

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA**

**Laima Buškauskaitė**

**Korporatyvinės įmonės duomenų saugyklos  
modelio sudarymas ir tyrimas**

Magistro darbas

**Vadovas  
doc. dr. A. Venčkauskas**

**KAUNAS, 2007**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA**

**TVIRTINU  
Katedros vedėjas  
prof. dr. E. Bareiša  
2007-05-28**

**Korporatyvinės įmonės duomenų saugyklos  
modelio sudarymas ir tyrimas**

Informatikos mokslo magistro baigiamasis darbas

**Recenzentas  
doc.dr. E. Toldinas  
2007-05-28**

**Vadovas  
doc. dr. A. Venčkauskas  
2007-05-28**

**Atliko  
IFM 1/2 gr. stud. L. Buškauskaitė  
2007-05-28**

**KAUNAS, 2007**

## SUMMARY

Business use a big amount of data in our days, that data become just ballast if we incapacity to sift it. Just data mining using special software, transform data in to information. That information become reason of correct business rule. OLAP software make data warehouse, that help analyse data quick and convenient, even data are from diferent environment. OLAP technology uses measures and tools to transform and store information, create and execute queries, and generate graphic user interface. Because OLAP systems are becoming more affordable companies have to face the issue of how to choose the best product and then design and implement OLAP systems according to their business requirements. This research aims at comparing different OLAP systems at functional and data structure level in order to design and implement an OLAP system.

What is OLAP? OLAP is On-Line Analitical Procesing. This term use to describe software products, thats have possibility to analyse information in real time. OLAP systems are one from a lot of Business Intelligence group products. In the worl are a lot products which are the part of this group: from MS SQL OLAP cubes to „Business Objects“, „Cognos“, Corporate Planner“ and „Microstrategy“ systems.

The purpose of this work is to create and analyze the OLAP data warehouse models of large corporations.

# TURINYS

<b>1. DUOMENŲ SAUGYKLOS MODELIŲ ANALIZĖ.....</b>	<b>9</b>
1.1. TYRIMO SRITIS, OBJEKTAS IR TIKSLAI.....	9
1.1.1. Tyrimo sritis bei darbo objektas.....	9
1.1.2. Darbo tikslas bei uždaviniai.....	9
1.2. ĖGZITUOJANČIOS PROBLEMOS IR GALIMŲ SPRENDIMŲ APŽVALGA.....	10
1.2.1. „Cognos“.....	11
1.2.2. „Business object“.....	11
1.2.3. OLAP ir OLTP duomenų saugyklų galimybės.....	13
1.3. OLAP DUOMENŲ SAUGYKLOS FUNKCIONALUMO APŽVALGA.....	14
1.3.1. Pjūviai.....	14
1.3.2. Skaičiuojamos reikšmės.....	15
1.3.3. Duomenų gavyba (data mining).....	18
1.4. DUOMENŲ SAUGYKLOS ARCHITEKTŪRA.....	21
1.5. OLAP KUBŲ ARCHITEKTŪROS MODELIAI.....	23
1.6. OLAP ĮRANKIŲ ANALIZĖ.....	25
1.6.1. MS SQL Server 2000 Analysis Services apžvalga.....	25
1.6.2. Oracle Express apžvalga.....	26
1.6.3. OLAP sistemų palyginimas.....	27
1.6.4. Išvados.....	28
1.7. KOMPIUTERIZUOJAMOS SISTEMOS VARIANTO PARINKIMAS.....	28
2.1. REIKALAVIMŲ MODELIS.....	29
2.1.1. Sistemos panaudojimo atvejų modelis.....	29
2.1.2. Nefunkciniai reikalavimai.....	30
2.1.3. Reikalavimai sistemos funkcionalumui palaikyti.....	31
2.2. LOGINĖ OLAP SISTEMOS ARCHITEKTŪRA.....	31
Vartotojo sąsajos komponentas.....	32
Veiklos komponentas.....	33
DB komponentas.....	33
2.3. PANAUDOJIMO ATVEJŲ SEKŲ DIAGRAMOS.....	33
2.4. DUOMENŲ STRUKTŪROS MODELIS.....	37
2.5. TESTAVIMO MODELIS BEI TESTAVIMO PROCEDŪROS.....	39
2.5.1. Vienetų testavimas.....	39
2.5.2. Integravimo testavimas.....	40
2.5.3. Priėmimo testavimas.....	41
2.5.4. Naudotojo sąsajos testavimas.....	42
2.5.5. Testavimo rezultatai.....	42
2.5.6. Išvados.....	43
2.6. REALIZACIJOS MODELIS.....	43
2.7. KOKYBĖS TYRIMAS.....	44
2.7.1. Realiai atlikto darbo kokybės analizės tikslai.....	44
2.7.2. Kokybės vertinimo procesas.....	45
2.7.3. Išvados.....	46
<b>3. OLAP KUBŲ ARCHITEKTŪROS EKSPERIMENTINIS TYRIMAS.....</b>	<b>48</b>
3.1. „ŽVAIGŽDĖS“ IR „SNAIGĖS“ TIPO DUOMENŲ SCHEMAS.....	49
3.2. MOLAP ARCHITEKTŪRA.....	50
3.2.1. MOLAP pranašumai ir trūkumai.....	52
3.3. ROLAP ARCHITEKTŪRA.....	52
3.3.1. ROLAP pranašumai ir trūkumai.....	54
3.4. HOLAP ARCHITEKTŪRA.....	55
3.5. IŠVADOS.....	56
<b>IŠVADOS.....</b>	<b>57</b>
<b>LITERATŪRA.....</b>	<b>59</b>
PANAUDOJIMO ATVEJŲ SĄRAŠAS.....	69
DB LENTELIŲ APRAŠAI.....	74
XML FAILŲ ŠABLONAI.....	76

TESTAVIMO PROCEDŪROS.....	77
OLAP DUOMENŲ SAUGYKLOS VERTINIMO ANKETA.....	83

## Lentelių sąrašas

1 LENTELĖ DUOMENŲ PAVYZDYS.....	16
2 LENTELĖ REIKALAVIMAI SERVERIUI.....	31
3 LENTELĖ REIKALAVIMAI VARTOTOJO KOMPIUTERIUI.....	31
4 LENTELĖ. TESTINIŲ „BALTOS“ DĖŽĖS METODO RINKINIŲ FORMA.....	40
5 LENTELĖ. TESTINIŲ „JUODOS“ DĖŽĖS METODO RINKINIŲ FORMA.....	40
6 LENTELĖ. SISTEMOS VEIKIMO TESTAVIMO TESTINIŲ DUOMENŲ RINKINIŲ FORMA..	40
7 LENTELĖ. FUNKCINIO TESTAVIMO TESTINIŲ ATVEJŲ RINKINIŲ FORMA .....	41
8 LENTELĖ SAUGUMO TESTAVIMO TESTINIŲ DUOMENŲ RINKINIŲ FORMA .....	41
9 LENTELĖ. PRIĖMIMO TESTAVIMO TESTINIŲ DUOMENŲ RINKINIŲ FORMA .....	41
10 LENTELĖ. NAUDOTOJO SĄSAJOS TESTINIŲ DUOMENŲ RINKINIŲ FORMA .....	42
11 LENTELĖ. PROJEKTO KOKYBĖS VERTINIMO PARAMETRAI.....	46
12 LENTELĖ. PRICEDISKTABLE LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	74
13 LENTELĖ. FORECASTSALES LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	75
14 LENTELĖ. CUSTINVOICETRANS LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	75
15 LENTELĖ. CUSTINVOICEJOUR LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	75
16 LENTELĖ. CUSTTABLE LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	75
17 LENTELĖ. MARKUPTRANS LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	75
18 LENTELĖ. INVENTDIM LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	76
19 LENTELĖ. COUNTRY LENTELĖS DUOMENŲ LAUKŲ APRAŠAS.....	76
20 LENTELĖ. SA_TIMEDIM KOMPONENTO TESTINIAI DUOMENŲ RINKINIAI.....	77
21 LENTELĖ. SA_CLIENTDIM KOMPONENTO TESTINIAI DUOMENŲ RINKINIAI.....	78
22 LENTELĖ. SA_FACT KOMPONENTO TESTINIAI DUOMENŲ RINKINIAI.....	78
23 LENTELĖ. „JUODOS DĖŽĖS“ METODO TESTINIAI DUOMENŲ RINKINIAI.....	80
24 LENTELĖ. VEIKIMO TESTAVIMO TESTINIAI DUOMENŲ RINKINIAI.....	80
25 LENTELĖ. FUNKCINIO TESTAVIMO TESTINIAI DUOMENŲ RINKINIAI.....	80
26 LENTELĖ. SAUGUMO TESTAVIMO TESTINIAI DUOMENŲ RINKINIAI.....	81
27 LENTELĖ. PRIĖMIMO TESTAVIMO TESTINIAI DUOMENŲ RINKINIAI.....	81
28 LENTELĖ. NAUDOTOJO SĄSAJOS TESTAVIMO TESTINIAI DUOMENŲ RINKINIAI.....	81

## **Paveikslėlių sąrašas**

1.

## ĮVADAS

Šiuolaikinis verslas naudoja didžiulį duomenų kiekį, tačiau įmonėje šie duomenys taip ir liks tik balastas, jeigu nesugebėsime jų išanalizuoti ir tinkamai interpretuoti. Tik duomenų analizė naudojant specialius programinius įrankius iš „žalios“ informacijos leis atrinkti naudingus grūdelius ir perdirbti juos į vertingas žinias, kuriuos taps teisingų verslo sprendimų pagrindu [1].

Nuolatos yra reikalinga tiksli ir vieninga verslo informacija, kurios pagalba būtų galima užtikrinti procesų efektyvumą, optimizuoti pinigų srautus ir sumažinti kaštus. Tad pagrindinė priemonė atremti visus rinkos iššūkius ir priimti efektyvius strateginius sprendimus yra disponavimas pagrindiniu turtu šiuolaikinėje ekonomikoje – informacija. Todėl informacijos saugojimas, užtikrinimas priėjimą prie jos reikiamu metu ir reikiamoje vietoje, taip pat patogi jau susistemintos informacijos analizė yra pagrindiniai uždaviniai, kurie įgyvendinami, tik išsirenkant tinkamą duomenų saugyklą, **bei tinkamas priemones informacijos išgavimui ir analizei.**

Šiame darbe bus analizuojami duomenų saugyklų modeliai, vykdoma jų charakteristikų apžvalga, apžvelgiami duomenų saugyklos kūrimo įrankiai, atliekamas architektūrų ir duomenų schemų palyginimas. Taip pat šiame darbe pateikiamas OLAP duomenų saugyklos modelis, jo realizacija bei atliktas sukurtosios sistemos tyrimas.



# 1. DUOMENŲ SAUGYKLOS MODELIŲ ANALIZĖ

## 1.1. Tyrimo sritis, objektas ir tikslai

### 1.1.1. Tyrimo sritis bei darbo objektas

Pradedant darbo analizę labai svarbu apibrėžti tyrimo sritį bei išsamiai ją iširti nusakant objektą, kuris bus suprojektuotas ar sukurtas kaip darbo rezultatas. Šiame darbe tyrimų sritis bus duomenų saugyklos modelio sudarymas ir jo realizacija.

Dauguma įmonės darbuotojų sugaišta daug laiko analizuodami darbo kokybę, skaičiuodami ir lygindami rodiklius ar tiesiog ieškant suvestos informacijos. Į šį procesą įtraukiama daug specialistų. Kartais tam naudojamos elektroninės lentelės (MS Excel), tačiau jos nėra patogios, nes duomenis reikia nuolatos atnaujinti, bei jos nėra universalios vykdant duomenų palyginimą.

Todėl ir buvo sukurti specializuoti verslo analitikos (Business Intelligence) produktai, skirti detaliam, išsamiam, greitai duomenų analizei, verslo modeliavimui bei planavimui, išsiskiriantys naudojimo paprastumu ir lankstumu – įmonių darbuotojai be informacinių technologijų specialistų pagalbos gali greitai ir lengvai modeliuoti įmonės pajamų ir išlaidų struktūrą, pinigų srautus, balanso straipsnius ir t.t. Pasaulyje yra sukurta tikrai nemažai sistemų, priklausančių šiai produktų grupei. Tai galima paminėti nuo paprasčiausių MS SQL OLAP kubų, iki tokių sistemų kaip „Business Objects“, „Cognos“, „Corporate Planner“, „Microstrategy“. Šių produktų pagalba galima modeliuoti įmonės pajamų ir kaštų struktūrą, sudaryti analitines ataskaitas, planuoti ir prognozuoti atsivėlgiant į praeities duomenis, nuolatos stebėti fakto nuokrypius nuo biudžeto.

Naudojant OLAP (On-line Analytical Processing) priemones, duomenų bazės duomenis galima greitai analizuoti pageidaujama pjūviais – iš reikiamų duomenų formuojami „duomenų kubai“. OLAP kubai sudaromi panaudojant duomenų pjūvius (pvz., padaliniai, projektai, darbuotojai, klientai...), lygius (pvz., metai, periodas...) ir rodiklius (pvz., pajamos, sąnaudos, pelnas...). Duomenų kubus galima peržiūrėti ir analizuoti MS Excel formato faile, o pateikiamą informaciją galima atnaujinti automatiškai, kadangi sudaroma kreiptis į duomenų bazę.

### 1.1.2. Darbo tikslas bei uždaviniai

Šio darbo tikslas – sudaryti korporatyvinės įmonės duomenų saugyklos, kurioje būtų saugomi duomenys iš nutolusių įmonės padalinių, bei kuri užtikrintų greitą bei patogią duomenų analizę vienoje alinkoje, modelį.

### **Sprendžiami uždaviniai:**

1. Apibrėžti ir išanalizuoti probleminę sritį.
2. Ištirti užsakovo reikalavimus būsimai sistemai.
3. Apibrėžti tyrimų sritį ir atlikti išsamų tyrimą toje srityje.
4. Suprojektuoti duomenų saugyklos modelį.
5. Realizuoti suprojektuotą modelį.
6. Atlikti OLAP duomenų saugyklos eksperimentinį tyrimą.

### **1.2. Egzistuojančios problemos ir galimų sprendimų apžvalga**

Šiandien daugumoje įmonių rasime įdiegtą verslo valdymo ar finansinę sistemą, kuri dažnai yra papildoma ir kitomis IT sistemomis: gamybos valdymo, sandėlių valdymo, santykių su klientais valdymo, personalo valdymo ar kt. Visa įmonės veiklos ir verslo analizė remiasi į aukščiau paminėtų sistemų ataskaitų posistemėmis, t.y. ataskaitų rinkiniais, kurias siūlo įdiegtos įvairios IT sistemos.

Iš čia kyla visa eilė kasdieninių problemų, kurias tenka sutikti dažnoje įmonėje:

- Ataskaitų ribotumas – jei nėra sukurtos ir įdiegtos reikalingos bei kompleksinės ataskaitos, įmonės analizė norimu pjūviu tampa sunkiai įvykdoma arba iš viso negalima.
- Naujos ataskaitos kuriamos labai ilgai, jei iš viso yra sukuriamos. Taip pat kiekvieną kartą tenka mokėti sistemos diegėjams ar prižiūrėtojams už naujų ataskaitų kūrimą ar jau esamų modifikavimą.
- Ataskaitų generavimas kasdien ilgėjantis procesas, kadangi duomenų kiekis sparčiai didėja. Neretai dienomis jau negalima daryti ataskaitų, nes tai dažnai stabdo kitų sistemų darbą.
- Dėl didelio duomenų kiekio įmonė ir jos verslas yra analizuojamas tik makro-lygiu, nes labiau detalizuoti informacijos neleidžia turimos technologijos, gaminių ir kitų analizės objektų nomenklatūra.
- Vartotojai, kuriems reikalingi įmonės duomenys, juos apdoroja įvairiausiomis priemonėmis. Todėl būtinai atsiranda klaidų, nesutapimų, skirtingo vienodų objektų traktavimo.
- Keičiant turimą finansinę sistemą kita ar apjungiant skirtingas sistemas, yra prarandama dalis įmonės istorinių duomenų bei galimybė analizuoti verslą,

nes visų duomenų perkėlimas į naują finansinę sistemą – sudėtingas uždavinys.

- Turint kelias IT sistemas nėra matomas vieningas įmonės vaizdas.
- Nutolę padaliniai neturi galimybės interaktyviai analizuoti norimus duomenis.
- Nepakanka arba visiškai nėra duomenų saugumo.

Norint išspėsti šias problemas, yra reikalinga verslo analizės sistema, kuri pirmiausia apima funkcionalų duomenų ir informacijos analizės bei vizualizavimo įrankį bei technologines priemones (duomenų saugykla), suteikiančias galimybę itin sparčiai analizuoti didelius duomenų kiekius, t.y. visą įmonę, įmonių grupę, visus įmonės procesus net iki detaliausio lygmens.

Pasaulyje yra sukurta daug įrankių, kurie apima duomenų ir informacijos analizę, bei technologines priemones - duomenų saugykla.

### **1.2.1. „Cognos“**

Cognos Inc. ([www.cognos.com](http://www.cognos.com)) siūlo įmonėms pilną paketą priemonių skirtų įmonės verslo planavimui, veiklos rodiklių stebėjimui, duomenų analizei ir atvaizdavimui, paprastų ir sudėtingų ataskaitų kūrimui, realaus laiko užklausoms ir kt.

Pagrindiniai Cognos šeimos produktai, skirti analizei bei ataskaitinei informacijai pateikti:

- Cognos ReportNet – ataskaitų kūrimo, generavimo ir pateikimo vartotojams platforma. Peržiūra, analizė ir kūrimas vyksta internetinėje aplinkoje su naršykle.
- PowerPlay – OLAP analizės įrankis, leidžiantis interaktyviai atlikti daugiamatę analizę. Abu šie produktai veikia per internetą ir nereikalauja papildomų komponentų diegimo vartotojų kompiuteriuose.

Cognos produktai yra skirti vidutinėms ir didelėms organizacijoms. Jie gali būti naudojami įvairiose srityse: bankininkystėje, vertybinių popierių, draudimo ir kitose finansinėse kompanijose, prekyboje, gamyboje, valstybiniame sektoriuje bei kt. Cognos sprendimais naudojami >135'000 klientų visame pasaulyje.

### **1.2.2. „Business object“**

Business Objects ([www.dbit.lt](http://www.dbit.lt); [www.businessobjects.com](http://www.businessobjects.com)) - pasaulyje pirmaujantis duomenų analizės ir informacijos vizualizavimo priemonių gamintojas. Business Objects veiklą vykdo 80-tyje šalių bei turi daugiau nei 20000 klientų visame pasaulyje.

Business Objects Enterprise Professional (toliau – BOE Pro) darbo vietą sudaro priemonių rinkinys, kurių pagalba yra sukuriama pilna infrastruktūra informacijos analizei:

- Vartotojų analitinės informacijos portalas – Infoview
- Automatinis ataskaitų apdorojimo ir atnaujinimo įrankis - Broadcast Agent Scheduler
- Administravimo bei programavimo priemonės – Designer (semantiniam lygmeniui) bei Supervisor (vartotojų ir informacijos saugumui)
- Integracijos priemonės, RDBMS jungtys

Vartotojai, įsigiję BOE Pro, gali naudotis analitikos portalu (Infoview), kurį gali personalizuoti pagal savo poreikius, taip pat turi teisę naudotis Business Objects repozitoriumu, iš kurio pasiėmę ataskaitas galės jas atnaujinti – užpildyti naujausiais duomenimis (refresh funkcija). BOE Pro vartotojams galioja visas Business Objects saugumas, jie taip pat gali pasinaudoti Broadcast Agent Scheduler funkcijomis.

Business Objects Enterprise Premium (toliau – BOE Premium) darbo vieta papildo BOE Pro galimybes naujomis funkcijomis, kurios, savo ruožtu, gali būti išigyjamos atskirai po vieną:

- Visos duomenų ir informacijos analizės galimybės – Explorer
- Business Objects instaliacijos apkrovimo, ataskaitų vartojimo, vartotojų veiksmų stebėjimo ir analizavimo priemonė – Auditor
- OLAP'inės jungtys

Explorer apima detalizavimo (Drill), filtravimo, rūšiavimo, sumavimo, procentinės išraiškos pateikimo, palyginimo procentais bei skirtumo išskaičiavimo, Top'ų išrinkimo, sąlygų ir išryškinių, matematinių funkcijų skaičiavimo ir kitas analitines funkcijas. Vartotojui suteikiamos galimybės informaciją ir duomenis pateikti lentelėje, dvimatėje lentelėje, grafike, nesudėtingai atlikti konvertavimą tarp šių formatų, pasirinkti standartinius ir vartotojų redaguojamus šablonus.

Business Objects saugykla suteikia kompanijoms konsoliduotus duomenis, kurie yra optimizuoti ataskaitų kūrimui ir analizei.

Tačiau įsigyti šias sistemas nebūtų tikslinga, nes egzistuojančią problemą galima išspręsti neperkant naujos programinės įrangos, o panaudojant „Microsoft SQL Server“ įrankį „Analysis Services“.

### 1.2.3. OLAP ir OLTP duomenų saugyklų galimybės

Duomenų saugykla – tai centralizuota duomenų bazė, surenkanti, integruojanti duomenis iš įvairių įmonės verslo procesų dalių bei duomenų bazių [2]. Duomenų saugykla yra vienas iš esminių faktorių lemiančių sėkmingą įmonės veiklą. Nes tik tinkamu laiku ir tinkamoje vietoje gauta informacija gali padėti nulemti esminius įmonės klausimus.

#### OLTP duomenų saugyklos

Kai duomenų yra nedaug, jų analizę galima atlikti pasitelkus ir tradicines reliacines duomenų bazių valdymo sistemas (RDBVS), kurios kartu naudojamos ir transakcijoms realiuoju laiku vykdyti (online transaction processing, OLTP). OLTP duomenų bazės pritaikytos tam, kad galėtų palaikyti daug vienu metu į jas besikreipiančių vartotojų, kurie intensyviai įterpia bei keičia duomenis. Paprastai tokios duomenų bazės turi sudėtingą struktūrą (didelė normalizacija), o duomenys jose parodo esamą įmonės situaciją, tačiau ne istoriją. Jose esančių duomenų analizė dažniausiai remiasi įprastų užklausų, parašytų SQL kalba, vykdymu.

Tačiau kai duomenų susikaupia daug, ir organizacija mėgina atlikti strateginę analizę naudodama tą pačią OLTP duomenų bazę, kaip ir einamojo duomenų apdorojimo atveju, dažnai susiduriama su keliomis problemomis:

- Didelių skaičiavimo pajėgumų reikalaujančios intensyviai vykdomos analitinės užklausos neigiamai veikia darbinės sistemos produktyvumą.
- Jeigu darbinėje duomenų bazėje saugoma istorinė informacija, nereikalinga kasdienėms operacijoms atlikti, tokios sistemos produktyvumas mažėja, nes duomenų laikui bėgant vis daugėja.

Aišku, istoriniai duomenys (pavyzdžiui, duomenys apie praėjusių metų pardavimus) atspindi organizacijos verslo istoriją ir yra labai svarbūs šio verslo analizei. Jais remiamasi priimant verslo sprendimus - ir strategiškai planuojant, ir vertinant konkrečių padalinių veiklą visos organizacijos laimėjimų kontekste. Siekiant išvengti minėtų problemų ir išsaugoti istorinius duomenis kuriamos atskiros duomenų saugyklos (Data Warehouses), kuriose telpa jau istorine tapusi informacija iš darbinės duomenų bazės. Istorinių duomenų saugojimą ir analizę perkėlus į duomenų saugyklas, išvengiama padidėjusių darbinių duomenų bazių serverių apkrovų. Tačiau sukūrus duomenų saugyklą išsprendžiama tik dalis problemų, nes:

- Duomenų analitikai dažnai neturi reikiamo išsilavinimo, kad galėtų savarankiškai parašyti ir vykdyti įvairias užklausas OLTP duomenų bazėje.

- Atsakant į sudėtingas ir didelių duomenų kiekio apdorojimo reikalaujančias užklausas, tokio tipo užduotims nepritaikytos OLTP sistemos pajėgumas gali būti per lėtas apibendrintų duomenų peržiūrai ir analizei.
- Vieną kartą jau atliktais skaičiavimais negali pasinaudoti kiti vartotojai.

## **OLAP duomenų saugyklos**

Daugelį aukščiau minėtų problemų gali išspręsti OLAP duomenų saugyklos. Tokiu atveju darbinėse reliacinėse duomenų bazėse ar duomenų saugyklose saugomi pradiniai duomenys transformuojami, ir sukuriama optimizuota duomenų saugojimo struktūra - OLAP duomenų kubai (data cubes), kurie specialiai pritaikyti greitai duomenų analizei .

OLAP kubuose didelė tarpinių skaičiavimų dalis atliekama dar iki duomenų kubo naudojimo, o vieną kartą atliktų tarpinių skaičiavimų rezultatais gali pasinaudoti visi prieigos teises turintys šios duomenų bazės vartotojai. Kadangi OLAP duomenų bazės saugo tarpines agregatines reikšmes, pagal duomenų kubo įgyvendinimo būdą, duomenų pasikeitimai ar naujų duomenų įkėlimas į OLAP duomenų bazę gali pareikalauti atnaujinti ar pertvarkyti kubo informaciją (turi būti perskaičiuotos kai kurios agregatinės reikšmės).

### **1.3. OLAP duomenų saugyklos funkcionalumo apžvalga**

#### **1.3.1. Pjūviai**

Aprašant OLAP duomenų struktūras (kubus) naudojama pjūvių sąvoka (dimensions). OLAP duomenų bazėje esantys įvykiai (eilutė lentelėje) gali būti analizuojami pagal įvairius parametrus - įvairiais pjūviais (dimensijomis). Naudodami pardavimų modelio pavyzdį, paaiškinsim pjūvio sąvoką.

Pvz. Užregistruotas prekių pardavimo įvykis apibūdinamas keliais parametrais (kada tai įvyko, koks produktas parduotas, kokioje parduotuvėje), todėl ir įmonės mastu pardavimus galima analizuoti įvairiais pjūviais:

- Pagal laiką. Galima sužinoti kaip sekėsi prekyba šį mėnesį, palyginti su praėjusiu, su praėjusiais metais ir pan.
- Pagal produktą. Parduodamos įvairios produktų grupės, jų tipai ir pagaliau - konkretūs produktai. Galima sužinoti, kokią pajamų dalį sugeneravo ne tik konkretus produktas, bet ir viena ar kita produktų grupė, koks perkamiausias produktas ir pan.
- Pagal vietovę. Didelės įmonės pardavimus vykdo didelėje teritorijoje, todėl konkrečios parduotuvės gali būti priskirtos prie apskričių, rajonų. Rajonuose gali būti numatyti mažesni struktūriniai vienetai, pavyzdžiui, miestai ir pan. Tokiu

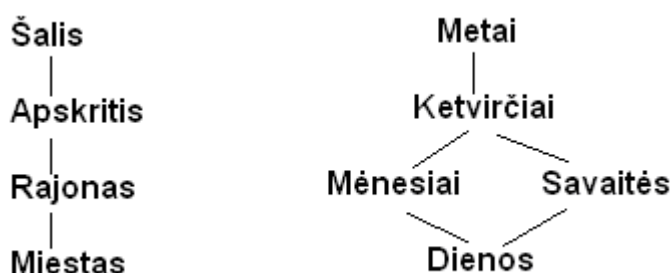
būdu galima atsakyti, kaip sekėsi parduoti tą ar kitą prekę apskrityje, rajone ar mieste.

Šiuo atveju sukūrėme trimatį duomenų kubą, kurio matmenis apibrėžia laiko, produkto ir vietovės dimensijos. Tačiau loginė duomenų kubo struktūra gali būti ne tik trimatė, bet ir daugiamatė (keturmatė, penkiamatė ir t. t.).

Kaip matome iš pavyzdžio, mūsų pjūviai gali turėti tam tikrą hierarchinę struktūrą, pavyzdžiui, vietovės ir laiko pjūviai turi net po tris hierarchinius lygius:

- apskritis > rajonas > miestas
- metai > ketvirtis > mėnuo

Tokia hierarchinė struktūra gali būti ir labai sudėtinga.



1 pav. Hierarchinės dimensijos

### 1.3.2. Skaičiuojamos reikšmės

Kita pagrindinė OLAP duomenų bazių sąvoka yra skaičiuojamoji vertė (**measure**). Galima pasirinkti, kokią su konkrečiu įvykiu susijusią skaitinę reikšmę analizuosime. Pateikiamame pavyzdyje galėtume analizuoti ne tik, pavyzdžiui, pajamas iš šio miesto per mėnesį parduoto konkretaus produkto, bet ir jo pardavimo pelną ar parduotų prekių vienetų skaičių.

Sukūrus duomenų kubą, OLAP duomenų bazės leis lengvai atlikti įvairius veiksmus:

- Perėjimą nuo bendresnių reikšmių iki individualių ir atvirkščiai, t. y. judėti hierarchine pjūvių (dimensijų) struktūra. Žinodami kurio nors produkto pardavimų pokytį per mėnesį visoje apskrityje, esant poreikiui galime pažiūrėti išsamiau - kaip produkto pardavimai kito skirtinguose šios apskrities rajonuose ar net atskirose parduotuvėse.

- Duomenų apibendrinimą pagal įvairius parametrus, t. y. analizę įvairiais pjūviais. Pavyzdžiui, ataskaitas apie pelną pardavus tam tikrą produktą galime analizuoti ne tik laike, bet pagal vietovę.
- Analizuojamojo matmens pakeitimą, t. y. analizę pagal skirtingas skaičiuojamas reikšmes. Vietoj pardavimų analizės pagal pajamas tuos pačius pardavimus toje pačioje vietovėje ir tuo pačiu metu galime analizuoti pagal jų sugeneruotą pelną.

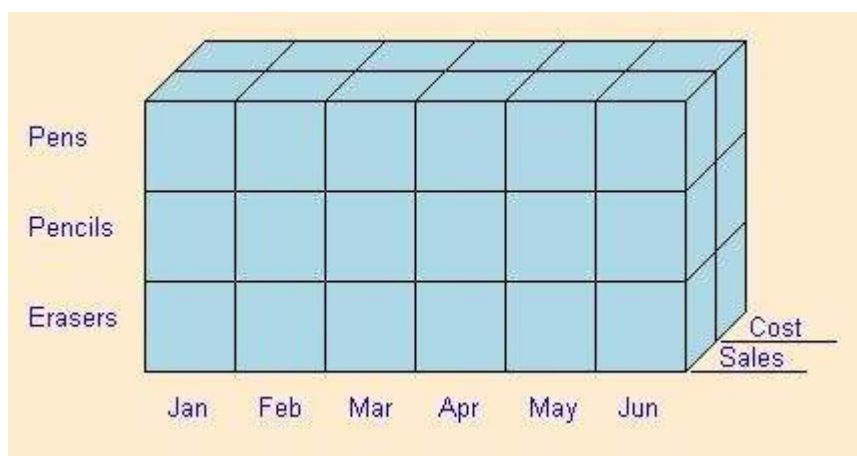
Turėdami kube saugomus duomenis, galėtume lengvai atsakyti į klausimą, kiek pajamų (skaičiuojamoji reikšmė) praėjusį pusmetį (laiko pjūvis) buvo gauta Vilniaus apskrities parduotuvėse (vietovės pjūvis) padažų produktų grupei (produktų pjūvis) iš visų pirkėjų.

Tarkime turime lentelę:

**1 lentelė Duomenų pavyzdys**

Product	Sales	Cost
Pens	250	191
Pencils	322	217
Erasers	108	83

Pagal pateiktus duomenis galime sudaryti kubą:



**2 pav. Duomenų kubas pagal duomenis patektus 1 lentelėje**

Duomenų kubą (2 pav.) galima analizuoti įvairiais pjūviais. Galima atlikti parduotų prekių analizę pagal mėnesius. Galima atsakyti į klausimą: kiek ir kokių prekių buvo parduota sausio mėnesį ir už kokią kainą?

Sausio mėnesio visų prekių pardavimai ir kainos pagal prekės kodą:





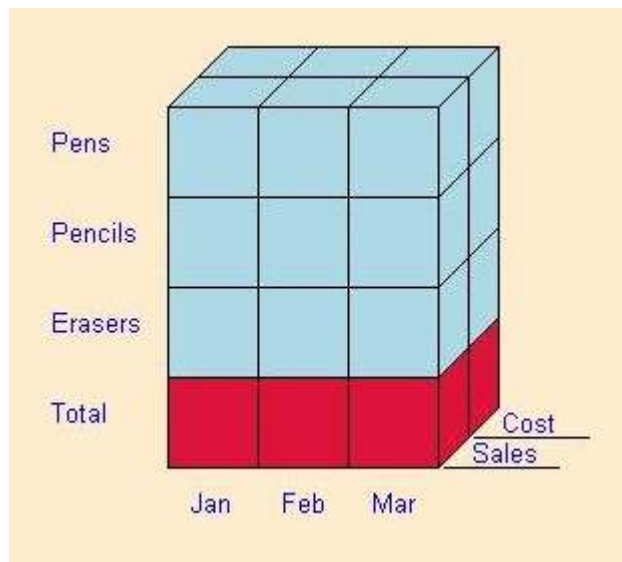
**3 pav. Prekių pardavimai ir kainos mėnesiais pagal prekių tipus**

Taip pat galima atlikti tik pardavimų analizę. Kiek kokių prekių buvo parduota koki mėnesį:



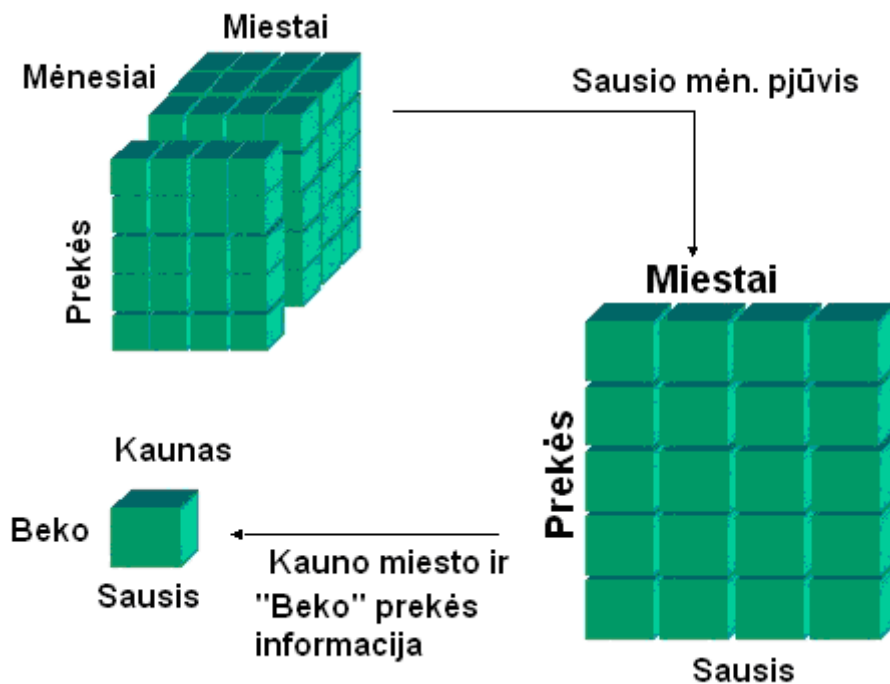
**4 pav. Pardavimai pagal prekių tipus ir mėnesius**

Galima analizuoti bendrus pardavimus ir kainas pagal mėnesius. Kiek buvo įvykdyta pardavimų ir už kokią sumą kiekvieną mėnesį.



5 pav. Kainos ir pardavimai pagal mėnesius

Kiekvienam vartotojui dažniausia reikalinga dar labiau konkretesnė analizė. Tarkim vartotojui yra įdomi tik jo parduodama prekė, o ne bendra visų prekių statistika, todėl jis analizuoja dar labiau susmulkintą informaciją:



6 pav. Vienos prekės pardavimų informacija viename mieste sausio mėnesį

### 1.3.3. Duomenų gavyba (data mining)

Duomenų kaupimas jų neapdorojus neturi prasmės, todėl duomenų analizė buvo atliekama visais laikais. Visų rūšių duomenų tvarkymas (kartotekos, rūšiavimas ir t.t.) yra pirmas žingsnis siekiant palengvinti duomenų analizę, apdoroti didelį informacijos srautą.

Duomenų gavyba – tai prasmingų dėsningumų, modelių ir tendencijų radimo procesas dideliuose informacijos kiekiuose, naudojant modelių atpažinimo, statistinius bei matematinius metodus [3].

Jei įprasti duomenų analizės metodai parodo norimų kintamųjų priklausomumą, tai duomenų gavyba unikali tuo, kad analizės rezultatas yra naujų priklausomybių, kurios neegzistavo, radimas.

Duomenų gavyba turi vieną savybę, išskiriančią ją iš kitų procesų ar operacijų, susijusių su duomenų bazėmis. Jei įprastos ataskaitos bei įvairios analitinės priemonės atsako į klausimus "kas yra?" ar "kas bus, jeigu?", tai duomenų gavybos rezultatas yra atsakymas į nepateiktus klausimus, t.y. naujų priklausomybių radimas. Tyrimai rodo, kad dažniausiai duomenų gavyba pasiteisina gerą finansinę padėtį turinčiose kompanijose. Tuomet sudaromos galimybės pagerinti veiklą arba nustatyti nedidelius nesklandumus[3].

Duomenų gavyba yra labai plati sritis, todėl yra ir daug jai skirtų metodų, algoritmų bei taikomųjų sistemų:

- **Asociacijų paieška** – tai dėsningumų analizė tam tikrose reiškinių ar daiktų grupėse. Pavyzdys — pirkimo analizė. Nagrinėjama, kokios prekės perkamos kartu, kokia tikimybė, kad bus būtent toks derinys ir t.t. Tokio tipo uždaviniai gali būti sprendžiami rengiant reklamos kampanijas, kuriant nuolaidų sistemas.
- **Eiliškumo analizė** – tai dėsningumų paieška atsižvelgiant į laiką. Šiuo atveju svarbu ne tik tai, kokiomis paslaugomis naudojasi klientas, bet ir kokia eilės tvarka. Šis metodas padeda efektyviau teikti paslaugas.
- **Grupavimas (klasterizavimas)** dažnai būna vienas pirmųjų duomenų gavybos žingsnių. Tai visos duomenų aibės suskaidymas į poaibius pagal skiriamuosius bruožus. Tai ir rinkos ar klientų segmentavimas, ir nekilnojamojo turto grupavimas pagal būdingus duomenis, ir daugelis kitų uždavinių.
- **Klasifikavimas** dažnai atliekamas po grupavimo. Kai nagrinėjama aibė jau padalyta į pogrupius, dažnai kyla klausimas, kam priskirti naujus elementus. Grupavimu anksčiau neįvardyti poaibiai išskiriami iš duomenų visumos, o klasifikuojant sprendžiama, kaip sudėti elementus į žinomas grupes.
- Po klasifikavimo atliekamas **įvertinimas**. Pavyzdžiui, finansinė institucija ne tik nori žinoti savo išduotų paskolų apibūdinimą "gera — bloga", bet ir jų įvertinimą.
- **Prognozavimas** taip pat labai svarbus duomenų gavybai. Atsižvelgiant į turimus duomenis bei pastebėtas tendencijas, bandomi prognozuoti ateities įvykiai.

Dažnai uždaviniui išspręsti taikomi keli metodai iš eilės ar net sudėtingi jų deriniai. Nė vienas jų nėra universalus ar nepriekaištingas. Vienų trūkumas — sudėtingumas, kitų - didelė modelių apimtis, daug sugaištama laiko [3].

### **1.3.3.1. Sprendimų medžiai (decision tree)[3]**

Šį algoritmą galima įsivaizduoti kaip struktūrą, panašią į medį. Kiekvienas išsišakojimas reiškia vieną ar kitą atsakymą į klausimą. Tokiu būdu sudaromos taisyklės, kurios nagrinėjamą duomenų aibę klasifikuoja atsižvelgdamos į elementų savybes. Proceso pradžioje turėta duomenų aibė skaidoma į šakas, kol kiekviena jų tampa homogeniška. Sprendimų medžių privalumas — jų aiškumas, jie suprantami tiek problema formuluojantiems, tiek bandantiems uždavinį realizuoti, tiek analizuojantiems gautus rezultatus.

Dažnai cituojamas sprendimo medžio pavyzdys - naudojantis istorine informacija apie banko klientus (pavyzdžiui, metinės pajamos, šeimyninė padėtis, išsilavinimas) ir tai, kaip jie anksčiau sugebėjo gražinti paimtus kreditus, nustatomi klientų patikimumo kriterijai. Algoritmas suranda priklausomybę tarp tam tikrų klientų apibūdinančių parametrų, taip pat nustato jų svarbą įvertinant kliento mokumą. Sprendimų medžio algoritmas leidžia numatyti kredito suteikimo riziką naujam klientui pagal jo pateiktą informaciją apie save.[3]

### **1.3.3.2. Grupavimas (Clustering)[3]**

Klasifikavimas, naudojant klasterinę analizę leidžia surasti įdomią ir neakivaizdžią informaciją. Klasifikavimo (klasterizavimo) algoritmas leidžia išskirti duomenų įrašų sutankėjimo grupes įvairiose dimensijose. Pavyzdžiui, turint informaciją apie pirkinių asortimentą ar užsakomas paslaugas, pagal elgseną esamus klientus galima suskirstyti į skirtingas grupes. Planuojant rinkodaros ir reklamines kampanijas dažnai svarbu žinoti tikslią vartotojų auditoriją ir numatyti, kokiais reklamos būdais tą ar kitą vartotojų grupę pasiekti, kokius produktus siūlyti ir pan. Gali būti, kad šis algoritmas išskirs vedusių, išsilavinusių žmonių su mažomis pajamomis grupę, grupę vienišų žmonių su didelėmis pajamomis ir pan. Naudojant tokio pobūdžio informaciją galima parengti mažai sąnaudų reikalaujančią tikslią rinkodaros kampaniją.

Pažymėtina, kad grupės nebūtinai identifikuojamos pagal visas dimensijas - gali paaiškėti, kad grupei kai kurių dimensijų reikšmės gali būti visiškai nesvarbios.

### 1.3.3.3. Neuroniniai tinklai [3]

Tai netiesiniai modeliai, sėkmingai naudojami klasifikavimo ir prognozavimo uždaviniams spręsti. Jų struktūra primena biologinius neuroninius tinklus. Dažnai dėl struktūros ir sudėtingumo neuroniniai tinklai vadinami "juodąja dėže".

Tinklas sudarytas iš kelių sluoksnių: įvesties, išvesties ir vieno ar daugiau paslėptų sluoksnių. Įvesties sluoksnyje pateikiama elementų aprašanti informacija, o išvesties sluoksnyje gaunamas rezultatas — priklausymas klasei (klasifikavimo uždaviniuose) ar numanomas rezultatas (prognozavimo uždaviniuose). Paslėpti sluoksniai realizuoja uždavinio logiką. Kiekvienas tinklo sluoksnio mazgas yra susijęs su kiekvienu šalia esančio sluoksnio mazgu. Nagrinėjami elemento duomenys turi didesnę ar mažesnę svorį štakos sprendimui, todėl šių svorių parinkimas ir yra pagrindinis algoritmo uždavinys.

### 1.3.3.4. Artimiausių kaimynų metodas [3]

Klasifikavimo uždavinius galima spręsti ne tik grupuojant elementus pagal nustatytas ar analizės metu rastas taisykles, bet ir pagal elemento panašumą į savo kaimynus. Šie algoritmai remiasi daikto, veiksmo ar kitaip apibrėžiamo nagrinėjamo vieneto lyginimu su prieš tai buvusiais. Analizė remiasi istorine informacija, todėl algoritmo trūkumas — sudaryto modelio dydis.

### 1.3.3.5. Taisyklių metodas [3]

Algoritmas remiasi taisyklių "jei — tai" taikymu. Panašiai kaip sprendimų medis, elementai grupuojami pagal jų duomenis.

Didelė metodų ir algoritmų įvairovė atskleidžia duomenų gavybos sudėtingumą ir leidžia šią technologiją pritaikyti įvairiose nagrinėjamose situacijose. Tenka nuvilti skeptikus, kurie teigė, jog ši sritis nesiplėtos ir išnyks.

Duomenų gavybos taikymas nesibaigia teisingo algoritmo parinkimu. Kaip ir daugelyje sričių, čia svarbūs visi etapai: problemos identifikavimas, kokybiški duomenys, kurių pagrindu bus atliekama analizė, modelio parinkimas ir realizavimas, gautų rezultatų interpretavimas.

## 1.4. Duomenų saugyklos architektūra

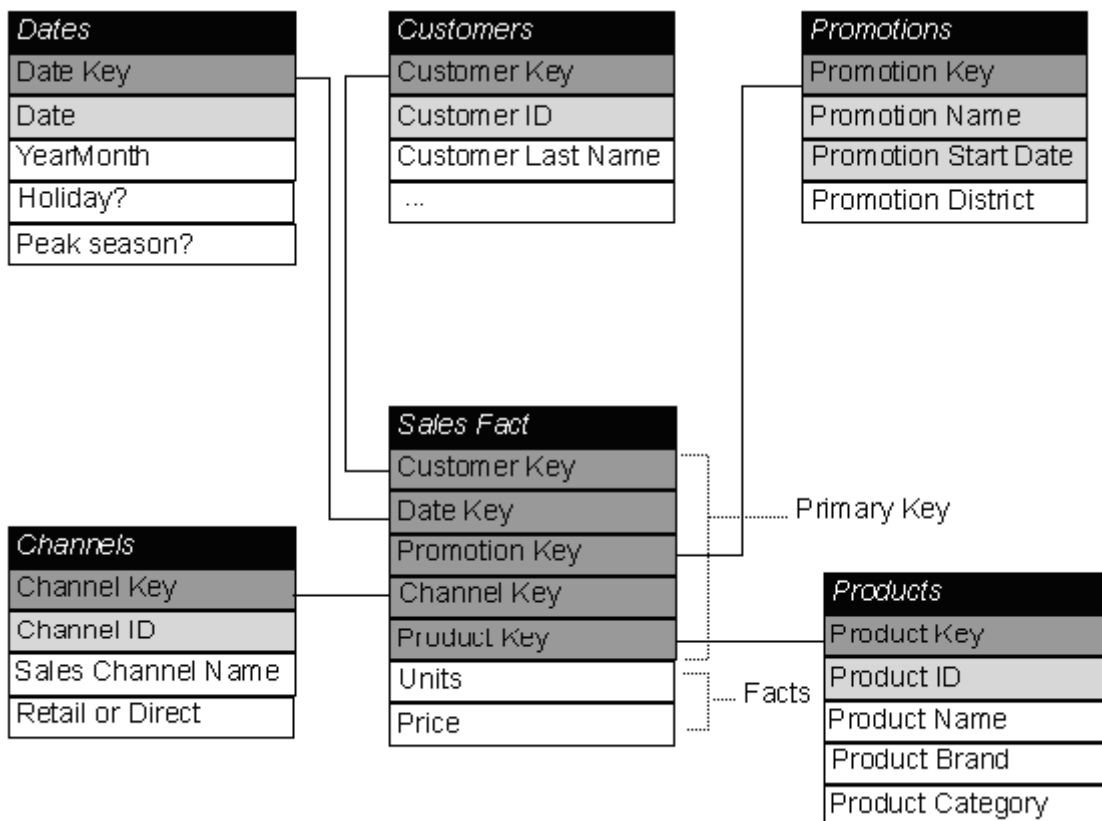
Pradiniai duomenys daugiamačiams kubams saugomi reliacinėse DB. Duomenys šiose DB buvo transformuoti į žvaigždės (star) arba sniegės (snowflake) schemą, kuri paprastai naudojama duomenų sandėliuose ir OLAP sistemose [26].

Pagrindinės struktūros modelio charakteristikos yra aibė smulkių verslo faktų, kurie susiję su daugialypėmis dimensijomis, kurios apibūdina tuos faktus. Realizuojamas saugyklos modelis turi pagrindinę faktų lentelę ir daug dimensijų lentelių. Toks struktūros modelis gali būti realizuojamas žvaigždės ir snaigės architektūra.

## Žvaigždės modelio architektūra

Žvaigždės schema - kai visos dimensijų lentelės tiesiogiai sujungtos su faktų lentele.

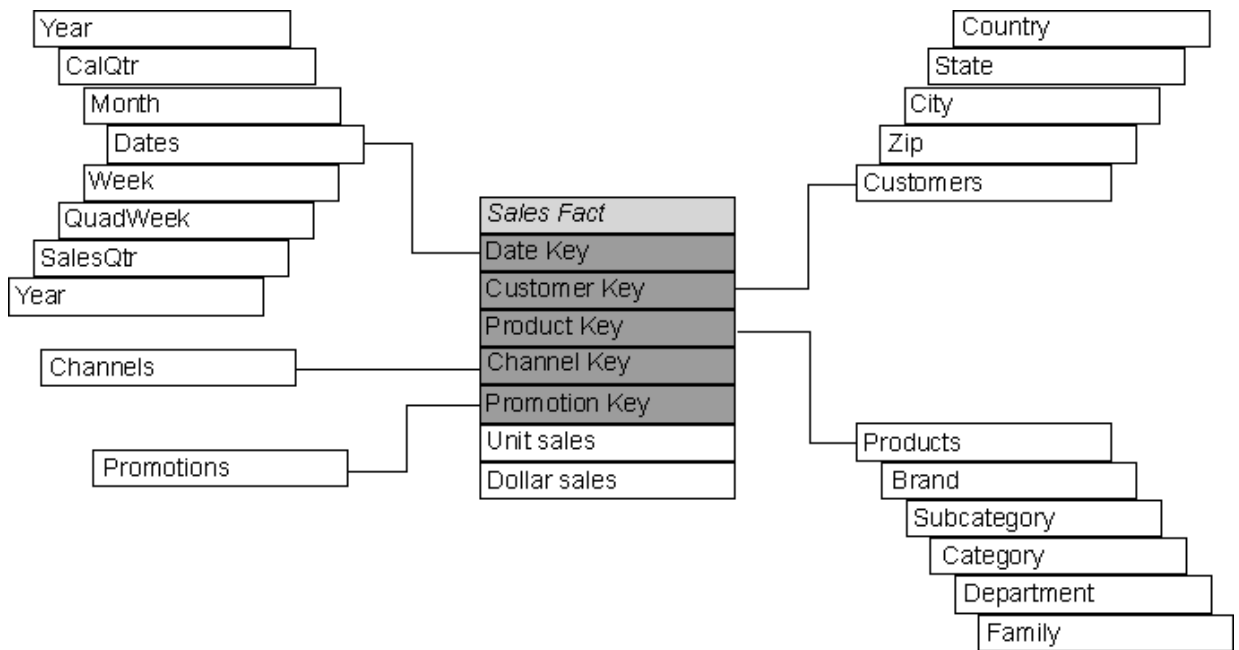
- Klasikinė žvaigždės architektūros schema



7 pav. Žvaigždės architektūra [26]

## Snaigės modelio architektūra

Snaigės schema - kai viena ar kelios dimensijų lentelės nėra tiesiogiai sujungtos su faktų lentele, o siejasi per kitas dimensijų lenteles.



8 pav. Snaigės architektūra [26]

Kuris duomenų saugyklos architektūros modelis yra pranašesnis ir kuo?

Snaigės tipo architektūroje yra lengviau palaikyti dimensijų lenteles, kai dimensijų lentelės yra labai didelės (sumažina bendrą duomenų dydį).

Žvaigždės tipo architektūra yra efektyvesnė atliekant kubo duomenų peržiūrą (browsing), nes mažiau ryšių, gali įtakoti eksploatacines sąlygas.

### 1.5. OLAP kubų architektūros modeliai

Galima išskirti tris pagrindines OLAP sistemų kubo saugojimo architektūras:

- MOLAP- daugiadimensinis OLAP. Duomenys saugomi specialiose daugiadimensinėse struktūrose. Privalumas – greitas indeksavimas iš anksto skaičiuojant agregatines reikšmes ir išsaugomos tik apskaičiuotos reikšmės. Trūkumas – nelabai kintamo dydžio, bei išsklaidytas.
- ROLAP – realiacinis OLAP. Dažniausiai naudoja žvaigždės schemą, palaiko SQL išplėtimus. Privalumas tas, kad kintančio dydžio (nera laisvų ląstelių išsklaidytiems duomenims), o trūkumas – nėra tiesaus priėjimo prie informacijos ląstelių.
- HOLAP – hibridnis OLAP. Tai kombinuota ROLAP ir MOLAP technologija. Tai ROLAP kintamumo ir MOLAP greito duomenų apskaičiavimo derinys.

MOLAP (multidimensional OLAP) tai duomenų analizės priemonė paremta duomenų kubo modeliu. MOLAP saugo jau susistemintus kubo duomenis, darant užklausą nereikia kiekvieną kartą skaičiuoti kubo reikšmių, todėl MOLAP priemonės pasižymi greitu atsakymu į užklausą. Tačiau į MOLAP kubą sudėtingiau pridėti naujas dimensijas, be to,

kube nesaugomi detalizuoti duomenys, todėl negalima pasinaudoti „Drill - In“ galimybėmis.

ROLAP veikia reliacinės duomenų bazės pagrindu kur sukuriamos užklausos, formuojančios duomenų kubus. Kiekvieną kartą analizuojant kubo duomenis, kubo reikšmės perskaičiuojamos. ROLAP užklausos nėra tokios greitos kaip MOLAP ypač dirbant su dideliais duomenų kubais, tačiau čia saugoma detali informacija ir galima pasinaudoti „Drill - In“ galimybėmis. ROLAP duomenų kubą paprasčiau papildyti naujomis dimensijomis.

HOLAP (hybrid OLAP) - tai ROLAP ir MOLAP priemonės apjungtos į vieną. HOLAP gali naudoti jau susistemintas reikšmes bei reliacinės bazės duomenų šaltinius.

### **Architektūrų palyginimas**

Dimensijų kiekis ROLAP sistemose yra neribotas, nes ROLAP duomenų modelis yra pakankamai dinamiškas ir lankstus. Tačiau MOLAP sistemose yra dimensijų kiekio problema, nes augant dimensijų skaičiui, sparčiai didėja duomenų bazės apimtis. Kadangi HOLAP architektūra yra ROLAP ir MOLAP sintezė, tai ji iš MOLAP paveldi tą pačią DB apimties problemą.

MOLAP sistemose užklausų greitis yra geresnis, nes jose saugomi agreguoti duomenys. Priešingai ROLAP sistemose yra saugomi patys detaliesi duomenys, todėl esant agreguotai užklausiai, turi būti sumuojami duomenys, dėl to užtrunka, arba turi būti lentelės agreguotų duomenų saugojimui, bet dėl to padidėja duomenų bazės apimtis.

HOLAP sistemose užklausų laikas yra geresnis nei ROLAP, nes čia HOLAP sistemos dalis duomenis saugo reliacinėje duomenų bazėje, o agreguotus daugiamačiuose masyvuose. Keičiantis organizacijos duomenų struktūrai, lengviausia pertvarkyti ROLAP sistemą, tuo tarpu MOLAP ir HOLAP sistemose reikalingas visiškas struktūros reorganizavimas. Tačiau ROLAP sistemas, palyginus su MOLAP, yra pakankamai sudėtinga administruoti.

Analizės galimybės yra geriausios MOLAP ir HOLAP sistemose, nes čia galima atlikti sudėtingas užklausas ir sudėtingą modeliavimą. ROLAP sistemose analizė mažiau lanksti, nes užklausoms naudojama SQL kalba, kurios galimybės yra ribotos. Tačiau MOLAP sistemos neturi vieningos užklausų sistemos



## **1.6. OLAP įrankių analizė**

### **1.6.1. MS SQL Server 2000 Analysis Services apžvalga**

MS SQL Server 2000 Analysis Services įrankio paskirtis yra užtikrinti greitą duomenų sandėlio informacijos analizę. Tuo tikslu OLAP serveryje kuriami daugiamačiai kubai, paremti duomenų sandėlio duomenimis. Taip pat kubuose saugojami ir konsoliduoti (aukštesnio hierarchijos lygio) duomenys. Priklausomai nuo naudojamo duomenų modelio, kube gali būti saugojama duomenų sandėlio duomenų kopija, konsoliduoti duomenys, arba informacija skaitoma tiesiai iš duomenų sandėlio. MS SQL Server 2000 sudaro OLAP (analizės) serveris, kuris valdo daugiamačius kubus, ir PivotTable paslauga, kurią naudoja Microsoft Excel ar kitos kliento programos, jungiantis prie serverio ir išgaunant kubo duomenis, arba kuriant kubo struktūrą serveryje.

Analysis Services palaiko tris pagrindines duomenų saugojimo architektūras - (MOLAP, ROLAP, HOLAP).

Taip pat sistema leidžia aprašyti skaičiuojamus kintamuosius, t.y. šių kintamųjų reikšmės nesaugomos šaltinio DB, o apskaičiuojamos pagal aprašytas formules. Ta pati galimybė užtikrinama ir dimensijų nariams. Dimensijų nariams taip pat galima aprašyti atributus, kurie vėliau gali būti panaudoti užklausoje. Dimensijų narių atributų pagrindu, gali būti konstruojamos virtualios dimensijos.

Apibendrinant visas Analysis Services įrankio teikiamas paslaugas, galima pasakyti, kad modelis lankstus ir tai užtikrina optimalų duomenų saugyklos veikimą (saugyklos apimtis, užklauso aptarnavimo greitis). Be to tai viena nedaugelio sistemų, kurios leidžia pasirinkti iš trijų duomenų saugojimo architektūrų. Tai įgalina naudoti Analysis Services įvairiose probleminėse srityse. Tačiau duomenys kubui išgaunami iš duomenų sandėlio, todėl norint užtikrinti korektišką kubo struktūrą, pats duomenų šaltinis turi būti teisingai suprojektuotas, priešingu atveju kubo projektavimas tampa sudėtingas.

Analysis Services pasižymi lanksčiomis duomenų peržiūros galimybėmis. Analysis Manager priemonėmis galima peržiūrėti kubo duomenis elektroninės lentelės pavidalu, tačiau į vieną stulpelį ar eilutę galima įkelti daugiau nei vieną dimensiją, duomenis filtruoti pagal tam tikrą dimensijos narį, keisti vietomis stulpelius ir eilutes ir kt. Be to tokioje duomenų peržiūroje ir dimensijos, ir kintamieji traktuojami vienodai, t.y. tiek su dimensijomis, tiek su kintamaisiais galima atlikti tuos pačius veiksmus.

Analysis Services realizuota MDX (Multidimensional Expressions) kalba, skirta manipuluoti duomenimis. Naudojant MDX sintaksę galima kurti daugiamačio kubo objektus, formuoti užklausas ir manipuluoti duomenimis. Manipuliavimo duomenimis

galimybes išplečia Analysis Services duomenų modifikavimo savybė, t.y. vartotojai gali keisti kubo duomenis analizės ir situacijų modeliavimo tikslais. Šiuo atveju duomenys tiesiogiai nepakeičiami, tačiau saugomas pats pakeitimo faktas ir užklausa atveju vartotojui pateikiami jau pakeisti duomenys.

Analysis Services pasižymi draugiška vartotojo aplinka. Sistemos administratoriui darbą palengvina meistras (wizards), bei redagavimo langai, leidžiantys intuityviai atlikti kubo projektavimą ir redagavimą. Taip pat realizuota pakankamai išsami duomenų saugumo sistema, tiek administratoriaus, tiek paprastų vartotojų atžvilgiu.[11]

### **1.6.2. Oracle Express apžvalga**

Oracle Express sudaro Express serveris, skirtas daugiamatį kubo duomenų saugojimui, kubų kūrimui ir modifikavimui, bei klientinės programos, skirtos šių duomenų peržiūrai ir analizei. Yra šios pagrindinės programos:

- Express Objects – objektiškai orientuota vizuali programų kūrimo aplinka, leidžianti kurti klientines programas, dirbančias tiek su OLAP duomenimis tiek su reliaciniais.
- Express Analyzer – programa skirta OLAP duomenų peržiūrai ir analizei, bei prezentacijų (briefings) kūrimui.
- Web Publisher – priemonė skirta kurti tinklapiams, kurios dėka OLAP duomenys gali būti analizuojami interneto naršyklės pagalba.

Oracle Express duomenims saugoti naudoja multikubo struktūrą. Tai struktūra, leidžianti vienoje DB saugoti kelias logines struktūras (šiuo atveju kubus). Reliacinėse DB analogija šiai struktūrai yra daug lentelių vienoje duomenų bazėje [11].

Kubas šiuo atveju yra daugiamatis kintamasis, analizuojamas pagal tam tikras dimensijas. Oracle Express atveju kiekvienam kintamajam reikia nurodyti dimensijų rinkinį, taigi galima teigti, kad viename kube gali būti saugojamos ne daugiau kaip vieno kintamojo reikšmės, priešingai nei Analysis Services atveju, kur tas pats kubas gali turėti daug kintamųjų. Iš esmės Oracle Express naudoja MOLAP saugojimo architektūrą, tačiau tuo atveju kai naudojamos dviejų tipų saugyklos (pastovi ir laikina), gaunama HOLAP architektūra, kadangi daugiamatėje struktūroje saugomi tik konsoliduoti duomenys, o detalūs duomenys išgaunami tiesiogiai iš duomenų sandėlio.

Oracle Express taip pat palaiko formules, nusakančias sąryšius tarp kintamųjų, tačiau skirtingai nei Analysis Services, formulių rezultatai nėra saugojami kubo struktūroje, o apskaičiuojami realiu laiku, kai suformuojama atitinkama užklausa.

Express Server administratoriaus aplinkoje pateikiamos ribotos duomenų peržiūros galimybės, kadangi lentelėje galima išdėstyti tik dvi dimensijas. Tačiau klientinės programos pateikia priemones grafiniam duomenų atvaizdavimui įvairių diagramų pavidalu.

Express Server aplinkoje naudojama vidinė programavimo kalba SPL (Stored Procedure Language), kurios pagalba galima manipuluoti ne tik duomenimis saugomais kube, bet ir apibrėžti naujus duomenis naudojant išraiškas, bei išplėsti Express Server funkcionalumą. Kalba leidžia vartotojui apibrėžti savas funkcijas, iškviešti išorines funkcijas, ar naudoti SQL užklausas. SPL kalba yra automatiškai generuojamos ir duomenų užkrovimo iš duomenų šaltinių programos.

Analysis Services atveju duomenų užkrovimo procesas yra paslėptas nuo vartotojo, tuo tarpu Oracle Express aplinkoje galima sugeneruotą užkrovimo programą modifikuoti pagal vartotojo poreikius.

Išskirtinė Oracle Express savybė yra galimybė užkrauti kubo duomenis iš struktūrizuotų failų. Galimi du failų tipai: duomenų bazės failai (saugoma informacija apie dimensijų narius ir kintamuosius) ir dimensijų failai (saugoma informacija apie dimensijų hierarchiją). Tai išplečia galimų duomenų šaltinių ribas, tačiau struktūrizuotiems failams paruošti reikia žinių ir laiko [11].

### **1.6.3. OLAP sistemų palyginimas**

Tiek Analysis Services, tiek Oracle Express turi savo privalumų. Oracle Express svarbus pranašumas yra daugiau galimų veikimo platformų. Express Server gali veikti Windows, Unix ir kitose aplinkose. Tuo tarpu Analysis Server gali veikti tik Windows platformoje. Šis faktas išplečia Oracle Express panaudojimo galimybes.

Probleminės srities požiūriu, daugiau problemų galima spręsti naudojant Analysis Services. Tai įmanoma dėl žymiai lankstesnio duomenų modelio. Nors ir Oracle Express, ir Analysis Services duomenų modelis yra multikubas, tačiau Analysis Services leidžia pasirinkti daugiau galimų duomenų saugojimo architektūrų. Oracle Express palaiko tik MOLAP ir HOLAP architektūras.

Duomenų saugyklos apimties požiūriu, Analysis Services taip pat siūlo efektyvesnį duomenų saugojimą, kubai naudoja mažiau atminties negu Oracle Express kubai. Be to, šiuo požiūriu svarbi ir Analysis Services galimybė padalinti kubą į kelias fizines dalis.

Savo ruožtu Oracle Express pasižymi didesnėmis analitinėmis galimybėmis, bei duomenų atvaizdavimo būdų įvairove. Tai įmanoma dėl papildomų klientinių programų.

Oracle Express taip pat pateikia atskirą objektiškai orientuotą programų kūrimo aplinką (Express Objects), pritaikytą klientinių OLAP taikomųjų programų kūrimui. Tačiau Express Server administravimo aplinka reikalauja daugiau pasiruošimo nei Analysis Services aplinka, kurios paprastumas leidžia nesudėtingai konstruoti daugiamačius kubus. Nors Analysis Services nėra atskiros programavimo aplinkos, tačiau PivotTable paslauga leidžia naudojant kitas programavimo aplinkas (MS Visual Basic, MS Visual C++) išgauti pilną OLAP sistemos funkcionalumą. Tam tikrais atvejais programuotojui patogiau naudoti jau įprastą programavimo aplinką ir nereikia įsisavinti naujos aplinkos.

Ir Oracle Express, ir Analysis Services leidžia išgauti duomenis iš įvairių DB. Oracle Express be įprastų DB leidžia užkrauti kubo duomenis iš struktūrizuotų failų, taip praplėsdamas galimų duomenų šaltinių aibę.

#### **1.6.4. Išvados**

Lyginant Microsoft SQL Server 2000 Analysis Services ir Oracle Express sistemas, naudingiau pasirinkti pirmąją sistemą dėl lankstesnio duomenų modelio. Be to, Analysis Services įrankis leidžia išgauti pilną OLAP funkcionalumą naudojant klientines programas, sukurtas kitose programavimo aplinkose.

### **1.7. *Kompiuterizuojamos sistemos varianto parinkimas***

Kompiuterizuojamos sistemos varianto pasirinkimas:

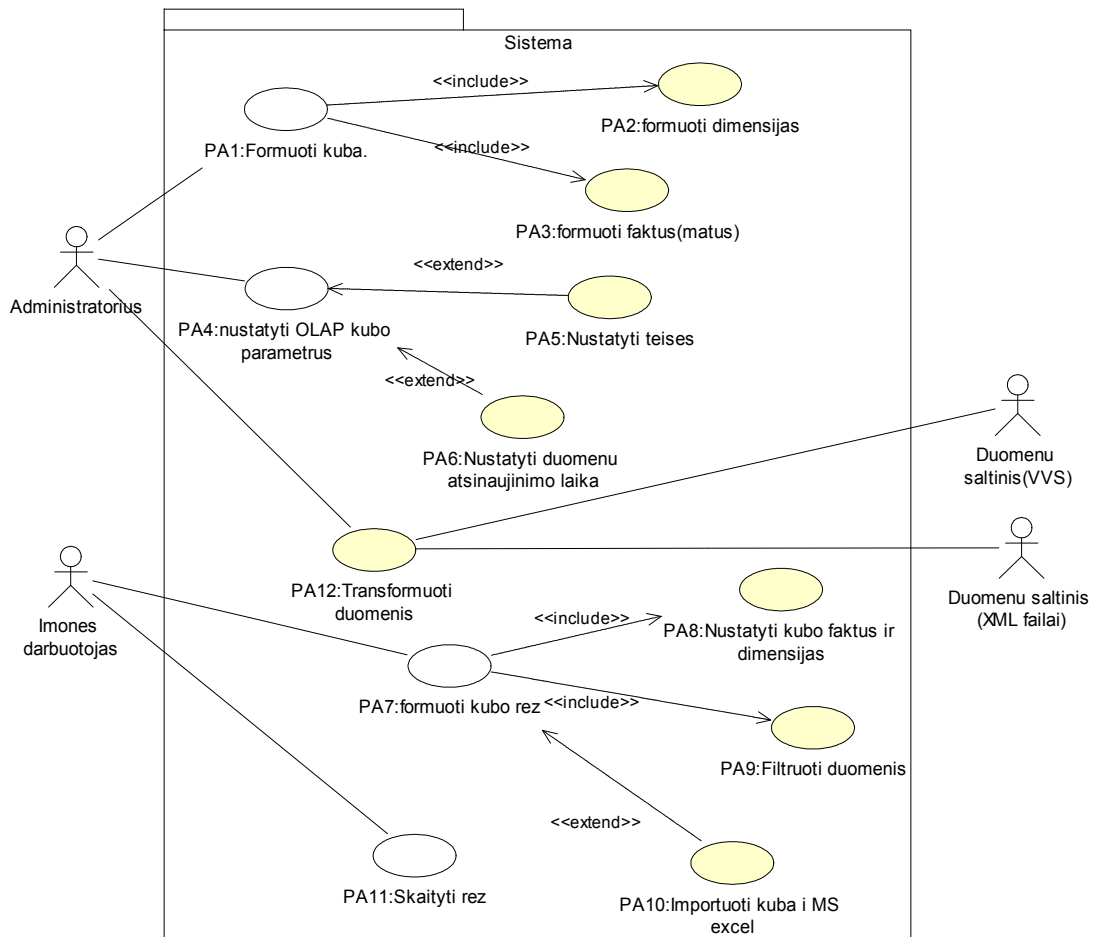
- Duomenų saugyklos modelio realizavimui pasirinkta Microsoft SQL Server 2000 Analysis Services įrankis, kuris teikia lenksčias duomenų saugyklos modelio galimybes ir užtikrina optimalų sistemos veikimą (saugyklos apimtis, užklausų aptarnavimo greitis).
- Dokumentacijos ruošimo bei projektavimo įrankiais pasirinkti „Rational Rose 2000 Enterprise Edition“ ir „Microsoft Visio Professional 2003“ paketai.
- Dokumentacijai rašyti buvo pasirinktas „Microsoft Word 2003“ paketas.

## 2. OLAP DUOMENŲ SAUGYKLOS MODELIS

### 2.1. Reikalavimų modelis

#### 2.1.1. Sistemos panaudojimo atvejų modelis

Sistemos panaudojimo atvejų diagrama pateikta sekančiame paveiksle. Joje pavaizduoti sistemos aktoriai ir jų atliekamos funkcijos.



9 pav. Veiklos panaudojimo atvejų modelis

- Administratorius – tai visos sistemos administratorius, kuris administruoja OLAP sistemą (nustato vartotojų teises, duomenų atsinaujinimo laiką, bei formuoja OLAP kubą);
- Įmonės darbuotojas- įmonės darbuotojas, kuris turi teises jungtis prie OLAP sistemos, formuoti OLAP kubo rezultatus, bei juos analizuoti.
- Duomenų šaltinis(VVS) – tai reliacinė DB, kuri saugo verslo valdymo sistemos transakcijų duomenis;

- Duomenų šaltinis(XML failai) – tai duomenys iš kitų šaltinių, kurie saugomi XML formato failuose, pagal nustatytus XML failų šablonus;

Detalus kiekvieno panaudojimo atvejo aprašymas, bei aktorių atliekamos funkcijos aptartos panaudojimo atvejų specifikacijose, kurios pateiktos priede (žiūrėti 2 Priedas).

## **2.1.2. Nefunkciniai reikalavimai**

### **2.1.2.1. Reikalavimai sistemos išvaizdai**

Jokių ypatingų reikalavimų sistemos išvaizdai nėra, kadangi duomenų atvaizdavimui naudojama įprasta MS Excel aplinka. Ji yra patogi realizavimui, reikalaujanti minimalių pastangų bei įgūdžių iš vartotojo (įmonės darbuotojų) pusės. Tai neperkauta, bei neerzinanti vartotojo sąsaja.

### **2.1.2.2. Reikalavimai panaudojamumui**

Kiekvienam vartotojui, prieš dirbant su sistema, reikėtų perskaityti atitinkamai vartotojo ar administratoriaus vadovą. Jokio papildomo pasiruošimo nereikia.

### **2.1.2.3. Reikalavimai vykdymo charakteristikoms**

- *Greičiui* - įgyvendinti technologinius sprendimus, kurie užtikrintų pakankamą duomenų išgavimo greitį turimomis interneto ir kompiuterio sąlygomis.
- *Duomenų saugumui* - kadangi duomenys yra labai svarbūs, bei konfidencialūs, turi būti užtikrintas maksimalus duomenų perdavimo saugumas.
- *Patikimumui* - reikia užtikrinti sistemos ir serverio darbo stabilumą. Pageidautina, kad sistema turi veikti 24h per parą, 7 dienas per savaitę. Kitu atveju duomenų saugykla nebus galima pasinaudoti.

### **2.1.2.4. Reikalavimai veikimo sąlygoms**

#### **Reikalavimai fizinei aplinkai**

Sistema bus naudojama įprastomis sąlygomis, todėl jai nekeliami ypatingi apribojimai ar reikalavimai. Pageidautina dirbti kambario sąlygomis prie normalaus apšvietimo.

#### **Reikalavimai technologinei aplinkai**

Bus įgyvendinti technologiniai sprendimai, kurie leistų dirbti turimomis interneto (teikiamo įmonės) ir kompiuterio sąlygomis. Jokie periferiniai įrenginiai, neskaitant pelės ir klaviatūros, nereikalingi.

### 2.1.2.5. Reikalavimai sistemos priežiūrai

Sistemos priežiūrai keliami reikalavimai:

- Sistema turi būti realizuota taip, kad jos veikimui užtikrinti nereikėtų nuolatinio programuotojo įsikišimo (keičiantis organizacijos veiklos taisyklėms šis reikalavimas netaikomas).
- Keičiantis organizacijos veiklos taisyklėms, turi būti įmanoma adaptuoti sistemą prie naujų poreikių

### 2.1.2.6. Reikalavimai saugumui

Veiksnius su sistemoje kaupiamais duomenimis gali atlikti tik vartotojai, kuriems suteiktos teisės.

Vartotojai, kurie yra ne lokaliame tinkle, jungiantis prie sistemos privalo naudoti VPN tinklą.

### 2.1.3. Reikalavimai sistemos funkcionalumui palaikyti

#### Reikalavimai serveriui

Techniniai ir programiniai reikalavimai serveriui pateikti sekančioje lentelėje.

2 lentelė Reikalavimai serveriui

#### Programinės įrangos reikalavimai

OS	Microsoft Windows Server 2000/2003
DBVS	Microsoft SQL Server 7.0/8.0
OLAP servisas	Microsoft SQL Server OLAP Service

#### Techninės įrangos reikalavimai

Turi atitikti programinės įrangos funkcionalumui keliamus reikalavimus

#### Reikalavimai vartotojo kompiuteriui

Techniniai ir programiniai reikalavimai vartotojo kompiuteriui pateikti sekančioje lentelėje.

3 lentelė Reikalavimai vartotojo kompiuteriui

#### Programinės įrangos reikalavimai

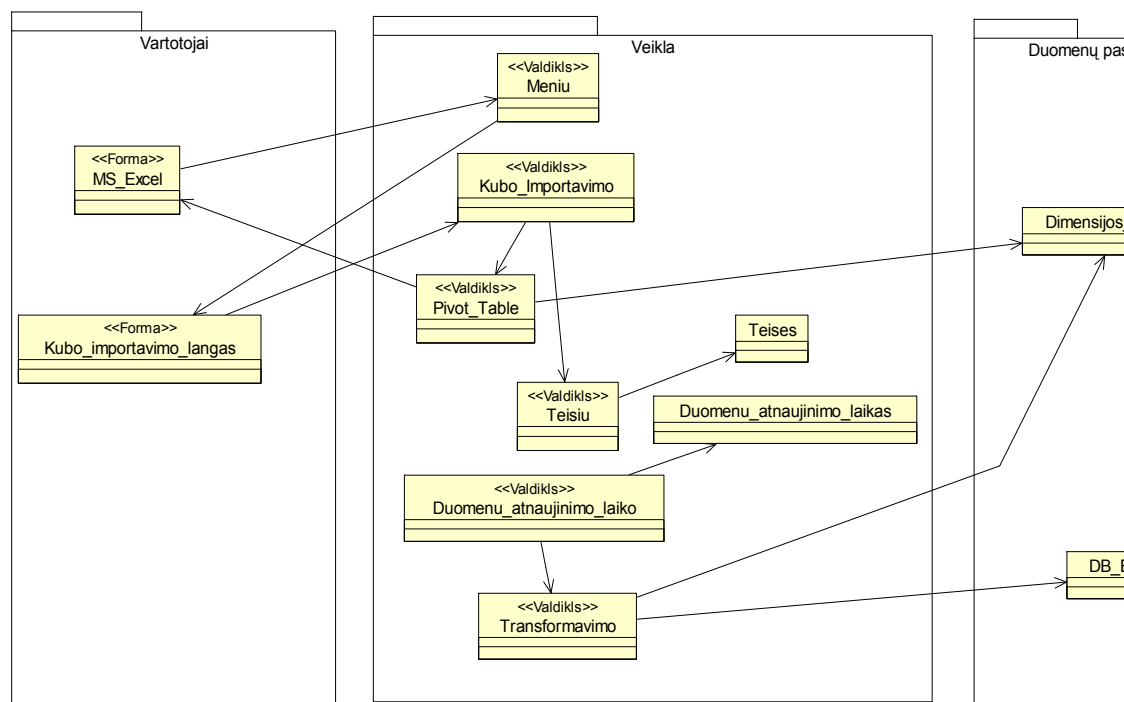
OS	Windows 98/2000/XP
Programa	Microsoft Excel 2000/XP, .NET framework (tik administratoriaus kompiuteryje)

#### Techninės įrangos reikalavimai

Turi atitikti programinės įrangos funkcionalumui keliamus reikalavimus

## 2.2. Loginė OLAP sistemos architektūra

Sistema pasirinkta trijų sluoksnių architektūra – atvaizdavimo, veiklos logikos ir duomenų sluoksniai :



10 pav. Sistemos loginė architektūra

Atvaizdavimo sluoksnis atsakingas už vartotojo paslaugas – realizuoja vartotojo sąsają, kad vartotojas galėtų pasirinkti norimas atlikti funkcijas ir matytų pasirinktą informaciją.

Veiklos logikos sluoksnis atsakingas už veiklos paslaugas – užtikrina sistemos funkcionalumą, t.y. vartotojo sistemoje pasirinktų funkcijų atlikimą. Vartotojo paslaugų elementai – ribinės klasės sąveikauja su atitinkamais valdikliais, kurie realizuoja metodus, atitinkančius tam tikrą sistemos funkciją.

Duomenų sluoksnis atsakingas už duomenų paslaugas – saugo sistemai reikalingus duomenis.

### Vartotojo sąsajos komponentas

Vartotojo sąsaja - tai sistemos paketas, atsakingas už sąsają su sistemos vartotoju. Jį sudaro sąsajos klasės.

Vartotojo sąsajos komponentai – tai MS Excel programos ir kubo importavimo lango klasės. Šių klasių pagalba vartotojas bendrauja su sistema ir atlieka norimus veiksmus.

Vartotojo sąsajos komponentas skirtas užtikrinti, kad sistemos vartotojas galėtų lengvai pasirinkti norimus atlikti veiksmus.



Vartotojo sąsają sudaro MS Excel programa ir kubo importavimo lango formos. Tai yra programinės įrangos produktai, kurių pagalba vyksta vartotojo prisijungimas prie serverio, duomenų importavimas į MS Excel programą, bei duomenų peržiūra ir analizė. Vartotojo sąsajos komponentas sąveikauja su veiklos komponentais.

**Resursai** - vartotojo sąsajos komponentai saugomi vartotojo kompiuteryje.

**Apribojimai:**

Turi turėti ryšį su veiklos komponentais.

Turi apsaugoti sistemos vartotoją nuo atsitiktinių klaidų atliekant svarbius veiksmus.

**Veiklos komponentas**

Veiklos komponentas užtikrina, kad duomenys būtų prieinami atitinkamiems vartotojams, bei jų korektiškumą. Tai yra tarpinis komponentas tarp vartotojo sąsajos ir duomenų bazės.

Veiklos taisyklių komponento pagrindinė funkcija –užkrauti duomenis iš duomenų bazės į veiklos objektus, užtikrinti duomenų ir naudojimosi duomenimis korektiškumą.

Veiklos taisyklių komponentai sąveikauja su vartotojo sąsajos komponentais ir DB komponentais.

**Resursai** - SQL server OLAP service įrankis, interneto tinklas.

**Apribojimai:**

Turi būti užtikrinamas apdorojamų duomenų korektiškumas.

Turi būti sąsaja su DB komponentais.

**DB komponentas**

DB komponentas skirtas visų sistemoje naudojamų duomenų saugojimui ir tvarkymui. DB komponento pagrindinė funkcija – duomenų saugojimas ir duomenų integralumo užtikrinimas. Duomenys saugomi lentelėse.

**Apribojimai:**

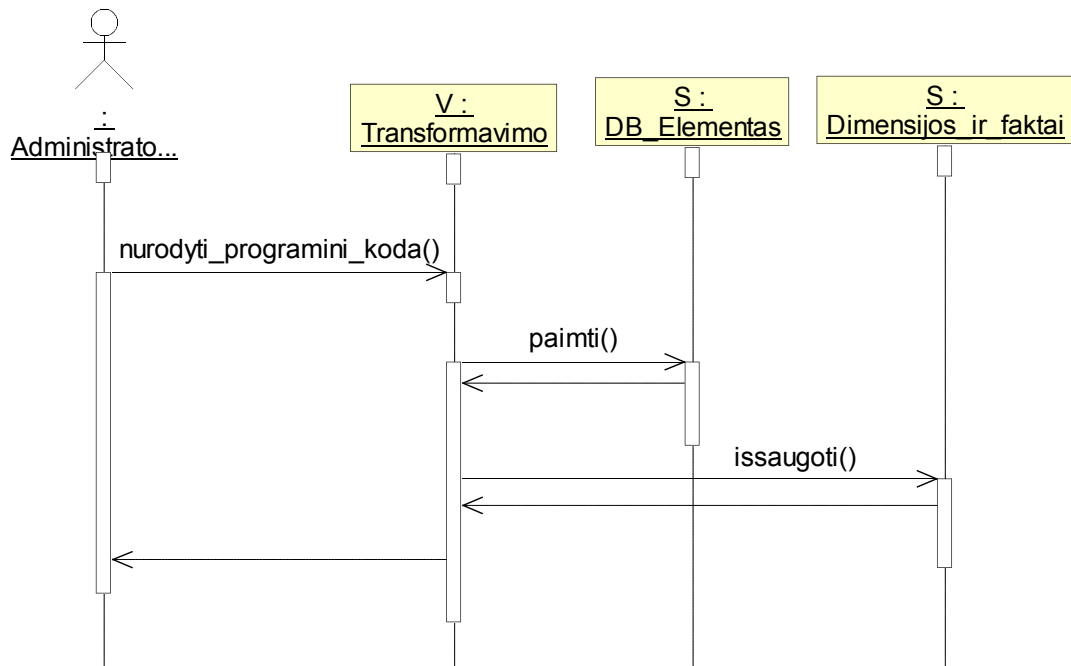
DB komponentas turi būti saugomas MS SQL serveryje.

**2.3. Panaudojimo atvejų sekų diagramos**

Žemiau pateiktos panaudojimo atvejų sekų diagramos, aprašančios kiekvieno panaudojimo atvejo transakcijų seką.

**PA „Formuoti kubą“** - šis panaudojimo atvejis susideda iš dviejų PA, tačiau sekų diagramose šie PA nėra išskirti atskirai (projektavimo etape faktų ir dimensijų formavimo principas nesiskiria, todėl yra tikslinga juos apjungti).

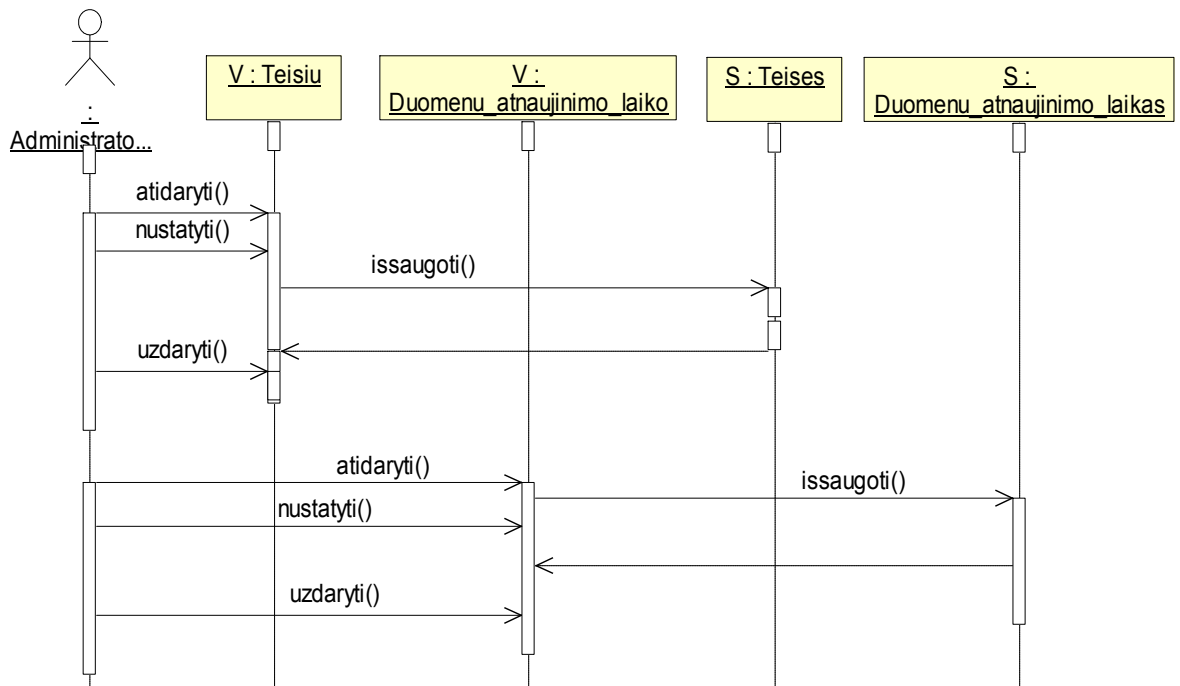
Administratorius suformuoja OLAP kubo dimensijas ir faktus, t.y. nurodo programinį kodą transformacijos valdikliui, o šis savo ruožtu paima DB elementą ir išsaugo jį kaip OLAP kubo elementą.



11 pav. PA „Formuoti kuba“ sekų diagrama

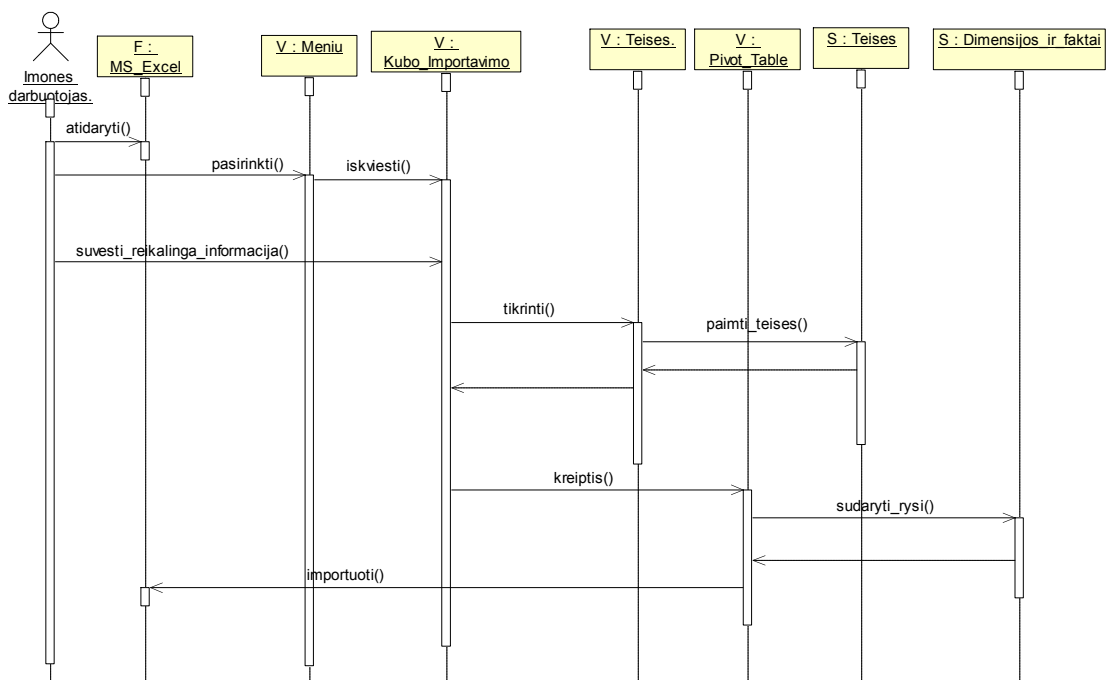
**PA „Nustatyti OLAP kubo parametrus“** - šis panaudojimo atvejis susideda iš dviejų PA, tačiau sekų diagramose neišskyrčiau šių PA atskirai (procesas vyksta taip pat, tik skiriasi funkcija, kurią atlieka administratorius – vienu atveju jis nustato vartotojų teises, kitu - kubo atnaujinimo laiką).

Administratorius nurodo vartotojų teises ir duomenų atsinaujinimo laiką (diena, valandą, periodiškumą), o teisių valdiklis ir duomenų atnaujinimo laiko valdiklis atitinkamai išsaugo nustatymus.



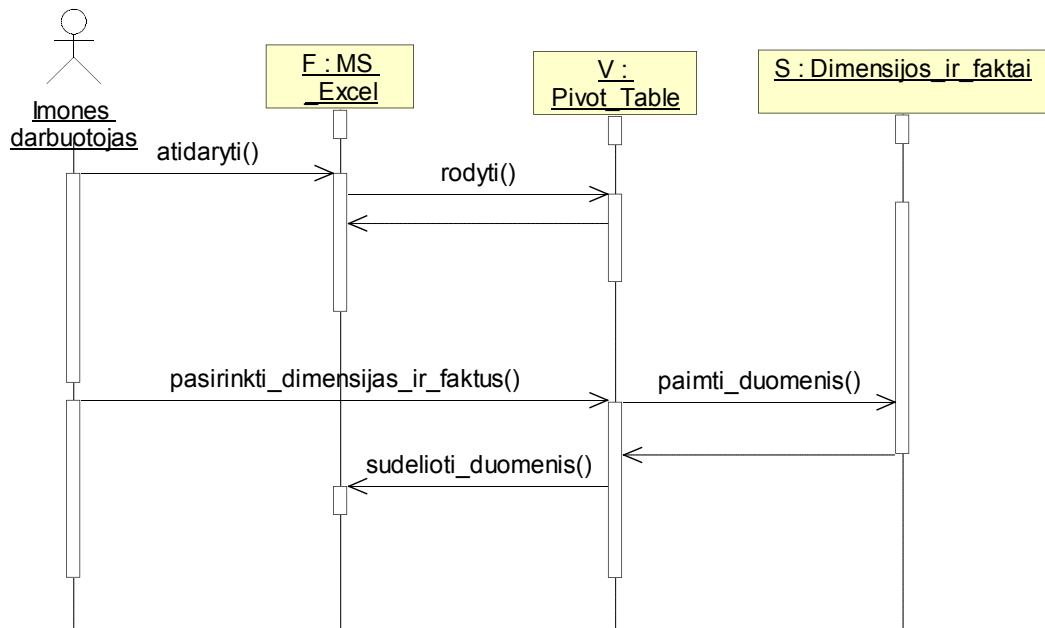
12 pav. PA „Nustatyti OLAP kubo parametrus“ sekų diagrama

**PA „Importuoti kubą“** – šis panaudojimo atvejis įeina į PA „Formuoti kubo rezultatus“. Įmonės darbuotojas MS Excel programoje per meniu punktą iškviečia kubo importavimo valdiklį. Kubo importavimo valdiklis sudaro ryšį su teisių valdikliu, kuris tikrina vartotojo teises, jei vartotojas neturi teisės jungtis prie sistemos išduodamas pranešimas. Jei turi teises tada, naudojantis kubo importavimo valdiklio vedliu, sudaro ryšį su OLAP kubo ir per Pivot Table importuoja kubo dimensijas ir faktus.



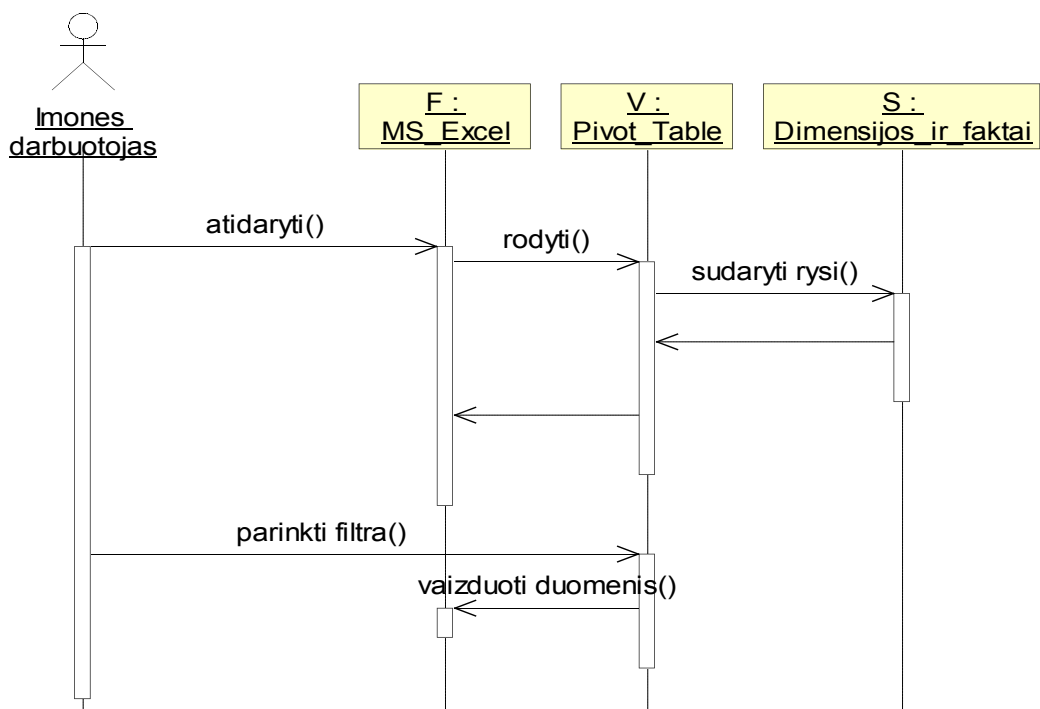
13 pav. PA „Importuoti kuba“ sekų diagrama

**PA „Nustatyti kubo faktus ir dimensijas“** – įmonės darbuotojas atidaro MS Excel programą, kurioje Pivot Table valdiklio pagalba pasirenka dimensijas ir faktus, kuriuos nori matyti.



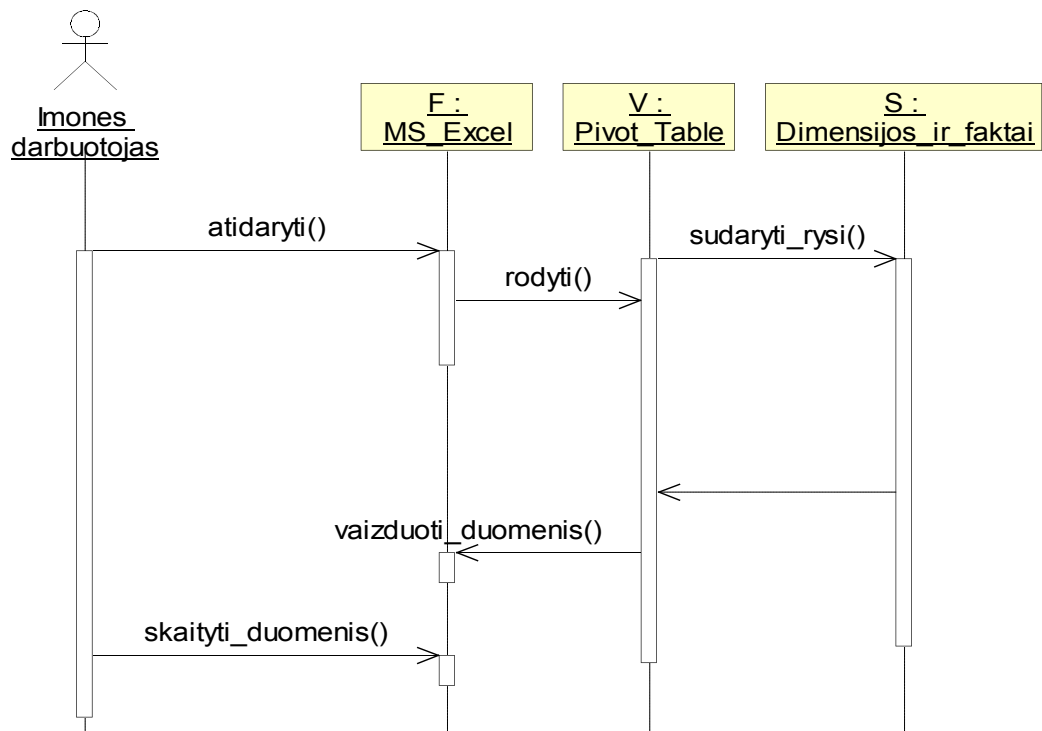
14 pav. PA „Nustatyti kubo faktus ir dimensijas“ sekų diagrama

**PA „Filtruoti duomenis“** - įmonės darbuotojas atidaręs MS Excel programa, Pivot Table nurodo norimą filtrą ir Pivot Table atfiltruoja duomenis, pagal nurodytą filtrą.



15 pav. PA „Filtruoti duomenis“ sekų diagrama

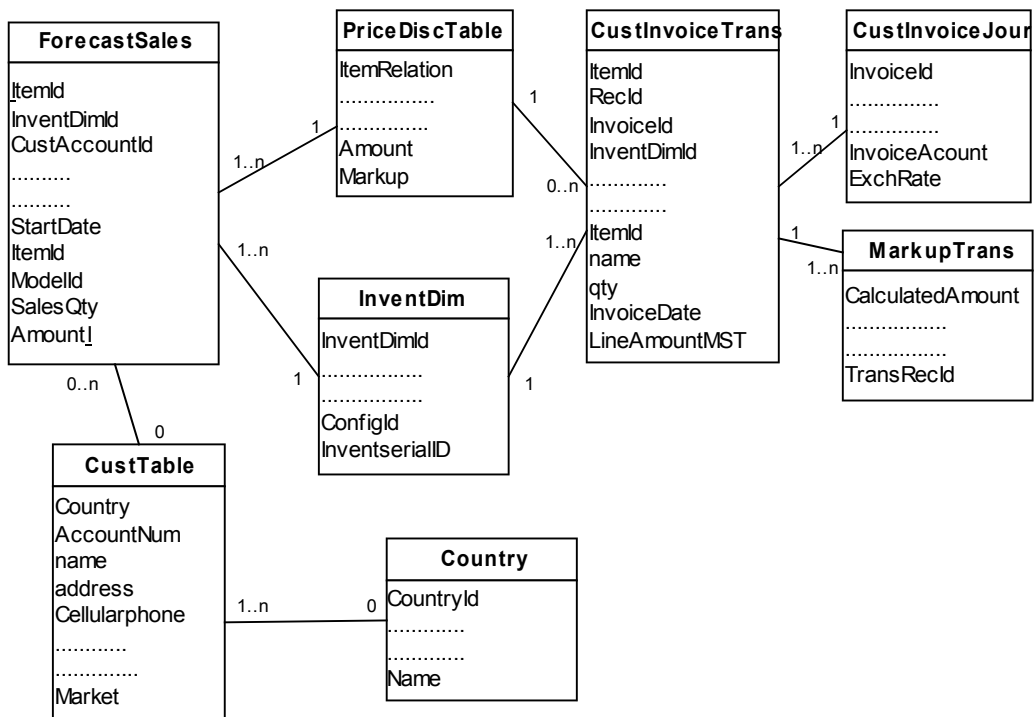
PA „Skaityti rez“ – įmonės darbuotojas atidaręs MS Excel programą, gali skaityti duomenis.



16 pav. PA „Skaityti rez“ sekų diagrama

## 2.4. Duomenų struktūros modelis

Pradiniai duomenys yra saugomi reliacinėse DB ir XML formato failuose. DB schemoje charakterizuojami tik tie atributai, kurie yra naudojami kuriant OLAP sistemą.

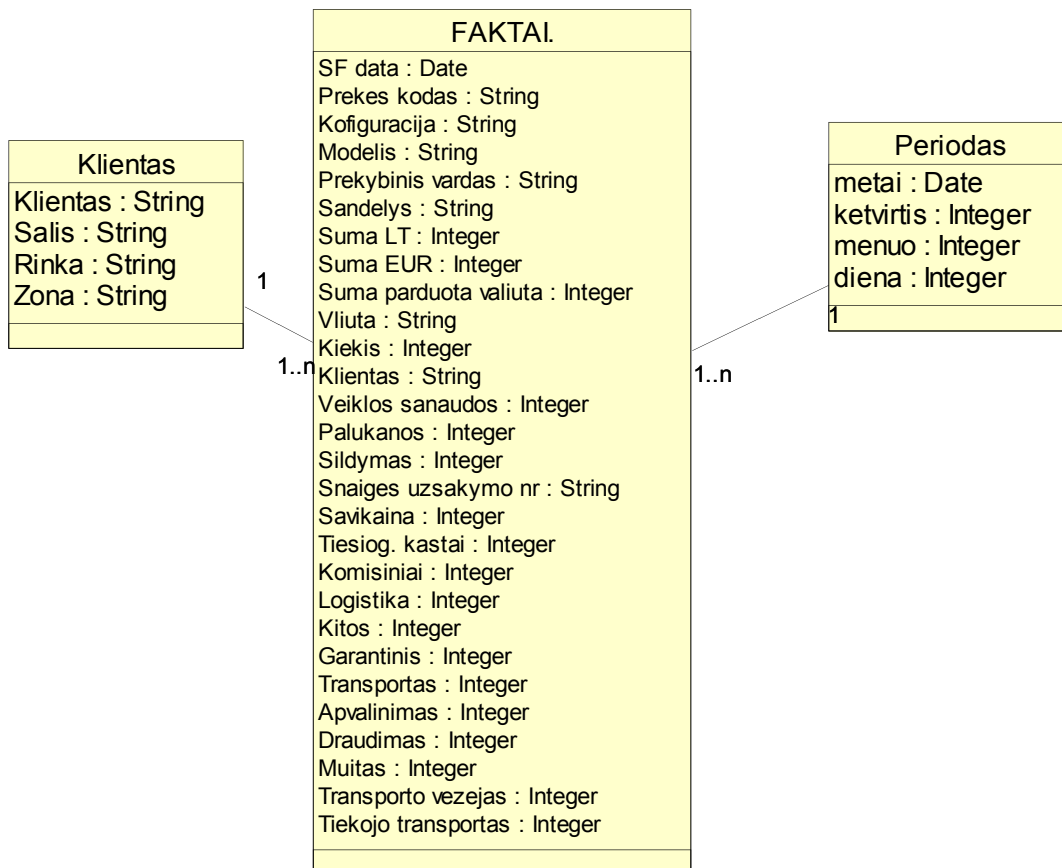


17 pav. DB schema

Pateikiama supaprastinta DB schema – lentelėse parodomas tik tos reikšmės, kurios yra naudojamos OLAP kube. Detalūs DB lentelių aprašai pateikti priede (žiūrėti 2 Priedas). Taip pat OLAP sistema duomenis ima iš XML formato failų. XML failų šablonai pateikiami priede (žiūrėti 2 PRIEDAS).

Duomenys iš duomenų bazės ir XML failų yra transformuojami į „Žvaigždės“ tipo duomenų schemą, kuri toliau naudojama OLAP sistemoje.

Transformuota DB schema pavaizduota sekančiame paveiksle.



18 pav. DB transformuota į „Žvaigždės“ tipo schemą

## 2.5. Testavimo modelis bei testavimo procedūros

Pasiruošimas sistemos testavimui susideda iš trijų pagrindinių šio proceso fazių:

- *Testavimo strategijos nustatymas*  
Nustatoma sistemos testavimo apimtis, apibrėžiama sistemai taikoma testavimo strategija, vykdomi veiksmai, reikalingi resursai ir naudojami procesai.
- *Testavimo planavimas*  
Detalizuojami darbai, poreikiai ir pastangos, reikalingos sistemos testavimo vykdymui.
- *Testavimo sąlygų/atvejų apibrėžimas*  
Aprašomi sistemai taikomi testai, apdorojami duomenys, automatinio testavimo apimtis ir laukiami rezultatai.

### 2.5.1. Vienetų testavimas

Šioje fazėje testuojami atskiri sistemos elementai – programinės klasės: formos, rodiniai, dokumentai. Sistemą sudarantys komponentai taip pat gali būti įvardinti kaip

atskiri sistemos objektai, tačiau jie susideda iš programinių klasių. Todėl jų testavimo principai pateikiami kartu su integracinio testavimo strategija.

Analizuojant elementų vidinę struktūrą - realizaciją, taikomas „baltos dėžės“ metodas. Testuojant elementų elgsenos charakteristikas, taikomas „juodos dėžės“ metodas.

- “Baltos” dėžės metodas

Tai – struktūrinis testavimo metodas. Šioje testavimo fazėje siekiama pereiti visus programinius sprendimo priėmimo taškus (sąlyginius operatorius, ciklus) taip, kad kiekvienas programos operatorius būtų įvykdytas bent vieną kartą. Testavimo atvejai sudaromi iš programinio kodo struktūros, atsižvelgiant į programiniams operatoriams keliamas sąlygas.

4 lentelė. Testinių „Baltos“ dėžės metodo rinkinių forma

Programinio kodo fragmentas	Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
...	...	...	...

- „Juodos” dėžės metodas

Šioje fazėje sistemos elementai testuojami neatsižvelgiant į jų programinę realizaciją. Įvedami duomenys ir išvedami rezultatai suskirstomi į atskiras ekvivalentines klases – programos elgesys su kiekvienu atitinkamos klasės nariu yra toks pats. Testavimo atvejai parenkami taip, kad apimtų reikšmes iš ekvivalentinių klasių, ribines ekvivalentinių klasių reikšmes, reikšmes, esančias už ekvivalentinių klasių ribų, neleistinas reikšmes.

5 lentelė. Testinių „Juodos“ dėžės metodo rinkinių forma

Testavimo atvejo ID	Įėjimo duomenys	Rezultatai
...	...	...

## 2.5.2. Integravimo testavimas

Integracinio testavimo fazėje analizuojamas programinių komponentų (posistemių kaip atskirų duomenų bazių) veikimas, taikant funkcinio, „juodos dėžės“, vykdymo bei sąsajų testavimo strategijas.

- Veikimo testavimas

Testuojant sistemos komponentų veikimo parametrus, vadovaujamesi sistemai keliamais nefunkciniais reikalavimais: duomenų bazės dydis, lygiagrečių sesijų skaičius, patikimumas, greitis, aparatūrinės ir programinės įrangos konfigūracija ir t.t.

6 lentelė. Sistemos veikimo testavimo testinių duomenų rinkinių forma

Testavimo	Sistemos veikimo	Sąlygos	Laukiamas rezultatas
-----------	------------------	---------	----------------------



<b>atvejo ID</b>	<b>(paleidimo) aplinka</b>		
...	...	...	...

- Funkcinis testavimas

Funkcijų testavimas pagrįstas sistemos panaudojimo atvejais. Testavimo atvejai sudaromi taip, kad kiekvienas panaudojimo atvejo scenarijus būtų patikrintas bent vieną kartą. Tokiu būdu sudarius kiekvienam panaudojimo atvejui testinių duomenų rinkinius, parenkamos konkrečios sąlyginių duomenų reikšmės, atitinkančios tam tikro scenarijaus prieš sąlygas. Taip gaunami testavimo atvejai funkciniam sistemos testavimui. Taip pat šioje testavimo fazėje atliekama duomenų korektiškumas kontrolė.

7 lentelė. Funkcinio testavimo testinių atvejų rinkinių forma

Testavimo atvejo ID	Scenarijus	Sąlygos	Laukiamas rezultatas
....	...	...	...

- Saugumo testavimas

Saugumo testavimas, kaip ir funkcinis testavimas, pagrįstas sistemos panaudojimo atvejais. Sudarant testavimo atvejus, atsižvelgiama į vartotojų atliekamas funkcijas – prieigos teises sistemoje.

8 lentelė Saugumo testavimo testinių duomenų rinkinių forma

Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
....	...	...

### 2.5.3. Priėmimo testavimas

Naudojant šį metodą programinė įranga bus demonstruojama klientui, kuris ją išbandys norint nuspręsti ar ji atitinka kliento poreikius, specifikaciją. Testavimas atliekamas peržiūrint kiekvieną programinės įrangos panaudojimo atvejį. Jei programinė įranga neatitinka specifikacijos, bus pildoma klaidos ataskaita.

9 lentelė. Priėmimo testavimo testinių duomenų rinkinių forma

PA ID	Priimtas/nepriimtas	Klaida
...	...	...

## 2.5.4. Naudotojo sąsajos testavimas

Naudotojo sąsajos testavimo metu yra tikrinama ar sąsaja atitinka numatytus sąsajos reikalavimus, ar naudotojui yra patogiu dirbti su programa. Šio testo tikslas yra įsitikinti, kad naudotojo sąsaja atitinka reikalavimus ir yra patogi naudojimui.

10 lentelė. Naudotojo sąsajos testinių duomenų rinkinių forma

NR.	Objektas	Laukiami rezultatai	Klaidos
...	...	...	...

## 2.5.5. Testavimo rezultatai

Visos testavimo procedūros pateiktos 2 Priede.

Testavimo metu buvo atliktas sistemos vienetų testavimas, integravimo testavimas, priėmimo testavimas ir naudotojo sąsajos testavimas.

- Vienetų testavimas

Vienetų testavimo metu buvo atliktas struktūrinis ir funkcinis vienetų testavimas. Komponentus/modulius testavo tie patys asmenys, kurie juos sukūrė. Atlikti testavimo veiksmai parodė, kad programą testuojant tiek funkciškai, tiek struktūriškai klaidų nerasta. Taip pat buvo atliktas automatinis vienetų testavimas, kuris klaidų neaptiko.

- Integravimo testavimas

Integravimo testavimo metu buvo atliktas sistemos veikimo testavimas, kuris leidžia teigti, kad sistema teisingai veikia, kai prie sistemos tuo pačiu metu yra prisijungę 50 naudotojų, bei kai duomenų bazės dydis siekia 30 GB. Taip pat buvo atliekamas funkcinis sistemos testavimas pagal PA scenarijus ir saugumo testavimas. Saugumo testavimo metu buvo atlikti testavimai su vartojais, kurie gali/negali prisijungti prie sistemos, vartotojais, kurie tik ką neteko teisės prisijunti ar tik ką ją įgijo. Sistema atitinka reikiamą saugumo lygį, nes nesklaidumų neaptikta.

- Priėmimo testavimas

Priėmimo testavimas buvo vykdomas remiantis panaudojimo atvejais ir tikrinama, ar visi panaudojimo atvejai atitinka specifikacijas, bei tenkina vartotoją. Priėmimo testavimą vykdė atstovas iš sistemos projektavimo grupės ir sistemos naudotojas. Šio testavimo metu išaiškėjo nauji naudotojų poreikiai, kurie galės būti įdiegti tolimesnėse sistemos versijose.

- Naudotojo sąsajos testavimas

Naudotojo sąsajos testavimo metu buvo atlikta visų laukų, mygtukų, duomenų užkrovimo, vientisumo, funkcijų, bei aiškumo patikrinimas. Daugiausia klaidų aptikta vykdant naudotojo sąsajos aiškumą.

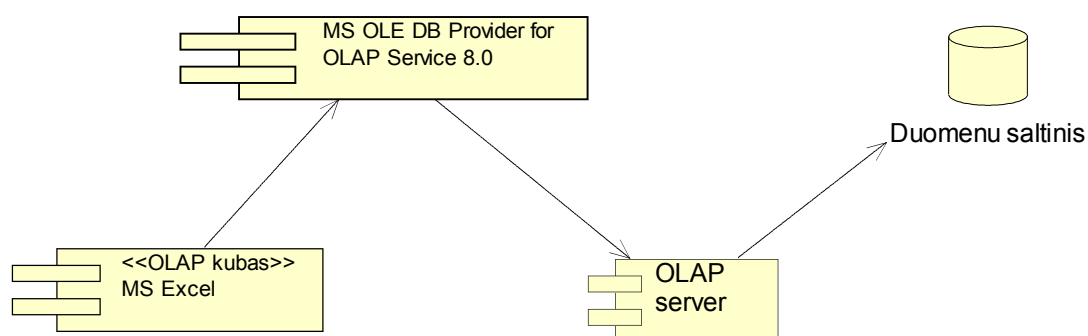
Vykdamas sistemos naudotojo sąsajos testavimą, buvo aptikta keletą neaiškių sistemos vartotojui punktų. Pavyzdžiui, jungiantis vartotojui prie serverio, reikia įvesti tik serverio vardą, o vartotojo duomenų nėra reikalaujama. Taip pat vartotojai importuojant OLAP kubą dažnai nepasirenka tinkamo darbo lapo ir klaidingai veda duomenis. Atsidarius importavimo langą, pagal nutylėjimą sistema pasiūlo „databases“ importavimą, o vartotojas turi pasirinkti „OLAP Cubes“: Todėl šios problemos yra akcentuotos vartotojo vadove.

### 2.5.6. Išvados

Testavimo metu buvo išsiaiškinta, kad sistema atitinka sistemos saugumo, funkcionalumo, vartotojo sąsajos naudojamumo lygį. Taip pat išaiškėjo, kad kai kurie pradžioje aprašyti ir specifiuoti vartotojo norai kinta. Nors ir buvo aptikta keletą sistemos klaidų, tačiau didelių sistemos neatitikimų nerasta atlikus sistemos testavimą.

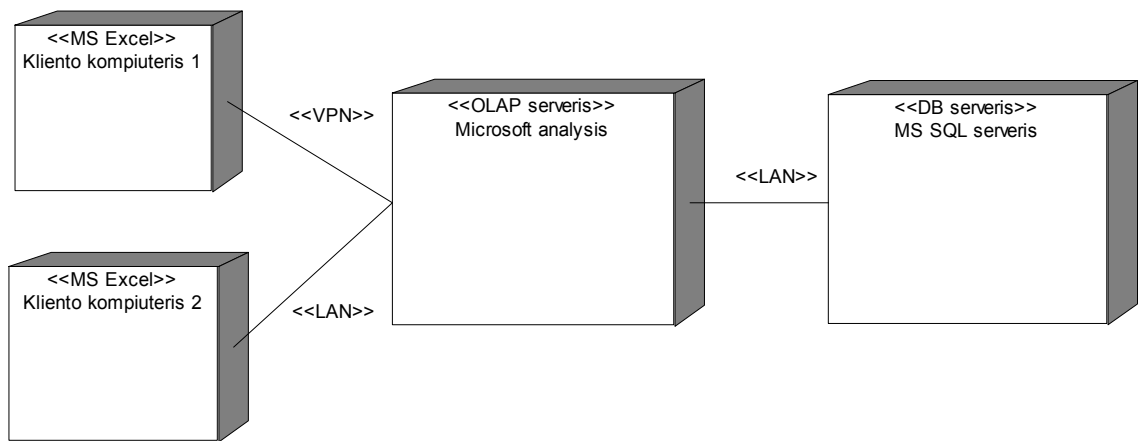
## 2.6. Realizacijos modelis

Komponentų diagrama rodo fizinę sistemos vaizdą: komponentus bei jų tarpusavio priklausomybes. Įdiegimo diagramoje (20 pav.) atvaizduojami procesoriai, kurių pagalba sistema veikia. Procesoriai – tai specialios paskirties kompiuteriai (serveriai) arba paslaugos (servisai).



19 pav. Sistemos komponentų diagrama

Sistemos išdėstymo diagrama:



20 pav. Sistemos įdiegimo diagrama

## 2.7. Kokybės tyrimas

Šiame skyriuje pateikiama projekto kokybės vertinimo ataskaita. Ataskaitoje pateikiami darbo kokybės vertinimo tikslai, procesas, rezultatai bei išvados.

### 2.7.1. Realiai atlikto darbo kokybės analizės tikslai.

- Aptikti klaidas sistemos funkcionavime, logikoje, realizacijoje.
- Patikrinti ar sistema atitinka reikalavimų specifikaciją.
- Įvertinti ar sistema atitinka užsibrėžtą kokybės lygį.

#### Funkcionalumo tyrimas

Tyrimo eigoje programų sistema naudojosi įvairaus lygio vartotojai. Buvo siekiama patikrinti sistemos funkcionalumo atitikimą vartotojo reikalavimams. Vartotojams tyrimo metu buvo pateiktas sistemos funkcijų sąrašas, kuriame jie pažymėjo programų sistemos atitikimą specifikacijai. Gauti tyrimo rezultatai patvirtino programų sistemos funkcionalumo atitikimą sistemos specifikacijai.

**Korektiškumas** – tai kokybės faktorius, atspindintis programos atitikimą specifikacijai ir vartotojo poreikių tenkinimą. Remiantis bandymų rezultatais, galima teikti, sistema yra korektiška.

**Saugumas** nusako autorizuotą priėjimą prie programinės įrangos ar jos duomenų. Priėjimą prie sistemos turi tik tie vartotojai, kuriems yra suteikta teisė. Teisės suteikia/atima sistemos administratorius, taip pat bet kada jau gali keisti. Todėl saugumas vertinamas kaip aukštas.

**Efektyvumas** apibrėžia santykį tarp kompiuterinių resursų, reikalingų programinei įrangai funkcionuoti ir programinės įrangos įmonėms teikiamos naudos. Projektas sukurtas taip, kad galutiniam vartotojų nereikia įdiegti papildomos programinės įrangos, užtenka standartinių operacinės sistemos priemonių (MS Excel ), bei interneto tinklo. Todėl efektyvumas vertinamas kaip aukštas.

**Sąsajos paprastumas** įvertinamas kaip pastangos, reikalingos išmokti dirbti su sistema, ja naudotis, išmokti paruošti duomenis bei interpretuoti rezultatus. Kvalifikuotam vartotojui užtenka perskaityti vartotojo vadovo dokumentą ir jis yra pasiruošęs vykdyti veiklą. Pradinis prisijungimas prie sistemos atliekamas tik vieną kartą, o po to lieka tik ja naudotis.

### **Standartų laikymasis**

Programų sistemos kūrimo procesas aprašytas projektinėje dalyje.

### **Atitikimas reikalavimų specifikacijai**

Norint sulaukti produkto pripažinimo, pirmiausiai sukurtas produktas turi idealiai veikti, atlikti tinkamus darbus tinkamu metu ir be abejonės tenkinti reikalavimų specifikaciją. Kad programinė įranga būtų be klaidų yra labai sunkiai pasiekama. Didelė kokybės dalis gali būti priskiriama testavimui, tai pasiekti padeda detalus testavimo etapo planavimas. Detaliame testavimo plane pabrėžiama, kokius objektus, duomenis, limitus, ribas ir suvaržymus būtina ištestuoti.

## **2.7.2. Kokybės vertinimo procesas.**

### **2.7.2.1. Peržiūros**

Norint įsitikinti ar sistema atitinka kokybės reikalavimus, buvo atlikta keletą sistemos peržiūrų. Peržiūra vyko keliais etapais:

- Reikalavimų specifikacijos peržiūra
- Architektūros specifikacijos peržiūra
- Detalios architektūros peržiūra
- Testavimo plano peržiūra
- Programinio kodo peržiūra

### **2.7.2.2. Interviu su užsakovu.**

Interviu metu aptariami reikalingi patobulinimai, sukuriamas pakeitimų sąrašas. Interviu metu užsakovui buvo pateikta bandomoji sistemos versija. Kartu su užsakovu vyko sistemos peržiūra ir buvo bandyta išsiaiškinti ar sistema atitinka užsakovo

reikalavimus. Taip pat buvo pateiktas papildomų reikalavimų sąrašas. Bei pageidaujamų pakeitimų sąrašas.

### 2.7.2.3. Projektavimo komandos narių peržiūrų aprašymas.

Programa buvo pateikta keletai šios srities specialistų, kurių buvo paprašyta išreikšti savo nuomonę atitinkamais programos kokybės vertinimo aspektais.

### 2.7.2.4. Apklausų anketos

Pateikus programos bandomąją versiją buvo pateikta daliai programos vartotojų užpildyti anketą (žiūrėti 2 Priedas). Paaiškėjo, kad naudojantis šia programa darbuotojai sutaupo laiką; domisi platesne įmonės veiklos sritimi, o ne tik tai kas tiesiogiai susiję su jo pareigomis; daugelis darbuotojų labai greitai įsisavino šią sistemą, tik vienetai atsakė, kad jiems tai užtruko apie keletą savaičių; didžioji dalis darbuotojų pateikė papildomų pageidavimų dėl sistemos tobulinimo.

### 2.7.2.5. Vertinimo rezultatai.

Projekto kokybę įvertinta pagal šiuos parametrus:

11 lentelė. Projekto kokybės vertinimo parametrai

Eil. Nr.	Parametras	Aprašymas
1.	Korektiškumas	Ar sistema pilnai realizuota? Ar tinkama naudojimui?
2.	Išplečiamumas	Galimybė praplėsti programinės įrangos funkcijas.
3.	Panaudojamumas	Ar lengva išmokti dirbti su programine įranga.
4.	Palaikomumas	Ar tai palaikoma vystoma sistema?
5.	Patikimumas	Teisingas sistemos darbas ilga laiką
6.	Greitaveika	Didelis duomenų išgavimo greitis

### Išvados

Galime teigti, kad :

- produktas atitinka užsibrėžtus kokybės reikalavimus;
- bei reikalavimų specifikacijoje apibrėžtus reikalavimus.

### 2.7.3. Išvados

Sudaryta sistemos architektūra tenkina pagrindinius kokybės reikalavimus:

- Sistema užtikrins duomenų saugumą.
- Sistemos naudotojui prieinami duomenys priklausys nuo vartotojo pareigų įmonėje.
- Sistemos funkcijos bus prieinamos iš bet kokios įmonės darbo vietos.
- Taip pat ši sistema bus funkcionali 24 valandas per parą, 7 dienas per savaitę.
- Aprašyta sistemos kūrimo strategija, paremta komponentiniu kūrimu, leidžia ateityje lengvai papildyti sistemą naujais komponentais. Apibrėžta duomenų sąsaja, naudojant XML formatą, leidžia atsiradus poreikiui su minimaliomis pastangomis prijungti naujus duomenų šaltinius.

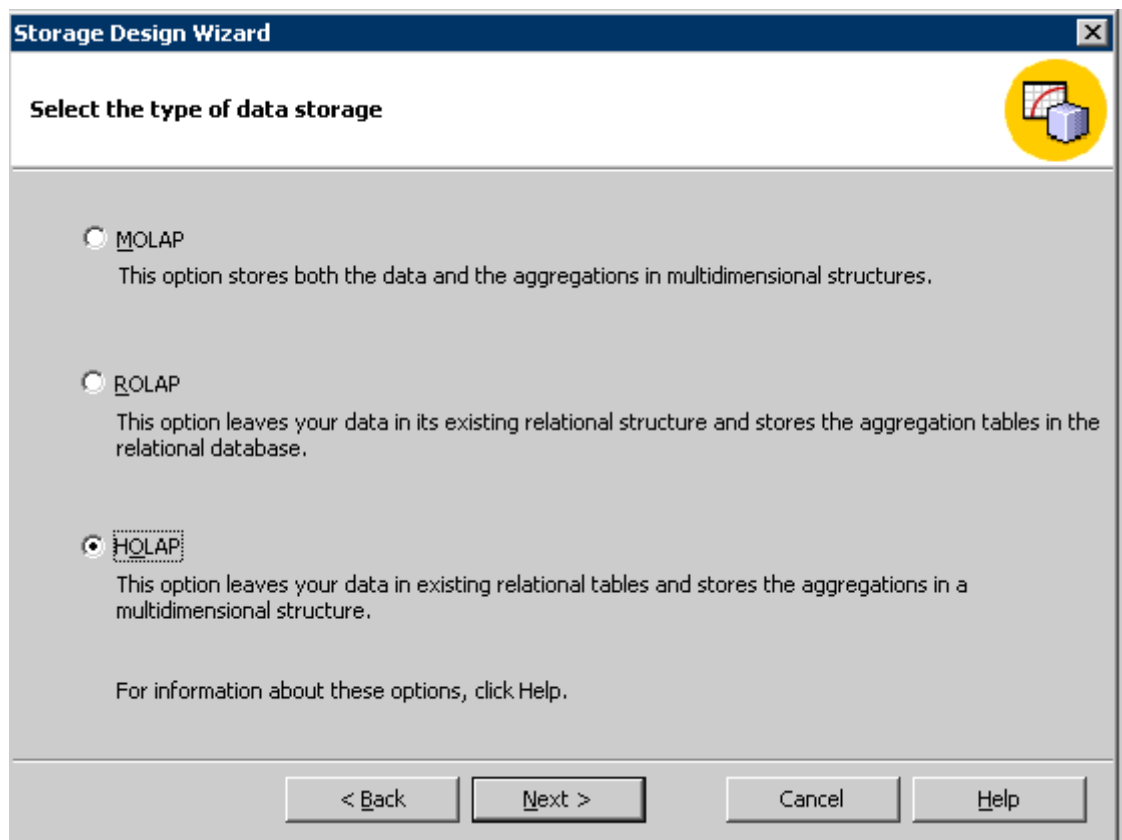
### 3. OLAP KUBŲ ARCHITEKTŪROS EKSPERIMENTINIS TYRIMAS

Naudojant MS Analysis Services įrankį buvo sudaryta skirtingų OLAP architektūrų kubai ir atliktas jų tyrimas.

Analysis Services palaiko tris pagrindines duomenų saugojimo architektūras:

- MOLAP – ir detalūs ir konsoliduoti duomenys saugomi daugiamatėje struktūroje. Šiuo atveju užtikrinamas trumpiausias užklausų aptarnavimo laikas;
- ROLAP – konsoliduoti ir detalūs duomenys saugomi reliacinėje struktūroje, be to detalūs duomenys gali būti išgaunami tiesiog iš pradinio duomenų šaltinio (duomenų sandėlio);
- HOLAP – detalūs duomenys saugomi reliacinėje saugykloje, o konsoliduoti duomenys – daugiamatčiame kube. Šiuo atveju kubo užimama atmintis mažesnė nei MOLAP atveju, o užklausos aptarnaujamos greičiau nei ROLAP atveju.

Tai viena nedaugelio sistemų, kurios leidžia pasirinkti iš trijų duomenų saugojimo architektūrų:



21 pav. Analysis Services įrankio kubų architektūros pasirinkimo vedlys

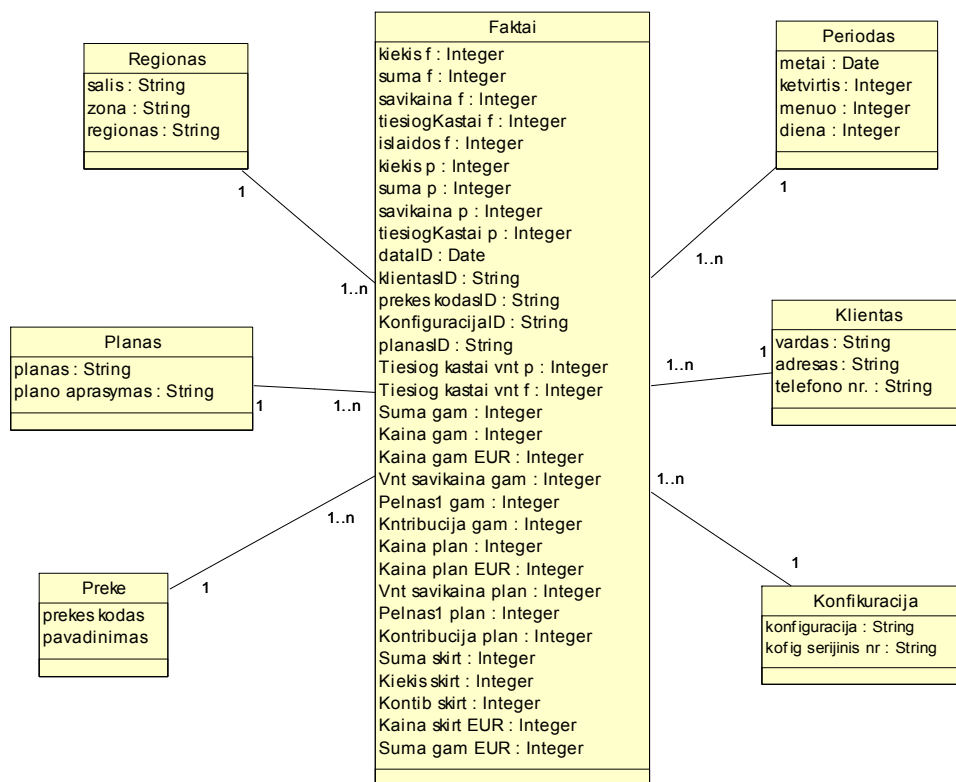


Tai įgalina naudoti Analysis Services įvairiose probleminėse srityse. Tačiau duomenys kubui išgaunami iš duomenų sandėlio, todėl norint užtikrinti korektišką kubo struktūrą, pats duomenų šaltinis turi būti teisingai suprojektuotas, priešingu atveju kubo projektavimas tampa sudėtingas. Todėl duomenys iš tradicinės reliacinės DB yra transformuojami į „Žvaigždės“ ar „Snaigės“ tipo architektūrą.

### 3.1. „Žvaigždės“ ir „Snaigės“ tipo duomenų schemas

Žvaigždės schema - kai visos dimensijų lentelės tiesiogiai sujungtos su faktų lentele.

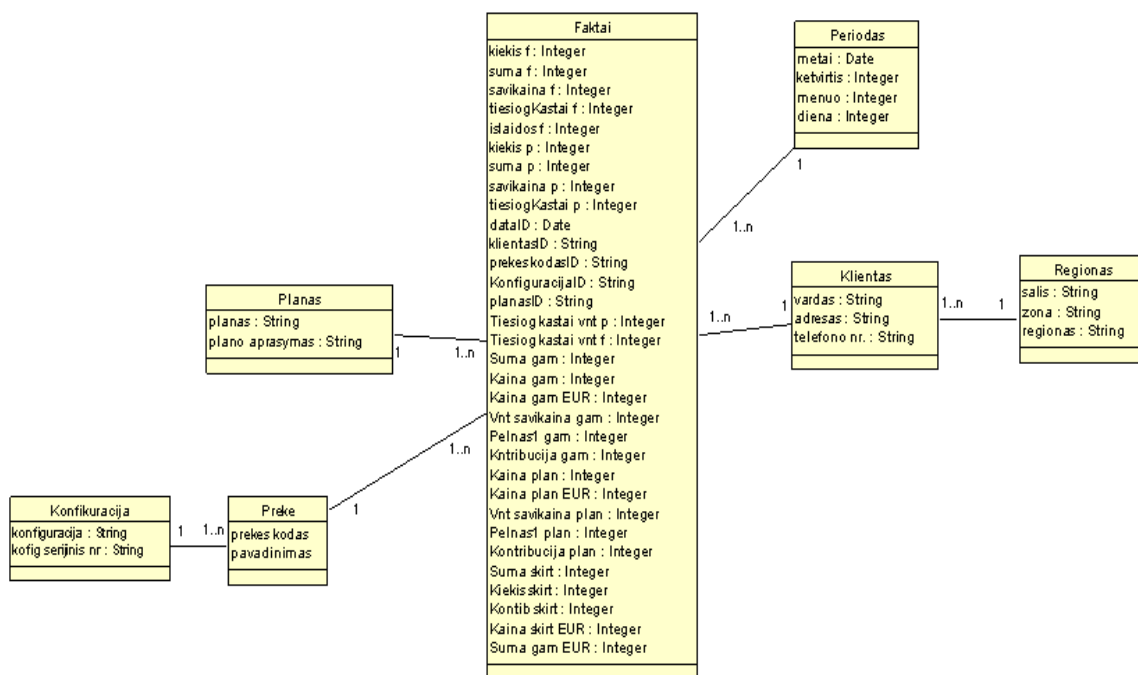
Duomenys buvo transformuoti į „Žvaigždės“ tipo duomenų schemą. Sekančiame paveiksle yra parodyta „Žvaigždės“ tipo duomenų schema iš kurios duomenis ima OLAP kubas.



22 pav. Žvaigždės tipo duomenų schema

Snaigės schema - kai viena ar kelios dimensijų lentelės nėra tiesiogiai sujungtos su faktų lentele, o siejasi per kitas dimensijų lenteles.

Sekančiame paveiksle pavaizduota „Snaigės“ tipo duomenų architektūra, kuri naudojama OLAP kube.



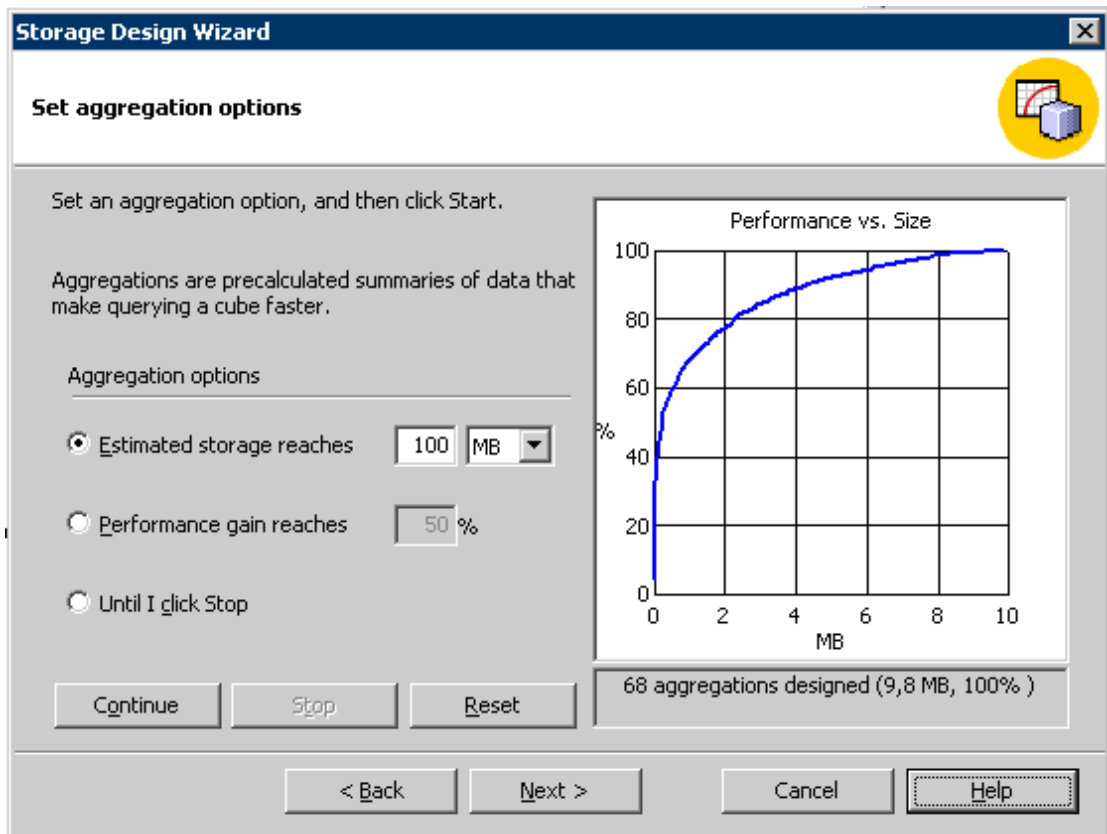
23 pav. Snaigės tipo duomenų schema

Patys duomenų kubai gali būti saugojami naudojant įvairias OLAP architektūras (ROLAP, MOLAP ar HOLAP).

### 3.2. MOLAP architektūra

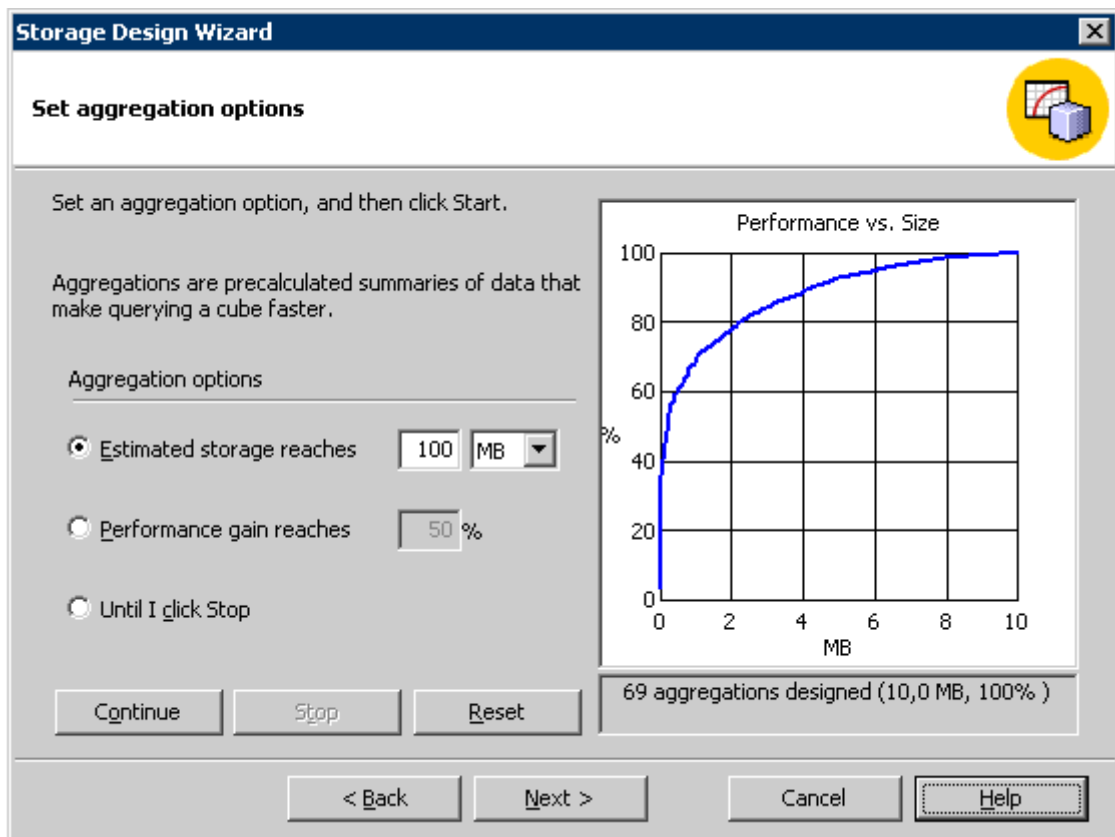
MOLAP (multidimensional OLAP) tai duomenų analizės priemonė paremta duomenų kubo modeliu. MOLAP saugo jau susistemintus kubo duomenis, darant užklausą nereikia kiekvieną kartą skaičiuoti kubo reikšmių, todėl MOLAP priemonės pasižymi greitu atsakymu į užklausą. Tačiau į MOLAP kubą sudėtingiau pridėti naujas dimensijas, be to, kube nesaugomi detalizuoti duomenys, todėl negalima pasinaudoti „Drill - In“ galimybėmis.

MOLAP architektūros kubo, kuris ima duomenis iš „Žvaigždės“ tipo duomenų schemas, apdoravimo charakteristikos pavaizduotos sekančiame paveiksle.



24 pav. MOLAP architektūros charakteristikos, kai naudojama „Žvaigždės“ tipo duomenų schema

MOLAP architektūros kubo, kuris ima duomenis iš „Snaigės“ tipo duomenų schemas, apdorojimo charakteristikos pavaizduotos sekančiame paveiksle.



25 pav. MOLAP architektūros charakteristikos, kai naudojama „Snaigės“ tipo duomenų schema

### **3.2.1. MOLAP pranašumai ir trūkumai**

#### **Pranašumai:**

- Puikus veikimas – MOLAP kubai greitai išgauna duomenis ir yra optimalūs informacijos išrinkimo ir pateikimo procese.
- Gali vykdyti daug sudėtingų skaičiavimų. Visos reikšmės yra išgaunamos sukūrus kubą.

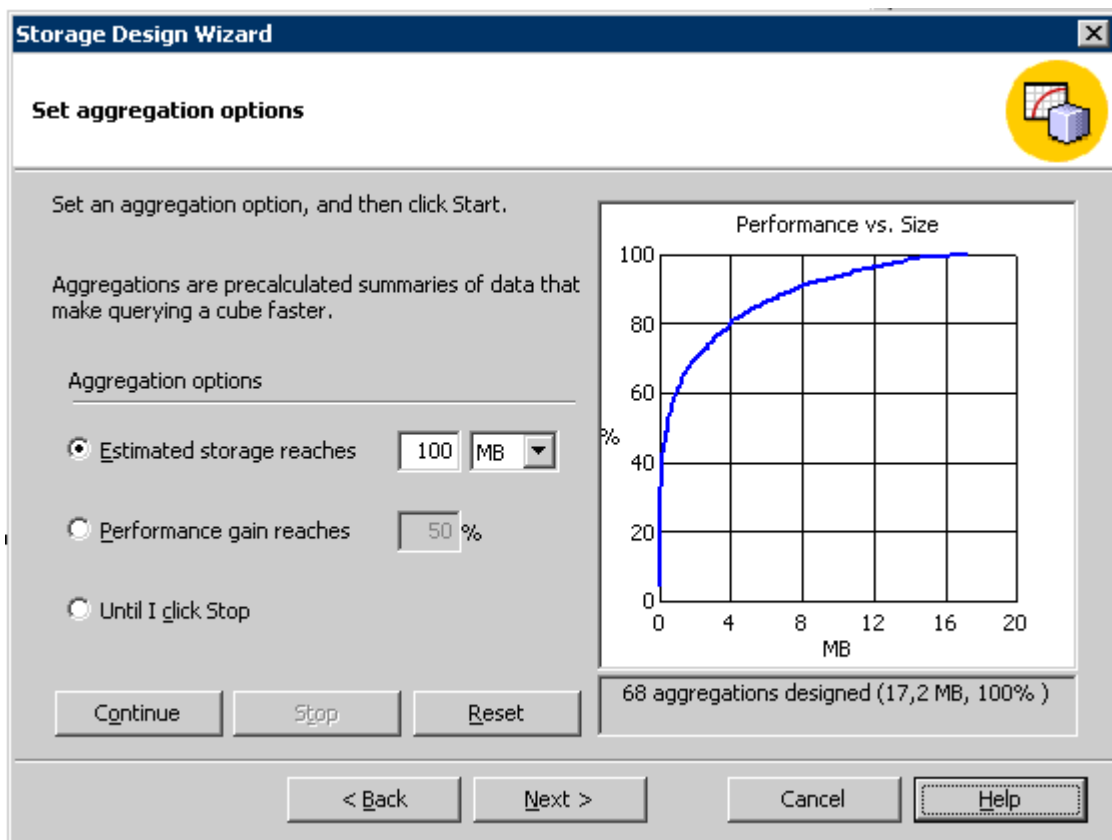
#### **Trūkumai:**

- Lėtai vykdo užklausas su neskaičiuojamais duomenimis.
- Ribotas duomenų kiekis, kadangi visos įmanomos reikšmės jau yra apdorotos jo kūrimo procese ir saugomos kube.
- Reikalauja papildomų investicijų – kubo technologijos dažnai reikalauja papildomų turto ir žmoniškojo kapitalo investicijų.

### **3.3. *ROLAP architektūra***

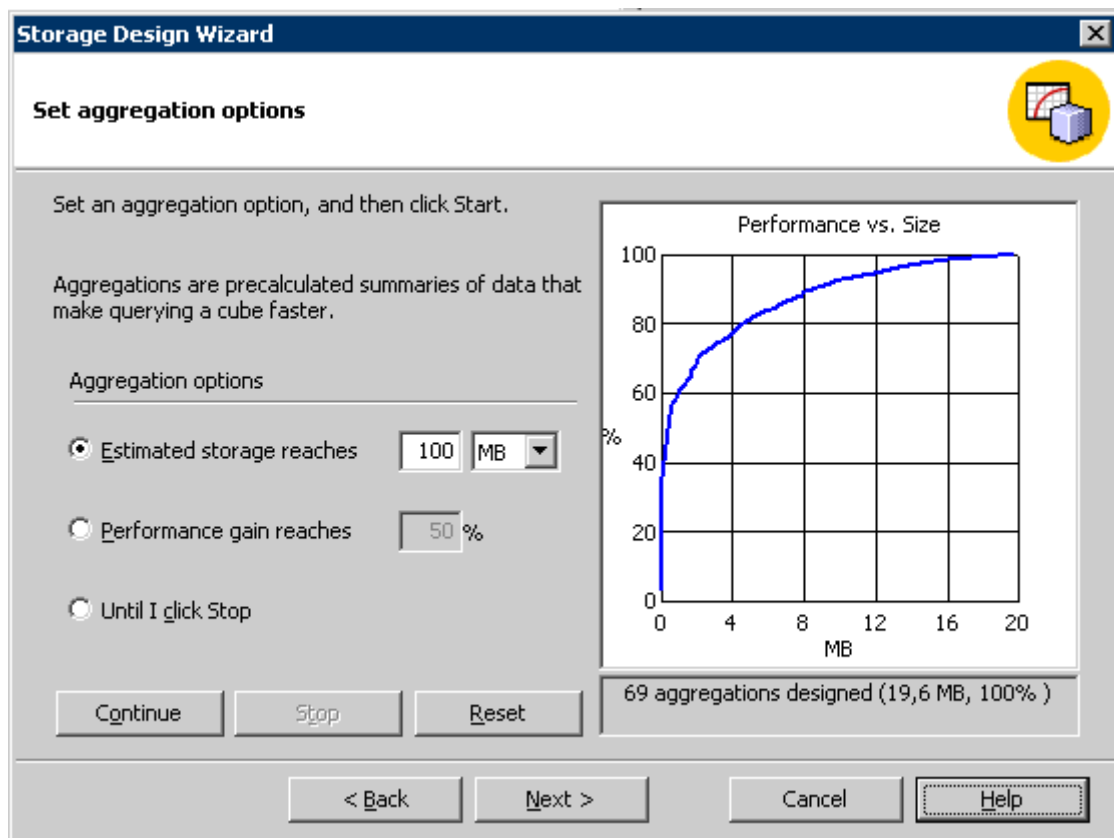
ROLAP veikia reliacinės duomenų bazės pagrindu kur sukuriamos užklausos, formuojančios duomenų kubus. Kiekvieną kartą analizuojant kubo duomenis, kubo reikšmės perskaičiuojamos. ROLAP užklausos nėra tokios greitos kaip MOLAP, ypač dirbant su dideliais duomenų kubais, tačiau čia saugoma detali informacija ir galima pasinaudoti „Drill - In“ galimybėmis. ROLAP duomenų kubą paprasčiau papildyti naujomis dimensijomis.

Sekančiame paveiksle pavaizduota ROLAP architektūros kubo vykdomo charakteristikos.



26 pav. ROLAP architektūros charakteristikos, kai naudojama „Žvaigždės“ tipo duomenų schema

ROLAP architektūros kubo, kuris ima duomenis iš „Snaigės“ tipo duomenų schemas, apdorojimo charakteristikos pavaizduotos sekančiame paveiksle.



27 pav. ROLAP architektūros charakteristikos, kai naudojama „Snaigės“ tipo duomenų schema

### 3.3.1. ROLAP pranašumai ir trūkumai

#### Pranašumai:

- Duomenys saugomi standartinėje reliacinėje duomenų bazėje todėl galima panaudoti SQL ataskaitų (Reporting) įrankius (Įrankiai nebūtinai turi būti OLAP įrankiai);
- ROLAP priemonės leidžia lanksčiau valdyti didelius duomenų kiekius (milijonai įrašų);
- ROLAP įrankiai geriau apdoroja neskaičiuojamas reikšmes (non-aggregatable facts), pvz. tekstinius duomenis. MOLAP priemonės lėtai vykdo užklausas su tokiais duomenimis;
- Gali įtakoti susietos duomenų bazės būdingą funkcionalumą. Dažnai, susieta duomenų bazė jau turi savo funkcionalumą. Kadangi ROLAP technologijos yra labiau veikiantis veiksnys nei duomenų bazės, tai gali įtakoti šį funkcionalumą.

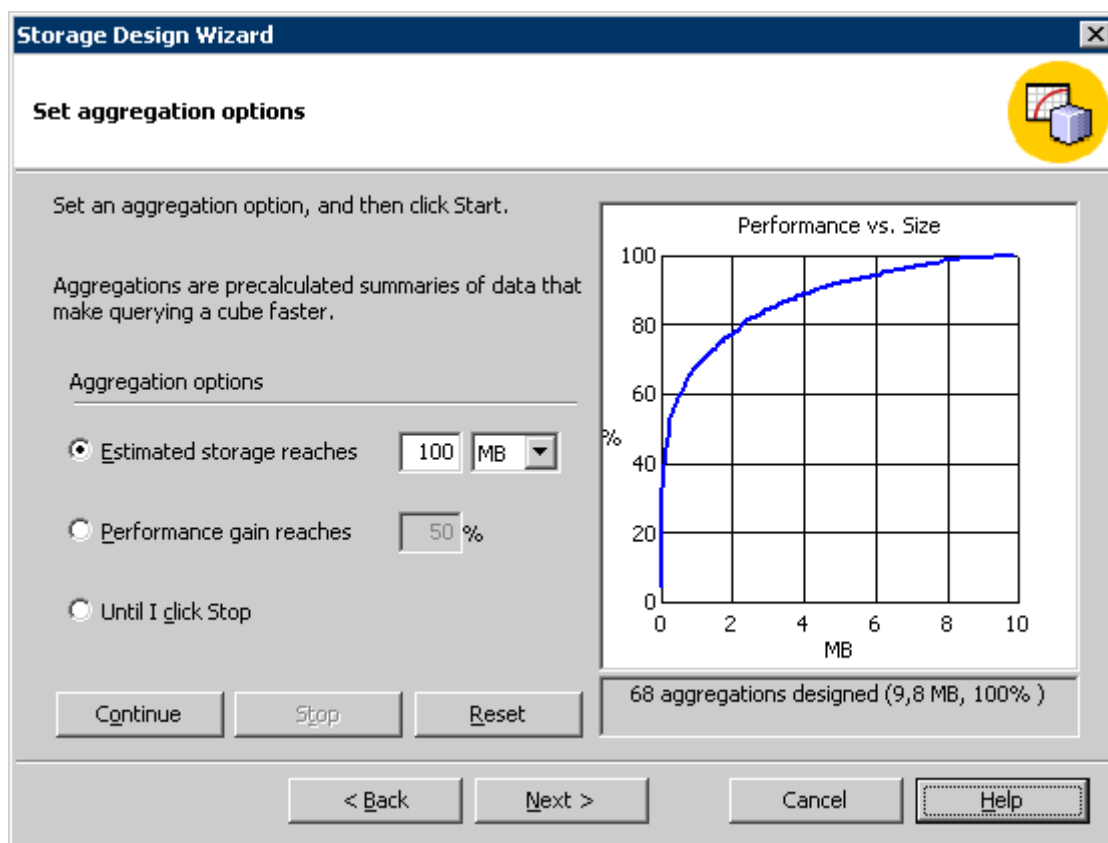
#### Trūkumai:

- ROLAP lėčiau veikia esant dideliame duomenų kiekiui;
- Dauguma ROLAP priemonių praleidžia skaičiuojamųjų reikšmių lentelių kūrimą, tada nukenčia užklausos vykdymo greitis nes reikia apdoroti didesnius duomenų kiekius.

- ROLAP naudoja bendros paskirties duomenų bazes, kurios neturi specialų MOLAP naudojamų priemonių (pvz. hierarchinis indeksavimas). Tačiau modernios ROLAP priemonės jau turi SQL kalboje numatytus specialius operatorius (CUBE, ROLLUP).
- ROLAP naudoja SQL užklausas visiems skaičiavimams, todėl ROLAP įrankiai netinka sudėtingiems modeliams, kuriuos aprašyti SQL kalba labai sudėtinga:

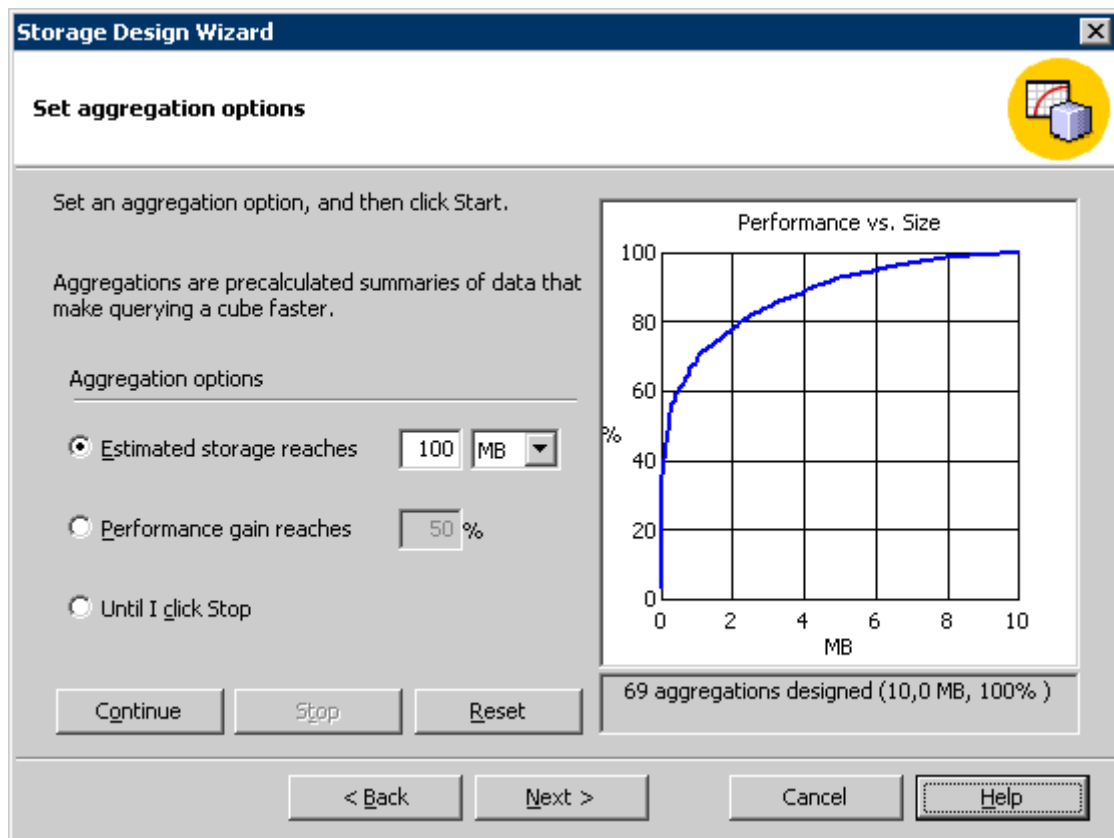
### 3.4. HOLAP architektūra

HOLAP technologijos, tai mėginimas suderinti MOLAP ir ROLAP privalumus. Suglausto tipo informacija ir HOLAP kubo įtakos technologija greitesniam veikimui. Kai reikalinga kažkokia detali informacija, HOLAP gali kreiptis iš kubo į bendros paskirties duomenų bazes ir išgauti reikiamą informaciją.



28 pav. HOLAP architektūros charakteristikos, kai naudojama „Žvaigždės“ tipo duomenų schema

HOLAP architektūros kubo, kuris ima duomenis iš „Snaigės“ tipo duomenų schemos, apdorojimo charakteristikos pavaizduotos sekančiame paveiksle.



29 pav. HOLAP architektūros charakteristikos, kai naudojama „Snaigės“ tipo duomenų schema

### 3.5. Išvados

Tyrimo metu buvo sudaryta „Snaigės“ ir „Žvaigždės“ tipo duomenų schemas ir pagal kiekvieną iš jų sukurti trijų skirtingų architektūrų OLAP kubai. Tarpusavyje palyginta MOLAP, ROLAP ir HOLAP architektūros duomenų apdorojimo priklausomybė nuo agreguotų duomenų kiekio atsižvelgiant į duomenų schemas struktūra, iš kurios imami kubo duomenys.

Kai naudojama „Snaigės“ tipo duomenų schema, nepriklausomai nuo OLAP architektūros, agreguotų duomenų kiekis yra didesnis, nei naudojant „Žvaigždės“ tipo schemą. Lyginat tarpusavyje OLAP architektūras MOLAP ir HOLAP architektūras agreguotų duomenų kiekis nesiskiria, tuo tarpu ROLAP architektūros duomenų kiekis beveik padvigubėja.

Remiantis atliktu tyrimu, galima teigti, kad geriausia naudoti „Žvaigždės“ duomenų modelį. Renkantis OLAP kubo architektūrą patartina naudoti MOLAP arba HOLAP architektūras.



## IŠVADOS

Duomenų saugyklos modelių analizės dalyje buvo atlikta egzistuojančių problemų ir galimų sprendimo variantų apžvalga. Taip pat atlikta OLAP duomenų saugyklos funkcionalumo analizė, apžvelgti architektūros modeliai. Paaiškintos OLAP kubų pjūvių ir skaičiuojamų reikšmių sąvokos. Apžvelgta duomenų gavybos funkcionalumas ir pagrindiniai metodai. Atlikta MS SQL Server 2000 Analysis Services ir Oracle Express OLAP įrankių funkcionalumo palyginimas. Apibendrinant Analysis Services teikiamas galimybes duomenų modelių sudarymui, galima teigti, kad ši sistema lanksti ir tai užtikrina optimalų duomenų saugyklos veikimą (saugyklos apimtis, užklausų aptarnavimo greitis).

Ištyrus vartotojų poreikius OLAP duomenų saugyklos modelio sudarymo dalyje, buvo apibrėžtas kuriamos sistemos reikalavimų modelis. Taip pat pateiktos pagrindinių panaudojimo atvejų sekų diagramos, bei duomenų saugyklos modelio detalus aprašymas ir realizacijos diagramos. Modelio testavimo dalyje detalai aprašytos testavimo procedūros, bei pateikti testavimo rezultatai, bei išvados. Taip pat atliktas kokybės tyrimas, kuris patvirtino, kad realizuota sistema atitika reikalavimų specifikacijoje aprašytus reikalavimus.

OLAP kubų architektūrų tyrimo dalyje buvo sudarytos „snaigės“ ir „žvaigždės“ tipo duomenų schemas ir pagal kiekvieną iš jų MOLAP, ROLAP ir HOLAP architektūros kubai. Atlikta skirtingų architektūrų duomenų apdorojimo priklausomybė nuo duomenų kiekio, atsižvelgiant į duomenų schemas struktūrą, iš kurios imami kubo duomenys. Remiantis atlikto tyrimo rezultatais, rekomenduojama „Žvaigždės“ tipo duomenų schema ir MOLAP arba HOLAP duomenų kubo architektūra.

Sudarytas OLAP duomenų saugyklos modelis pasižymi greitu informacijos išgavimu iš didelių duomenų kiekiu ir patogia vartotojo sąsaja. Realizuotas OLAP kubas yra įdiegtas gamybinėje įmonėje ir apima logistinius ir gamybinius procesus. Taip yra sutaupomas darbuotojų laikas, padidėja darbo efektyvumas, atliekant detalesnę ir gilesnę duomenų analizę yra didinamas įmonės pelningumas. Be to jį galima panaudoti kitose gamybinėse įmonėse.

Šio darbo metu atlikta OLAP duomenų saugyklų funkcionalumo analizė aprašytas straipsnyje, kuris buvo paskelbtas magistrantų ir doktorantų konferencijoje „Informacinė visuomenė ir universitetinės studijos 2007“ ir išspausdintas konferencijos medžiagoje.

## Terminų ir santrumpų žodynas

**OLTP** – transakcinės sistemos (angl *Online transaction processing*)

**OLAP** – analitinės sistemos (angl *Online analytical procesing*)

**MOLAP** – angl. Multidimensional OLAP

**ROLAP** – angl. Relacional OLAP

**HOLAP**- angl. OLAP híbrido

**VPN** – virtualus privatus tinklas(angl *Virtual private network*)

**RDBVS** - reliacinės duomenų bazių valdymo sistemos

**IT** – informacinės technologijos

**PĮ** – programinė įranga

**Dimensija** - Denormalizuota lentelė, gaunama iš grupuojančių duomenų ir skirta kubo analizei

**MS** – angl. *Mocrosoft*

**WWW** - pasaulinis interneto tinklas (angl *World Wide Web*)

**XML** yra W3C rekomenduojama bendros paskirties ženklinimo kalba, skirta specialios paskirties ženklinimo kalbų kūrimui bei duomenų keitimuisi tarp skirtingų sistemų (angl *eXtensible Markup Language*).

**VVS** – Verslo valdymo sistema

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

**DBVS** – duomenų bazės valdymo sistema

**OS** – operacinė sistema (angl *Operating system*)

**HTML** - hiperteksto žymėjimo kalba. Tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete (*Hypertext Markup Language*)

## LITERATŪRA

- [1] E. Sysojevas. Interaktyvios duomenų analizės įrankiai: OLAP duomenų bazės, Naujoji komunikacija 2003m. Nr6
- [2] Žinių technologijų pagrindinių terminų žodynelis, S. Maskeliūnas, Vilnius 2006  
[interaktyvus] , [žiūrėta 2007-03-07] Prieiga per internetą:  
[http://www.mii.lt/files/mii\\_prep\\_2006\\_35.doc](http://www.mii.lt/files/mii_prep_2006_35.doc)
- [3] Naujausios publikacijos, Duomenų bazės. [interaktyvus] 2006,  
[žiūrėta 2006-11-12] Prieiga per internetą: <http://itekspertas.projektas.lt/>
- [4] Verslo analitika [interaktyvus], [žiūrėta 2007-03-07] Prieiga per internetą:  
[http://www.dbit.lt/intelligence\\_BO1.htm](http://www.dbit.lt/intelligence_BO1.htm)
- [5] L.Kalinauskienė, OLAP sistemų projektavimas žinių modelio pagrindu: Informacinių sistemų projektavimo metodai ir technologijos, 146-150psl.
- [6] R. Šileikienė Duomenų gavyba – kas tai? Informacinės technologijos, 2000, 13-16psl.
- [7] Data Warehouse Information Centre. [interaktyvus] 2006 , [žiūrėta 2007-02-15],  
Prieiga per internetą: <http://www.dwinfocenter.org>
- [8] Planning Anglysis Reporting Informatikon System, Solutions. [interaktyvus] 2006,  
[žiūrėta 2007-03-12] Prieiga per internetą: <http://www.olap.com/>
- [9] Vikipedija, laisvoji enciklopedija. [interaktyvus] [žiūrėta 2007-04-15]  
Prieiga per internetą: <http://lt.wikipedia.org>
- [10] Brief XML tutorial. [interaktyvus] 2006, [žiūrėta 2006-03-07]  
Prieiga per internetą: <http://www.xml.lt/>
- [11] A.Sabalys, OLAP sistemų MS SQL Server 2000 Analysis services ir Oracle Express palyginimas: Informacinių sistemų projektavimo metodai ir technologijos 151-156psl.
- [12] S.Goil, A.Choudhary, „High Performance Multidimensional Analysis and Data Mining“ .
- [13] Introducing the SQL Server 2000 [Žiūrėta 2005 10 25], prieiga per internetą  
<http://www.microsoft.com/technet/prodtechnol/sql/2000/reskit/part5/c1761.mspxf>
- [14] Leased Lines and Private Networks. [interaktyvus] 2005, kovas [žiūrėta 2005-05-05]  
Prieiga per internetą: [http://www.flexwork.eu.com/members/tech\\_brief/tb20.pdf](http://www.flexwork.eu.com/members/tech_brief/tb20.pdf)
- [15] OLAP [interaktyvus]. Prieiga per internetą: <http://www.olap.ru/basic/home.asp>
- [16] Secect (T\_SQL) [interaktyvus]. Prieiga per internetą:  
[http://doc.ddart.net/mssql/sql70/sases\\_3.htm#\\_select\\_clause](http://doc.ddart.net/mssql/sql70/sases_3.htm#_select_clause)

- [17] „Cognos“ produkto informacinis puslapis [Žiūrėta 2005 11 10], prieiga per internetą <http://www.cognos.com/products/index.html?lid=//Products//Overview>
- [18] J.Tomkūnaitė, L. Nemuraitė „Duomenų saugyklos konceptų modelis“, Kauno technologijos universitetas, 2005
- [19] The Microsoft Data Warehouse Toolkit, The Toolset, © 2006 by Wiley Publishing
- [20] Microsoft Dynamics AX overview, 2007 [interaktyvus], [žiūrėta 2007-01-23], Prieiga per internetą: <http://www.itpiast.com/solution1.html>
- [21] The Microsoft Data Warehouse Toolkit Second Edition, © 2006 by Wiley Publishing, 16- 27 psl.
- [22] T.Bain, M.Benkovich ir kiti, „Professional SQL Server 2000 Data Warehousing with Analysis Srevises“ 2nd. ed., Wrox Press, 2001
- [23] S.Goil, A.Choudhary, „High Performance Multidimensional Analysis and Data Mining“ .
- [24] Data Warehouse Design Considerations [Žiūrėta 2005 11 03], prieiga per internetą <http://www.microsoft.com/technet/prodtechnol/sql/2000/reskit/part5/c1761.msp>
- [25] A.Venčkauskas, M.Mikučionis „Korporacinių įmonių informacinės saugos architektūrų modeliavimas“, „Informacijos mokslai“, 2005
- [26] W.H.Imon, OLAP and Data warehouse, 2000
- [27] Verslo analitikos naujovės ir technologijos, 2007, [interaktyvus] [žiūrėta 2007-01-05], prieiga per internetą: <http://www.olap.ru/seagate/home.asp>
- .

# 1 PRIEDAS. Straipsnis

Tarpuniversitetinė magistrantų ir doktorantų konferencija “Informacinė visuomenė ir universitetinės studijos” (IVUS‘07) 11-16psl.

## OLAP kubų taikymas korporatyvinėje įmonėje

**Laima Buškauskaitė**

*Kauno technologinis universitetas,*

*Studentų 50, LT-51368, Kaunas*

*Telefonas: 8-611-19876*

*E-paštas: laimma@one.lt*

Šiuolaikinis verslas naudoja didžiulį duomenų kiekį, tačiau įmonėje šie duomenys taip ir liks tik balastas, jeigu nesugebėsime jų išanalizuoti ir tinkamai interpretuoti. Tik duomenų analizė, naudojant specialius programinius įrankius, iš „žalios“ informacijos leis atrinkti naudingus grūdėlius ir perdirbti juos į vertingas žinias, kuriuos taps teisingų verslo sprendimų pagrindu. Naudojant OLAP (On-line Analytical Processing) priemones, sukuriama duomenų saugykla, kuri leidžia greitai, bei patogiai analizuoti duomenis. Taip pat šis produktas leidžia analizuoti duomenis, kurie yra gauti iš skirtingų verslo valdymo sistemų, kurios yra naudojamos skirtinguose geografiškai nutolusiuose įmonės padalinuose. Kas tai - OLAP? Šis terminas naudojamas norint apibūdinti programinius produktus, kurie leidžia visapusiškai analizuoti verslo informaciją realiuoju laiku. Sąveika su tokiais sistemomis vyksta interaktyviai, atsakymai net į daug skaičiavimų reikalaujančias užklausas gaunami per kelias sekundes.

OLAP sistemos yra vienas iš daugelio verslo analitikos (Business Intelligence) grupės priskiriamų produktų. Pasaulyje yra sukurta nemažai sistemų, priklausančių šiai produktų grupei: nuo paprasčiausių MS SQL OLAP kubų iki tokių sistemų kaip „Business Objects“, „Cognos“, Corporate Planner“, „Microstrategy“.

### Egzistuojančios problemos

Šiandien beveik kiekvienoje įmonėje rasime įdiegtą verslo valdymo ar finansinę sistemą, kuri dažnai yra papildoma ir kitomis IT sistemomis: gamybos valdymo, sandėlių valdymo, santykių su klientais valdymo, personalo valdymo ar kt. Visa įmonės veiklos ir verslo analizė remiasi į aukščiau paminėtų sistemų ataskaitų posistemėmis, t.y. ataskaitų rinkinius, kurias siūlo įdiegtos įvairios IT sistemos.

Iš čia kyla visa eilė kasdieninių problemų, kurias tenka sutikti dažnoje įmonėje:

- Ataskaitų ribotumas – jei nėra sukurtos ir įdiegtos reikalingos bei kompleksinės ataskaitos, įmonės analizė norimu pjūviu tampa sunkiai įvykdoma arba iš viso negalima.
- Naujos ataskaitos kuriamos neleistinai ilgai, jei iš viso yra sukuriamos. Taip pat kiekvieną kartą tenka mokėti sistemos diegėjams ar prižiūrėtojams už naujų ataskaitų kūrimą ar esamų modifikavimą.
- Ataskaitų generavimas kasdien ilgėjantis procesas, kadangi duomenų kiekis sparčiai didėja. Neretai dienomis jau negalima daryti ataskaitų, nes tai itin stabdo kitų sistemų darbą.
- Dėl didelio duomenų kiekio įmonė ir jos verslas yra analizuojamas tik makrolygiu, nes labiau detalizuoti informacijos neleidžia turimos technologijos, gaminių ir kitų analizės objektų nomenklatūra.
- Vartotojai, kuriems reikalingi įmonės duomenys, juos apdoroja įvairiausiomis priemonėmis. Todėl būtinai atsiranda klaidų, nesutapimų, skirtingo vienodų objektų traktavimo.
- Keičiant turimą finansinę sistemą kita ar apjungiant skirtingas sistemas, yra prarandama dalis įmonės istorinių duomenų bei galimybė analizuoti verslą, nes visų duomenų perkėlimas į naują finansinę sistemą – sudėtingas uždavinys.
- Turint kelias IT sistemas nėra matomas vieningas įmonės vaizdas.
- Nutolę padaliniai neturi galimybės interaktyviai analizuoti norimus duomenis.
- Nepakankamai arba jokio duomenų saugumo.

Norint spręsti šias problemas, yra reikalinga verslo duomenų analizės sistema, kuri pirmiausia apima funkcionalų duomenų ir informacijos analizės bei vizualizavimo įrankį bei technologines priemones (duomenų saugykla), suteikiančias galimybę itin sparčiai

analizuoti didelius duomenų kiekius, t.y. visą įmonę, įmonių grupę, visus įmonės procesus net iki detaliausio lygmens.

### **Duomenų saugyklos**

Kai duomenų yra nedaug, jų analizę galima atlikti pasitelkus ir tradicines reliacines duomenų bazių valdymo sistemas (RDBVS), kurios kartu naudojamos ir transakcijoms realiuoju laiku vykdyti (online transaction processing, OLTP). OLTP duomenų bazės pritaikytos tam, kad galėtų palaikyti daug vienu metu į jas besikreipiančių vartotojų, kurie intensyviai įterpia bei keičia duomenis. Paprastai tokios duomenų bazės turi sudėtingą struktūrą (didelė normalizacija), o duomenys jose parodo esamą įmonės situaciją, tačiau ne istoriją. Jose esančių duomenų analizė dažniausiai remiasi įprastų užklausų, parašytų SQL kalba, vykdymu.

Tačiau kai duomenų susikaupia daug, ir organizacija mėgina atlikti strateginę analizę naudodama tą pačią OLTP duomenų bazę, kaip ir einamojo duomenų apdorojimo atveju, dažnai susiduriama su keliomis problemomis:

- Didelių skaičiavimo pajėgumų reikalaujančios intensyviai vykdomos analitinės užklausos neigiamai veikia darbinės sistemos produktyvumą.
- Jeigu darbinėje duomenų bazėje saugoma istorinė informacija, nereikalinga kasdienėms operacijoms atlikti, tokios sistemos produktyvumas mažėja, nes duomenų laikui bėgant vis daugėja.

Aišku, istoriniai duomenys (pavyzdžiui, duomenys apie praėjusių metų pardavimus) atspindi organizacijos verslo istoriją ir yra labai svarbūs šio verslo analizei. Jais remiamasi priimant verslo sprendimus - ir strategiškai planuojant, ir vertinant konkrečių padalinių veiklą visos organizacijos laimėjimų kontekste.

Siekiant išvengti minėtų problemų ir išsaugoti istorinius duomenis kuriamos atskiros duomenų saugyklos (Data Warehouses), kuriose telpa jau istorine tapusi informacija iš darbinės duomenų bazės. Istorinių duomenų saugojimą ir analizę perkėlus į duomenų saugyklas, išvengiama padidėjusių darbinių duomenų bazių serverių apkrovų.

Tačiau sukūrus duomenų saugyklą išsprendžiama tik dalis problemų, nes:

- Duomenų analitikai dažnai neturi reikiamo techninio išsilavinimo, kad galėtų savarankiškai parašyti ir vykdyti įvairias užklausas OLTP duomenų bazėje.
- Atsakant į sudėtingas ir didelių duomenų kiekio apdorojimo reikalaujančias užklausas, tokio tipo užduotims nepritaikytos OLTP sistemos pajėgumas gali būti per lėtas apibendrintų duomenų peržiūrai ir analizei.
- Vieną kartą jau atliktais skaičiavimais negali pasinaudoti kiti vartotojai.

### **OLAP duomenų saugyklos**

Daugelį aukščiau minėtų problemų gali išspręsti OLAP kubai. Tokiu atveju darbinėse reliacinėse duomenų bazėse ar duomenų saugyklose saugomi pradiniai duomenys transformuojami, ir sukuriama optimizuotos duomenų saugojimo struktūros - OLAP duomenų kubai, kurie specialiai pritaikyti greitai duomenų analizei.

OLAP kubuose didelė tarpinių skaičiavimų dalis atliekama dar iki duomenų kubo naudojimo, o vieną kartą atliktų tarpinių skaičiavimų rezultatais gali pasinaudoti visi prieigos teisės turintys šios duomenų bazės vartotojai.

Kadangi OLAP duomenų bazės saugo tarpines agregatines reikšmes, pagal duomenų kubo įgyvendinimo būdą duomenų pasikeitimai ar naujų duomenų įkėlimas į OLAP duomenų bazę gali pareikalauti atnaujinti ar pertvarkyti kubo informaciją (turi būti perskaičiuotos kai kurios agregatinės reikšmės).

Naudojant OLAP kubus yra užtikrinama:

- Galimybės analizuoti visą įmonę ar įmonių grupę vienoje aplinkoje;
- Duomenų ir informacijos analizės funkcionalumas;
- Duomenų gavimo greitis.

### **Pjūviai ir skaičiuojamosios reikšmės**

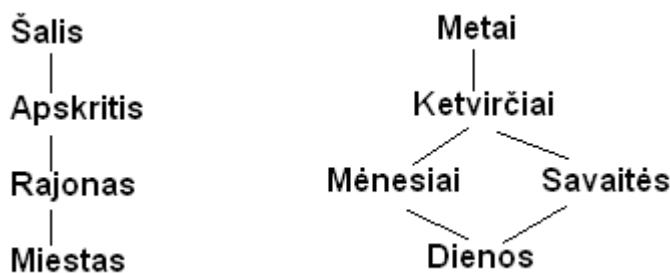
Aprašant OLAP duomenų struktūras (kubus) naudojama pjūvių sąvoka (dimensions). OLAP duomenų bazėje esantys įvykiai (eilutė lentelėje) gali būti analizuojami pagal įvairius parametrus - įvairiais pjūviais (dimensijomis). Pamėginsime paaiškinti pjūvių sąvoką naudodami pardavimų modelio pavyzdį. Užregistruotas prekių pardavimo įvykis apibūdinamas keliais parametrais (kada tai įvyko, koks produktas parduotas, kokioje parduotuvėje), todėl ir įmonės mastu pardavimus galima analizuoti įvairiais pjūviais:

**Pagal laiką.** Galima sužinoti, kaip prekyba sekėsi šį mėnesį, palyginti su praėjusiu, su praėjusiais metais ir pan.

**Pagal produktą.** Parduodamos įvairios produktų grupės, jų tipai ir pagaliau - konkretūs produktai. Galima sužinoti, kokią pajamų dalį sugeneravo ne tik konkretus produktas, bet ir viena ar kita produktų grupė, koks perkamiausias produktas ir pan.

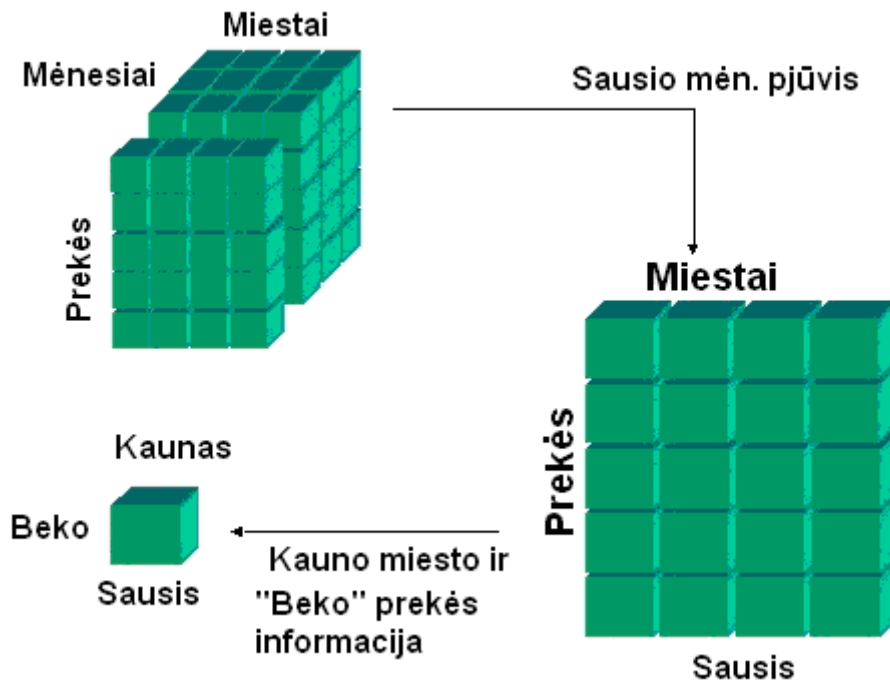
**Pagal vietovę.** Didelės organizacijos pardavimus vykdo didelėje teritorijoje, todėl konkrečios parduotuvės gali būti priskirtos prie apskričių, rajonų, rajonuose gali būti numatyti mažesni struktūriniai vienetai, pavyzdžiui, miestai ir pan. Tokiu būdu galima atsakyti, kaip sekėsi parduoti tą ar kitą prekę apskrityje, rajone ar mieste.

Kaip matome iš pavyzdžio, mūsų pjūviai gali turėti tam tikrą hierarchinę struktūrą, pavyzdžiui, vietovės ir laiko pjūviai turi net po tris hierarchinius lygius (apskritis > rajonas > miestas ir metai > ketvirtis > mėnuo). Tokia hierarchinė struktūra gali būti ir labai sudėtinga.



1 pav. Hierarchinė pjūvių (dimensijų) struktūra

Šiuo atveju sukūrėme trimatį duomenų kubą, kurio matmenis apibrėžia laiko, produkto ir vietovės dimensijos.



2 pav. Trimačio duomenų kubo vaizdas

Tačiau loginė duomenų kubo struktūra gali būti ne tik trimatė, bet ir daugiamatė (keturmatė, penkiamatė ir t. t.).

Kita pagrindinė OLAP duomenų bazių sąvoka yra skaičiuojamoji vertė (measure). Galima pasirinkti, kokią su konkrečiu įvykiu susijusią skaitinę reikšmę analizuosime. Pateikiamame pavyzdyje galėtume analizuoti ne tik, pavyzdžiui, pajamas iš šio miesto per mėnesį parduoto konkretaus produkto, bet ir jo pardavimo pelną ar parduotų prekių vienetų skaičių.

Sukūrus duomenų kubą, OLAP duomenų bazės leis lengvai atlikti įvairius veiksmus:

- Perėjimą nuo bendresnių reikšmių iki individualių ir atvirkščiai, t. y. judėti hierarchine pjūvių (dimensijų) struktūra. Žinodami kurio nors produkto pardavimų pokytį per mėnesį visoje apskrityje, esant reikalui galime pažiūrėti išsamiau - kaip produkto pardavimai kito skirtinguose šios apskrities rajonuose ar net atskirose parduotuvėse.
- Duomenų apibendrinimą pagal įvairius parametrus, t. y. analizę įvairiais pjūviais. Pavyzdžiui, ataskaitas apie pelną, pardavus, tam tikrą produktą galime analizuoti ne laike, bet pagal vietovę.
- Analizuojamojo matmens pakeitimą, t. y. analizę pagal skirtingas skaičiuojamąsias reikšmes. Vietoj pardavimų analizės pagal pajamas tuos pačius pardavimus toje pačioje vietovėje ir tuo pačiu metu galime analizuoti pagal jų sugeneruotą pelną.
- Turėdami kube saugomus duomenis, galėtume lengvai atsakyti į klausimą, kiek pajamų (skaičiuojamoji reikšmė) praėjusį pusmetį (laiko pjūvis) buvo gauta Kauno apskrities parduotuvėse (vietovės pjūvis) „Beko“ produktų grupei (produktų pjūvis) iš visų pirkėjų.

### Sudėtingesnė duomenų analizė (data mining)

Duomenų gavyba – tai prasmingų dėsnų, modelių ir tendencijų radimo procesas dideliuose informacijos kiekiuose, naudojant modelių atpažinimo, statistinius bei matematinius metodus.

Jei įprasti duomenų analizės metodai parodo norimų kintamųjų priklausomumą, tai duomenų gavyba unikali tuo, kad analizės rezultatas yra naujų priklausomybių, apie kurių egzistavimą buvo ar net nebuvo įtariama, radimas.

Duomenų gavyba turi vieną savybę, išskiriančią ją iš kitų procesų ar operacijų, susijusių su duomenų bazėmis. Jei įprastos ataskaitos bei įvairios analitinės priemonės atsako į



klausimus "kas yra?" ar "kas bus, jeigu?", tai duomenų gavybos rezultatas yra atsakymas į nepateiktus klausimus, t.y. naujų priklausomybių radimas. Tuomet sudaromos galimybės pagerinti veiklą arba nustatyti nedidelius nesklandumus.

Duomenų gavyba yra labai plati sritis, todėl yra ir daug jai skirtų metodų, algoritmų bei taikomųjų sistemų:

**Asociacijų paieška** – tai dėsningumų analizė tam tikrose reiškinių ar daiktų grupėse. Pavyzdys — pirkimo analizė. Nagrinėjama, kokios prekės perkamos kartu, kokia tikimybė, kad bus būtent toks derinys ir t.t. Tokio tipo uždaviniai gali būti sprendžiami rengiant reklamos kampanijas, kuriant nuolaidų sistemas.

**Eiliškumo analizė** – tai dėsningumų paieška atsižvelgiant į laiką. Šiuo atveju svarbu ne tik tai, kokiomis paslaugomis naudojasi klientas, bet ir kokia eilės tvarka. Šis metodas padeda efektyviau teikti paslaugas.

**Grupavimas (klasterizavimas)** dažnai būna vienas pirmųjų duomenų gavybos žingsnių. Tai visos duomenų aibės suskaidymas į poaibius pagal skiriamuosius bruožus. Tai ir rinkos ar klientų segmentavimas, ir nekilnojamojo turto grupavimas pagal būdingus duomenis, ir daugelis kitų uždavinių.

**Klasifikavimas** dažnai atliekamas po grupavimo. Kai nagrinėjama aibė jau padalyta į pogrupius, dažnai kyla klausimas, kam priskirti naujus elementus. Grupavimu anksčiau neįvardyti poaibiai išskiriami iš duomenų visumos, o klasifikuojant sprendžiama, kaip sudėti elementus į žinomas grupes.

Po klasifikavimo atliekamas **įvertinimas**. Pavyzdžiui, finansinė institucija ne tik nori žinoti savo išduotų paskolų apibūdinimą "gera — bloga", bet ir jų įvertinimą.

**Prognozavimas** taip pat labai svarbus duomenų gavybai. Atsižvelgiant į turimus duomenis bei pastebėtas tendencijas, bandomi prognozuoti ateities įvykiai.

Dažnai uždaviniui išspręsti taikomi keli metodai iš eilės ar net sudėtingi jų deriniai. Nė vienas jų nėra universalus ar neprikaištingas. Vienu trūkumas — sudėtingumas, kitų — didelė modelių apimtis, daug sugaištama laiko.

### **Pasaulinė OLAP rinka**

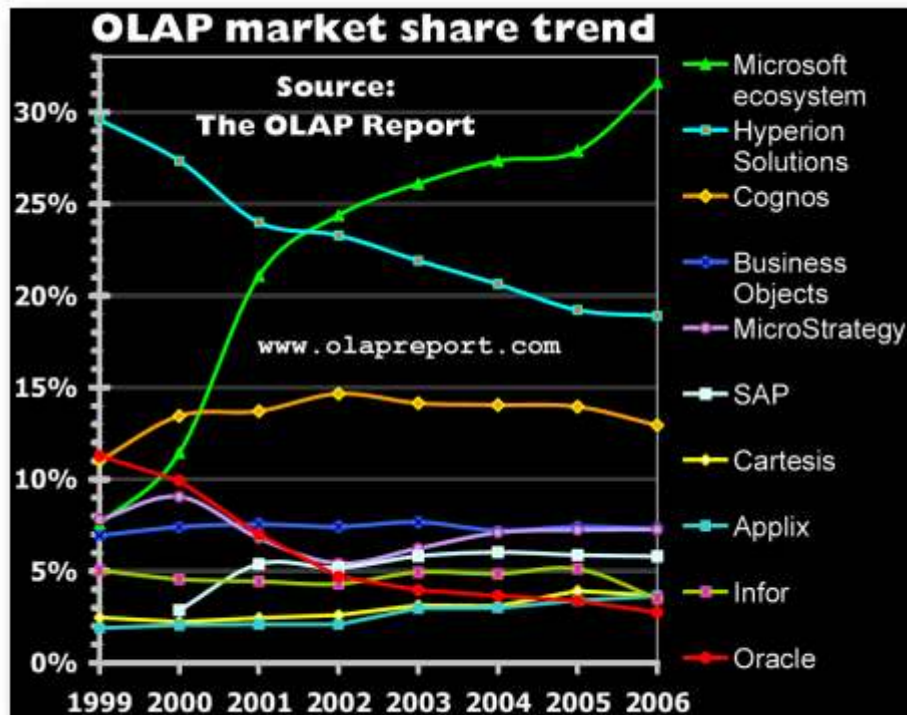
Seniausias OLAP produktas "Express" pasirodė 1970 metais - beveik tokiu pačiu metu, kaip ir reliacinių duomenų bazių valdymo sistemos. Tačiau pasaulinis OLAP rinkos pagyvėjimas ir plėtra ypač juntama nuo 2002 metų, kuri vis nepaliaujamai auga.

Atliktas "The OLAP Report" ([www.olapreport.com](http://www.olapreport.com)) nepriklausomas tyrimas parodė, kad OLAP rinka pasaulyje auga maždaug 20-40 proc. per metus - nuo 0,5 bilijono dolerių 1994 metais ji užaugo iki 5.7 bilijonų dolerių 2006-aisiais. Prognozuojama, kad 2008m. pasieks beveik 7 bilijonų dolerių ribą.



3 pav. OLAP rinka (Pagal "The OLAP Report" [www.olapreport.com](http://www.olapreport.com))

Tas pats šaltinis pateikia ir šešių didžiausių OLAP programinės įrangos gamintojų rinkos dalį, atsižvelgiant ne tik į parduotą programinę įrangą, bet ir teikiamas OLAP sistemų projektavimo, diegimo ir priežiūros paslaugas:



4 pav. OLAP programinės įrangos rinkos pasiskirstymas tarp gamintojų

Didžiausias pokytis šioje rinkoje įvyko 1998 metais su "MS SQL Server 7" į OLAP rinką įžengus bendrovei "Microsoft". 2001 metais ji pasivijo didžiausią rinkos žaidėją "Hyperion Solutions" ([www.hyperion.com](http://www.hyperion.com)), siūlančią "Essbase" paketą. OLAP sprendimų tiekėjų rinkoje lyderio pozicijas užimančio "Microsoft" rezultatai yra puikūs, tai parodė 2006-ųjų rinkos analizė.

Dažnas gali pasakyti, kad tokia programinė įranga paprastam vartotojui yra neprieinama dėl naudojimosi sudėtingumo ar didelės kainos. Iš tikrųjų ilgai taip ir buvo. Istoriskai duomenų analizė (data mining) buvo prieinama tik aukštos kvalifikacijos vartotojams, kurie naudojo specialius statistinius paketus ar brangią ne masinei rinkai skirtą OLAP programinę įrangą. Aišku, darbui su šiais paketais reikėjo ir specialaus pasiruošimo. "Microsoft" išleidus duomenų bazių valdymo paketą "MS SQL Server 7.0", įvyko tam tikras lūžis, nes su įprasta reliacine duomenų bazių valdymo sistema (RDBVS) vartotojai nemokamai gavo ir OLAP funkcijas atliekantį priedą.

Išleidus "MS SQL Server 2000" šis OLAP įrankis buvo pavadintas "Analysis Services", tačiau ir toliau liko nemokamas. Būna, kad net kai kurie IT profesionalai, kasdien dirbdami su "MS SQL Server" programine įranga, nežino, kad tame pačiame instaliaciniame CD yra "Analysis Services" - ją tik reikia papildomai įdiegti.

## Išvados

Dėl vis didėjančių reikalavimų duomenų analizei atsiranda poreikis kurti duomenų saugyklas. Duomenų saugykloje gali būti kaupiami skirtingų organizacijos filialų, esančių geografiškai nutolusiose vietovėse, duomenys ir vykdoma bendra duomenų analizė.

OLAP duomenų struktūra galima panaudoti bet kurioje įmonėje, nepriklausomai nuo joje vykstančių verslo procesų tipo ir sudėtingumo. Duomenų saugykla realizuota OLAP struktūra pasižymi greitu informacijos išgavimu iš didelių duomenų kiekių, bei patogia informacijos analize.

Naudojant duomenų saugyklą yra sutaupomas darbuotojų laikas, padidėja darbo efektyvumas, atliekant detalesnę ir gilesnę duomenų analizę yra didinamas įmonės pelningumas.

## Literatūros sąrašas

- [1] **J. Mundy, W. Thornthwaite, R. Kimball**, The Microsoft Data Warehouse Toolkit, *John Wiley&Sons*, 2006.
- [2] **E. Sysojevas**, Interaktyvios duomenų analizės įrankiai: OLAP duomenų bazės, *Naujoji komunikacija*, 2003m. Nr6(127)
- [3] **S. Maskeliūnas**, Žinių technologijų pagrindinių terminų žodynelis, Vilnius, 2006 [interaktyvus], Prieiga per internetą: [http://www.mii.lt/files/mii\\_prep\\_2006\\_35.doc](http://www.mii.lt/files/mii_prep_2006_35.doc)
- [4] Naujausios publikacijos, Duomenų bazės. [interaktyvus] 2006, Prieiga per internetą: <http://itekspertas.projektas.lt/>
- [5] The OLAP Report [interaktyvus] Prieiga per internetą: <http://www.olapreport.com/>
- [6] Data Warehouse Design Considerations [interaktyvus], Prieiga per internetą <http://www.microsoft.com/technet/prodtechnol/sql/2000/reskit/part5/c1761.msp>
- [7] **W. H. Inmon**, OLAP and Data Warehouse, 2000,[3-7]

## OLAP kubas application in corporative enterprise

Business use a big amount of data in our days, that data become just ballast if we incapacity to sift it. Just data mining using special software, transform data in to information. That information become reason of correct business rule.

OLAP software make data warehouse, that help analyse data quick and convenient, even data are from diferent environment.

What is OLAP? OLAP is On-Line Analytical Procesing. This term use to describe software products, thats have possibility to analyse information in real time. OLAP systems are one from a lot of Business Intelligence group products. In the worl are a lot products which are the part of this group: from MS SQL OLAP cubes to „Business Objects“, „Cognos“, Corporate Planner“ and „Microstrategy“ systems.



## 2 PRIEDAS

### ***Panaudojimo atvejų sąrašas***

1. **PANAUDOJIMO ATVEJIS:** Formuoti kubą

**Aktorius:** Administratorius

**Tikslas:** suformuoti kubo duomenis

**Ryšiai su kitais PA:** apibendrina PA2 ir PA3.

**Prieš-sąlygos:** administratorius turi būti įsijungęs kubo formavimo paketą.

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką.

**Po-sąlyga:** suformuotas kubas.

**Pagrindinis scenarijus:**

1. Vartotojas pasirenka norimą atlikti veiksmą
2. Atsidaro reikiamas langas
3. Vartotojas pasirenka dimensiją
4. Sistema atlieka dimensijos suformavimą
5. Vartotojas pasirenka faktą
6. Sistema atlieka fakto suformavimą
7. Vartotojas pasirenka matą (nurodo apskaičiavimo formulę)
8. Sistema atlieka mato suformavimą
9. Vartotojas patvirtina, kad teisingai atliko veiksmus
10. Sistema išsaugo įvestą informaciją ir suformuoja kubą

**Alternatyvūs scenarijai:**

- Vartotojas blogai suformavo duomenis.
- Vartotojas nusprendė nutraukti darbą dar nepabaigus formuoti kubo.

2. **PANAUDOJIMO ATVEJIS:** Formuoti dimensijas

**Aktorius:** Administratorius

**Tikslas:** formuoti kubo dimensijas.

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)

PA1

**Prieš-sąlygos:** administratorius turi būti įsijungęs kubo formavimo paketą

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką.

**Po-sąlyga:** suformuotos dimensijos

**Pagrindinis scenarijus:**

1. Administratorius sukuria dimensiją.

2. Sistema sugeneruoja dimensiją.
3. Sistema išsaugo pakeitimus.

**Alternatyvūs scenarijai:**

- Administratorius neteisingai suformavo dimensiją.
- Įvyko programos klaida.

3. **PANAUDOJIMO ATVEJIS:** Formuoti faktus(matus)

**Aktorius:** Administratorius

**Tikslas:** formuoti kubo faktus.

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)

PA1

**Prieš-sąlygos:** administratorius turi būti įsijungęs kubo formavimo paketą

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką.

**Po-sąlyga:** suformuoti faktai

**Pagrindinis scenarijus:**

1. Administratorius sukuria faktą.
2. Sistema sugeneruoja faktą.
3. Sistema išsaugo pakeitimus.

**Alternatyvūs scenarijai:**

- Administratorius neteisingai suformavo faktą.
- Įvyko programos klaida.

4. **PANAUDOJIMO ATVEJIS:** Nustatyti OLAP kubo parametrus

**Aktorius:** Administratorius

**Tikslas:** nustatyti OLAP kubo parametrus

**Ryšiai su kitais PA:** apibendrina PA5 ir PA6

**Prieš-sąlygos:** turi būti suteiktos teisės administruoti kubą

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** nustatyti OLAP kubo parametrai

**Pagrindinis scenarijus:**

1. Nustatyti duomenų atnaujinimo laiką .
2. Sistema išsaugo nustatymus.
3. Nustatyti OLAP kubo vartotojų sąrašą
4. Sistema išsaugo nustatymus.

**Alternatyvūs scenarijai:**

- Administratorius neteisingai nustatė teises/duomenų atsinaujinimo laiką
- Įvyko programos klaida.

#### 5. PANAUDOJIMO ATVEJIS: Nustatyti teises

**Aktorius:** Administratorius

**Tikslas:** nustatyti/koreguoti vartotojų teises

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)  
PA4

**Prieš-sąlygos:** turi būti suteiktos teisės administruoti kubą

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** nustatytos/koreguotos teisės.

**Pagrindinis scenarijus:**

1. Administratorius prisijungia prie kubo.
2. Sistema atidaro teisių nustatymo įrankį
3. Administratorius nustato teises
4. Sistema išsaugo nustatymus.

**Alternatyvūs scenarijai:**

- Administratorius neteisingai nustatė teises
- Įvyko programos klaida.

#### 6. PANAUDOJIMO ATVEJIS: Nustatyti duomenų atsinaujinimo laiką

**Aktorius:** Administratorius

**Tikslas:** nustatyti/koreguoti duomenų atsinaujinimo laiką.

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)  
PA4

**Prieš-sąlygos:** turi būti suteiktos teisės administruoti kubą

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** nustatytas/koreguotas duomenų atsinaujinimo laikas.

**Pagrindinis scenarijus:**

1. Administratorius parenka kubo atsinaujinimo laiką.
2. Sistema išsaugo nustatymus

**Alternatyvūs scenarijai:**

- Administratorius neteisingai nustatė duomenų atsinaujinimo laiką
- Įvyko programos klaida.

## 7. PANAUDOJIMO ATVEJIS: Formuoti kubo rez

**Aktorius:** Įmonės darbuotojas

**Tikslas:** suformuoti kubo rezultatus reikiamu pjūviu

**Ryšiai su kitais PA:** apibendrina PA8, PA9 ir PA10

**Prieš-sąlygos:** vartotojui turi būti suteiktos teisės naudotis kubu

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** suformuoti kubo rezultatai

### **Pagrindinis scenarijus:**

1. Vartotojas atidaro MS Excel failą.
2. Per meniu punktą atidaro kubo importavimo langą.
3. Vartotojas įveda arba pasirenka reikalingą informaciją
4. Sistema išsaugo įvestą informaciją ir išveda importuotą kubą.
5. Vartotojas išdėlioja dimensijas.
6. Sistema užsaugo pakeitimus ir pateikia faktus pagal išdėliotas dimensijas
7. Vartotojas nurodo faktus kuriuos nori matyti
8. Sistema pateikia duomenis pagal nurodytus faktus
9. Vartotojas nurodo duomenų filtravimą
10. Sistema pateikia nufiltruotus duomenis

### **Alternatyvūs scenarijai:**

- Vartotojui nepavyko prisijungti prie kubo
- Nutruko ryšys su serveriu

## 8. PANAUDOJIMO ATVEJIS: Nustatyti kubo faktus ir dimensijas

**Aktorius:** Įmonės darbuotojas

**Tikslas:** nustatyti faktus ir dimensijas pagal norimą matyti pjūvį

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)  
PA7

**Prieš-sąlygos:** vartotojui turi būti suteiktos teisės naudotis kubu

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** nustatyti faktai ir dimensijos pagal norimą matyti pjūvį

### **Pagrindinis scenarijus:**

1. Vartotojas išdėlioja dimensijas.
2. Sistema užsaugo pakeitimus ir pateikia faktus pagal išdėliotas dimensijas
3. Vartotojas nurodo faktus kuriuos nori matyti



4. Sistema pateikia duomenis pagal nurodytus faktus

**Alternatyvūs scenarijai:**

- Vartotojui nepavyko prisijungti prie kubo
- Nutruko ryšys su serveriu

9. **PANAUDOJIMO ATVEJIS:** Filtruoti duomenis

**Aktorius:** Įmonės darbuotojas

**Tikslas:** filtruoti duomenis pagal norimus kriterijus

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<include>>)

PA7

**Prieš-sąlygos:** vartotojui turi būti suteiktos teisės naudotis kubu

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** nufiltruoti duomenys

**Pagrindinis scenarijus:**

1. Vartotojas nurodo filtravimo kriterijus.
2. Sistema pateikia nufiltruotus duomenis, pagal pasirinktus filtravimo kriterijus

**Alternatyvūs scenarijai:**

- Vartotojui nepavyko prisijungti prie kubo
- Nutruko ryšys su serveriu

10. **PANAUDOJIMO ATVEJIS:** Importuoti kubą į MS Excel

**Aktorius:** Įmonės darbuotojas

**Tikslas:** Importuoti Kuba į MS Excel.

**Ryšiai su kitais PA:** Įeina į pagrindinį sistemos veiklos panaudojimo atvejį (<<extend>>)

PA7

**Prieš-sąlygos:** vartotojui turi būti suteiktos teisės naudotis kubu

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** importuotas kubas į MS Excel

**Pagrindinis scenarijus:**

1. Vartotojas atidaro MS Excel failą.
2. Per meniu punktą atidaro kubo importavimo langą.
3. Vartotojas įveda arba pasirenka reikalingą informaciją
4. Sistema išsaugo įvestą informaciją ir išveda importuotą kubą.

**Alternatyvūs scenarijai:**

- Vartotojui nepavyko prisijungti prie kubo

- Nutruko ryšys su serveriu

#### 11. PANAUDOJIMO ATVEJIS: Skaityti rez

**Aktorius:** Įmonės darbuotojas

**Tikslas:** analizuoti duomenis

**Ryšiai su kitais PA:** nėra

**Prieš-sąlygos:** turi būti suformuoti kubo rezultatai.

**Sužadinimo sąlyga:** vartotojas įeina i MS Excel failą

**Po-sąlyga:** perskaityti/išanalizuoti duomenys.

**Pagrindinis scenarijus:**

1. Vartotojas atsidaro MS Excel failą su suformuotais duomenimis
2. Vartotojas skaito/analizuoja duomenis

**Alternatyvūs scenarijai:**

- nėra.

#### 12. PANAUDOJIMO ATVEJIS: Transformuoti duomenis

**Aktorius:** Administratorius

**Tikslas:** transformuoti duomenis iš XML struktūros failų ir vvs duomenų bazės į bendrą duomenų schemą

**Ryšiai su kitais PA:** nėra

**Prieš-sąlygos:** nėra

**Sužadinimo sąlyga:** vartotojas paspaudžia atitinkamą programos mygtuką

**Po-sąlyga:** duomenys transformuoti į vienodą duomenų formatą

**Pagrindinis scenarijus:**

1. Sukurti bendrą duomenų formatą

**Alternatyvūs scenarijai:**

- nėra.

### ***DB lentelių aprašai***

- PriceDiskTable struktūra

12 lentelė. PriceDiskTable lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
Amount	Numeric	13	0
Markup	Numeric	13	0
ItemRelation	Varchar	20	„ „

- ForecastSales struktūra

13 lentelė. ForecastSales lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
StartDate	DateTime	8	1900 01 01 00 00 00
ItemId	Varchar	25	„ „
ModelId	Varchar	10	„ „
SalesQty	Numeric	13	0
Amount	Numeric	13	0
ItemId	Varchar	25	„ „
InventDimId	Varchar	20	„ „
CustAccountId	Varchar	10	„ „

- CustInvoiceTrans struktūra

14 lentelė. CustInvoiceTrans lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
ItemId	Varchar	20	„ „
qty	Numeric	13	0
InvoiceDate	DateTime	8	1900 01 01 00 00 00
LineAmountMST	Numeric	13	0
name	Varchar	15	„ „
RecId	Int	4	-
InvoiceId	Varchar	20	„ „
InventDimId	Varchar	20	„ „

- CustInvoiceJour struktūra

15 lentelė. CustInvoiceJour lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
InvoiceAccount	Varchar	10	„ „
ExchRate	Numeric	13	0
InvoiceId	Varchar	20	„ „

- CustTable struktūra

16 lentelė. CustTable lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
AccountNum	Varchar	10	„ „
Name	Varchar	30	„ „
address	Varchar	50	„ „
Market	Varchar	10	„ „
Cellularphone	Varchar	15	„ „
Country	Varchar	10	„ „

- MarkupTrans struktūra

17 lentelė. MarkupTrans lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
CalculatedAmount	Numeric	13	0
TransRecId	Int	4	0

- InventDim struktūra

18 lentelė. InventDim lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
ConfigId	Varchar	10	„ „
InventserialId	Varchar	10	„ „
InventDimId	Varchar	20	„ „

- Country struktūra

19 lentelė. Country lentelės duomenų laukų aprašas

Lauko vardas	Lauko tipas	Ilgis	Reikšmė pagal nutylėjimą
Name	Varchar	10	„ „
CountryId	Varchar	30	„ „

## XML failų šablonai

Fact XML failo pavyzdys:

```

<?xml version="1.0" ?>
<DataContainer>
<Fact>
<Items>
<SalesReportFact>
<SaskaitosFakturosData>Data</SaskaitosFakturosData>
<PrekesKodas>Prekes kodas</PrekesKodas>
<Konfiguracija>Konfiguracija</Konfiguracija>
<Modeliai>Modelis</Modeliai>
<PrekybinisVardas>prekybinis vardas</PrekybinisVardas>
<Sandelys>sandelys</Sandelys>
<SumaLt>Suma LT</SumaLt>
<SumaEur>Suma EUR</SumaEur>
<SumaParduotaValiuta>Suma parduota valiuta</SumaParduotaValiuta>
<Valiuta>Valiutos kodas</Valiuta>
<Kiekis>Kiekis</Kiekis>
<Klientas>Kliento kodas</Klientas>
<VeiklosSanaudos>Veiklos sanaudos</VeiklosSanaudos>
<Palukanos>Palukanos</Palukanos>
<Sildymas>Sildymas</Sildymas>
<SnaigesUzsakymoNr>snaiges uzsakymo nr </SnaigesUzsakymoNr>
<Savikaina>savikaina</Savikaina>
<TiesiogKastai>Tiesioginiai kastai</TiesiogKastai>
<Komisiniai>Komisiniai</Komisiniai>
<Logistika>Logistika</Logistika>
<Kitos>Kitos</Kitos>
<Garantinis>Garantinis</Garantinis>
<Transportas>Transportas</Transportas>
<Apvalinimas>Apvalinimas</Apvalinimas>
<Draudimas>Draudimas</Draudimas>
<Muitas>Muitas</Muitas>
<TransportoVezejas>Transportas vezejas</TransportoVezejas>
<TiekejoTransportas>Tiekėjo transportas</TiekejoTransportas>
</SalesReportFact>
</Items>
</Fact>

```

```
</DataContainer>
```

Time XML failo pavyzdys:

```
<?xml version="1.0" ?>  
= <DataContainer>  
= <Time>  
= <Items>  
= <SalesReportTime>  
  <SaskaitosFakturosData>Data</SaskaitosFakturosData>  
  <Metai>Metai</Metai>  
  <Ketvirtis>Ketvirtis</Ketvirtis>  
  <Menuo>Menuo</Menuo>  
  <Diena>Diena</Diena>  
  </SalesReportTime>  
  </Items>  
  </Time>  
  </DataContainer>
```

Client XML failo pavyzdys:

```
<?xml version="1.0" ?>  
= <DataContainer>  
= <Client>  
= <Items>  
= <SalesReportClient>  
  <Klientas>Kliento kodas</Klientas>  
  <Rinka>Rinka</Rinka>  
  <Šalis>Salis</Šalis>  
  <Zona>Zona</Zona>  
  </SalesReportClient>  
  </Items>  
  </Client>  
  </DataContainer>
```

## Testavimo procedūros

Šiame priede yra pateiktos detalios programinės įrangos testavimo procedūros.

### Vienetų testavimas

Komponentus/modulius testuos tie patys asmenys, kurie juos sukūrė.

„Baltos“ dėžės metodas:

View „SA\_TimeDIM“

20 lentelė. SA\_TimeDIM komponento testiniai duomenų rinkiniai

Programinio kodo fragmentas	Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
CASE DATEPART(year, [bmssa].CustInvoiceTrans.Invoic eDate)	TA_1	true	Išvedama pagaidaujama reikšmė
	TA_2	false	Išvedama „null“ reikšmė

CASE DATEPART(Quarter,	TA_1	WHEN 1	Išvedama reikšmė '1 Ketvirtis'
[bmssa].CustInvoiceTrans.InvoiceDate)	TA_2	WHEN 2	Išvedama reikšmė '2 Ketvirtis'
	TA_3	WHEN 3	Išvedama reikšmė '3 Ketvirtis'
	TA_4	WHEN 4	Išvedama reikšmė '4 Ketvirtis'
	TA_5	false	Išvedamas pranešimas paie klaidą
CASE DATEPART(month,	TA_1	WHEN 1	Išvedama reikšmė „Sausis“
[bmssa].CustInvoiceTrans.InvoiceDate)	TA_2	WHEN 2	Išvedama reikšmė „Vasaris“....
	.....	.....	.....
	TA_13	false	Išvedamas pranešimas apie klaidą
CASE DATEPART(day,	TA_1	WHEN 1	Išvedama reikšmė „1“
[bmssa].CustInvoiceTrans.InvoiceDate)	TA_2	WHEN 2	Išvedama reikšmė „2“....
	.....	.....	.....
	TA_32	false	Išvedamas pranešimas apie klaidą
WHERE	TA_1	true	Įvykdoma sąlyga
[bmssa].CustInvoiceTrans.dataAreaId='dat' and (DATEPART(year,	TA_2	False	Klaida
[bmssa].CustInvoiceTrans.InvoiceDate)			
> 2003)			

## View „SA\_ClientDIM“

21 lentelė. SA\_ClientDIM komponento testiniai duomenų rinkiniai

Programinio kodo fragmentas	Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
CASE [bmssa].CustTable.Market	TA_1	WHEN 'Baltijos'	Išvedama reikšmė „Vakarai“
	TA_2	WHEN 'C.Euro pa'	Išvedama reikšmė „Vakarai“
	TA_3	WHEN 'C.Euro pa 2'	Išvedama reikšmė „Vakarai“
	TA_4	WHEN 'Vakara i2'	Išvedama reikšmė „Vakarai“
	TA_5	WHEN 'Vakata i3'	Išvedama reikšmė „Vakarai“
	TA_7	WHEN 'Rytai'	Išvedama reikšmė „Rytai“
	TA_8	false	Klaida
WHERE	TA_1	true	Įvykdoma sąlyga
[bmssa].Custtable.dataAreaId='dat' and	TA_2	False	Klaida
substring([bmssa].custtable.accountnum			
,1,1) != '_'			

## View „SA\_Fact“

22 lentelė. SA\_Fact komponento testiniai duomenų rinkiniai

Programinio kodo fragmentas	Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
-----------------------------	---------------------	---------	----------------------

select (SUM(PriceDiscTable.Amount)/count(*)* [bmssa].CustInvoiceTrans.qty from PriceDiscTable where PriceDiscTable.ItemRelation = [bmssa].CustInvoiceTrans.ItemId AND PriceDiscTable .DataAreaId = PriceDiscTable.DataAreaId AND PriceDiscTable .AccountRelation='Savikainos' ) AS [Savikaina]	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	false	Klaida
(select (PriceDiscTable.Markup* [bmssa].CustInvoiceTrans.qty) from PriceDiscTable where PriceDiscTable.ItemRelation = [bmssa].CustInvoiceTrans.ItemId AND PriceDiscTable .DataAreaId = PriceDiscTable.DataAreaId AND PriceDiscTable .AccountRelation='Savikainos' ) AS [TiesiogKastai] ,	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	False	Klaida
(PriceDiscTable.Markup* [bmssa].CustInvoiceTrans.qty)	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	False	Klaida
select (( (sum(MarkupTrans.CalculatedAmount)) * [bmssa].CustInvoiceJour.ExchRate)/100) from MarkupTrans where MarkupTrans.TransRecId= [bmssa].CustInvoiceTrans.RecId ) AS [Išlaidos Viso]	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	False	Klaida
(select (( isnull((sum(MarkupTrans.CalculatedA mount)),0)* [bmssa].CustInvoiceJour.ExchRate)/100) from MarkupTrans where MarkupTrans.TransRecId= [bmssa].CustInvoiceTrans.RecId and MarkupTrans.MarkupCode='Commision' ) AS [Komisiniai],	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	False	Klaida
select (( isnull((sum(MarkupTrans.CalculatedA mount)),0)* [bmssa].CustInvoiceJour.ExchRate)/100) from MarkupTrans where MarkupTrans.TransRecId= [bmssa].CustInvoiceTrans.RecId and MarkupTrans.MarkupCode='Service' ) AS [Garantinis],	TA_1	true	Išvedama pageidaujama reikšmė
	TA_2	False	Klaida
case (select distinct markupcategory from MarkupTrans	TA_1	WHEN 0	Vykdoma „sum(MarkupTrans.valu e))*[bmssa].CustInvoi ceJour.ExchRate)/100) „operacija
	TA_2	WHEN 1	Vykdoma „sum(MarkupTrans.valu e))*[bmssa].CustInvoi ceTrans.qty*[bmssa].C ustInvoiceJour.ExchRa te)/100)„operacija
	TA_3	false	Klaida

```

WHERE
[bmssa].CustInvoiceTrans.dataAreaId='d
at' and
(InventTable.ItemGroupId='300' or
InventTable.ItemGroupId='310' or
InventTable.ItemGroupId='320' or
InventTable.ItemGroupId='330')
and
( [bmssa].CustInvoiceTrans.LineAmountM
ST != 0) and (DATEPART(year,
[bmssa].CustInvoiceTrans.InvoiceDate)
> 2003)
and Inventdim.InventLocationId!=4003

```

TA\_1            true            Įvykdoma sąlyga  
TA\_2            False           Klaida

„Juodos“ dėžės metodas – kiekvienam moduliui yra sudaromi testai ir lyginami išeities rezultatis su testų duomenimis, sutampa ar ne.

23 lentelė. „Juodos dėžės“ metodo testiniai duomenų rinkiniai

Testavimo atvejo ID	Įėjimo duomenys	Rezultatai
TA_1	Test „SA TimeDIM“	Teisingi/neteisigi
TA_2	Tets „SA ClientDIM“	Teisingi/neteisigi
TA_3	Test „SA Fact“	Teisingi/neteisigi

## Integravimo testavimas

- Veikimo testavimas

24 lentelė. Veikimo testavimo testiniai duomenų rinkiniai

Testavimo atvejo ID	Sistemos veikimo (paleidimo) aplinka	Sąlygos	Laukiamas rezultatas
TA_1	50 prisijungusių vartotojų	Duomenų bazės dydis >= 30 GB	Ispėjantis pranešimas sistemos administratoriui
TA_2	50 prisijungusių vartotojų	Vartotojai prisijungę prie tos pačios bazės	Operacijų vykdymo laikas 20-60 s
...	...	...	...

- Funkcinis testavimas

25 lentelė. Funkcinio testavimo testiniai duomenų rinkiniai

Testavimo atvejo ID	Scenarijus	Sąlygos	Laukiamas rezultatas
TA_1	Formuoti dimensijas	Pavyko suformuoti	Suformuota dimensija
TA_2	Formuoti dimensijas	Nepavyko suformuoti	Klaida
TA_3	Formuoti faktus	Pavyko suformuoti	Suformuotas faktas
TA_4	Formuoti faktus	Nepavyko suformuoti	Klaida
TA_5	Nustatyti teises	Pavyko nustatyti	Teisės nustatytos
TA_6	Nustatyti teises	Nepavyko nustatyti	Klaida
TA_7	Nustatyti duomenų atsinaujinimo laiką	Pavyko nustatyti	Nustatytas duomenų atsinaujinimo laikas
TA_8	Nustatyti duomenų atsinaujinimo laiką	Nepavyko nustatyti	Klaida
TA_9	Transformuoti duomenis	Pavyko transformuoti	Teisingi duomenys
TA_10	Transformuoti duomenis	Nepavyko transformuoti	Klaida
TA_11	Nustatyti faktus ir dimensijas	Pavyko nustatyti	Nustatyti faktai ir dimensijos
TA_12	Nustatyti faktus ir dimensijas	Nepavyko nustatyti	Klaida



TA_13	Filtruoti duomenis	Pavyko filtruoti	Teisingai filtruoti duomenys
TA_14	Filtruoti duomenis	Nepavyko filtruoti	Klaida
TA_15	Inportuoti kuba į MS Excel	Pavyko inportuoti	Teisingai importuotas kubas
TA_16	Inportuoti kuba į MS Excel	Nepavyko inportuoti	Klaida
TA_17	Skaityti rezultatus	Pavyko	Teisingi rezultatai
TA_18	Skaityti rezultatus	Nepavyko	Klaida

- Saugumo testavimas

26 lentelė. Saugumo testavimo testiniai duomenų rinkiniai

Testavimo atvejo ID	Sąlygos	Laukiamas rezultatas
TA_1	Prie sistemos prisijungė vartotojas, turintis prisijungimo teises	Leidžia sistema prisijungti
TA_2	Prie sistemos prisijungė vartotojas, neturintis prisijungimo teisių	Neleidžia sistema prisijungti

## Priėmimo testavimas

Priėmimo testavimą atlieka užsakovas kartu su programinės įrangos vadovu.

27 lentelė. Priėmimo testavimo testiniai duomenų rinkiniai

PA ID	Scenarijus	Priimtas/nepriimtas	Pastaba
TA_1	Formuoti dimensijas	Taip/ne	Reikalingi pataisymai/ne
TA_2	Formuoti faktus	Taip/ne	Reikalingi pataisymai/ne
TA_3	Nustatyti teises	Taip/ne	Reikalingi pataisymai/ne
TA_4	Nustatyti duomenų atsinaujinimo laiką	Taip/ne	Reikalingi pataisymai/ne
TA_5	Transformuoti duomenis	Taip/ne	Reikalingi pataisymai/ne
TA_6	Nustatyti faktus ir dimensijas	Taip/ne	Reikalingi pataisymai/ne
TA_7	Filtruoti duomenis	Taip/ne	Reikalingi pataisymai/ne
TA_8	Inportuoti kuba į MS Excel	Taip/ne	Reikalingi pataisymai/ne
TA_9	Skaityti rezultatus	Taip/ne	Reikalingi pataisymai/ne

## Naudotojo sąsajos testavimas

Naudotojo sąsajos testavimas susideda iš:

28 lentelė. Naudotojo sąsajos testavimo testiniai duomenų rinkiniai

Nr.	Objektas	Testavimo sąlygos, atvejais	Laukiami rezultatai
1.	L a u k a i	Patikrinti spalvas, dydžius, pavadinimus, gramatiką, stilių.	Nėra gramatinių klaidų. Korektiški laukų dydžiai. Spalvos yra neryškios, nevargina akių.
2.	M y g t u	Patikrinti pavadinimus, spalvas, dydžius, gramatiką, išdėstymą.	Mygtuko pavadinimas – vienas žodis – veiksmažodžio bendratis. Vienodi mygtukų dydžiai.

		k a i		
3.		D u o m e n ų u ž k r o v i m a s	Patikrinti, ar užkraunami reikalingi duomenys.	Paspaudus mygtuką [Pakelti] užkraunami reikiami duomenys.
4.		V i e n t i s u m a s	Patikrinti, ar visi langai, laukai, mygtukai ir kiti programos objektai realizuoti pagal vienodas taisykles.	Elementai išdėstyti laikantis tos pačios tvarkos, vienodos spalvos.
5.		F u n k c ij a	Patikrinti, ar veiksmas, kurį turi iššaukti mygtuko paspaudimas yra teisinga ir ar atidaromi reikiami programos langai. Taip pat patikrinti ar yra pranešimai įspėjantys apie klaidas ar neteisingai atliktus veiksmus.	Programa turi neleisti įvesti netinkamų duomenų, taip pat jei naudotojas neteisinga tvarka paspaudė mygtukus, programa neturėtų nustoti funkcionavus.
6.		A iš k u m a s	Jei bet vienas iš laukų yra nevisiškai aiškus naudotojui, jis turi rasti pagalbą ar aprašus ką reiškia tas laukas.	Programa turi rodyti pranešimus apie lauko aprašymą, kad naudotojui būtų aišku ką įvesti ir kas vyksta programoje.

## **OLAP duomenų saugyklos vertinimo anketa**

Ar atitinka Jūsų lūkesčius realizuota OLAP sistema?	TAIP	NE	NEŽINAU
Ar sparčiau atliekate savo pareigas įmonėje naudojantis šia sistema?	TAIP	NE	NEŽINAU
Ar peržvelgiate statistinius duomenis(balansines vertes už metus, mėnesius, savaites) savo įdomumui?	TAIP	NE	NEŽINAU
Ar lengva naudotis sistema?	TAIP	NE	NEŽINAU
Ar sunku buvo priprasti prie sistemos?	TAIP	NE	NEŽINAU
Kiek laiko užtruko sistemos įsisavinimas?	Keletą dienų	Savatė	Keletą savaičių
Kokias klaidas/netikslumus pastebėjote sistemoje?	.....		
	.....		
	.....		
	.....		
Ką norėtumėte keisti/ papildyti sistemoje?	.....		
	.....		
	.....		
	.....		
	.....		