KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Marius Vilimas

Duomenų analizės priemonių tyrimas ir taikymas interneto sistemose

Magistro darbas

Darbo vadovas
doc. dr. L. Nemuraitė

Kaunas
2004

1 Įvadas

Didelių informacijos kiekų apdorojimo problema atsirado vos tik pradėjus vystytis informacinėms sistemoms. Sukaupiamų duomenų kiekiai didėja daug greičiau nei vystosi duomenų saugojimo technologijos. Fiziškai neįmanoma sukaupti ir saugoti visus bet kokios organizacijos duomenis. Yra sistemų, kuriose per dieną sukaupiami šimtai GB informacijos. Tačiau tokie duomenys yra vertingi ir reikalingi palyginus trumą laiko tarpą (dieną, savaitę, mėnesį). Vėliau jų saugojimas tokiame pačiame detaliame pavidale ir tokia pačia struktūra, kokia jie pateko į sistemą, netenka prasmės. Be to, praktiškai neįmanoma analitiškai analizuoti tokių didelių duomenų kiekių įvairiais pjūviais, nes šios operacijos būtų per brangios, užimtų per daug laiko ir kai kuriais atvejais galėtų turėti neigiamų padarinių visai sistemai.

Šias duomenų kaupimo ir analitinio apdorojimo problemas sprendžia taip vadinamos duomenų saugyklų (angl. *warehouse*) sistemos. Jose duomenys analizuojami naudojant specialią programinę įrangą - OLAP sistemas. OLAP (angl. *On-Line Analytical Processing*) - tai sistemos, analitiškai apdorojančios didelius duomenų kiekius. Plačiau ši sąvoka aptariama analitinėje magistrinio darbo dalyje.

Palyginus su kitomis duomenų bazių valdymo sistemų srityje sprendžiamomis problemomis, duomenų saugyklų kūrimo problema atsirado gana neseniai. Pats terminas atsirado tik dešimtojo dešimtmečio pradžioje. Užsienio autorių tiriamųjų darbų šioje srityje yra parašyta daug, tačiau praktinio taikymo patirtis Lietuvoje yra maža.

Šiame darbe tiriamos duomenų perkėlimo į duomenų saugyklas bei OLAP priemonės, įeinančios į Lietuvoje plačiausiai paplitusių duomenų bazių valdymo sistemų Oracle ir MS SQL Server sudėtį. Analizuojamos galimybės šias priemones taikyti interneto svetainių lankomumo duomenų apdorojimui. Apkrautose interneto svetainėse tikslinga stebėti, kokios informacijos vartotojas dažniausiai ieško, kokie puslapiai lankomiausi. Svetainės administratorių dažnai domina svetainėje pateikiamos informacijos poreikiai paros meto, savaitės dienų, vartotojų amžiaus pjūviais. Norint analizuoti šiuos duomenis, tikslinga pasitelkti saugyklas ir OLAP.

Svetainių lankomumo duomenų analizė turi didelę reikšmę svetainės reklamai, leidžia koreguoti jos turinį, išdėstymą, informacijos pateikimo stilių priklausomai nuo besilankančios auditorijos. Lietuvos interneto svetainių rinkoje labai trūksta tikslių duomenų apie besilankančiųjų amžiaus grupes, pomėgius ir pan.

Pagrindinis šio darbo rezultatas yra MS SQL serverio ir Oracle saugyklų kūrimo ir duomenų analizės priemonių galimybių išaiškinimas bei naudojimo rekomendacijos. Praktinis rezultatas yra šių priemonių pritaikymas interneto svetainės duomenų analizei.

Šį darbą sudaro trys pagrindinės dalys. Analitinėje dalyje nagrinėjamos analitinio duomenų apdorojimo priemonės, duomenų saugyklos ir jų kūrimo principai.

Tyrimo dalyje analizuojama kokios saugyklų kūrimo ir duomenų analitinio apdorojimo priemonės įeina į MS SQL ir ORACLE duomenų bazių valdymo sistemas.

Eksperimento dalyje aprašyti du analitinio duomenų apdorojimo sistemos prototipai. Vienas sukurtas MS SQL, kitas ORACLE priemonėmis. Nagrinėjami šių priemonių privalumai bei trūkumai. Pateikiamos trumpos rekomendacijos kada verta naudoti vieną, kada kitą priemonių rinkinį.

2 Duomenų saugyklų naudojimo interneto sistemose galimybių analizė

2.1 Tyrimo tikslas

Tyrimo sritis: OLAP priemonės bei jų naudojimas interneto sistemose.
Tyrimo objektas: MSSQL ir Oracle duomenų bazių valdymo sistemos.

Problema: Norint detaliai ištyrinėti OLAP priemones, reikia sukurti sistemą, kurioje jos būtų pritaikytos. Pasirinkau gana naują OLAP ir Duomenų saugyklų pritaikymo sritį: Interneto svetainės vartotojų duomenų apdorojimą.

2.2 Duomenų saugyklos sąvoka

2.2.1 Veiklos duomenų saugyklos apibrėžimas

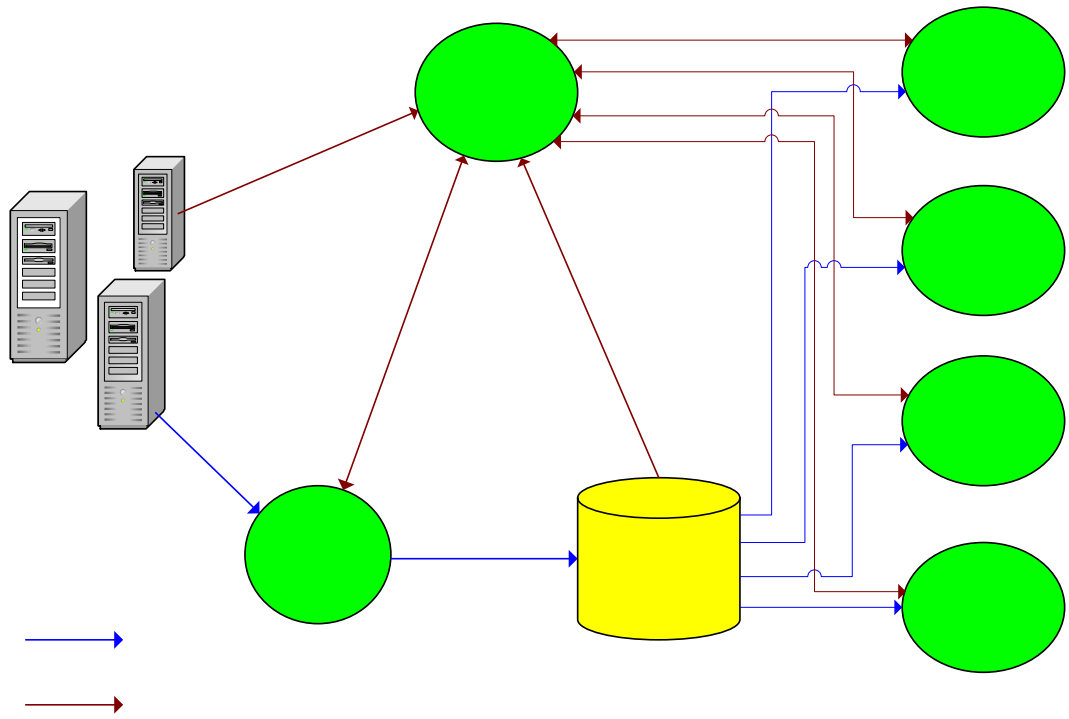
Terminą *Data Warehouse* (Duomenų saugykla) 1990 metais įvedė B. Inmon. Jis duomenų saugyklą apibrėžė: „Duomenų saugykla yra į veiklos sritis orientuotų, integruotų ir nekintančių, turinčių laiko matą duomenų rinkinys, naudojamas sprendimų priėmimo procese“. B. Inmon įvardino šiuos duomenų saugykloje saugomų duomenų požymius:

- **Į veiklos objektus orientuoti :** duomenys suteikiantys informacijos apie tam tikra sistemos (įmonės, organizacijos) objektą, o ne apie sistemos vykdomas operacijas .
- **Integruoti :** duomenys surenkami iš įvairių šaltinių į vieną saugyklą ir joje sudaro prasmingą visumą.
- **Turintys laiko matą :** saugykloje saugomos kintančių laike duomenų apibendrintos reikšmės, todėl duomenys turi laiko identifikatorių (pavyzdžiui periodą)/
- **Nekintantys :** duomenys saugykloje nekinta. Gali būti įdedama naujų duomenų, tačiau saugomų duomenų faktai nemodifikuojami.

Nors šis apibrėžimas paskelbtas beveik prieš 10 metų, jis ir dabar gan tiksliai charakterizuoja duomenų saugyklą.

Kiek paprastesnį apibrėžimą yra pateikęs R. Kimbalas: „Warehouse (duomenų saugykla) yra specialiai struktūrizuotų analizei operacinių duomenų kopija“. Abudu apibrėžimai nėra griežti. Pavyzdžiui duomenys iš saugyklos gali būti ir ištrinami, dėl per didelio jų kiekio ir brangios saugojimo terpės. Duomenų saugyklos schema pateikta 2.1 paveiksle.

2.2.2 Saugyklos sudedamosios dalys ir kūrimo procesas



2.1 Pav. Duomenų saugyklos komponentai

Duomenų kėlimas į saugyklą (angl. *Data Warehousing*) tai procesas reikalingas duomenų saugyklos sukūrimui. Jis susideda iš saugyklos kūrimo, pildymo ir užklausų vykdymo. Procese galima išskirti keletą žingsnių:

- **Duomenų šaltinio identifikavimas.** Norint sukurti duomenų saugyklą reikia turėti tam tinkamus duomenis. Dažniausiai imami kasdien kaupiami ir naudojami duomenys bei „istoriniai“ ankstesnių periodų duomenys, kurie gali būti senose „liktinėse“ sistemose. Tokių duomenų išgavimas kartais gali būti labai brangus procesas.
- **Saugyklos projektavimas ir kūrimas.** Tai procesas, kurio metu kuriama saugykla, Didžiausias dėmesys kreipiamas į tai kokios užklausos saugykloje bus vykdomos. Tam, kad etapas būtų sėkmingas reikalingas kuriamos duomenų struktūros supratimas ir nuolatinis bendravimas su galutiniu sistemos vartotoju. Dažniausiai šis žingsnis atliekamas iteracijomis. Jis turi būti atliekamas itin kruopščiai. Vieną kartą sukūrus

duomenų modelį ir jį užpildžius dideliais duomenų kiekiais vėliau būna labai sunku, o kartais ir neįmanoma tą modelį keisti.

- **Užpildymas duomenimis.** Tai duomenų perkėlimo procesas iš šaltinio į saugyklą. Dažniausiai šis žingsnis yra brangiausias ir ilgiausiai trunkantis. Naudojamos taip vadinamos ETL (Extract/Transform/Load) (Išgauk/Tranformuok/Įdėk) programinės priemonės.
- **Pakitimų sekimas.** Periodinis saugyklos atnaujinimas duomenimis iš operacinės aplinkos. Problemos kyla sekant kuriuos duomenis reikia atnaujinti. Ne visose komercinėse sistemose ši problema sėkmingai išspręsta.
- **Duomenų valymas (cleaning).** Tai procesas vykdomas kartu su saugyklos užpildymu. Jo metu stengiamasi panaikinti neteisingus ar netikslius duomenis. Pavyzdžiui duomenys apie tą patį subjektą gauti iš skirtingų šaltinių gali sintaksiškai nesisieti.
- **Duomenų agregavimas.** Saugykloje gali būti saugomi skirtingo detalumo duomenys. Kai kurios sumos gali būti iš anksto paskaičiuotos. Tada užklausos su šiais duomenimis bus atliekamos žymiai greičiau.

2.2.3 Duomenų analizės saugyklose priemonės

Duomenų saugomų saugyklose analizei naudojamos šios pagrindinės priemonės:

- Sprendimų priėmimo sistemos (*Decision Support Systems (DSS)*)
- Vykdomosios informacinės sistemos (*Executive Information Systems (EIS)*)
- Duomenų gavybos sistemos (*Data Mining*)
- Analitinės sistemos (*On-Line Analytical Processing (OLAP*))

Šiame darbe bus plačiau apžvelgtos ir tiriamos OLAP sistemos.

2.2.4 Saugyklos metaduomenys

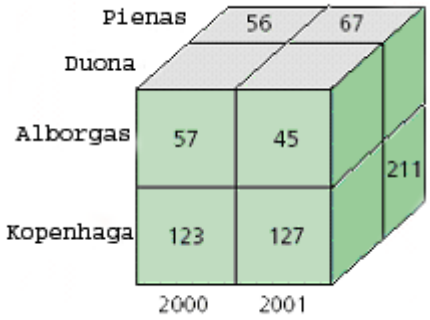
Didelę reikšmę duomenų saugyklų kūrime turi metaduomenų apie operacinę (apskaitos) duomenų bazę ir duomenų saugyklą kaupimas. Metaduomenys – tai duomenys apie duomenis. Juose sukaupta informacija apie naudojamus duomenų tipus, tam tikrų įrašų fizinę ir loginę vietą, jų prasme. Metaduomenys saugomi specialioje saugykloje (angl. *repository*). Jie padeda organizacijai sekti kur kokie duomenys kaupiami, keisti jų struktūrą, padeda išsiaiškinti kokios užklausos duomenų saugykloje gali būti atliekamos.

2.3 Daugiamačiai duomenų modeliai (duomenų kubai).

OLAP sistemose naudojamas daugiamatis duomenų modelis. Šių sistemų charakteristikos skiriasi nuo charakteristikų naudojamų OLTP (angl. *On-line Transaction Processing*) (operacinių duomenų transakcijų apdorojimo) sistemose. OLTP sistemos skirtos vykdyti pasikartojantiems veiksmams su duomenimis, kurių metu atnaujinami detalūs įrašai. Svarbiausias čia yra vienu metu apdorojamų transakcijų skaičius ir duomenų pilnumo bei neprieštaringumo užtikrinimas. Tipinių OLTP sistemų dydis yra šimtų megabaitų ar gigabaitų eilės.[1] Kitaip nei OLTP, duomenų saugyklų sistemos kuriamos pagelbėjimui priimant sprendimus ir turi ilgalaikius istorinius duomenis. Taigi jos yra žymiai didesnės. Čia žymiai svarbesnės paskaičiuotos tarpinės sumos nei detalūs įrašai. Saugyklose svarbiausias yra užklausų vykdymo laikas, o ne atliekamų transakcijų skaičius. Užklausos potencialiai darosi

labai sudėtingos, turinčios daug ryšių tarp lentelių ir agregavimų. Be to jos turi būti vykdomos lentelėse, saugančiose milijonus įrašų, o jų rezultatai turi būti interaktyvūs ir lengvai suprantami verslo analitikams, o ne informacinių technologijų specialistams. Skirtingi reikalavimai OLTP ir OLAP sistemoms lėmė skirtingų duomenų modelių ir realizacijos metodų naudojimą. Netinkamas OLAP sistemose pasirodė ir plačiai OLTP sistemose naudojamas ER (esybių-ryšių) duomenų modelis. Išpopuliarėjo daugiamatis duomenų modelis arba kitaip duomenų kubas.

Duomenų kubai apibendrina duomenų peržiūros lenteles ir leidžia peržiūrėti duomenis daugeliu pjūvių (dimensijų). Dimensija kube yra pirminė koncepcija. Todėl kubai lengvai valdomi, į juos galima įdėti naujus dimensijas. Teoriškai jų skaičius gali būti neribotas, tačiau praktikoje dirbti su kubais didesniais nei sudarytais iš 10-12 dimensijų yra sudėtinga. Kubo dimensijų skaičius apsprendžia jo ląstelių skaičių. 2.2 paveiksle parodytas paprastas 3 dimensijų kubas su vietovių, prekės tipų ir laikotarpių informacija skirtingiems pirkimams. Kubo ląstelėse parodomas vienas matavimas - parduotas prekių kiekis.



2.2 Pav. Trijų dimensijų (prekės tipo, vietovės ir laikotarpio) kubas su pardavimų informacija.

Praktiškai kube vaizduojama dviejų, trijų dimensijų informacija, tačiau ji gaunama agreguojant didesnį dimensijų kiekį.

2.3.1 Dimensijos

Dimensija - viena pagrindinių sąvokų daugiamačiame duomenų modelyje. Kiekviena dimensija saugo informaciją apie tą patį faktą skirtingame kontekste. Tai padeda gauti ar apskaičiuoti kažkokią reikšmę tam tikrame detalizacijos lygyje. Dimensijos dažnai turi hierarchinę struktūra. Pavyzdžiui, nagrinėjant tam tikro laikotarpio pardavimus, savaitės dimensijos jungiamos į mėnesių dimensijas, o šios savo ruožtu į metų ir taip toliau. Tam tikras

duomenų lygis be jį sudarančių dimensijų gali turėti papildomų savybių, pavyzdžiui prekės matavimo vienetą, kuris tam tikroje dimensijoje nekinta ir nedidina dimensijų skaičiaus.

2.3.2 Faktai

Faktas kube vaizduoja subjektą – analizuojamą šabloną ar įvykį. Faktą dažniausiai identifikuoja dimensijų kombinacija. Galutinai faktą identifikuoja dimensijų reikšmių kombinacija. Faktas egzistuoja tik tada, jei yra jį atitinkanti visų dimensijų reikšmių kombinacija. Dauguma daugiamačių duomenų modelių reikalauja, kad faktas turėtų bent jau žemiausio hierarchijos lygio dimensijų reikšmių kombinaciją (Aukštesnio lygio dimensijų reikšmės gali būti gaunamos agreguojant žemesniųjų lygių reikšmes).

Kiekvienas faktas turi savo detalumą. Kurį nulemia jį sudarančių dimensijų detalumas bei vieta dimensijų hierarchijoje. 2 paveikslėlio pavyzdyje fakto detalumas yra: metai pagal miestą ir produktą. Detalumai sudaryti iš aukštesnių arba žemesnių hierarchijų dimensijų yra detalesni (angl. *finer*) arba labiau apibendrinti (angl. *coarser*).

Duomenų saugyklose gali būti kaupiami trijų tipų faktai:

- Įvykiai (angl. *events*) – egzistuoja bent jau žemiausiame detalumo lygyje. Dažniausiai tai realaus pasaulio įvykiai. Juos atitinkantis faktas - konkretus įvykis su savo matavimais. Įvykiu gali būti pardavimas, WEB puslapio nuorodos paspaudimas ir pan.
- Momentiniai vaizdai (angl. *snapshot*) – modeliuoja esybės būseną tam tikru laiko momentu. Pavyzdžiui WEB puslapio vartotojų skaičių tam tikru laiko momentu. Tokia pati būsena gali pasikartoti įvairiais laiko momentais, tačiau duomenų bazėje tai bus atskiri įrašai.
- Kaupiantieji momentiniai vaizdai (angl. *cumulative snapshots*) – saugo informaciją apie veiksmą iki tam tiko momento. Pavyzdžiui pardavimai ar vartotojų vizitai iki šio mėnesio pradžios. Tokia informacija lengvai palyginama su kitų laikotarpių informacija, pavyzdžiui praėjusių metų.[2]

2.3.3 Matavimai

Matavimus sudaro dvi dalys:

- Skaitmeninė išraiška (pavyzdžiui prekės kaina)
- Formulė – dažniausiai parasta aritmetinė išraiška, tokia kaip „*sum“* funkcija, kuri gali apjungti keletą matavimų reikšmių į vieną.[3]

Daugiamačiame duomenų modelyje matavimai dažniausiai išreiškia faktų savybes, kurias vartotojas nori optimizuoti. Tai yra skirtingos reikšmės įvairioms matavimų kombinacijoms. Savybė ir formulė parenkama taip, kad matavimas turėtų prasmę įvairiose dimensijų kombinacijose. Pati formulė yra modelio metaduomuo. Matavimai gali būti skirstomi į tokias klases:

- Sumuojami matavimai (angl. *additive measures*)- tokie matavimai kurie gali būti sumuojami ir jų sumos turės reikšme bet kurioje dimensijų kombinacijoje. Pavyzdžiui, pardavimų sumos, svetainės nuorodų paspaudimai.
- Dalinai sumuojami matavimai (angl. *semiadditive measures*)- matavimai kuriuos galima apjungti kai kuriose dimensijų kombinacijose, tačiau egzistuoja viena ar kelios kombinacijos kai toks apjungimas neteks prasmės. Pavyzdžiui prekių grupių sumavimas galimas produktų ar sandėlio dimensijoje, tačiau neteks prasmės laiko dimensijoje, nes tos pačios reikšmės bus susumuotos keletą kartų.
- Nesumuojami matavimai (angl. *nonadditive measures*)- matavimai kurie negali būti sumuojami nei vienoje dimensijų kombinacijoje. Dažniausiai tai įvairūs vidurkiai.[4]

Sumuojami ir nesumuojami matavimai gali egzistuoti su visais faktų tipais. Tuo tarpu dalinai sumuojami dažniausiai egzistuoja tik kartu su momentiniais vaizdais ir kaupiančiaisiais momentiniais vaizdais.

2.3.4 Užklausos

Daugiamatėje duomenų bazėje naudojamos specialios užklausos kurios gali būti keleto tipų:

- Dalinančios ir projektuojančios (angl. *slice-and-dice*). Šios užklausos mažina kubą. Pavyzdžiui galime paimti tik tą kubo dalį kurijoje yra informaciją apie duoną (2

paveikslas) arba tik apie 2000 metus. Vienos dimensijos reikšmių išrinkimas mažina kubo dimensijų skaičių. Galimos ir labiau apibendrintos užklausos.

- Einančios gilyn (detalizuojančios) (angl. *drill-down*) ir Einančios į viršų (apibendrinančios) (angl. *roll-up*) užklausos. Šie du tipai yra vienas kito inversija. Jos naudoja dimensijų hierarchiją skaičiavimams (dažniausiai sumavimams) atlikti. Pavyzdžiui ėjimas nuo atskirų miestų prie šalies (į viršų einanti užklausa) sumuoja miestų reikšmes ir gauna vieną šalies reikšmę.
- Einančios aplink (angl. *drill-across*) užklausos apjungia kelis kubus turinčius tą pačią dimensiją. Tai atitinka ryšį tarp lentelių įprastinėje reliacinėje algebroje.
- Ranguojančios arba pirmų n / paskutinių n užklausos (angl. *ranking or top n/bottom n*). Grąžina tik tuos įrašus kurie yra tam tikro sutvarkymo viršuje arba apačioje. Pavyzdžiui geriausiai parduodamų kažkuriame mieste prekių dešimtuką.
- Sukančios (angl. *rotating*) užklausos leidžia vartotojui peržiūrėti duomenis, sugrupuotus kitomis dimensijomis. [5]

Einančios gilyn ir į viršų užklausos gali būti apjungtos su dalinančiomis ir projektuojančiomis užklausomis.

2.3.5 OLAP priemonių įgyvendinimo būdai

MOLAP (angl. *Multidimensional Online Analytical Processing*) **-** kubo duomenis saugo daugiamatėje struktūroje. Skaičiavimai saugomi kartu su duomenimis.
Toks duomenų saugojimas leidžia potencialiai greičiausiai vykdyti užklausas. Užklausų vykdymo greitis priklauso tik nuo to, kiek procentų skaičiavimų atlikta iš anksto. MOLAP taikomas kubams, kurie dažnai peržiūrimi ir kur reikia greito atsako į užklausas.

ROLAP (angl. *Relational Online Analytical Processing*) **-** naudojamas kai išrenkami duomenys saugomi reliacinėje duomenų bazėje. Čia saugomi ir skaičiavimai.
Užklausos tokioje duomenų struktūroje vykdomos žymiai lėčiau. ROLAP naudojamas dideliems duomenų kiekiams arba tada, kai užklausos nėra dažnos. Šis būdas tinkamas archyviniuose duomenyse.

HOLAP (angl. *Hybrid Online Analytical Processing*) **-** naudojamas saugant duomenis reliacinėje duomenų bazėje, o iš anksto paskaičiuotas tarpines sumas daugiamatėje struktūroje. Užklauso kurios išgauna susumuotus duomenis šioje struktūroje įvykdomos taip pat greitai kaip ir MOLAP. Užklausos kurios naudoja bazinius duomenis, pavyzdžiui gilyn einančios iki tam tikro fakto, turi gauti duomenis iš reliacinės struktūros ir nėra tokios greitos. Kubai saugomi HOLAP yra mažesni, nei jiems ekvivalentiški saugomi MOLAP struktūroje ir atsako į užklausas greičiau, nei kubai saugomi ROLAP struktūrose, kai užklausiami išanksto susumuoti duomenys. HOLAP taikomas kai dirbama su kubais, kuriuose reikia greito atsako į išanksto susumuotų duomenų užklausas dideliuose duomenų kiekiuose.

2.3.6 Klientinės OLAP priemonės

Klientinės OLAP priemonės yra aplikacijos, atliekančios agreguojančius skaičiavimus (skaičiuojančios sumas, vidurkius, minimumus, maksimumus) ir juos atvaizduojančios. Patys agreguoti duomenys laikomi laikinoje atmintyje (angl. *cache*), OLAP priemonės adresų erdvėje.

Tuo atveju, kai pradiniai duomenys saugomi DBVS serveryje dauguma klientinių OLAP priemonių siunčia į jį SQL užklausas su GROUP BY operatoriais ir gauna serveryje paskaičiuotus duomenis.

Dauguma šių priemonių pateikia klasių bibliotekas ar jų komponentus, leidžiančius kurti aplikacijas, realizuojančias paprasčiausią OLAP funkcionalumą. Taip pat dauguma kompanijų pateikia AcitveX ir kitas priemonių valdymo elementų bibliotekas.

Klientinės OLAP priemonės dažniausiai taikomos turint nedidelį matavimų skaičių (dažniausiai rekomenduojama ne daugiau šešių) ir nedaug reikšmių juose. Gauti agreguoti duomenys turi tilpti aplikacijos adresų erdvėje, o didinant matavimų skaičių šių duomenų dydis auga eksponentiškai. Todėl beveik visose klientinėse OLAP priemonėse realizuota galimybė paskaičiuoti, kiek atminties prireiks užpildant tam tikrą duomenų kubą.

Klientinėms OLAP priemonėms taip pat būdinga galimybė išsaugoti skaičiavimus faile. Tai leidžia pasinaudoti jau atliktais skaičiavimais kitą kartą, bei perduoti šiuos duomenis kitoms sistemoms[6].

2.3.7 OLAP serveriai

Serverinėse OLAP sistemose buvo toliau išplėtota skaičiavimų išsaugojimo idėja. Jose agreguotų duomenų saugojimą ir keitimą bei šių duomenų saugyklos valdymą atlieka atskira taikomoji programa ar operacinės sistemos procesas, vadinamas OLAP serveriu. Kliento aplikacijos gali siųsti užklausas į šią daugiamačių duomenų saugyklą ir gauti reikiamus duomenis. Kai kurios klientinės aplikacijos netgi gali kurti tokias saugyklas arba keisti jų turinį, pasikeitus pradiniams duomenims.

Serverinių OLAP priemonių privalumai yra panašūs į privalumus naudojant serverines DBVS lyginant su failinėmis, viename kompiuteryje saugomomis duomenų bazėmis. Naudojant OLAP serverį skaičiavimai ir agreguotų duomenų saugojimas vykdomi serveryje, o klientinės aplikacijos gauna tik užklausų pasiųstų šiam serveriui rezultatus. Tai leidžia sumažinti tinklo apkrovimą, užklausų vykdymo laiką ir sistemos resursų, reikalingų kliento aplikacijai kiekį. Dauguma klientinių OLAP priemonių gali kreiptis į OLAP serverį ir gauti duomenis iš jo. Tokiu atveju šios priemonės tiesiog atlieka OLAP kliento vaidmenį.[7]

2.4 Duomenų saugyklos pasaulio tinkle

WEB Duomenų saugyklos (WEB Warehousing) apibrėžiamos kaip WEB ir duomenų saugyklų technologijų kombinacija, išryškinanti kiekvienos šių technologijų privalumus. Tikslesnis apibrėžimas:

WEB duomenų saugyklos: tai požiūris į kompiuterinių sistemų projektavimą, kurių pagrindinės funkcijos yra informacijos (duomenų, teksto, grafikų, piešinių, garsų vaizdų ar kitokių daugiaterpės aplinkos (angl. *multimedia*) objektų) identifikavimas, katalogizavimas, išgavimas (gali būti) saugojimas ir analizė, naudojant WEB technologiją tam, kad vartotojas galėtų lengviau rasti ir efektyviau analizuoti informaciją.[8]

Šis apibrėžimas apima kelis svarbius aspektus:

- WEB duomenų saugyklos (angl. *Web Warehousing*) yra architektūra, kuri apibrėžia įrankių ir procesų, naudojamų WEB technologijų pagrindu kuriant duomenų saugyklas, aibę.

- Duomenys saugomi WEB duomenų saugykloje gali būti ne tik tekstinio, bet ir grafinio, garsinio ar kitokio pavidalo.
- WEB duomenų saugyklos nekuria informacijos. Jos dirba su ja.
- WEB duomenų saugyklos valdo informacijos vienetus, bet jų nesurenka. Tuo WEB duomenų saugyklos skiriasi nuo įprastinių, nes pirminė duomenų saugyklų funkcija yra duomenų rinkimas, identifikacija ir saugojimas. WEB saugykla gali neturėti šių funkcijų. Tai priklauso nuo to, kiek pastangų reikės norint WEB saugyklą padaryti prieinamą vartotojams.

WEB saugyklos paveldėjo dalį savybių iš įprastų duomenų saugyklų. Pavyzdžiui, orientaciją į veiklos analizės sritis. Tačiau jose atsirado ir daug naujų savybių, perimtų iš WEB technologijų: pavyzdžiui, greita ir paprasta duomenų prieiga. 2.1 lentelėje pateikiamos pagrindinės WEB saugyklų savybės, perimtos iš duomenų saugyklų ir WEB technologijų.

Charakteristikos perimtos iš duomenų saugyklų		**Charakteristikos perimtos iš WEB technologijų**	
Charakteristika	**Aprašymas**	**Charakteristika**	**Aprašymas**
Į subjektą orientuota	Duomenys organizuojami taip, kad apibūdintų dalykinės srities subjektus	Paprasta prieiga	Paprasta prieiga per įprastinę WEB naršyklę
Integruota	Informacija integruota ir išvalyta nuo nereikalingų duomenų	Išplėstas interaktyvumas	Interaktyvi sąsaja tarp vartotojo ir WEB saugyklos
Nekintanti	Informacija tik skaitoma. Vartotojai jos neatnaujina	Paskirstyta	Skaičiavimai paskirstyti tarp kompiuterių
Laiko matas	Informacija priklauso nuo laiko mato	Saugumas	Duomenys apsaugomi santykinai atviroje sistemoje

2.1 Lentelė. WEB saugyklų charakteristikos

2.5 Analizės išvados

1. Dideli transakcijų vykdymo sistemose (angl. *OLTP*) sistemose sukaupiami duomenų kiekiai negali būti efektyviai apdorojami be specialiai tam skirtų priemonių: lentelių transformavimo, perkėlimo į duomenų saugyklą, daugiamačių kubų formavimo ir jų peržiūros. Šias priemonės ir aptartos analitinėje dalyje.
2. Didelių duomenų kiekių analizei geriausiai tinka daugiamatis duomenų modelis, kuris skirtingai nei reliaciniai modeliai, yra optimizuotas analitinių užklausų vykdymui. Šio modelio esmė yra duomenų kubai, sudaryti iš faktų ir dimensijų lentelių bei turintys tam tikrus matavimus.
3. Transformuoti reliaciniai duomenys saugomi duomenų saugykloje.. Analizės dalyje nagrinėtos pagrindinės duomenų saugyklų savybės bei jų kūrimo etapai.
4. Pastaruoju metu vis plačiau naudojama duomenų saugyklų atmaina: WEB saugyklos. Aptarta kokios yra duomenų saugyklų perkėlimo į WEB galimybės ir kuo pastarosios skiriasi nuo įprastinių duomenų saugyklų.
5. Duomenų saugyklos gali būti naudojamos ne tik duomenų analizei, bet ir duomenų, ateinančių iš skirtingų šaltinių, surinkimui bei integracijai. Darbe tiriamas pirmasis duomenų saugyklų panaudojimo atvejis.
6. Tirti duomenų saugyklų kūrimo principai bei OLAP priemonių naudojimas jose. Šios priemonės yra tinkamos ir interneto svetainės lankomumo duomenų analitiniam apdorojimui.
7. Darbe giliau bus tiriamos OLAP priemonės MSSQL ir Oracle duomenų bazių valdymo sistemose.

3 Svetainės lankomumo duomenų analizės sistemos modelis

3.1 Formalus daugiamačio duomenų modelio aprašas

Duomenų bazėse plačiausiai naudojamos SQL kalbos užklausos yra pritaikytos reliacinėms duomenų struktūroms. Jos nelabai tinka darbui su daugiamačiais duomenimis. Daugiamačių duomenų apdorojimui įvairūs gamintojai sukūrė daug sistemų, tačiau jose nėra vieningo užklausų modelio. Daugiamačių duomenų peržiūrose egzistuoja du atributų tipai: dimensijų parametrai ir matavimai arba metrikos. Dauguma OLAP produktų matavimus išreiškia kaip dimensijų funkcijas. Tai reiškia kad dimensijų ir matavimų aibės yra statiški. Toks modelis neleidžia vartotojui vykdyti matavimais paremtų užklausų. Kadangi tokios užklausos yra būtinos, reikalingas vienodas matavimų ir dimensijų traktavimas.

Duomenų kubas yra visur pripažįstamas daugiamačio duomenų modelio loginis vienetas (panašiai kaip ryšys tarp lentelių reliaciniame modelyje). Taigi visi daugiamačių duomenų algebroje naudojami operatoriai yra taikomi duomenų kubams.

Apibrėžimas 1: Kubas yra fundamentalus daugiamačio duomenų modelio elementas. Jis yra pagrindinė esybė, naudojama daugiamačių duomenų operatorių įeigai ir išeigai. Kubas apibrėžiamas 4 kortežais : <D, M, A, f>. Taigi yra 4 komponentai charakterizuojantys kubą:

- D- n dimensijų aibė d={d_1, d_2,...d_n}, kur kiekviena d_i priklauso domenui $dom_{dim(i)}$.
- M- k matavimų aibė m={m_1, m_2, … m_k}, kur kiekvienas m_i priklauso domenui $dom_{measure(i)}$
- Dimensijų ir matavimų aibės nesikerta $D \cap M = 0$.
- A- t atributų aibė a={a_1, a_2,… a_t}, kur kiekvienas a_i atributo pavadinimas iš domeno $dom_{attr(i)}$
- Vienas-su-daug vaizdas f: $D \to A$ egzistuoja kiekvienai dimensijai priklausanti atributų aibė. Vaizdas yra toks, kad atributų aibės, priklausančios dimensijoms poromis nesusikerta: $\forall i, j, i \neq j, f(d_i) \cap f(d_j) = 0$

Pavyzdys: pardavimų kubas (3.1. pav.):

D={LAIKAS, PREKĖ, VIETA}

Vartotoją domina matavimai: M={pardavimų_suma, kiekis}

Kubo dimensijos turi tokius atributus: A={diena, metai, mėnuo, prekės_pavadinimas, svoris, spalva, parduotuvės_pavadinimas, miestas, valstybė}

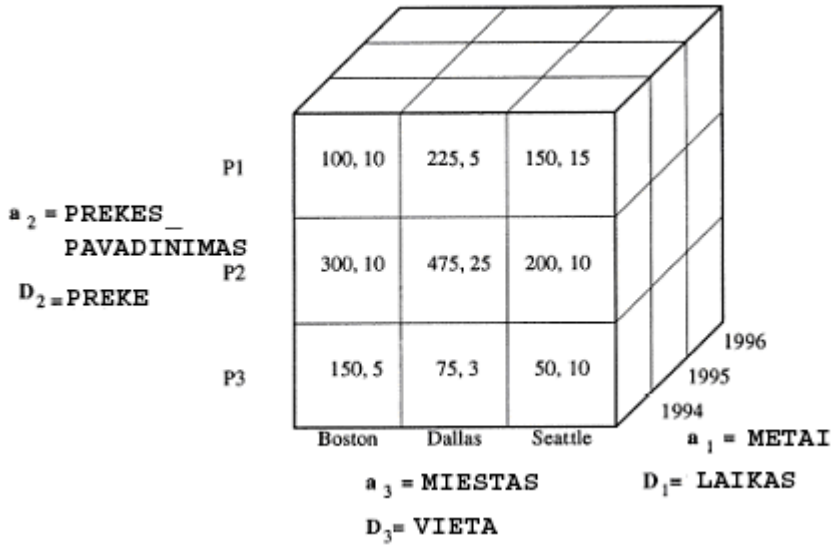
Vaizdas f priskiria atributus konkrečiai dimensijai:

f(LAIKAS) = {diena, metai, mėnuo}

f(PREKĖ) = {prekės_pavadinimas, svoris, spalva}

f(VIETA) = {parduotuvės_pavadinimas, miestas, valstybė}

Reikia pastebėti, kad pateiktos atributų aibės nesikerta.



3.1 Pav. Pardavimų kubas

Aukščiau pateiktas apibrėžimas aprašo abstrakčią kubo struktūrą. Tam, kad kubas materializuotųsi, reikia turėti matavimų reikšmes visose dimensijose. Materializuotas kubas vadinamas kubo-egzemplioriumi (angl. *cube-instance*). Jis aprašomas 6 kortežais: <D, M, A, f, V,g>. D, M, A ir f elementai yra paveldimi iš „tėvinio“ kubo, V yra reikšmių aibė, panaudota kubo materializavimui. Kiekvienas $v_i \notin V$ yra k-kortežis $< \mu_1, \mu_2, ... \mu_k >$ (k - matavimų skaičius), kur kiekvienas μ_i yra i-tojo m_i matavimo egzempliorius. g yra vaizdas: g: $dom_{dim(1)}$ X $dom_{dim(2)}$ X $dom_{dim(n)} \rightarrow V$. Taigi g vaizdas parodo kurios reikšmės su kuriomis konkrečiomis kubo „ląstelėmis“ yra susijusios.

Du kubo egzemplioriai, kilę iš to paties kubo, skiriasi tik 2 kortežais : <V, g>. Toliau pateikiant daugiamačių duomenų modelių algebrą kubu vadinsiu kubo egzempliorių.

OLAP algebroje naudojami šie operatoriai:

Apribojimas (angl. *restriction*)– apribojimo operatorius apriboja reikšmių aibę viename ar keliuose matavimuose.

Sakykime turime predikatą P sudarytą iš atominių predikatų *p*:
P=p_1 <op> p_2 <op> … <op>p_i (<op> loginis operatorius $\wedge$ *arba* $\vee$)

Pradinis (įvedamas) kubas : C_I=< *D*, *M*, *A*, *f*, *V*, *g*> . Panaudoję operatorių gausime kubą C_o=< D_o, M_o, A_o, f_o ,V_o, g_o>, kur D_o=D, M_o=M, A_o=A, f_o=f, $V_o \subseteq$ Vir g_o=g, kuriame kiekvienas elementas $g_p^{-1}(V_p)$ tenkina P.

Matematiškai: $\sigma_p(C_I) = C_O$

Pavyzdys: Norime rasti pardavimus visuose miestuose 1994 metais. Tam pardavimų kube turėsime įvykdyti apribojimo operatorių: $\sigma_{(metai=1994)}(pardavimai)$.

Keletas pastebėjimų:

1) Apribojimo operatoriaus pradinis kubas ir kubas rezultatas abu yra to paties tėvinio kubo egzemplioriai.
2) Apribojimo operatorius realizuoja svarbią OLAP operaciją – išpjovimą (angl. *dicing*), palikdamas tik pradinių reikšmių poaibį tenkinantį tam tikras sąlygas.
3) Jei nėra reikšmių, tenkinančių predikatą P, rezultate gautas kubas yra tuščias.
4) Apribojimo operatorius gali būti taikomas ne tik dimensijose, bet ir matavimuose. Tam kubas turi būti transformuojamas.

Agregavimas (angl. *agregation*)- vykdo aritmetines operacijas viename ar daugiau matavimų. Jis yra sukurtas reliacinės algebros operatoriaus vykdančio MAX, MIN, AVG, SUM funkcijas bazėje ir taikomas kubuose su viena ar daugiau dimensijų sudaromų iš „grupuojančių atributų“ (angl. *grouping atributes*).
Pavyzdžiui, jei naudodamas pardavimų kubą vartotojas norėtų žinoti vidutinę pardavimų sumą per kiekvienus metus, tada metai būtų grupuojantis atributas. Agregacija bus atliekama grupuojant pagal likusių dimensijų atributus: vietovė ir prekės_pavadinimas. Operatorius labai naudingas į viršų einančiose (angl. *roll-up*) užklausose.
Tegul *h* būna agreguojanti funkcija, naudojama vienam matavimui m_i, o *S* aibė grupuojančių atributų $\{a_1, a_2, \ldots a_q\}$, kur $S \subseteq A$. Įvesime vaizdą $\delta : A \to D$, kur δ atvaizduoja a_i į dimensiją d_i. Tada agregacijos operatoriaus algebra bus apibrėžta taip:
Įvestis: kubas C_I=< *D*, *M*, *A*, *f*, *V*, *g*>, matavimas agregacijai m_i ir grupavimo atributų aibe *S*.

Rezultatas: kubas C_o=< D_o, M_o, A_o, f_o ,V_o, g_o>, kur D_o={d_1, d_2, … , d_q}, $q = |S|$ ir $\forall a_i \subseteq S$, $d_i = \delta(a_i)$. Be to M_o={m_i}, $A_o = \bigcup_{\forall d_i \in D_o} f(d_i)$ ir $f_o = f.V_o$ vaizduoja reikšmes gautas pritaikius agreguojančią funkcija h V elementams o g_o yra vazdas: g_o: $dom_{dim(1)}$ X $dom_{dim(2)}$ X $dom_{dim(n)} \rightarrow V_o$

Matematinė notacija: $\alpha_{h,m_i,S}(C_I) = C_o$

Pavyzdys: vartotojas nori sužinoti bendras pardavimų sumas kiekvienam produktui, nepriklausomai nuo miesto. Užklausa gali būti įvykdyta panaudojus tokį operatorių paradavimų kube:

$\alpha_{[SUM(suma),\{prekės\_pavadinimas,metai\}]}(Pardavimai)$

Dekarto sandauga (X): Dekarto sandauga yra dvejetainis operatorius, kuris gali būti panaudotas susiejant du kubus. Operatoriaus algebra apibrėžiam taip:

Įvestis: kubai C_{I1}=< D_1,M_1, A_1,f_1,V_1, g_1> ir C_{I2}=< D_2,M_2, A_2,f_2,V_2, g_2>

Rezultatas: kubas C_o=< D_o, M_o, A_o, f_o ,V_o, g_o>, kur $D_o = D_1 \cup D_2$, $M_o = M_1 \cup M_2$, $A_o = A_1 \cup A_2$, $V_o = V_1 \times V_2$ ir $|V_o| = |V|_1 \times |V_2|$. f_o gali būti gaunamas iš f_1 ir f_2 . Kaip ir kitur g_o yra vaizdas: g_o: $dom_{dim(1)}$ X $dom_{dim(2)}$ X $dom_{dim(q)} \rightarrow V_o$., kur $q = |D_o|$

Matematinė notacija: $C_{I1} \times C_{I2} = C_o$

Pavyzdys: Imkime dar viena kubą *Nuolaidos*, kuriame yra duomenys apie nuolaidas tam tikroms prekėms tam tikruose miestuose. (D={PREKĖ, VIETA}, M={*nuolaida*}, A={prekės_pavadinimas, miesto_id}, prekės_pavadinimas priklauso dimensijai PREKĖ, o miesto_id dimensijai VIETA). Sakykim vartotojas nori žinoti kokios nuolaidos prekei taikomos skirtinguose miestuose. Norint atsakyti į šią užklausą reikia atlikti Dekarto sandaugą tarp *nuolaidų* ir *pardavimų* kubų. Šios sandaugos rezultatas bus atsakymui reikalingos rezultatų aibės superaibė. Taigi norint gauti tikslų atsakymą reikia daugiau operacijų. Dekarto sandaugos operatorius neuždeda jokių apribojimų atsakyme. Atskiras Dekarto sąjungos atvejis yra ryšio (angl. *join*) operatorius leidžiantis nurodyti apribojimus .

Ryšio operatorius (angl. *join*) – tai ypatingas Dekarto sandaugos operatoriaus atvejis, naudojamas susieti dviems kubams, turintiems po vieną ar dvi tokius pačias dimensijas ir identiškus atitikmenis į tų dimensijų atributų rinkinius. Kitaip tariant du kubus

$C_1 = \langle D_1, \ , \ , f_1, \ , \ \rangle$ ir $C_2 = \langle D_2, \ , \ , f_2, \ , \ \rangle$ galime susieti ryšiu *join,* jei $D_1 \cap D_2 \neq 0$ ir $\forall d_1 \in (D_1 \cap D_2), f_1(d_1) = f_2(d_2)$.

Matematinė notacija: $C_1 \otimes C_2 = \sigma_p (C_1 \times C_2)$

Pavyzdys: Imkime praeito pavyzdžio užklausą: vartotojas nori sužinoti, kokias nuolaidų sumas jis gautų kiekviename mieste iš *pardavimų* kubo. Atsakymas yra užklausa, kuri gaunama susiejus ryšio operatoriumi *pardavimų* ir *nuolaidų* kubus – (*Pardavimai* $\otimes$ *Nuolaidos*).

Kiti du operatoriai yra dvejetainiai operatoriai, kurių įvestis turi būti du kubai, kuriems galima pritaikyti **sąjungos** operaciją. Kitaip tariant, jie turi turėti tiek pat dimensijų ir matmenų bei tarp dimensijų ir matmenų tuose dviejuose kubuose yra atitikmuo 1 prie 1, neformaliai – kubai turi tą pačią struktūrą.

Sąjungos operatorius $(\cup)$**:** sąjungos operatorius suranda dviejų kubų sąjungą. **Matematinis apibrėžimas:** Įvestis: Kubai $C_{I1} = \langle D_1, M_1 A_1, f_1, V_1, g_1 \rangle$ ir $C_{I2} = \langle D_2, M_2 A_2, f_2, V_2, g_2 \rangle$, kuriems galima pritaikyti sąjungos operaciją. Rezultatas: Kubas $C_0 = \langle D_0, M_0 A_0, f_0, V_0, g_0 \rangle$, toks, kad $D_0 / M_0 / A_0 = D_1 / M_1 / A_1 = D_2 / M_2 / A_2$ ir $V_0 = V_1 \cup V_2$.

Matematinė notacija: $C_{I1} \cup C_{I2} = C_0$

Pavyzdys: Imkime du kubus – *Rytų_Pardavimai* ir *Vakarų_Pardavimai***,** kurių struktūra yra tokia pati kaip *pardavimų* kubo, o kube *Rytų_Pardavimai* saugomi duomenis apie pardavimus šalies rytų regione, kube *Vakarų_Pardavimai* duomenys apie pardavimus vakarų regione. Tarkime vartotojas norės apibendrinti pardavimų duomenis iš abiejų regionų viename kube. Šiam apibendrinimui tiks sąjungos operacija: *Rytų_Pardavimai* $\cup$ *Vakarų_Pardavimai***,** kurios rezultatas yra vienas kubas su duomenimis iš kiekvieno jungiamo kubo arba su tais duomenimis, kurie bendri abiem kubams (jei pastarasis atvejis galimas).

Skirtumo (angl. ***difference*** **) operatorius (-):** skirtumo operatorius suranda skirtumą tarp dviejų kubų. **Matematinis apibrėžimas:** Įvestis: Kubai $C_{I1} = \langle D_1, M_1 A_1, f_1, V_1, g_1 \rangle$ ir $C_{I2} = \langle D_2, M_2 A_2, f_2, V_2, g_2 \rangle$, kuriems galima pritaikyti sąjungos operaciją. Rezultatas: Kubas $C_0 = \langle D_0, M_0 A_0, f_0, V_0, g_0 \rangle$, toks, kad $D_0 / M_0 / A_0 = D_1 / M_1 / A_1 = D_2 / M_2 / A_2, f_0 = f$ ir $V_0 = V_1 - V_2$. Skirtumo operatorius panaikina tą kubo C_{I1} dalį, kuri bendrą abiem kubams.

Matematinė notacija: $C_{I1} - C_{I2} = C_0$

Pavyzdys: Imkime du kubus –*Vakarų_Pardavimai* ir *CA_Pardavimai***,** kurių struktūra yra tokia pati kaip *pardavimų* kubo, kube *Vakarų_Pardavimai* duomenys apie pardavimus vakarų regione, o kube *CA_Pardavimai* turime duomenis apie pardavimus Kalifornijos valstijoje. Tarkime vartotojas nori išimti Kalifornijos pardavimų duomenis iš Vakarų regiono pardavimų. Tam naudojama skirtumo operacija: *Vakarų_Pardavimai* **-** *CA_Pardavimai.*

Pastaba: Papildomą sankirtos (*intersection*)- $(\cap)$ operaciją galima išreikšti naudojant skirtumo operatorių: $C_{I1} - (C_{I1} - C_{I2}) = C_0$. Sankirtos operatorius nėra pagrindinis/ fundamentalus operatorius, kadangi jį galima išreikšti naudojant kitus operatorius. Patogumo dėlei sankirtą galima išreikšti ir taip: $C_{I1} \cap C_{I2} = C_0$

Kiti du operatoriai yra priskiriami **transformacijos** operatoriams. OLAP užklausoms dažnai reikia, kubo matmenis traktuoti kaip dimensijas, ir atvirkščiai. Traukos (*pull*) ir stūmimo (*push*) operatoriai naudojami tokioms transformacijoms atlikti.

Traukos operatorius (angl. *pull* - ϕ) – konvertuoja matavimus į dimensijas. Tarkime, kad D_R yra dimensijų aibė $D_R = \{d_{R1}, d_{R2}, ..., d_{Rq}\}$. Tarkime, kad R matų rinkinys, toks, kad $R \subseteq M$. Apibrėžiame papildomą ryšį $\kappa : R \longrightarrow D_R$, kur κ atitinka matavimo $m_1 \subseteq R$ atvaizdavimą į dimensiją $d_1 \subseteq D_R$.

Matematinis apibrėžimas: Įvestis: Kubas $C_I = \langle D, M\ A, f, V, g \rangle$, matavimų rinkinys R transformacijai, dimensijų aibė D_R ir ryšys tarp dimensijų κ . Rezultatas: Kubas $C_0 = \langle D_0, M_0 A_0, f_0, V_0, g_0 \rangle$, kuriame $D_0 = D \cup \kappa(d_{R1}); M_0 = M - R; A_0 = A \cup f_0(\kappa[d_{R1}])$.

Matematinė notacija: $\phi_{[R, D_{R,\kappa}]}(C_1) = C_0$.

Pavyzdys: Tarkim vartotojas nori sužinoti kokių produktų parduota daugiau nei 100. Kadangi pardavimų apimtis yra matmuo, negalima pritaikyti apribojimo operacijos. Pirmiausia reikia pritaikyti **traukos (*pull*)** operatorių:

$\phi_{[pardavimu\_apimtis, \{Pardavimai\}, \kappa(pardavimu\_apimtis) = Pardavimai]}(Pardavimai)$. Ši operacija sukuria naują dimensiją *pardavimai* ir naują dimensijos atributą – **pardavimų_apimtis.** Dabar galima atlikti apribojimo operaciją: $\sigma_{(pardavimu\_apimtis > 100}(Pardavimai)$.

Stūmimo operatorius (angl. *push* - ψ) – konvertuoja dimensijas į matavimus.

Matematinis apibrėžimas: Įvestis: Kubas $C_I = \langle D, M\ A, f, V, g \rangle$ ir dimensijos į kurią transformuojame pavadinimas D_t. Rezultatas: Kubas $C_0 = \langle D_0, M_0 A_0, f_0, V_0, g_0 \rangle$, toks, kad $D_0 = D - d_t; M_0 = M \cup f(d_t); A_0 = A - f_0(d_t)$ ir $f_0 = f$.

Matematinė notacija: $\psi_{d_t}(C_1) = C_0$.

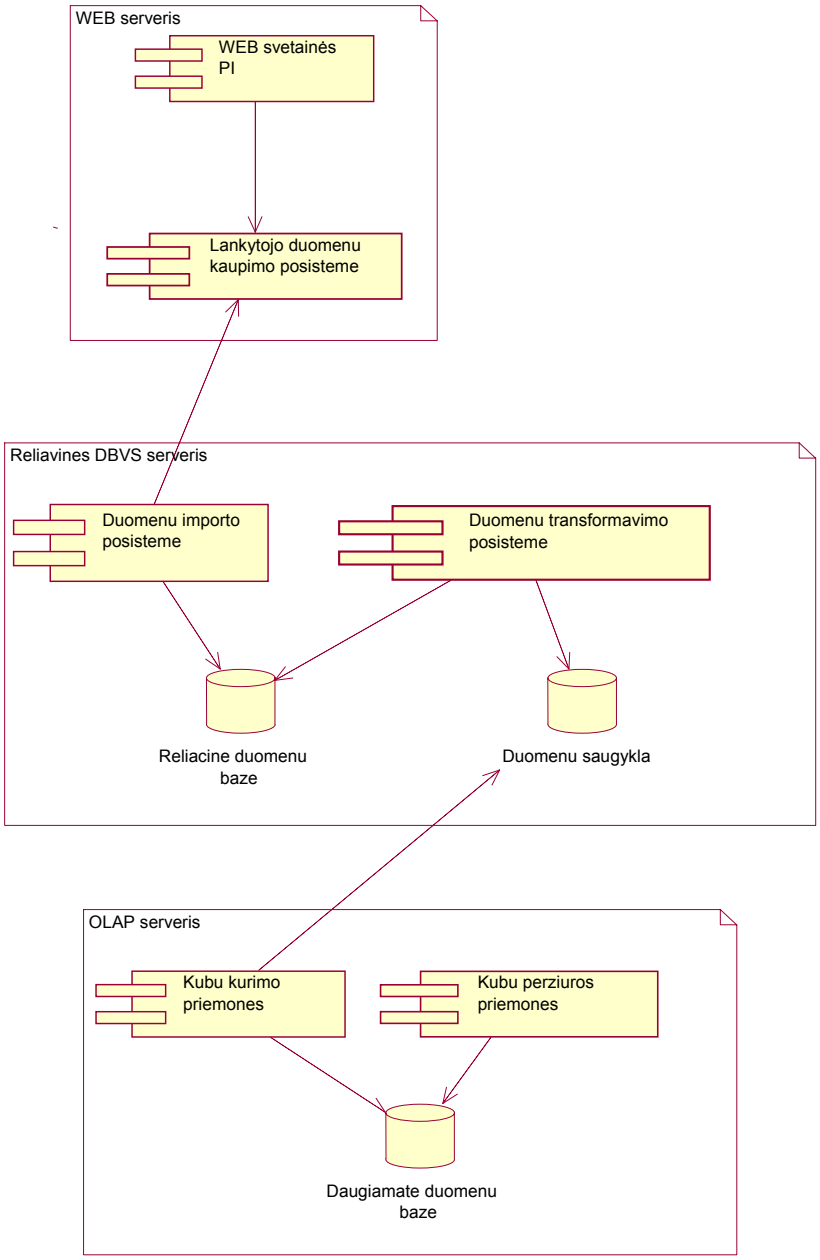
Pavyzdys: tęsiant prieš tai buvusį pavyzdį, tarkime, kad vartotojas nori *įstumti pardavimų_apimtį* atgal į kubą, kaip matavimą. Reikia taikyti **stūmimo (*push*)** operatorių: $\psi_{Pardavimai}(Pardavimai)$[9].

Šis formalus aprašas tinkamas operacijoms su daugiamačiais duomenimis aprašyti bet kokioje daugiamačiu duomenų modeliu pagrįstoje analitinio duomenų apdorojimo sistemoje.

3.2 Svetainės duomenų analizės sistemos architektūrinis modelis

Analizės sistema susideda iš trijų pagrindinių dalių (3.2 pav.):

- **WEB serveryje** įdiegta interneto svetainės programinė įranga, kurios paslaugomis naudojasi vartotojai. Šioje sistemoje failuose kaupiami duomenys apie joje besilankančius vartotojus. Svetainės lankomumo duomenys bus analizuojami, naudojant MS SQL ir Oracle OLAP įrankius, tai yra, jie tarnaus kaip eksperimentiniai duomenys, tiriant OLAP savybes.
- **Reliacinės DBVS serveris** – sukaupta informacija toliau perduodama šiam serveriui, kur ji patalpinama į reliacines struktūras, iš kurių transformuojama į saugyklos duomenų modelį (snaigės tipo schemą). Transformuoti duomenys saugomi duomenų saugykloje.
- **OLAP serveris –** šiame serveryje duomenys iš saugyklos perkeliami į daugiamatę duomenų bazę. Tokioje bazėje analitinės užklausos vykdomos žymiai greičiau nei kreipiantis į duomenų saugyklą esančią reliacinėje duomenų bazėje.

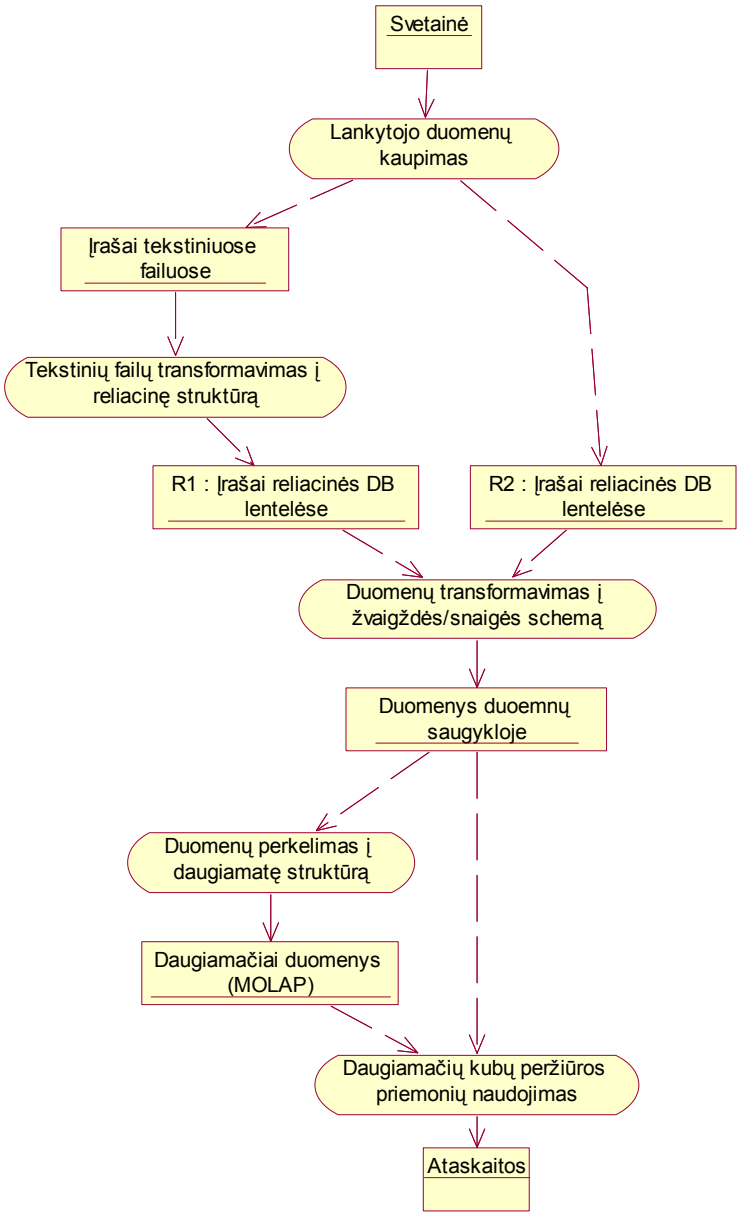


3.2 Pav. Svetainės lankomumo duomenų analizės sistemos architektūrinis modelis

Aprašytas architektūrinis modelis yra labai bendras, jis tinka bet kuriam interneto svetainėje sukauptų duomenų analizės uždaviniui. Visos šios sistemos dalys gali būti viename arba keliuose kompiuteriuose.

3.3 Duomenų perkėlimo į saugyklą ir pateikimo analizės įrankiuose procesas.

Šiame darbe svetainės lankomumo duomenys bus analizuojami OLAP priemonėmis. Pradžioje jie kaupiami tekstiniuose failuose - lankomumo žurnaluose. Toliau aprašytas šių duomenų transformavimo procesas, kurio rezultate jie patenka į OLAP priemones ir tampa prieinami analitikui (pvz., svetainės administratoriui). Procesas pavaizduotas UML veiklos diagrama (3.3 paveikslas).



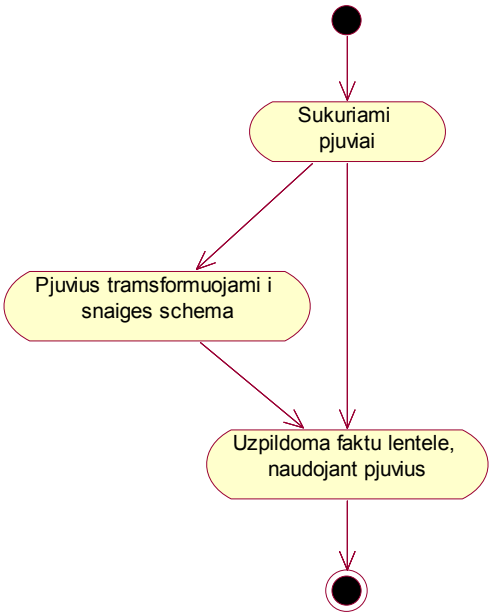
3.3 Pav. Duomenų transformacijų seka

Transformavimo žingsnių aprašymas:

1. Lankomumo duomenų kaupimas – Daugumoje internete veikiančių sistemų fiksuojamas kiekvienas puslapio parodymas. Galime sužinoti kokiu metu ir koks svetainės puslapis buvo žiūrėtas. Kartais turime ir daugiau informacijos: koks vartotojas puslapį žiūrėjo, kokia geografinė vartotojo būvimo vieta, jo amžius ir t.t.
Dažniausiai šie duomenys rašomi į tekstinius failus. Tai mažiausiai sistemos resursų reikalaujanti ir greičiausiai atliekama duomenų saugojimo operacija. Kai kuriose svetainėse, kurių apkrautumas (vartotojų skaičius) santykinai nedidelis, lankomumo duomenys gali būti rašomi tiesiai į kokios nors DBVS reliacines lenteles. Tokiu atveju tekstinių duomenų failų transformavimo į reliacinę struktūrą žingsnis praleidžiamas.

2. Tekstinių failų transformavimas į reliacinę struktūrą – Norint analizuoti tekstinių failų pavidalu sukauptus lankomumo duomenis pirmiausia juos reikia transformuoti į reliacinę struktūrą. Tekstiniuose failuose duomenys atskiriami tam tikrais skyrikliais (dažniausiai kableliais arba kabliataškiais) skyrikliais atskirti duomenys importuojami į atitinkamus reliacinės lentelės stulpelius. Kiekvienas naujas įrašas faile pradedamas nauja eilute. Taigi, lentelėje po importo turėsime tiek pat įrašų, kiek eilučių buvo tekstiniame faile.

3. Duomenų transformavimas į žvaigždės/snaigės struktūra – Iš tekstinių failų importuoti duomenys patenka į vieną duomenų bazės lentelę. Tam, kad galėtume juos analizuoti naudojant OLAP priemones, duomenys turi būtų žvaigždės arba snaigės daugiamatėje duomenų struktūroje. Formuojamos dimensijos, kuriomis duomenis norėsime nagrinėti, bei faktų lentelė. Pirmiausiai suformuojamos dimensijos atrenkant skirtingus (angl. *DISTINCT*) duomenis, pagal kuriuos darysim dimensijas iš bendros lankomumo duomenų lentelės. Tada formuojama faktų lentelė, kurioje kiekvienas įrašas atitinka viena svetainės puslapio parodymą. Visų rišančiųjų su dimensijomis stulpelių duomenys gaunami paimant (angl. *look-up*) atitinkamus identifikatorius iš dimensijų lentelių. Šį procesą detaliai galima pavaizduoti veiksmų diagrama pateikta 3.4 paveiksle.



3.4 Pav. Duomenų transformavimas į žvaigždės struktūrą

4. Duomenų perkėlimas į daugiamatę struktūra (MOLAP). Daugiamačius duomenis galima peržiūrėti vykdant užklausas reliacinėje duomenų bazėje (ROLAP). Tačiau procesas žymiai pagreitėja duomenis perkėlus į daugiamačių duomenų saugojimo bazę ir paskaičiavus kai kurias sumas (MOLAP). Šiame žingsnyje reliacinius duomenis perkeliame į daugiamatę saugyklą. Kai kuriose sistemose toks perkėlimas gali būti ir neatliekamas. Pvz. MS SQL Analizės servisuose (angl. *Analysis services*) daugiamačiai duomenys būtinai turi būti transformuojami į MOLAP kubus tolimesniam apdorojimui. ORACLE duomenų bazės naujausioje versijoje 9iR2 daugiamačių duomenų užklausos atliekamos tiesiog reliacinėje duomenų bazėje. Skaičiavimams pagreitinti gali būti naudojamos tarpinės iš anksto paskaičiuotos lentelės - materializuoti vaizdai (angl. *Metherialized Views*). Taigi šiuo atveju turime ROLAP arba HOLAP. ROLAP atveju užklausos atliekamos tiesiog reliacinėje duomenų bazėje, o HOLAP atveju prieš jas atliekant dar reikia suformuoti tarpines lenteles.

5. Daugiamačių kubų peržiūros priemonių naudojimas. Galutinis mūsų duomenų transformacijų tikslas – patogiai juos peržiūrėti naudojant analitinės duomenų peržiūros ir ataskaitų generavimo priemones. Šios priemonės daugiamačius duomenis saugomus reliacinėje ar daugiamatėje duomenų bazėje pateikia vartotojui suprantama interaktyvių kubų bei ataskaitų forma.

4 Duomenų analizės priemonių MSSQL serveryje ir ORACLE tyrimas

Šiame skyriuje analizuojamos konkrečios MS SQL serverio ir ORACLE priemonės duomenų analitiniam apdorojimui.

4.1 *Saugyklų kūrimo ir OLAP priemonės MS SQL duomenų bazių valdymo sistemoje.*

4.1.1 Duomenų transformacijų servisai.

MSSQL duomenų bazių valdymo sistemoje duomenų transformacijoms, importavimui ir eksportavimui naudojami DTS (angl. *data transformation services*) (duomenų transformavimo servisai). Jie puikiai tinka ir OLTP sistemos duomenų perkėlimui į duomenų saugyklą, kurioje vėliau bus atliekama duomenų analizė.

DTS pateikia priemonių rinkinį, kuris leidžia išgauti, transformuoti ir susieti duomenis esančiuose skirtinguose šaltiniuose, prisijungimą prie kurių palaiko DTS prisijungimų posistemė.

DTS paketas tai prisijungimų prie duomenų bazės, DTS užduočių (angl. *tasks*) ir DTS transformacijų rinkinys. Taip pat jame nurodoma DTS darbų seka (angl. *workflow*). DTS paketas dažniausiai saugomas MSSQL duomenų bazės metaduomenų saugykloje (angl. *repository*).

DTS užduotis (angl. *task*) yra diskreti funkcionalumo aibė, vykdoma kaip vienas žingsnis *DTS pakete*. Kiekviena užduotis apibrėžia bendro duomenų perkėlimo darbo sudėtinę dalį, kurios metu perkeliama ar transformuojama kokia nors duomenų dalis arba duomenų bazėje atliekamas koks nors darbas. Duomenų perkėlimo į duomenų saugyklą metu dažniausiai naudojami šie DTS užduočių tipai[10]:

- *Duomenų importas ir eksportas* – DTS gali importuoti/eksportuoti duomenis esančius bet kokiuose OLE DB prieiga pasiekiamuose šaltiniuose, nutolusiuose ir lokaliuose SQL serveriuose bei tekstiniuose failuose.

- *Duomenų transformavimas* – Šis užduočių tipas leidžia paimti bet kokiame duomenų rinkinyje esančius duomenis, arba kombinuoti keliuose rinkiniuose esančius duomenis SQL užklausų pagalba. Duomenys gali būti padedami į kita duomenų rinkinį. Tarp rinkinių galima sudaryti ryšius (angl. *mappings*). Perkeliamų duomenų įrašus galima papildomai transformuoti panaudojant įvairias duomenų konvertavimo bei agregavimo funkcijas.
- *Duomenų bazės objektų kopijavimas* – Leidžia perkelti iš vienos duomenų bazės į kitą tokius duomenų bazėse saugomus objektus kaip procedūros, vaizdai (angl. *views*) ir pan.
- Transact SQL ar ActiveX skriptų vykdymas - DTS leidžia vykdyti įvairius skriptus duomenų šaltinyje arba duomenų imtuve (jei tai yra duomenų bazė su DBVS).

Naudojant DTS servisus galime suformuoti tolesniam duomenų apdorojimui reikalingas struktūras - duomenų kubus. Duomenų kubų užpildymui reliacinėse duomenų bazėse saugomais duomenimis ir kubų peržiūrai naudojami į MS SQL programų paketą įeinantys analizės servisai (angl. *Analysis services*).

4.1.2 Analizės servisai

Visų duomenų saugyklų ir OLAP sistemų naudojimo pagrindinis tikslas yra duomenų analizė bei analizės rezultatų pateikimas vartotojui suprantama ir patogia sprendimų priėmimui forma. Tiesioginis kliento taikomosios programos, pateikiančios analizės rezultatus, kreipimasis į duomenų saugyklą įmanomas, tačiau šiuo atveju taikomojoje programoje turi būti realizuotos analizės priemonės, kitaip tariant ši aplikacija turėtų būti klientinė OLAP priemonė. Progresyvesnis yra OLAP serverių panaudojimas. OLAP serveris yra tarpinė grandis tarp duomenų saugyklos (realizuotos DBVS pagalba) ir kliento aplikacijos. Tokiu atveju OLAP serveris turi versti duomenis iš reliacinio pavidalo, į pavidalą patogesnį analitinių ataskaitų formavimui – OLAP kubus. Microsoft Analizės Servisai realizuoja OLAP serverį.[11]
Pagrindinis šio paketo komponentas yra Analizės serveris (angl. *Analysis Server*) - operacinės sistemos Windows NT/2000 servisas. Šis serveris skirtas OLAP kubų kūrimui iš reliacinės duomenų bazės duomenų bei prieigai prie šių duomenų iš klientinių aplikacijų.

Teoriškai OLAP kubas, sukurtas naudojant Microsoft analizės servisus, gali talpinti visus faktų lentelės duomenis ir agreguotas išraiškas toms įrašų grupėms iš šios lentelės, kurios atitinka viršutinį matavimų hierarchijos lygį[12]. Kai reikia, galima dinamiškai atnaujinti kubą, jeigu faktų lentelėje įvyko duomenų pakeitimai. Taip pat leidžiama pasirinkti ar žemesniu hierarchijos lygių duomenys bus saugomi pačiame kube (atitinka MOLAP duomenų saugojimo modelį), ar bus gaunami iš faktų lentelė (ROLAP ar HOLAP). Kubo duomenis peržiūrinčiam vartotojui nėra jokio skirtumo, koks duomenų saugojimo modelis naudojamas kube.

Analizės servisai saugo tik paprasčiausių agreguojančių funkcijų agreguotus duomenis (sumas, įrašų skaičių minimalias ir maksimalias reikšmes). Tačiau esant reikalui galima sudaryti taip vadinamus skaičiuojamus elementus (angl. *calculated members*) panaudojant žymiai daugiau analitinių funkcijų.

Sukūrus keletą kubų, turinčių tas pačias dimensijas, juos galima sugrupuoti į vieną daugiamatę duomenų bazę, o dimensijas apjungti į vieną biblioteką (angl. *library*). Tokios dimensijos bus prieinamos visiems kubams (angl. *shared dimensions*)

Galiausiai *Analizės servisai* leidžia sudaryti taip vadinamus virtualius kubus, kurie yra vaizdų (angl. *views*) analogas reliacinėje duomenų bazėje.

4.1.3 Kubų saugomų Analizės servisuose peržiūros priemonės

Kubo naršyklė (angl. *Cube Browser*).

Tai paprasta priemonė, leidžianti peržiūrėti analizės serveryje sukurtus kubus. Jos pagalba galima atlikti visus pagrindinius veiksmus, taikomus duomenų kubams. Ji leidžia vizualiai keisti kubo dimensijas, eiti gilyn arba aukštyn kubo dimensijų lygiais. Norint išmokti naudotis šia priemone nereikia jokių ilgų apmokymų. Ji yra interaktyvi ir aiški.

Pivot Tables komponentas.

Tai yra COM+ komponentas, kurį galime panaudoti Microsoft Office taikomosiose programose, pavyzdžiui Excel. Jis leidžia prisijungti prie Analizės serveryje saugomų duomenų kubų, vykdyti čia daugiamačių duomenų (MDX) užklausas ir pateikia duomenis lenteline forma arba grafikais. Šio komponento privalumas, kad jis pilnai integruojasi į tokias vartotojams įprastas programas kaip Microsoft Excel. Juo paprasta naudotis tiems, kas turi bent bazines Microsoft Excel žinias. Šis komponentas kaip ir *Kubo Naršyklė* leidžia keisti

kubo dimensijas, kubo detalumus (atlikti einančias gilyn (angl. *drill-down*), ir apibendrinančias (angl. *drill-up*) užklausas).

4.2 Saugyklų kūrimo ir OLAP priemonė ORACLE duomenų bazių valdymo sistemoje

4.2.1 Duomenų saugyklų kūrėjas (angl. *Warehouse Builder*)

Saugyklų kūrėjas yra bendra duomenų saugyklų ir verslo analizės sistemų projektavimo ir realizacijos priemonė. Joje apjungiami pagrindiniai duomenų išgavimo, transformacijos ir įdėjimo (ETL) komponentai ir projektavimo aplinka.
Saugyklų kūrėjas architektūriškai susideda iš dviejų komponentų: kūrimo ir vykdymo aplinkų. Kūrimo aplinka dirba su saugyklos metaduomenis, o vykdymo aplinka su fiziniais duomenimis.

Pagrindinės saugyklų kūrėjo funkcijos:

- Duomenų šaltinių aprašų importas.
- Duomenų bazės schemos projektavimas ir kūrimas
- Duomenų perkėlimo ir transformavimo tarp šaltinio ir imtuvo aprašymas
- Priklausomybių tarp ETL procesų aprašymas
- Duomenų šaltinių aprašymų tvarkymas ir atnaujinimas
- Analitinių (ad-hoc) užklausų aplinkos kūrimas
- OLAP aplinkos kūrimas

Saugyklų kūrėjas generuoja DDL ir PL/SQL kodus kurie vėliau vykdomi ORACLE duomenų bazėse. Šie kodai optimizuojami, norint pasiekti kuo didesnį duomenų bazės produktyvumą.

Duomenų saugyklų kūrėjo naudojimas teikia tokius privalumus:

- **Greitas kūrimas:** Saugyklų kūrėjas sumažina kūrimo laiką. Jis pateikia lengvai naudojamus vizualius redaktorius, vedlius ir iš anksto paruoštų transformacijų bibliotekas.

- **Centralizuotas kūrimas:** visa informacija apie sistemą saugoma vienoje centralizuotoje saugyklų kūrėjo saugykloje (angl. *repository*).
- **Sumažina laiką reikalingą pakeitimams:** sistemos gyvavimo ciklo valdymo priemonės, paremtos vieninga saugykla užtikrina sklandų palaikymo procesą.
- **Neklaidingas kodas:** kadangi kodas generuojamas vienoje vietoje, jis yra ne tik be klaidų, bet ir lengvai perkuriamas, atnaujinamas ir palaikomas.
- **Mažina investicijas į technologijas.** Naudodamas ORACLE duomenų bazę kaip transformavimo variklį ir duomenų saugojimo vietą, Saugyklų kūrėjas išnaudoja visas ORACLE plėtimo, produktyvumo saugumo bei patikimumo galimybes.[13]

4.2.2 Kubų kūrimo priemonės OEM konsolėje

Visi pagrindiniai ORACLE duomenų bazės administravimo bei duomenų struktūrų keitimo veiksmai gali būti atliekami naudojant vieningą administravimo priemonę – OEM (angl. *Oracle Enterprise Management*) konsolę. Ji taip pat leidžia kurti OLAP dimensijas bei kubus. Kuriant šiuos OLAP objektus galima naudotis vedliais. Į OEM programinių priemonių sudėtį įeina ir duomenų kubų peržiūros priemonė *Cube Viewer*. Jos pagalba galima peržiūrėti ką tik sukurtus duomenų kubus. ORACLE‘e duomenų bazėje kubai neperkeliami į kitą saugyklą (naudojamas HOLAP). Todėl kubus galima peržiūrėti vos tik juos sukūrus. Nereikia jokio papildomo duomenų perkėlimo (angl. *cube processing*). Kubų peržiūroms pagreitinti naudojamos specialios ORACLE duomenų bazės tarpinės lentelės – Materializuoti vaizdai (angl. *Matherialized Views*). Juose gali būti saugomos visos iš anksto paskaičiuotos kubų sumos.

4.2.3 „BI Beans“ OLAP prieiga

BI Beans tai JAVA komponentų rinkinys leidžiantis naudotis ORACLE OLAP API (programinę prieigą). Šie komponentai turi vizualias duomenų kubų atvaizdavimo priemones. Juos galima naudoti tiek įprastinėse JAVA programose tiek JSP technologijos pagalba rašomose internete veikiančiose programose.

5 Interneto svetainės lankomumo duomenų analizės sistemos eksperimentinis tyrimas

Interneto svetainės lankomumo analizės sistemoje naudojamos dvi pagrindinės metrikos: vartotojo vizitų ir puslapio užklausų skaičius. Vizitas – tai vartotojo apsilankymas sistemoje apskritai. Pavyzdžiui, jei vartotojas naršyklėje surinko *http://www.banga.lt/* ir pateko į kažkurį šios interneto svetainės puslapį tai jis atliko vizitą. Puslapio užklausa - tai vartotojo kreipimasis į kažkurį svetainės puslapį. Vieno vizito metu vartotojas gali atlikti daug užklausų skirtinguose puslapiuose. Pagal vizito metu atliekamas užklausas galima daryti išvadas kokia informacija vartotoją labiausiai domina, kuriuos puslapius jis dažniausiai lanko.

Pagrindinės vertės, kurias pageidaujama turėti kiekvienam svetainės puslapiui:

1. Kiek apsilankė naujų vartotojų. (per laiko vienetą)
2. Kiek apsilankė jau pažįstamų (ne naujų) vartotojų (per laiko vienetą)
3. Kiek vartotojų iš visų apsilankiusių nuėjo toliau, į kitus sistemos puslapius (procentais)
4. Kiek vartotojų iš visų apsilankiusių paliko sistemą (iš jos išėjo) šioje vietoje(puslapyje)(procentais).

Papildoma informacija:

Pagal vartotoją:

1. Kiek kartų parodytas konkretus puslapis
2. Vartotojų pasiskirstymas pagal amžiaus grupes (procentais)
3. Vartotojų pasiskirstymas pagal lytį (procentais)
4. Vartotojų pasiskirstymas pagal pomėgius
5. Vartotojų geografinis pasiskirstymas

Visa informacija apie vartotoją turėtų būti nurodyta jo profilyje. Kai kuriose svetainėse to nėra. Taigi ne visada šią informaciją įmanoma gauti. Geografinę informaciją galima gauti pagal vartotojo IP adresą, tačiau tai nėra patikimas būdas. Nėra patikimos informacijos kokios šalies

bei miesto interneto tiekėjui priklauso tam tikras IP adresas. Be to vartotojai gali naudoti tarpines ‘proxy’ tarnybines stotis. Šiuo atveju neįmanoma nustatyti vartotojo geografinės vietos pagal IP.

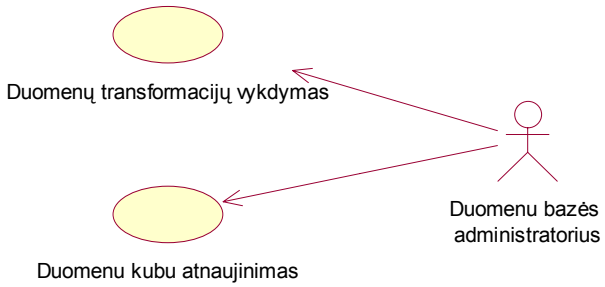
Prototipe bus apsiribota keliomis pagrindinėmis lankomumą charakterizuojančiomis vertėmis:

- Kažkurio puslapio parodymų skaičius
- Parodymo laikas
- Lankytojo amžius
- Lankytojo lytis

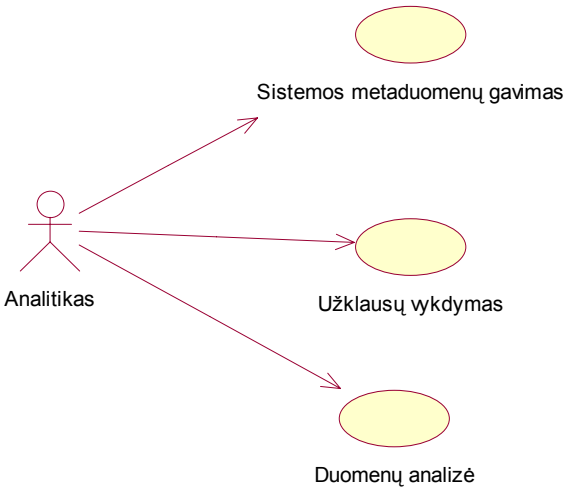
5.1 Svetainės lankomumo duomenis analizuojančios sistemos veiklos aprašymas

Sistemos panaudojimo atvejai:

Duomenų apdorojimo sistemoje egzistuoja dvi pagrindinės vartotojų rolės: administratorius ir analitikas. Administratoriaus sistemos panaudojimo atvejai pateikti 5.1 paveiksle, o analitiko 5.2 paveiksle:



5.1 Pav. Administratoriaus veiksmai

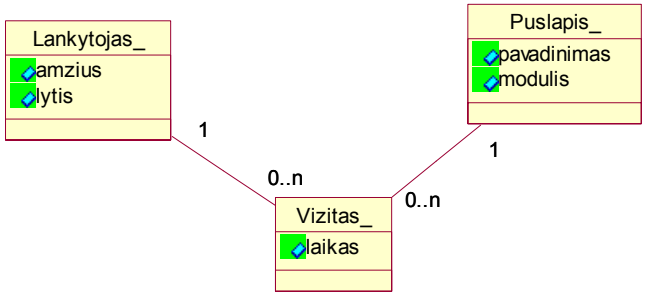


5.2 Pav. Duomenų analitiko veiksmai

Diagramose pateikti nedetalizuoti pagrindiniai sistemos prototipo panaudojimo atvejai. Realioje sistemoje panaudojimo atvejų turėtų būti gerokai daugiau.

Sistemos lankomumo duomenų modelis

5.3 paveiksle pateikta sistemos duomenų modelis.



5.3 Pav. Veiklos klasių diagrama

Svetainės lankomumo duomenų struktūra ir surinkimas

Skirtingo detalumo lankytojų duomenys kaupiami kiekvienoje svetainėje. Netgi jei svetainės kūrėjas neįdiegė duomenų kaupimo posistemės, tokius duomenis kaupia pats svetainę aptarnaujantis WEB serveris. Tačiau WEB serverių kaupiami duomenys yra bendro pobūdžio. Pavyzdžiui juos analizuodami negalime nieko pasakyti apie besijungiantį vartotoją, išskyrus vartotojo IP adresą.

Norint turėti detalesnius duomenis, juos turi kaupti atskira posistemė. Ji turi identifikuoti vartotoją pagal svetainėje saugoma vartotojų duomenų bazę. Tik susiejęs kiekvieną užklausą su

konkrečiu vartotoju galėsime tiksliai analizuoti įvairių vartotojų grupių elgseną svetainėje bei jų poreikius. Tokio pobūdžio duomenys kaupiami ir mano analizuojamoje svetainėje.
Duomenys čia laikomi CSV (angl. *coma separated values*) failuose. Kiekvieną puslapio užkrovimą (užklausą) atitinka viena duomenų failo eilutė. Tekstiniai failai nėra patogi apdorojimui duomenų saugojimo forma. Todėl pirmiausia šiuos duomenis perkeliu į duomenų bazių valdymo sistemą – MSSQL serverį.

Kaupiamų duomenų struktūra

Mano nagrinėjamoje svetainėje viena puslapio užklausa sugeneruoja tokį įrašą lankomumo žurnale:

F;T=1050979533;SID=3de9ed1451b1f;IP=193.219.11.163;UID=1;sys.iamback=1;user_id=80104;user_sex=V;user_bday=1962-04-03 00:00:00;user_mobile=XXXXXXXXXXX;type=F;p_title= ar kimba kur nors kokia nors zuvis?;EV=2notice.ViewAlert|1050831364.E.238.2forum.showPosts.150269.41;REF=;

Laukai kuriuos importuoju į duomenų bazę:

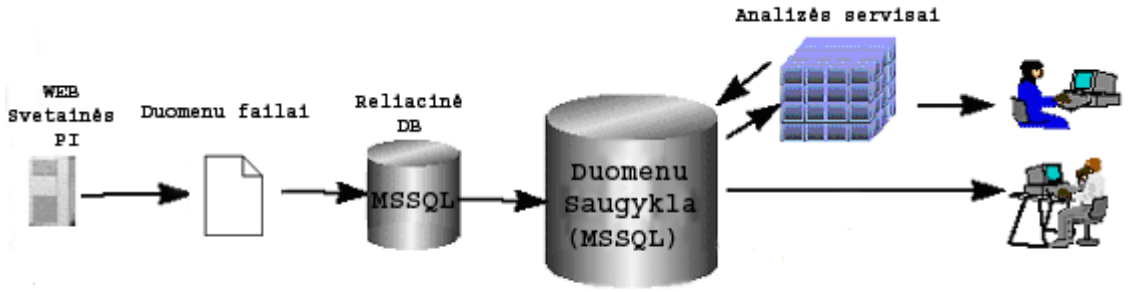
Laukas	Aprašymas
T	apsilankymo laikas išreikštas UNIX TIMESTAMP'u. Jį konvertuoju į MSSQL datos formatą
SID	vartotojo sesijos svetainėje identifikatorius
IP	adresas iš kurio vartotojas jungiasi
User_id	vartotojo identifikatorius svetainėje (jo pagalba vartotojas susiejamas su svetainės vartotojų duomenų bazėje saugoma informacija)
User_sex	vartotojo lytis
User_bday	vartotojo gimimo data
P_title	vartotojo užklausto puslapio pavadinimas
EV	vartotojo užklausos sukeltas įvykis

Įvykiai svetainėje yra panašūs į įvykius įprastoje programoje su grafine vartotojo sąsaja (GUI). Kai vartotojas prisijungia, užpildo formą ar tiesiog paspaudžia nuorodą svetainėje, įvyksta tam tikras įvykis. Įvykių grupės sujungiamos į modulius. Pagal sukeltus įvykius galima nustatyti kokiuose svetainės puslapiuose vartotojas lankėsi ir kokias nuorodas juose spaudė.

Sukūriau du analitinio lankomumo informacijos apdorojimo sistemos prototipus. Vienas jų pagrįstas MS SQL, kitas ORACLE analitinio duomenų apdorojimo priemonėmis.

5.2 Sistemos prototipo įgyvendinimas naudojant MSSQL priemones

Prototipo architektūra parodyta 5.4 paveiksle.



5.4 Pav. Prototipo naudojant MSSQL priemones architektūra

Pagrindinės sistemos dalys kuriamos naudojant į MS SQL paketą įeinančias priemones. Papildomai kuriama tik duomenų transformavimo ir perkėlimo iš WEB svetainės į duomenų bazę priemonė. Ji realizuojama C# programavimo kalba, .NET aplinkoje.

Lankomumo duomenų saugojimui duomenų bazėje sukūriau lentelę, kurios struktūra atkartoja tekstiniuose failuose kaupiamų lankomumo duomenų struktūrą

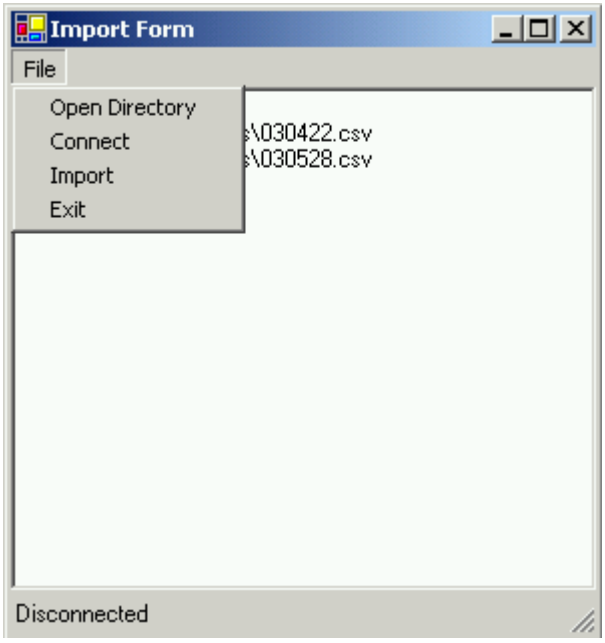
Duomenų bazės lentelės stulpeliai apsilankymų informacijai saugoti:

Stulpelis	Duomenų tipas	Papildoma informacija
ID	Int	IDENTITY (1, 1) NOT NULL
visit_date	Datetime	NULL
sid	char(15)	NULL
ip	char(15)	NULL
uid	Int	NULL
type	char(1)	NULL
iamback	Tinyint	NULL
newbie	char(1)	NULL
user_id	Int	NULL

user_sex	char(1)	NULL
user_bday	Datetime	NULL
p_title	varchar(1000)	NULL
event	varchar(255)	NULL
ref	varchar(1000)	NULL
sh_pattern	varchar(255)	NULL

5.2.1 Duomenų importo į duomenų bazių serverį posistemė

Pagrindinio duomenis importuojančios programos lango vaizdas pateiktas 5.5 paveiksle.



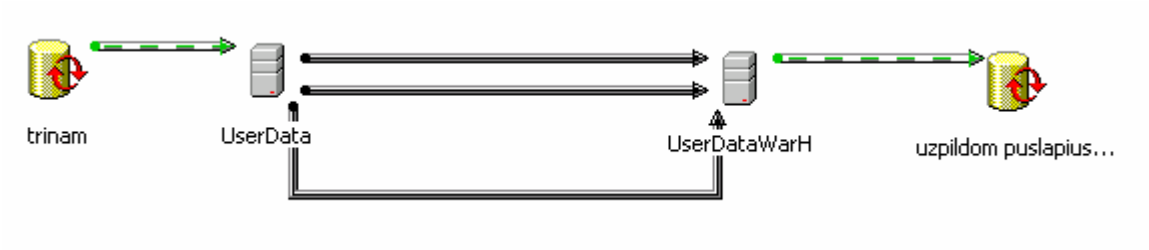
5.5 Pav. Duomenis importuojanti programa

Ši programa nurodytus CSV failus importuoja į MS SQL duomenų bazės lentelę.

5.2.2 Duomenų transformavimas ir perkėlimas į saugyklą

Gautus iš svetainės duomenų kaupiklio "plokščius" duomenis reikia transformuoti į daugiamatę duomenų struktūrą. Tik po tokios transformacijos duomenų rinkinyje bus galima atlikti analitines užklausas. Duomenų transformacijoms naudoju į Microsoft SQL serverio programinių priemonių rinkinį - DTS (duomenų transformavimo servisus).

Dimensijas formuojančio paketo vaizdas:



5.6 Pav. Dimensijas formuojantis DTS (duomenų transformacijų servisų) paketas

Dimensijas formuojančiame pakete (5.6 Pav.) pakete sukuriami du prisijungimai prie duomenų bazių. Viena jų yra duomenų šaltinis, kita duomenų imtuvas. Duomenų transformavimas susideda iš tokių pagrindinių žingsnių:

1. Duomenų imtuvo matavimų lentelių išvalymas.
2. Duomenų perkėlimas. Matavimams reikalingi duomenys gaunami panaudojant grupuojančias SQL užklausas. Šiame pakete sukuriami du matavimai: Vartotojo peržiūrėtų puslapių bei tų puslapių modulių ir užklausų laiko.
3. Duomenų bazės procedūros pagalba puslapiai priskiriami moduliams. (Užpildomas modulio identifikatoriaus stulpelis puslapių lentelėje).

Puslapių lentelė (puslapis):

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
event_id	Int	IDENTITY (1, 1) NOT NULL	Puslapio identifikatorius
Module_id	Int	NULL	Modulio į kurį įeina įvykis identifikatorius
Name	varchar(255)	NULL	Įvykio pavadinimas

PASTABA: puslapio identifikatorius vadinamas „event_id“, nes svetainės programinėje įrangoje kiekvieno puslapio parodymas traktuojamas kaip įvykis.

Modulių lentelė (modulis):

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
Module_id	Int	IDENTITY (1, 1) NOT NULL	Modulio identifikatorius
Name	varchar(255)	NULL	Modulio pavadinimas

Vizitų laikų lentelė (laikas):

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
time_id	Int	IDENTITY (1, 1) NOT NULL	Laiko identifikatorius
time_time	Smalldatetime	NOT NULL	Laikas

PASTABA: laiko momentų įrašai šioje lentelėje surašyti minučių tikslumu. Apvalinimas atliktas ignoruojant sekundinę laiko išraiškos dalį.

Sukuriame dar dvi dimensijas, kuriose duomenys nėra priklausomi nuo svetainės lankomumo įrašų. DTS paketo šių dimensijų užpildymui nekuriame.

Lankytojų amžiaus lentelė (vart_amzius):

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
user_age_id	Int	IDENTITY (1, 1) NOT NULL	Vartotojo amžiaus grupės identifikatorius
user_age	Varchar(50)	NOT NULL	Vartotojų amžiaus grupės pavadinimas.

Į šią lentelę įvedame tokias vartotojų amžiaus grupes:

- **1 - Iki 10 metų**
- **2 - 10-20 metų**
- **3 - 20-30 metų**
- **4 - 30-50 metų**
- **5 - virš 50 metų**

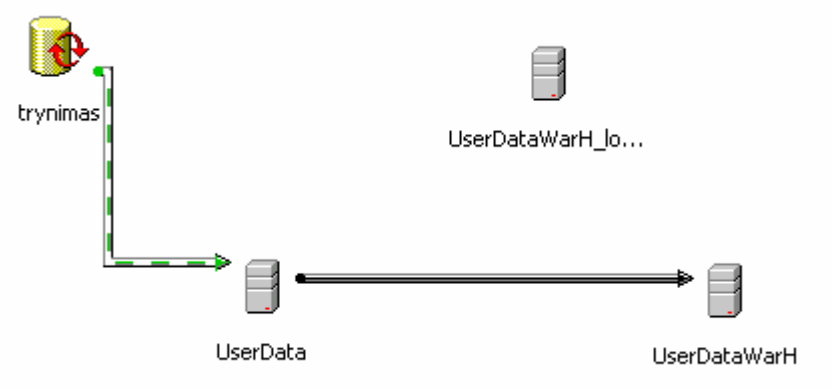
Lankytojų lyties lentelė (vart_lytis):

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
user_sex_id	Int	IDENTITY (1, 1) NOT NULL	Vartotojo lyties identifikatorius
user_sex	Varchar(50)	NOT NULL	Vartotojo lyties pavadinimas.

Į šią lentelę įvedame tokius įrašus:

- **1 - Vyras**
- **2 - Moteris**

Turėdami matavimus, galime suformuoti faktų lentelę. Tam naudojamas kitas DTS paketas (5.7 Pav.)



5.7 Pav. Faktų lentelę formuojantis DTS

Čia naudojama tas pats duomenų šaltinis ir tas pats imtuvas. Pagrindiniai transformavimo žingsniai yra šie:

1. Duomenų imtuvo faktų lentelės išvalymas
2. Duomenų perkėlimas. Šiame žingsnyje labai svarbūs yra susieti (angl. *look-up*) duomenų šaltiniai. Susieto šaltinio pagalba, peržiūrint matavimų lenteles sukuriami faktų ir matavimų lentelių ryšio laukai.

Faktų lentelė (vizitai):

Stulpelis	**Duomenų tipas**	**Papildoma informacija**	**Aprašymas**
ID	Int	NOT NULL	Įrašo identifikatorius
time_id	int	NULL	Laiko identifikatorius
event_id	Int	NULL	Puslapio identifikatorius
user_age_id	int	NULL	Vartotojo amžiaus identifikatorius
user_sex_id	int	NULL	Vartotojo lyties identifikatorius
Click	Int	NOT NULL	Stulpelis nurodantis apsilankymą puslapyje (visada =1)

Faktų lentelėje yra dimensijų identifikatoriai (su dimensijomis rišantys stulpeliai). Čia sukuriamas laukas *click* kuriam visada priskiriamas 1. Sumuojant šio lauko duomenis kube ir gauname įvairius suminius matavimus: apsilankymų skaičių per tam tikrą laikotarpį, apsilankymų skaičių kažkuriame puslapyje ir t.t.

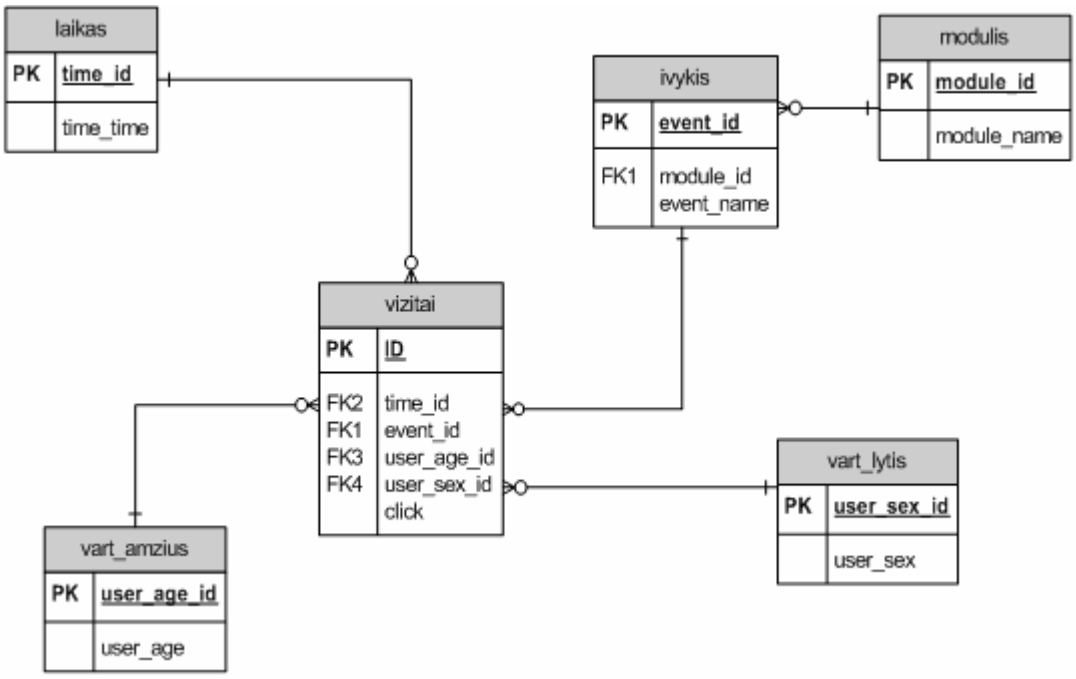
Abi transformacijas (dimensijų lentelių formavimą ir faktų lentelės formavimą) reikia atlikti viena po kitos. Kad nereikėtu atskirai vykdyti kiekvieno duomenų transformacijas atliekančio paketo sukuriame bendrą transformavimo procesą valdantį paketą (5.8 Pav.).

5.8 Pav. Transformaciją valdantis paketas

Atlikę transformaciją turim analitiniam apdorojimui paruoštą duomenų struktūrą. Duomenis perkeliame į Analizės servisų daugiamatę duomenų bazę.

5.2.3 Svetainės lankomumo duomenų analizės kubas

Iš turimų lentelių galime suformuoti kubą kurio struktūra pateikta 5.9 paveiksle.



5.9 Pav. Svetainės lankomumo duomenų analizės kubas.

Tai kombinuotos struktūros kubas. Visos dimensijos, išskyrus puslapio suformuotos pagal žvaigždės schemą. Puslapio dimensija formuojama snaigės schemos principu.

Naudodamasis Analizės servisų kubo kūrimo priemone - kubo redaktoriumi perkeliu šio kubo struktūrą į daugiamačių duomenų serverį. Jame duomenys saugomi MOLAP būdu. Kubo vaizdas kubų redaktoriuje pateiktas 5.10 paveiksle. Formaliai remiantis daugiamačio duomenų modelio algebra šį kubą galima aprašyti taip:

Kubą žymime C_{UD}

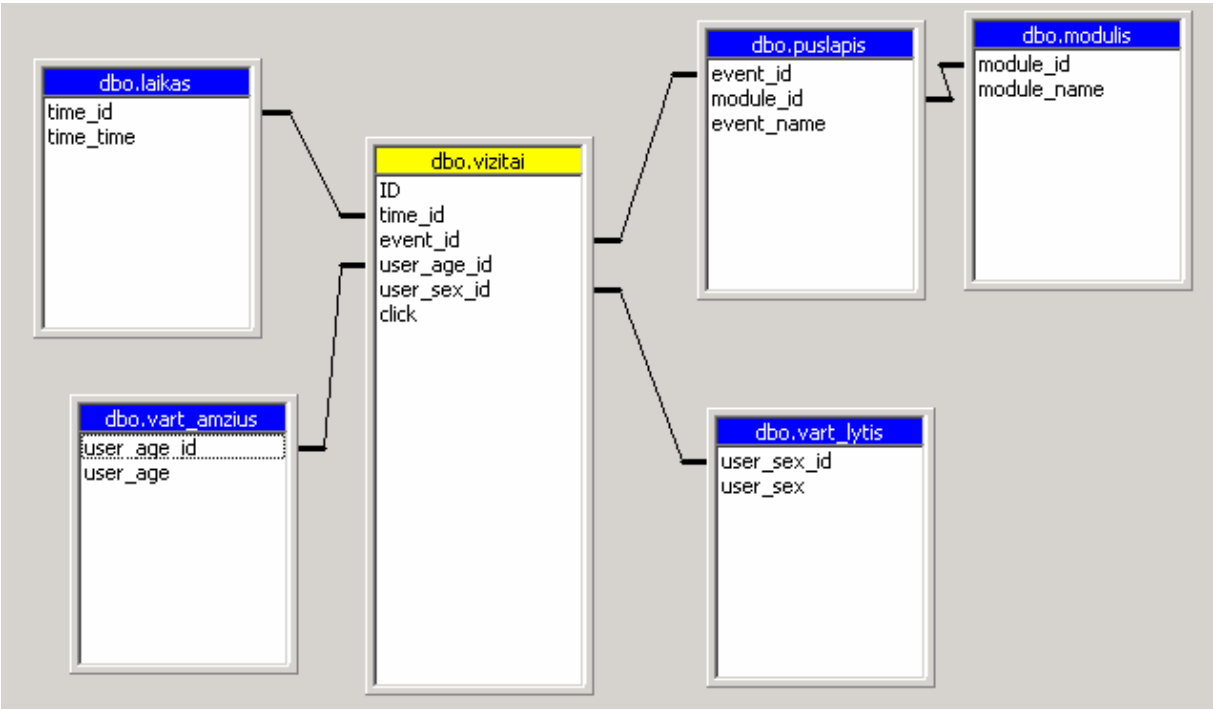
$$C_{UD} =< D_{UD}, M_{UD}, A_{UD}, f_{UD} >$$

$$D_{UD} = \{LAIKAS, PUSLAPIS, VART\_AMZIUS, VART\_LYTIS\}$$

$$M_{UD} = \{click\}$$

$$A_{UD} = \{time\_time, module\_name, event\_name, user\_age, user\_sex\}$$

$$f(LAIKAS) = \{time\_time\}$$
$$f(PUSLAPIS) = \{module\_name, event\_name\}$$
$$f(VART\_AMZIUS) = \{user\_age\}$$
$$f(VART\_LYTIS) = \{user\_sex\}$$

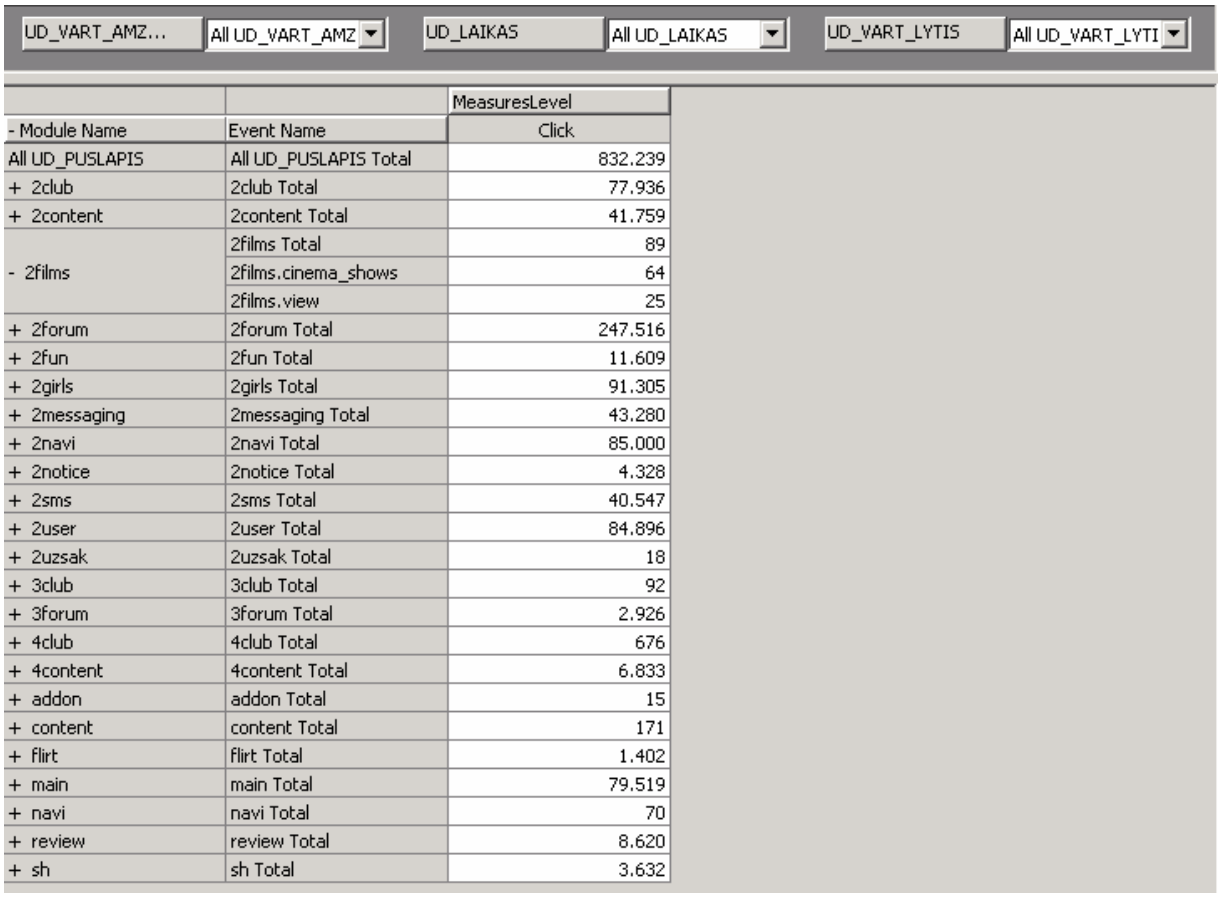


5.10 Pav. Kubo vaizdas kubų redaktoriuje

Analizės servisų kubo naršyklė leidžia peržiūrėti kubo duomenis įvairiomis dimensijomis ir dimensijų kombinacijomis. Pateiksime keletą tokių peržiūrų pavyzdžių

Puslapių dimensija kube.

Puslapių dimensijos vaizdas pateiktas 5.11 paveiksle



UD_VART_AMZ...	All UD_VART_AMZ	UD_LAIKAS	All UD_LAIKAS	UD_VART_LYTIS	All UD_VART_LYTI

		MeasuresLevel
- Module Name	Event Name	Click
All UD_PUSLAPIS	All UD_PUSLAPIS Total	832.239
+ 2club	2club Total	77.936
+ 2content	2content Total	41.759
- 2films	2films Total	89
	2films.cinema_shows	64
	2films.view	25
+ 2forum	2forum Total	247.516
+ 2fun	2fun Total	11.609
+ 2girls	2girls Total	91.305
+ 2messaging	2messaging Total	43.280
+ 2navi	2navi Total	85.000
+ 2notice	2notice Total	4.328
+ 2sms	2sms Total	40.547
+ 2user	2user Total	84.896
+ 2uzsak	2uzsak Total	18
+ 3club	3club Total	92
+ 3forum	3forum Total	2.926
+ 4club	4club Total	676
+ 4content	4content Total	6.833
+ addon	addon Total	15
+ content	content Total	171
+ flirt	flirt Total	1.402
+ main	main Total	79.519
+ navi	navi Total	70
+ review	review Total	8.620
+ sh	sh Total	3.632

5.11 Pav. Puslapių dimensija

Formaliai kube atlikta daugiamatės algebros operaciją :

$$res = \alpha_{[SUM(click),\mathrm{mod}ule\_name,event\_name]}(C_{UD})$$

Kubo pirmame stulpelyje matomas modulių sąrašas. Moduliui *2films* įvykdyta einanti gilyn užklausa (angl. *drill-down*) ir matomi į šį modulį įeinantys įvykiai. Formaliai tai galima būtu aprašyti tokia veiksmų seka:

$$C_{UD1} = \sigma_{[\mathrm{mod}ule\_name=2films]}(C_{UD})$$
$$res = \alpha_{[SUM(click),\mathrm{mod}ule\_name,event\_name]}(C_{UD1})$$

Kubo duomenų (dešiniame) stulpelyje matomos puslapio užklausų sumos moduliams arba konkretiems įvykiams (kai įvykdyta einanti gilyn užklausa).

Kubo naršyklė leidžia peržiūrėti kubo duomenis iš karto dvejomis dimensijomis. 5.12 paveiksle pateiktas kubo vaizdas *PUSLAPIU* ir *LAIKO* dimensijomis.

		- Year	+ Month		
		All UD_LAIKAS	- 2003		
- Module Name	Event Name	All UD_LAIKAS Total	2003 Total	+ April	+ May
All UD_PUSLAPIS	All UD_PUSLAPIS Total	832.239	832.239	435.026	397.213
+ 2club	2club Total	77.936	77.936	38.592	39.344
+ 2content	2content Total	41.759	41.759	20.792	20.967
- 2films	2films Total	89	89	42	47
	2films.cinema_shows	64	64	34	30
	2films.view	25	25	8	17
+ 2forum	2forum Total	247.516	247.516	129.323	118.193
+ 2fun	2fun Total	11.609	11.609	6.873	4.736
+ 2girls	2girls Total	91.305	91.305	52.951	38.354
+ 2messaging	2messaging Total	43.280	43.280	23.552	19.728
+ 2navi	2navi Total	85.000	85.000	44.181	40.819
+ 2notice	2notice Total	4.328	4.328	2.500	1.828
+ 2sms	2sms Total	40.547	40.547	17.184	23.363
+ 2user	2user Total	84.896	84.896	46.373	38.523
+ 2uzsak	2uzsak Total	18	18	12	6
+ 3club	3club Total	92	92	28	64
+ 3forum	3forum Total	2.926	2.926	167	2.759
+ 4club	4club Total	676	676	271	405
+ 4content	4content Total	6.833	6.833	3.633	3.200
+ addon	addon Total	15	15	10	5
+ content	content Total	171	171	82	89
+ flirt	flirt Total	1.402	1.402	776	626
+ main	main Total	79.519	79.519	41.857	37.662
+ navi	navi Total	70	70	29	41
+ review	review Total	8.620	8.620	3.995	4.625
+ sh	sh Total	3.632	3.632	1.803	1.829

5.12 Pav. Kubo vaizdas dviem dimensijomis

Čia įvykdyta operacija:

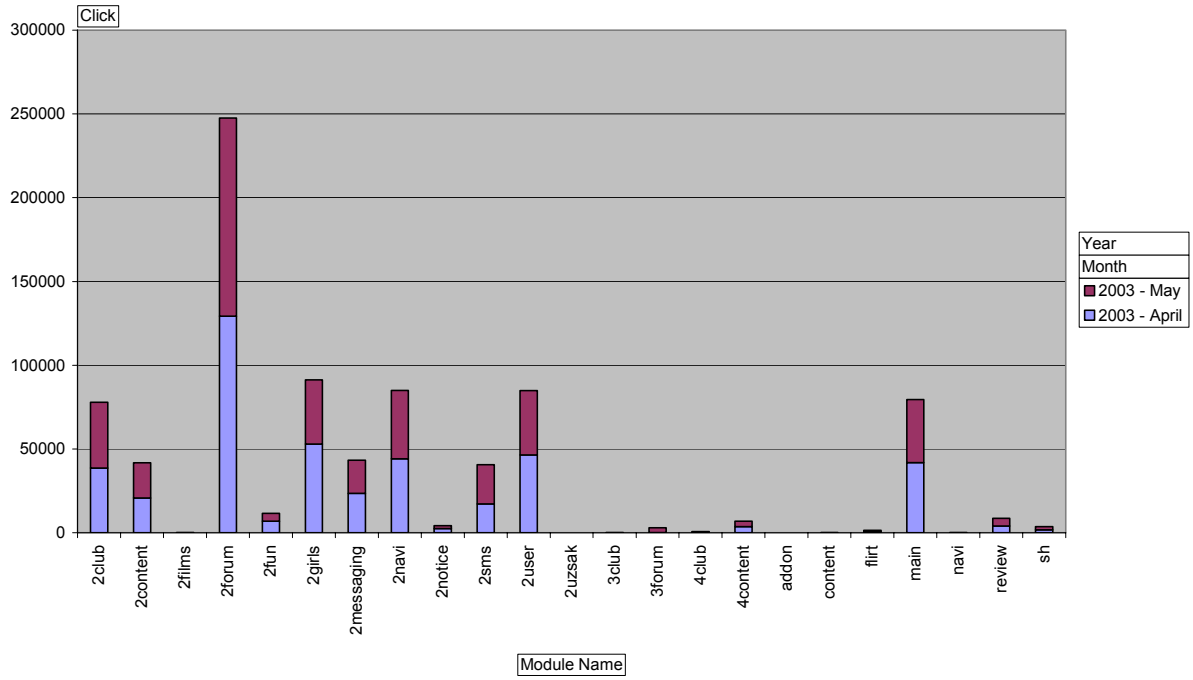
$$res = \alpha_{[SUM(click),\mathrm{mod}ule\_name,event\_name,time\_time]}(C_{UD})$$

Suformuotame kube galime matyti kiek puslapių įeinančių į modulius buvo peržiūrėta kažkurį laiko tarpą. Ėjimas gilyn (angl. *drill-down*) ir ėjimas į viršų (angl. *drill-up*) leidžia peržiūrėti detalesnius arba mažiau detalius laikotarpių ir puslapių modulių duomenis.

5.2.4 Grafinis kubo duomenų atvaizdavimas

Naudojant Microsoft Office sudėtyje esančiomis PivotTable serviso priemones galima suformuoti grafinę gautų duomenų išraišką. Servisas lengviausiai pasiekiamas ir valdomas naudojantis Microsoft Excel. PivotTable ataskaitų generavimo posistemė gali jungtis prie analizės serveryje saugomų duomenų kubų ir išgauti jų duomenis. Sukūriau grafinę duomenų

išraiška naudodamas dvi anksčiau sukurto kubo dimensijas. Šie duomenys pavaizduoti 5.13 paveiksle.

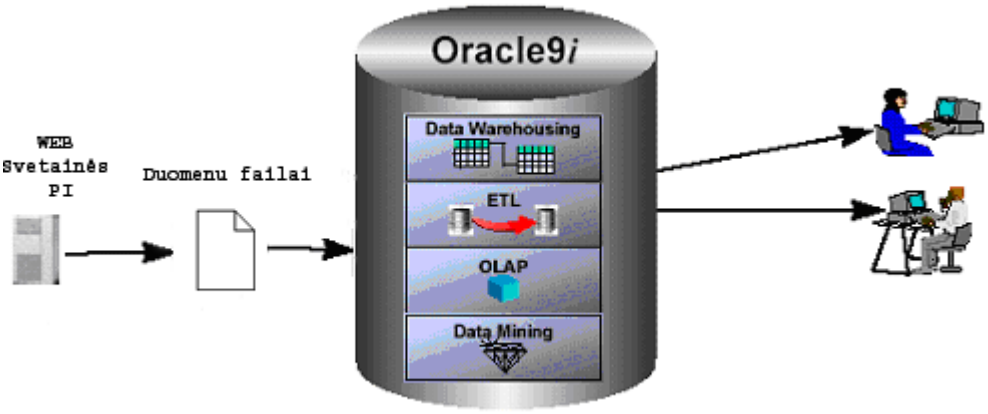


5.13 Pav. Grafinė kubo duomenų išraiška

5.3 Sistemos prototipo įgyvendinimas naudojant ORACLE priemones

Prototipo architektūra

Šio prototipo architektūra skiriasi nuo MS SQL priemonėmis kurto prototipo architektūros, nes analitinio informacijos apdorojimo priemonės ORACLE serveryje įgyvendintos kitaip. Architektūros vaizdas pateiktas 5. 14 paveiksle.



5.14 Pav. Prototipo naudojant ORACLE priemones architektūra

Šiame sistemos prototipe taip pat papildomai panaudotos duomenų perkėlimo iš WEB svetainės priemonės. Kitos analitinio duomenų apdorojimo priemonės įeina į ORACLE duomenų bazės sudėtį

Kad būtų lengviau lyginti, į ORACLE reliacinę duomenų bazę DTS pagalba importuotos lygiai tokios pačias dimensijų ir faktų lenteles, kaip ir dirbant su MSSQL. Tačiau laiko matavimui reikėjo papildomų transformacijų, nes ORACLE dimensijų kūrimo priemonės pačios neišskaido laiko dimensijos iki reikiamo detalumo. Taigi, laiko lentelė įgavo tokią struktūrą:

Lentelė LAIKAS_DET

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
TIME_ID	NUMBER(10)	NOT NULL	Laiko identifikatorius
TIME_LAB	VARCHAR2(16)	NOT NULL	Laiko simbolinė išraiška
YEAR	NUMBER(10)	NOT NULL	Metai
YEAR_LAB	VARCHAR2(4)	NOT NULL	Metų simbolinė išraiška
MONTH	NUMBER(10)	NOT NULL	Mėnuo
MONTH_LAB	VARCHAR2(4)	NOT NULL	Mėnesio simbolinė išraiška
DAY	NUMBER(10)	NOT NULL	Diena
DAY_LAB	VARCHAR2(4)	NOT NULL	Dienos simbolinė išraiška
HOUR	NUMBER(10)	NOT NULL	Valanda
HOUR_LAB	VARCHAR2(4)	NOT NULL	Valandos simbolinė išraiška
MINUTE	NUMBER(10)	NOT NULL	Minutė
MINUTE_LAB	VARCHAR2(4)	NOT NULL	Minutės simbolinė išraiška

Gal būt dėl to, kad Oracle OLAP versija 9.2.0.4 dar nėra sertifikuota, joje blogai veikia vieno lygio dimensijos. Todėl *VART_AMZIUS* ir *VART_LYTIS* lentelėse teko sukurti dar viena porą stulpelių, apibendrinančių pagrindinius šių dimensijų hierarchijos lygius. Tokių būdu dimensijos tapo dviejų lygių:

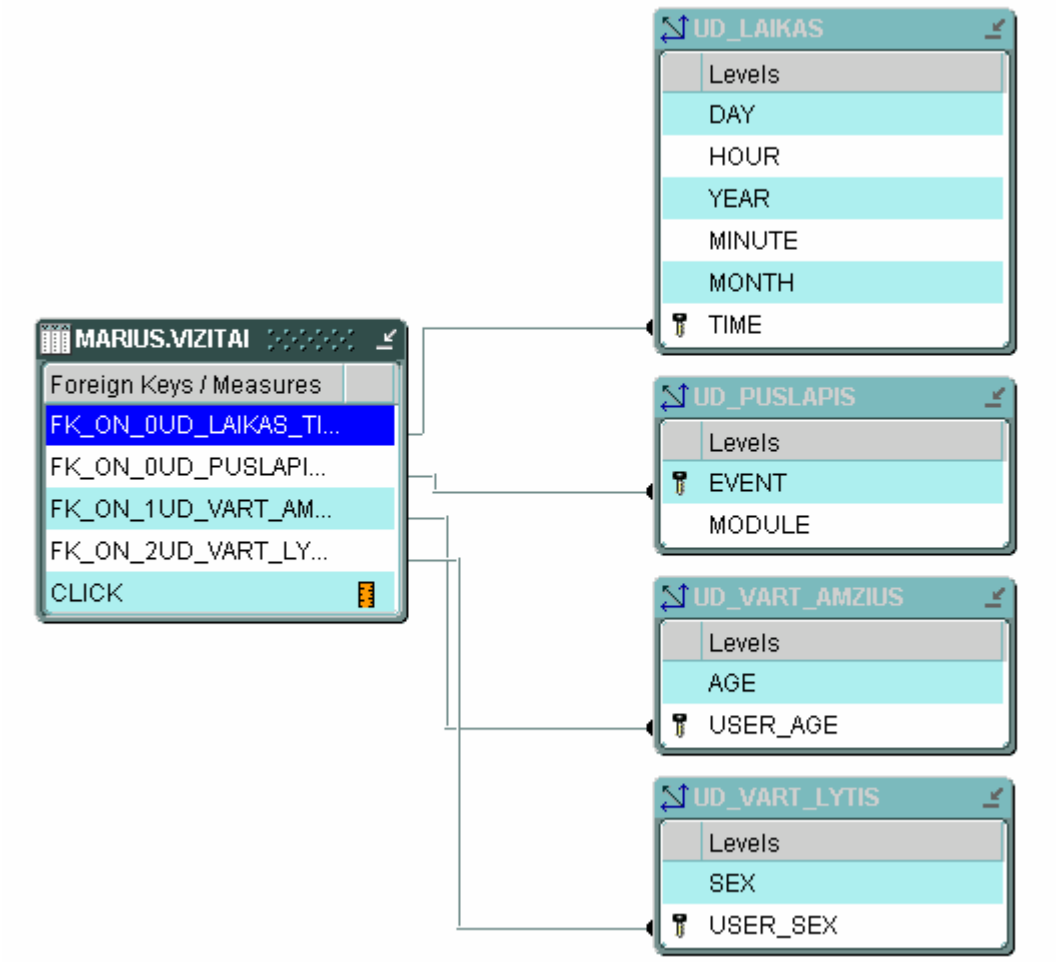
Lankytojų amžiaus dimensijos lentelė *VART_AMZIUS*:

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
USER_AGE_ID	NUMBER(10)	NOT NULL	Vartotojo amžiaus grupės identifikatorius
USER_AGE	VARCHAR2(50)	NOT NULL	Vartotojų amžiaus grupės pavadinimas.
BULK_ID	NUMBER(10)	NOT NULL	Apibendrinančios amžiaus grupės identifikatorius
BULK	VARCHAR2(10)	DEFAULT('Amzius')	Apibendrinančios amžiaus grupės pavadinimas

Lankytojų lyties dimensijos lentelė *VART_LYTIS*:

Stulpelis	Duomenų tipas	Papildoma informacija	Aprašymas
USER_SEX_ID	NUMBER(10)	NOT NULL	Vartotojo lyties identifikatorius
USER_SEX	VARCHAR2(50)	NOT NULL	Vartotojo lyties pavadinimas.
BULK_ID	NUMBER(10)	NOT NULL	Apibendrinantis identifikatorius
BULK	VARCHAR2(10)	DEFAULT('Lytis')	Apibendrinantis pavadinimas

Pasinaudodamas OEM (Oracle Enterprise Manager) priemonėmis aprašiau kubo struktūrą. Ji parodyta 5.15 Paveiksle.



5.15 Pav. Svetainės lankomumo duomenų kubas ORACLE OLAP priemonėse

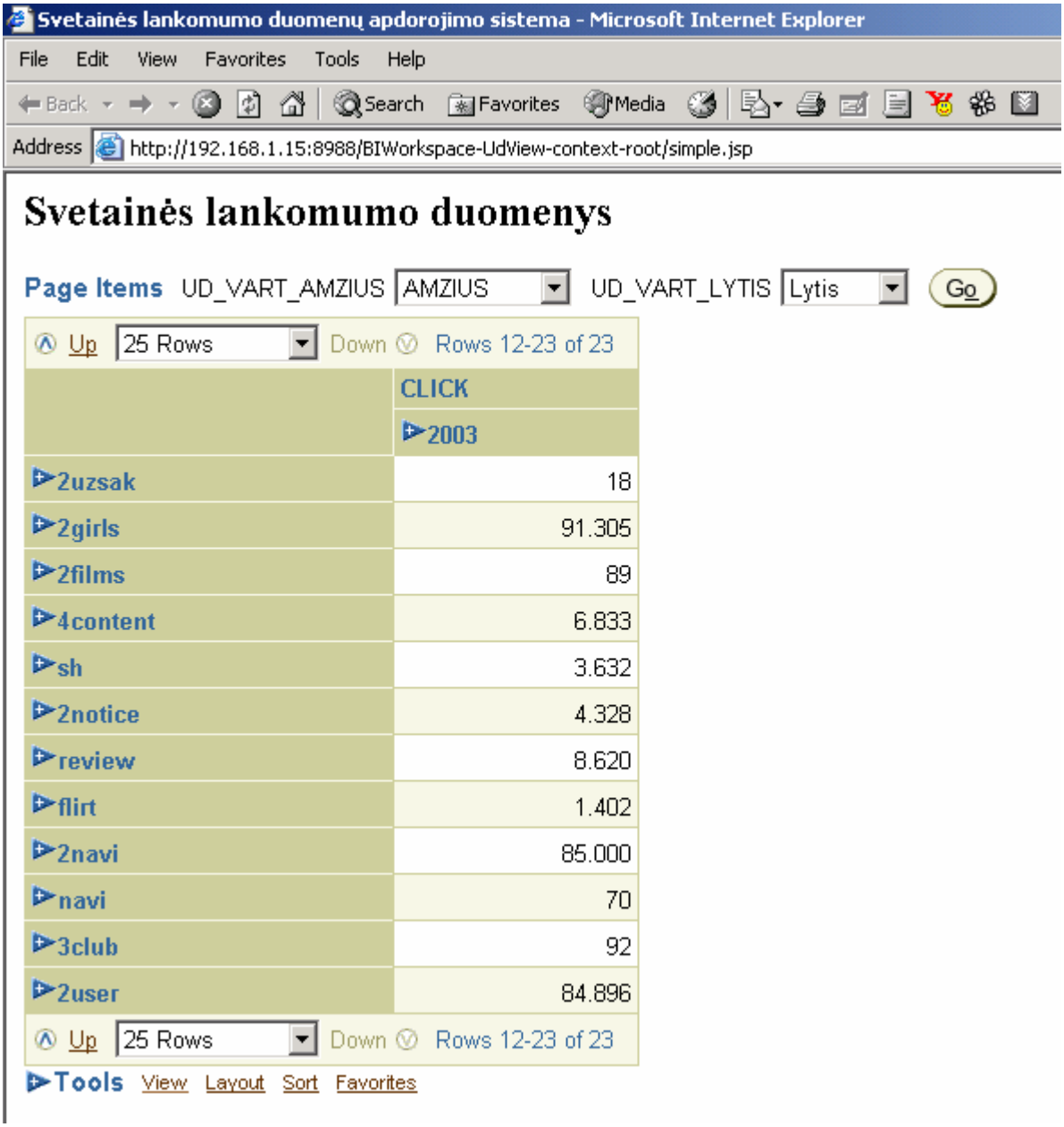
Kuriant dimensijas, ORACLE OLAP priemonės nedetalizuoja jų į lenteles. Dėl to kubas visada atrodo taip, tarsi būtų žvaigždės struktūros. Tačiau fiziškai *UD_PUSLAPIS* matavimas susideda iš dviejų reliacinės duomenų bazės lentelių.

Taip pat, kaip ir naudojant Analizės servisų kubo naršyklę, šį kubą galima peržiūrėti naudojant ORACLE “Kubo vaizduotoją” (Cube viewer). Puslapių dimensijos vaizdas šioje priemonėje pateiktas 5.16 paveiksle.

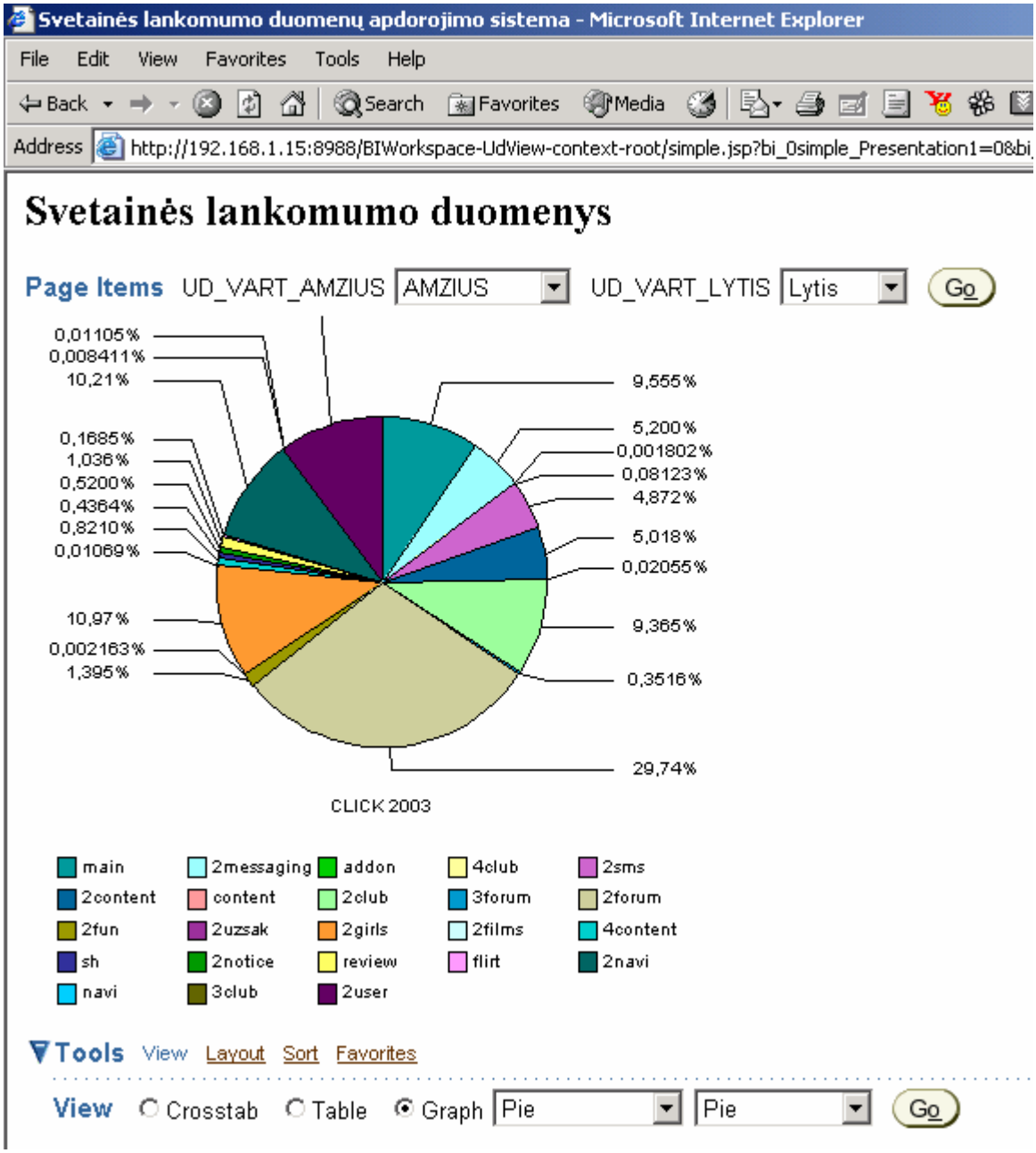
	CLICK		
	2003	4	5
main	87.122	45.772	41.350
2messaging	43.298	23.564	19.734
addon	15	10	5
4club	676	271	405
2sms	40.634	17.223	23.411
2content	41.793	20.801	20.992
content	171	82	89
2club	78.014	38.641	39.373
3forum	2.926	167	2.759
2forum	247.893	129.528	118.365
2fun	11.622	6.882	4.740
2uzsak	18	12	6
2girls	91.333	52.978	38.355
2films	89	42	47
2films.cinema_shows	64	34	30
2films.view	25	8	17
4content	6.833	3.633	3.200
sh	3.644	1.809	1.835
2notice	4.332	2.504	1.828
review	8.621	3.996	4.625
flirt	1.402	776	626
2navi	85.103	44.239	40.864
navi	70	29	41
3club	92	28	64
2user	84.934	46.398	38.536

5.16 Pav. Puslapių dimensija ORACLE Cube Viewer priemonėje

Oracle turi patogia programinę sąsają prieigai prie OLAP duomenų. Tai verslo intelekto komponentai (angl. *BI beans*). Jie leidžia pasiekti duomenų kubuose saugomus duomenis tiek iš paprastų, tiek ir iš interneto taikomųjų programų. Norint pademonstruoti šias galimybes, buvo sukurta paprasta sistema, kubų peržiurai WEB svetainėje. Duomenis galima peržiūrėti lentelinėje (5.17 Pav.) arba grafinėje formoje (5.18 Pav.)



5.17 Pav. Duomenys interneto taikomojoje programoje (lentelinė forma)



5.18 Pav. Duomenys interneto taikomojoje programoje (grafinė forma)

Duomenys čia pateikti Puslapių ir Laiko dimensijomis (Kaip ir ankstesniuose pavyzdžiuose).

5.4 Eksperimentų išvados

1. Sukurti interneto svetainės lankomumo duomenų sistemos prototipai naudojant MS SQL ir Oracle, analitinio duomenų apdorojimo priemones. Taigi šios priemonės yra tinkamos norint sukurti tokio tipo pilnai veikiančia sistemą.
2. Sistema įgyvendinta dviejuose skirtingų gamintojų duomenų bazių valdymo sistemose, naudojant tose sistemose esančias analitinio duomenų apdorojimo priemones. Toks sistemos įgyvendinimas leido palyginti šių priemonių galimybes.
3. Loginis duomenų transformacijų procesas naudojant abiejų gamintojų priemones yra beveik toks pat. Tačiau šių gamintojų architektūriniai sprendimai skiriasi, taigi skiriasi ir sukurtų prototipų architektūra.
4. Sukurtos sistemos nėra išbaigtos. Jos sukurtos OLAP galimybių skirtingose duomenų bazių valdymo sistemose demonstravimui. Šių prototipų pagalba parodoma, kaip galima analitiškai apdoroti interneto svetainės lankomumo duomenis.
5. Kuriant išbaigtą lankomumo duomenų apdorojimo sistemą reiktų sukurti daugiau skirtingų dimensijų ir matavimų. Iš jų galima būtų sudaryti daugiau nei vieną kubą. (Reiktų turėti ne tik puslapio peržiūros (CLICK), bet ir vizitų bei naujų vartotojų apsilankymų skaičiaus matavimus, ne tik puslapių aplankymo, bet ir vartotojų sesijų faktų lenteles).
6. Kuriant realią sistemą reiktų apsispręsti, kurioje sistemoje ORACLE ar MSSQL duomenis bus apdorojami (kurti sistemą paremtą kelių gamintojų OLAP priemonėmis yra netikslinga). Vienų ar kitų priemonių privalumai bei trūkumai nagrinėjami sekančioje šio darbo dalyje.

6 Oracle ir MSSQL duomenų analizės galimybių palyginimas.

6.1 OLAP priemonių įgyvendinimas

Savybė	MSSQL	Oracle
Daugiamačių duomenų saugyklos vieta	Atskiras servisas (Analizės servisai)	Reliacinėje ORACLE duomenų bazėje
Užklausų vykdymo mechanizmas	Vykdomos analizės servisuose	Vykdomos pagrindinės duomenų bazės branduolyje

6.1 Lentelė. Pagrindiniai priemonių architektūrinio įgyvendinimo skirtumai

Microsoft SQL serverio analizės serveriai praktiškai yra atskirti nuo paties DBVS serverio. Tuo tarpu ORACLE kompanija renkasi kitokią strategiją ir naujausioje duomenų bazės versijoje Oracle9iR2 OLAP serverį integravo į pagrindinės duomenų bazės branduolį (6.1 lentelė). Šis sprendimas suteikia tokius privalumus lyginant su dviejų atskirų (reliacinės ir daugiamatės) bazių naudojimu:

- **Supaprastėjęs konfigūravimas ir priežiūra.** Visi priežiūros veiksmai atliekami vienoje duomenų bazėje, naudojant vieną programinę priemonę: angl. *Oracle Enterprise Manager,* PL/SQL konsolę ar k.t.
- **Aukštas patikimumas.** Oracle OLAP turi tas pačias plėtimo bei prieigos galimybės kaip ir Oracle reliacinė duomenų bazė. Oracle OLAP patikimumas yra labai aukštas. Ši sistema gali veikti realiuose sistemų klasteriuose.
- **Aukštas saugumo lygis.** Oracle užtikrina vieningą saugumo mechanizmą visiems duomenims esantiems duomenų bazėje. Tai taikoma ir daugiamačiams kubams. Informacija apie visus reliacinės ir daugiamatės duomenų bazės vartotojus saugoma bendrame kataloge, jiems bendrai suteikiamos atitinkamos rolės ir privilegijos.
- **Atvira prieiga.** Ir reliaciniai ir daugiamačiai duomenys gali būti pasiekiami per SQL arba OLAP API. Programų kūrėjai gali rinktis, ar naudoti OLAP API teikiamus privalumus ar kreiptis į duomenų bazę naudojant paprastas SQL užklausas. [14]

Operacinės sistemos, kuriose veikia priemonės

Savybė	MSSQL	Oracle
Operacinės sistemos, kuriose galima instaliuoti priemones	Tik MS Windows šeimos OS	Visos plačiau paplitusios pramoninės OS

6.2 Lentelė. Operacinės sistemos, kuriose veikia priemonės

Oracle duomenų bazę ir OLAP priemones galima instaliuoti praktiškai visose pramoninėse operacinėse sistemose (Solaris, HP unix, AIX, Linux, Windows 2000 …). MS SQL duomenų bazė ir OLAP priemonės, kaip ir dauguma kitu Microsoft firmos produktų veikia tik Windows operacinių sistemų šeimoje (6.2 lentelė.).

6.2 Bendrųjų MS SQL ir ORACLE OLAP savybių palyginimas

Duomenų perkėlimo bei transformavimo priemonės

Pagrindiniai ETL priemonių skirtumai pateikti 6.3 lentelėje

Savybė	DTS (MS SQL)	Saugyklų kūrėjas (ORACLE)
Pagrindinės metaduomenų saugyklos vieta	MSSQL duomenų bazė (msdb)	ORACLE duomenų bazė (schema bet kurioje bazėje)
Galimos alternatyvios metaduomenų saugojimo vietos	Struktūriniai failai, VisualBasic failai	Struktūriniai failai, OMG CWM standartą suprantančios sistemos
Objektų vykdymo vieta	Operacinė sistema (Windows)	Duomenų bazė (Oracle)
Naudojamos programavimo kalbos	VB script	PL/SQL

6.3 Lentelė. Duomenų perkėlimo ir transformavimo priemonių skirtumai

Didžiausias skirtumas tarp ETL priemonių įgyvendinimo MS SQL ir Oracle yra šiomis priemonėmis sukurtų objektų vykdymo būdas. DTS (duomenų transformavimo servisų) objektai vykdomi operacinėje sistemoje. Taikant tokį vykdymo būdą sumažinama duomenų bazės apkrova bei supaprastinamas šių objektų įdiegimas, tačiau atsiranda daug kitų problemų:

- **Tinklo apkrovimas** (jei objektai vykdomi ne toje pačioje sistemoje kurioje yra duomenų bazės).
- **Prisijungimų prie duomenų bazių nevienareikšmiškumas.** Objektai gali būti vykdomi bet kuriame kompiuteryje kuriam leidžiama prisijungti prie duomenų bazių serverio saugančio DTS. Tačiau jungiantis iš skirtingų potinklių prisijungimų aprašai esantys DTS‘uose gali skirtis. Tokiu atveju duomenų šaltiniai ar imtuvai bus

nepasiekiami, arba dar blogiau, bus naudojami ne tie šaltiniai ar imtuvai kuriuos DTS kūrėjas buvo numatęs naudoti.

- **Vieningos vykdymo terpės nebuvimas.** Nėra apsaugos nuo situacijos, kai tas pats objektas dviejuose skirtinguose kompiuteriuose vykdomas tuo pačiu metu. Tokiu atveju imtuvo duomenys gali būti negrįžtamai sugadinami.

ORACLE ETL priemonėje „Saugyklų kūrėjas“ (angl. *Warehouse Builder*) išvengiama aukščiau išvardintų problemų. Tačiau sukurtų objektų diegimas (ir vykdymas) yra žymiai sudėtingesnis.

Lyginant funkcines šių dviejų ETL priemonių savybes, Warehouse Builder turi didesnes iš anksto sukurtų transformacijų bibliotekas ir leidžia transformacijų pagalba gauti daugiau duomenų bazės objektų tipų. (ne tik lenteles ir vaizdus (angl. *view)*, bet ir dimensijas, kubus ir k.t.)

Kubų peržiūros priemonės

6.4 lentelėje pateikiamos pagrindinės kubų peržiūros priemonių savybės.

Savybė	**Cube Browser (MSSQL)**	**CubeViewer (ORACLE)**
Valdančiosios taikomosios programos pavadinimas	Analizės valdytojas (angl *Analysis Manager*)	Įmonės valdymo konsolė (angl. *Enterprise Management Console*)
Kubo struktūros peržiūros galimybė	Kubo struktūros peržiūrai naudojama kita Analizės valdytojo sudėtinė dalis- Kubo redaktorius (angl. *Cube Editror*)	Kubo struktūra gali būti peržiūrima OEM konsolės kubo redagavimo priemonėmis
Užklausos, išrenkančios vaizduojamus duomenis, keitimo galimybė	Leidžia keisti kubo dimensijas	Leidžia keisti kubo dimensijas bei dimensijų sudėtį (išrenkamus narius)
Vartotojų prieigos prie duomenų valdymas	Prieigos prie kubo valdymui naudojamas rolių valdytojas (angl. *Role Manager*)	Rolės valdomos centralizuotai, kartu su prieiga prie reliacinės duomenų bazės

6.4. Lentelė. Pagrindinės *CubeBrowser* ir *CubeViewer* savybės

Pagrindinės programavimo technologijos ir kvalifikacija, reikalaujama naudojantis OLAP ir saugyklų kūrimo priemonėmis.

Pagrindinės programavimo kalbos, naudojamos analitinio duomenų apdorojimo priemonėse pateiktos 6.5 lentelėje

Savybė	MS SQL	ORACLE
Duomenų bazės užklausų ir saugomų programų kalbos	Transact SQL, MDX	PL/SQL, DML
ETL priemonių pagrindinės kalbos	Transact SQL, VB script	PL/SQL
Rekomenduojamos programavimo kalbos ir technologijos.	.NET priemonės ir šioje aplinkoje palaikomos kalbos (VB, C++, C#)	JAVA 2EE aplinka ir JAVA programavimo kalba

6.5 Lentelė. Saugyklų ir OLAP priemonėse naudojamos programavimo kalbos

Transformuojant duomenis dažniausiai nepakanka paprasčiausių transformacijų. Sudėtingesnėms transformacijoms aprašyti duomenų transformacijų servisuose naudojama VB script programavimo kalba, o Oracle Warehouse Builder'yje PL/SQL'as. Oracle ELT priemonė šiuo požiūrių yra išbaigtesnė. Ta pati programavimo kalba naudojama ir duomenų bazėje saugomose programose (angl. *stored procedures*), trigeriuose bei SQL skriptuose. Tuo tarpu DTS'uose naudojamas VB script yra labiau būdingas pačiai Windows operacinei sistemai. Programavimui SQL serveryje jis nėra naudojamas, o T-SQL kalbos naudojamos duomenų bazėje saugomų programų ir trigerių rašymui konstrukcijos gerokai skiriasi nuo VB script. Veiksmai su DTS objektais nėra atliekami duomenų bazėje. Juos galima laikyti atskiru komponentu. Duomenų saugyklų kūrėjo (angl. *Warehouse Builder*) sugeneruotas kodas vykdomas naudojantis kitokiais principais. Šis kodas negali veikti atskirai nuo duomenų bazės.

Norint išnaudoti visas OLAP priemonių teikiamas galimybes, nepakanka paprasčiausių kubų peržiūros priemonių. Kuriant sudėtingesnes duomenų apdorojimo sistemas reikalingas ne tik OLAP priemonių išmanymas, bet ir programavimo kalbų žinios.

Naudojant Microsoft SQL OLAP priemonės rekomenduojama naudoti tos pačios firmos programavimo priemones t.y. .NET programavimo aplinką ir jos palaikomas programavimo kalbas. Pagrindinė kubų peržiūros priemonė – „Pivot Tables" servisas gali būti programiškai valdomas kaip COM+ objektas.

Dirbant su Oracle OLAP rekomenduojama naudoti JAVA programavimo kalbą. Šia kalba parašyti visi pagrindiniai Oracle komponentai, skirti darbui su duomenų kubais. Kubų peržiūros priemonė „BI beans" taip pat yra JAVA bean programinis komponentas, kurį patogiausia valdyti naudojant JAVA programavimo kalbą.

Kubų peržiūros per WEB galimybės

Kubų peržiūros per WEB galimybės palygintos 6.6 lentelėje.

Savybė	Pivot Tables servisas	BI beans
Jungimosi prie DB būdas	Per WEB servisą iš kliento pusės	Tiesiogiai prie DB iš serverio
Serverio pusės objektai	Neturi	Turi lenteliniai ir grafiniai pateikimo formai
Kliento pusės objektai	Turi lentelinei ir grafinei duomenų pateikimo formai	Neturi

6.6 Lentelė. Kubų peržiūros per WEB galimybės

ORACLE priemonės yra universalesnės, nes norint jomis naudotis nereikia jokių papildomų komponentų kliento pusėje. Duomenis peržiūrėti galima su bet kokia Interneto naršykle, palaikančia HTML ir JavaScript. Tuo tarpu norint peržiūrėti „Pivot Tables“ komponentu paremtus WEB puslapius, šis komponentas turi būti kiekvieno kliento kompiuteryje (instaliuojamas kartu su Office XP). Kadangi komponentas pasiekiamas naudojant COM+, jis gali būti matomas tik Internet Explorer 5.0 (arba naujesnėje) naršyklėje. Naudojant tokį modelį kiekviename kompiuteryje turį būti Office XP (kurio reikia norint naudoti „Pivot Tables“ komponentą) licenzijuota versija. Taigi tuo atveju, kai turime daug darbo vietų tokios technologijos naudojimas yra žymiai brangesnis nei technologijos paremtos „*BI beans“*.

Kubų peržiūros naudojant ofiso programas galimybės

Savybė	MSSQL	Oracle
Kubo peržiūra MS Excel pagalba	„Pivot Tables“ komponentas	Neturi

6.7 lentelė kubų peržiūros galimybės naudojant ofiso programas

Plačiausiai naudojama ofiso analizės programa Microsoft Excel yra to paties gamintojo kaip ir SQL serveris. Taigi šie produktai gali būti integruojami duomenų lygyje. „Pivot Tables“ komponento pagalba MS Excel programoje galima peržiūrėti analizės servisuose saugomus kubus. Oracle priemonės tokių galimybių neturi (6.7 lentelė).

Užklausų kalbos

Užklausoms daugiamačių duomenų kubuose vieningo standarto nėra (tokios kaip SQL kalba reliacinių duomenų užklausoms). 6.8 lentelėje pateiktos MS SQL ir Oracle daugiamačių duomenų užklausų kalbos:

Savybė	MS SQL	Oracle
OLAP užklausų kalba	MDX užklausos	DML kalba. Naudojama daugiau darbui su analitinėmis aplinkomis. Paprastos užklausos atliekamos SQL priemonėmis

6.8 Lentelė. Daugiamačių užklausų kalbos

MS SQL serveryje vykdyti užklausoms duomenų kubuose naudojama MDX (Multidimensional Expressions) (daugiamačių išraiškų) kalba. Ši kalba yra labai panaši į SQL kalbą. Ji leidžia išrinkti duomenis įvairiomis dimensijomis su įvairiais apribojimais. Iš esmės tai yra SQL adaptacija daugiamačiams duomenims. Ji leidžia ne tik peržiūrėti duomenis, bet ir keisti kubus bei jų dimensijas.

Naudojant ORACLE OLAP dauguma veiksmų su daugiamačiais duomenimis galima atlikti naudojantis įprastomis SQL komandomis ir specialiai darbui su šiais duomenimis sukurtais procedūrų ir funkcijų paketais. ORACLE OLAP taip pat turi specialią DML užklausų kalbą. Ji daugiau skirta darbui su daugiamačių duomenų darbo aplinkomis (angl. *workspaces*). Analitinės darbo aplinkos (angl. *analytical workspaces*) yra duomenų bazėje saugomos aplinkos kuriose vykdomas analitinis daugiamačių duomenų apdorojimas.

Dokumentacija ir mokymo priemonės

Abi šios programinės įrangos kompanijos specializuojasi daugelyje veiklos sričių. Todėl jų interneto svetainėse pateikiama labai daug ir įvairios informacijos apie produktus. Sunkumų ieškant informacijos apie vieną ar kitą produktą iškyla abiejose svetainėse. Microsoft kompanijos svetainėje daugiau pavyksta rasti pasinaudojus paieška. Dokumentai čia yra žymiai mažesnės apimties nei ORACLE svetainėje. Tačiau Microsoft dokumentacija yra labai smarkiai struktūrizuota (suskaidyta į dalis) ir dažnai sunku vienoj vietoj surasti pilną informaciją apie vieną ar kitą produktą.

ORACLE dokumentacija suskirstyta atskirais dokumentais apie kiekvieno produkto instaliavimą ir naudojimą. Tai išsamūs dokumentai, tačiau jų trūkumas – labai didelė apimtis ir dažniausiai naudojantis vienu ar kitu produktu šie dokumentai neperskaitomi iki galo.

Abiejų firmų greito apmokymo priemonės (angl. *tutorials*) yra gan neblogos ir padeda greitai įsisavinti produktus. MS SQL priemonės čia išsiskiria iliustracijų gausa, kurių ORACLE produktų mokymo priemonėse dažniausiai iš viso nebūna. Iliustracijos su ekranų vaizdais padeda žymiai greičiau atrasti vieną ar kitą meniu punktą bei įsitikinti, kad veiksmai atliekami teisingai.

Kaina

Remiantis Microsoft firmos skaičiavimais, galima pateikti ORACLE ir MSSQL duomenų bazių pramoninių versijų kainų lentelę (lentelė 6.9).

Procesorių skaičius	Oracle9i Enterprise versija	Oracle9i Enterprise versija su OLAP arba Data Mining	Oracle9i Enterprise versija su OLAP ir Data Mining	SQL Server 2000 Enterprise versija
1	40.000USD	60.000USD	80.000USD	20.000USD
2	80.000USD	120.000USD	160.000USD	40.000USD
4	160.000USD	240.000USD	320.000USD	80.000USD
8	320.000USD	480.000USD	640.000USD	160.000USD
16	640.000USD	960.000USD	1.280.000USD	320.000USD
32	1.280.000USD	1.920.000USD	2.560.000USD	640.000USD

6.9 Lentelė. Oracle ir MS SQL priemonių kainos

Atskirų MS SQL serverių kainų su OLAP ir duomenų gavybos priemonėmis (angl. *Data Mining*) nėra, nes šios priemonės įeina į standartinį priemonių paketą.

Bandomosios versijos ir programinių paketų pataisymai

Microsoft ir Oracle kompanijos laikosi skirtingos politikos pateikiant vartotojams bandomąsias paketų versijas. Microsoft SQL serverio bandomosios versijos neįmanoma parsisiųsti iš šios firmos svetainės. Bandomąsias versijas gauna užsiregistravę testuotojai, programinės įrangos platintojai ir pan. Jiems atsiunčiamos kopijos kompaktiniuose diskuose. Šios kopijos dažniausiai turi veikimo laiko limitą.

Oracle korporacija leidžią visą jų pagamintą programinę įrangą parsisiųsti iš Oracle interneto svetainės (http://otn.oracle.com/) ir naudoti ją vieno sistemos prototipo kūrimui. Svetainėje pateikiami veikiantys programiniai paketai, be jokių laikinių ar funkcinių apribojimų.

Skirtinga yra firmų politika ir programinių paketų atnaujinimams bei pataisymams. Iš Microsoft firmos svetainės galima parsisiųsti visus MS SQL serverio pataisymus, kai tik jie pasirodo. Tuo tarpu ORACLE leidžia pataisymus atsisiųsti tik vartotojams turintiems kliento identifikacijos numerį ir užsiregistravusiems svetainėje http://metalink.oracle.com.

Instaliavimas

Visuotinai sutinkama kad MS SQL serveris yra lengviau instaliuojamas negu ORACLE. Tai lemia keletas priežasčių:

- Veikia tik vienoje operacinių sistemų šeimoje (Windows)
- Operacinės sistemos ir duomenų bazės gamintojas yra tas pats
- Turi mažiau konfigūracijos alternatyvų

Instaliuojant OLAP priemones įeinančias tiek į MSSQL tiek į ORACLE sudėtį susiduriama su ta pačia problema: į išleisto programinio paketo sudėtį įeinančios priemonės neveikia. Tam, kad jomis būtu galima naudotis ir MSSQL ir ORACLE serveriuose reikia įdiegti papildomus pataisymus (angl. *patches*).

6.3 Priemonių palyginimo išvados

MSSQL privalumai ir trūkumai

Privalumai:

- Lengviau instaliuojamas.
- Mažesnė kaina už panašaus funkcionalumo paketą.
- OLAP priemonės gali būti prieinamos naudojant plačiausiai paplitusius biuro programinės įrangos rinkinius Microsoft Office 2000 ir Microsfot Office XP.
- Mažesnės dokumentacijos apimtys.

Trūkumai:

- Veikia tik Windows operacinėje sistemoje.
- Ne itin patogios OLAP priemonėmis sukurtų ataskaitų pateikimo Interneto svetainėje galimybės.
- Nėra bendro reliacinės duomenų bazės ir daugiamatės duomenų bazės serverių saugumo mechanizmo.

ORACLE privalumai ir trūkumai

Privalumai:

- Veikia beveik visose plačiau paplitusiose operacinėse sistemose.
- Lankstesnė architektūra. Didesnis instaliavimo ir konfigūracijos variantų pasirinkimas, daugiau procesų galima paskirstyti keliems kompiuteriams.
- Patogios OLAP priemonėmis suformuotų ataskaitų pateikimo Interneto svetainėje priemonės. Ataskaitas galima peržiūrėti bet kokia naršykle.

- OLAP serverio integravimas į pagrindinį duomenų bazės serverį užtikrina didesnį saugumą ir sistemos stabilumą.

Trūkumai:

- Sudėtingiau instaliuoti ir konfiguruoti.
- Didelė kaina.
- Nėra integracijos į biuro programų paketus priemonių.

7 Išvados

1. Darbe tirtos duomenų saugyklos ir analitinio duomenų apdorojimo priemonės. Jos pritaikytos analitiniam Interneto svetainės lankomumo duomenų apdorojimui.
2. Išnagrinėtas ne tik pats daugiamatis duomenų modelis, bet ir daugiamačių duomenų algebros operacijos. Kai kurios jų pritaikytos eksperimentinėje dalyje, užklausų formalizavimui.
3. Saugyklos kuriamoje sistemoje naudojamos ne kaip duomenų integravimo ir atvaizdavimo pasauliniame tinkle priemonė, bet kaip svetainėje kaupiamų duomenų analizės ir atvaizdavimo priemonė.
4. Pateiktas duomenų transformavimo proceso aprašymas. Šis procesas gali būti sėkmingai taikomas transformuojant svetainėje surinktus lankomumo duomenis tam, kad juos būtų galima apdoroti analitiškai.
5. Darbe palygintos dviejų Lietuvoje labiausiai paplitusių duomenų bazių valdymo sistemų: MS SQL serverio ir Oracle analitinio informacijos apdorojimo priemonės. Nesistengta atsakyti į klausimą, kuri iš šių sistemų yra geresnė ar blogesnė. Tai priklauso nuo sistemos kūrimo aplinkos ir nuo sistemai keliamų reikalavimų bei turimų lėšų.
6. Sukurti du analitinio svetainės lankomumo duomenų apdorojimo sistemos prototipai. Vienas naudojantis MS SQL, kitas Oracle priemonėmis. Abu šie priemonių rinkiniai yra tinkami tokių sistemų įgyvendinimui.
7. Pateiktos abiejų analitinių duomenų apdorojimo sistemų savybės, jų privalumų ir trūkumų nagrinėjimas turėtų padėtį OLAP sistemų kūrėjui apsispręsti, kurią iš nagrinėtų sistemų naudoti vienu ar kitu atveju.
8. Oracle priemonės yra labiau tinkamos kuriant analitinio informacijos apdorojimo sistemą su prieiga per Internetą. MS SQL priemonės geriau tinka sistemose, kurios kaip OLAP klientai naudojamos ofiso programos.
9. Bendra pateikta duomenų transformavimo metodika tinka abiejų sistemų naudojimo atveju. Tačiau MS SQL ir Oracle analitinio informacijos apdorojimo priemonių architektūra skiriasi, todėl skiriasi ir sukurtų prototipinių sistemų architektūra.
10. Pateikti bendras rekomendacijas, kurias priemones naudoti kuriant išbaigtą svetainės analitinio informacijos apdorojimo sistemą, yra sudėtinga. Tai priklauso nuo daugelio kriterijų.

11. Parašytas straipsnis tema „*Lyginamoji OLAP priemonių analizė ir taikymas interneto svetainėse*“. Jis bus pristatytas konferencijoje INFORMACINĖS TECHNOLOGIJOS‘2004.
12. Paruošta MS SQL OLAP priemonių savarankiško mokymosi medžiaga (angl. *tutorial*).

8 Literatūra

1. Inmon, W. H. Building the data warehouse - Boston : QED Technical Publishing Group, 1992 - 272 p

2. Ralph Kimball, Richard Merz TheData Warehouse Toolkit. Building the Web-Enabled Data Warehouse - New York: WILEY 2000. 401p.

3. Майкл Оутей, Поль Конте. Эффективная работа SQL Server 2000. Киев: BHV 2002.

4. Инни Чанг, Рене Ковингтон... Oracle 8. Энциклопедия пользователя. ДиаСщфт 1998.

5. Torben Bach Pedersen, Cristian S. Jensen. Multidimensional Database Technology. Straipsnis išspausdintas žurnale “Computer” 2001-12. Psl. 40.

6. Anindya Datta, Helen Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Straipsnis. Decision Support Systems 27 1999 Psl. 289–301

7. Xin Tan, David C. Yen., Xiang Fang. Web warehousing: Web technology meets data warehousing. Straipsnis Technology in Society 25 2003 Psl. 131–148

8. *Nigel Pendse.* What is OLAP? Straipsnis. *2002-07-27.* Žiūrėta 2002-11-27. Prieiga per: <http://www.olapreport.com>

9. *Nigel Pendse* Multidimensional data structures.. Straipsnis *2001-03-19*. Žiūrėta 2002-11-28. Prieiga per: http://www.olapreport.com

10. Microsoft corporation. DTS Overview. Straipsnis. Žiūrėta 2002-12-01. Prieiga per: <http://msdn.microsoft.com>

11. Microsoft corporation SQL Server 2000 Online documentation. 2001

12. Oracle corporation Oracle OLAP Technical documentation . Žiūrėta *2002-12-02*. Priega per: <http://otn.oracle.com>

13. Microsoft corporation. OLE DB. Online Analytical Data Processing overview. Žiūrėta 2003-01-17. Prieiga per: <http://msdn.microsoft.com>

14. Microsoft corporation. DTS Basics. Straipsnis. Žiūrėta *2003-12-10*. Prieiga per : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dtssql/dts_basic_71v7.asp >

15. Oracle corporation. Oracle9*i* Warehouse Builder User’s Guide Release 2 (9.2) 2003-07.

16. Oracle corporation. Oracle9*i* OLAP User’s Guide Release 2 (9.2) 2002-07.

17. Oracle corporation Oracle9*i* OLAP Developer's Guide to the OLAP DML 2002-07

18. Oracle corporation. Oracle9i Data Warehousing Guide. 2002-07

19. Алексей Федоров, Наталия Елманова Введение в OLAP: часть 3. Архитектура Microsoft Analysis Services., Straipsnis КомпьютерПресс 6'2001. Žiūrėta *2003-10-20* Priega per: <http://www.olap.ru/>

20. Marius Vilimas. Lyginamoji OLAP priemonių analizė ir taikymas interneto svetainėse. Straipsnis 2004.

21. Marius Vilimas. MS SQL OLAP priemonių savarankiško mokymosi medžiaga. 2004.

Summary

Data analysis tools research and usage in Internet systems

Every day Internet sites collect and store lots of user data. For effective usage of that data, special tools are required.

Online Analytical Processing (OLAP) tools are suggested for that purpose. Usage of these tools is related to usage of Data Warehousing tools. Therefore issues of Data Warehouse design, data transformation and transferring to data warehouses were discussed.

In the present years there are new trends in business computer systems industry – WEB Data Warehouses. New characteristics of such systems were analyzed.

OLAP tools usage is not possible without multidimensional data model. Main entities and operations with these entities were reviewed and mathematical definitions given.

Data transformation process was proposed. This process flow shows how transformations can be used for transferring data from WEB site to multidimensional database.

OLAP tools of Microsoft SQL server and Oracle database server (the most popular database management systems in Lithuania) were analyzed in experimental part of work. Data transformation and reviewing in desktop applications, WEB systems and office applications tools were compared and recommendations given.

9 Priedai

Pranešimo autoriaus anketa

KTU kviečia Jus į konferenciją

INFORMACINĖS TECHNOLOGIJOS'2004

Iš konferencijų ciklo "LIETUVOS MOKSLAS IR PRAMONĖ"

Kauno technologijos universitetas
Informatikos fakultetas
2004 m. sausio 28-29 d.

PRANEŠIMO AUTORIŲ ANKETA

Vardas Pavardė, Mokslo vardas ir laipsnis, Organizacija, Pareigos, Telefonas, E-paštas.

1. *Nemuraitė Lina, doc. t.m.dr., KTU, Informacijos sistemų katedra, docentė, 300397, nemur@soften.ktu.lt*
2. *Vilimas Marius, - , KTU, Informacijos sistemų katedra, magistrantas, 300397, marius@axella.no*

Pageidautume dalyvauti konferencijoje (*sekcijoje* ***Duomenų bazės ir modeliai***) ir skaityti pranešimą tema: *Lyginamoji OLAP priemonių analizė ir taikymas interneto svetainėse*

Pranešimo anotacija: Straipsnyje palyginamos duomenų saugyklų kūrimo ir OLAP priemonės komercinėse DBVS Oracle ir MS SQL serveryje. Pateikiama metodika šioms priemonėms taikyti interneto svetainių duomenų analizei bei eksperimentinio tyrimo rezultatai

10 Išnašos

[1]. Anindya Datta, Helen Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Straipsnis. Decision Support Systems 27 1999 Psl. 291

[2] Torben Bach Pedersen, Cristian S. Jensen. Multidimensional Database Technology. Straipsnis išspausdintas žurnale “Computer” 2001-12. Psl. 40.

[3] Ten pat.

[4] Ten pat.

[5] Ten pat.

[6] Алексей Федоров, Наталия Елманова. Введение в OLAP: часть 3. Архитектура Microsoft Analysis Services.Straipsnis. КомпьютерПресс 6'2001

[7] Ten pat.

[8] . Xin Tan, David C. Yen., Xiang Fang. Web warehousing: Web technology meets data warehousing. Straipsnis. Technology in Society 25 2003 Psl. 132

[9] Anindya Datta, Helen Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Straipsnis. Decision Support Systems 27 1999 Psl. 289–301

[10]. Microsoft corporation. DTS Basics. Straipsnis .

Prieiga per: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dtssql/dts_basic_71v7.asp >

[11] . Алексей Федоров, Наталия Елманова. Введение в OLAP: часть 3. Архитектура Microsoft Analysis Services.

[12] Ten pat.

[13] .Oracle corporation. Oracle9*i* Warehouse Builder User’s Guide Release 2 (9.2)

[14]. Oracle corporation. Oracle9*i* OLAP User’s Guide Release 2 (9.2)