LÄHESTYMISTAPOJA OLAP -KIELIIN

Kaarlo Kanerva

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Pro gradu -tutkielma
Lokakuu 2003

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tekijän Nimi: Kaarlo Kanerva
Pro gradu -tutkielma, 88 sivua
Lokakuu 2003

OLAP –sovelluksissa operatiivisista järjestelmistä tuotettava tieto muokataan moniulotteisen loogisen mallin mukaiseksi tietorakenteeksi tietovarastoon. OLAP –käsittelyssä kyselyitä tehdään tietovaraston sisältämiin yhteenvetotietoihin. Siinä hierarkkisesti järjestetyistä dimensioista ja faktoista koostuva tietokuutio antaa mahdollisuuden saada lukuisia erilaisia tarkastelukulmia eli näkymiä organisaation tietoihin.

Tässä tutkielmassa tarkastellaan erilaisia moniulotteisia tietorakenteita ja niiden ominaisuuksia, jotka takaavat virheettömät kyselyjen tulokset. Tutkielmassa tarkastellaan myös mahdollisuutta käyttää universaalirelaatiota ja normalisoimatonta kuutiota moniulotteisena loogisena mallina. Tutkielmassa näytetään, että SQL –kielellä tehdyt kyselyt muodostuvat samankaltaisiksi sekä universaalirelaatio- että lumihiutalemallissa. Lisäksi normalisoimaton kuutiomalli osoittautui kyselyjen muodostamisen kannalta parhaimmaksi kuitenkin tietojen ylläpidon ja rakenteellisten muutosten kustannuksella.

Tutkielmassa tuodaan esille laajalti erilaisia SQL –kieleen kehitettyjä OLAP –laajennuksia. Nämä osoittavat, että SQL –kieli on OLAP –käsittelyyn hyvin soveltuva kieli. Analyyttiset laskennat antavat mahdollisuuden tehdä monimutkaisia kyselyjä.

Lisäksi havaitaan, että Prologin pohjalta tehty OLAP -kieli soveltuu OLAP –käyttöön mm. rivi- ja sarakeyhteissummien tulostamisessa huomattavasti joustavammin kuin SQL.

Avainsanat ja -sanonnat: Moniulotteinen looginen malli, tietokuutio, OLAP, ROLAP, MOLAP, SQL.

SISÄLLYS

1. JOHDANTO

Moniulotteisen tiedon analysoinnin (On –Line Analytical Processing, OLAP) tavoitteena on tukea päätöksentekoa. Moniulotteisesti organisoitua tietoa analysoitaessa tietoja yhdistellään ja summataan (aggregoidaan) useiden ulottuvuuksien suhteen. Ulottuvuudella ymmärretään tekijää, jonka erilaisten arvojen perusteella koottua yhteenvetotietoa halutaan tarkastella. Perinteiset tietokannan hallintajärjestelmät on suunniteltu ensisijaisesti tapahtumakäsittelyyn (On –Line Transaction Processing, OLTP) perustuviin tietojärjestelmiin, joilla hoidetaan organisaation operatiivisia tehtäviä. OLTP –tietojärjestelmissä tietoja käsitellään hyvin yksityiskohtaisella tasolla, ja kyselyissä käsiteltävät tietomäärät ovat vähäisiä. Moniulotteisessa tietojen analysoinnissa käsitellään operatiivisista järjestelmistä tuotettua ennalta yhdistettyä ja summattua tietoa. Organisaation tilasta ja toimintaympäristön muutoksista saadaan uutta arvokasta tietoa yhdistelemällä ja analysoimalla operatiivisten järjestelmien historiatietoja.

Tapahtumakäsittelyyn perustuvien järjestelmien tietokaavioiden ja tietokantojen rakenne on toteutettu optimoiden operatiivisten järjestelmien tietokantaoperaatioiden suorituskykyä. Näitä järjestelmiä ei ole suunniteltu palvelemaan moniulotteista tietojenkäsittelyä. OLAP –kyselyjen kohdistaminen operatiivisten tietokantojen perustietoihin voi kestää tunteja tai päiviä ja jotkut kyselyt ovat jopa mahdottomia toteuttaa. Moniulotteinen tietojenkäsittely perustuu nopeaan tietojen saantiin, usein suurten tietomäärien hakuun, käsittelyyn ja yhdistelyyn sekä tietojen tarkasteluun monien eri ulottuvuuksien hierarkiatasojen suhteen ja joustavien, käyttäjäystävällisten näkymien tuottamiseen. OLAP –käsittely edellyttää, että tiedot on organisoitu rakenteeltaan moniulotteiseksi. Tiedon moniulotteinen rakenne esitetään kaaviotasolla loogisena mallina, jonka mukaisesti organisoituihin tietoihin OLAP –kyselyt tehdään. Moniulotteinen tietojenkäsittely asettaa kyselykielelle uusia toisenlaisia toiminnallisia vaatimuksia kuin OLTP:ssa käytetyltä SQL –kieleltä edellytetään. OLAP –käsittelyssä käytettävän kyselykielen tulee olla helppokäyttöinen ja ilmaisuvoimainen.

Tietovarastointi (data warehousing) ja OLAP liittyvät lähes erottamattomasti toisiinsa. Tietojen varastoinnin tarkoituksena on siirtää ja muokata analysoinnissa tarvittavat tiedot operatiivisista järjestelmistä tietovarastoon, jossa ne on organisoitu rakenteeltaan moniulotteiseksi. Siirtoprosessi vastaa siitä, että tietovarastoon siirretyt tiedot ovat oikeita, tarkkoja, ajankohtaisia ja luotettavia. Tietovarastossa tiedot ovat usein pidemmältä aikajaksolta kuin

operatiivisissa järjestelmissä säilytettävät tiedot. Tämä antaa mahdollisuuden tarkastella tiedoissa tapahtuneita muutoksia pitkällä aikajänteellä esim. trendien havaitsemiseksi.

Moniulotteisessa tietojen analysoinnissa käyttäjä tekee kyselyjä tietyltä kohdealueelta kerättyihin yhdistelmätietoihin. Nämä tiedot ovat numeerisia mitta-arvoja (measure), joihin tehdään laskentaoperaatioita. Mitta-arvoja tarkastellaan erilaisten ulottuvuuksien eli dimensioiden perusteella. Dimensiot ovat rakenteeltaan hierarkkisia. Esim. aikadimensiosta voidaan muodostaa hierarkkinen rakenne: päivä, kuukausi ja vuosi. OLAP –käsittelyn keskeisimpiä ominaisuuksia on mahdollisuus tarkastella mitta-arvojen yhteenvetotietoja dimensioiden eri tasoilla. Käyttäjä valitsee yhden tai useamman dimension, jonka suhteen hän tarkastelee mitta-arvoja. Dimensioita vaihtelemalla käyttäjä voi analysoida tietoja moniulotteisesti lukuisista erilaisista näkökulmista. Tietovarastointi tarjoaa käyttäjälle kohdealueesta joitakin perusnäkymiä, joiden sisältö määräytyy loogisen mallin kaaviotasoon valittujen dimensioiden ja mitta-arvojen perusteella. Moniulotteisella käsittelyllä on yhteyksiä tilastollisiin tietokantoihin ja tilastotieteeseen [Shoshani, 1997]. Näissä dimensioita vastaavat riippumattomat muuttujat ja mitta-arvoja riippuvat muuttujat.

OLAP –käsittely yhdistetään usein pelkästään liiketoiminnassa tapahtuvaan tietojen analysointiin. Tavanomaisimpia ovat mm. markkinoinnin, myynnin, asiakastietojen ja taloustietojen analysointitehtävät (ks. esim. Thomsen [1997]). OLAP -käsittelyä voidaan soveltaa lukuisilla muilla aloilla. McCabe *et al.* [2000] esittävät menetelmän, jossa dokumenttien hierarkkista rakennetta voidaan käyttää perinteiseen tekstitiedon hakuun. Stefanovic [1993] soveltaa OLAP käsittelyä alueellisiin tietoihin ja tietokantoihin. Pedersen ja Jensen [1999] käyttävät moniulotteista analysointia potilas- ja diagnoositietojen yhteydessä kliinisessä ympäristössä ja Huyn [2001] soveltaa sitä biotekniikan alalla. Niemi T. *et al.* [2003] ovat soveltaneet sitä informetriikan tietojen analysointiin.

Tämän tutkielman tavoitteena on kartoittaa ja tuoda esille erilaisia SQL –kieleen kehitettyjä ja toteutettuja OLAP –laajennuksia. Tutkimuksessa esitellään kompleksisten kyselyjen tekemiseen soveltuvia SQL –laajennuksia. SQL ei välttämättä ole ainoa OLAP –käsittelyyn soveltuva kieli, joten tutkielmassa tarkastellaan lyhyesti muitakin kyselykielilähestymistapoja. SQL –kielen helppokäyttöisyyteen, ilmaisukykyyn ja kyselyn muotoiluun vaikuttaa oleellisesti myös kyselyn taustalla oleva moniulotteinen looginen malli.

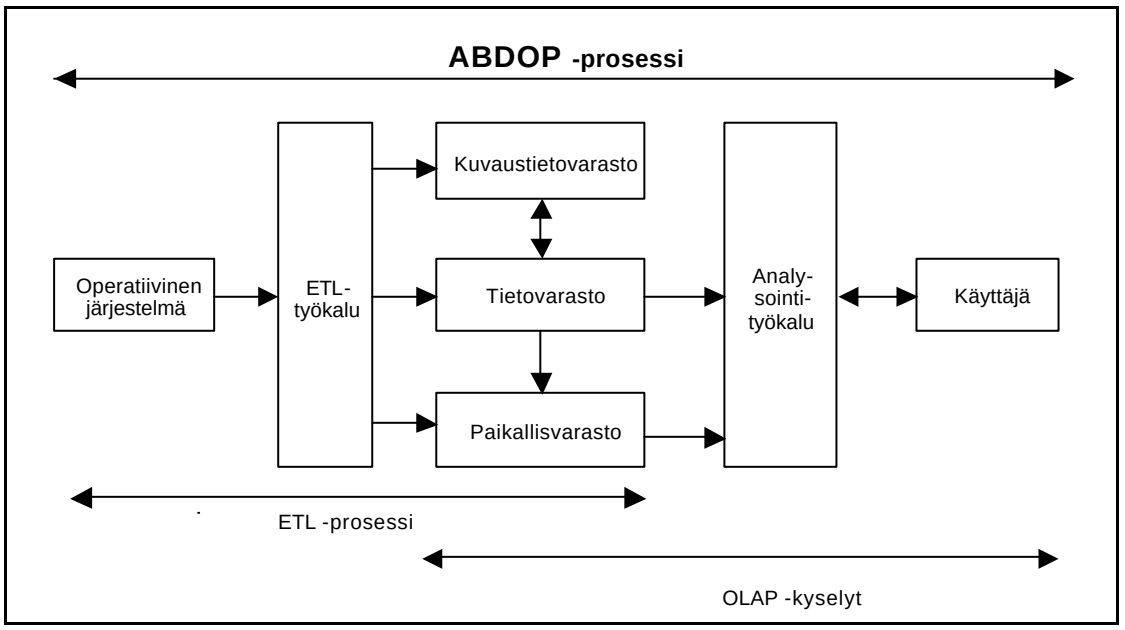
Tutkimuksessa tarkastellaan moniulotteisten tietorakenteiden mukaisia erilaisia loogisia malleja ja näiden vaikutusta kyselyjen muotoiluun. Tutkielman tavoitteena on löytää sellaiset loogiset mallit, joihin perustuen kyselyt voidaan muotoilla helposti, selkeästi ja yksinkertaisesti.

Tutkielma organisoidaan seuraavasti. Luvussa kaksi käsitellään OLAP –ympäristöä lähtien operatiivisista järjestelmistä ja päätyen analysointityökaluihin. Siinä myös kuvataan moniulotteisessa tietojenkäsittelyssä käytettyjä käsitteitä ja termejä. Luvussa kolme esitellään edellytykset laadukkaalle ja tehokkaalle OLAP –järjestelmälle. Neljäs luku sisältää kuvauksen keskeisistä OLAP –operaatioista. Viidennessä luvussa tarkastellaan sekä laskentaoperaatioita (aggregointifunktioita), joilla mitta-arvoista tehdään yhteenvetotietoja, että laskentaoperaatioiden summautuvuuden edellytyksiä. Lisäksi tarkastellaan dimensioiden hierarkiaan liittyviä rajoituksia. Kuudennessa luvussa tutkitaan erilaisia moniulotteisen tiedon loogisia malleja ja niiden vaikutusta kyselykieleen. Luvussa seitsemän esitetään SQL –kielen laajennuksia, jotka kykenevät moniulotteiseen tietojen analysointiin. Luvussa kahdeksan esitellään muita OLAP –käsittelyyn kehitettyjä kyselykielilähestymistapoja. Luku yhdeksän sisältää yhteenvedon tutkimuksen tuloksista.

2. OLAP -YMPÄRISTÖ

Tässä luvussa annetaan yleiskatsaus OLAP –ympäristöstä ja siihen liittyvästä käsitteistöstä sekä toiminnasta. Ensin kuvataan tietojen muokkaamista ja siirtämistä operatiivisista järjestelmistä tietovarastoon. Kuvauksessa esitetään tietovaraston keskeinen rooli OLAP –toiminnassa. Tämän jälkeen tarkastellaan moniulotteista tietorakennetta, josta käytetään nimeä tietokuutio tai kuutio [OLAP Council]. Kuution ominaisuuksia käsitellään sekä kaavio- että ilmentymätasolla. Luvun lopussa esitellään analysointityökalujen toimintaa ja OLAP –kyselykielen osuutta niissä.

OLAP on ohjelmistoteknologia, joka antaa analysoijille, johtajille ja muille käyttäjille mahdollisuuden saada kokonaisuuksiin liittyvää tietoa monin erilaisin näkymin nopealla, yhdenmukaisella ja interaktiivisella tavalla raakatiedoista tuotetusta informaatiosta, joka sisältää liiketoiminnan niitä ulottuvuuksia, joita käyttäjät haluavat tarkastella [OLAP Council]. Thomsen [1997, s. 12] kuvaa analysointiin perustuvaa päätöksentekoa tukevaa tietojenkäsittelyä ABDOP -prosessina (Analysis-Based, Decision-Oriented Processing). Kuva 1 esittää OLAP -ympäristön yleisellä tasolla.



Kuva 1: OLAP –ympäristö.

OLAP –ympäristö perustuu kiinteästi tietovarastoarkkitehtuuriin, jossa voidaan tunnistaa seuraavia komponentteja. ETL -prosessissa operatiivisen järjestelmän tiedot transformoidaan ETL -työkaluilla (Extraction,

Transformation and Loading) ja siirretään tietovarastoon [Dinter *et al.*, 1998]. Samassa yhteydessä muodostetaan tai ylläpidetään tietovaraston tietoja kuvaavaa erillistä kuvaustietovarastoa (metadatavarasto).

Kaikki analysoinnissa olennaiset tiedot pyritään keskittämään tietovarastoon. Tietovarastosta voidaan organisoida erillisiä tietovaraston osa-alueita, joita kutsutaan paikallisvarastoiksi eli data marteiksi. Tietovaraston rakentaminen on aikaa vievä prosessi. Jotta kohdealueen tietoja päästään hyödyntämään nopeasti perustuen tiettyyn tietojoukkoon voidaan operatiivisen järjestelmän tiedoista tuottaa data martteja. Periaatteessa tietovarastoon voidaan siirtää tietoa muistakin tietolähteistä kuin toimintayksikön operatiivisista järjestelmistä. Tavallisesti tietolähteet ovat tietokantoja, joista siirron aikana muokataan moniulotteisen rakenteen mukaisia tietovarastoja.

Kuvaustietovarasto sisältää kuvauksen tietovaraston sisältämistä tiedoista. Liiketoimintaa kuvaavaa metadataa ovat käytettävät termit, niiden määrittely. Operatiivinen metadata sisältää mm. tietoa tiedon alkuperästä, tietovarastoon siirron aikana tietoon tehdyistä muunnoksista, tiedon ajankohtaisuudesta, tietojen arkistoinnista sekä tietovarastosta poistetuista tiedoista. Metadatan avulla tietovaraston sisältämää tietoa voidaan jäljittää tiedon alkulähteelle. Metadataa käyttämällä hallitaan tietovaraston tietoja ja sen avulla myös käyttäjät jäsentävät tietovaraston sisältöä [Chaudhuri *et al.*, 2001].

2.1. ETL- prosessi

Ennen tietojen siirtämistä tietovarastoon on analysoitava ja päätettävä, mitkä operatiivisten järjestelmien tiedot kuvaavat yrityksen keskeistä toimintaa. Lisäksi on päätettävä tietovarastoon siirrettävän tiedon yksityiskohtaisuuden taso. Toiminnan kannalta avainasemassa olevia ja suorituskykyä kuvaavia indikaattoreita (Key Performance Indicators, KPI) voidaan eri menetelmin tunnistaa operatiivisista järjestelmistä (ks. esim. [Shouten, 1999] ja [Last ja Maimon, 2000]).

ETL –prosessi sisältää tietojen siirron operatiivisista järjestelmistä tietovarastoon sekä tietojen muokkauksen. Siirron aikana tietoja yhdistellään, yhtenäistetään, yhdenmukaistetaan ja puhdistetaan. Tietojen väliset ristiriitaisuudet ja havaitut virheellisyydet korjataan. Tietojen transformoinnissa käytetään sääntöjä ja ajojonoja, joiden mukaan tiedot muunnetaan ennalta määrätyn kaaviotason mukaiseen rakenteeseen. Eheyssääntöjen perusteella tarkistetaan ilmentymätason tietojen eheys. ETL -

prosessiin on kehitetty ohjelmistotyökaluja, joilla luodaan tietovaraston kaavio- ja ilmentymätaso.

2.2. Tietovarasto/tietovarastointi

Tietovarasto on kokoelma kohdealueorientoitua, integroitua, harvoin päivitettävää ja aikaan sidottua tietoa, jota käytetään johdon päätöksenteon tukena [Inmon, 1996, s. 33]. Tietovarasto sisältää useista operatiivisista järjestelmistä tuotettua ja yhdistettyä tietoa. Sillä on taipumus kasvaa kooltaan moninkertaiseksi operatiivisiin tietokantoihin verrattuna. Tietovaraston koko voi olla useita giga- tai teratavuja [Chaudhuri *et al.*, 2001]. Tietovarastoa ylläpidetään tyypillisesti erillään operatiivisista tietokannoista, koska OLAP – käsittely poikkeaa täysin operatiivisten tietokantojen toiminnasta. Vastaukset tietotarpeisiin ja analysointiin yhdestä yhteisestä moniulotteisesti organisoidusta tietolähteestä ovat useimmiten saatavissa huomattavasti nopeammin kuin operatiivisista järjestelmistä. Erillinen tietovarasto vähentää lisäksi operatiivisten järjestelmien kuormitusta erityisesti silloin, kun OLAP – kyselyitä tekee monta käyttäjää samanaikaisesti.

Tietojen pitäminen tietovarastossa erillään operatiivisista järjestelmistä aiheuttaa uudenlaisia vaatimuksia tietorakenteiden organisoinnille. Kimball [200b] on esittänyt 20 kriteeriään, jotka käyttäjäystävällisen tietovaraston pitäisi täyttää. Kaaviotason loogiset mallit ja ilmentymätason tiedot määräävät, millaisia näkymiä tietovaraston moniulotteisesta tietorakenteesta voidaan johtaa. Loogisen mallin suunnittelun aikana valitaan ne dimensiot, joiden suhteen mitta-arvoja halutaan tarkastella. Dimensiohierarkia antaa mahdollisuuden kumuloida mitta-arvoja ja tuottaa niistä yhteenvetotietoja eri dimensiotasoilla. Eri sovellus- tai osa-alueiden tietoja voidaan yhdistellä OLAP –analysoinnissa yhteisten ulottuvuuksien perusteella. Tietojen tulee olla ajantasaista, oikea-aikaista, luotettavaa ja oikein perusjärjestelmistä johdettua, jotta tietoihin perustuvat analysoinnit voivat tukea päätöksentekoa. OLAP -järjestelmän käytettävyys heikentyy, jos käyttäjät eivät voi luottaa vastauksena saamiensa tietojen oikeellisuuteen. Tietovaraston tietojen täytyy mukautua ja muuttua lähtöjärjestelmiensä muutoksiin. Tietovaraston dynaamisuus edellyttää, että sen käytön aikana lähtöjärjestelmissä tapahtuneet tietovaraston tietorakenteisiin vaikuttavat muutokset saadaan toteutettua tietovarastossa. Widom [1995] on tarkastellut tietovaraston käyttöön ja ylläpitoon liittyviä ongelmatilanteita.

Tietovarastoa rakennetaan tyypillisesti asteittain tuomalla sinne tietoja operatiivisten järjestelmien uusilta osa-alueilta. Tietovaraston tietojen virkistämisellä (refreshing) tarkoitetaan tietojen päivittämistä muuttuneilla operatiivisista järjestelmistä hankituilla tiedoilla. Tietojen virkistämiseen liittyy kaksi tärkeää seikkaa. On ratkaistava milloin ja miten tietovaraston tiedot päivitetään. Päivitys tehdään tavallisesti päivittäin tai viikoittain ennalta määrätyn aikataulun mukaan. Tietoja voidaan päivittää eri aikoina ja eri tietolähteistä riippuen operatiivisista järjestelmistä. Tietojen päivitystapa voi olla lisäävä tai korvaava. Lisäävässä päivityksessä tietovarastoon siirretään uudet tiedot operatiivisista järjestelmistä. Tässä päivityksessä siirrettävien tietojen määrä on pienempi kuin korvaavassa päivityksessä. Toisaalta tietovaraston tietyn osa-alueen tietojen päivittäminen uusilla tiedoilla on vaativampi tehtävä kuin osa-alueen kaikkien tietojen korvaaminen operatiivisten järjestelmien tiedoilla. Lisäävää päivitystä voi olla vaikea hallita, koska päivitysajankohta täytyy olla operatiivisen järjestelmän kanssa oikea-aikainen. Korvaavassa päivityksessä tietyn tietovaraston osa-alueen kaikki tiedot korvataan uusilla tiedoilla [Chaudhuri *et al.*, 2001]. Korvaava päivitys soveltuu pienehköjen tietomäärien siirtoon operatiivisista järjestelmistä.

Tietovaraston looginen malli on riippumaton fyysisestä toteutuksesta. Fyysisen toteutuksen suunnittelulla tavoitellaan tehokasta tietorakenteiden toteutusta, jolla tietojen haku ja käsittely tietovarastosta pyritään saamaan mahdollisimman nopeaksi. Tähän voidaan vaikuttaa tiedostorakenteilla, tiedostojen indeksoinneilla ja tietokantavalinnoilla. Relaatiotietokannalla toteutettua OLAP järjestelmää sanotaan ROLAP toteutukseksi (Relational OLAP), jossa kyselykielenä käytetään SQL –tyyppistä kieltä. Käyttäjien kyselyt ovat usein ennalta ennustamattomia ja monimutkaisia. Täten kyselyiden toteuttaminen edellyttää monivaiheista SQL –toteutusta (multi-pass sql). Siksi tietokannan yhteyteen on tässä lähestymistavassa rakennettu erillinen ns. moniulotteinen palvelinmoottori (multidimensional server engine), joka transformoi käyttäjän kyselyn yhdeksi tai useammaksi SQL –lauseeksi, jotka suoritetaan perättäin tai samanaikaisesti tietokannassa [Pendse, 2000a]. Schwarz *et al.* [2000] ovat suunnitelleet ns. optimoidun systeemiarkkitehtuurin, jossa kyselygeneraattori muodostaa SQL –kyselyn. Testen [2000] mallissa kyselyn kääntäjä muodostaa moniulotteisesta kyselystä relaatiotietokannan mukaisen kyselyn. Vastaavasti saadut vastaukset muotoillaan samaisen moottorin toimesta sellaiseen muotoon, että tulokset voidaan näyttää käyttäjälle. Palvelinmoottorin tavoitteena on nopeuttaa kyselyn suorittamista.

MOLAP -järjestelmät (Multidimensional OLAP) käyttävät ROLAP:ista poikkeavaa erityistä moniulotteisen tiedon hallintamenettelyä, jota sanotaan moniulotteiseksi tietokannan hallintajärjestelmäksi (Multidimensional Database Management Systems, MDDBMS). Moniulotteisessa tietokannassa kaikki mahdolliset aggregoinnit on laskettu ennalta [Vassiliadis ja Sellis, 1999]. Tässä lähestymistavassa kyselykieli ei välttämättä ole SQL:n kaltainen. MOLAP –arkkitehtuuriin perustuvissa järjestelmissä tieto tallennetaan moniulotteisiin taulukoihin. Moniulotteisen tiedon ilmentymätason vaatiman tilan tiivistäminen tapahtuu linearisoimalla taulukot [Shoshani, 1997]. Tämän menettelyn lisäksi käytetään erilaisia tietojen tiivistämistekniikoita. Niiden tarkoituksena on poistaa nolla- ja null -arvot ilmentymätason tilasta. Molempien menettelyjen käyttö johtaa siihen, että MOLAP –järjestelmissä tietojen fyysinen tallennusrakenne on useimmiten kaksitasoinen [Dinter *et al.*, 1999]. Tiheät paljon dataa sisältävät dimensiotaulukot indeksoidaan rakenteeksi, joka sisältää harvojen taulukoiden yhdistelmät. Tämän ylemmän tason muodostaman rakenteen osoitteet viittaavat alemman tason taulukoihin, jotka muodostetaan tiheistä dimensioista. Alemman tason taulukot sisältävät tiheät dimensiot mitta-arvoineen.

Hasan *et al.* [2000] vertailevat MOLAP - ja ROLAP -järjestelmien eroavuuksia. Heidän mukaan näiden kahden lähestymistavan keskeisimmät ero ovat tallennettujen tietojen prosessointiominaisuuksissa ja tietojen ajantasaisuudessa. MOLAP –järjestelmässä saavutetaan optimaalinen tietojen käsittelynopeus ja joustavuus, koska mitta-arvojen yhteenvetotietoja ei tarvitse laskea OLAP –kyselyjen aikana. MOLAP -järjestelmissä kyselyt suoritetaan nopeammin kuin relaatiotietokantoihin perustuvissa järjestelmissä [Colliat, 1996]. MOLAP –järjestelmille on tyypillistä, että harvemmin tietoja päivitetään, ja useimmiten tietoja ainoastaan luetaan. ROLAP –järjestelmissä voidaan usein porautua (drill-down) ainakin teoriassa tiedon yksityiskohtaisimmalle tasolle, jossa nopea prosessointi vaatii kalliita ja huipputehokkaita laitteistoja. ROLAP -järjestelmän etuna on se, että sillä on kiinteä yhteys relaatiotietokantoihin liittyviin standardeihin. Tästä on esimerkkinä mahdollisuus SQL –tyyppiseen kyselyn formulointiin. MOLAP –arkkitehtuuria on yleisesti arvosteltu siitä, että moniulotteisiin tietokantoihin perustuvista tuotteista puuttuvat standardit.

Näiden kahden yhdistelmällä HOLAP:illa (Hybrid OLAP) pyritään yhdistämään sekä ROLAP - että MOLAP -järjestelmien vahvuuksia. HOLAP –järjestelmä tunnistaa moniulotteisessa tilassa harvat (sparse) ja tiheät (dense)

alueet. Näiden perusteella tietojen käsittely hoidetaan harvoilla alueilla ROLAP:illa ja tiheillä alueilla MOLAP:illa [Chaudhuri *et al.*, 2001].

2.3. Paikallisvarasto

Tietovarasto toteutetaan eri tavalla kuin OLTP –sovelluksissa. Tietoja tuodaan tietovarastoon jatkuvasti uusilta aihealueilta, ja siksi varaston saattaminen lopulliseen tietosisällön laajuuteensa on pitkäaikainen prosessi. Käyttäjät eivät ole kiinnostuneita kaikista tietovaraston tiedoista, vaan tietyn rajatun alueen tiedoista. Tällaisia alueita voivat olla esim. markkinointi ja myynti. Operatiivisista järjestelmistä tai tietovarastosta johdettua yhteen aihealueeseen tai suppeaan joukkoon aihealueita rajautuvaa tietovaraston osaa sanotaan paikallisvarastoksi (data mart) [SYSTA / TIHA, 1997] [Hovi, 1997, s. 36].

OLAP -kysely voi kohdistua tietovarastoon tai paikallisvarastoon. Paikallisvarastossa tiedot voivat olla organisoidut karkeammalla tasolla kuin tietovarastossa, jos tietojen hyväksikäyttäjät eivät tarvitse tietovarastoon tallennettujen tietojen yksityiskohtaisuuden tasoa.

2.4. Kuutio

OLAP -käsittelyssä käytetään yleisesti metaforana käsitettä kuutio, joka kuvaa moniulotteista tietorakennetta havainnollisella ja ymmärrettävällä tavalla. Moniulotteinen tietorakenne voi olla kaksi- tai useampiulotteinen. Kaksiulotteisessa visualisoinnissa tiedot esitetään sarakkeina ja riveinä. Kun ulottuvuuksia on useampi kuin kolme puhutaan usein myös hyperkuutiosta. Kuutio sisältää ulottuvuuksia eli dimensioita ja näihin liittyviä mitta-arvoja. Kuution akselit muodostuvat dimensioista. Jokainen alkio kuutiossa on joidenkin dimensioiden tai dimensioattribuuttien arvojen yhdistelmään liittyvä mitta-arvo (ks. kuva 2). Dimensioiden hierarkkinen rakenne antaa mahdollisuuden koostaa eli aggregoida mitta-arvoja alimmalta hierarkiatasolta ylemmille tasoille tai vastaavasti porautua karkeammilta tasoilta alemmille yksityiskohtaisemmille tasoille. Aggregointi tapahtuu dimensioiden eri tasojen yhdistelmillä.

Formaalisti esitettynä OLAP kuutio on relaatio, joka on osajoukko karteesisesta tulosta $C \subseteq \overline{D}_1 X \overline{D}_2 X \ldots X \overline{D}_j X \overline{M}_1 X \overline{M}_2 X \ldots X \overline{M}_k$, jossa $\overline{D}_m$ $(1 = m = j)$ on dimensioattribuutin D_m arvojoukko ja $\overline{M}_n$ $(1 = n = k)$ on mitta-attribuutin M_n arvojoukko. Kuva 2 esittää tietovarastoon tallennetun peruskuution taulukkona, josta voidaan johtaa uusia kuutiota.

Dimensioattribuutit								Mitta-attribuutit	
Dimensio 1			*Dimensio 2*			*Dimensio 3*			
Aika			*Alue*			*Tuote*			
Vuosi	**Kuukausi**	**Päivä**	**Maanosa**	**Valtio**	**Kaupunki**	**Ryhmä**	**Tyyppi**	**Määrä**	**Hinta**
2001	2001-01	2001-01-10	Eurooppa	Ruotsi	Tukholma	B	Zafira	3	28 500
2001	2001-01	2001-01-15	Eurooppa	Ruotsi	Tukholma	A	Astra	1	32 100
2001	2001-01	2001-01-16	Eurooppa	Suomi	Helsinki	A	Vectra	2	19 900
2001	2001-01	2001-01-25	Eurooppa	Suomi	Helsinki	A	Vectra	1	28 500
2001	2001-02	2001-02-15	Eurooppa	Suomi	Helsinki	A	Vectra	1	25 600
2001	2001-02	2001-02-10	Pohj.Amerikka	USA	New York	A	Astra	5	22 000
2001	2001-02	2001-02-15	Pohj.Amerikka	USA	Chicago	A	Vectra	2	23 000
2001	2001-02	2001-02-27	Pohj.Amerikka	USA	Dallas	A	Vectra	4	23 000
2002	2002-01	2002-01-17	Eurooppa	Ruotsi	Tukholma	B	Zafira	2	29 000
2002	2002-01	2002-01-18	Eurooppa	Suomi	Helsinki	A	Vectra	2	18 900

Kuva 2: OLAP -kuutio taulukkoesityksenä.

Kuvassa 2 aika, alue ja tuote ovat dimensioita. Aikadimension hierarkkinen rakenne on vuosi, kuukausi ja päivä, aluedimension maanosa, valtio ja kaupunki ja tuotedimension ryhmä ja tyyppi. Mitta-attribuutteja ovat määrä ja hinta. Kuution kaaviotaso (intensio) muodostuu dimensioattribuuttien ja mitta-attribuuttien nimistä, ja sen ilmentymätaso (ekstensio) muodostuu dimensioattribuuttien ja mitta-attribuuttien arvoista.

Kuutio voi sisältää paljon mitta-arvojen yhteenvetotietoina alkioita, joista tieto puuttuu. Sovitusta esitystavasta riippuen tällaisten alkioiden arvo on nolla tai null. Tällaisten alkioiden määrä vaikuttaa kuution harvuuteen. OLAP –kuution harvuus ilmoitetaan suhdelukuna, joka on kuution tyhjien alkioiden määrä jaettuna kaikkien alkioiden määrällä. Esim. kaksiulotteisen taulun harvuus on 0, kun taulussa on kaksi saraketta, kolme riviä ja kaikissa sarakkeiden ja rivien leikkauspisteiden muodostamissa alkioissa mitta-arvo on muu kuin nolla tai null. Kuution harvuus on välillä 0–1. Lukua yksi lähestyttäessä kuution harvuus kasvaa. Lehner *et al.* [1998] ovat tutkineet kuution dimensioattribuuttien välisiä funktionaalisia riippuvuuksia ja niiden vaikutuksia loogisen mallin ja kuution suunnitteluun sekä kuution harvuuteen. He esittävät dimensionaalisen normaalimuodon (dimensional normal form, DNF) ja moniulotteisen dimensionaalisen normaalimuodon (multidimensional normal form, MNF). Nämä normaalimuodot edellyttävät, että dimension hierarkkisten tasojen välillä on funktionaalinen riippuvuus. Dimensioiden välistä funktionaalista riippuvuutta ei sallita, jotta DNF – ja MNF –ehto täyttyvät. Tämän lisäksi Niemi *et al.* [2001] ovat tarkentaneet dimensioattribuuttien funktionaalisten riippuvuuksien vaikutusta kuution suunnitteluun todeten, että kuutio on ei-harvassa normaalimuodossa (non-sparse normal form), jos kaikki riippuvuudet ovat dimension sisäisiä. Näiden

rajoitusten ja sääntöjen avulla kuution kaaviotason suunnittelussa pyritään minimoimaan ilmentymätason harvuus ja varmistamaan, että mitta-arvojen yhteenvetotiedot lasketaan oikein.

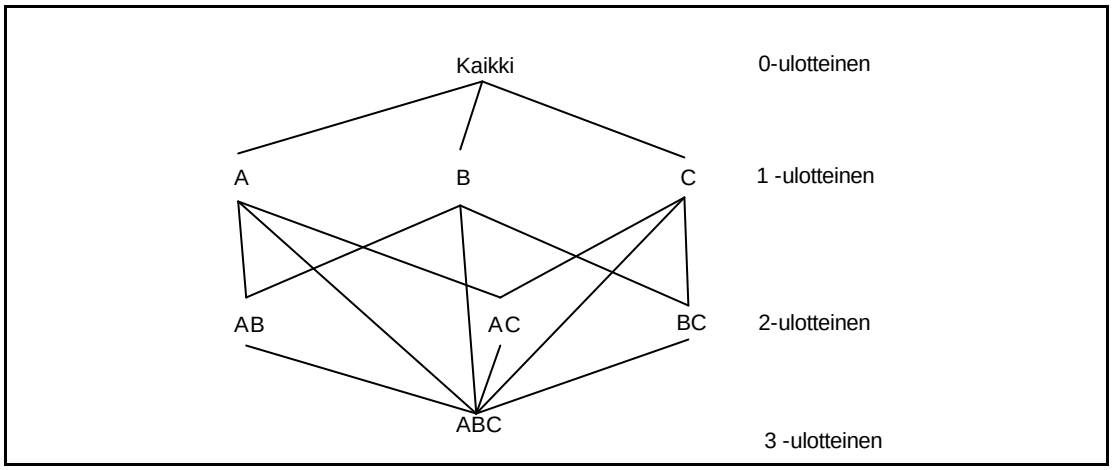
Kuution ilmentymätaso voi olla globaalinen, osittain esilaskettu, tai peruskuutio. Ilmentymätason globaalisessa kuutiossa kaikille mahdollisille dimensioattribuuttien arvojen yhdistelmille on ennalta laskettu mitta-arvojen yhteenvetotiedot. Tällaista kuutiota sanotaan myös täysin materialisoiduksi [Harinarayan *et al.*, 1995]. Globaaliseen kuutioon tehtyjen kyselyjen vastausajat ovat kahteen muuhun toteutustapaan verrattuna lyhyempiä, koska yhteenvetotietoja ei lasketa kyselyn aikana. Globaalinen kuutio ei välttämättä ole paras vaihtoehto kun käsitellään isoja tietomääriä, koska kuution mitta-arvojen yhteenvetotietojen ennalta laskeminen vaatii runsaasti aikaa, ja täysin ennalta laskettu kuutio vaatii paljon ulkoista muistitilaa.

Täydellisesti laskettu globaalinen kuutio saattaa muodostua moninkertaiseksi peruskuutioon verrattuna. Tästä syystä kuutio pyritään suunnittelemaan mahdollisimman tiheäksi niin, että mitta-arvojen yhteenvetotiedot lasketaan kaikilla dimensioiden yhdistelmillä oikein. OLAP –raportissa on määritelty kasvukerroin (compound growth factor, CGF) [Pendse, 2000b]. CGF ilmaisee kuinka paljon kuution koko kasvaa, kun kuutioon lisätään yksi dimensio. Tutkimusten mukaan yhden dimension lisäys kasvattaa kuution kokoa n. 1,5 – 2,5 -kertaiseksi.

Osittain esilasketussa, materialisoidussa, kuutiossa vain tietyt mitta-arvojen yhteenvetotiedot on ennalta laskettu [Harinarayan *et al.*, 1995]. Tällaisen kuution muodostamisessa täytyy tietää mitkä ovat ne keskeiset ennalta lasketut yhteenvetotiedot, joita kyselyissä voidaan hyödyntää siten, että niistä edelleen saadaan johdettua uusia kuutioita.

Kuvan 3 kolmiulotteinen kuutio C ={A, B, C} on esitetty hilana, jossa kuutio ABC on peruskuutio. Kuutiosta ABC voidaan esim. johtaa seitsemän eri kuutiota. Jos kaksiulotteinen taso (AB, AC ja BC) on esilaskettu, voidaan esim. AB:sta johtaa kuutio A tai kuutio B. Mitä enemmän mitta-arvojen yhteenvetotietoja on ennalta laskettu sitä lyhyempiä ovat kyselyjen vastausajat [Baralis *et al.*, 1997]. Suurissa tietomäärissä voidaan laskea vain pieni osa globaalin kuution alkioista ennalta ulkoisen muistitilan tarpeen vuoksi. On kehitetty myös menetelmiä, joiden avulla käyttäjän kyselyn perusteella

muodostettua kuutiota käytetään seuraavan kyselyn osakuutiona [Harinarayan *et al.*, 1995] [Agarwal *et al.*, 1996].



Kuva 3: Kuutio hilarakenteena.

Niemi *et al.* [2001] ovat tutkineet kyselyihin perustuvaa OLAP kuutioiden muodostamista. Käyttäjät tekevät yleensä sarjan toisiinsa liittyviä kyselyjä. Heidän tavoitteenaan on muodostaa annetusta kyselyjoukosta OLAP –kuutio tai kuutioita, jotka antavat vastaukset näihin kyselyihin. Heidän kehittämänsä algoritmin mukaan samantyyppiset kyselyt muodostavat oman kuutionsa, josta kyselyillä johdetaan uusia kuutioita.

Peruskuutiossa mitta-arvot on laskettu dimensioiden alimmille hierarkiatasoille. Muita mitta-arvojen yhteenvetotietoja ei ole laskettu ennalta. Mitta-arvojen vyöryttäminen ja laskenta ylempien dimensioattribuuttiarvojen tasoille yhteenvetotiedoiksi tehdään yleensä peruskuution dimensioiden alimman tason mitta-arvoista. Peruskuutioon tehtyjen kyselyjen vastinajat voivat muodostua pitkiksi. Tämän vaihtoehdon etuna on kuitenkin se, että se vaatii vain peruskuution tarvitseman tilan ulkoista muistia. Tietovarastossa tiedot ovat yleensä peruskuution muodossa, ja erilaiset moniulotteisen tietorakenteen loogiset mallit esittävät rakenteen peruskuutiona.

Keskeisiä kuution käsittelyyn liittyviä ongelmia ovat kuution tietojen ylläpito, kyselyjen prosessointiaika ja kyselykielen ilmaisuvoimaisuus. Näitä pyritään ratkaisemaan ja optimoimaan erilaisin menetelmin.

Kuution tietojen ylläpitoa vaikeuttaa kuution rakenteessa tapahtuva muutos. Mitta-arvot tietovarastossa ovat luonteeltaan dynaamisia ja dimensiot enemmänkin staattisia [Mendelzon ja Vaisman, 2000]. Operatiivisissa

järjestelmissä tapahtuvat muutokset voivat saada aikaan sen, että dimensioita on lisättävä tai poistettava kuutiosta. Dimensioiden sisäinen rakenne voi muuttua. Hierarkiatasoja on lisättävä tai yhdistettävä. Samalle dimensiolle on ehkä otettava käyttöön toinenkin hierarkia. Esimerkkinä mainittakoon kauppadimensio. Se voi muuttua, kun uusia kauppoja avataan tai vanhoja kauppoja suljetaan. Saattaa olla myös tarvetta ryhmitellä kauppoja aiemmasta poikkeavalla tavalla.

Kyselyjen vaatimaan prosessointiaikaan voidaan vaikuttaa suunnittelemalla optimaalinen kaaviotason looginen malli mm. ennakoimalla kyselytarpeita, ja valitsemalla ilmentymätasolle paras toteuttamistapa. Näihin valintoihin vaikuttaa myös kohdealueen tietojen väliset rakenteet ja kohdealueella tapahtuvat muutokset. Loogisessa mallissa taulujen määrä vaikuttaa siihen, kuinka paljon liitoksia kyselyissä joudutaan tekemään taulujen välillä. Jos taulujen välisiä liitoksia tehdään paljon, kyselyjen prosessointiaika kasvaa. Toisaalta looginen malli, jossa tietokannan taulut ovat normalisoimattomassa muodossa aiheuttaa kuution tietojen ylläpidossa vaikeuksia. Ilmentymätaso voidaan toteuttaa ROLAP - , MOLAP – tai HOLAP –ratkaisuna. ROLAP – vaihtoehdossa järjestelmä on tietosisällön kasvaessa helposti ja joustavasti laajennettavissa kyselyjen vaatiman prosessointiajan kustannuksella. MOLAP - ratkaisu on suorituskyvyltään edellistä tehokkaampi, mutta tietojen ylläpito on vaikeampaa kuin ROLAP:issa.

Kyselykielellä tulee voida ilmaista kaikki OLAP –käsittelyssä tarvittavat operaatiot. Kyselyt tulee myös voida formuloida käyttäen selkeää ja yksinkertaista syntaksia. OLAP –kyselyt voivat olla hyvinkin kompleksisia. Relaatioalgebraan perustuva SQL –kieli ei välttämättä ole paras OLAP – käsittelyyn sopiva kieli. Monesti SQL –kielellä esitettävä kysely joudutaan jakamaan useaksi erilliseksi lausekkeeksi (multi-phase sql), jotta kysely saadaan toteutettua. SQL -kyselyä prosessoitaessa samaa relaatiota voidaan joutua käymään läpi useaan kertaan. Näin tapahtuu, kun kyselyssä tehdään operaatioita saman relaation eri rivien kesken. Näistä syistä SQL –kieleen on kehitetty laajennuksia sen helppokäyttöisyyden ja ilmaisuvoimaisuuden lisäämiseksi. MDX –kyselykieli on kehitetty nimenomaan OLAP –käsittelyä varten.

2.5. Analysointityökalut

Tietojen tarkastelu pyritään tekemään analysointityökaluilla mahdollisimman joustavaksi ja helpoksi. Käyttäjän ei tarvitse tietää taustalla olevaa tietokannan

rakennetta. Analysointityökalu näyttää tarkasteltavan kuution tai kuutioiden dimensioattribuuttien ja mitta-attribuuttien arvot. Taustalla olevaan kuutioon kohdistuvat kyselyt tuottavat usein peruskuutiosta johdettuja uusia näkymiä. Mitta-arvojen yhteenvetotiedot näytetään sarakkeiden ja rivien leikkauspisteissä alkioina. Tavallisesti mitta-arvoja summataan tai ne esitetään esim. prosentuaalisina osuuksina ylemmästä dimensiotasosta. Mitta-arvojen yhteenvetotietoja laskettaessa käytetään muitakin funktioita kuin yhteenlaskua.

Käyttäjät voivat kysellä OLAP –analysointityökaluilla tietoja tietovarastosta tai paikallisvarastosta. Analysointityökalu yleensä näyttää käyttäjälle mitä dimensioita ja mitta-arvoja on käytettävissä. Analysoitavat tiedot voivat sisältää tietoja yhdestä tai useammasta kuutiosta. Yleensä dimensiot ilmaisevat aikaa, paikkaa ja jotain muuta ominaisuutta, jonka suhteen yhteenvetotietoja halutaan tarkastella. Tietojen analysoinnissa dimensiot antavat useimmiten vastauksen kysymyksiin mitä, missä, milloin, miten ja kuka. Aikadimensio on moniulotteisessa tietojen analysoinnissa välttämätön. Sillä voidaan tarkastella toiminnan kehitystä ja trendejä eri aikajaksoilta ja verrata näiden mitta-arvoja toisiinsa. Tiedoista muodostetut näkymät esitetään loppukäyttäjälle usein kaksiulotteisena tai sisennettynä taulukkorakenteena tai graafisina kaavioina. Kolmiulotteinen vaikutelma saadaan aikaan siten, että kolmannen dimensioattribuutin arvot näytetään näytöllä kukin eri sivulla. Näkymiä voidaan muunnella joustavasti esim. muuttamalla rivejä ja sarakkeita keskenään.

Tietojen analysointi ja tarkastelu alkaa yleensä karkealta tasolta muutamalla dimensiolla. Analysoinnissa pyritään saamaan selville mistä mitta-arvojen poikkeukselliset eroavuudet johtuvat. Kun tällainen eroavuus on havaittu, hierarkiatasoja vaihtelemalla poraudutaan (drill-down) yksityiskohtaisimmille tasoille tarkasteltavan dimension suhteen. Tietoja voidaan myös yhdistellä dimensiohierarkian sallimissa rajoissa ylemmille karkeammille tasoille (roll-up). Navigointi on siirtymistä näkymästä toiseen kuution rajoittamassa moniulotteisessa avaruudessa dimensioiden ja dimensiotasojen välillä. Kyselyjen tulokset ovat aina tämän avaruuden osakuutioita. OLAP –operaatiot näyttävät siirtymissuunnan kuution avaruudessa tai niillä rajoitetaan dimensiohierarkiatason arvojoukkoa. Analysointityökalun tehtävänä on muuntaa käyttäjän ilmaisemat toiminnot kyselyiksi, jotka kohdistetaan tarkasteltavaan kuutioon. Relaatiotietokantaan perustuvassa ratkaisussa analysointityökalu generoi käyttäjän ilmaisemien toimintojen perusteella SQL

–kyselyjä, jotka sitten toteutetaan kuutioon. Kyselyt tuottavat yhden tai useampia tulosrelaatioita, joista ohjelmisto muokkaa tiedot näytön edellyttämään muotoon.

3. OLAP –JÄRJESTELMÄLLE ESITETYT VAATIMUKSET

E. F. Codd Associates [1993] on esittänyt OLAP -käsittelyn perusvaatimukset. Seuraavien kriteerien perusteella voidaan arvioida, miten hyvin tietty OLAP –ohjelmistotuote soveltuu OLAP -käsittelyyn.

Moniulotteinen käsitteellinen näkymä. Tietoja analysoivan käyttäjän näkemys kohdealueen toiminnasta on moniulotteinen. Käyttäjä näkee OLAP tiedot moniulotteisena käsitteellisenä mallina, jossa tietoja voidaan joustavasti ja intuitiivisesti käsitellä mallin tarjoamien ulottuvuuksien suhteen.

Läpinäkyvyys. OLAP:in tulee olla käytettävissä avoimen systeemiarkkitehtuurin mukaisesti, ja sen tulee antaa käyttäjälleen mahdollisuus liittää analysointityökalu käyttäjän haluamaan ympäristöön. Lisäksi käyttäjän ei tarvitse tietää, saako analysointityökalu syötteensä homogeenisestä tai heterogeenisestä tietokantaympäristöstä.

Saatavuus. OLAP –työkalun tulee tarjota useasta heterogeenisestä fyysisestä tietolähteestä koottu yksi, yhdenmukainen ja yhtenäinen tietolähde, jolla on oma looginen kuvauksensa. Työkalu vastaa lähtötietojen hankkimisesta käyttäjälle. Käyttäjälle lähtötietojen tuottaminen on näkymätöntä.

Raporteissa yhtenäinen vastinaika. Ulottuvuuksien määrän tai kuution koon kasvu ei saa merkittävästi pidentää vastinaikoja raportoinnissa. OLAP -käyttäjälle raporttien yhtenäiset vastinajat ovat kriittisiä. Yhtenäisillä vastinajoilla OLAP –käsittely säilytetään helppokäyttöisenä kompleksisuutta välttäen.

Työasema/palvelinarkkitehtuuri. OLAP –tuotteiden on voitava toimia asiakas / palvelin ympäristössä siten, että palvelinosaan voidaan liittyä erilaisilta työasemilta joustavasti.

Ulottuvuuden yleisyys. Eri ulottuvuuksien tulee olla rakenteeltaan ja käsittelyominaisuuksiltaan yhdenmukaiset. Laskentafunktiot tulee toimia kaikilla ulottuvuuksilla.

Dynaaminen harvan kuution käsittely. OLAP –työkalun tulee käsitellä harvaa kuutiota optimaalisesti. Vastinaika tulee olla yhtenäinen ja melko vakio

kuution tietoalkioiden hakujärjestyksestä, ulottuvuuksien määrästä ja tiedostojen koosta riippumatta.

Monen käyttäjän tuki. OLAP –välineen tulee tarjota yhtäaikainen tietojen saatavuus, päivitys ja haku monelle käyttäjälle.

Ulottuvuuksien väliset operaatiot. Työkalun tulee tarjota kaikille ulottuvuuksille valmiita funktioita ja antaa käyttäjälle mahdollisuus määritellä niille omia funktioita.

Tiedon intuitiivinen käsittely. Käyttöliittymän käsittelytavat tulee olla käyttäjäystävällisiä, suoraviivaisia ja ne tulee toteuttaa niin, että tietoalkioita haettaessa käytetään nopeinta tapaa. Mm. DRILL-DOWN - tai ROLL-UP - operaatiot tulee voida suorittaa sekä riveittäin että sarakkeittain.

Joustava raportointi. Näyttöruudun riveillä, sarakkeilla ja sivuotsikoilla tulee voida esittää ulottuvuuksia nollasta N:ään, jossa N ilmaisee ulottuvuuksien kokonaismäärän.

Ulottuvuudet ja rajoittamaton summaustasojen määrä. Tutkimukset osoittavat, että kuution käsittelyssä saatetaan samanaikaisesti tarvita jopa 19 eri dimensiota. Siksi suositellaan, että kehittyneellä OLAP –työkalulla voidaan kuutiota käsitellessä käyttää vähintään 15 mieluiten 20 ulottuvuutta. Käyttäjän tulee voida määritellä rajoittamaton määrä hierarkiatasoja kullekin ulottuvuudelle.

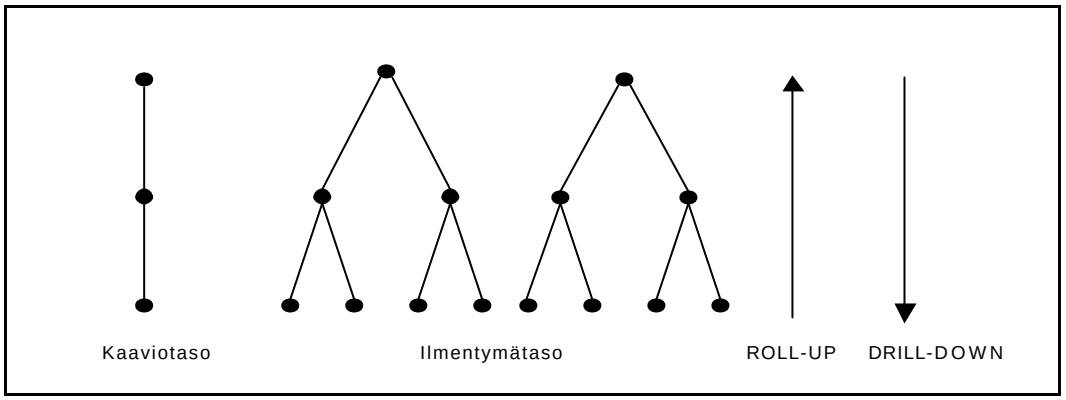
4. OLAP -OPERAATIOT

Luonteeltaan OLAP -operaatiot ovat enemmänkin kuvailevia kuin täsmällisiin formaaleihin määrittelyihin perustuvia, mistä johtuen samalla operaatiolla voi olla useampiakin merkityksiä. Tässä luvussa tarkastellaan tyypillisimpiä OLAP –operaatioita ja niiden SQL-kielisiä ilmauksia. Samalla tuodaan esille ne operaatioiden piirteet, jotka SQL-kielellä ovat vaikeasti tai jopa mahdottomia toteuttaa. Yleisimmät aggregointifunktiot SQL –kielessä ovat AVERAGE, COUNT, MAX, MIN ja SUM [Date ja Darwen, 1997]. SQL-99 standardi sisältää joukon muitakin funktioita edellisten tavanomaisimpien lisäksi [Winter, 2000a].

4.1. ROLL-UP

ROLL-UP –operaatio on tietojen tarkastelutasoa karkeistava jalostusmenettely. Siinä yhdistetään tietoja dimensiohierarkiassa karkeammalle tasolle yleensä seuraavaksi ylemmälle tasolle aggregoimalla mitta-attribuuttien arvot. Kuva 4 esittää tietojen yhdistelyn suunnan. SQL-kielessä yhdistelyä vastaa GROUP BY -operaatio, jossa SELECT –lausekkeessa ilmoitetaan aggregoitavat dimensioattribuutit. Kyselyn tulosrelaation dimensioattribuuttien arvoja voidaan rajoittaa WHERE lauseella.

SQL -lause tuottaa aina kyselyn tuloksena taulukon, jossa mitta-attribuuttien arvoja ei ole laskettu yhteen rivien ja sarakkeiden yhteissummiksi. Tietojen analysoinnin helpottamiseksi ja havainnollisuuden vuoksi loppukäyttäjä haluaa nähdä myös tällaiset mitta-attribuuttien yhteenvetotiedot. SQL-92 standardin mukaisella kyselykielellä tätä ei ole mahdollista tuottaa yhdellä lausekkeella [Date ja Darwen, 1997]. Edeltävän kyselyn tuloksia ei useinkaan voi käyttää hyväksi seuraavassa ROLL-UP –kyselyssä, koska käyttäjä voi muuttaa kyselyn dimensioita. Siksi ROLL-UP –operaatiolla peruskuutiosta johdetut yhteenvetotiedot joudutaan laskemaan uudelleen kyselyn prosessoinnin yhteydessä. Erityisesti raskaita ovat sellaiset ROLL-UP -kyselyt, joissa summaus tapahtuu tietojen alimmalta yksityiskohtaisimmalta tasolta dimensionhierarkian ylimmille tasoille. Tähän vaikuttaa luonnollisesti alimman tason ja aggregoitavan tason välisten hierarkiatasojen ja kyselyssä käytettyjen dimensioiden määrä. Edeltävän kyselyn tulosrelaatiota voidaan käyttää sitä seuraavan ROLL-UP –kyselyn syötteenä vain silloin, kun jälkimmäisessä kyselyssä tarvittavan kuution tiedot sisältyvät aiemman kyselyn tuottaman kuution tietoihin.



Kuva 4: ROLL-UP - ja DRILL-DOWN –operaatiot.

4.2. DRILL-DOWN

DRILL-DOWN –operaatio on vastakkainen ROLL-UP –operaatiolle. Tällä operaatiolla dimensiohierarkiassa siirrytään karkeammalta tasolta yksityiskohtaisemmalle tasolle. Tämä operaatio on tietojen analysoinnissa merkittävä, koska analysoinnin alussa käyttäjät haluavat ensin nähdä tiedot usein karkealla tasolla ja valikoiden sen jälkeen siirtyä yksityiskohtaisemmille tasoille.

DRILL-DOWN –operaatiossa aggregoidut mitta-attribuuttien arvojen yhteenvetotiedot joudutaan purkamaan alemmalle dimensiohierarkiatasolle. Tälle operaatiolle ei löydy vastinetta SQL –kielessä. Kun DRILL-DOWN –operaation yhteydessä laskentaoperaatiot joudutaan useimmiten aloittamaan dimensiohierarkian alimman tason mitta-attribuuttien arvoista GROUP BY -lausekkeella, niin tietomääristä riippuen kyselyn prosessointiaika voi olla hyvinkin pitkä. Täysin materialisoidusta kuutiosta DRILL-DOWN –operaatiossa tarvitut tiedot saadaan sellaisenaan ilman laskentaa. Osittain materialisoidusta kuutiosta haetaan ylin mahdollinen esilaskettu dimensiotaso, josta GROUP BY –operaatiolla johdetaan DRILL-DOWN –operaatiolla kyseltävät tiedot.

4.3. SLICE

Nimensä mukaisesti SLICE –toiminnolla kuutiosta otetaan se osa, jota halutaan tarkastella. SLICE –operaatio poistaa kuutiosta yhden tai useamman dimension, jolloin kaikki tuloksena saadun osakuution alkiot on laskettava uudelleen. Käyttäjän näkökulmasta katsottuna SLICE –operaatio tuottaa kuutiosta kaksiulotteisen sivun [OLAP Council]. Kuvan 5 kysely tuottaa osakuution kuvan 2 taulukkomuotoisesta OLAP –kuutiosta rajaamalla aluedimensio pois.

Myytyjen autojen määrä vuosittain ja tuotetyypeittäin.

```
SELECT vuosi, tyyppi, SUM(määrä) AS Määrä
FROM kuutio
GROUP BY vuosi, tyyppi
```

Kyselyn tulos:

Vuosi	Tyyppi	Määrä
2001	Astra	6
2001	Vectra	10
2001	Zafira	3
2002	Vectra	2
2002	Zafira	2

Kuva 5: SLICE –operaatio SQL –määrittelynä ja sen tuottama tulos kuvan 2 esimerkkikuutioon perustuen.

4.4. DICE

DICE –operaatio on toiminto, jolla kuutiosta rajataan ja tuotetaan osajoukko. Rajauksessa kuitenkin koko dimensioattribuutin sijasta rajataan osa pois perustuen dimensioattribuutin arvoihin [Pourabbas ja Rafanelli, 2000]. DICE –operaatio muistuttaa rajaukseltaan SLICE -operaatiota. Em. toiminnolla dimensioattribuuttitason arvojoukosta otetaan osajoukko joko valitsemalla sen alkiot eksplisiittisesti tai halutulta arvoväliltä. Kuvan 6 kyselyssä aikadimensiosta on otettu mukaan vuosi 2001 perustuen kuvan 2 esimerkkikuutioon.

Myytyjen autojen määrä vuodelta 2001 tuotetyypeittäin

```
SELECT vuosi, tyyppi, SUM(määrä) AS Määrä
FROM kuutio
WHERE vuosi IN('2001')
GROUP BY vuosi, tyyppi
```

Kyselyn tulos:

Vuosi	Tyyppi	Määrä
2001	Astra	6
2001	Vectra	10
2001	Zafira	3

Kuva 6: DICE –operaatio SQL –määrittelynä ja sen tuottama tulos kuvan 2 erimerkkikuutioon perustuen.

SLICE - ja DICE –operaatioilla kyselyn tuloksena saadaan aiemmalla kyselyllä tuotettua tulosta pienempi kuutio. SLICE - ja DICE –operaatioiden toteuttaminen on käyttäjän kannalta tietojen valintaa kuution tilan sallimissa rajoissa, jossa käyttäjä vaihtelee näytöllä dimensioita ja dimensioattribuuttien arvoja riveiltä sarakkeiksi ja päinvastoin [OLAP Council].

4.5. PIVOT -OPERAATIO

Agrawal *et al.* [1997] mainitsevat hyvin yleisellä tasolla, että PIVOT –operaatiolla kuutiosta näytetään toisenlainen näkymä. Thomsen [1997 s. 510] kuvaa pivot -operaatiota niin, että dimensioita vaihdetaan tai järjestetään näyttöruudulla eri tavoin. OLAP Council määrittelee PIVOT –operaation niin, että näyttöruudun näkymää muutetaan dimensioita vaihtamalla ja siirtelemällä. PIVOT –operaatiota tarvitaan kun peruskuution dimensioattribuuttien arvoja käytetään tuloskuution sarakkeina. Kuvissa 7-9 on kolme esimerkkiä PIVOT –operaation käytöstä, kun sitä sovelletaan kuvan 2 OLAP –kuutioon.

Tyyppi	Astra	Vectra	Zafira
Vuosi			
2001	6	10	3
2002	0	2	2

Vuosi	2001	2002
Tyyppi		
Astra	6	0
Vectra	10	2
Zafira	3	2

Kuva 7: PIVOT -operaatio rivien ja sarakkeiden vaihtamisessa.

Tyyppi	Astra		Vectra		Zafira	
Vuosi	2001	2002	2001	2002	2001	2002
Maanosa						
Eurooppa	1	0	4	2	3	2
Pohj.Amerikka	5	0	6	0	0	0

Tyyppi		Astra	Vectra	Zafira
Vuosi	**Maanosa**			
2001	Eurooppa	1	4	3
2001	Pohj.Amerikka	5	6	0
2002	Eurooppa	0	2	2
2002	Pohj.Amerikka	0	0	0

Kuva 8: PIVOT –operaatio, missä lähtötaulun sarakkeiden nimet muutetaan tulostaulun sarakkeen arvoiksi.

Tyyppi	Astra	Vectra	Zafira
Vuosi			
2001	6	10	3
2002	0	2	2

Tyyppi	Astra	Vectra	Zafira
Maanosa			
Eurooppa	1	6	5
Pohj.Amerikka	5	6	0

Kuva 9: PIVOT -operaatio, jossa dimensioattribuutti vaihdetaan.

Kolmannessa esimerkissä kuvassa 9 näyttöruudun rividimensio vuosi korvataan maanosadimensiolla. Tässä on esitelty vain osa näyttöruudun näkymien muunteluvaihtoehdoista.

PIVOT -operaatiota ei ole formaalisti määritelty. Se on yksi operaatio muiden OLAP –operaatioiden joukossa, jolla näkymien muuntelu saadaan joustavaksi ja käyttäjäystävälliseksi. Yhteenvetotietojen ryhmittely voi siis tapahtua kaavio- tai ilmentymätasoon perustuen. PIVOT –operaatio edellyttää kaavio- ja ilmentymätason vaihtamista keskenään. SQL –kielen lausekkein PIVOT –operaatiota ei saa ilmaistua yhdellä lauseella. Tällaiset muutokset voidaan saada aikaiseksi ns. monivaiheisella SQL –menetelmällä, jossa SQL –lausekkeita suoritetaan useita peräkkäin. Em. prosessissa tehdään tarvittava määrä aputauluja ja näitä yhdistelemällä kuutiosta voidaan tuottaa PIVOT –operaation tavoin erilaisia näkymiä. SQL –määrittelynä PIVOT –operaatio on erityisen työlästä. Analysointityökalu sisältää ensisijaisesti pivotointiominaisuuden.

4.6. PUSH JA PULL

PUSH- ja PULL –operaatioilla dimensioattribuutteja ja mitta-arvoattribuutteja käsitellään symmetrisesti [Agrawal *et al.*, 1997] [Pourabbas ja Rafanelli, 2000]. Tällöin dimensioattribuutit ja mitta-attribuutit ovat käsittelyn kannalta samanarvoisia.

PUSH –operaatiolla dimensioattribuutin arvo siirretään tai kopioidaan mitta-attribuuttiarvoksi Kuvan 2 OLAP –kuutioon sovellettuna PUSH –operaatiolla tyyppi voidaan siirtää tai kopioida kuution alkioihin mitta-arvoattribuuttien arvojen lisäksi. Tällöin yksittäisessä kuution alkiossa voisi esiintyä esim. arvot <Zafira, 3, 28 500>, jossa 3 ilmaisee määrää ja 28 500 ilmaisee ahinnan.

PULL –operaatiolla mitta-attribuutti siirretään dimensioattribuutiksi. Tällöin esim. kuvan 2 OLAP –kuution mitta-arvoattribuutti määrä voidaan siirtää em. peruskuutiosta johdettuun uuteen kuutioon dimensioattribuutiksi.

Usein on tarve ryhmitellä arvoja ja suorittaa aggregointeja näiden ryhmien mukaan. Esimerkiksi autojen arvoja, yksikköhintaryhmiä, pitäisi voida käyttää dimensioina aggregoinnissa. PUSH- ja PULL –operaatioita voidaan saada aikaiseksi SQL –kielellä, kun looginen malli on rakenteeltaan sellainen, että numeerisen dimensioattribuutin arvot voidaan muuttaa mitta-attribuutin arvoiksi, ja päinvastoin.

SQL –kielellä WHERE –ehdossa voidaan määritellä lukuarvon ala- ja yläraja ja siten saada muodostetuksi yksi mielivaltainen ryhmä, jonka mukaan aggregointi suoritetaan. PULL –operaatio voidaan toteuttaa SQL –kielellä siten,

että tietyllä välillä olevista lukuarvoista muodostetaan dimensioita. Kuvan 10 kyselyssä tiedot tulostetaan tietyltä mitta-attribuuttien arvoväliltä. SQL:ssä kyselyllä voidaan tuottaa vain yhden ryhmän tiedot.

Laske maanosittain tuotetyypeittäin hintatason 25 000-30 000 välillä myytyjen autojen lukumäärä.

```
SELECT maanosa, yyppi, SUM(määrä) AS Määrä_25000_30000
FROM kuutio
WHERE hinta BETWEEN 25000 AND 30000
GROUP BY  maanosa, tyyppi
```

Kyselyn tulos:

Maanosa	Tyyppi	Määrä_25000_30000
Eurooppa	Astra	1
	Vectra	2
	Zafira	5

Kuva 10: PULL –operaation määrittäminen SQL:llä.

PUSH- ja PULL –operaatioilla saadaan haluttua lisätoiminnallisuutta moniulotteisen tiedon käsittelyyn. Näillä operaatioilla pyritään symmetriseen dimensioiden ja mitta-arvojen käsittelyyn. SQL –kielessä näitä ominaisuuksia ei ole.

4.7. SELECT

SELECT –operaatio on DICE –operaatiota vastaava operaatio. Tämä operaatio on analoginen relaatioalgebran valintaoperaatiolle. Siinä valinnasta jätetään pois ne dimensioattribuuttien tai mitta-attribuuttien arvot, jotka eivät täytä valinnan ehtoja [Pourabbas ja Rafanelli, 2000]. SELECT -operaatiolla kuutiosta valitaan tarkasteltavaksi SELECT -listan sisältämät dimensioattribuuttien ja mitta-arvoattribuuttien arvot. Kuvan 10 esimerkkikysely ilmaisee, miten valintaehto määritellään SQL:ssä.

5. AGGREGOINTIFUNKTIOT JA SUMMAUTUVUUS

OLAP –järjestelmän analysointikyky perustuu osittain tiettyihin aggregointifunktiohin, joiden avulla mitta-attribuuttien arvoja aggregoidaan eri dimensiotasoille. Lenz ja Thalheim [2001] luokittelevat aggregointifunktiot niiden laskennan monimutkaisuuden perusteella yksinkertaisiin ja kompleksisiin funktioihin. Ne voidaan luokitella myös toisin perustein eli distributiivisiin, algebrallisiin ja holistisiin funktioihin [Gray *et al.*, 1995].

Näiden funktioiden tulee luonnollisesti tuottaa oikea tulos mitta-attribuuttien yhteenvetotietoina, kun niitä sovelletaan tietokuutioon. Summautuvuudella tarkoitetaan sitä, että mitta-attribuuttien arvoihin sovellettavat funktiot antavat tulokseksi aina oikeat yhteenvetotiedot. Summautuvuus on OLAP –järjestelmille ja tilastollisille tietokannoille välttämätön ominaisuus. Jos summautuvuuden ehdot eivät toteudu, kyselyjen tuloksista tehdyt yhteenvedot ja johtopäätökset voivat olla virheellisiä.

5.1. Aggregointifunktiot

Aggregointifunktioiden käyttö OLAP –kyselyissä on välttämätöntä silloin, kun numeerisia tietoja lasketaan dimensiohierarkian mukaisesti karkeammalle tasolle ROLL-UP –operaatiolla tai yksityiskohtaisemmalle tasolle DRILL-DOWN –operaatiolla. SQL –kielen funktiot AVERAGE (aritmeettinen keskiarvo), COUNT (lukumäärä), MAX (suurin arvo), MIN (pienin arvo) ja SUM (lukujen summa) ovat yksinkertaisia funktioita. Monimutkaisemmat funktiot voidaan jaotella seuraaviin luokkiin: liikkuvat yhteissummat (moving totals), kumulatiiviset prosentuaaliset osuudet (culumulative percentage), arvotus (rank), moodi (mode), keskihajonta (standard deviation), varianssi (variance) sekä keskimääräinen poikkeama (average deviation) [Lenz ja Thalheim, 2001].

5.1.1. Distributiiviset funktiot

Distributiiviset funktiot sallivat aggregoinnin aiemmin aggregoiduille luvuille. Funktio f() on distributiivinen, jos jaettaessa joukon X alkiot erillisiin osajoukkoihin $X = X_1 \cup X_2 \cup \ldots \cup X_{n,,}$ ja sovellettaessa näihin erikseen funktiota f(),on olemassa sellainen funktio g(), että $f(X) = g(f(X_1), f(X_2), \ldots, f(X_n))$. SQL:n distributiivisia funktioita ovat COUNT, MAX, MIN ja SUM. Funktio f = g kaikilla muilla funktioilla paitsi COUNT –funktiolla, jolla g on SUM –funktio.

5.1.2. Algebralliset funktiot

Funktio f() on algebrallinen, jos sen jokainen argumentti on distributiivisen funktion tulos. Algebrallisissa funktioissa välitulokset lasketaan distributiivisilla funktioilla, ja algebrallinen lauseke lasketaan näiden välitulosten perusteella. Keskiarvoa laskettaessa lukujen summa ja lukujen määrä ovat keskiarvofunktion argumentteja. Summa jaetaan lukumäärällä, josta funktion tuloksena saadaan keskiarvo. Tällainen funktio on SQL –kielen AVG (keskiarvo). SQL –kielestä puuttuvia mutta usein analysoinnissa tarvittavia funktioita ovat keskihajonta (standard deviation), MaxN ja MinN. MaxN –funktio antaa tuloksena N suurinta arvoa ja MinN vastaavasti N pienintä arvoa.

5.1.3. Holistiset funktiot

Kaikki muut kuin distributiiviset ja algebralliset funktiot ovat holistisia funktioita. Holistisen funktion toteuttamisen aikana funktion vaatimien välitulosten määrä ja niiden tarvitsema muistitila on rajoittamaton. Näitä funktioita ovat mediaani, moodi ja arvotus. Arvotusfunktio (rank) tuo N määrän rivejä alkaen arvojärjestyksessä suurimmasta luvusta. Tästä on esimerkkinä kysely, joka antaa luettelon viidestä eniten myydystä automerkistä laskevassa järjestyksessä. Tällaiset funktiot puuttuvat SQL –kielestä.

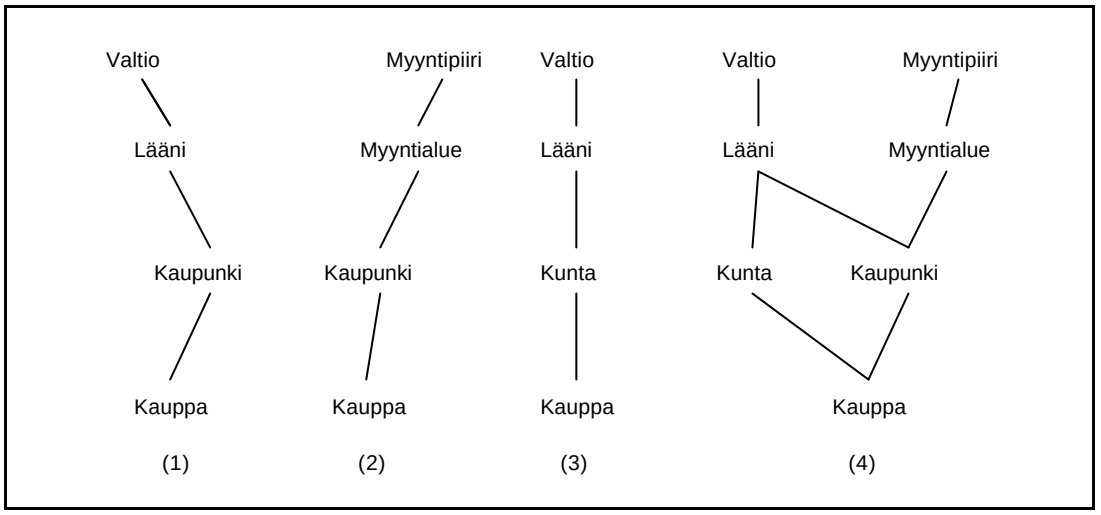
5.2. Dimension hierarkkisuus

Dimensiohierarkiatasot antavat mahdollisuuden tarkastella mitta-attribuutteihin liittyviä arvoja eri karkeisuustasoilla. Hurtadon ja Mendelzonin [2002] mukaan dimensio voi olla homogeeninen tai heterogeeninen. Homogeenisessä dimensiossa jokainen alemman tason dimensioattribuutin arvo sisältyy yhteen ja vain yhteen sitä välittömästi seuraavaan ylemmän tason dimensioattribuutin arvoon. Tällaisessa homogeenisessä summautuvuuden kolme ehtoa täyttävässä dimensiossa tietoja karkeistava ROLL-UP –operaatio tuottaa oikean tuloksen. Homogeenisesta dimensiohierarkiasta puuttuu monihierarkkisuus [Pourabbas ja Rafanelli, 2000]. Monihierarkisessa dimensiossa tietoja voidaan aggregoida ylemmille tasoille useampaa kuin yhtä polkua pitkin. Kuvan 11 esittämässä monihierarkisessa dimensiossa (4) on hierarkiat (1), (2) ja (3). Monihierarkiasta (4) voidaan muodostaa mm. seuraavanlaisia hierarkioita:

Kauppa > Kunta > Valtio

Kauppa > Lääni > Valtio

Kauppa > Myyntialue > Myyntipiiri



Kuva 11: Monihierarkinen dimensio.

Heterogeenisessa dimensiossa summautuvuuden toteutuminen on kompleksisempaa. Hurtado ja Mendelzon [2002] esittävät heterogeenisille dimensioille eheyssääntöjä, jotka sisällytetään OLAP –operaatioiden yhteyteen, jotta summautuvuus toteutuu.

Pourabbas ja Rafanelli [2000] luokittelevat dimensiohierarkian täydelliseksi tai osittaiseksi seuraavasti.

5.2.1. Täydellinen luokitteluhierarkia

Dimension kahden hierarkiatason välinen suhde muodostaa muuttujien osalta sisältyvyysfunktion. Jos jokainen alemman tason dimensioattribuutin arvo sisältyy yhteen ja vain yhteen ylemmän tason dimensioattribuutin arvoon ja jokaiseen ylemmän tason dimensioattribuutin arvoon sisältyy vähintään yksi alemman tason dimensioattribuutin arvo, kyseessä on täydellinen luokitteluhierarkia, jota sanotaan täydelliseksi kuvaukseksi. Tästä esimerkkinä on hierarkiasuhde osavaltio–kaupunki, kun hierarkiatason osavaltio tietty osavaltio sisältää tietyn osavaltion kaikki kaupungit ja jokaiseen osavaltioon sisältyy vähintään yksi kaupunki.

5.2.2. Osittainen luokitteluhierarkia

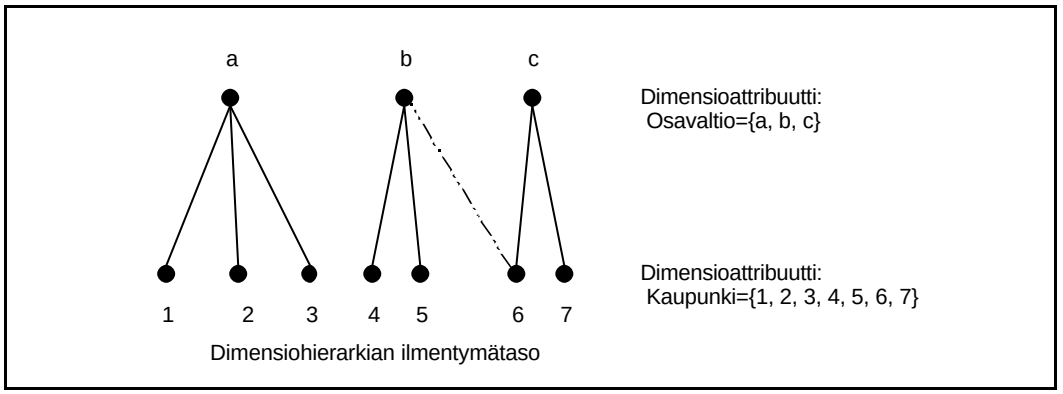
Osittaisessa luokitteluhierarkiassa dimension kahden peräkkäisen hierarkiatason välisistä dimensioattribuuttien arvoista vähintään yhdellä ei ole täydellistä kuvausta. Esimerkkinä tästä on hierarkiataso osavaltio, joka ei

sisälläkään kaikkia tietyn osavaltion kaupunkeja. Tällöin ylemmällä osavaltiotasolla ei ole tietoa siitä, että kaikki osavaltion kaupungit eivät sisälly kaupunkien joukkoon. Tällaisessa tapauksessa osavaltion mitta-attribuuttien arvojen yhteenvetotiedot voidaan tulkita virheellisesti esimerkiksi laskettaessa osavaltion asukasmäärää.

5.3. Aggregointifunktioiden summautuvuuden edellytykset

Tilastollisissa tietokannoissa ja OLAP –käsittelyssä tietojen summautuvuus on erittäin tärkeä ominaisuus, jonka täyttämättä jättäminen voi antaa virheellistä tietoa ja johtaa vääriin yhteenvetoihin ja päätöksiin. Lenz ja Shoshani [1997] esittävät kolme summautuvuuden ehtoa, jotka ovat riittäviä, ja takaavat kyselyjen summautuvuuden oikeellisuuden. Tässä tutkimuksessa heidän käyttämäänsä käsitettä luokittelurakenne (classification structure) vastaa dimensiohierarkia, ja käsitettä kategoria-attribuutti vastaa dimensioattribuutti.

Ensimmäinen ehto on dimensioattribuuttien erillisyys. Tämän ehdon mukaan dimensioattribuuttien arvojen joukosta muodostetaan seuraavalle ylemmälle dimensiohierarkiatasolle erillisiä osajoukkoja siten, että alemman dimensioattribuutin tietty arvo voi sisältyä yhteen ja vain yhteen ylemmän dimensioattribuutin arvoon. Tämä on sama kuin täydellinen luokitteluhierarkia. Kuvassa 12 lehtisolmutasolla dimensioattribuuttien arvojen joukkoon kuuluva arvo 6 sisältyy kahteen välittömästi ylemmän dimensioattribuuttitason joukkoon b ja c. Tässä hierarkiassa ei toteudu dimensioattribuuttien erillisyysehto.



Kuva 12: Erillisten osajoukkojen vastainen ilmentymätaso.

Dimensioattribuuttien erillisyysehto sisältää myös vaatimuksen, jonka mukaan kullakin hierarkiatasolla dimensioattribuuttien arvot tulee olla yksikäsitteisiä. Tästä esimerkkinä on ehtoa rikkova tilanne kaupunkien ja osavaltioiden välisessä suhteessa, kun eri osavaltioissa on samannimisiä kaupunkeja.

Tällaiset tapaukset saadaan helposti korjattua muuttamalla ristiriidan aiheuttaman dimensioattribuutin arvo toisenlaiseksi niin, että siitä tulee yksikäsitteinen. Tällaiset tapaukset tulevat useimmiten esille jo funktionaalisten riippuvuusanalysointien yhteydessä.

Toinen ehto on täydellisyyden toteutuminen. Täydellisyydellä tarkoitetaan, että ilmentymätasolla dimensioattribuuttien arvojen joukosta ei puutu yhtään arvoa, ja että jokainen arvo sisältyy aina johonkin ylemmän hierarkiatason dimensioattribuutin arvoon. Jos esim. autotehtaan valmistamista automalleista jätetään yksi pois, saadaan virheellinen lukumäärä tehtaan valmistamista autojen yhteismäärästä. Täydellisyyden toteutumiseen vaikuttaa myös mitta-attribuutin tyyppi. Jos mitta-attribuuttina on liikenneonnettomuuksissa kuolleiden määrä ja nämä arvot on tallennettu kaupungeittain, ylemmälle osavaltiohierarkiatasolle lasketut yhteenvetotiedot antavat virheellisen tuloksen osavaltiossa liikenneonnettomuuksissa kuolleiden määrästä. Yhteenvetotiedoista puuttuu onnettomuudet, jotka tapahtuivat kaupunkien ulkopuolella. Jos mitta-attribuuttina on museoiden määrä ja tiedetään, että museoita on vain kaupungeissa, saadaan ylemmille dimensiohierarkiatasoille yhteenvetotiedot laskettua oikein.

Kolmas välttämätön summautuvuuden ehto riippuu dimensioattribuutin ja summattavan mitta-attribuutin tyypistä ja sovellettavasta funktiosta. Summautuvuutta sovelletaan aivan eri tavalla aikadimensioihin (kuten päivä tai kuukausi) kuin ei ajallisiin dimensioihin (kuten autotehdas tai valtio). Mitta-attribuutit voidaan luokitella tyyppeihin: flow, stock tai value-per-unit.

Flow -tyyppiset mitta-attribuutit viittaavat tiettynä ajanjaksona rekisteröityihin lukuihin. Ne viittaavat tiettynä aikana tapahtuviin kumulatiivisiin määriin, jotka tallennetaan mitta-attribuuttien arvoina ajanjakson lopussa. Tätä tyyppiä ovat esim. kuukauden aikana myytyjen autojen määrä, tai päivittäin lähetettyjen tekstiviestien määrä. Flow -tyyppiset attribuutit ovat summautuvia ja antavat mielekkäitä tuloksia.

Stock –tyyppiset mitta-attribuutit kuvaavat määriä tai tilannetta tiettynä ajankohtana. Henkilölukumäärä ja varastotilanne kuukauden viimeisenä päivänä ovat esimerkkejä stock –tyyppisistä mitta-attribuuteista. Kuukauden viimeisenä päivänä laskettujen henkilöiden määrä ei ole summautuva, koska summattaessa tällaista mitta-attribuuttia kahdentoista eri kuukauden ajalta

summa ei anna mielekästä tulosta vuosittaisesta henkilömäärästä. Stock –tyyppinen mitta-attribuutti ei ole summautuva.

Value-per-unit –tyypin arvo on jonkin objektin yksikköarvo. Koska yksikköarvo luonnollisesti muuttuu eri ajanjaksoina siihen ei voi soveltaa summausfunktiota temporaalisen dimension eikä muunkaan dimension suhteen.

Jotta kolmas summautuvuuden ehto täyttyisi, on tutkittava dimensioattribuutin tyyppi (temporaalinen, ei-temporaalinen), mitta-attribuutin tyyppi (flow, stock, value-per-unit), sekä näihin sovellettava funktio.

6. TIETOVARASTON LOOGISET MALLIT

OLAP –tietojenkäsittely poikkeaa merkittävästi perinteisestä tapahtumaorientoituneesta käsittelystä. Kyselyt kohdistuvat suuriin tietomääriin, jotka sisältävät historiatietoja koskien useita ajanjaksoja. Lisäksi kyselyjen käsittelemiin tietoihin kohdistuu runsaasti laskentaa. Kyselyt ovat usein kompleksisia ja ennalta arvaamattomia. Näistä seikoista huolimatta kyselyjen vastinajat pitää olla kohtuullisia ja ennakoitavia tietojen vuorovaikutteisessa analysoinnissa. Nämä seikat edellyttävät, että tietovaraston looginen malli on suunniteltu moniulotteisuutta tukevaksi.

Tapahtumakäsittelyyn perustuvissa järjestelmissä tietojen looginen rakenne on käyttäjiltä näkymättömissä. Tietovaraston loogisen mallin tarkoituksena on kuvata moniulotteisesti organisoitua tietoa käyttäjälle havainnollisella ja ymmärrettävällä tavalla. Käyttäjien tulee nähdä millä tavalla ryhmiteltyinä he voivat tarkastella mitta-arvojen yhteenvetotietoja. Moniulotteisesta loogisesta mallista tulee voida kyselyin tuottaa mahdollisimman monia näkymiä vuorovaikutteisessa analysoinnissa. Kimball [2000a] [2000b] nimittää moniulotteista loogista mallintamista ja tietovarastoa dimensionaaliseksi, jossa dimensiot ja mitta-arvot ovat keskeiset peruskäsitteet.

Tässä kappaleessa tarkastellaan erilaisia OLAP –käsittelyyn kehitettyjä loogisia malleja ja niiden vaikutusta kyselyn määrittelyyn kyselykielellä. Lisäksi tutkitaan miten dimensioiden määrä, dimensiohierarkian syvyys ja leveys vaikuttaa kyselyjen määrittelyyn kyselykielellä. Tarkasteltavien loogisten mallien lähtökohtana on se, että tietovarasto sisältää mallien mukaiset peruskuutiot, joihin kyselyt kohdistuvat [Niemi, 2001, s. 24]. Näistä peruskuutioista johdetaan kyselyjen tuloksena uusia aggregoituja kuutioita, joissa mitta-attribuuttien arvoja tarkastellaan karkeammalla tai yksityiskohtaisemmalla tasolla. Tarkasteltaviin loogisiin malleihin ei sisälly peruskuutiosta johdettuja materialisoituja näkymiä, joihin kyselyjä voidaan myös kohdistaa. Tarkasteltavana kyselykielenä on SQL.

Tietty moniulotteinen looginen malli vaikuttaa kyselyjen formulointiin joko suotuisasti tai haitallisesti. Suotuinen vaikutus merkitsee sitä, että kyselyn muodostamiseen tarvitaan vain muutama ilmaisurakenne, kyselyn tekeminen on intuitiivista ja se on selkeästi tulkittavissa. Jos kyselyn muodostamiseen tarvitaan monia rakenteita, tällöin vaarana on, että niistä muodostuu

monimutkaisia, vaikeasti käsitettäviä ja hallittavia ja lisäksi ne voidaan tulkita monella eri tavalla.

Aluksi tarkastellaan universaalirelaatiota tietovaraston loogisena mallina. Seuraavaksi tarkastellaan muita yleisimpiä loogisia malleja: normalisoimaton kuutio, tähtimalli, lumihiutalemalli, konstellaatiomalli ja monikuutiomalli. Näitä yleisimpiä malleja yhdistelemällä voidaan tietysti luoda erilaisia muunnelmia loogisista malleista.

6.1. Universaalirelaatio

Universaalirelaation käsitteellä tarkoitetaan relaatiokaaviota, jossa kaikki tietokannan tiedot on sijoitettu yhteen isoon relaatioon [Ullman, 1989, s. 1026]. Elmasri ja Navathe [1994, s. 401] esittävät universaalirelaation R kaaviona, joka on muotoa $R = \{A_1, A_2, \ldots, A_n\}$, jossa A_n on relaation attribuutti. Koko tietovarasto ja tietokuutio voidaan esittää myös universaalirelaationa.

Tarkastellaan kuvan 13 esittämää esimerkkiä siitä, kuinka tietokuutio voidaan esittää universaalirelaationa. Sen kaaviotaso (rivi 1) muodostuu kolmesta dimensioattribuutista (D -alkuiset) ja kahdesta mitta-attribuutista (M -alkuiset). Dimensiot (D1-, D2- ja D3- alkuiset) muodostavat vastaavat dimensiohierarkiat esitettynä alimmalta tasolta ylimmälle tasolle seuraavasti:

D13 > D12 > D11
D23 > D22 > D21
D32 > D31

Dimensiohierarkiassa dimensioattribuuttien välillä on funktionaalinen rakenteellinen riippuvuus. Kuvan 13 rivit 2-13 muodostavat relaation ilmentymätason. Ilmentymätasolla mitta-attribuuttien arvot liittyvät jokaisen dimension D1, D2 ja D3 alimman tason johonkin arvoon (rivit 2 ja 8). Esim. riveillä 2, 6 ja 7 on dimension D1 dimensioattribuuttien arvot, jotka liittyvät rivin 2 mitta-attribuuttien arvoihin.

	D1			D2			D3			
(1)	**D11**	**D12**	**D13**	**D21**	**D22**	**D23**	**D31**	**D32**	**M1**	**M2**
(2)			d13_1			d23_1		d32_1	m1_1	m2_1
(3)							d31_1	d32_1		
(4)					d22_1	d23_1				
(5)				d21_1	d22_1					
(6)		d12_1	d13_1							
(7)	d11_1	d12_1								
(8)			d13_1			d23_2		d32_2	m1_2	m2_2
(9)							d31_2	d32_2		
(10)					d22_2	d23_2				
(11)				d21_2	d22_2					
(12)		d12_2	d13_2							
(13)	d11_2	d12_2								

Kuva 13: Universaalirelaatioesitystapa kuutiolle.

Kuvan 13 esimerkkitapauksessa relaatiossa on 10 * 12 = 120 alkiota, joista ¾ on tyhjää. Kun dimensiohierarkiatasoja lisätään, niin tyhjien alkioiden osuus alkioiden kokonaismäärästä kasvaa. Samoin tapahtuu kun dimensioattribuutin arvojen lukumäärä suurenee. Kahden hierarkiatason dimensioattribuuttien välinen suhde ilmaistaan yhdellä rivillä (esim. rivi 6: d12_1 ja d13_1), jolloin muut rivin alkiot ovat ilman tietoa. Todellisessa käytännön tilanteessa mitta-attribuuttien arvoja sisältävien rivien (kuva 13 rivit 2 ja 8) osuus ilmentymätason kokonaisrivimäärästä on huomattavan suuri ja dimensiohierarkiaa kuvaavien rivien määrä (kuva 13 rivit 3-7 ja 9-13) pysyy melko vakiona. Tällöin tyhjien alkioiden osuus alkioiden kokonaismäärästä pienenee.

Kun kuvan 13 esittämään ilmentymätasoon tehdään kaksi esimerkkikyselyä a ja b, niin kyselyt muodostuvat seuraavanlaisiksi:

a) Mitta-attribuutin M1 arvot on laskettu yhteen dimension D1 ylimmälle tasolle:

```
SELECT R2.D11,
SUM(M1)
FROM    R, R R1, R R2
WHERE R.M1 IS NOT NULL
AND     R.D13   = R1.D13
AND     R1.D12 = R2.D12
GROUP BY R2.D11
```

b) Mitta-attribuutin M1 arvot on laskettu yhteen dimensioiden D1, D2, ja D3 ylimmille tasoille:

```
SELECT R2.D11, R4.D21, R5.D31,
SUM(M1)
FROM R, R R1, R R2, R R3, R R4, R R5
WHERE R.M1 IS NOT NULL
AND     R.D13   = R1.D13
AND     R1.D12  = R2.D12
AND     R.D23   = R3.D23
AND     R3.D22  = R4.D22
AND     R.D32   = R5.D32
GROUP BY R2.D11, R4.D21, R5.D31
```

Em. kyselyissä peruskuutiona on yksi ja sama universaalirelaatio. Kyselyssä a) mitta-attribuutin M1 arvo aggregoidaan yhden dimension (D1) ylimmälle tasolle. Tällöin universaalirelaatioon itseensä täytyy tehdä liitos kahdesti. Kohdan a) tapauksessa kyselyn formulointi pysyy selkeänä kun dimensiohierarkia on matala.

Kohdan b) kyselyssä mitta-attribuutin M1 arvot on summattu jokaisen kolmen dimension ylimmälle tasolle. Tästä esimerkistä näemme, että universaalirelaation osalta mitta-attribuuttien arvojen summaus ylimmälle tasolle vaatii liitosoperaation jokaisen kahden dimensioattribuuttitason välille.

Universaalirelaatioon itseensä tehtävien liitosoperaatioiden määrä kyselyssä kasvaa dimensiohierarkian syvyyden mukaan, kun mitta-arvoja yhdistellään dimension ylimmille tasoille. Kyselyt voivat muodostua hyvinkin pitkiksi määrittelyiksi, kun yhdistelyssä on mukana useita dimensioita.

6.2. Normalisoimaton kuutio

Peruskuution dimensioattribuutit ja mitta-attribuutit voidaan esittää yhtenä relaatiokaaviona, joka ei ole normalisoidussa muodossa. Moody ja Kortink [2000] nimittävät tällaista latteaksi kaavioksi (flat schema). Tässä tutkielmassa siitä käytetään nimitystä normalisoimaton kuutio. Seuraava relaatiokaavio esittää tällaisen loogisen mallin kaaviotason:

$$C = \{D1_1, D1_2, \ldots, D1_n, D2_1, D2_2, \ldots, D2_n, \ldots, Dm_1, Dm_2, \ldots, Dm_n, M_1, M_2, \ldots, M_n\}$$

C on relaatiokaavio, jossa D –alkuiset attribuutit ovat dimensioattribuutteja ja M –alkuiset attribuutit mitta-attribuutteja. Dimensiot D1, D2, …, Dm kukin muodostavat dimensiohierarkian. Relaatiokaavion C attribuutit $D1_1$, $D1_2$, …, $D1_n$ muodostavat dimensiohierarkian $D1_n > D1_{n-1} > \ldots > D1_1$, jossa $D1_n$ on hierarkian alin ja $D1_1$ hierarkian ylin dimensioattribuutti.

	D11	D12	D13	D21	D22	D23	D31	D32	M1	M2
(1)										
(2)	d11_1	d12_1	d13_1	d21_1	d22_1	d23_1	d31_1	d32_1	m1_1	m2_1
(3)	d11_2	d12_2	d13_1	d21_2	d22_2	d23_2	d31_2	d32_2	m1_2	m2_2

Kuva 14: Normalisoimaton kuutio

Kuvassa 14 esitetyn normalisoimattoman kuution ilmentymätaso ja kaaviotaso on sama kuin edellä esitetty universaalirelaatio kuvassa 13. Normalisoimattomassa kuutiossa mitta-attribuutilla on jokaisella rivillä arvo. Samoin jokaisella mitta-attribuutin arvoon liittyvällä dimensioattribuutilla on rivillä arvo. Normalisoimattoman kuution ilmentymätasolla jokaisen dimension dimensiohierarkia on esillä eksplisiittisesti kullakin rivillä. Tällaisen loogisen mallin ilmentymätasossa rivejä on aina talletettujen mitta-arvojen määrä, joten dimensiohierarkian esittäminen ei vaadi erillisiä rivejä kuten universaalirelaatiossa.

Normalisoimattoman kuution haittana voidaan pitää sitä, että se sisältää hyvin paljon attribuutteja ilmentymätasolla. Sen sisällön tulkittavuuden ilmeisyys vähenee, kun siihen lisätään attribuutteja. Tästä kaaviosta moniulotteisuus ei ole helposti havaittavissa. Dimensioita ja dimensiohierarkiaa on vaikea hahmottaa ekstensionaalisella tasolla vaikka dimensioattribuutit on järjestetty hierarkian mukaan nousevaan tai laskevaan järjestykseen. Selkeys vähenee entisestään, jos dimensiohierarkia on syvä tai dimensio on monihierarkinen. Samoin käy kun normalisoimattomaan kuutioon lisätään dimensioattribuuttiin liittyvää lisätietoa, joka lisää relaation attribuuttien määrää. Esimerkiksi dimensioattribuuttiin kauppa voidaan liittää tieto kaupan osoitteesta tai kaupan liiketilan neliömäärästä. Normalisoimattomassa kuutiossa sama tieto toistuu monta kertaa. Hierarkiatasojen suuri määrä ja yksittäisen dimensioattribuutin iso arvojoukko lisäävät redundanttia tietoa ja taulun kokoa. Dimensiohierarkian rakenteelliset muutokset on työlästä toteuttaa normalisoimattomaan kuutiomalliin.

On mahdollista, että kahdella eri normalisoimattomalla kuutiolla on yksi tai useampi yhteinen dimensio, jonka suhteen relaatiot voisi yhdistää toisiinsa.

Tästä seuraa, että molemmissa relaatioissa on tallennettu tiedot samasta dimensiosta ja sen hierarkiasta. Dimensiotietojen ylläpito ja ajantasaisuus saattaa aiheuttaa ongelmia tällaisessa ratkaisussa.

SQL –kielellä tehtävien OLAP -kyselyjen kannalta normalisoimaton kuutio loogisena mallina on selkeä. FROM –lauseessa tarvitaan vain yhden taulun nimi. Koska tauluja on vain yksi ja jokaisella dimensioattribuutin rivillä on arvo, vältytään taulujen välisiltä liitosoperaatioilta. Kun kuvan 14 esittämään relaatioon tehdään vastaavat kaksi kyselyä kuin edellä kuvattuun universaalirelaatioon, saadaan seuraavat SQL –lausekkeet:

a) Mitta-attribuutin M1 arvot on laskettu dimensioattribuutin D1 ylimmälle tasolle:

```
SELECT R.D11,
SUM(M1)
FROM R
WHERE R.M1 IS NOT NULL
GROUP BY R.D11
```

b) Mitta-attribuutin M1 arvot on laskettu dimensioattribuuttien D1, D2, ja D3 ylimmille tasoille:

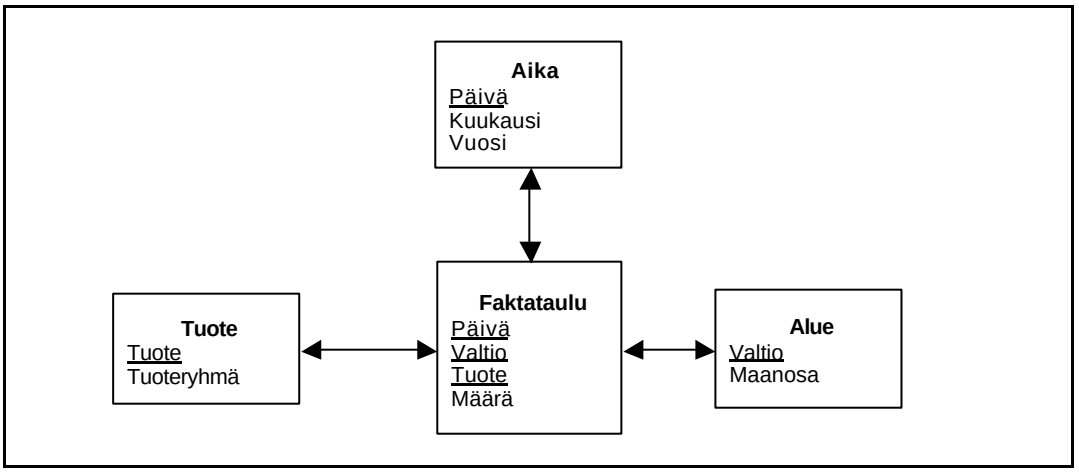
```
SELECT R.D11, R.D21, R.D31, SUM(M1)
FROM R
WHERE R.M1 IS NOT NULL
GROUP BY R.D11, R.D21, R.D31
```

WHERE –lauseke yksinkertaistuu, koska taulujen välisiä liitoksi ei tarvita. Tällöin WHERE –lausekkeella ilmaistaan enimmäkseen DICE- ja SLICE –operaatioita soveltamalla SQL:n IN, NOT IN, BETWEEN, AND tai OR predikaatteja. Normalisoimaton kuutio on kyselyn muodostamisen kannalta selkeä. Rivimäärältään pienehköissä relaatioissa normalisoimaton kuutio on selkeä ja käyttökelpoinen.

6.3. Tähtimalli

Normalisoimattomasta kuutiomallista voidaan johtaa looginen tähtimalli. Tähtimallissa jokainen dimensio kuvataan omana taulunaan, joka sisältää dimensiohierarkian. Tässä mallissa dimensiohierarkian sisältämät dimensioattribuutit ovat kaikki samassa taulussa. Mitta-arvot ja niihin liittyvät dimensiotiedot esitetään monidimensionaalisesti organisoituna rakenteena faktataulussa. Faktataulu ja siihen liittyvät dimensiotaulut muistuttavat visualisoituna tähden muotoa, josta malli on saanut nimensä. Tällaista loogista mallia sanotaan tähtimalliksi, kun dimensiot esitetään normalisoimattomassa muodossa [Chaudhuri ja Dayal, 1997].

Kuvan 15 esittämässä esimerkissä faktataulu on kuvion keskellä ja siihen liittyvät aika-, tuote- ja aluedimensiot muodostavat tähden sakarat. Dimensiotaulujen ensisijaisten avainten yhdistelmä muodostaa faktataulun ensisijaisen avaimen. Kuvassa 15 dimensioattribuutit päivä, valtio ja tuote muodostavat yhdessä faktataulun ensisijaisen avaimen. Ensisijainen avain määrittelee yksiselitteisesti faktataulun yksittäisen rivin mitta-attribuutin arvon. Relaationa faktataulu on normalisoidussa muodossa. Faktataulun kukin dimensioattribuutti on vierasavain sitä vastaavaan dimensiotauluun. Dimensiotaulun ensisijainen avain viittaa tällaiseen faktataulun dimensioattribuuttiin. Dimensiotaulun hierarkiatasoja merkitsevien dimensioattribuuttien välillä vallitsee transitiivinen riippuvuus. Dimensiotaulu sisältää kaikki dimensioon liittyvät tiedot, dimensiotasot ja mahdolliset dimensiotasoihin liitetyt lisätiedot.



Kuva 15: Tähtimalli.

Tähtimallin etuna pidetään yksinkertaista ja selkeästi tulkittavaa rakennetta, jossa taulujen määrä muodostuu dimensioiden määrästä ja faktataulusta. Tähtimallin mieltäminen moniulotteiseksi tietorakenteeksi on edellisiin malleihin nähden havainnollisempi. Toisaalta dimensiohierarkia ei ole tästä mallista helposti nähtävissä, koska dimension kaikki dimensioattribuutit ovat samassa taulussa. Tätä voidaan pitää tähtimallin haittana. Dimensiohierarkiaa voi havainnollistaa sijoittamalla dimensioattribuutit hierarkian mukaiseen järjestykseen. Toisena haittaavana tekijänä on se, että dimensiot sisältävät paljon redundanttia tietoa, mikä hankaloittaa normalisoimattomien dimensiotaulujen ylläpitoa. Dimension hierarkian syvyys tässäkin mallissa kasvattaa attribuuttien määrää ja leveys vastaavasti rivien määrää, mutta vain dimensioiden osalta. Tällaisen mallin ilmentymätasoon tietoja lisättäessä ensisijaisesti faktataulun ja aikadimension rivien määrä kasvaa. Muiden

dimensioiden rivimäärä ei paljoakaan lisäänny. Dimensioille on luonteenomaista niiden staattisuus. Tämän mallin ilmentymätaso on vähemmän redundanttinen verrattuna normalisoimattomaan kuutioon ja siksi sen ulkoisen muistin tilatarve on pienempi kuin edellisissä universaalirelaation ja normalisoimattoman kuution mukaisissa loogisissa malleissa.

Koska tähän malliin kohdistetuissa kyselyissä taulujen välisiä liitoksia on enintään dimensioiden määrä, se lisää liitosoperaatioiden osalta kyselyn prosessointiaikaa normalisoimattomaan kuutioon verrattuna. Mitä enemmän kyselyssä on dimensioita mukana, sitä enemmän on tauluja FROM –lausekkeessa, ja liitosehtojen ilmaisemista WHERE -rakenteessa. Jokaista SELECT –listassa ilmaistua dimensiota kohden joudutaan tekemään liitosoperaatio faktataulun ja dimensiotaulun välille, jotta halutun hierarkiatason dimensioattribuuttien arvot saadaan valintalausekkeessa käyttöön. Taulujen välisten liitosoperaatioiden määrä lisää kyselyn kokoa. Toisaalta yhdellä liitosoperaatiolla saadaan käyttöön dimension kaikkien hierarkiatasojen dimensioattribuuttien arvot ja niihin liittyvät mahdolliset lisätiedot projektio-operaatiota varten. Dimensiohierarkian syvyys ei tässä mallissa lisää kyselyn rivimäärää.

Tietoja suodattavat operaatiot kuten SLICE ja DICE lisäävät kyselyn kokoa. Tällaisia SQL:n lausekkeita ovat IN-, ja NOT IN- predikaatein varustetut WHERE ehdon lausekkeet. Ns. arvoaluekyselyt (range sum queries) lisäävät kyselyyn lisärivejä. SQL:n BETWEEN –predikaatti on tästä esimerkki. Arvoaluekyselyssä dimensioattribuutin arvojoukosta valitaan halutulla arvovälillä sijaitseva osajoukko [Lee *et al.*, 2000] [Li *et al.*, 2001]. Tällainen kysely soveltuu vain numeerisille dimensioattribuuttien arvoille.

6.4. Lumihiutalemalli

Lumihiutalemalli voidaan johtaa tähtimallista. Tässä loogisessa mallissa on yksi faktataulu ja siihen liittyvät dimensiotaulut. Jokaisen dimensiohierarkiatason dimensioattribuutit ovat omassa taulussaan. Dimension hierarkiatasojen välillä on suhde 1:M, jonka mukaan ylemmän tason dimensioattribuutin arvoon sisältyy yksi tai useampia alemman tason dimensioattribuutin arvoja. Alemman tason relaation vierasavain on sitä ylemmän tason relaation ensisijainen avain. Tässä mallissa kaikki relaatiot ovat normalisoidussa muodossa, jolloin saman tiedon esittäminen on minimoitu. Lumihiutalemallin ilmentymätason päivittäminen ja uusien rivien lisääminen

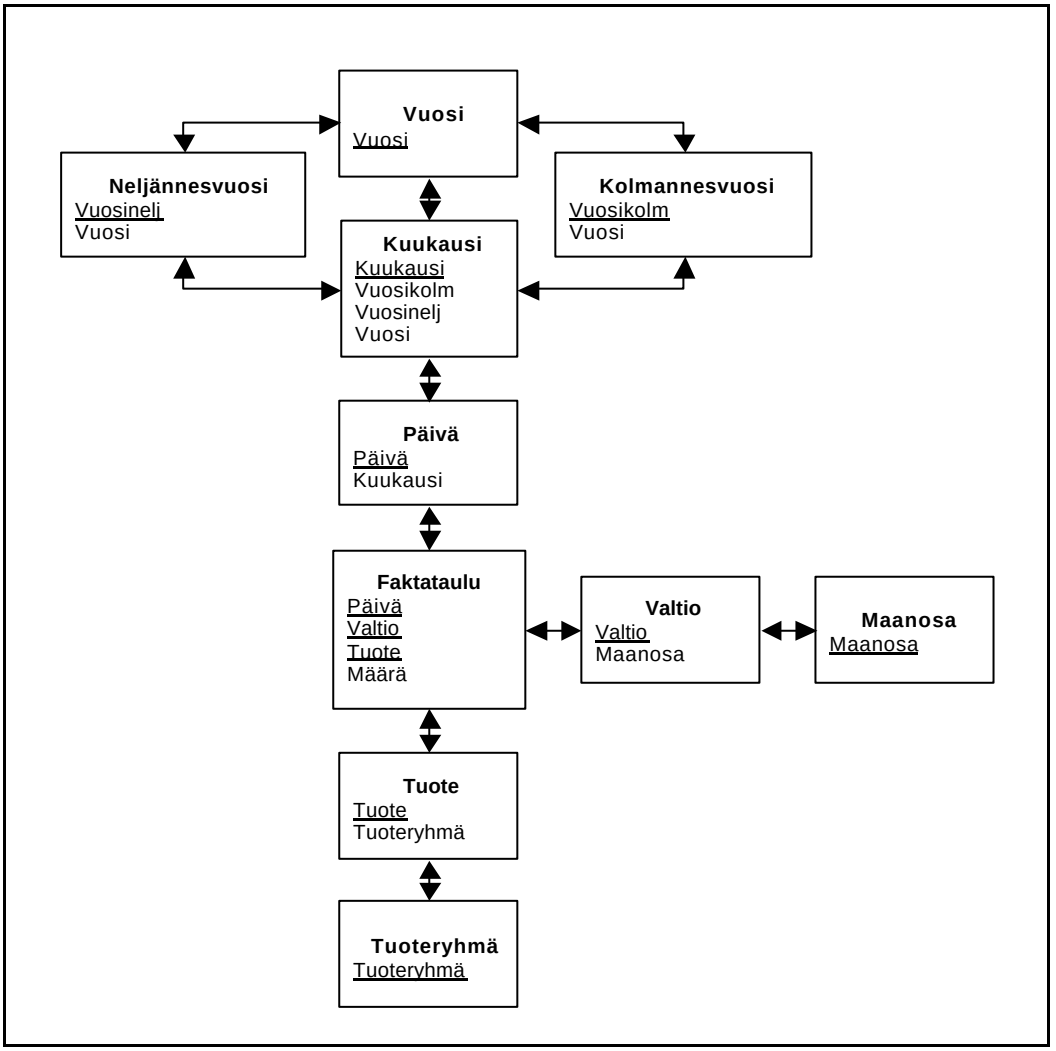
on selkeätä normalisoitujen taulujen ansiosta. Dimensiotaulujen hierarkiatasojen välinen eheys on helppo toteuttaa. Monihierarkkisuus lisää malliin taulun jokaista uutta dimensiotasoa kohti.

Kuvassa 16 esitetään lumihiutalemalli, jossa aikadimensio on monihierarkinen. Aikadimensiossa on seuraavat polut, joita pitkin mitta-attribuuttien arvoja voidaan aggregoida alimmalta hierarkiatasolta ylimmälle:

Päivä > Kuukausi > Vuosi

Päivä > Kuukausi > Vuosikolmannes > vuosi

Päivä > Kuukausi > Vuosineljännes > Vuosi



Kuva 16: Lumihiutalemalli.

Lumihiutalemalli on havainnollinen ja helppo käyttäjänkin käsittää moniulotteisena rakenteena, koska siinä dimensiohierarkia on eksplisiittisesti

kuvattu. Siitä käy selkeästi ilmi mille hierarkiatasoille mitta-attribuuttien arvoja on mahdollista yhdistellä.

Sen heikkoutena voidaan pitää taulujen paljoutta, jonka seurauksena SQL – kyselyssä taulujen välisten liitosoperaatioiden määrä voi kasvaa merkittävästi. Tähän seikkaan vaikuttaa dimensioiden hierarkiatasojen määrä. Jos hierarkiatasoja tai monihierarkisuutta on mallissa runsaasti, se lisää taulujen välisten liitosoperaatioiden määrää erityisesti silloin kun tietojen yhdistely tehdään samanaikaisesti useampien dimensioiden ylemmille hierarkiatasoille. Kun kyselyn valintalista sisältää ylempiä hierarkiatasoja, niiden dimensioattribuuttien arvot saa käyttöön vain hierarkiapolkua pitkin kirjoittamalla kyselyyn liitosoperaatiot tasojen välille alkaen alimmalta tasolta. Siksi kyselystä voi tulla hyvinkin pitkä. Jos kysely sisältää tietoja suodattavia määrittelyjä, kyselyn formulointi saattaa muodostua pitkäksi ja kyselyn tulkinta vaikeutuu. Jos kyselyssä tietoja yhdistellään alimmille dimensioiden hierarkiatasoille tai kyselyssä käytetään vain muutamia dimensioita, kyselyt pysyvät lyhyinä ja selkeinä.

Kyselyn muodostamisen kannalta universaalirelaatio ja lumihiutalemalli ovat samanlaisia. Samanlaisen käsitetason ja ilmentymätason sisältämässä universaalirelaatiossa ja lumihiutalemallissa samanlaisen tuloksen tuottavat kyselyt ovat muodoltaan samanlaisia.

6.5. Konstellaatiomalli

Konstellaatiomalli sisältää kaksi tai useampia faktatauluja ja näihin liittyviä dimensioita, joista jotkut ovat kahdelle tai useammalle faktataululle yhteisiä. Konstellaatiomallissa vähintään yksi dimensio on yhteinen eri faktatauluille. Konstellaatio voi koostua tähtikaavioista, lumihiutalekaavioista tai näiden yhdistelmistä. Tarkasteltavissa konstellaatiomallin muunnelmissa dimensiohierarkia voi sisältyä yhteen tauluun tai se voidaan jakaa lumihiutalemallin mukaisesti niin, että jokainen hierarkiataso esitetään omana taulunaan. Konstellaatiomallissa faktataulut voivat liittyä toisiinsa hierarkkisesti tai ne voi olla itsenäisiä toisistaan riippumattomia.

Moody ja Kortink [2000] kuvaavat konstellaatiomallin, jossa faktataulut liittyvät hierarkkisesti toisiinsa. Tällaisessa tapauksessa ylemmän tason faktataulun mitta-attribuuttien arvoja ei voida esittää alemman tason faktataulun tarkkuudella. Heidän esimerkkimallissaan laskun rivi on hierarkiassa alemmassa faktataulussa ja laskun kokonaissummasta saatava

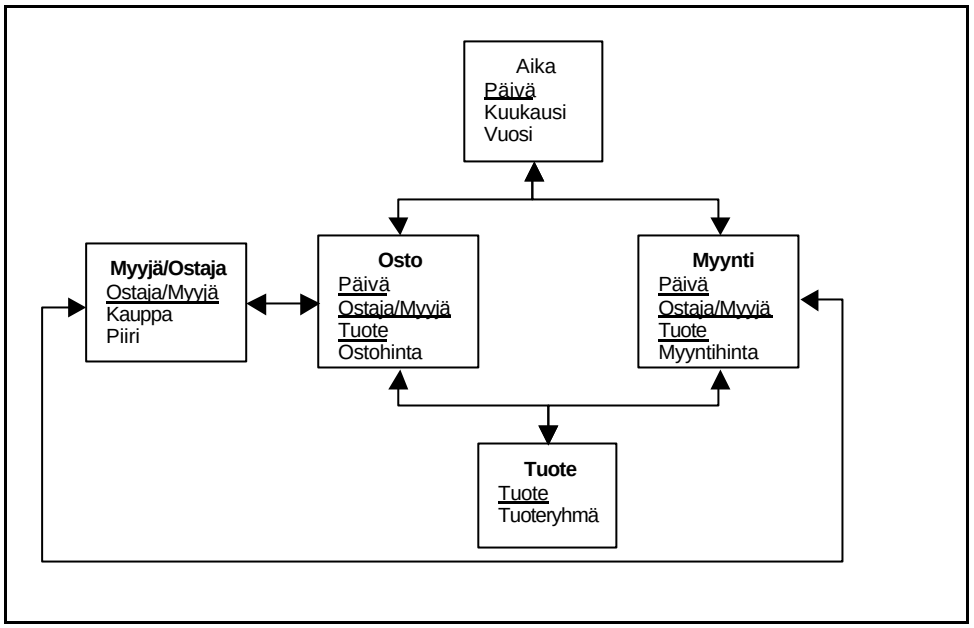
alennus on ylemmässä faktataulussa. Tässä esimerkissä laskufaktataulun ensisijainen avain koostuu laskun riviä esittävän faktataulun osa-avaimesta. Heidän esimerkkitapauksessaan mitta-attribuuttien arvoja voidaan aggregoida ylöspäin molemmista faktatauluista lähtien.

He nimeävät linnunratamalliksi (galaxy) sellaisen tähti- tai lumihiutalemalleista muodostuvan yhdistelmän, jossa faktataulut eivät ole hierarkkisessa suhteessa toisiinsa. Faktatauluja ainoana yhdistävänä tekijänä on tällöin yksi tai useampi yhteinen dimensio. Tässä mallissa kaikki dimensiot liittyvät faktatauluun dimension alimmalla tasolla.

Pokorny ja Sokolowsky [1999] esittävät toisenlaisen ns. faktakonstellaatiomallin, jossa faktatauluja yhdistävinä tekijöinä ovat myös yhteiset dimensiot. Tässä mallissa faktataulu liittyy kuitenkin johonkin dimensiotasoon, joka on alinta tasoa ylempi. Tällainen heidän esimerkkinsä mukainen faktataulu on aggregoitu alimman faktataulun tiedoista ko. dimension ylemmälle tasolle. Tämä vastaa materialisoitua näkymää (materialized view). Tätä mallia ei voi luokitella ainakaan pelkistetyksi konstellaatiomalliksi. Jos konstellaatiomallissa sallitaan materialisoidut näkymät, erilaisia loogisia malleja voidaan esittää lukematon määrä. Materialisoitujen näkymien tarkoituksena on vähentää tietojen yhdistelytarvetta dimension alimmalta tasolta, ja näin nopeuttaa kyselyjen prosessointiaikaa. Kohdealueen mallintaminen tietojen analysointinäkökulmasta ja ennakoitavissa olevien todennäköisimpien kyselyjen tunnistaminen määrää tällaiseen loogiseen malliin toteutusvaiheessa tehtävät materialisoidut näkymät.

Faktakonstellaatiota voidaan pitää aitona konstellaatiomallin muunnelmana, jos dimensiohierarkian alinta tasoa ylempään tasoon liittyvä faktataulu ei ole yhteenveto alimman faktataulun mitta-attribuuttien arvoista.

Kuvassa 17 on esitetty konstellaatiomalliin perustuva esimerkki osto- ja myyntitoiminnasta. Siinä molemmilla toisistaan riippumattomilla faktatauluilla (osto ja myynti) on samat dimensiotaulut (aika, ostaja / myyjä ja tuote). Tässä loogisessa mallissa jokainen dimensio muodostaa oman normalisoimattomassa muodossa olevan taulun.



Kuva 17: Konstellaatiomalli.

Konstellaatiomallin ymmärrettävyyttä ja havainnollisuutta saadaan lisättyä kun dimensiotasot esitetään eksplisiittisesti omina tauluinaan. Tämä malli on luonnollisesti tietosisällön ja perusmallista johdettavien mitta-arvojen yhteenvetotietojen osalta rikkaampi kuin yhden faktataulun sisältävät mallit. Mahdollisten erilaisten kyselyjen joukko on huomattavasti suurempi kuin yhdessä tähti- tai lumihiutalemallissa. Faktataulujen yhteiset dimensiot vähentävät dimensiotaulujen määrää, kun ne esitetään konstellaatiomallissa vain kertaalleen. Tämä taas edellyttää, että yhteisen dimension osalta dimensiohierarkia ja dimensioattribuutit ovat käsitteinä yhtenäistetty. Saman sovelluksen osalta tämä seikka harvemmin on ongelma. Tuotantokäytön alkuvaiheessa tietovarasto sisältää enimmäkseen tähti- tai lumihiutalemallin mukaisia rakenteita. Tietovaraston sisällön laajetessa tulee tarvetta yhdistellä eri sovellusalueiden tietoja. Dimensioihin liittyvien käsitteiden yhtenäistäminen saattaa tulla yllätyksenä, kun tietojen yhdistelytarve kasvaa.

Konstellaatiomallin rakenteessa kyselyjen formulointiin vaikuttavat samat seikat kuin tähti- ja lumihiutalemallissa. Yhdellä taululla esitettynä normalisoimaton dimensio vähentää SQL –lausekkeessa taulujen välisten liitosten määrää ja kyselyn prosessointiaikaa lumihiutalemalliin verrattuna. Kyselyn prosessointiaikaa pyritään minimoimaan mm. normalisoimattomilla relaatioilla ja ennalta aggregoiduilla yhteenvetotiedoilla eli materialisoiduilla näkymillä.

Tähtimallin sisältämässä konstellaatiossa, jossa faktataulun dimensioattribuutti viittaa dimensiohierarkiassa muulle kuin alimmalle tasolle tulee SQL –kyselyn toteutuksessa ongelmia. Tällaisessa tapauksessa liitosoperaation tuloksena kysely palauttaa dimensiotaulusta useita dimensioattribuuttirivejä. Lumihiutalemallissa tätä ongelmaa ei ole koska viittaus tapahtuu dimensiotaulun dimensioattribuuttiin, joka on rivin yksilöivä avain.

6.6. Monikuutiomalli

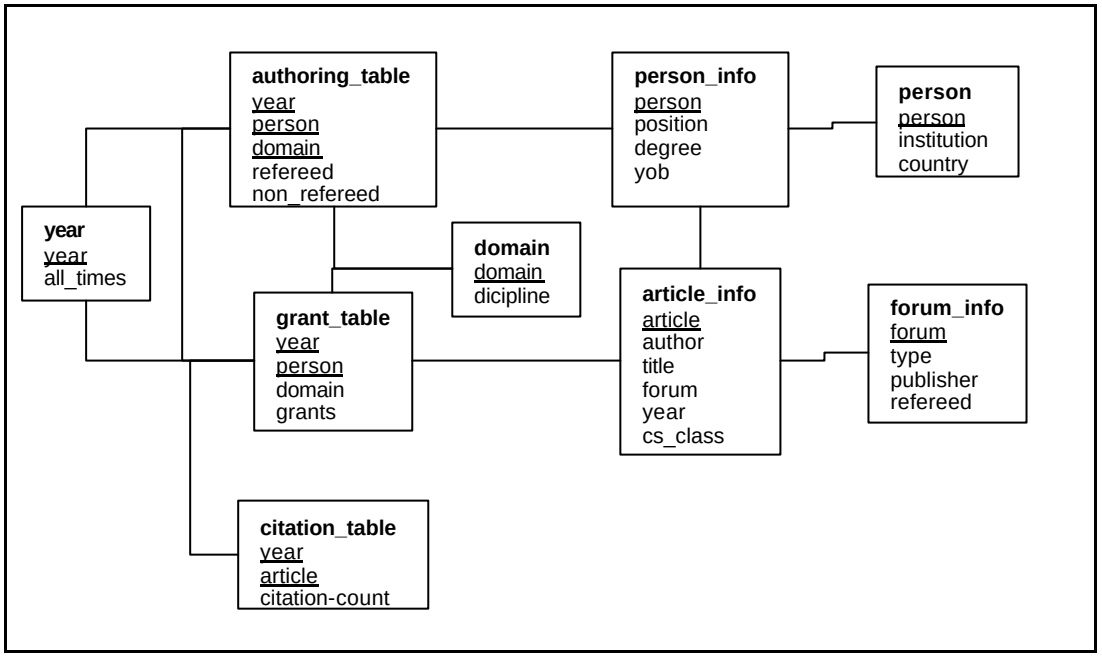
Niemi T. *et. al.* [2003] ovat mallintaneet informetriikkaan liittyvät tiedot moniulotteiseksi tietokannaksi (multidimensional database, MDD). Siinä on piirteitä tähti- ja lumihiutalemallista ja yhtäläisyyksiä konstellaatiomallin kanssa. Tässä mallissa mitta-attribuutteja esiintyy kolmessa taulussa. Siksi käytän siitä nimeä monikuutiomalli erotukseksi konstellaatiomallista.

Monikuutiomallissa on kolme eri taululajia: tietokuutiotaulut (data cubes), dimensiotaulut (dimension tables) ja hierarkkiset taulut (hierarchical tables). Tietokuutiotaulut vastaavat faktatauluja, koska ne sisältävät mitta-attribuutteja ja näihin liittyviä dimensioattribuutteja. Jokainen dimensiotaulu sisältää vain alimman dimensiotason dimensioattribuutit ja ko. dimensiotasoon liittyvät mahdolliset lisätiedot, joiden perusteella myös yhteenvetotietoja saadaan tuotettua. Dimensiohierarkia sisältyy kukin omaan hierarkiatauluunsa, joka vastaa tähtimallin normalisoimatonta dimensiotaulua. Dimensiotauluilla voi olla yhteisiä dimensioattribuutteja, joiden kautta dimensiotaulujen tietoja voidaan yhdistellä myös dimensioiden välillä.

Kuvassa 18 on esitetty monikuutiomalli. Siinä taulut *authoring_table, grant_table* ja *citation_table* ovat tietokuutioita. Dimensiotauluja ovat *person_info, article_info* ja *forum_info.* Taulut *year, person* ja *domain* ovat hierarkiatauluja. Attribuutti *author* ja *person* ovat samoja dimensioattribuuttien arvojen osalta ja tarkoittavat samaa asiaa. Dimensiotauluja voidaan yhdistää toisiinsa esim. attribuuttien *person_info.person* ja *article_info.author* kesken tai *forum_info.forum* ja *article_info.forum* kesken. *Person_info* –taulun attribuutit *position* (asema), *degree* (oppiarvo) ja *yob* (year of birth, syntymävuosi) ovat henkilöön liitettyjä lisätietoja. Toisin kuin muissa edellä esitetyissä loogisissa malleissa käyttäjä voi itse kyselykielellä määritellä omia hierarkiatasoja ryhmittelemällä dimensioattribuuttien arvoja määriteltävien kriteerien perusteella. Monikuutiomalli on toteutettavissa relaatiokaaviona, jolloin kyselyt voidaan tehdä SQL –kielellä. Tähän loogiseen malliin perustuvat kyselyt on tehty

deklaratiivisella ja näkymäorientoituneella kielellä, joka pohjautuu logiikkaohjelmoinnin muuttujakäsitteeseen.

Kun tässä mallissa dimensiotaulut eivät ole lumihiutalemallin mukaisesti täysin normalisoidussa muodossa, kyselyjen muotoilu ei myöskään muodostu pitkäksi lausekejonoksi.



Kuva 18: Monikuutiomalli.

6.7. Loogisten mallien yhteisiä piirteitä

Kohdealueen moniulotteinen looginen mallintaminen tehdään aina kohdealueen keskeisten käsitteiden perusteella, mutta tulokseksi muodostuu aina tietorakenne, jossa on sekä faktoiksi että dimensioiksi luokiteltuja tietoja. Edellä mainituissa keskeisissä perusmalleissa voidaan nähdä yhteisiä piirteitä.

Universaalirelaatiomallissa ja lumihiutalemallissa faktatiedot ovat molemmissa omilla riveillään siten, että mitta-attribuutteihin liittyvät dimensioattribuutit muodostuvat käsitetasolla lähes samanlaisiksi. Näitä kahta mallia yhdistää vielä se seikka, että molemmissa dimensiohierarkia on eksplisiittisesti esillä. Universaalirelaatiossa kahden dimensiohierarkiatason välinen yhteys esitetään samalla rivillä dimensioattribuuttien arvoilla kuten lumihiutalemallissa. On huomattava, että konstellaatiomallin sisältämät dimensioattribuutit ja mitta-attribuutit voidaan muuttaa universaalirelaatiomallin mukaiseksi rakenteeksi.

Normalisoimattomalla kuutiomallilla ja tähtimallilla on yhteistä se, että molemmissa dimensiot ovat normalisoimattomassa muodossa. Erona kuitenkin on se, että normalisoimattomassa kuutiomallissa faktoja edustavat mitta-attribuutit ja dimensiot ovat kaikki samassa taulussa.

Tähti- ja lumihiutalemallit on helppo muuntaa monikuutiomalliksi ja päinvastoin.

Tässä tutkielmassa on tarkasteltu moniulotteisen tietorakenteen tavanomaisimpia loogisia malleja. Materialisoiduilla näkymillä saadaan aikaiseksi erilaisia muunnelmia näistä malleista. Oman erikoisen loogisen konstellaatiomallimuunnelman muodostaa ns. faktakonstellaatio, jossa faktataulu ei ole muodostettu materialisoituna näkymänä.

7. SQL –KIELEN OLAP -LAAJENNUKSET

SQL –kieli on alun alkaen suunniteltu tapahtumakäsittelyyn perustuvien sovellusten kyselykieleksi. Tapahtumakäsittelyssä tietokannan tietojen haku ja ylläpito kohdistuvat useimmiten muutamiin tauluihin. Näissä tauluissa tapahtumakohtaisesti käsiteltävien rivien lukumäärä on vähäinen OLAP –käsittelyyn verrattuna.

OLAP –kyselyssä käsitellään samanaikaisesti useita tauluja. Tietovaraston taulut sisältävät historiatietoa ja siksi niissä on miljoonia rivejä. Haku- ja laskentaoperaatioita tehdään kyselyissä useimmiten suurimmalle osalle taulujen riveistä. Tietovaraston taulujen koko voi kasvaa useiksi teratavuiksi. Huolimatta mittavista tietomääristä OLAP –kyselyn vastinaika tulee olla kestoltaan samaa luokkaa kuin tapahtumakäsittelyssä.

Tässä kappaleessa esitetään SQL –kielen puutteet ja rajoitukset OLAP –käsittelyn kannalta. Lisäksi esitetään erilaisia SQL –kielen laajennuksia, joilla kielen toiminnallisuutta pyritään laajentamaan niin, että se sisältäisi keskeisimmät OLAP –kyselyissä tarvittavat ominaisuudet [Dinter *et al.*, 1998]. Joissakin laajennuksissa kyselyn prosessointiajan minimointiin pyritään erilaisin algoritmein.

7.1. SQL –kielen puutteet ja rajoitukset

Ensisijaisesti tapahtumakäsittelykieleksi suunniteltu SQL ei sisällä moniulotteisessa analysoinnissa tarvittavia ominaisuuksia. Kimball ja Strehlo [1995] ovat esittäneet syitä, miksi päätöksenteon tukijärjestelmien toteutukset epäonnistuvat ja kuinka puutteellinen SQL –kieli on näihin tehtäviin nähden.

Kielellä ei kykene vertaamaan tietyn dimensioattribuutin kahden tai useamman eri arvon suhteen laskettujen mitta-attribuuttien yhteenvetotietoja keskenään Esimerkiksi myytyjen tuotteiden kokonaissumma viime vuodelta ei anna riittävästi informaatiota. Kun käytettävissä on myös sitä edeltävän vuoden vastaavat tiedot, se antaa tärkeätä tietoa yrityksen toiminnassa tapahtuneesta kehityksestä. Myöskään saman dimension eri hierarkiatasojen dimensioattribuuttien arvojen suhteen laskettujen mitta-attribuuttien yhteenvetotietojen vertailu toisiinsa on mahdotonta. Tästä esimerkkinä on tarve verrata tuotteen kuukausimyyntimäärää vastaavan tuotteen vuosimyyntimäärään. Tällainen tietojen vertailu on keskeinen menettely, jolla liiketoimintaa voidaan ymmärtää.

SQL-92 standardin mukaisesta kielestä puuttuu mahdollisuus tehdä rivikohtaisesti laskentaoperaatioita tulosrelaation muodostamisen aikana. Tulosrelaation rivejä ei voi numeroida. Tämä merkitsee, että arvotusta jonkin tietyn mitta-attribuutin osalta ei saada toteutettua. SQL:llä ei saada aikaan tulostetta kymmenen eniten myydystä tuotteesta. Kyselyn tulokseen ei voi laskea liikkuvia aggregaatteja, joita ovat esim. summa ja keskiarvo. Esimerkiksi kuukausittain vuoden alusta kumuloitua tuotteen myyntimäärää on mahdoton tuottaa. Tulosrelaation muodostamisen aikana ei voi laskea välisummia dimensioattribuuttien tasokatkoilla eikä niitä saa lisättyä riveiksi kyselyn tulokseen. Taulukkolaskennasta tuttuja rivi- ja sarakesummia ei saa tulostettua tulosrelaation rivien ja sarakkeiden viimeisiksi yhteenvetotiedoiksi. Mitta-arvojen yhteenvetotietoja ei saada laskettua riittävän monipuolisesti OLAP –käsittelytarpeiden edellyttämällä tavalla. SQL-99 standardin mukana osa näistä puutteista poistuvat.

SQL-99 standardiin on ehdotettu sisällytettäväksi useita analyyttisiä funktioita. Winter [2000b] esittelee SQL-99 –standardiin ehdotettuja uusia funktioita, joista hän mainitsee seuraavat uudet tilastolliset funktiot: korrelaatio (*correlation*), kovarianssi (*covariance*), kumulatiivinen jakauma (*cumulative distribution*), sijaluku tai arvotus (*ranking*), prosentuaalinen osuus (*percentile*), lineaarinen regressio (*linear regression*), keskihajonta (*standard deviation*) ja varianssi (*variance*). Lisäksi on ehdotettu numeerisia funktioita: ceiling palauttaa pienimmän kokonaisluvun, joka on yhtä suuri tai suurempi kuin funktiolle parametrina annettu luku (*ceiling*), floor palauttaa pienimmän kokonaisluvun, joka on pienempi tai yhtä suuri kuin funktiolle parametrina annettu luku (*floor*), luonnollinen logaritmi (*natural logarithm*), neliöjuuri (*square root*), eksponentti (*exponent*) ja potenssi (*power*).

7.2. CUBE ja ROLL-UP

Gray *et al.* [1995] esittävät SQL-kieleen operaatioita CUBE ja ROLL-UP. Nämä ovat yleistyksiä GROUP BY –lausekkeesta sekä ristiintaulukointeja ja välisummia tuottavista operaatioista. CUBE –operaatio sisältyy SQL-99 standardiin.

CUBE –operaatio muodostaa tulosrelaation, joka sisältää aggregoinnin jokaista sarakkeiden potenssijoukon alkiota kohti, ja muodostaa siten kaikki mahdolliset SELECT –lauseessa esiintyvien dimensioattribuuttien kombinaatioihin liittyvät summaukset. Kun SELECT –lausekkeen listassa on N

kappaletta dimensioattribuutteja, niin erilaisia dimensioattribuuttiyhdistelmiä saadaan $2^N - 1$ kappaletta. Oletetaan, että SELECT –lausekkeen lista sisältää kolme dimensioattribuuttia (A, B, C). Ilman CUBE –operaatiota tämä edellyttää seitsemän eri GROUP –BY lausekkeen formulointia, jotka UNION –operaatiolla liitetään yhdeksi kyselyksi. Jos N:n attribuutin arvojoukkojen mahtavuudet (cardinality) ovat C_1, C_2, …, C_N, CUBE –operaatiolla muodostetun tuloskuution mahtavuus on $(C_1 +1)*(C_2+1)* \ldots * (C_N+1)$. Jos esimerkiksi SELECT –listan attribuuttien A, B ja C kardinaliteetit ovat 5, 4 ja 3, tuloskuution kardinaliteetti on 6 * 5 * 4 = 120.

```
SELECT Malli, Vuosi, Väri, SUM(Myynti)
FROM Myynti
GROUP BY Malli, Vuosi, Väri WITH CUBE;
```

Peruskuutio:

	Malli	Vuosi	Väri	Myynti
(1)	Chevy	1990	Sininen	10
(2)	Chevy	1990	Punainen	11
(3)	Chevy	1991	Sininen	12
(4)	Chevy	1991	Punainen	13
(5)	Ford	1990	Sininen	14
(6)	Ford	1990	Punainen	15
(7)	Ford	1991	Sininen	16
(8)	Ford	1991	Punainen	17

Tuloskuutio:

	Malli	Vuosi	Väri	Myynti
(1)	Chevy	1990	Sininen	10
(2)	Chevy	1990	Punainen	11
(3)	Chevy	1990	ALL	21
(4)	Chevy	1991	Sininen	12
(5)	Chevy	1991	Punainen	13
(6)	Chevy	1991	ALL	25
(7)	Chevy	ALL	Sininen	22
(8)	Chevy	ALL	Punainen	24
(9)	Chevy	ALL	ALL	46
(10)	Ford	1990	Sininen	14
(11)	Ford	1990	Punainen	15
(12)	Ford	1990	ALL	29
(13)	Ford	1991	Sininen	16
(14)	Ford	1991	Punainen	17
(15)	Ford	1991	ALL	33
(16)	Ford	ALL	Sininen	30
(17)	Ford	ALL	Punainen	32
(18)	Ford	ALL	ALL	62
(19)	ALL	1990	Sininen	24
(20)	ALL	1990	Punainen	26
(21)	ALL	1991	Sininen	28
(22)	ALL	1991	Punainen	30
(23)	ALL	1990	ALL	51
(24)	ALL	1991	ALL	58
(25)	ALL	ALL	Sininen	52
(26)	ALL	ALL	Punainen	56
(27)	ALL	ALL	ALL	108

Kuva 19: CUBE –operaatio.

Kuvassa 19 myynti –taulu sisältää kahdeksan riviä, josta CUBE –lausekkeella tulosrelaatioon muodostuu 27 riviä ((2+1)*(2+1)*(2+1)=27). Kaikki tulosrelaation rivit, joissa jossakin sarakkeessa on ALL –arvo sisältävät mitta-arvojen yhteenvetotietoja myytyjen autojen määristä. Tulosrelaation rivimäärät muodostuvat dimensioattribuuttien malli, vuosi ja väri osalta seuraavasti:

malli, vuosi		4 riviä
malli, väri		4 riviä
vuosi, väri		4 riviä
malli		2 riviä
väri		2 riviä
vuosi		2 riviä
malli, vuosi, väri	8 riviä	
ei ryhmittelyä minkään suhteen		1 rivi
Yhteensä		27 riviä

CUBE –operaatio aggregoi ensin kaikki SELECT –lausekkeen attribuutit GROUP BY –lausekkeen mukaisesti. Sen jälkeen se muodostaa UNION –operaation mukaan ylemmät yhdistelmärivit. Jotta kuution kaikki mahdolliset rivit saadaan esitettyä tulosrelaatiossa, otetaan käyttöön ALL –arvo. ALL –arvolla saadaan esitettyä jokaisen dimensioattribuuttiin liittyvän hierarkian ylin taso. Tämä tuo SQL –kieleen mutkikkuutta. ALL arvo on merkitykseltään sama kuin NULL- arvo.

Gray *et al.* [1995] ehdottavat, että ALL –arvo korvataan NULL –arvolla, ja otetaan käyttöön GROUPING() –funktio, jonka avulla tehdään ero näiden kahden arvon kesken. Jokaista SELECT –lauseen dimensioattribuuttia kohden lausekkeeseen lisätään GROUPING() –funktio, jonka parametrina on vastaava dimensioattribuutti. Boolean -tyyppinen GROUPING() –funktio palauttaa arvon TRUE, jos SELECT –listan vastaava dimensioattribuuttiarvo on NULL, ja arvon FALSE, jos ko. attribuuttiarvo ei ole NULL.

```
(1)  SELECT Model, Year,Color, SUM(Sales)
(2)          GROUPING(Model),
(3)          GROUPING (Year),
(4)          GROUPING (Color)
(5)  FROM Sales
(6)  GROUP BY Model,Year,Color
(7)  WITH CUBE;
```

Kuva 20: CUBE ja GROUPING -funktio.

Kuva 20 on esimerkki GROUPING() –funktion käytöstä. Lausekkeen kokonaissumman muodostama ylin aggregoitu rivi (ks. kuva19 rivi 27) on muotoa (NULL, NULL, NULL, 108, TRUE, TRUE, TRUE).

Toinen operaatio on ROLL-UP, jolla SELECT -listassa luetelluista attribuuteista muodostetaan kaikki mahdolliset ylempien karkeampien tasojen yhdistelmärivit annetun aggregointifunktion mukaan. ROLL-UP operaatio tulostaa SELECT –listan dimensioattribuuteista mitta-attribuuttien arvojen yhteenvetotiedot välisummina oikealta vasemmalle (esim. kuvan 19 rivit 3, 6, 9, 12 ,15 18 ja 27). Luettelot välisummatasokatkoineen ovat ominaisuuksiltaan lineaarisia. ROLL-UP -operaatio vaatii CUBE –operaatiota vähemmän laskentaa ja on siten näistä kahdesta käyttökelpoisin ratkaisu, kun syöteaineistosta halutaan raporttityyppistä listaa tasokatkoineen ja välisummineen. ROLL-UP –operaatio tuottaa vain välisummat seuraavan mallin mukaisesti:

(f1, f2, …,ALL),
…
(f1, ALL, …,ALL),
(ALL, ALL, …,ALL)

Gray *et al.* [1995] ehdottavat otettavaksi käyttöön ns. decoration –sarakkeet. Nämä sarakkeet eivät esiinny GROUP BY -lausekkeessa ja ne ovat funktionaalisesti riippuvaisia ryhmittelysarakkeista. Jos tällainen decoration –sarake on funktionaalisesti riippuvainen aggregoitavista sarakkeista, se voi esiintyä SELECT –listassa (kuva 21). Decoration –sarake vastaa dimensiohierarkiatasoon liitettyä tasoa kuvaavaa lisäattribuuttia. Nämä lisäävät kuutiosta saatavien näkymien määrää.

```
(1)  SELECT department.name, SUM(sales)
(2)  FROM sales JOIN department
(3)                  USING (department_number)
(4)  GROUP BY sales.department_number
```

Kuva 21: Decorations –sarake.

Kuvan 21 rivin 1 decoration sarake department_name saadaan liittämällä sales- ja department –taulut toisiinsa riveillä 2 ja 3. Molemmissa tauluissa on attribuutti department_number, jonka perusteella osaston nimi saadaan department – taulusta tulosrelaatioon.

CUBE –operaatiolla saadaan tuotettua kuutio, joka on SELECT –listassa esiintyvien dimensiotasojen potenssijoukko, ja sisältää kaikki alkioiden mahdolliset kombinaatiot. CUBE -operaatio vaatii raskasta prosessointia, jos

kyselyssä on paljon dimensioita, kussakin tasossa on paljon dimensioattribuuttien arvoja ja peruskuutiossa on paljon rivejä. Prosessointiaikaa saadaan pienennettyä, jos käytetään esilaskettuja eli materialisoituja näkymiä. Tästä johtuen CUBE -operaatio soveltuu parhaiten tietojen analysointiin peruskuutiosta, jossa on muutamia sarakkeita ja niiden sisältämä arvojoukko on pieni. Analysointiohjelmisto muokkaa CUBE -operaation tuloksena syntyvästä relaatiosta tiedot loppukäyttäjän haluamaan muotoon näytölle. Esim. rivi- ja sarakeyhteissummat ovat tällaista tietoa. ROLL-UP –operaatio on erityisen käyttökelpoinen, kun halutaan perinteistä listamallista tulostetta, jossa mitta-attribuuttien yhteenvetotiedot kumuloidaan dimension alimmilta tasoilta ylimmille tasoille.

7.3. nD-SQL

Gingras ja Lakshmanan [1998] lisäävät SQL –kieleen ominaisuuksia, joiden avulla kysely voi kohdistua useampaan kuin yhteen relaatiotietokantaan. Samalla he lisäävät SQL –kieleen ominaisuuksia, joilla käyttäjä saa muotoiltua mitta-arvojen yhteenvetotietoja joustavammin ja vapaammin kuin mihin CUBE – ja ROLL-UP –operaatiot antavat mahdollisuuden. Tässä kuvauksessa keskitytään esittämään OLAP –ominaisuuksiin ehdotettuja laajennuksia.

nD-SQL:ssä keskeisiä käsitteitä ovat mm. relaatio- ja dimensiomuuttujat. Relaatiomuuttuja sisältää kaikki tietokantakaavion relaatiot ja dimensiomuuttuja yksittäisen relaation kaikki attribuutit. Esimerkeissä käytetään seuraavaa relaatiota:

nyse::prices(Date, Ticker, Measure, Price)

```
(1)  SELECT X, SUM(T.Price)
(2)  FROM nyse::prices T, DIM  X
(3)  GROUP BY X
```

Kuva 22: Kuva 22: nD-SQL –ilmaisu: aggregointi eri dimensioilla.

Kuvan 22 kysely kohdistuu tietokantaan nimeltä nyse ja sen yhteen tauluun prices (rivi 2). Rivillä 1 dimensiomuuttuja X sisältää relaatiomuuttujan T dimensioattribuutit Date, Ticker ja Measure. Mitta-arvoattribuutti price sisältyy myös muuttujaan T. Tämä kysely vastaa kolmea GROUP BY –lauseketta, joissa price –mitta-attribuutin arvo aggregoidaan kunkin dimensioattibuutin suhteen.

```
(1)  SELECT (AVG(T.Price) AS Y FOR Y) AS X FOR X
(2)  FROM nyse::prices T, DIM X, Y
(3)   WHERE DIMS IN {T.date, T.measure, T.ticker}
(4)  GROUP BY X, Y
```

Kuva 23: nD-SQL –ilmaisu: kahden dimension mahdolliset yhdistelmät sovellettuna mitta-attribuuttiin.

Kuvan 23 kysely on esimerkki monirakenteisesta aggregointikyselystä. Rivin 2 FROM lausekkeessa nyse::prices T tarkoittaa, että nyse -nimisen tietokannan prices –relaation price mitta-attribuutista tehdään yhteenvetotiedot (rivi 1). T on ko. relaation relaatiomuuttuja ja X ja Y ovat dimensiomuuttujia. Rivillä 3 varatulla sanalla DIMS IN esitetään T –relaatiomuuttujan sisältämät dimensioattribuutit, joiden suhteen price –mitta-attribuutin arvoista lasketaan keskiarvot. Kysely generoi keskihinnan kahden dimension kaikille mahdollisille yhdistelmille, jotka ovat muodostettavissa dimensiojoukosta date, measure ja ticker.

```
(1)  SELECT W, X, Y, Z, SUM(G)
(2)  FROM db::rel T, DIM W, X, Y, Z
(3)  WHERE W < X < Y < Z AND W IN{A, B, C} AND
(4)            X IN {A, B, C} AND Y IN{C, NONE} AND
(5)           Z IN{D, E, F, NONE}
```

Kuva 24: nD-SQL ja kysely: "lähimmät naapurit".

Kuvan 24 esimerkkikysely kohdistuu relaatioon db::rel(A, B, C, D, E, F, G). Esimerkissä G –attribuutti on mitta-attribuutti ja muut attribuutit ovat dimensioattribuutteja. Rivin 3 kohta W < X < Y < Z asettaa dimensioattribuutit nimen mukaiseen järjestykseen (A, B ,C D, E F) tulosrelaatiossa. Rivin 4 varattu sana NONE vastaa ALL –arvoa, jolloin SELECT –lauseen kolmannesta dimensioattribuuttimuuttajasta Y (dimensioattribuutti C) lasketaan yhteenvetotiedot ylöspäin kahden dimensioattribuutin osalta. Kysely tuottaa yhteenvetotiedot dimensioattribuuttien A, B ja C ”lähimmistä naapureista”. Dimensioattribuuttien A, B, C –ryhmittelyn lisäksi kysely laskee mitta-attribuutin summan ryhmille {A, B, C, D}, {A, B, C, E}, {A, B, C, F}, {A, B}, {A, C} ja {B, C}.

nD-SQL perustuu relaatioalgebran laajennukseen, jota he kutsutaan uudelleen strukturointialgebraksi (Restructuring Relational Algebra, RRA). RRA koostuu

RA -operaatioista ja uudelleen strukturointioperaatioista. nD-SQL –kysely käännetään RRA:n mukaiseen muotoon ja siitä edelleen syötteeksi SQL –kääntäjälle.

SQL:n ominaisuuksia on laajennettu siten, että kysely voi kohdistua tietokantojen yhdistelmiin (federation of databases). Tutkijat toteavat, että tavoitteena pitää olla yksi tietovarasto, johon tiedot on yhdistelty haluttuun muotoon. Moniulotteisessa OLAP –käsittelyssä kyselyt harvoin kohdistuvat suoraan operatiivisiin järjestelmiin, joten tietojen haku yhdistetyistä tietokannoista on harvinaista. nD-SQL:llä on mahdollisuus saada aikaan samanlaisia tuloksia kuin CUBE – ja ROLL-UP –operaatioilla [Gray *et al.* ,1995]. nD-SQL:n laajennusten hyvänä puolena on , että ne sisältävät useampia OLAP –käsittelyyn liittyviä operaatioita. Kyselykielen ilmaisut ovat tiiviitä ja ilmaisuvoimaisia, ja niissä käyttäjä voi rajoittaa kyselyn tulosta melko vapaasti. nD-SQL:stä puuttuu dimensiohierarkian käsittely ja mahdollisuus käyttää useampia faktatauluja samassa kyselyssä.

7.4. Extended Multi-Feature SQL, EMF SQL

Johnson ja Chatziantoniou [1999] ovat kehittäneet SQL –kieleen OLAP –toiminnallisuutta uusilla lausekkeilla. Chatziantoniou [1999b] esittelee PanQ -työkalun, jolla EMF SQL –kyselyjä voidaan toteuttaa. Keskeiset SQL:n syntaksiin tehdyt laajennukset ovat: GROUP BY –lausekkeen ryhmämuuttuja, SUCH THAT –lauseke, SELECT –lauseke ja HAVING –lauseke.

EMF SQL –kielessä käytetään käsitettä ryhmämuuttuja (grouping variable) [Chatziantoniou, 1999a]. GROUP BY –lausekkeeseen voidaan lisätä puolipisteellä alkaen ryhmämuuttujia, jotka voivat esiintyä myös SELECT lausekkeessa. Ryhmämuuttuja GROUP BY –lauseessa esitetään seuraavan syntaksin mukaisesti:

GROUP BY dimensionattribute ; X_1, X_2, …,X_n

Toinen SQL –kieleen tehty laajennus on SUCH THAT –lauseke. Sillä voidaan määritellä ryhmämuuttujien sallimien arvojen vaihteluvälejä. WHERE –lausekkeen kaltaisessa SUCH THAT –lausekkeessa esitetään ryhmämuuttujien ehdot. Se on muotoa:

SUCH THAT C_1, C_2, …,C_n

Kukin C_i on ehto, jolla määritellään ryhmämuuttuja X_i (i = 1, 2, …, n). Tämän lausekkeen avulla voidaan tehdä hyvinkin kompleksisia ja ilmaisuvoimaisia kyselyjä.

SELECT –lausekkeessa on mahdollista ilmaista ryhmämuuttuja-attribuutteja ja aggregaatteja.

HAVING –lauseke voi sisältää ryhmämuuttujissa esiintyviä aggregaatteja.

Seuraavassa esitetään muutamia esimerkkejä EMF SQL –kyselyistä. Esimerkeissä käytetään taulua [Chatziantoniou, 1999b]:

Sales(cust, prod, state, day, month, year, sale)

Sales –taulun dimensioattribuutteina ovat cust (asiakas), prod (tuote), state (valtio), day (päivä), month (kuukausi) ja year (vuosi). Mitta-attribuuttina on sale (myyntimäärä).

```
(1)  SELECT cust, avg(x.sale), avg(y.sale), avg(z.sale)
(2)  FROM sales
(3)  WHERE year = 1997
(4)  GROUP BY cust; x, y, z
(5)  SUCH THAT x.cust = cust AND x.state =  “NY”
(6)                y.cust = cust AND y.state = “CT”
(7)                z.cust = cust AND z.state = “NJ”
```

Kuva 25: EMF SQL –kysely: Riviryhmien muuttaminen sarakkeiksi.

Kuvan 25 kysely tuottaa asiakkaittain myynnin keskiarvot vuodelta 1997 kolmessa eri osavaltiossa NY, CT ja NJ. GROUP BY –lausekkeessa (rivi 4) x, y ja z ovat ryhmämuuttujia. Riveillä 5, 6 ja 7 määritellään ryhmämuuttujien ehdot ,joilla dimensioattribuutin (state) arvojoukosta sallitaan vain tietyt arvot (NY, CT ja NJ) ryhmämuuttujiksi ja tuloksen sarakkeiksi. Ryhmämuuttujille lasketaan asiakaskohtainen keskiarvo SELECT lauseessa (rivi 1). Ryhmämuuttujilla ja niiden sisällön määrittelevillä ehdoilla dimensioattribuutin arvojoukosta voidaan muodostaa myös osajoukkoja tulosrelaation sarakkeiksi.

```
(1)  SELECT prod, month, sum(x.sale) / sum(y.sale)
(2)  FROM sales
(3)  WHERE year = 1997
(4)  GROUP BY prod, month; x, y
(5)  SUCH THAT x.prod = prod AND x.month =  month
(6)                y.prod = prod
```

Kuva 26: EMF SQL –kysely: Dimensiohierarkia-aggregoinnin ilmaiseminen.

Kuvan 26 kysely laskee vuoden 1997 tuotteiden kuukausittaisen myynnin prosentuaalisen osuuden suhteessa tuotteen vuoden kokonaismyyntiin. Rivin 1 sarakkeen sum(x.sale) sisältö määräytyy rivin 5 ehdon mukaan, jossa aggregointi tehdään tuotteittain ja kuukausittain. Tämä mitta-arvon yhteenvetotieto jaetaan rivin 6 ehdon mukaan määräytyvällä tuotteen kokonaismyynnin arvolla. SQL –toteutuksena tämä edellyttäisi useita kyselyjä vastaavan tuloksen tuottamiseksi.

```
(1)  SELECT cust, prod, avg(x.sale), avg(y.sale)
(2)  FROM sales
(3)  GROUP BY cust, prod; x, y
(4)  SUCH THAT x.cust = cust AND x.prod = prod
(5)              y.cust < > cust AND y.prod = prod
```

Kuva 27:EMF SQL –kysely: Dimension sisäinen vertailu.

Kuvan 27 kysely näyttää asiakkaittain ja tuotteittain asiakkaalle myydyn tuotteen myynnin keskiarvon ja muille asiakkaille myydyn vastaavan tuotteen myynnin keskiarvon. Rivillä 1 sarakkeet cust, prod ja avg(x.sale) sisältävät asiakkaalle myydyn tuotteen keskimääräisen myyntiarvon. Sarake avg(y.sale) sisältää rivin 5 ehdon mukaan kaikkien muiden asiakkaiden vastaavan tuotteen keskimääräisen myyntiarvon.

Seuraavassa kyselyesimerkki liittyy useampaan kuin yhteen faktatauluun. Kyselyesimerkeissä käytetään seuraavia tauluja:

Purchases(account, prod-cat, day, month, year, amount)

WebLog(account, webSite, type, day, month, year, length)

Purchase –taulu (hankinnat) sisältää dimensioattribuutit: account (tili), prod_cat (tuoteluokka), day (päivä), month (kuukausi) ja year (vuosi). Mitta-attribuuttina on amount (ostetun tuotteen määrä). WebLog –taulu (weblogi) sisältää websivuilla käynneistä logitiedoston, jossa ovat dimensioattribuutit: account (tili), webSite (webpaikka), type (webpaikan tyyppi) ja length (webpaikassa käytetty aika).

```
(1)  SELECT account, month, type
(2)  FROM WebLog
(3)  GROUP BY account, month, type; X(Purchase), Y(Purchase), Z, Q
(4)  SUCH THAT X.account = account AND X.month = month,
(5)               Y.account = account,
(6)               Z.account = account AND Z.month = month AND Z.type= type,
(7)               Q.account = account AND Q.month = month
(8)  HAVING avg(X.amount) > avg(Y.amount) AND avg(Z.length) > avg(Q.length)
```

Kuva 28: EMF SQL –kysely sisältäen useita faktatauluja.

Kuvan 28 melko kompleksisessa kyselyssä tulostetaan asiakas, kuukausi ja website –tyyppi, kun asiakkaan kuukauden keskimääräinen ostomäärä on suurempi kuin vuosittainen keskimääräinen ostomäärä, ja käyttäjän website:lla kuukaudessa käyttämän ajan (length) keskiarvo on suurempi kuin asiakkaan käyttämä keskimääräinen aika ko. kuukautena. FROM –lausekkeessa voi esiintyä vain yksi faktataulu. Jos ryhmämuuttujan vaikutusalue liittyy muihin tauluihin kuin FROM –lausekkeessa esiintyvään tauluun, nämä faktataulut täytyy esitellä ryhmämuuttujien määrittelyssä kaarisuluin (rivi 3). Purchase –taulun sarakkeet on X –ryhmämuuttujan käytettävissä. Riveillä 4-7 määritellään ryhmämuuttujien sisältö ja rivin 8 ehdon mukaan tiedot suodatetaan tulosrelaatioon.

```
( 1)  SELECT prod-cat, MAX(SUM(X.amount))
(2)   FROM purchases
(3)   GROUP BY prod-cat
(4)   SUCH THAT [ (X.prod-cat 0 prod-cat AND X.month = month) GROUP BY month; X]
```

Kysely tuottaa seuraavanlaisen tuloksen:

prod-cat	MAX(SUM(X.amount))	month	SUM(X.amount)
Shoes	22	1	22
		2	17
		3	12
Socks	19	2	15
		4	19
Coats	8	5	8
Hats	4	6	4

Kuva 29: EMF-SQL –kysely: Sisennetty aggregointi.

Kuvassa 29 on sisennetty aggregointikysely. Sisennetty aggregointi ilmaistaan SUCH THAT –lausekkeen sisältämällä GROUP BY –lausekkeella alku- ja loppuhakasulkujen välissä (rivi 4). Ilmaisua rajoittaa se, että GROUP BY –lausekkeen attribuutit saa esiintyä vain yhdessä taulussa. Kysely tulostaa

tuotteittain tuotetta eniten myydyn kuukauden myyntimäärän ja kuukausittaisen myynnin yhteismäärän.

Chatziantoniou *et al.* [2001] ehdottavat otettavaksi käyttöön GROUP BY - ja CUBE -lausekkeita korvaavan ANALYZE BY –lausekkeen, jonka syntaksi EMF -kielessä on seuraava:

ANALYZE BY <ryhmittelyoperaatio | taulun nimi> (attribuuttilista)

Ensimmäinen argumentti eli ryhmittelyoperaatio tai taulun nimi voi olla jokin operaatio (esim. GROUP BY, CUBE BY, UNPIVOT, ROLL-UP, GROUPING SETS), taulu (table) tai näkymä (view). Se voi olla mikä tahansa lauseke, joka palauttaa tuloksena taulun.

```
(1)  SELECT prod, month, state, SUM(sale)
(2)  FROM Sales
(3)  ANALYZE BY cube(prod, month, state)

(4)  SELECT prod, month, SUM(sale)
(5)  FROM Sales
(6)  ANALYZE BY UNPIVOT(prod, month,state)

(7)  SELECT prod, month, state, SUM(sale)
(8)  FROM Sales
(9)  ANALYZE BY T(prod, month, state)
```

Kuva 30:Esimerkkejä EMF -kielen ja ANALYZE –lausekkeen käytöstä.

Kuvassa 30 on kolme esimerkkiä lausekkeen käytöstä. Rivien 1-3 kyselyllä saadaan samanlainen tulos kuin korvaamalla rivi 3 lausekkeella CUBE BY prod, month, state. Riveillä 4-6 esiintyvä kysely tuottaa halutun osan tai osakuution CUBE BY –lausekkeen tuottamasta kuutiosta. UNPIVOT –vastaa tässä kyselyssä GROUPING SETS –lauseketta (kuva 39). Tämä kysely tuottaa tulosrelaatioon kolmenlaista yhteenvetotietoa myynnin summasta: rivit myynnin summasta tuotteittain, rivit myynnin summasta kuukausittain ja rivit myynnin summasta osavaltioittain. Rivien 7-9 kyselyssä sales –taulusta on valmiiksi laskettu tietyt mitta-arvojen yhteenvetotiedot tauluun T (rivi 9). Tässä kyselyssä faktatauluina käytetään peruskuutiota (sales) ja materialisoitua näkymää (T) ja sillä tuotetaan tuotteittain, kuukausittain ja osavaltioittain myynnin yhteissummat.

EMF –kielen laajennukset ovat monipuolisia. Syötetaulun riveistä voidaan muodostaa tulostauluun sarakkeita. Samasta dimensiohierarkiasta saadaan eri tason aggregointeja tulossarakkeiksi. Kyselyssä voidaan tehdä dimension

sisäisiä vertailuja. Kysely voi käsitellä useita faktatauluja. Sisennetty aggregointi antaa mahdollisuuden tehdä monipuolisia tuloksia. Lisäksi ANALYZE BY –lauseke yleistää toimintoja kuten operaatiot GROUP BY, CUBE ja ROLL-UP.

7.5. SQL/MX

NonStop SQL / MX on tiedonhallintajärjestelmä, joka on lähinnä suunniteltu käytettäväksi tietojen louhintaan [Clear *et al.*, 1999]. Kyselyjen käsittelyn suorittava tietokantamoottori on kehitetty tehokkaaksi. Tietojen louhinta on voimakkaasti tietointensiivistä prosessointia, jossa raskaat kyselyt toteutetaan tietokantapalvelimella ja tulosten visualisointi käyttäjän työasemalla. SQL –kieleen on lisätty primitiivejä, joilla kyselyn prosessointi suoritetaan yhdellä taulun läpikäynnillä. Ilmentymätasoltaan suurien taulujen käsittelyssä prosessointiaika on siksi lyhyempi kuin perinteisellä SQL:llä tehty kysely, jossa samaan tauluun joudutaan tekemään useita liitosoperaatioita. Tietojen louhintaa edesauttavat mm. seuraavat kieleen lisätyt primitiivit: vaihdos (transpose), otanta (sampling) ja peräkkäisfunktiot (sequence functions).

TRANSPOSE –operaatio muodostaa yhdestä relaation rivistä useita rivejä tulosrelaatioon. Tällä operaatiolla samasta taulusta yhdellä kyselyllä voidaan tulostaa esim. haluttujen sarakkeiden arvojen frekvenssit. Tämä edellyttäisi tavanomaisella SQL:llä jokaista taulun saraketta kohden yhden kyselyn.

```
(1)  SELECT attr_id, attr_val, count(*)
(2)  FROM customer
(3)  TRANSPOSE (1, ACCT_STATUS), (2, GENDER)
(4)    AS (attr_id, attr_val)
(5)  GROUP BY attr_id, attr_val
(6)  ORDER BY attr_id, attr_val;
```

Kuva 31: Esimerkki SQL / MX:n TRANSPOSE –operaatiosta.

Kuvan 31 kyselyssä TRANSPOSE -operaatio tekee taulun (customer) jokaista riviä kohden useita rivejä tulosrelaatioon. Riveillä 3-4 TRANSPOSE –operaatio asettaa arvon 1 muuttujaan attr_id ja dimensioattribuutin ACCT-STATUS –arvon muuttujaan attr_val. Vastaavasti arvo 2 sijoitetaan muuttujaan attr_id ja dimensioattribuutin GENDER –arvo muuttujaan attr_val. Näin saadut välitulokset ryhmitellään (rivi 5) ja lajitellaan (rivi 6) attr_id - ja attr_val –muuttujien mukaan. Tulostaulu sisältää arvon 1 ja yksiselitteisen ACCT_STATUS –attribuutin arvon, frekvenssiluvun ja vastaavat tiedot GENDER –attribuutista.

Otantaoperaatiolla SAMPLE tiedot voidaan poimia otantaan useilla eri tavoilla. Otanta sisältää viisi eri ominaisuutta: otantatyyppi, otantamenetelmä, otantasyöte, otantasuhde ja otantakoko. Esimerkiksi prosentin suuruinen otanta voidaan toteuttaa hajautetulla haulla.

Peräkkäisfunktioilla voidaan laskea liikkuvan ikkunan (moving window) tiedot ja jatkuvat yhteenvetotiedot (running aggregates). Liikkuvalla ikkunalla tarkoitetaan window –funktion (ks. kohta 7.6.2) kaltaista operaatiota, jolla tulosrelaation kullekin riville voidaan laskea ja lisätä sarakkeena yhteenvetotieto tietyn määrän ko. riviä edeltävien tai sitä seuraavien rivien muodostamasta ryhmästä rivejä.

```
(1)  SELECT account, history_month
(2)    MOVINGAVG(balance, rows
(3)  SINCE(THIS(account) < > account), 3)
(4)  FROM customer_account_history
(5)  SEQUENCE BY account, history_month;
```

Kuva 32: Esimerkki SQL / MX:n peräkkäisfunktion soveltamisesta.

Kuvan 32 kyselyssä tulostetaan asiakkaan tilin kolmen kuukauden liikkuva keskiarvo tilin ja kuukauden mukaisessa järjestyksessä. Syötetaulu customer_account_history (rivi 4) sisältää kuukausittain tallennetut asiakkaan tilin saldotiedot (balance). MOVINGAVG –funktio (rivi 2) laskee asiakkaan tilin saldon keskiarvon kolmen kuukauden ajalta SELECT –lausekkeen sarakkeen historiakuukaudelta (history_month) ja sitä edeltäviltä kahdelta kuukaudelta. Rivillä 3 ilmoitetaan ikkunan koko (window size) parametrivakiolla 3. Jos asiakkaalla on tilin kuukausitietoja tallennettu vähemmän kuin kolme, keskiarvoon lasketaan tiedot kahden tai yhden kuukauden ajalta. Rivin 3 SINCE –funktiolla määritellään mistä alkaen ja miltä ajalta asiakkaan tilin kuukauden saldoista keskiarvot lasketaan. Esimerkki vastaa myöhemmin esitettävää window –funktiota (ks. kohta 7.6.2).

SQL / MX:n kehityksen lähtökohtana on ollut ensisijaisesti tietojen louhintaan liittyvien operaatioiden kuten otanta (SAMPLE) ja vaihdos (TRANSPOSE) toteuttaminen. Näistä kolmesta uudesta ominaisuudesta peräkkäisfunktiot ovat lähinnä OLAP –laajennuksia. Näitä ovat liikkuvat yhteenvetotiedot (moving aggregates) ja ikkunarakenne (window construct). SQL / MX:stä puuttuu paljon keskeisiä OLAP –ominaisuuksia, kuten CUBE- ja ROLL-UP -operaatiot.

7.6. Analyyttiset funktiot

SQL –kielestä puuttuu mahdollisuus tehdä analyyttisiä laskentoja [Bellamkonda *et al.*,2000]. Analyyttisessä laskennassa käytettäviä funktioita ovat mm. liikkuvat keskiarvot (*moving averages*), kumulatiiviset summat (*cumulative sums*), arvotukset (*ranking*), prosenttiosuudet (*percentiles*) ja ryhmän ensimmäisen ja viimeisen alkion käsittelyt (*lead/lag functions*). Kaikille näille funktioille on yhteistä, että niissä käsitellään järjestettyjä tietojoukkoja. SQL-99 standardi sisältää näistä jo osan. Näillä laajennuksilla voidaan välttää monivaiheisen (multi phase sql) SQL –kielen käyttäminen, jossa useilla peräkkäisillä kyselyillä muodostetaan välituloksia aputauluiksi tai näkymiksi, ja näitä yhdistelemällä saadaan haluttu lopputulos.

Analyyttiset laskennat voidaan luokitella seuraavasti:

1. Rank –funktiot: Rank –funktiot arvottavat tiedot jollakin perusteella. Näillä saadaan vastaus esim. kyselyyn ”hae myyntialueen 10 eniten myynyttä myyntimiestä”.
2. Window –funktiot: Nämä funktiot toimivat ns. liikkuvalla ikkunalla, joka sisältää rivejä tietyltä aikajaksolta. Tällaisia kyselyjä ovat esim. ”tulosta tätä viikkoa edeltävien 13 viikon ajalta varaston arvon liikkuva keskiarvo”. Näillä funktioilla voidaan laskea liikkuvia tai kumulatiivisia aggregointeja.
3. Raportointifunktiot: Näillä funktioilla verrataan dimensiohierarkian eri tasojen mitta-attribuuttien yhteenvetotietoja toisiinsa. Tyypillinen esimerkki tästä on kysely ”hae alueittain kaupungit, joissa myyntimäärä on vähintään 10% tarkastelun kohteena olevan alueen myyntimäärästä”.
4. Ensimmäinen/viimeinen –funktiot (lag/lead functions): Näiden funktioiden avulla mistä tahansa taulun rivistä saadaan käsiteltäväksi muita saman taulun rivejä tekemättä tauluun itseensä liitosoperaatioita. Kysely ”hae asiakkaan tilin tämän kuukauden saldon ja sitä edeltävän kuukauden saldon erotus” on tästä esimerkki.
5. Käänteinen jakauma (inverse distribution) ja first/last -funktiot: Käänteinen jakauma antaa mm. mahdollisuuden laskea mediaanin. First/last –funktioiden avulla lasketaan yhteenvetotiedot järjestetyn joukon ensimmäiselle ja viimeiselle arvolle. Tästä on esimerkkinä kysely ”hae jokaisen vuoden ensimmäisen kuukauden tilin saldon keskiarvo”.

Analyyttisissä funktioissa SQL –syntaksi on seuraavanlainen [Bellamkonda *et al.*,2000]:

```
function(<arguments>) OVER
    ([<partition by clause>]
     [<order by clause>
     [<aggregate group clause>]])
```

Funktiomäärittelyssä lausekkeet <partition by clause> ja <order by clause> määrittelevät kyselyn dimensioattribuuttien arvojen osajoukot ja näiden järjestyksen. Ensimmäinen näistä jakaa arvot joukkoihin, jotka sen jälkeen lajitellaan toisen lausekkeen mukaiseen järjestykseen. <aggregate group clause> -lauseke valitsee järjestetystä joukosta osajoukon, jonka jokaiselle riville määritellään päätepisteet. Päätepisteellä varustettua riviä sanotaan nykyiseksi riviksi (current row). Esimerkki <aggregate group clause> -rakenteesta on ilmaus ROWS 1 PRECEDING, joka määrittelee ”fyysisen” ikkunan koon (ks. kuva 35), ts. se sisältää edellisen ja nykyisen rivin. Vastaavasti lauseke ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING määrittelee kolme riviä sisältävän ”fyysisen” ikkunan. Tämän ikkunan keskusta on nykyinen rivi ja se sisältää nykyisen rivin, sitä edeltävän ja seuraavan rivin. Lauseke RANGE INTERVAL ’1’ MONTH PRECEDING määrittelee loogisen ikkunan, joka sisältää kaikki rivit edellisen kuukauden ja tämänhetkisen kuukauden väliltä. Kun fyysinen tai looginen ikkuna on määritelty sen sisältämään joukkoon sovelletaan aggregointifunktiota.

Seuraavat esimerkkikyselyt perustuvat taulukaavioon:

```
salesTable (region, state, product, salesperson, date,sales)
```

Tauluun on rekisteröity jokainen myyntimiehen myyntitapahtuma. Attribuutit region (alue), state (valtio), product (tuote), salesperson (myyjä) ja date (päiväys) ovat dimensioattribuutteja. Sales (myyntimäärä) on mitta-attribuutti.

7.6.1. Rank –funktiot

Rank –funktioissa kullekin tulostettavalle riville liitetään luku, joka kuvaa arvojärjestystä muihin riveihin nähden. Rivit lajitellaan jonkin dimensioattribuutin ja mitta-attribuutin laskettujen arvojen mukaan nousevaan tai laskevaan järjestykseen.

```
(1)   SELECT * FROM
(2)    (SELECT region, salesperson, SUM(sales) sum_sales
(3)          RANK() OVER
(4)             (PARTITION BY region
(5)              ORDER BY sum(sales) DESC) rank
(6)      FROM salesTable
(7)      GROUP BY region, salesperson)
(8)    WHERE rank <= 3;
```

Kuva 33: Esimerkki RANK –funktion soveltamisesta.

Kuvan 33 kyselyssä haetaan kolmen eniten myyneen myyjän (salesperson) myyntisummat (sum_sales) alueittain (region) rivin 2 mukaisesti. Rivillä 4 taulu ositetaan alueittain (region), ja rivillä 5 kukin alueen muodostama ryhmä lajitellaan laskevaan järjestykseen myyntisumman perusteella. Rivillä 7 tiedot ryhmitellään uudelleen alueen ja myyntimiehen mukaan. Rivin 8 lauseke suodattaa tulosrelaatioon kustakin alueesta vain kolme eniten myynyttä myyjää. Tulostettavassa luettelossa eniten myyneen myyjän rank –arvo on 1, ja seuraavaksi eniten myydyllä vastaava arvo on 2 jne. . Muita rank –funktioita ovat mm. rivinumerointi ja arvojärjestys prosentuaalisen osuuden mukaan.

7.6.2. Window -funktiot

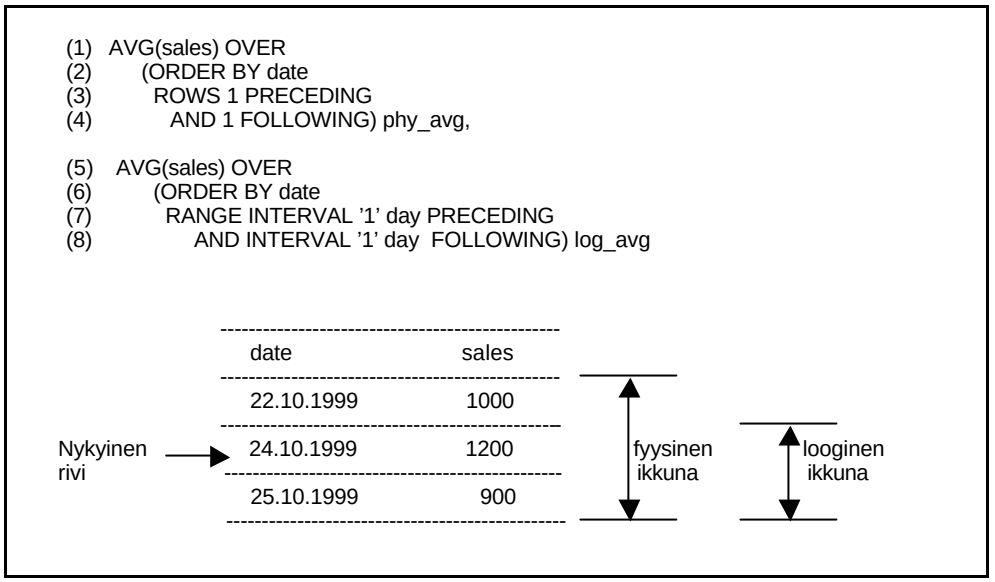
Window –funktio tuo liikkuvan ikkunan, joka sisältää joukon rivejä ositetusta joukosta. Avainsana ROW märittelee fyysisen ikkunatyypin, jossa fyysisiä siirtymiä käsitellään eteen tai taaksepäin ns. nykyisestä rivistä. RANGE määrittelee loogisen ikkunatyypin loogisella arvovälillä [nykyinen arvo – x, nykyinen arvo], jossa x on jokin aikayksikkö. Lauseke BETWEEN x PRECEEDING ja y FOLLOWING määrittelee ikkunan koon.

```
(1)   SELECT product, date, SUM(sales) sum_sales,
(2)         SUM(SUM(sales)) OVER
(3)               (PARTITION BY product ORDER BY date
(4)                  RANGE INTERVAL '1' MONTH PRECEEDING) mavg
(5)   FROM salesTable
(6)   GROUP BY product, date;
```

Kuva 34: Esimerkki Window –funktion soveltamisesta.

Kuvan 34 kyselyssä rivillä 2 mitta-attribuutti sales summataan tuotteittain ositetuissa joukoissa, jotka järjestetään päivämäärän mukaiseen järjestykseen rivillä 3. Tämän jälkeen järjestetyn joukon jokaiseen riviin sovelletaan <aggregate group clause> -rakennetta RANGE INTERVAL ’1’ MONTH PRECEEDING (rivi 4). Tulokseksi saadaan ”looginen” ikkuna, joka sisältää

jokaisesta päivästä kuukauden ajalta taaksepäin myydyn tuotteen myyntimäärän liikkuvan summan (moving sum) tuotteittain ja päivämäärittäin.



Kuva 35: Fyysinen ja looginen ikkuna.

Kuvan 35 ensimmäinen kysely (rivit 1-4) tuottaa fyysisen ikkunan, jossa on kolme riviä. Toinen kysely (rivit 5-8) tuottaa loogisen ikkunan, jossa rivejä on kaksi, koska taulussa salesTable ei ole rivejä 24.10.1999 edeltävälle päivälle.

7.6.3. Raportointifunktiot

Raportointifunktiot muodostavat oman ryhmänsä window -funktioita. Niissä ikkunaan sisältyy ositetun joukon kaikki rivit. Jokaiseen ositetun joukon riviin on liitetty ositetusta joukosta laskettu yhteenvetotieto.

```
(1)  SELECT *
(2)  FROM (SELECT year(date), product, SUM(sales) s_sales,
(3)                MAX(SUM(sales)) OVER
(4)                     (PARTITION BY product) m_sales
(5)        FROM salestTable
(6)        GROUP BY year(date) , product
(7)   WHERE s_sales = m_sales;
```

Kyselyn tulos:

year	product	s_sales	m_sales
1998	P1	1800	1800
1999	P3	2000	2000

Kuva 36: Esimerkki raportointifunktiosta.

Kuvan 36 raportointifunktio tulostaa jokaiselta vuodelta vuoden aikana eniten myydyn tuotteen ja sen myyntimäärän. Rivillä 3-4 muuttujaan m_sales

lasketaan eniten vuoden aikana myydyn tuotteen myyntimäärä. Rivillä 6 aineisto ryhmitellään vuoden ja tuotteen mukaiseen järjestykseen. Lopuksi rivillä 7 tulosrelaatioon otetaan mukaan vain rivit, joissa mitta-attribuutin sales summattu yhteenvetotieto (s_sales) on yhtä suuri kuin eniten myydyn tuotteen myyntisumma (m_sales).

7.6.4. Lag / Lead funktiot

Lag / Lead funktioilla saadaan käsiteltäväksi mikä tahansa rivi, joka on nykyisestä rivistä tietyn siirtymän päässä taaksepäin tai eteenpäin järjestetyssä ositetussa joukossa. Näitä funktioita käytetään, kun joukon eri rivejä verrataan keskenään. Näillä voidaan välttää kompleksissa SQL –kyselyissä usein käytetyt liitokset samaan tauluun.

```
(1)  SELECT product, month(date), SUM(sales) sum_sales
(2)  SUM(sales) - LAG(SUM(sales), 1) OVER
(3)                  (PARTITION BY product
(4)                ORDER BY month(date)) diff
(5)   FROM salesTable
(6)   GROUP BY product, month(date);
```

Kuva 37: Esimerkki Lag / Lead –funktiosta.

Kuvan 37 kyselyssä jokaisen tuotteen kuukauden myyntimäärä tulostetaan muuttujaan sum_sales (rivi 1), ja tämän myyntimäärän ja sitä edeltävän kuukauden myyntimäärän erotus muuttujaan diff (rivit 2-4).

7.6.5. Käänteinen prosentuaalinen osuus

Käänteisessä prosentuaalisessa osuudessa (inverse percentile) kyselyssä ilmoitetaan annettu prosentuaalinen osuus joukon alkioista. Kysely palauttaa tuloksena ehdon täyttävät rivin tai rivit.

Ilmaisu PERCENTILE_CONT laskee jatkuvan prosentuaalisen arvon ja PERCENTILE_DISC palauttaa varsinaisen arvon, joka on lähinnä annettua prosentuaalista lukua, joka on nollan ja yhden välillä.

Homes taulun sisältö:

Area	Address	Price
Uptown	15 Peak St	456 000
Uptown	27 Primrose Path	349 000
Uptown	44 Shady Lane	341 000
Uptown	23301 Hihway 64	244 000
Uptown	34 Design Rd	244 000
Uptown	77 Sunset Strip	102 000
Downtown	72 Easy St	509 000
Downtown	29 Wire Way	402 000
Downtown	45 Diamond Lane	203 000
Downtown	76 Blind Alley	201 000
Downtown	15 Tern Pike	199 000
Downtown	44 Kanga Rd	102 000

```
(1)   SELECT Homes.Area, Avg(Homes.price)
(2)   PERCENTILE_DISC (0,5)   WITHIN GROUP (ORDER BY Homes.price DESC),
(3)   PERCENTILE_CONT (0,5) WITHIN GROUP (ORDER BY Homes.price DESC)
(4)   FROM Homes GROUP BY Area;
```

Kysellyn tulos:

Area	Avg	PERCENTILE_DISC	PERCENTILE_CONT
Uptown	289 333	341 000	292 500
Downtown	269 333	203 000	202 000

Kuva 38: Esimerkki käänteisestä prosentuaalisesta osuudesta.

Kuvan 38 esimerkkikysely laskee jokaiselle alueelle (area) keskiarvon ja mediaanin. Rivillä 1 lasketaan asuntojen hintojen keskiarvo. Rivi 2 tulostaa mediaanin (50%) asuntojen hinnoista alueittain funktiolla PERCENTILE_DISC (discrete percentile). Rivillä 3 tulostetaan jatkuva prosentuaalinen arvo funktiolla PERCENTILE_CONT (continuous percentile).

7.6.6. GROUP BY -lausekkeen Grouping set laajennukset

CUBE –operaatio tuo kumulatiiviset summat jokaista mahdollista dimensioattribuuttien yhdistelmää kohti, ja ROLL-UP –operaatio laskee dimensioattribuuteista välisummat oikealta vasemmalle. GROUPING SETS –operaatiolla käyttäjä määrittelee mitta-attribuuttien välisummat haluamilleen dimensioattribuuteille.

Operaatio ROLLUP (a, b, c) tuottaa saman tuloksen kuin GROUPING SETS ((a, b, c), (a, b), ()). Ryhmittely () merkitsee ylintä dimension yhteenvetotasoa.

```
(1)   SELECT Time, Region Department, SUM(Profit)
(2)   FROM Sales
(3)   GROUP BY GROUPING SETS ((Time, Region, Department),
(4)   (Time,Department), (Region, Department));
```

Kyselyn tulos:

Time	Region	Department	Profit
1998	Central	VIdeoRental	75 000
1998	Central	VIdeoSales	74 000
1998	East	VIdeoRental	89 000
1998	East	VIdeoSales	115 000
1998	NULL	VIdeoRental	164 000
1998	NULL	VIdeoSales	189 000
1999	Central	VIdeoRental	82 000
1999	Central	VIdeoSales	85 000
1999	East	VIdeoRental	101 000
1999	East	VIdeoSales	137 000
1998	NULL	VIdeoRental	183 000
1998	NULL	VIdeoSales	222 000
NULL	Central	VIdeoRental	157 000
NULL	Central	VIdeoSales	159 000
NULL	East	VIdeoRental	190 000
NULL	East	VIdeoSales	252 000

Kuva 39: GROUPING SETS –ilmauksen soveltaminen Sales –kuutioon.

Kuvan 39 kyselyn peruskuutio on muotoa:

```
Sales(Time, Region, Department, Profit)
```

Sales -kuutiossa Time (aika), Region (alue) ja Department (osasto) ovat dimensioattribuutteja, ja Profit (tuotto) on mitta-attribuutti. Kuvan 39 kysely laskee mitta-attribuutille Profit yhteenvetotiedot kolmelle eri ryhmälle: rivillä 3 (Time, Region, Department), rivillä 4 (Time, Department) ja (Region, Department). Tulosrelaation NULL –arvoiset rivit ovat yhteenvetotietorivejä. Kyselyssä ei tulosteta ryhmittelyä (Time, Region).

7.6.7. Yhdistetyt sarakkeet (composite columns)

Yhdistetty sarake on sulkumerkein ympäröity joukko sarakkeita, joita käsitellään yhtenä yksikkönä laskettaessa ryhmän yhteenvetotietoja.
Esimerkki, jossa (quarter, month) on yhdistetty sarake:

```
GROUP BY ROLLUP (year, (quarter, month), day)
```

Em. lauseke ei tee erottelua ROLL-UP –operaatiolla neljännesvuosi– ja kuukausisarakkeiden välillä, koska ne ovat yhdistettyjä sarakkeita (composite columns). Tästä seuraa, että em. lauseke ei tulosta yhteenvetotietoja vuosi-neljännesvuosi (year, quarter) –ryhmittelystä. Kyselyn tuloksena saadaan seuraavanlaiset yhteenvetotiedot:

(year, quarter, month, day),

(year, quarter, month),

(year)

```
(1)   SELECT Year, Quarter, Month, SUM(Profit) AS Profit
(2)   FROM Sales
(3)   GROUP BY ROLLUP(Year, (Quarter, Month))
```

Kyselyn tulos:

Year	Quarter	Month	Profit
2002	Winter	Jan	55 000
2002	Winter	Feb	64 000
2002	Winter	Mar	71 000
2002	Spring	Apr	75 000
2002	Spring	May	86 000
2002	Spring	Jun	88 000
2002	Summer	Jul	91 000
2002	Summer	Aug	87 000
2002	Summer	Sep	101 000
2002	Fall	Oct	109 000
2002	Fall	Nov	114 000
2002	Fall	Dec	133 000
2002	NULL	NULL	1 074 000

Kuva 40: Esimerkki yhdistetyn sarakkeen ilmaisemisesta.

Kuvan 40 kyselyssä yhteenvetotiedot tulostetaan aikadimension hierarkiatasoille: (year, quarter, month) ja (year). Tasolle (year,quarter) ryhmittelyä ei lasketa.

7.6.8. Yhdistetty ryhmä (concatenated grouping)

Yhdistetty ryhmä antaa mahdollisuuden muodostaa dimensioattribuuteista tiettyjä ryhmiä. Ryhmien yhdistely antaa tulokseksi ryhmien tulon:

```
GROUP BY GROUPING SETS (a, b), GROUPING SETS (c, d)
```

Kun a, b, c ja d ovat lähtökuutiossa olevia dimensioattribuutteja, yo. SQL –lause tuottaa tuloskuutioon seuraavat ryhmittelyt:

(a, c), (a, d), (b, c), ja (b, d)

Yhdistetyissä ryhmissä voidaan käyttää useita ROLLUP -, CUBE - tai GROUPING SET –operaatioita, jotka erotetaan toisistaan pilkuilla.

7.6.9. GROUPING_ID - ja GROUP_ID –funktiot

GROUP BY –lausekkeella tuotetun tietyn dimensiohierarkiatason rivin löytäminen tulosrelaatiosta edellyttää, että jokaista GROUP BY –lausekkeen saraketta kohti määritellään GROUPING –funktio, joka palauttaa tiedon (true tai false) siitä onko kyseisellä rivillä sarake yhdistelty ylimmälle tasolle (ks. kuva 20). Yhdessä käytettynä nämä kaksi funktiota korvaavat GROUPING –funktion. Näillä saadaan tulosrelaatiosta tunnistettua eri ryhmittelytasojen sisältämät tuplarivit, joita voi syntyä mutkikkaissa kyselyissä.

GROUPING –funktioiden sarakkeet saadaan korvattua GROUPING_ID -funktiolla yhdellä sarakkeella. GROUPING_ID –funktio palauttaa luvun, jonka perusteella voidaan päätellä tietty GROUP BY –taso. Jokaiselle riville GROUPING-ID palauttaa jokaista sisältämäänsä dimensioattribuuttia kohden yhden bitin, jonka arvo on 0 tai 1 ja yhdistää nämä arvot yhdeksi bittivektoriksi. GROUPING_ID –funktio palauttaa sarakkeeseen tämän bittivektorin arvon kymmenjärjestelmän lukuna.

Kysely voi tuottaa tulosrelaatioon tuplarivejä. Tuplarivien tunnistamisessa käytetään GROUP_ID –funktiota. Se numeroi jokaisen tuplarivin yhden välein alkaen luvusta 1. GROUP_ID:n sisältämä arvo 0 on merkki siitä, että tämä on yksilöity rivi jatkokäsittelyä varten. Tuplarivejä ovat ne rivit, joissa vastaavan sarakkeen arvo on suurempi kuin 0.

```
(1)   SELECT Region, State, SUM(sales) AS sum_sales,
(2)   GROUPING_ID (state, region),
(3)   GROUP_ID()
(3)   FROM salesTable
(4)   GROUP BY GROUPING SETS (region, ROLLUP(region, state));

Kyselyn tulos:

      Region      State     sum_sales      grouping_id      group_id
      ------------------------------------------------------------------
(5)   W           CA        2 000          0                0
(6)   E           NULL      1 000          2                0
(7)   E           NULL      1 000          2                1
(8)   W           NULL      2 000          2                0
(9)   W           NULL      2 000          2                1
(10)  NULL        NULL      3 000          3                0
```

Kuva 41: GROUPING_ID – ja GROUP_ID – funktio.

Kuvan 41 kysely tulostaa rivin 4 mukaan yhteenvetotiedot alueesta (region) ja ROLLUP –funktiolla tiedot alueesta ja valtiosta (state). Kysely tuottaa seuraavat ryhmittelyt: (region, state), (region), (region) ja (). Alueen (region) yhteenvetotiedoista syntyy siis tulosrelaatioon tuplarivejä. Esimerkkikyselyssä näitä region –ryhmittelyn tuplarivejä ovat rivit 6 ja 7 sekä rivit 8 ja 9. Sarakkeen

group_id:n nolla-arvoiset rivit ovat kunkin tason yksikäsitteisiä rivejä, joilla tuplariveistä saadaan eroteltua halutut rivit jatkokäsittelyyn. Kyselyn tuloksen grouping_id –sarake sisältää ryhmittelytasojen numeroinnin.

7.7. Multidimensional SQL, SQL_M

Pedersen *et. al.* [2002] ovat kehittäneet SQL_M –kielen, jonka tavoitteena on ollut ilmaisuvoimaisuus, summautuvuuden oikeellisuuden tarkistava automaattinen aggregointi ja XML –dokumenttien integrointi OLAP –käsittelyyn. Kieli on yhteensopiva SQL –standardin kanssa ja on sen osajoukko.

Pedersen *et. al.* [2002] ovat kehittäneet moniulotteisen OLAP –tietokaavion, siihen perustuvan formaalin algebran ja kyselykielen. Näiden avulla pystytään käsittelemään dimensiohierarkioita, jotka eivät ole tiukkoja (non-strict) eivätkä kattavia (non-covering). Dimensiohierarkia ei ole tiukka tai ehdoton, kun alemman tason dimensioattribuutin arvo voi sisältyä useampaan kuin yhteen ylemmän tason dimensioattribuuttiin. Dimensiohierarkia ei ole kattava, kun hierarkiassa yksikin dimensioattribuutin arvo ei sisälly jonkin välittömästi seuraavan ylemmän tason dimensioattribuutin arvoon. Summautuvuuden edellytykset toteutetaan siten, että jokaisen dimension ja mitta-attribuutin välille luodaan tieto siitä, mitä aggregointifunktioita näiden yhteydessä voidaan käyttää. Aggregointifunktiot on tyypitetty seuraaviin aggregointityyppeihin: ? –tyyppinen tieto, jota ei voida aggregoida, koska summautuvuuden ehdot eivät toteudu, ø –tyyppinen tieto, josta voidaan laskea keskiarvo, mutta ei sitä ei voi käyttää yhteenlaskussa ja S –tyyppinen tieto, jota voidaan käyttää myös yhteenlaskussa. SQL –funktiot sisältyvät näihin tyyppeihin seuraavasti: S = {AVG, COUNT, MAX, MIN, SUM}, ø = {AVG, COUNT, MAX, MIN} ja ? = {ø}. Kun tiettyä mitta-attribuuttia ja dimensiota aggregoidaan funktio palauttaa hyväksyttävän aggregointityypin. Jos kyselyssä käytetty aggregointifunktio ei ole hyväksyttyä tyyppiä, ohjelmisto estää tai varoittaa käyttäjää virheellisestä käytöstä. Dimensioon ja mitta-attribuuttiin voidaan liittää myös oletusaggregointifunktio, jota aggregoitaessa käytetään.

Seuraavissa kahdessa kyselyssä käytetään Purchases –relaatiota (ostot):

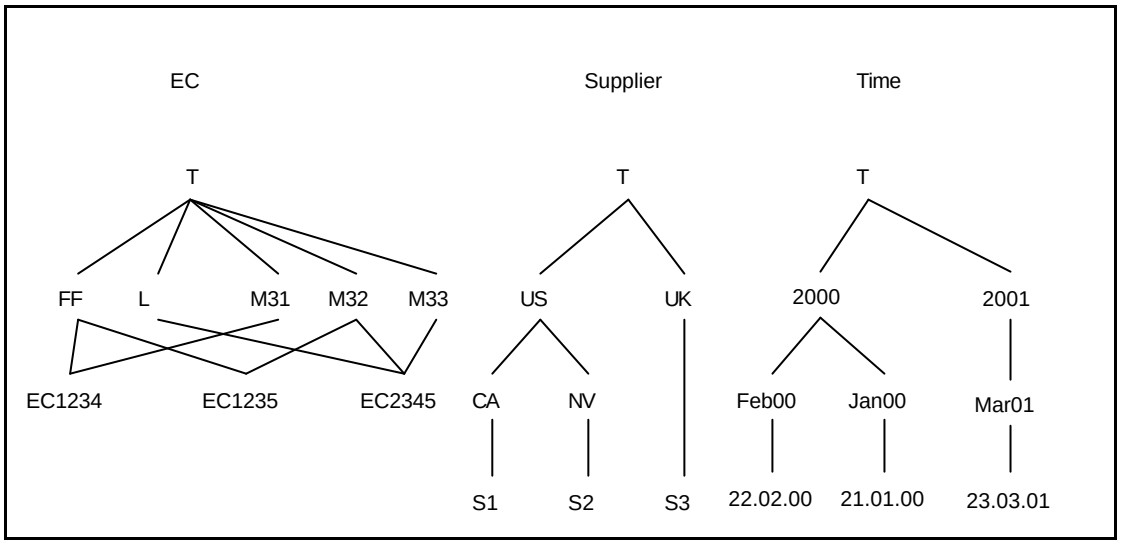
Purchases(Day, Supplier, EC, Cost, No. Of Units)

Purchases –relaatiossa Day (päivä), Supplier (toimittaja), ja EC (elektroninen komponentti) ovat dimensioattribuutteja ja Cost (ostohinta) ja No.Of Units (yksikkömäärä) ovat mitta-attribuutteja.
Toimittajadimension (Supplier) hierarkiatasot ovat seuraavat:

Supplier > State > Country > All tai
Supplier > Country > All.

EC –dimensiossa on seuraavat hierarkiat:

EC > Class > All
EC > Manufacturer > All



Kuva 42: Purchases -taulun ilmentymätaso.

Kuvassa 42 esitetään purchases –taulun ilmentymätaso, jossa dimension EC luokkatason (class) dimensioattribuuttiarvojoukko on {FF, L}, ja manufacturer dimensiotason arvojoukko on {M31, M32, M33}. Hierarkian ylin taso koostuu T:stä. Kuviosta nähdään, että komponenttia EC2345 toimittaa kaksi eri toimittajaa (EC:n dimensiohierarkia ei ole tiukka). Dimensiossa Supplier hierarkia ei ole kattava, koska toimittajan S3 välitön ylempi taso on UK.

```
(1)  SELECT DEFAULT(cost), supplier, class(EC)
(2)  FROM purchases
(3)  WHERE country(supplier) = 'UK'
(4)  GROUP BY supplier, class(EC)
(5)  HAVING DEFAULT(cost) > 10000
```

Kuva 43: Multidimensional SQL.

Kuvan 43 kyselyssä lasketaan tuotteen luokittelun (class) ja toimittajan mukaan ostojen kustannukset (cost) niistä toimittajista, joiden sijaintipaikka on UK ja kustannusten yhteismäärä ylittää 10 000 (rivi 5). Rivillä 1 class(EC) -ilmaisu tarkoittaa ROLL-UP –operaatiota, jossa kustannukset aggregoidaan elektronisen komponenttidimension (EC) alimmalta tasolta luokkatasolle (class). Mitta-arvon yhteenvetotietoja laskettaessa käytetään oletusaggregointifunktiota DEFAULT(cost), joka tässä tapauksessa korvataan SUM(cost) funktiolla (rivit 1 ja 5).

SQL_M:ssä tuloskuutioon voidaan hakea tietoa XML –dokumentista [WC3, 2000]. XML –dokumentin integrointi kuutioon perustuu XPath lausekkeiden käyttöön SQL_M -kyselykielessä. XPath –kielen avulla voidaan käsitellä XML –dokumentteja [WC3, 1999]. Esimerkiksi Xpath -lauseke //Component/Manufacturer [@Mcode=”M31”] valitsee XML –dokumentista kaikki komponentit, joissa Mcode -attribuutin arvona on ”M31”.

Kuvan 44 kyselyssä luodaan yhteys kuution dimensioattribuutin EC arvojoukon ja XML –dokumentin yksikköhintojen (unit price) välille. Jokaiselle dimensiotasolle voi olla määritelty oletusyhteys, jolloin yhteyttä ei tarvitse määritellä kyselylausekkeessa. Jos esimerkiksi EC –tasolle oletusyhteydeksi on määritelty ”EC –link”, lauseke ”EC/EC-link/Description” voidaan kyselyssä ilmaista merkkijonolla ”EC/Description”. SQL_M -kuutiota, yhteystietoja ja näihin liittyviä XML –dokumentteja sanotaan federaatioksi (federation).

```
(1)  SELECT DEFAULT(cost), supplier, EC
(2)  FROM PurchaseFederation
(3)  WHERE EC/UnitPrice[@Currency='euro']<3.00
(4)  GROUP BY supplier, EC
```

Kuva 44: Multidimensional SQL:n ja XML:n ilmausten yhdistäminen.

Kuvan 44 WHERE –lausekkeessa rivillä 3 ilmaistaan oletusyhteys kuution EC -dimensioattribuutin ja XML –dokumentin elementin UnitPrice välillä. Kysely tuottaa tulosrelaation, jossa on tiedot toimittajittain komponenteista, joiden yksikköhinta on alle 3,00 euroa (rivi 3).

Samalla tapaa XML –dokumentista voidaan linkin mukaan tuoda myös dimensiotasoon liittyvää muuta tietoa, kuten esim. toimittajan osoite- tai yhteyshenkilötiedot. Nämä ovat ns. decoration –sarakkeita. XML –dataa voidaan ryhmitellä ja käyttää valintaperusteena.

Kyselyssä voi käyttää vain yhtä faktataulua, mutta dimensioiden liittämistä faktatauluun ei tarvitse tehdä eksplisiittisesti kuten SQL –99:ssä. Tämän ansiosta kyselyjen muotoilu on tiivistä ja ilmaisuvoimaista.

8. MUITA OLAP –KÄSITTELYYN SOVELLETTUJA KYSELYKIELIÄ

SQL ei ole ainoa vaikkakin käytetyin kyselykieli OLAP –ympäristössä. Seuraavissa kappaleissa tarkastellaan OLAP –käsittelyä logiikkaohjelmointikielillä, joiden toteutukset perustuvat Prologiin tai Datalogiin. Kolmantena esitetään Microsoftin kehittämää MDX kieltä, joka on kehitetty OLAP –kyselykieleksi.

8.1. Prolog

Niemi T. *et.al.* [2003] toteuttivat näkymäorientoituneen ja ilmaisuvoimaisen kyselykielen Prologilla. Heidän kielensä käyttöliittymän avulla käyttäjä voi tehdä kyselyjä operaatio-orientoituneita kieliä deklaratiivisemmalla tavalla.

Kyselyä muodostaessaan käyttäjä antaa kyselyn tulostaululle nimen (Result Table), dimensioattribuuttien arvojen poimintaehdot (Dimension Conditions), tulostaulun sarakkeiden nimet (Columns), sarakemäärittelyn (Column Definition) ja taulun aggregoinnin (Table Aggregation).

Kielen ilmaisuvoimaa kuvaa se, että käyttäjä voi muodostaa tulostauluun mitta-attribuuttien yhteenvetotietoina sarakkeita, jotka ovat aggregoitu jonkun tai joidenkin kaaviotason dimensioattribuuttien arvojen ryhmittelyn perusteella. Käyttöliittymässä voidaan siis tuottaa tulosrelaatioon riveistä sarakkeita, mikä SQL:llä on mutkikasta tehdä. Käyttäjä rajoittaa dimensioattribuutin sisältöä sarakemäärittelyllä ja kysely summaa oletusarvoisesti mitta-attribuuttien arvot halutuille ryhmille. Käyttäjän ei myöskään aina tarvitse kyselyssä ilmoittaa tietokuution taulujen nimiä, vaan sarakkeiden nimet riittävät.

Taulun aggregointimäärittelyllä kyselyn tulokseen voidaan laskea sekä rivien että sarakkeiden summat ja keskiarvot varatuilla sanoilla col_sums, row_sums, col_avg ja row_avg. SQL –kielellä vastaavan tulosrelaation tuottaminen edellyttää monivaiheista kyselyä, jossa joudutaan käyttämään useita aputauluja.

8.2. TOLAP –temporaalinen kyselykieli

Dimensiot ymmärretään OLAP –käsittelyssä useimmiten staattisiksi tiedoiksi. Reaalimailmassa dimensioiden rakenne kuitenkin muuttuu hitaasti. Jotta tietojen analysointi tuottaisi luotettavaa ja validia tietoa on tietoihin liitettävä

niiden kelpoisuusaika. Mendelson ja Vaisman [2000] käsittelevät tätä aihetta esittäessään temporaalisen moniulotteisen mallin.

Dimensioissa rakenne voi muuttua siten, että hierarkiatason dimensioattribuutin arvoista yksi tai useampi jäsen siirretään sisältyväksi johonkin toiseen ylemmän tason dimensioattribuutin arvoon. Kun tällainen muutos tapahtuu muutosajankohtaa, aikaisemmatkin mitta-arvot yhdistyvät muutetun dimensioattribuutin arvon mukana ylemmän tason dimensioattribuutin arvoon. Tässä tapauksessa kadotetaan tietojen historiaan liittyvä oikea tieto. Näin käy, kun esimerkiksi kauppa dimension jokin kauppa siirretään toisesta alueesta toiseen. Tältä osin tietorakenteeseen tulisi lisätä tiedon kelpoisuuden alkamis- ja päättymisaika.

Mendelson ja Vaisman [2000] ovat käyttäneet Datalogia kyselykielenä, jota he nimittävät temporaaliseksi OLAP –kieleksi (TOLAP). Sillä tehdyt kyselyt voidaan muuntaa SQL –kielelle tai TSQL2:lle (Temporal SQL2).

Mendelson ja Vaisman [2000] argumentoivat, että OLAP –järjestelmässä täytyy olla aikaan liittyviä ominaisuuksia, joiden avulla tietovaraston tila tiedetään sen elinkaaren aikana.

8.3. Multidimensional expressions, MDX

Microsoft on kehittänyt moniulotteisen tietokannan kyselykielen Multidimensional Expressions (MDX) [Spofford, 2001, s. 1-19]. MDX on kehitetty yksinomaan OLAP –käsittelyä varten ja se ei ole SQL –kielen laajennus vaikka siinä on joitakin SQL –kielen varattuja sanoja. Tässä esitetään vain muutamia kielen piirteitä ja funktioita, jotka toisaalta tuovat esille SQL –kielen puutteet ja MDX -kielen mahdollisuudet muodostaa erilaisia kuutioita kyselyjen tuloksena. Tämä kuvaus ei ole kattava, vaan tarkoitus on esittää lyhyesti johdatuksen tapaan MDX –kielen joitakin keskeisiä piirteitä ja ominaisuuksia. Yksinkertainen MDX –lauseke on muodoltaan seuraava:

```
SELECT <axis specification> [,axis specification, … ]
FROM <cube specification>
WHERE <slicer specification>
```

Jokainen <axis specification> määrittelee kuinka dimensio esitetään kyselyn tuloskuutiossa. Tässä määrittelyssä annetaan mm. dimensioiden nimet ja hierarkiatasot, joille aggregointi tehdään. SELECT lauseessa määritellään ensin

tuloskuution sarakkeet ja seuraavaksi rivit. <axis specification> -rakenne määrittelee moniulotteisen avaruuden akselit. MDX sisältää useita tapoja määritellä dimension hierarkiatasoja akselille. Määrittely tapahtuu esim. funktioilla: Children(), Ancestor(), Ancestors(), Acendants() ja Decendants().

Kielessä on viisi nimettyä akselia, COLUMNS, ROWS, PAGES, CHAPTERS ja SECTIONS, jotka tulee olla lausekkeessa tässä järjestyksessä. Kuva 45 esittää yksinkertaisen kyselyn. FROM –lausekkeessa voi olla vain yksi kuution nimi. WHERE –lausekkeen <slicer specification> -rakenne toimii mm. SLICE – operaationa, jolla kuutiosta otetaan “siivu”. Jos tuloskuutio sisältää akseleita, esim. rivejä tai sarakkeita, joissa solujen sisältö on tyhjä, voidaan kyselyssä antaa komento, joka jättää tällaiset rivit pois tuloksesta.

```
(1)   SELECT
(2)      { [Time] . [June-2001], [Time] . [July-2001], [Time] . [August-2001] }
(3)    on columns
(4)      { [Stores] . [Dowtown], [Stores] . [Uptown] }
(5)   on rows
(6)   FROM Cube
(7)   WHERE  [Measures] . [Costs] }
```

Kyselyn tulos:

Measure: Cost	June2001	July-2001	August-2001
Downtown	100	1050	1050
Uptown	800	900	1050

Kuva 45: Esimerkkikysely MDX –kielellä.

Kuvan 45 kyselyssä Time ja Stores ovat dimensioiden nimiä. Niiden jäljessä esitetään halutun hierarkiatason mukaan poimittavat dimensioattribuutin tietyt arvot (rivit 2 ja 4). WHERE –lausekkeessa tuloskuutioon tulostetaan vain Costs – mitta-attribuutista yhteenvetotiedot (rivi 7).

Kielessä on runsaasti tuloskuution määrittelyyn vaikuttavia funktioita, joista seuraavaan luetteloon on poimittu vain muutamia kuvaamaan kielen ilmaisukykyä:

- CrossJoin() tuottaa jäsenten kaikki mahdolliset yhdistelmät.
- Filter() suodattaa halutun rajauksen perusteella kuution dimensioattribuuttien tai mitta-attribuuttien arvoja.
- Order() – funktiolla tuloskuution akselit saadaan haluttuun järjestykseen.

Koska MDX –kieltä ei ole kehitetty SQL:n laajennuksena on luonnollista, että siihen on saatu sisällytettyä mahdollisimman runsaasti OLAP –käsittelyssä tarvittavia ominaisuuksia. Dimensioattribuuttien vapaa sijoittelu tuloskuution riveille tai sarakkeille on SQL:n perusrajoitus. MDX –kielessä tämä on mahdollista. Sillä tulos voidaan esittää myös sisennettyinä taulukkoina. MDX sisältää runsaasti muitakin SQL –kieleen sisältymättömiä toimintoja, mutta niiden kaikkien yksityiskohtainen kuvaaminen ei ole tämän tutkimuksen aiheena.

9. YHTEENVETO

OLAP –järjestelmän käyttöä edeltää tietojen kokoaminen ja muokkaaminen operatiivisista järjestelmistä moniulotteiseksi tietorakenteeksi. Moniulotteisesti organisoitujen tietojen tallentaminen tietovarastoon on tämän ABDOP –prosessin lopputulos. Tässä tutkimuksessa esitettiin moniulotteiseen tietorakenteeseen liittyviä ominaisuuksia ja sääntöjä, joilla tietojen analysointiin liittyvät tulokset pyritään saamaan virheettömiksi. Tutkielmassa kuvattiin yleisimmin tunnettuja loogisia malleja ja näiden rakenteen sisältämiä sääntöjä. Moniulotteisista loogisista rakenteista esiteltiin keskeisimmät tietovaraston kaaviotason mallit. Lisäksi tarkasteltiin OLAP –käsittelyn keskeistä käsitettä kuutiota. Tutkielmassa tarkasteltiin SQL –kieleen kehitettyjä laajennuksia, joista osa jo sisältyy SQL-99 standardiin, sekä muita OLAP –käsittelyssä käytettäviä kieliä.

Työssä tutkittiin erilaisten loogisten mallien vaikutusta SQL –kyselykielen formulointiin. Tutkittavina perusmalleina olivat universaalimalli, normalisoimaton kuutio, tähtimalli, lumihiutalemalli, konstellaatiomalli ja tämän kanssa yhdenmukainen monikuutiomalli. Uusina perusmalleina tutkimuksessa tarkasteltiin universaalirelaation ja normalisoimattoman kuution käyttämistä moniulotteisena loogisena mallina. Tulokseksi saatiin, että universaalirelaatioon ja lumihiutalemalliin tehdyt kyselyt esitetään lähes samanlaisina SQL -kielellä. Nämä mallit eivät ole kuitenkaan kyselyn muotoilun kannalta optimaalisimpia malleja. Normalisoimaton kuutio ja tähtimalli antavat selkeimmän ja yksinkertaisimman tavan formuloida kysely.

Loogisten mallien vaikutusta kyselykieleen tarkasteltiin myös suhteessa dimensiotasojen määrään ja kardinaliteettiin sekä ilmentymätasoon. Tarkastelu osoitti, että jos loogisessa mallissa relaatioiden määrä kasvaa, niin vastaavasti kyselyt muodostuvat pitkiksi ja kompleksisiksi. Tämä lisää osaltaan kyselyn prosessointiaikaa. Normalisoitujen relaatioiden ylläpito on yksinkertaisempaa kuin normalisoimattomien relaatioiden. Normalisoimaton kuutio sisältää paljon redundanttia tietoa, mutta toisaalta kyselyt tällaiseen malliin kohdistettuna muodostuvat selkeiksi, tiiviiksi ja intuitiivisiksi.

SQL:n sisältämän CUBE –operaation toimintaa arvioitiin kyselyn tuottaman tulosrelaation kannalta. SQL –kieleen on lisätty runsaasti ns. analyyttisiä funktioita, joilla tulosrelaation sisältöä voidaan määritellä ja hallita CUBE –operaatiota yksityiskohtaisemmin. SQL –kieleen tehdyillä laajennuksilla

voidaan tehdä hyvinkin kompleksisia kyselyjä. Siksi SQL:n OLAP – laajennuksia sisällytetään SQL:n standardiin. SQL on edelleenkin käyttökelpoinen kieli ROLAP -lähestymistavassa, jossa tietojen ylläpito ja järjestelmän laajentaminen on joustavampaa kuin MOLAP -lähestymistavassa.

Tässä tutkimuksessa tarkasteltiin myös sitä, että logiikkaohjelmointikielellä voidaan toteuttaa käyttöliittymä, jolla voidaan tuottaa tuloskuutioita. SQL -kielellä tätä on työlästä määritellä. Kieli perustuu logiikkaohjelmoinnin muuttujakäsitteeseen. Tästä annettiin esimerkkejä.

Tässä tutkimuksessa sivuttiin tietovaraston evoluutiota mainitsemalla, että OLAP -järjestelmän käytön aikana on varauduttava moniulotteisen tietorakenteen muutoksiin. OLAP dimensiot ovat useimmiten staattisia, mutta käytännössä niiden tulee muuttua vastaamaan reaalimaailman muutoksia. Eräs jatkotutkimuksen aihe on se, miten tietovaraston tietoihin sisällytetään tiedon kelpoisuusaika, ja miten se vaikuttaa loogisiin malleihin, ja mitä uusia ominaisuuksia se edellyttää kyselykieleltä.

Viiteluettelo

[Agarwal *et al.*, 1996] Sameet Agarwal, Rakesh Agrawal, Prasad M. Deshpande, Ashish Gupta, Jeffrey F. Naughton, Raghu Ramakrishnan, Sunita Sarawagi. On the computation of multidimensional aggregates, *Proceedings of the International Conference on Very Large Databases, Bombay, India, September 1996*, 506–521.

[Agrawal *et al.*, 1997] Rakesh Agrawal, Ashish Gupta, and Sunita Sarawagi. Modeling multidimensional databases, *Proceedings of the 13th International Conference on Data Engineering, Birmingham, U.K. April 1997.*
http://www.almaden.ibm.com/cs/quest
10.9.2002.

[Baralis *et al.*, 1997] Elena Baralis, Stefano Paraboschi, Ernest Teniente. Materialized view selection in a multidimensional database, *Proceedings of the 23rd Conference on Very Large Databases, Athens, Greece, 1997.*

[Bellamkonda *et al.*, 2000] Srikanth Bellamkonda, Tolga Bozkaya, Bhaskar Ghosh, Abhinav Gupta, John Haydu, Sankar Subramanian, Andrew Witkowski. Analytic functions in Oracle 8*i* .
http://www-db.stanford.edu/dbseminar/Archive/SpringY2000/speakers/agupta/paper.pdf
20.03.2003.

[Chatziantoniou, 1999a] Damianos Chatziantoniou. Evaluation of ad hoc OLAP: in-place computation. *Proceedings of the 11th International Conference on Scientific and Statistical Database Management, 1999*, 34-43.

[Chatziantoniou, 1999b] Damianos Chatziantoniou. The PanQ tool EMF SQL for complex data management, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, August 1999.*

[Chatziantoniou *et al.*, 2001] Damianos Chatziantoniou, Michael Akinde, Theodore Johnson, Samuel Kim. The MD-join: an operator for complex OLAP. *The 17th International Conference on Data Engineering, April 02-06, 2001 Heidelberg, Germany*, 524-533.

[Chaudhuri ja Dayal, 1997] Surajit Chaudhuri, Umeshwar Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record, 26:6, 1997*, 55–74.
ftp://ftp.research.microsoft.com/users/surajitc/sigrecord.pdf
10.09.2002.

[Chaudhuri *et al.*, 2001] Surajit Chaudhuri, Umeshwar Dayal, Venkatesh Ganti. Database technology for decision support systems. *Computer*, Volume **34**, Issue 12, December 2001. IEEE Computer Society Press, Los Alamitos, CA, USA, 48-55.
http://dlib2.computer.org/co/books/co2001/pdf/rz048.pdf
10.04.2003.

[Clear *et al.*, 1999] John Clear, Debbie Dunn, Brad Harvey, Michael Heytens, Peter Lohman, Abhay Mehta, Mark Melton, Lars Rohrberg, Ashok Savasere, Robert Wehrmeister, Melody Xu. Nonstop SQL/MX primitives for knowledge discovery, *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August, 1999.*

[Colliat, 1996] George Colliat. OLAP, relational, and multidimensional database systems. *SIGMOD Record, Vol. **25**, No. 3, September 1996.*

[Date ja Darwen, 1997] C. J. Date, Hugh Darwen. *A Guide to THE SQL STANDARD*, Fourth Edition, Addison Wesley Longman, Inc., 1997.

[Dinter *et al.*, 1998] Barbara Dinter, Carsten Sapia, Gabriele Höfling, Marcus Blaschka. The OLAP market: state of the art and research issues, *DOLAP'98 Washington DC, USA, 1998.*

[E. F. Codd Associates, 1993] E. F. Codd Associates. Providing OLAP to user-analysts: an it mandate.
http://www.hyperion.com/downloads/OLAPCoddwp.pdf
10.09.2002.

[Elmasri ja Navathe, 1994] Ramez Elmasri, Shamkant B. Navathe. *Fundamentals of Database Systems*, Second Edition, The Benjamin/Cummings Publishing Company, Inc. ISBN 0-0853-1753-8.

[Gingras ja Lakshmanan, 1998] Frederic Gingras, Laks V. S. Lakshmanan. nD-SQL: a multi-dimensional language for interoperability and OLAP, *Proceedings of the 24th Conference on Very Large Databases, New York, USA, 1998.*

[Gray *et al.*, 1995] Jim Gray, Adam Bosworth, Andrew Layman, Hamid Pirahesh. Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals, *Microsoft Research Advanced Technology Division, Technical Report MSR-TR-95-22.*
http://www.research.microsoft.com/ gray
10.02.2002.

[Harinarayan *et al.*, 1995] Venky Harinarayan, Anand Rajaraman, Jeffrey D. Ullman. Implementing data cubes efficiently, *Proceedings of the ACM SIGMOD Conference on Management of Data, 1995*, 205–216.

[Hasan *et al.*, 2000] Helen Hasan, Peter Hyland, David Dodds, and Raja Veeraraghavan. Approaches to the development of multi-dimensional databases: lessons from four case studies, *The DATA BASE for Advances in Information Systems, Summer 2000*, Vol. **31**, No. 3**.**

[Hovi, 1997] Ari Hovi. *Data Warehousing - Tietovarastotekniikka*, Suomen Atk-kustannus Oy, ISBN 951-762-509-X.

[Hurtado ja Mendelzon, 2002] Carlos A. Hurtado, Alberto O. Mendelzon. OLAP dimension constraints, *Proceedings of the twenty-first ACM SIGMOS-SIGACT-SIGART symposium on Principles of database systems 2002, Madison, Wisconsin.*

[Huyn, 2001] Nam Huyn. Scientific OLAP for biotech domain, *Proceedings of the 27th Conference on Very Large Databases, Roma, Italy, 2001.*

[Inmon, 1996] W. H. Inmon. *Building the Data Warehouse*, Second Edition, Wiley Computer Publishing, ISBN No 0471-14161-5.

[Johnson ja Chatziantoniou, 1999] Theodore Johnson, Damianos Chatziantoniou. Extending complex ad-hoc OLAP, *Conference on the 8h International Conference on Information Knowledge Management, 1999.*

[Kimball, 2000a] Ralph Kimball. Rating your dimensional data warehouse, *Intelligent Enterprise Magazine, April 28, 2000, Volume **3**, Number 7***.*** *http://www.intelligententerprise.com/000428/webhouse.shtml* 10.09.2002.

[Kimball, 2000b] Ralph Kimball. Is your dimensional data warehouse expressive?, *Intelligent Enterprise Magazine, May 15, 2000, Volume **3**, Number 8.* *http://www.intelligententerprise.com/000515/webhouse.shtml* 10.09.2002.

[Kimball ja Strehlo, 1995] Ralph Kimball, Kevin Strehlo. Why decision support fails and how to fix it, *SIGMOD Record, Vol. **24**, No. 3, September 1995.*

[Last ja Maimon, 2000] Mark Last, Oded Maimon. Automated dimensionality reduction of data warehouses, *Proceedings of the 2nd Intl. Workshop DMDW'2000, Stockholm, Sweden, June 5.-6.2000.* (7-1 - 7-9). *http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-28/paper7.pdf* 10.09.2002.

[Lee, *et al.*, 2000] Sin Yeung Lee, Tok Wang Ling, HuaGang Li. Hierarchical compact cube for range-max queries, *Proceedings of the 26th Conference on Very Large Databases, Cairo, Egypt, 2000.*

[Lehner *et al.*, 1998] W. Lehner, J. Albrecht, H. Wedekind. Normal forms for multidimensional databases, *Proceedings of the 10th International Conference on Scientific and Statistical Data Management (SSDBM'98),Capri, Italy, 1998.*

[Lenz ja Shoshani, 1997] Hans-J. Lenz, Arie Shoshani. Summarizability in OLAP and statistical data bases, *Ninth International Conference On Scientific And Statistical Database Management (SSDBM), 1997,* 132-143.

[Lenz ja Thalheim, 2001] Hans-J. Lenz, Bernhard Thalheim. OLAP databases and aggregation functions.

[Li *et.al.*, 2001] Hua-Gang Li, Tok Wang Ling, Sin Yeung Lee, Zheng Xuan Loh, Range sum queries in dynamic OLAP data cubes.

[McCabe *et al.*, 2000] M. Catherine McCabe, Jinho Lee, Abdur Chowdhury, David Grossman, Ophir Frieder. On the design and evaluation of a multi-dimensional approach to information retrieval, *Proceedings on the ACM SIGIR Conference, Athens, 2000,* 363-365.

[Mendelzon ja Vaisman, 2000] Alberto O. Mendelzon, Alejandro A. Vaisman. Temporal queries in OLAP, *Proceedings of the 26th Conference on Very Large Databases, Cairo, Egypt, 2000.*

[Moody ja Kortink, 2000] Daniel L. Moody, Mark A. R. Kortink. From enterprise models to dimensional models: a methodology for data warehouse and data mart design. *Proceedings of the 2nd Intl. Workshop DMDW'2000, Stockholm, Sweden, June 5-6.2000,* 5-1 – 5-12.
http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-28/paper5.pdf
06.10.2001.

[Niemi, 2001] Tapio Niemi. Methods for logical OLAP design, Department of Computer and Information Sciences, FIN-33014 University of Tampere, Finland.

[Niemi *et al.*, 2003] Tapio Niemi, Jyrki Nummenmaa, Peter Thanisch. Normalising OLAP cubes for controlling sparsity, *Data & Knowledge Engineering, 46 (2003),* 317-343.

[Niemi T. *et al.*, 2003] T. Niemi, L. Hirvonen, K. Järvelin. Multidimensional data model and query language for informetrics, *Journal of the American Society for Information Sciences and Technology, 2003, Issue 54, Number 10,* 939-951**.**

[OLAP Council] OLAP and OLAP server definitions, OLAP: on-line analytical processing.
http://www.olapcouncil.org/research/glossary.htm
10.09.2002.

[Pedersen ja Jensen, 1999] Torben Bach Pedersen, Cristian S. Jensen. Multidimensional data modeling for complex data, *Proceedings of the 15th International Conference on Data Engineering, Sidney, Australia, 1999,* 336-345.

[Pedersen *et. al.*, 2002] Dennis Pedersen, Karsten Riis, Torben Bach Pedersen. A powerful and SQL–compatible data model and query language for OLAP, *The Thirteenth Australian Database Conference (ADC2002), Melbourne, Australia.*

[Pendse, 2000a] Nigel Pendse. OLAP architectures. *The OLAP Report*
http://www.olapreport.com/Acrhitectures.htm
15.09.2002.

[Pendse, 2000b] Nigel Pendse. Database explosion, *The OLAP Report 2000.*
http://www.olapreport.com/DatabaseExplosion.htm
15.09.2002.

[Pokorny ja Sokolowsky, 1999] Jaroslav Pokorny, Peter Sokolowsky. A conceptual modelling perspective for data warehouses, *Electronic Business Engineering 4 Internationale Wirtsharftsinformatik 1999,Hrsg.a-W. Scheer, M. Nuttgens, Heidelberg Physica-Verlag 1999,* 666-684.

[Pourabbas ja Rafanelli, 2000] Elaheh Pourabbas, Maurizio Rafanelli. Hierarchies and relative operators in the OLAP environment, *SIGMOD Record, Vol.* ***29****, No. 1, March 2000.*

[Schouten, 1999] Hans Schouten. Analysis and design of data warehouses, *Proceedings of the international Workshop on Design and Management of Data Warehouses (DMDW'99), Heidelberg, Germany, 14.-15.6.1999.*
http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-19/paper5.pdf
10.09.2002.

[Schwarz *et al.*,2000] Holger Schwarz, Ralph Wagner, Bernhard Mitschang. Improving the processing of decision support queries: the case for DSS optimizer, *Institute of Parallel and Distributed High-Performance Systems, University of Stuttgart, D-70565, Stuttgart, Germany.*
http://www.informatik.uni-stuttgart.de/ipvr/as/personen/schwarz/ideas01.pdf
10.09.2002.

[Shoshani, 1997] Arie Shoshani. OLAP and statistical databases: similarities and differences, *International Conference on Management of Data and Symposium on Principles of Database Systems, Proceedings of the sixteenth ACM SIGACT-*

SIGMOD-SIGART symposium on Principles of database systems, Tucson, Arizona, United States, 1997, 185-196.

[Spofford, 2001] George Spofford. *MDX Solutions*, John Wiley & Sons, Inc. ISBN: 0-471-40046-7.

[SQL Extensions, 1999] SQL extensions for analytic calculations, data warehousing and business intelligence.
http://www.cs.utexas.edu/users/dsb/CD-update/Oracle/Oracle_Aggregation.pdf
12.12.2002.

[Stefanovic, 1993] Nebojsa Stefanovic. Design and implementation of on-line analytical processing (OLAP) of spatial data, 1993.

[SYSTA / TIHA, 1997] Relaatiotietokantasanasto (1997-12), (toim. Harri Laine), Systeemityön standardointi- ja kehittämiskeskuksen tiedonhallintaryhmä (SYSTA / TIHA), Tietotekniikan kehittämiskeskus ry, Helsinki, 1997. *http://www.cs.helsinki.fi/relaatiosanasto/*
05.05.2003.

[Teste, 2000] Olivier Teste. Towards conceptual multidimensional design in decision support systems, *Universite Paul Sabatier –IRIT/SIG*.

[Thomsen, 1997] Erik Thomsen. *Solutions Building Multidimensional Systems*, John Wiley and Sons, Inc., USA, 1997.

[Ullman, 1989] Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume II*, Computer Science Press, ISBN 0-7167-8162-X(v. 2).

[Vassiliadis ja Sellis, 1999] Panos Vassiliadis, Timos Sellis. A survey of logical models for OLAP databases, *National Technical University of Athens, Department of Electrical and Computer Engineering Computer Science Division, Knowledge and Database Systems Laboratory*.

[W3C, 2000] Extensible markup language (xml) 1.0 (second edition).
http://www.w3.org/TR/REC-xml
07.08.2003.

[W3C, 1999] WC3. XPath language (XPath) Version 1.0.
http://www.w3.org/TR/xpath
07.08.2003.

[Widom, 1995] Jennifer Widom. Research problems in data warehousing, *Proceedings of the 1995 conference on International Conference on Information and Knowledge Management, 1995*, 25–30.
http://www.acm.org/pubs/articles/proceedings/cikm/221270/p25-widom/p25-widom.pdf
06.12.2000.

[Winter, 2000a] Winter Richard. SQL-99's new OLAP functions. *Intelligent Enterprise, January 20, 2000*, Volume **3,** Number 3.
http://www.intelligententerprise.com/000120/scalable.shtml?scale
05.02.2003.

[Winter, 2000b] Winter Richard. The extra mile. *Intelligent Enterprise, June 26, 2000*, Volume **3,** Number 10.
http://www.intelligententerprise.com/000626/scalable.shtml?scale
05.02.2003.