**TEHNOLOGIA OLAP (ON-LINE ANALYTICAL PROCESSING)**

1. **INTRODUCERE.**

Conceptul de On-line Analytical Processing a aparut incepand cu anii 60-70 din dorinta de a modela prin functii analitice activitatile financiare. Primul limbaj multidimensional, *A Programming Language* (APL) a fost dezvoltat de firma IBM si utilizat pe mainframe-uri  inca din 1962, multe din conceptele acestuia fiind si astazi implementate in unele limbaje, cum ar fi Adaytum Planning si Lex 2000.

In 1993, E.F.Codd observa diferenta de procesare dintre modelele relationale si cele multidimensionale si introduce termenul de OLAP fundamentat pe 12 reguli, pe care sistemele de analiza multidimensionala ar trebui sa le respecte. Intr-un articol in revista Computerworld, Codd mentioneaza faptul ca: “oricat de puternice ar fi pentru utilizatori sistemele relationale, acestea nu au fost proiectate pentru a asigura functii puternice de sinteza, analiza siconsolidare a datelor, functii cunoscute colectiv sub denumirea de analiza multidimenionla a datelor”.

In 1995 se infiinteaza Consiliul OLAP, un consortiul al firmelor dezvoltatoare de produse OLAP,cu rolul de a standariza aceste tehnologii prin stabilirea unor standarde deschise (OLAP API). Consiliul OLAP a publicat urmatoarea definitie [OLAP95]:

***“On-Line Analytical Processing este o tehnologie software ce permite analistilor, managerilor si persoanelor cu functie de conducere sa analizeze datele printr-un acces rapid, consistent si interactiv si sa le vizualizeze intr-un mod cat mai variat. “***

        Tehnologia OLAP este caracterizata de o dinamica analiza multidimendionala in sprijinul utilizatorului final printr-o serie de activitati:

       Aplicarea de formule si modele asupra dimensiunilor si ierarhiilor;

       Previziuni pe perioade diferite de timp;

       Analiza in adancime (*drill-down*);

       Extragerea unui subset de date pentru vizualizare;

       Rotatii in cadrul dimensiunilor;

Tehnologia OLAP reprezinta o modalitate de prelucrare si analiza dinamica si avansata a datelor, oferind decidentilor posibilitatea de a obtine propria perspectiva asupra datelor, de creare flexibila si obtinere directa a  situatiilor centralizate si sintetice, dar si cu posibilitatea de navigare in detaliu, cu facilitati de previzionare si simulare a unor situatii viitoare, fiind o solutie eficienta de analiza a datelor din depozitele de date.

     Sistemele OLAP ajuta utilizatorul sa sintetizeze informatiile organizatiei printr-o vizualizare comparativa si personalizata ca si printr-o analiza a datelor istorice folosind scenarii de tipul “ce se intampla daca?” (“what-if?”). Acestea se obtin cu ajutorul serverului de OLAP special conceput pentru manipularea structurilor de date multidimensionale.   Arhitectura serverului si structura datelor sunt optimizate pentru regasiri rapide, analize ad-hoc, calcule flexibile si transformari ale datelor.

     Spre deosebire de sistemul operational care functioneaza pe baza unor proceduri prestabilite (exista o gama relativ limitata de tranzactii operate de o organizatie) un sistem de analiza on-line (OLAP) ofera suport pentru o varietate de cerinte care nu pot fi prevazute decat intr-o mica masura.

     Sistemele OLAP ajuta managerii in urma analizei datelor sa-si fundamenteze deciziile astfel incat sa-si comercializeze mai bine produsele, sa-si planifice productia intr-un mod mai eficient, sa controleze costurile, sa descopere evolutiile viitoare ale unor factori. OLAP poate fi utilizat in orice domeniu al afacerilor: analiza vanzarilor si studii de piata, evolutii ale indicatorilor financiari ai intreprinderii, ca suport de decizii: previziuni ale veniturilor si cheltuielilor. Se poate studia volumul vanzarilor in functie de produse, arii geografice si timp, etc.

1. ***CERINTELE FUNCTIONALE ALE SISTEMELOR OLAP***

Datorita particularitatilor cerintelor de raportare ale managerilor,  sistemele OLAP trebuie sa prezinte urmatoarele caracteristici:

* ***Analiza dinamica a datelor*** - Aceasta cere existenta diferitelor instrumente de analiza si implica dimensiuni multiple concentrandu-se asupra manipularii modelelor de date ale intreprinderii. Analiza dinamica a datelor ofera o intelegere mai buna asupra schimbarilor intervenite in cadrul afacerilor intreprinderii si pot fi utilizate pentru identificarea solutiilor, pentru planificarea tactica si strategica la nivelul intreprinderii.
* ***Acces rapid la date****-*Aplicatiile OLAP necesita un volum mare de date care trebuie sa fie acesate foarte rapid, ceea ce presupune de obicei ca acestea sa fie stocate in structuri separate, optimizate care pot fi accesate fara sa afecteze raspunsul din sistem.
* ***Surse de date multiple*** *-*Majoritatea aplicatiilor OLAP cer surse de date din sisteme multiple, incluzand surse externe si aplicatii realizate in medii de programare diferite. Procesul fuzionarii acestor surse multiple poate fi foarte complex datorita sistemelor de codificare diferite si calitatii diferite a datelor.
* ***Sincronizarea surselor de date*** *-*Daca datele dintr-o aplicatie OLAP provin din mai multe baze de date, este foarte probabil ca acestea sa fie modificate la cicluri diferite. Ca analiza sa fie bazata pe date consistente, datele trebuie incarcate impreuna in depozitele de date.
* ***Analiza istorica*** *-*Majoritatea aplicatiilor OLAP includ timpul ca o dimensiune, si multe rezultate utile sunt obtinute din analize de serii de timp. Dar pentru ca acest lucru sa fie util ar fi necesar ca datele sa fie stocate intr-un depozit sau data mart pe o perioada de cel putin 2-3 ani. Aceasta presupune un effort de localizare a datelor istorice si in general, trebuie ajustate datorita modificarilor din organizatie si a structurilor ierarhice.
* ***Grad de generalizare ridicat*** *–*Cerintele de analiza ale managerilor impun ca informatiile sa fie grupate, agregate si reprezentate cat mai sintetic. Pentru a creste eficienta si a reduce timpul de raspuns, de obicei este util sa stocam datele fuzionate si ajustate la un nivel superior de agregare, dand insa posibilitatea managerilor sa poata vedea la cerere si nivelele de detaliu.

In lucrarea [THOM02] Erik Thomsen structureaza cerintele functionale ale sistemelor OLAP in doua mari categorii si anume *cerinte logice si cerinte fizice*.

**Cerintele logice** se refera la modalitatile de prelucrare a datelor din dimensiuni, la structurarea datelor si flexibilitatea sistemelor, astfel sunt identificate urmatoarele cerinte:

* ***Structurare completa a dimensiunilor prin ierarhizare*** – se refera la capacitatea unui sistem OLAP de a modela dimensiunile existente in mediul organizational pe diferite niveluri in functie de anumite ierarhii, pornind de la nivelul cel mai detaliat pana la un nivel superior, generalizat si abstarctizat.
* ***Realizarea eficienta a calculelor si prelucrarilor*** *–*sistemul OLAP trebuie sa implementeze functionalitati  complexe de analiza, de comparare si previzionare a  datelor, pe langa abilitatile de agregare si segmentare a acestora.
* ***Flexibilitate*** *–* modul de prezentare a datelor rezultate in urma prelucrarilor trebuie sa tina cont de utilizator. Interfata poate fi grafica, tabelara, complexa in functie de cerinte. Flexibilitatea se refera si la posibilitatile de modificare a modelului de catre utilizator fara a fi necesara re-proiectarea intregului sistem.
* ***Independenta reprezentarilor fata de structura modelului*** *–* referitor la aceasta cerinta, un sistem OLAP trebuie sa ofere posibilitatea modificarii reprezentarii fara a afecta structura datelor.

**Cerintele fizice** ale sistemelor OLAP sunt referitoare la accesul si timpul de raspuns al sistemului si la suportul multiutilizator al acesuia:

* ***Acces rapid si direct*** *–*principalul obiectiv al sistemelor OLAP este de a realiza analize ad-hoc pe un volum mare de date. Accesul la aceste analize artrebui sa se realizeze direct de catre utilizatorii finali, fara interventii suplimentare, intr-un timp cat mai scurt.
* ***Suport multiutilizator*** *–*datoritavolumului de date si a faptului ca acestea sunt centralizate si prelucrate dintr-un depozit de date, iar asupra depozitului au acces diversi utilizatori, sistemul OLAP trebuie sa permita accesul concurential si distribuit la prelucrarile analitice.

O alta abordare a cerintelor sistemelor OLAP este realizata chiar de parintele conceptului, E.F. Codd in 1993 prin intermediul unui set de 12 reguli. Mai tarziu, in 1995, setul a fost extins la 18 reguli ce surprind caracteristicile sistemelor OLAP. Voi prezenta mai jos acest set de reguli, dupa cum urmeaza:

***A. Caracteristici de baza***

***Regula 1: O viziune conceptuala multidimensionala***

Viziunea conceptuala a modelelor OLAP trebuie sa fie multidimensionala bazata pe viziunea sau modelul existent in organizatie.

***Regula 2: Manipularea intuitiva a datelor***

Sistemele OLAP trebuie sa permita operatii intuitive si flexibile de manipulare a datelor, cum ar fi navigarea penivelurile ierarhiilor (operatii de drill down, drill up, drill across), analize pe sectiuni din date, etc.

***Regula 3: Accesibilitate***

Sistemele OLAP trebuie sa ofere acces la o singura viziune logica a datelor din organizatie. Sursele de date, in modelul OLAP, trebuie sa fie transparente utilizatorilor.

***Regula 4: Surse de date variate***

Un sistem OLAP trebuie sa fie capabil sa lucreze cu date stocate fie in baze de date multidimensionale (MOLAP) cat si in baze de date relationale (ROLAP) sau chiar sisteme hibride (HOLAP).

***Regula 5: Modele de analiza OLAP***

Sistemele OLAP trebuie sa suporte patru modele de analiza: explicativ, direct, contemplativ si formativ in sensul ca un trebuie sa permita cel putin realizarea rapoartelor parametrizate, analize de tip “ce se intampla daca..?”, operatii de tip drill-down/roll-up si slice/dice.

***Regula 6: Arhitectura client/server***

Orice sistem OLAP ar trebui sa fie bazat pe o arhitectura client/server, oferind accesul utilizatorilor prin intermediul unui client, iar prelucrarea multidimensionala sa fie realizata de un server specializat.

***Regula 7: Transparenta***

Accesul la sursele de date eterogene ar trebui sa fie transparente pentru utilizatori, iar analiza datelor sa poata fi realizata si prin intermediul diverselor instrumente client ca: grafice, calcul tabelar, procesoare de text, etc.

***Regula 8: Suport multiutilizator***

Sistemele OLAP trebuie sa asigure acces concurent si distribuit la sursele de date,  fiind asigurate insa integritatea si securitatea acestora.

***B.Caracteristici speciale***

***Regula 9: Denormalizarea datelor***

Prelucrarea datelor intr-un mediu OLAP nu trebuie sa afecteze sursele externe din care provin acestea. Procesarea colectiilor mari de date, actualizate periodic trebuie sa fie realizata prin intermediul unor legaturi persistente cu sursele externe de date, pentru a asigura sincronizarea intre acestea si cubul de date. Deoarece sistemele OLAP sunt in general separate de sistemele sursa, legaturile servesc ca functii de transformare ce precizeaza modul de transformare a datelor din tabele sau foi de calcul tabelar in date multidimensionale. Legaturile pot descrie relatii structurale, atributele membrilor sau continutul cuburilor si pot fi unidirectionale (de citire) sau bidirectionale (citire/scriere).

***Regula 10: Stocarea rezultatelor generate de sistemul OLAP***

Datele supuse analizei trebui stocate si prelucrate separat de sursele relationale sau de fisierele din care provin datorita diferentelor existente intre modele si a cerintelor de procesare.

***Regula 11: Manipularea valorilor lipsa***

Termenul de imprastiere a fost utilizat cu semnificatia de valoare lipsa, valoare inaplicabila si valoare zero. Primele doua cazuri sunt considerate date invalide (conceptul de null). Al treilea caz, unde termenul de imprastiere a fost utilizat cu semnificatia de existenta a multor valori zero, este un caz special al modului in care este stocat un numar mare de valori care se repeta, in cazul de fata valoarea zero. Insa valoarea zero este valida ca orice alt numar. Confuzia a aparut deoarece in aplicatiile OLAP apar un numar mare de valori zero, precum si volume mari de date lipsa si invalide. Tehnicile pentru optimizarea fizica a stocarii unui numar mare de valori repetate sunt similare si uneori aceleasi cu tehnicile pentru optimizarea fizica a stocarii de volume mari de date lipsa si invalide. Totusi valorile lipsa si cele invalide nu sunt date valide. Ele nu pot fi tratate in acelasi mod ca orice alta valoare. De aceea, sunt necesare tehnici speciale pentru aceste cazuri. [MUNT04]

***Regula 12: Modul de tratare a valorilor lipsa***

Tratamentul impropriu al valorilor null poate cauza calcule incorecte. Acuratetea calculelor este de o importanta cruciala pentru analiza oricarui set de date, indiferent ca este sau nu multidimensional. Problema tratarii datelor imprastiate este una foarte importanta si este frecvent dezbatuta in domeniul bazelor de date. Cele doua tipuri de date (lipsa si invalide) trebuie totusi sa fie tratate individual, deoarece ele afecteaza calculele in diferite moduri [MUNT04]

***C. Modul de prezentare a datelor***

***Regula 13: Flexibilitatea rapoartelor***

Modul de prezentare a datelor supuse analizei trebuie sa fie accesibil utilizatorilor astfel incat acestia sa poata aranja cu usurinta datele pe diverse dimensiuni pe axele disponibile.

***Regula 14: Performanta raportarii***

Dimesiunea sau modul de organizare a datelor nu ar trebui sa influenteze performanta in raportare. Exista insa doi factori importanti care afecteaza performanta raportarii si anume: modul in care sunt realizate calculele (antecalculate sau la momentul interogarii) si locul unde sunt procesate calculele (client/server). Acesti factori sunt mai importanti decat dimensiunea bazei de date, numarul de dimensiuni sau complexitatea raportului.

***Regula 15: Ajustarea automata a nivelului fizic***

Sistemele OLAP ar trebui sa-si modifice automat schema fizica a bazei de date in functie de tipul modelului logic si de volumul datelor.

***D. Controlul dimensiunilor***

***Regula 16: Dimensionalitate generica***

Dimensiunile proiectate trebuie sa fie echivalente structural si operational, adica sa permita ierarhii multiple si toate tipurile de operatii multidimensionale si in acelasi timp sa poate fi actualizate (adaugarea/stergerea unui membru, adaugarea/stergerea unei ierarhii, modificarea unui membru/ierarhie etc).

***Regula 17: Dimensiuni si niveluri de agregare nelimitate***

Codd recomanda utilizarea un numar maxim de 15-20 de dimensiuni. In practica insa exista o multitudine de alte cerinte si limitari ale instrumentelor OLAP, astfel incat problema numarului maxim de dimensiuni poate deveni o cerinta minora, cesemnificativa.

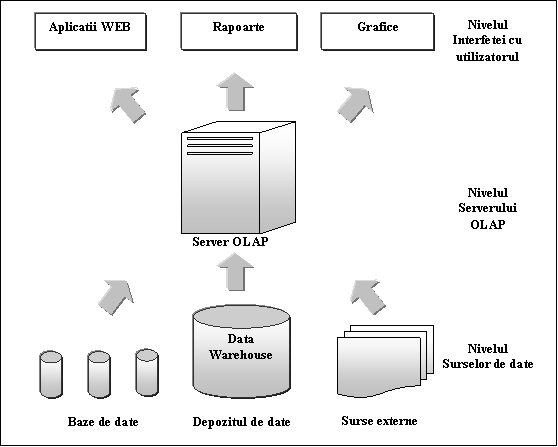
***Regula 18: Operatii intre dimensiuni nerestrictive***

Sistemele OLAP ar trebui sa permita relizarea de operatii intre diverse dimensiuni, fara restrictii.

1. **ARHITECTURA SISTEMELOR OLAP**

Datorita caracteristicilor functionale si a particularitatilor sistemelor existente in cadrul fiecarei organizatii se disting mai multe***tipuri de arhitecturi ale sistemelor OLAP*.** Acestea difera in functie de modalitatea de stocare a datelor si de tipul prelucrarii acestora, insa generalizand se pot identifica 3 niveluri ale arhitecturii: nivelul surselor de date, al serverului OLAP si al prezentarii datelor sau interfata cu utilizatorul.

Figura urmatoare prezinta rolul serverului OLAP in extragerea datelor din diferite surse si prezentarea informatiilor obtinute in diverse moduri pe cele trei niveluri mentionate anterior.



*Arhitectura Sistemelor OLAP*

Multe confuzii exista in legatura cu arhitecturile OLAP si termeni ca ROLAP, HOLAP, DOLAP. De fapt exista mai multe optiuni in care datele OLAP ar putea fi stocate si unde ar putea fi procesate. Sunt mai multe variante rezultate in urma combinatiilor intre modalitatile de stocare si cele de prelucrare a datelor din sistem.

In functie de ***modalitatea de*** ***organizare si stocare a datelor*** pot exista trei optiuni:

1. ***Fisiere client*** *-*in acest caz, extragerile de date relativ mici sunt stocate local pe calculatorul client sub forma de fisiere (de exemplu foi de calcul) care pot fi utilizate direct, prelucrate si transformate pentru analiza. In acest caz exista o serie de limitari cum ar fi: voumul redus de date care poate fi prelucrat, timpul relative mare de procesare a informatiilor, securitate redusa, prelucrari rudimentare datorate inexistentei unor functii puternice de analiza multidimensionala.
2. ***Baze de date relationale*** – aceasta varianta se recomanda in cazul in care datele provin dintr-un SGBD relational iar depozitul de date a fost implementat utilizand un model relational sau este implementat ca deposit de date virtual. In acest caz, datele ar fi stocate intr-o structura denormalizata cum ar fi o schema stea sau una din variantele sale: o baza de date normalizata nu ar fi potrivita pentru performante.
3. ***Baze de date multidimensionale*** *-* in acest caz datele sunt stocate intr-un depozit de date pe un server dedicate, denumit server multidimensional. In acest caz putem vorbi de un deposit de date format din obiecte multidimensionale asupra carora pot fi aplicate direct operatiile multidimensionale. Sarcina realizarii acestor operatii cade in seama serverului multidimensional. Datele sunt extrase din surse diverse (baze de date relationale, fisiere), transformate si incarcate in tabelele de fapte si dimensiuni, aggregate pe diverse nivele, preprocesate si pregatite pentru analiza.  Este varianta optima datorita avantajelor oferite: capacitatea de procesare a unui volum mare de date,  existenta procesului ETL pentru transformarea si incarcarea datelor, implementarea operatiilor la nivel de server multidimensional optimizat pentru analiza.

        Asa cum exista trei modalitati de stocare pentru datele OLAP, tot trei optiuni sunt si pentru ***procesarea datelor***. Asa cum se va observa, operatiile multidimensionale nu trebuie neaparat sa aiba loc unde sunt stocate datele din acest motiv exista urmatoarele variante:

       ***Nucleul SQL*** *-*Aceasta este departe de a fi o optiune optima pentru a efectua calcule multidimensionale complexe, chiar daca datele OLAP sunt stocate intr-o baza de date relationala. Limbajul SQL nu are implementate facilitatile de a efectua direct calcule multidimensionale si sunt necesari mai multi pasi pentru a obtine aceleasi rezultate cu cele obtinute prin aplicarea functiilor si operatiilor multidimensionale.

       ***Motorul client multidimensional*** *-*Presupunand ca majoritatea utilizatorilor au sisteme relativ puternice, se pot efectua local unele operatii multidimensionale, de exemplu  pivotarea sau filtrarea in cadrul foilor de calcul. Insa aceasta varianta presupune cunostinte avansate in domeniu si lasa practic sarcina construirii si aplicarii functiilor de analiza pe seama utilizatorului final.

       ***Motorul server multidimensional*** - Aceasta este alegerea optima pentru efectuarea operatiilor multidimensionale intr-o aplicatie OLAP client/server. Executia operatiilor multidimensionale de catre serverul dedicat degreveaza sistemul client si utilizatorul final de sarcina construirii acestora, asigura accesul concurent la aceleasi resurse, iar procesarea cererilor de analiza se realizeaza in timp real si informatiile sunt disponibile pentru vizualizare prin intermediul unor interfete standardizate si prietenoase pentru utilizatorii finali.

In functie de optiune de stocare si procesare a datelor teoretic sunt posibile noua arhitecturi de baza, din care doar sase au sens. Aceste combinatii precum si cateva dintre produsele software care le utilizeaza sunt prezentate in tabelul de mai jos [HUHA99]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **OPTIUNI DE STOCARE A DATELOR** | | |
| **OPTIUNI DE PROCESARE** | **Fisiere** | **SGBDR** | **Baza de date multidimensionale** |
| **Nucleul SQL** | 1 | **2**  Cartesis Magnitude MicroStrategy | 3 |
| **Motorul client Multidimensional** | **4**  Brio.Enterprise BusinessObjects Cognos PowerPlay Oracle Personal Express iTM1 Perspectives  Microsoft Excel | **5**  Oracle Discoverer Informix MetaCube | **6**  Comshare FDC Dimensional Insight Hyperion Enterprise Hyperion Pillar PwC CLIME |
| **Motorul server Multidimensional** | 7 | **8**  Crystal Holos (ROLAP mode) IBM DB2 OLAP Server  CA EUREKA:Strategy  Longview Khalix Informix MetaCube Speedware Media/MR Microsoft Analysis Services Pilot Analysis Server  Sagent Applix iTM1 WhiteLight  Oracle Express (ROLAP mode)  Oracle Warehouse Builder  Oracle Discoverer | **9**  SAS CFO Vision  Crystal Holos Comshare Decision Hyperion Essbase Gentia Speedware Media/M Microsoft Analysis Services PowerPlay Enterprise Server  Pilot Analysis Server Applix iTM1  Oracle Express  Oracle Warehouse Builder  Oracle Discoverer |

*Tabel 2.1: Variante de implementare ale sistemelor OLAP*

     Arhitecturile cele mai utilizate dintre aceste tipuri de combinatii sunt urmatoarele:

* *OLAP relational (ROLAP)* (2, 5, 8) din care *OLAP hibrid (Hybrid OLAP sau HOLAP)* (5, 8)
* *OLAP multidimensional (MOLAP)*(6, 9) din care *OLAP client (Desktop OLAP sau DOLAP)* (6)
* *OLAP client (DOLAP)* (4)

1. **MODELUL DE DATE MULTIDIMENSIONAL**

Pentru definirea unui model de date este necesara specificarea urmatoarelor elemente:

       ***Structura modelului*** constituita din obiectele modelului precum si relatiile dintre ele;

       ***Operatorii***care actioneaza asupra structurii;

       ***Restrictiile de integritate*** formate din totalitatea de regului si constrangeri impuse modelului pentru asigurarea corectitudinii datelor.

***Structura modelului*** contine in principal obiectele referitoare la tabele de fapte cu atributele de tip masuri sau metrici, tabelele de tip dimensiune in care regasim nivele ierarhice, attribute de descriere, etc. Aceste obiecte vor fi prezentate in continuare.

In cadrul modelului multidimensional se intalnesc mai multe ***tipuri obiecte*** care prezinta o importanta deosebita in analiza [KIRE98]:

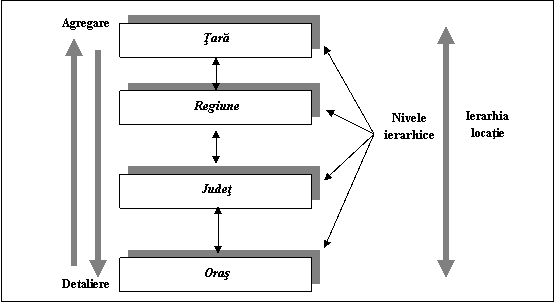
***Dimensiunile* –**reprezinta structuri compuse atribute structurate pe diverse niveluri ierarhice in functie de care sunt grupate datele. Aceste atribute sunt de obicei descriptive si sunt folosite ca sursa pentru restrictii si pentru randurile din rapoarte. Sunt considerate tabele secundare datorita dimensiunilor reduse. Consiliul OLAP defineste conceptul de dimensiune ca fiind  “*un atribut structural al unui cub ce consta dintr-o lista de membrii, pe care utilizatorii ii percepe ca fiind de acelasi tip (de exemplu toate lunile, trimestrele, anii formeaza dimensiunea Timp). Dimensiunile reprezninta un mod foarte concis, intuitiv de organizare si selectare a datelor pentru explorare si analiza.”* [OLAP95]. Datele sunt de obicei colectate la nivelul cel mai detaliat si apoi agregate pe nivelele superioare pentru analiza.

In cadrul dimensiunilor se regasesc si conceptele de ierarhie, nivel, atribut, concepte care vor fi prezentate in continuare:

***Ierarhiile* –**sunt structuri logice utilizate pentru ordonarea nivelelor de reprezentare a datelor. Sunt utilizate si pentru definirea cailor de navigare in interiorul datelor. Nivelele ierarhice sunt utilizate de instrumentele de analiza OLAP permitand detalierea graduala a datelor. Tot in definitiile date de Consiliul OLAP se mentioneaza ca  “*membrii dimensiunilor pot fi organizati pe baza relatiilor de tip parinte-copil, unde un membru parinte reprezinta agregarea membrilor copil. Rezultatul este o ierarhie si relatiile parinte-copil sunt relatii ierarhice”.* [OLAP95]

Ierarhia definita pe o dimensiune determina aranjarea membrilor dimensiunii intr-o configuratie piramidala. pe orizontala se plaseaza rezultatele corespunzatoare masurilor de pe acelasi nivel in ierarhia dimensiunii, iar pe verticala se plaseaza rezultatele avand niveluri diferite in ierarhia dimensiunii.

***Nivelele* –** reprezinta pozitii in cadrul ierarhiilor (Vezi figura ce urmează). De exemplu dimensiunea *Timp* poate avea trei nivele de ierarhizare: an, trimestru si luna. Nivelele se structureaza in functie de ierarhie de la general la specific, radacina fiind reprezentata de nivelul superior, cel mai inalt al ierarhiei. Relatiile intre diferite nivele sunt relatii de tipul parinte-copil. Se pot defini ierarhii in care datele fiecarui nivel sunt agregate la un nivel superior sau se pot sari anumite nivele care sunt independente.



***Ierarhii si nivele***

***Atribute* –**dimensiunile contin atribute care reprezinta calificative specifice. Orice atribut se asociaza unei singure dimensiuni, iar o dimensiune se poate exprima prin mai multe atribute. Cu cat aceste atribute sunt mai descriptive cu atat depozitele de date vor fi mai performante.

***Tabelele de fapte*** – sunt tabelele centrale. Acestea contin atribute de tip masuri (metrici) si chei externe catre tabelele dimensiuni. Faptele sunt de obicei date numerice care pot fi insumate si analizate pe diferite nivele.

***Metricile (masurile)***corespund atributelor (faptelor) din tabelele de fapte si sunt de regula de natura numerica (de exemplu: volumul vanzarilor, costurile, stocurile disponibile). Aceste variabile au sens numai in contextul unor anumite dimensiuni. Masurile reprezinta valorile centrale care sunt analizate prin cubul de date. Valoarea masurii este calculata pentru un punct dat prin agregarea datelor corespondente perechii respective valoare-dimensiune, diferite pentru punctul dat.

Masurile pot fi clasificate dupa *modalitatea de calcul* in *masuri de baza* care se regasesc sub forma atributelor din tabelele de fapte si care provin din sursele de date si  *masuri derivate (virtuale)* care se obtin prin combinarea masurilor de baza si care in tabelele de fapte au precizata formula de calcul prin care se obtin.

Masurile pot fi organizate in trei categorii bazate pe *tipurile de functii agregate utilizate*: *distributive, algebrice, holistice*.

***Masurile distributive*** *–*sunt calculate cu ajutorul unorfunctii de agregare distributive. Presupunem ca datele sunt impartite in *n* seturi. Calcularea functiei pe fiecare partitie determina o valoare agregata. Daca rezultatul obtinut prin aplicarea functiei asupra a n valori agregate este acelasi cu cel obtinut prin aplicarea functiei asupra tuturor datelor fara partitionare, functia poate fi calculata in maniera distributiva. De exemplu, functia count( ) poate fi calculata pentru cubul de date printr-o prima partitionare a cubului intr-un set de subcuburi, calculand count( ) pentru fiecare subcub si apoi insumand rezultatele obtinute pentru fiecare subcub. Din acest motiv functia count( ) este o functie agregata distributiva.

***Masuri algebrice*** *-* sunt calculate cu ajutorul unor functii algebrice cu M argumente (unde M este un intreg pozitiv), fiecare din ele obtinuta prin aplicarea unei functii agregate distributive. De exemplu, AVG( ) poate fi calculata prin sum()/count() unde ambele functii sum( ) si count( ) sunt functii agregate distributive. In mod similar se poate demonstra ca min( ), max( ) si abaterea standard sunt functii algebrice agregate. Masura este algebrica daca este obtinuta prin aplicarea unei functii algebrice agregate.

***Masuri holistice*** *-* sunt calculate cu ajutorul unor functii holistice. O functie agregata este holistica, daca aceasta nu este limitata constant pe spatiul de stocare cerut de deschiderea subagregarii. In acest caz nu exista o functie algebrica avand M argumente (unde M este o constanta) care caracterizeaza calculul. Exemple comune de functii holistice sunt: median( ), mode ( ), rank( ). O masura holistica este obtinuta prin aplicarea unei functii agregate de tip holistic. Cele mai multe aplicatii necesita calcularea eficienta a masurilor distributive si algebrice. Exista mai multe tehnici eficiente pentru aceasta, in contrast, poate fi mai dificil de calculat in mod eficient masuri holistice. Exista totusi anumite tehnici eficiente de aproximare a calculului masurilor holistice. De exemplu, in loc de a calcula exact median( ), exista tehnici care pot determina aproximativ valoarea mediana pentru un set foarte mare de date, cu rezultate satisfacatoare.

Din punctul de vedere al *modalitatii de insumare si agregare in functie de dimensiuni,*Ralph Kimball in lucrarea “The Data Warehouse Toolkit” [KIMB96] clasifica metricile in trei categorii: *indicatori aditivi*care se pot insuma dupa toate dimensiunile*, indicatori semiaditivi*care se pot insuma numai dupa unele dimensiuni si*indicatori neaditivi* care nu se pot insuma dupa nici o dimensiune dar care pot fi combinate cu alte variabile pentru a deveni aditive.

***Metadatele –***reprezinta poate cea mai importanta componenta a depozitului de date. Pentru a putea utiliza depozitul de date, utilizatorii trebuie sa cunoasca ce date se gasesc aici, iar metadatele nu sunt altceva decat date despre date, date care descriu continutul depozitului si furnizeaza trimiteri directe la date. Tot la nivelul metadatelor se definesc si diverse vederi*(views)* asociate unor categorii specifice de utilizatori.

Dar metadatele nu sunt utile doar utilizatorului final. Ele sunt intens folosite pentru administrarea depozitului de date, continand informatii despre provenienta datelor, algoritmii de agregare si insumare, statistici privind utilizarea si multe altele.

Cand se utilizeaza intr-un depozit de date, metadatele sunt date care definesc obiectele depozitului. Metadatele sunt create pentru numele de date si definitiile din depozit. Metadatele aditionale sunt create pentru a asocia intervale de timp la datele extrase si alte campuri care vor fi adaugate prin curatirea datelor sau prin procesele de integrare. Nivelul metadatelor trebuie sa contina conform [JAJE98]:

* O descriere a structurii datelor din depozit, incluzand schema depozitului, dimensiunile, ierarhiile, definitiile datelor derivate;
* Metadatele operationale, care includ date privind evolutia in timp (istoricul datelor si secventa de transformare aplicata asupra lor), situatia datelor (active, arhivate sau sterse) si informatii de monitorizare (statistici privind utilizarea depozitului de date, rapoarte de erori, imprastierea datelor etc.);
* Algoritmi utilizati pentru insumare, care includ masura si dimensiunea algoritmilor definiti, date despre granularitate, partitii, arii de subiecte, agregari, sumarizari, rapoarte si filtre predefinite;
* Transformarile datelor de la mediul operational la depozitul de date si care includ bazele de date sursa si continutul lor, partitionarea datelor, extragerea datelor, curatirea datelor, regulile de intretinere si securitate a datelor;
* Date relative la performantele sistemului care includ indicatori si profiluri care imbunatatesc accesul la date si performantele de cautare;
* Metadate economice (business metadata), care includ termeni economici si definitii, expresii si formule de calcul ale indicatorilor.

Metadatele se aplica pentru sursele de date, pentru programele si regulile de extragere si transformare, pentru structura datelor si pentru continutul propriu-zis al depozitului de date. Importanta metadatelor pentru depozitul de date reiese din faptul ca acestea stabilesc contextul depozitului de date, usureaza procesul de analiza, mentin si cresc calitatea datelor dar si din faptul ca sunt o forma de auditare a transformarii datelor.

Metadatele ajuta administratorii si utilizatorii depozitului sa localizeze si sa inteleaga secventele de date atat in sistemele sursa cat si in structura depozitului. Daca metadatele care descriu formatul datelor din depozite sunt disponibile, atunci se elimina orice ambiguitate legata de semnificatia datelor.

Metadatele mentin si cresc calitatea datelor, fapt ce se realizeaza prin definirea valorilor valide pentru fiecare camp din depozit. Inainte de a fi efectiv incarcate in depozit, datele pot fi revazute si erorile pot fi corectate, regulile de corectie a erorilor pot fi documentate tot prin metadate. Se pot deosebi mai multe tipuri de metadate:

***Metadate administrative*.** Acestea contin descrieri ale bazelor de date sursa si ale continutului, ale obiectelor depozitului de date si ale regulilor folosite pentru a transforma datele din sistemul sursa in depozit. Printre exemple de astfel de metadate mentionez: descrirea tuturor sursele de date folosite, trecerea campurilor sursa in campuri destinatie, schema depozitului de date, structura datelor din back-end, programe si instrumente back-end, reguli si formule de calcul, reguli de securitate si de acces.

***Metadate pentru utilizatorii finali*.** Aceste metadate au rolul de a ajuta pe utilizatori sa-si creeze propriile lor interogari si sa interpreteze rezultatele. Pentru aceasta, ei au nevoie sa cunoasca definitiile datelor din depozit, descrierea lor, precum si orice ierarhie care poate exista in diferite dimensiuni. Exemple de astfel de metadate sunt urmatoarele: continutul depozitului de date, rapoarte si interogari predefinite, definitiile ierarhiilor, calitatea datelor, istoricul incarcarii depozitului de date, reguli de eliminare.

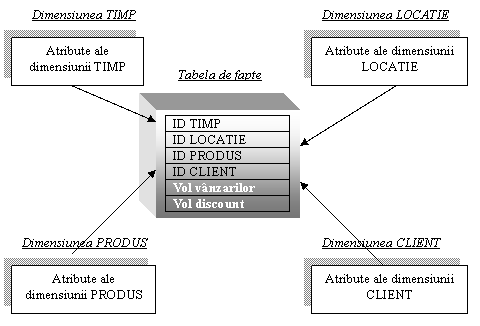
***Metadate pentru optimizare*.** Aceasta categorie de metadate are rolul de a creste performantele depozitului de date. Exemple de astfel de metadate sunt: definitiile agregarilor si colectii de statistici.

Un depozit de date contine date pentru diferite perioade de timp si de aceea este important sa avem in vedere efectul pe care il poate avea timpul asupra regulilor de trecere a campurilor sursa in campuri destinatie, asupra agregarilor etc. Utilizatorii trebuie sa aiba acces la metadatele corecte pentru perioada de timp pe care o studiaza.     Echipa IT are nevoie de aceste informatii pentru a putea intretine depozitul de date, iar ceea ce la prima vedere ar parea sa fie o eroare in transformarea datelor poate fi de fapt rezultatul schimbarii regulilor de transformare a datelor. De aceea este important ca metadatele sa fie corect gestionate din punct de vedere al versiunilor.

Desi in mod traditional metadatele reprezinta o componenta dezvoltata spre sfarsitul ciclului de dezvoltare, la ora actuala exista o tendinta puternica de a atribui metadatelor un rol mai important. Utilizatorii instrumentelor de extragere si transformare pot specifica modul de trecere din campurile sursa in campurile destinatie si pot introduce toate regulile care guverzeaza transformarea. Tabelul sursa-destinatie poate servi ca baza pentru generarea codului de program folosit apoi la extragerea si transformarea efectiva a datelor. Utilizatorii instrumentelor pentru calitatea datelor pot specifica valorile valide pentru diferite secvente de date atat in sistemele sursa, cat si in depozitul de date. Aceste instrumente pot folosi metadatele ca baza de pornire in identificarea si corectarea erorilor. Utilizatorii specifica metadatele referitoare la schema depozitului de date (fapte, dimensiuni etc), iar aplicatile pot folosi aceste specificatii ca intrare pentru a genera efectiv schema (tabele, indecsi, agregari etc.).

***Schema modelului*** este o colectie de obiecte, incluzand tabelele, viziunile, indecsi si sinonime. Exista mai multe tipuri de scheme utilizate in modelarea multidimensionala acestea diferind de modurile in care se pot aranja obiectele in cadrul schemei.

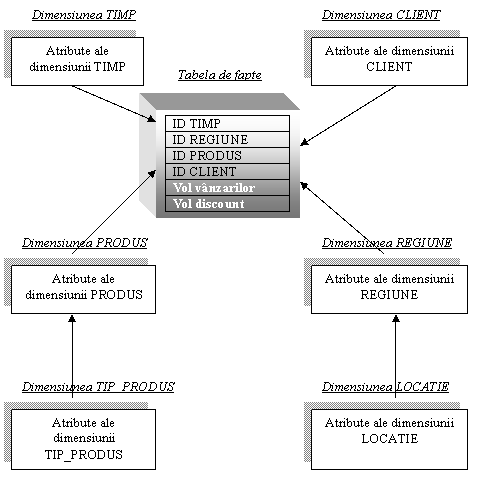
***Schema de tip “Stea“* -**Acesta este cel mai simplu si mai frecvent utilizat model (vezi figura ce urmează). Obiectele sale sunt dispuse in forma de stea, in centru aflandu-se una sau mai multe tabele de fapte de care sunt legate dimensiunile. O schema de “jonctiune stea“suporta doua tipuri de interogari: consultare si jonctiuni multiple. Operatia de consultare se realizeaza pe o singura tabela de fapte si nu necesita jonctiuni. O cerere de interogare tipica apare atunci cand un utilizator final solicita o lista derulanta. Interogarile de tip jonctiune multipla apar dupa o serie de consultari si implica restrictii plasate in cateva tabele dimensiune care sunt puse in legatura simultan, prin operatia de jonctiune, cu tabela de fapte. Scopul este de a aduce sute si mii de inregistrari de baza intr-un set de raspunsuri de dimensiune mica.



*Figura a:  Schema de jonctiune stea*

Dimensiunile in acest caz sunt denormalizate, ele avand date redundante care elimina necesitatea unor legaturi multiple intre tabele. Intr-o schema stea nu exista decat o singura legatura intre tabela de fapte si dimensiuni. Optimizarea performantei de raspuns la interogari este principalul avantaj al acestui model.

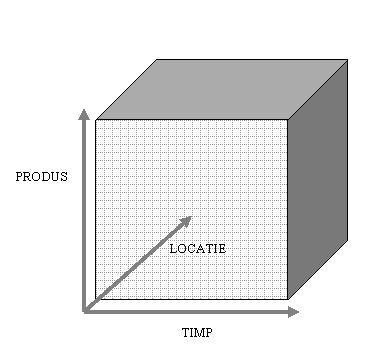
***Schema de tip “Fulg de Nea”* -**este o varianta a modelului stea in care o parte din tabelele dimensiune sunt normalizate, iar datele sunt distrinuite in tabele suplimentare (vezi figura b). Rezulta o schema reprezentata intr-un grafic similar unui fulg de zapada. Diferenta intre modelul stea si modelul fulg de nea este ca tabelele dimensiune din acesta pot fi pastrate in forma normalizata, ceea ce determina o redundanta redusa. Asemenea tabele sunt usor de intretinut si astfel se economiseste spatiu de stocare. Totusi aceasta economie de spatiu este neglijabila in comparatie cu volumul foarte mare de date din tabelul de fapte. Mai mult, structura fulg de nea poate reduce performanta extragerii de date deoarece sunt necesare mai multe jonctiuni intre tabele la o singura interogare.



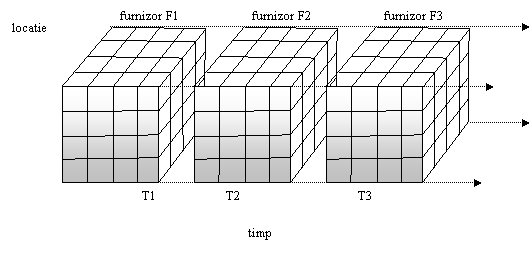
*Figura b:  Schema de jonctiune fulg de nea*

***Cuburi de date****-*Un mod mai simplu de vizualizare a datelor este reprezentarea intr-un spatiu cartezian definit pe toate dimensiunile depozitului de date (Vezi figurile c si d). Acesta poate fi numit cub de date, fiind un spatiu de date logic si nu unul fizic. Sectiunile bidimensionale sunt numite tablouri. Axele cubului sunt reprezentate de dimensiuni, la intersectia acestora fiind variabilele sau masurile.

In analiza multidimensionala cubul de date cu mai mult de trei dimensiuni poarta denumirea de cub n-dimensional sau hipercub (hypercub). Consiliul OLAP defineste cubul n-dimensional ca fiind ”*un grup de celule de date aranjate dupa dimensiunile datelor. O matrice tridimensionala poate fi vizualizata ca un cub cu fiecare dimensiune formand o fata a cubului*” [OLAP95]. Tot in aceeasi definisie se mentioneaza ca dimensiunile tipice ale datelor dintro intreprindere sunt timpul, masurile, produsele, regiunile geografice, canalele de distributie.



*Figura c:  Cub de date cu trei dimensiuni*

Text Box: produs

*Figura d:  Cub de date cu patru dimensiuni*

1. **OPERATII REALIZATE ASUPRA MODELULUI MULTIDIMENSIONAL**

Aplicatiile de analiza OLAP trebuie sa asigure o utilizatorilor o viziune multidimensionala asupra datelor, indiferent daca modalitatea de stocare este relationala sau multidimensionala. Pentru utilizarea viziunilor multidimensionale nu este necesara o stocare a datelor in aceasta forma. Bazele de date relationale si cele multidimensionale folosesc modele asemanatoare ceea ce permite o trasformare usoara a datelor. Prin aplicarea unor operatii specifice asupra modelului multidimensional utilizatorului i se ofera posibilitatea de a vedea si de a analiza din perspective multiple datele, de a naviga in cadrul ierarhiilor definite, de a extrage un subset de date, de a interschimba axele sau dimensiunile pentru a obtine o alta detaliere a datelor. Toate aceste operatii multidimensionale impementate in cadrul modelului multidimensional sunt prezentate in paragrafele urmatoare.

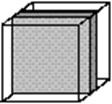
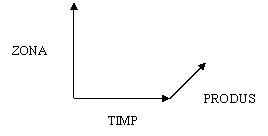
***Navigarea pe nivelele ierarhice* *(Drill Down si Roll Up)*** – reprezinta operatii de navigare in cadrul ierarhiilor dimensiunilor, prin agregare pe nivelele superioare sau detaliere pe nivelele inferioare. Orice baza de date multidimensionala trebuie sa permita navigarea pe diferite nivele ale ierarhiilor. Aceasta tehnica se numeste *roll up* sau *drill down*, in functie de directie, spre varful sau baza ierarhiei. Acestea sunt operatii de schimbare a vederii de-a lungul nivelelor unei ierarhii. Prin facilitatea de *drill down*, utilizatorii pot naviga pe nivele cu un grad de detaliu mai accentuat. Prin *roll up* se pot vizualiza datele la un nivel agregat. Cu toate ca instrumentele OLAP pot realiza dinamic toate operatiile necesare analizei, pentru a economisi timp si resurse se prefera uneori pre-calcularea unor valori globale. Aceasta operatie este numita consolidare (cand se refera la aspectul conceptual) sau insumare (din perspectiva procedurala), fie agregare (din perspectiva structurala). Aceste agregari se refera la o anumita masura si se realizeaza dupa dimensiunile corespunzatoare acesteia. Pentru atributele organizate ierarhic, consolidarea se face nivel cu nivel. Aceste  operatii implica de cele mai multe ori doar calcularea unor totaluri, dar exista si exceptii in care se utilizeaza formule sau procedee statistice. Nivelul la care se face insumarea in cazul in care sunt implicate ierarhii se numeste granularitate. Procesul de agregare creaza o redundanta in cadrul bazei de date, dar volumul acesteia nu este semnificativ deoarece scade exponential cu fiecare nivel de insumare. Castigul de performanta obtinut la accesarea datelor este deosebit de important in analiza.

***Rotatii*** – reprezinta operatiile cele mai uzuale in structurile de date  multidimensionale si ofera utilizatorului posibilitatea de a alege perspectiva asupra datelor pe care o va utiliza. De exemplu in cazul bidimensional exista doua posibilitati de vizualizare, iar in cazul tridimensional se pot utiliza 6 rotatii pentru a vizualiza datele din diferite perspective, iar pentru patru dimensiuni exista 24 de perspective posibile. Fiecare rotatie pune in evidenta o noua perspectiva, aducand in prim plan o structura bidimensionala, o fateta (*slice*). Din acest motiv rotatia se mai numeste si “*data slicing*”. Aceste operatii nu implica o reorganizare a datelor stocate, ci o schimbare a modalitatii de reprezentare, spre deosebire de cazul unor structuri relationale, pentru care o noua fateta poate fi obtinuta doar in urma unor interogari complexe.

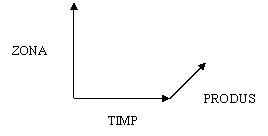
***Sectiuni*** - reprezinta viziuni sau imagini (views) specifice diverselor categorii de utilizatori, prin operatii de sectionare prin care se obtin 'felii' bidimensionale (*slices*). Astfel, un manager de produs poate avea la indemana datele legate de produsul pe care-l supervizeaza, pe toate zonele, pe toata perioada analizata. In schimb, un manager regional, va fi interesat de toate produsele, dar numai pe toate zonele pe care le coordoneaza. Tehnica aceasta consta in limitarea unor atribute la anumite valori si obtinerea unui cub de date redus (procedeu numit *data dicing*) (vezi figura a, ce urmează).

|  |
| --- |
|  |
|  | http://www.rasfoiesc.com/files/informatica/260_poze/image034.gif |

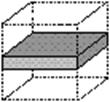
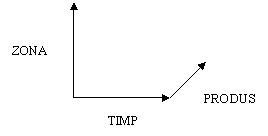
*Figura a: Cub de date tridimensional. Dimensiunile reprezinta timpul, produsele  si zonele de desfacere.*



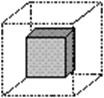
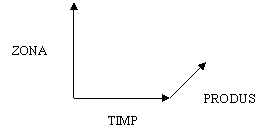
*Figura b: Viziunea managerului de produs: acesta poate obtine o viziune a datelor ce reflecta doar vanzarile anumitor produse in toata regiunea si in toata perioada de timp considerata.*



*Figura c: Viziunea managerului financiar: poate restrictiona analiza la un anumit trimestru pe toate produsele si toate zonele.*



*Figura   d: Viziunea managerului regional: poate vedea vanzarile intregii game de produse in regiunea de care raspunde, pe toata perioada de timp considerata.*



*Figura e: O viziune ad-hoc: diferite cerinte pot duce la selectarea unor anumite valori ale atributelor. Rezultatul consta in subseturi de date si din acest motiv aceste operatii se mai numesc si “data dicing”.*

1. **MODELE DE DATE MULTIDIMENSIONALE UTILIZATE IN SISTEMELE OLAP**

Modelele de date utilizate in sistemele OLAP au cunoscut o diversitate destul de mare atat din punctul de vedere al teoretizarii conceptelor cat mai ales din punctul de vedere al aplicarii diferitelor tipuri de modele in practica. Doua directii importante au clasificat totusi aceasta diversitate de modele si anume *dezvoltarea unor extensii ale modelului relational* si utilizarea acestora in cadrul sistemelor OLAP si a doua directie – *dezvoltarea modelelor bazate pe cuburi n-dimensionale*.

Printre *extensiile modelului relational* se pot mentiona:*modelul propus de Gray* la baza caruia sunt operatorii CUBE si ROLLUP din clauza GROUP BY din limbajul SQL care presupune agregarea datelor pe atributele clauzei group by; *modelul propus de Li si Wang* sau *modelul lui Gyssens si Lakshmanan* care sunt extensii ale algebei relationale [MUNT04]. Insa cel mai important model si cel mai reprezentativ este cel propus de *Ralph Kimball* in lucrarea [KIMB96] care defineste *schema tip stea* ca o reprezentare relationala a cubului n-dimensional. Schema tip stea prezentata anterior si in cadrul acestui capitol cuprinde in viziunea lui Kimball o tabela centrala si mai multe tabele dimensiune legate radial de tabela de fapte prin jonctiuni asemanator cu modelul ER. Din modelul de tip stea a derivat mai tarziu si *modelul tip fulg de nea* care extinde facilitatile oferite de modelul anterior. Ulterior au aparut notiuni ca *schema galaxie* care este o schema stea cu mai multe tabele de fapte sau *schema constelatie* (*fact constellation scheme*) in care exista tabele de fapte suplimentare ce stocheaza date agregate. O constelatie este o colectie de stele si consta dintr-o stea centrala inconjurata de alte stele. Steaua centrala contine datele la nivel atomic, iar celelalte stele contin date agregate [MUNT04].

Printre *modelele bazate pe cub* se poate aminti *modelul lui Agrawal, Gupta si Sarawagi* care are la baza un set minimal de operatori asemanatori cu cei din algebra relationala, insa organizarea datelor se bazeaza pe unul sau mai multe cuburi n-dimensionale. In viziunea lui Agrawal [MUNT04] cubul are urmatoarele componente: dimensiunile definite prin nume si domeniu de valori si elementele cubului care sunt definite printr-o functie ce asociaza multimea valorilor dimensiunilor la un n-tuplu reprezentat de celulele cubului.

Tot in categoria modelelor bazate pe cub se situeaza si *modelul propus de Cabibbo si Torlone* [MUNT04] in care dimensiunile sunt categorii lingvistice ce descriu diferite moduri de prezentare si de analiza a informatiilor, iar fiecare edimensiune este organizata pe ierarhii. Modelul are la baza o schema multidimensionala formata din setul de dimensiuni, tabelele de fapte (f-table) si descrierile nevelurilor ierarhice.

*Modelul propus de Blaschka* [MUNT04] introduce o extensie a tehnicii de modelare entitate – asociere a modelului relational. *Tehnica ME/R* pentru proiectarea schemei multidimensionale contine o entitate denumita *nivel al dimensiunii*(*dimension level*), o relatie tip 1:n denumita *fact relationship*si o relatie binara denumita *relatie de clasificare* a doua niveluri ierarhice.

**Din *punct de vedere al nivelului de realizare* modelele multidimensionale utilizate in cadrul sistemelor OLAP sunt impartite pe cele trei niveluri: *conceptual, logic si fizic***[MUNT04]:

1. ***modele conceptuale*** ofera concepte apropiate de modul in care utilizatorii percep datele si sunt independente de implementare. La acest nivel se pot considera ca modele conceptuale modelul lui Cabibbo si cel propus de Blaschka.
2. ***modele logice*** ofera concepte ce pot fi intelese de utilizatorii finali dar depind de tipul de SGBD utilizat. Dintre modelele multidimensionale la nivel logic se pot considera modelul lui Kimball, cel propus de Li si Wang si cel al lui Agrawal.
3. ***modele fizice*** ofera concepte legate de modul in care sunt stocate fizic datele (descrierea datelor pe suport fizic), depinzand de SGBD-ul utilizat.

Tipul de model multidimensional utilizat de diverse tehnologii si produse software ce implementeaza sistemele OLAP difera atat din punct de vedere al SGBD-ului utilizat cat si din punct de vedere al operatiilor realizate asupra datelor si a arhitecturii implementate (MOLAP, ROLAP, HOLAP).

1. **LOCUL TEHNOLOGIEI OLAP  IN ARHITECTURA DEPOZITULUI DE DATE**

In cartea “Building the Data Warehouse”, W.H. Inmon mentioneaza: “*Sunt patru niveluri in cadrul mediului arhitectural:* *operational, atomic sau al depozitului de date, departamental si individual*” [INMO96].

***Nivelul operational****-*            Sistemele operationale sunt reprezentate de sursele, datele care populeaza depozitul de date. Datele operationale sunt supuse tranzactiilor, volatile, stocate la nivel de tranzactie in forma normalizata sau proprie in sistem OLTP.

***Nivelul depozitului de date****-*Acest nivel contine date cu caracter istoric ale nivelului tranzactional, prelucrate si transformate intr-un format multidimensional mult mai potrivit pentru suportul de decizii. O singura tabela de fapte poate avea o inregistrare pentru fiecare tranzactie si fiecare inregistrare va contine valorile sau masurile si alte campuri descriptive ce vor reprezenta cat mai fidel intregul potential al dimensiunilor caracterizand afacerile (timp, zone, clienti, produs) tinzand catre un continut complet al ariei subiectelor (date despre vanzarea produselor, date despre cost, tipuri de venituri, tipuri de cheltuieli). Insa cu un volum foarte mare de date este imposibil sa se furnizeze un timp de raspuns rapid la cererile managerilor. De aceea este nevoie de nivelul departamental.

***Nivelul departamental, data mart sau OLAP****-*In aceeasi carte, W.H.Inmon scrie: “Nivelul departamental este uneori denumit ‘nivelul data mart’, ‘OLAP’, ‘baza de date multidimensionala’”. Tehnologia OLAP ar trebui folosita la acest nivel in arhitectura. Un data mart OLAP va fi limitat la submultimea marimilor statistice disponibile si dimensiunilor necesare pentru a studia problemele specifice afacerilor. Intr-un mediu inteligent, bine proiectat al afacerilor, 80% sau mai mult din totalul de cereri sunt transmise data mart-ului si server-ului OLAP. Cand datele ajung in depozit, ele trebuie sa fie pregatite pentru a fi redistribuite imediat in data mart. Structura dimensionala trebuie sa fie deja definita si reprezentata in depozit prin schema stea a bazei de date relationale. Ar trebui sa existe un depozit central care catalogheaza continutul si statutul depozitului. Serverul OLAP ar trebui sa poata citi direct din tabelele depozitului, atat datele cat si metadatele necesare pentru restructurarea si actualizarea data mart-ului cu submultimea ceruta de masuri, dimensiuni, inregistrari.

Mai mult, arhitectura trebuie sa fie cuprinzatoare si flexibila suficient pentru ca noile date mart-uri sa poata fi create rapid si cele existente sa fie redefinite rapid, simplu, prin selectarea noilor combinatii de masuri si dimensiuni din cele deja existente in depozitul metadatelor ca raspuns la cererea noua sau redefinita. Cand bazele de date ale sistemului OLAP sunt incomplete (acest lucru se intampla des), proiectantii incearca sa anticipeze toate cererile posibile pentru toate domeniile posibile (subiecte) si apoi incearca sa reuneasca continuturile intregului depozit intr-unul singur numindu-se OLAP ‘data waremart’. Depozitul de date este un proces continuu de dezvoltare iterativ, ceea ce trebuie sa conduca la posibilitatea de a adapta modelele data mart la necesitatile afacerilor. In orice caz, pentru o mai mare flexibilitate se recomanda ca arhitectura sa includa atat nivelul depozitului de date cat si data mart.

***Nivelul individual****-*La ultimul nivel al arhitecturii, datele sunt prezentate managerilor pentru interpretare. Instrumentele de vizualizare a cererilor, precum grafice, prezentari, rapoarte dinamice, browserele Web, toate apartin acestui nivel. Aplicatiile clientilor, care contin informatii despre bugete, prognoze, recomandari cu privire la alocarea resurselor si multe altele se afla in data mart la acest nivel al arhitecturii.

Din punctul de vedere al modalitatilor de realizare a sistemelor informatice executive, consider ca locul tehnologiei OLAP in cadrul depozitelor de date ale organizatiei este esential, acesta acoperind practic cele doua nivele superioare identificate de W.H. Inmon. Analiza datelor din depozite fara tehnologia OLAP este ar fi extrem de grea, implicand metode si modele statistice si matematice laborioase, functii de analiza dezvoltate de programatori, interfete speciale, dezvoltate separate de restul sistemului.

**Sursa:** Sisteme informatice executive, editura ASE, 2007, autori Ion Lungu, Adela Bara

**DOCUMENTE DEREFERINTA OLAP, DATAWAREHOUSE, DSS**

|  |  |  |
| --- | --- | --- |
|  | [ANDE97] | Anahory, S., Dennis, M. - *Data Warehousing in the Real World*, Addison Wesley Longman, Reading, Mass, 1997 |
|  | [BARA03/2] | Bara A. - *Rolul sistemelor OLAP si a depozitelor de date in managementul strategic*, Sesiunea tinerilor cercetatori “Evolutii economice si financiare in contextul integrarii si globalizarii”, Centrul de Cercetari Financiare si Monetare “Victor Slavescu”, Bucuresti, 2003 |
|  | [CRAB99] | Crabone, L., P. - *Data Warehouses: Many of the common failures*, White paper, mai 1999 |
|  | [DEVL97] | Devlin, B. -  *Data Warehouse – from Architecture to Implementation*, Addison Wesley Longman, Reading, Mass, 1997 |
|  | [DOBR99] | Dobre I. - *Suport de curs postuniversitar, Metode si tehnici de analiza a*  *sistemelor social economice*, Academia de Studii Economice, Facultatea  de Cibernetica, Statistica si Informatica Economica, 1999 |
|  | [EDIS06] | Edison Group Inc - *Comparative management Cost study of Oracle Database 10g release 2 and Microsoft SQL Server 2005*, 6 martie 2006 |
|  | [FUBA03] | Fusaru D., Bara A. - *Tehnici si arhitecturi pentru micsorarea timpului de raspuns in sistemele cu depozite de date* - Comunicare la Sesiunea de Comunicari Stiintifice a cadrelor didactice “Economia Romaniei – criterii de functionalitate si competitivitate”, Universitatea “Spiru Haret”, Bucuresti, mai 2003, publicatǎ in Analele Universitǎtii Spiru Haret, Seria Economie, Anul 3, nr. 3, pag. 378-385 Editura si Tipografia Fundatiei Romania de Maine, 2003, ISSN 1582-8336. |
|  | [FUBA04] | Fusaru D., Bara A. - *Sisteme informatice de asistare a deciziei in managementul modern al organizatiilor economice* - Comunicare la Sesiunea de Comunicari Stiintifice “Realizari si perspective in procesul fauririi economiei de piata, functionale, competitive si durabile in Romania”*,* Universitatea Spiru Haret, Bucuresti, mai 2004, publicatǎ in Analele Universitǎtii Spiru Haret, Seria Economie, Anul 4,nr. 4, pag. 393-399 Editura si Tipografia Fundatiei Romania de Maine, 2004, ISSN 1582-8336. |
|  | [HACH98] | J. Han, S. Chee, J. Y. Chiang. - *Issues for on-line analytical mining of data warehouses*. In Proc. 1998 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'98), Seattle, Washington, iunie 1998. |
|  | [HUHA99] | Humphries, M., Hawkins, M., Dz, M., - *Data Warehousing. Architecture and Implementation*, Prentice Hall PTR,Upper Saddle River, New Jersez, 1999 |
|  | [INMO96] | Inmon, W.H. -  *Building the Data Warehouse*, John Wiley & Sons, New York,  1996 |
|  | [INMO99] | Inmon, B. - *Data mart does not equal data warehouse*, DM Direct Newsletter, November, 1999 |
|  | [JAJE98] | Jarke, M.,  Jeusfeld, M.A., Quix, C., Vassiliadis, P*.-  Architecture and quality in data warehouses,*Proceedings CaiSE 98, Pisa, Italy, 1998 |
|  | [KIMB96] | Kimball, R. -  *The Data Warehouse Toolkit*, John Wiley & Sons, New York, 1996 |
|  | [KIRE98] | Kimball, R., Reeves, L., Ross, M., Thornthwaite, W. - *The data Warehouse Lifecycle Toolkit*, JohnWiley&Sons, Inc., New York, 1998. |
|  | [LUNG05] | Lungu, I - *Metode de dezvoltare a sistemelor informatice*, Editura Universitas, Petrosani, 2005 |
|  | [LUSA04] | Lungu, I, Sabau Gh, Velicanu M, Muntean M, Ionescu S, Posdarie E, Sandu D - *Sisteme informatice. Analiza, proiectare si implementare*, Ed. Economica, 2004 |
|  | [MUNT04] | Muntean M. - *Initiere in tehnologia OLAP: teorie si practica*, Editura ASE, Bucuresti, 2004 |
|  | [OLAP95] | The OLAP Council Definitions, ianuarie 1995   www.olapcouncil.org |
|  | [ORA10G] | ORACLE Corporation – documentatie produse Business Intelligence 10g - User’s Guide, Concepts, Internet seminars.  www.oracle.com |
|  | [ORLI90] | Orlikowski w. J. - *The Duality of Technology. Rethinking the concept of technology in organization,*Sloan School of Management Working paper, No. 3141, MIT 1990. |
|  | [ORRO91] | Orlikowski W. J. and Robey D. - *Information Technology and the Structuring of organisations.*Information Systems Research. Vol. 2, 1991, pp. 143-169. |
|  | [OWP06] | Oracle Database 10g Product Family – *Oracle White Paper*, Oracle Corporation, august 2006 |
|  | [PODE89] | Poole, M. S. and Desanctis, G. - *Understanding the use of Group Decision Support Systems: The Theory of Adaptive Structuration,*in Stein field C. And Fulk J., (eds.) Perspectives on Organisations and New Information Technology*.*Sage, 1989. |
|  | [POWE00] | Power  D.J. - *Decision Support Systems: Concepts and Resources*, Cedar  Falls, IA: DSSresources.com, http://dssresources.com/dssbook/ |
|  | [RAPA72] | Rapaport A. - *The use mathematical isomorphism in general systems theory, Trends in general systems theory*, New York, 1972 |
|  | [RYAN99] | Ryan, J. - *Building and deploying an enterprise data warehouse*, White Paper, 1999. |
|  | [THIE91] | Thierauf Robert J.  - *Executive Information Systems: A Guide for Senior Management and Mis Professionals,*Hardcover / Quorum Books, 1991 |
|  | [THOM02] | Thomsen E. *- OLAP Solutions: Building Multidimensional Information Systems*, John Wiley&Sons, New York, 2002, second edition |
|  | [TRPA01] | J. Trujillo, M. Palomar, J. Gómez, Il-Yeol Song - *Designing Data Warehouses with OO Conceptual Models*. IEEE Computer, special issue on Data Warehouses, 2001. |
|  | [TURB98] | Turban, E. - *Decision Support Systems and Intelligent Systems*, 5th ed.,  Englewood Cliffs, New Jersey, Prentice Hall, 1998 |
|  | [VILA97] | Vilan, A. - *Data warehouses, data marts si data mining*, Revista Computerworld Romania, nr. 18 (88), 21 Octombrie 1997 |
|  | [ZADE74] | Zadeh, L, A. - *Notiunile de sistem, subsistem si stare in teoria sistemelor*, Ed Tehnica, 1974 |