

Tehnologia OLAP

Prep. Daniela-Ioana SANDU, prep. Elena POSDARIE
Catedra de Informatica Economica, A.S.E. Bucuresti

OLAP enables analysts, managers and executives to gain insight into data through fast consistent, interactive access to a wide variety of possible views of information. OLAP transforms raw data to useful information so that it reflects the real factors affecting or enhancing the line of business of the enterprise. Nowadays many corporations adopted databases for their mission-critical data needs, but the queries needed to summarize the data and realize the management reports are extremely slow because of the large amount of data. The solution is to separate the data used for reporting and decision making from the operational systems into a datawarehouse designed and built to house this kind of data.

In this article we proposed to present the general terms used in OLAP. The examples are made using the Analysis Services component of Microsoft SQL Server 2000.

Keywords: OLAP, cube, fact table, measure, dimension, aggregation, members, cells, ROLAP, HOLAP, MOLAP.

Introducere

On Line Analytical Processing (OLAP) este tehnologia care permite utilizatorilor sa analizeze o baza de date larga sau un depozit de date. Baza de date a unui sistem OLAP este proiectata pentru stocarea datelor de tip static. O unitate de stocare OLAP este o unitate multidimensionala numita *cub* - spre deosebire de denumirea de *tabela* din teoria relationala. Cubul permite stocarea datelor agregate si ierarhice. Un cub este format din mai multe *dimensiuni* - pâna la 64 - iar agregarea datelor se face dupa attribute denumite *masuri* (*measures*). O dimensiune poate avea mai multe *nive-luri*.

De exemplu, daca ne intereseaza vânzarile la un anumit produs pe un anumit semestru al anului, cubul OLAP ne permite sa defalcam/agregam aceste vânzari pe luni, săptămâni si zile.

Definitie. Un cub poate fi definit în astfel:

1. O structura de memorare multidimensională folosită în OLAP care crește viteza de regasire a datelor oferind acces facil la diferite niveluri ale unei ierarhii de date.
2. Un subset de date dintr-un depozit de date, stocat într-o structură multidimensională.

Principalul avantaj în folosirea cuburilor este oferit de raspunsul rapid la cererile complexe datorita modului de organizare a datelor pe dimensiuni si masuri permitând si agregarea acestora.

Pentru a înțelege mai usor notiunea de cub, sa consideram ca managementul unui magazin de desfacere doreste sa cunoasca evolutia vânzarilor de mere din Detroit între trimestrele 1 si 3. Se vor construi 3 dimensiuni: "Product_dim", "Time_dim", "Location_dim" si o masura: "Sales_fact".

Product_dim	Time_dim	Location_dim	Sales_fact
ProductKey	TimeKey	LocKey	ProductKey
Grapes	Quarter 1	Atlanta	TimeKey
Cherries	Quarter 2	Denver	LocKey
Melons	Quarter 3	Detroit	Sales
Apples	Quarter 4		Units Sold
Pears			
All			

Fig.1. Trei dimensiuni si tabela "fact_table"

Dimensiunea "Product_dim" are ca membri toate produsele distribuite de firma. Membrii dimensiunii "Time_dim" sunt trimestrele anului, iar ai dimensiunii "Location_dim" localitatile de distributie. Masurile cubului sunt vânzările "sales" si unitatile vândute "units sold". O masura poate fi orice atribut al depozitului de date cu singura restrictie ca acest atribut trebuie sa fie o data numerica care poate fi însumata, agregata.

Deși denumirea înseala, un cub poate avea până la 64 de dimensiuni cu tot cu dimensiunea masurilor. Fiecare celula a cubului contine o valoare ce reprezinta intersectia dimensiunilor, fiind de fapt o valoare agregata.

Exemplu: o celula ce contine vânzările de mere din Detroit în trimestrul 3.

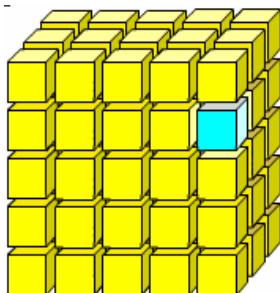


Fig.2. O celula a unui cub

Componentele cubului

Componentele unui cub sunt dimensiunile, membrii, celulele, ierarhiile si proprietatile. În cadrul unui cub o dimensiune descrie o axa individuala în cub reprezentând o

perspectiva din care pot fi vizualizate datele. De exemplu, vânzările pot fi vazute dupa data vânzării sau dupa produsul vândut.

Membrul reprezinta o valoare anume a dimensiunii. De exemplu dimensiunea "product_dim" contine membrii Grapes, Cherrie, Melons, Apples, Peras si All. Membrul "all" reprezinta totalitatea membrilor dimensiunii.

Fiecare celula este identificata de niste coordonate si anume câte un membru din fiecare dimensiune. Daca avem în vedere un cub bidimensional care reprezinta vânzările unei bacanii, axa Ox reprezinta data vânzării, axa Oy produsul vândut iar celula, intersectia dintre cele 2 axe, cantitatea vânduta.

În cadrul fiecărei dimensiuni se pot defini ierarhii, adica grupari de membri. Un exemplu la îndemână este gruparea datelor în saptamâni, a saptamânilor în luni, a lunilor în trimestre iar a trimestrelor în ani.

Când se defineste o ierarhie se foloseste notiunea de nivel. Un nivel descrie ordinea membrilor, de la nivelul care contine cele mai multe grupari de date până la cel mai detaliat nivel. De exemplu, daca magazinul nostru are o dimensiune timp formata din an, trimestru, luna si saptamâna, celula de la intersectia "Grapes (struguri)" cu "Ianuarie 1998" va contine totalul vânzărilor de struguri pentru Ianuarie 1998. Vânzările totale pentru fiecare produs vor fi stocate pentru fiecare an, trimestru, luna si saptamâna.

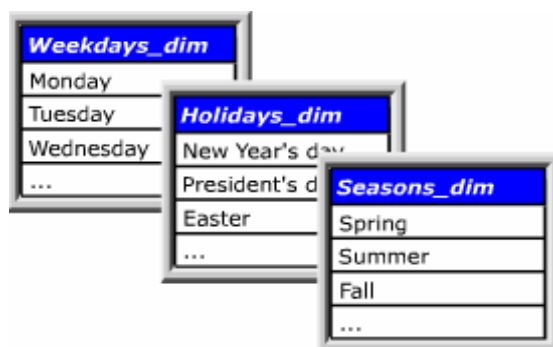


Fig.3. Dimensiuni din cadrul unui cub

Unele dimensiuni pot contine la rândul lor mai multe dimensiuni. Valorile din dimen-

siunea de timp pot fi agregate dupa ziua saptamânii, dupa zilele de sarbatoare, sau

dupa anotimpuri. Ierarhiile multiple adauga noi puncte de intersectie cu celelalte dimensiuni.

Proprietatile reprezinta attributele unui membru al unei dimensiuni.

Time	Start	End
Quarter 1	July 1	September 30
Quarter 2	October 1	December 31
Quarter 3	January 1	March 31
Quarter 4	April 1	June 30

Fig.4. Proprietatea Quarter (trimestru)

De exemplu, primul trimestru al unei companii poate începe la 1 Iulie si poate sfârși la 30 Septembrie. Fiecare nivel al unei ierarhii poate avea un set diferit de proprietati, dar un membru al unui nivel trebuie sa aiba proprietati similare cu ale celorlalti membri ai nivelului, desi proprietatile vor avea valori diferite. De exemplu, la nivelul "trimestru" fiecare nivel va avea date de început si de sfârșit diferite.

Etape în crearea unui cub

Primul pas în crearea unui cub este identificarea (daca exista)/ crearea (daca nu exista) a bazelor de date si a surselor de date.

La proiectarea unui cub se identifica întâi sursele de date si tabela de masuri: "fact table".

Masurile pe care le alegem depind de tipul de informatii cerute de utilizator. Fiecare masura este creata dintr-o coloana tabeli "fact table" (trebuie sa avem grija când cream aceasta tabela, deoarece ea trebuie sa contina toate informatiile care la un moment dat ar putea deveni masuri). Câteva masuri foarte des folosite sunt vânzările, costurile, productia. Masurile unui cub nu pot fi alese decât dintr-o singura tabela, (tabela de masuri). O masura poate fi si o expresie – mai multe coloane ale tabeli pot fi combinate într-o expresie pentru a forma o masura noua. De exemplu: profitul=venit-cheltuieli. SQL Server permite utilizarea atât a coloanelor însumabile cât si a celor neînsumabile drept masuri. Coloanele însumabile sunt coloanele numerice a caror însumare are o anumita semnifi-

catie. Vânzările zilnice pot fi însumate în vânzări săptămânale sau lunare.

Coloanele neînsumabile nu pot fi folosite drept masuri într-un cub deoarece agregarea lor nu are nici o semnificatie. O coloana numerica continând un identificator, (un numar de cont bancar) nu este însumabila. Coloanele de acest tip pot fi combinate cu alte functii – de exemplu functia *count* – iar rezultatul poate fi folosit ca o masura.

Dupa identificarea sursele de date trebuie create dimensiunile. O dimensiune poate fi creata pentru a fi folosita doar de un cub – *dimensiune privata* - sau pentru a fi folosita de mai multe cuburi ale bazei de date – *dimensiune partajata*.

Dimensiunile partajate tipice sunt dimensiunile de timp si cele geografice. De exemplu, daca o companie are un numar de puncte de desfacere în diverse tari, iar în cadrul fiecarei tari în diverse orase, ierarhia care defineste aceste locatii poate fi creata o singura data si apoi utilizata ori de câte ori este nevoie. Dimensiunile partajate permit atât evitarea duplicarii dimensiunilor private în cuburi diferite, cât si standardizarea procedurii de calcul a masurilor. Daca pentru toate departamentele organizatiei se foloseste aceeasi dimensiune partajata "profit" putem fi siguri astfel ca functia de calcul a profitului este aceeasi pentru toate departamentele.

Etapele de creare a unei dimensiuni cu Analysis Services sunt:

1. Se specifica daca dimensiunea se creeaza din una sau mai multe tabele. Daca dimensiunea se creeaza dintr-o singura tabela va fi o dimensiune de tip "star schema". Daca se creeaza din mai multe tabele va fi o dimensiune "snow flake", iar în acest caz toate tabelele trebuie sa provina din aceeasi sursa de date.
2. Se selecteaza tabela/tabelele din care se va crea dimensiunea.
3. Se alege tipul dimensiunii: standard sau dimensiune de timp.
4. Se identifica coloanele din tabele care vor deveni nivelurile dimensiunii.
5. Se stabileste daca dimensiunea va fi partajata sau privata.

Explozia datelor

Câmpurile agregate ale cuburilor pot îmbunătăți timpul de răspuns la cereri întrucât răspunsul la o cerere există deja înainte ca cererea să fie formulată. Agregările sunt stocate în cub la coordonate specificate de dimensiune. Un număr mare de agregări duce la un timp scăzut de răspuns, dar necesită mai mult spațiu de stocare. Numărul de agregări trebuie ținut sub control, altfel apare pericolul exploziei datelor. Creșterea în volum a datelor unui cub este exponențială cu numărul de valori agregate adăugate la cubul respectiv. De exemplu, pentru analiza vânzărilor unui produs folosim două dimensiuni: "timp" și "produs". Dimensiunea "timp" conține nivelurile zile, luni, trimestre și ani. Pentru fiecare membru al dimensiunii "produs" vom avea câte o valoare stocată în cub pentru fiecare membru al fiecărui nivel al dimensiunii "timp". Dacă avem 40000 de produse acest număr trebuie înmulțit cu numărul de zile dintr-un an (365) și așa mai departe pentru fiecare dimensiune în parte. Dacă ar trebui să mai adăugăm o dimensiune, cubul ar crește în mod exponențial, deoarece mărimea lui ar trebui multiplicată cu numărul de membri ai noii dimensiuni. Controlul

dimensiunii cubului se poate face agregând doar vânzarile zilnice și calculând în mod dinamic vânzarile lunare, anuale și pe semestre. Această strategie reduce explozia datelor și nu are nici un efect asupra vitezei de răspuns la cereri. Instrumentul SQL Server care permite agregarea datelor, Wizard-ul pentru Storage Design face și calculul câștigului de performanță pentru un anumit procent de agregări analizând numărul de niveluri pentru fiecare dimensiune și rata parinte-copil pentru fiecare nivel. În general 20% din toate agregările posibile asigură 80% din performanță. Storage Design oferă trei metode de control a agregărilor:

1. Mai întâi se stabilește spațiul de stocare disponibil iar instrumentul determină ce agregări va stoca.
2. Se stabilește procentul de performanță pe care dorim să-l obținem prin agregări și permitem tabelului cu agregări să ocupe spațiul de care are nevoie.
3. Determinarea manuală a performanței și spațiului fizic ocupat pe disc urmărind graficul "Performance vs. Size" pus la dispoziție de wizard cu oprirea acestuia când s-a atins raportul performanță/mărime optim.

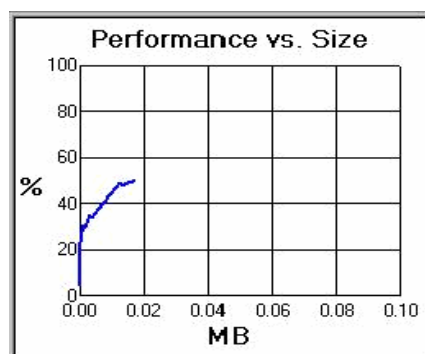


Fig.5. Wizard-ul Storage Designer din cadrul SQL Server 2000

Securitatea cuburilor

Securitatea cuburilor presupune autentificarea utilizatorilor, controlul drepturilor de acces și definirea și controlul rolurilor.

Autentificarea utilizatorilor se face la nivelul sistemului de operare fiind asigurat de Windows NT/2000 care folosește schemele normale de autentificare.

Controlul accesului la date se face prin acordarea de drepturi de acces atribuite utilizatorilor și rolurilor. Drepturile de acces sunt posibile numai pe sistemele de fișiere NTFS și sunt stocate în liste de control al accesului ACL (access control lists).

Accesul la cub poate fi controlat atribuind utilizatorului privilegii de Read sau Admin. Nivelul "Read" asigură acces read-

only permitând doar vizualizarea datelor si împiedicând modificarea si procesarea lor. Nivelul "Admin" ofera acces complet pentru a manipula si modifica un cub. În mod implicit, contul utilizator creat la instalarea SQL Server are privilegiile Admin.

Rolurile mapeaza conturile utilizator si grupurile din WindowsNT/2000 pe anumite drepturi de acces. Conturile utilizator si grupurile se definesc în WNT User Manager iar apoi pentru fiecare grup în parte se selecteaza rolurile care pot accesa cubul.

Structuri de memorare

La constructia unui cub se poate alege între trei structuri de memorare:

- OLAP Relational (ROLAP);
- OLAP Multidimensional (MOLAP);
- OLAP Hibrid (HOLAP).

Structura de memorare afecteaza atât performanta cât si spatiul fizic ocupat de cub.

• **MOLAP** (Multidimensional Online Analytical Processing). Structura pastreaza

atât o copie a datelor de baza cât si a agregarilor în structuri multidimensionale. Aceste structuri multidimensionale vor fi stocate în afara depozitului de date. Atunci când se interogheaza un cub, calculele sunt facute în cadrul cubului. Când se creeaza un cub în OLAP Services, daca nu se specifica structura de memorare se considera implicit o structura MOLAP. În cadrul unei structuri MOLAP toate masurile sunt în acelasi articol ceea ce reduce timpul de regasire datorita unor algoritmi specifici de compresie a datelor.

Structura MOLAP pastreaza date într-un format ce include segmente rând de 64KB. Structura este folosita atunci când datele sunt frecvent interogate si se cere viteza mare de raspuns. Desi spatiul ocupat pe disc va fi redus, cuburile MOLAP vor ocupa multa memorie interna datorita nivelurilor de agregare numeroase care duc la cresterea dimensiunii acestora.

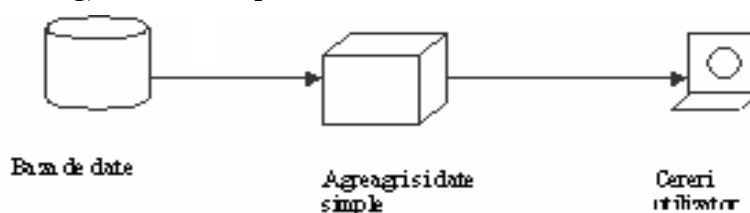


Fig.6. Structura de memorare MOLAP

• **ROLAP** (Relational Online Analytical Processing). Structurile ROLAP stocheaza atât baza de date cât si agregările în tabele ale unei baze de date relationale. Când utilizatorul interogheaza un cub, calculele sunt executate în cadrul acestor tabele. În loc sa pastreze o copie a datelor de baza, ROLAP acceseaza tabelele originale pentru a formula raspunsurile la cereri. Desi un cub este stocat într-o baza de date de tip warehouse datele de baza si agregările sunt gestionate de serviciile OLAP.

Structura ROLAP este formata din tabele ale bazei de date si din indecuri. Pentru fiecare nivel (atribut) al unei dimensiuni se creeaza o tabela iar fiecare masura este o

coloana în tabela. OLAP Services creeaza automat pentru fiecare nivel al dimensiunii un index.

Avantajele folosirii ROLAP sunt:

- eliminarea datelor duplicate;
- utilizarea agregarilor existente;
- scalabilitate;
- suporta SQL Server v2000, Oracle, Microsoft Access, ODBC;
- economiseste spatiul pe disc.

Este indicat sa se utilizeze ROLAP atunci când datele nu sunt interogate cu o frecventa mare.

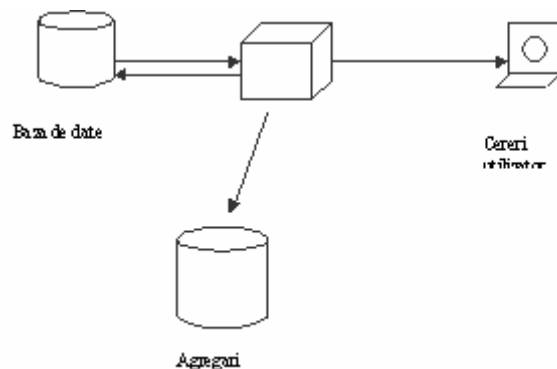


Fig.7. Structura de memorare ROLAP

• **HOLAP** (Hybrid Online Analytical Processing). Este o combinatie între MOLAP și ROLAP. HOLAP stochează datele în tabele relationale în aceeași bază de date în care sunt stocate și tabelele depozitului de date, însă agregările multidimensionale sunt stocate în afara bazei de date (depozitului de date). Când se execută o interogare asupra cubului calculele sunt executate în cadrul cubului.

Structura hibridă este foarte folosită când

au loc interogări frecvente ale datelor agregate pe baza unui volum foarte mare de date de bază.

O structură HOLAP regăsește rapid datele din cub utilizând același procesor de cereri folosit și de MOLAP însă ocupă mult mai mult spațiu decât acesta (MOLAP).

Structurile HOLAP evită duplicarea datelor deoarece datele de bază sunt în tabelele relationale.

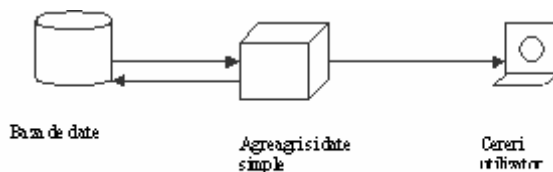


Fig.8. Structura de memorare HOLAP

Bibliografie

- Bain, Toni – *SQL Server 2000 Data-Warehousing with Analysis Services*, Wrox Press 2001.
- Patton, Robert – *Designing SQL Server 2000*, Syngress Publishing 2001.
- Seidman, Claude – *Data Mining with Microsoft SQL Server 2000*, Microsoft Press 2001.

- Speshoch, Carl – *Microsoft SQL Server 2000*, PrenticeHall 2002.
- Todman, Chris – *Designing a data warehouse*, PrenticeHall 2001.
- *Microsoft SQL 7.0 Data Warehousing on-line training kit* – Microsoft Press 1999.