UNIVERSITATEA POLITEHNICA DIN BUCURESTI FACULTATEA DE AUTOMATICA SI CALCULATOARE

|  |  |  |  |
| --- | --- | --- | --- |
| Sistem de procesare a datelor folosind Business Intelligence  Inteligenta Business Intelligence  Absolvent: Cristina Aciubotaritei | | |  |
|  |  |  | 2010 |
|  | | |  |

C O O R D O N A T O R S T I I N T I F I C : S . L . D R . I N G . M A D A L I N V L A D

1

**UNIVERSITATEA POLITEHNICA DIN BUCURESTI FACULTATEA DE AUTOMATICA SI CALCULATOARE**

Sistem de procesare a datelor folosind

Business Intelligence

Coordonator Stiintific:

S.L. Dr. Ing. Madalin Vlad

Absolvent: Aciubotaritei Cristina

Cuprins

1. [Introducere in Business Intelligence 4](#_bookmark0)
   1. [Introducere 4](#_bookmark1)
2. [Modele de date multidimensional 7](#_bookmark2)
   1. [Concepte de bază 7](#_bookmark3)
      1. [Conceptul de cub n-dimensional 7](#_bookmark4)
      2. [Conceptul de dimensiune 10](#_bookmark5)
      3. [Conceptul de ierarhie 10](#_bookmark6)
      4. [Conceptul de măsură 13](#_bookmark7)
      5. [Conceptul de multicub 14](#_bookmark8)
   2. [Arhitectura sistemelor OLAP 15](#_bookmark9)
   3. [Sisteme informatice pentru inteligenta afacerilor (Business Intelligence) 21](#_bookmark10)
3. [Arhitectura ORACLE Business Intelligence 25](#_bookmark11)
   1. [Serverul de Business Intelligence 25](#_bookmark12)
   2. [Tipuri de Layere din Depozitul de Date 26](#_bookmark13)
   3. [Planificarea unui Model Business 27](#_bookmark14)
   4. [Analiza cerintelor unui model business 27](#_bookmark15)
   5. [Identificarea Structurii Modelului Business 28](#_bookmark16)
   6. [Identificarea continutului bazei de date pentru modelul business 30](#_bookmark17)
   7. [Unelte BI specializate 34](#_bookmark18)
4. [Depozite de date (Data warehouse) 39](#_bookmark19)
   1. [Arhitectura depozitului de date 39](#_bookmark20)
   2. [Metadate 42](#_bookmark21)
   3. [Cuburile şi analiza multidimensională 42](#_bookmark22)
   4. [Integrarea tehnologiei relationale cu tehnologia OLAP 48](#_bookmark23)
5. [Structura bazei de date 53](#_bookmark24)
6. [Dezvoltarea aplicatiei de Business Intelligence (Sales Trend) 56](#_bookmark25)
   1. [Crearea layerului fizic 57](#_bookmark26)
   2. [Crearea Layerului de Modelare si Mapare Business 59](#_bookmark27)
   3. [Crearea Layerului de Prezentare 61](#_bookmark28)
   4. [Dezvoltarea arhitecturii de business 61](#_bookmark29)
   5. [Diagrame UML 74](#_bookmark30)
   6. [Prezentarea aplicatiei Sales Trends 81](#_bookmark31)
7. [Avantajele oferite de Business Intelligence. Concluzii 84](#_bookmark32)
8. [Bibliografie 86](#_bookmark33)

## Abstract

*In order to achieve a competitive advantage, a company has to rapidly identify market opportunities and take advantage of these opportunities in a rapid and efficient manner. More and more companies realize that having large amount of data does not imply a better understanding of the market or of the improvement of the operational performances. Business intelligence (BI) is a common word in today’s IT world. Yet, is a concept that not everybody understands. And many questions regarding the nature of this concept are rising. So let’s take a closer look inside business intelligence…*

# Introducere in Business Intelligence

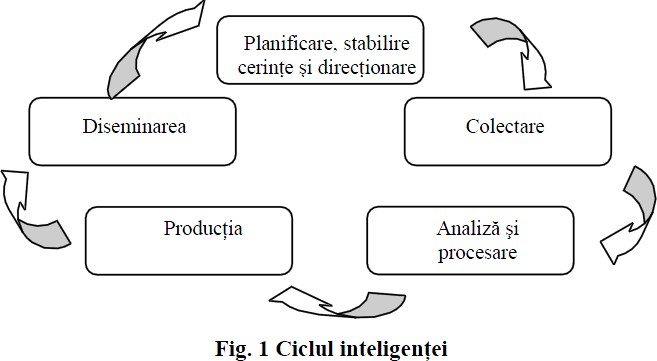
# Introducere

Termenul de business intelligence a fost introdus de către Gartner Group la mijlocul anilor ‘90s. Ca şi concept însă, business intelligence a existat cu mult timp înainte, încă din anii ’70 \*Zaman, 2005+, folosit în sistemele de raportare cu ajutorul mainframe-urilor. La acea vreme, sistemele de raportare erau statice, bidimensionale, fără a avea capacităţi analitice. Cererea de sisteme multidimensionale dinamice, care să sprijine procesele decizionale inteligente şi cu abilităţi predictive, a determinat dezvoltarea sistemelor de tip business intelligence. Aceste sisteme devin din ce în ce mai complexe, fiind capabile de analiză multidimensională a datelor, dispunând de capacităţi de analiză statistică şi predictivă pentru a servi mult mai bine sistemelor de asistare a deciziilor. Nevoia de sisteme de tip business intelligence poate fi cu mare uşurinţă explicată: pentru a supravieţui pe piaţă în actualele condiţii concurenţiale, o companie trebuie să încerce să dezvolte o strategie de succes; pentru a dezvolta o strategie de succes, e nevoie de capacitatea de a anticipa condiţiile viitoare; înţelegerea trecutului este modul cel mai bun de a fi în stare a prezice viitorul. Business intelligence face acest lucru.

# Natura inteligenţei

În timp ce există o largă plajă de definiţii date inteligenţei, poate că cea mai reprezentativă este cea oferită de U.S. Central Intelligence Agency (CIA) \*Waltz, 2003+: „*redus la cei mai simpli termeni, inteligenţa este cunoştinţa şi modul în care (pre-) simţim lumea din jurul nostru – preludiul deciziilor şi acţiunilor politicienilor\*...+*”. Aceste componente clasice ale inteligenţei furnizează înţelegere şi determină leaderii în a lua deciziile care furnizează securitate pentru afaceri sau pentru state.

„*Inteligenţa*” presupune cunoaşterea informaţiilor despre competiţie, informaţii precumprofitabilitatea sau venitul acestora [Raisinghani, 2004]. Beneficiul suprem al inteligenţeieste reprezentat de cunoaşterea clientului şi a potenţialului client. Această cunoştinţă ajută la îmbunătăţirea serviciilor acordate clienţilor şi la o mai bună orientare a afacerii pe nevoileacestor clienţi. Procesul obţinerii informaţiei inteligente începe cu nevoia de cunoaştere a decidentului (consumatorul informaţiei) şi se termină cu livrarea respectivei cunoştinţe. Nevoia poate fi o cerinţă stabilă, o solicitare specială sau o necesitate urgentă în situaţia unei crize.



1. *Planificare, stabilire cerinţe şi direcţionare*: definirea de către decident, la un nivel înalt de abstractizare, a cunoştinţelor necesare pentru a lua decizii. Cerinţele sunt traduse întermeni de informaţii solicitate, apoi în date care trebuie să fie colectate.
2. *Colectare*: sursele tehnice şi umane sunt adresate pentru a se colecta datele brute cerute. Surse pot fi disponibile în mod deschis sau închis, fiind accesate prin diverse metode.Aceste surse şi metode sunt cele mai fragile şi protejate elemente ale procesului. Surse deinteligenţă pot fi: human intelligence (HUMINT), imagery intelligence (IMINT), signals intelligence (SIGINT), electromagnetic signals monitoring (ELINT), open source intelligence (OSINT), şi multe altele.
3. *Analiză şi procesare*: datele colectate sunt procesate (ex. traduceri din alte limbi, decriptări), indexate şi organizate. Progresul în atingerea cerinţelor planului de colectare este monitorizat, iar modul de abordare poate fi rafinat pe baza datelor primite.
4. *Producţia*: baza de informaţii este procesată folosindu-se tehnici de estimare sau inferenţiale care combină datele surselor în încercarea de a da răspuns la întrebarea solicitatorului. Datele sunt analizate (descompuse pe componente şi studiate) şi soluţiile sunt sintetizate (construite plecând de la evidenţele acumulate). Topicele subiectelor de studiu sunt modelate şi se pot face noi cerinţe pentru colectări şi procesări adiţionale.
5. *Diseminarea*. În cele din urmă, informaţia inteligentă este diseminată către consumatori în formate diverse, plecând de la imagini dinamice ale sistemelor militare de război şi până la rapoarte formale către politicieni. Se pot distinge trei categorii de rapoarte de inteligenţă tactică şi strategică formale: *current intelligence reports* sunt rapoarte tip ştiri, care descriu evenimente recente sau indicatori şi avertismente; *basic intelligence reports* furnizează descrieri complete ale unei situaţii specifice (ex. ordinea de luptă sau situaţii politice) şi *intelligence estimates*, rapoarte care încearcă să prevadă posibile situaţii viitoare ca rezultat a stării şi constrângerilor curente.

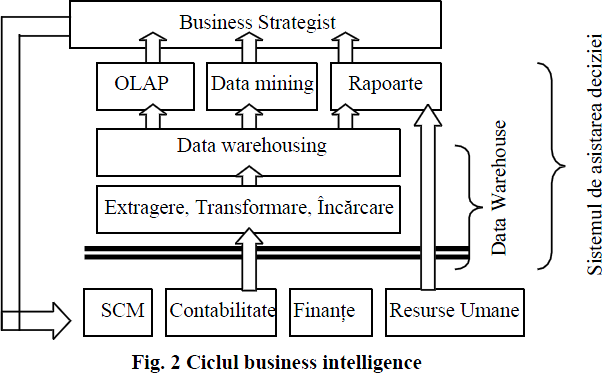
Produsele inteligenţei sunt diseminate către utilizator, furnizând răspunsuri la interogări şi estimări ale acurateţei produsului livrat. Facem o observaţie: chiar dacă procesul este prezentat sub forma unui ciclu, în realitate procesul operează ca acţiuni continue, cu multe feedback-uri (reacţii inverse) şi feedforward (reacţii în avans) care solicită colaborare între consumatori, colectori şi analişti.

# Concept

Business intelligence este un concept vag şi poate reprezenta folosirea de software de mare clasă pentru aplicaţiile de afacere. Într-o altă opinie, business intelligence reprezintă colecţia de tehnologii dintre cele mai noi care ajută sistemele de a deveni mult mai „inteligente”. Conform IBM

business intelligence înseamnă folosirea valorilor de tip date pentru a lua decizii mai bune. Este vorba despre acces, analiză şi descoperirea de noi oportunităţi. Conform Asociaţiei Române de Inteligenţă Economică, business intelligence este ansamblul acţiunilor de cercetare, colectare, tratare şi difuzare a informaţiei utile agenţilor economici, în scopul de a obţine avantaje concurenţiale, prin exploatarea ei în manieră defensivă sau/şi ofensivă.

Business intelligence este un proces iterativ: se porneşte de la mediul operaţional; datele sunt extrase din acest mediu şi depozitate în depozite de date (acest depozit de date se prezintă sub forma unui container central de date, separat de datele operaţionale); decidentul foloseşte sistemele de asistare a deciziilor pentru a extrage datele din depozitul de date; deţinând aceste informaţii, un decident poate să creeze planuri de acţiune; această schimbare la nivelul informaţiilor operaţionale duce la o nouă iteraţie a ciclului business intelligence.

Acest ciclu este prezentat în fig. 2 .

.

# Viitorul sistemelor de business intelligence

Un alt aspect deosebit de important referitor la business intelligence este cel referitor la performanţele acestor sisteme într-un context al creşterii utilizării lor. Se pare că viitorul va pune aceste sisteme în faţa imposibilităţii de a oferi avantajele promise. De ce? Din ce în ce mai multe companii încep să folosească sistemele de business intelligence. Iar aceste sisteme devin din ce în ce mai performante. În astfel de condiţii, o companie nu va putea niciodată să deţină un avantaj competiţional executând aceleaşi activităţi pe care şi alte companii le execută. Iar în al doilea rând, business intelligence este exclusiv concentrat pe oferirea de înţelegere asupra datelor; business intelligence nu oferă instrumentele necesare pentru implementările de schimbări operaţionale.

Legendarul investitor, Warren Buffet afirma: *„nu se câştigă prin prezicerea ploii. Se câştigă prin construirea unei ambarcaţiuni.”* \*Hyperion+ Şi rândurile următoare vin cu explicaţia: *„în timp ce business intelligence poate furniza înţelegerea condiţiilor atmosferice, BPM (n.a. business performance management) este cel care în cele din urmă va împuternici companiile să deţină un avantaj prin construirea unei ambarcaţiuni construirea mult mai rapidă a acestora, cu o mai mare eficienţă din punct de vedere al costurilor şi proiectate mult mai adecvat pentru a face faţă furtunii.”* Conform unui studiu recent [sap.info], soluţia pentru aceste limitări ale business intelligence, stă în modelul de afacere „Corporate Performance Management (CPM)”, un model care combină business intelligence cu business performance management. Acest nou model de afacere permite companiilor să-şi alinieze scopurile şi procesele de afacere cu activităţile zilnice de derulare a afacerii. Doar câteva din beneficiile acestui nou model de afacere \*technology+: un răspuns mult mai rapid la condiţiile şi

oportunităţile schimbătoare de pe piaţă, o puternică orientare către client, eficienţă operaţională sporită, o aliniere mai bună a bugetului, strategiilor de afacere şi planificare, profit sporit de pe urma investiţiilor în tehnologii informaţionale.

În afară de corporate performance management, o altă tendinţă în business intelligence este *analitica previzională* [Zaman, 2005] – folosită la determinarea rezultatelor viitoare posibile ale unui eveniment sau a probabilităţii de apariţie a unei anumite situaţii. Analitica previzională este folosită la analiza automatizată a unei cantităţi imense de date cu diferite variabile; această tehnică include arbori decizionali, analiza coşului de piaţă, reţele neuronale, algoritmii genetici, text mining etc.

# Modele de date multidimensional

# Concepte de bază

Pentru a descrie în detaliu modelele de date multidimensionale (extensii ale modelului relaţional sau orientate pe cub) şi a putea fi înţelese, s-a considerat necesar a se prezenta o serie de concepte utilizate şi anume: *hypercubul (cub ndimensional),multicubul, dimensiunile*, *ierarhiile*, *măsurile.* Aceste concepte apar în majoritatea modelelor OLAP, deşi modul lor de definire diferă uneori foarte mult. De asemenea, consiliul OLAP a propus un glosar de termeni care se doreşte standardizat.

# Conceptul de cub n-dimensional

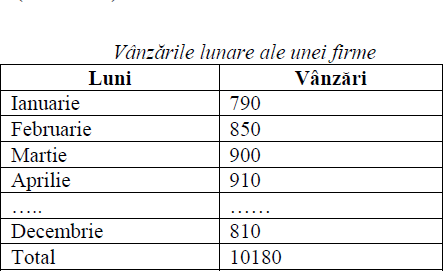
Conceptul *de hypercub* sau cub cu mai mult de trei dimensiuni (*cub ndimensional*) sau *structură multidimensională* este fundamental pentru înţelegerea sistemelor OLAP şi a modelului multidimensional. Instrumentele OLAP folosesc conceptul de *hypercub* în acelaşi mod în care foile de calcul tabelar folosesc conceptul de *foaie de lucru* (worksheet) şi bazele de date relaţionale conceptul de *tabelă.* Toate vizualizările, rapoartele şi analizele sunt făcute în termeni de *hypercuburi (cuburi n- dimensionale).*

Consiliul OLAP defineşte *hypercubul* ca „*un grup de celule de date aranjate după dimensiunile datelor. De exemplu, o foaie de calcul tabelar exemplifică o matrice bidimensională cu celulele de date aranjate în rânduri şi coloane, fiecare fiind o dimensiune. O matrice tridimensională poate fi vizualizată ca un cub cu fiecare dimensiune formând o faţă a cubului. Dimensiunile tipice ale datelor dintro întreprindere sunt timpul, măsurile, produsele, regiunile geografice, canalele de distribuţie etc.”* [OLAP97].

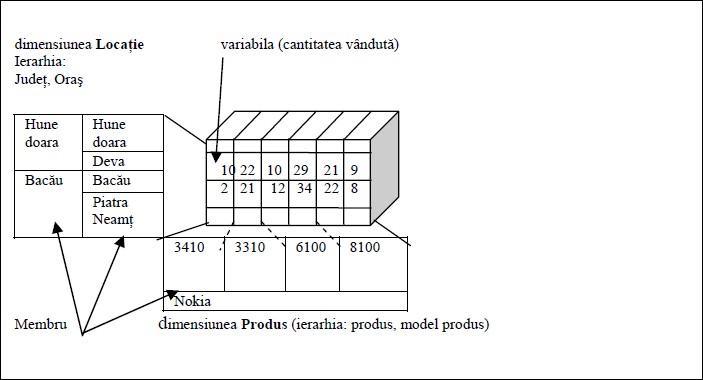
Pentru a înţelege conceptul de *hypercub* (*cub n-dimensional*) trebuie să ţinem cont de următoarele aspecte:

***Afişarea pe ecranul calculatorului nu este identică cu metaforele vizuale.*** Se consideră un exemplu de date bidimensionale şi anume informaţiile despre vânzările lunare (cantitatea vândută) de telefoane mobile Nokia ale unei firme.

Exemplul poate fi realizat foarte uşor cu ajutorul unei foi de calcul tabelar şi afişat pe orice ecran (tabelul 2.1).



Totalul vânzărilor este afişat pe ultimul rând. Este o singură coloană de date: coloana *Vânzări.* Setul de date are două dimensiuni: o dimensiune *Luna* aranjată pe rânduri şi o dimensiune *Vânzări* aranjată pe coloane. Lunile reprezintă modul cum sunt organizate datele. Lunile sunt un tip de cheie sau identificator. În concluzie, modelul prezentat are două dimensiuni: o dimensiune de identificare şi una pentru variabile. În figura 2.1 variabila “cantitatea vândută” este determinată în funcţie de trei dimensiuni: *Locaţie, Produs* şi *Timp***.** Dimensiunea *Locaţie* include o ierarhie cu două niveluri “judeţ” şi “oraş”, iar dimensiunea *Produs* ierarhia cu nivelurile „produs” şi “model produs”. Deşi nu se reprezintă în figură, dimensiunea *Timp* se referă la anul 2003. Fiecare subcub conţine cantitatea vândută pentru o anumită combinaţie de valori ale dimensiunilor. De exemplu, într-o anumită perioadă de timp, în oraşul Deva, din judeţul Hunedoara, s-au vândut 10 telefoane celulare, model Nokia 3410.

Ecranul calculatorului, la fel ca o hârtie, are numai două dimensiuni fizice. Putem crea o reprezentare bidimensională pe ecran a unui cub tridimensional. Afişarea datelor pe ecranul bidimensional al calculatorului este diferită de metaforele folosite pentru a vizualiza datele. Cubul vizualizat în figura 2.1 este o metaforă vizuală. Acest aspect îi face pe dezvoltatorii de software să caute un mod optim de reprezentare a structurilor numerice cu mai mult de două dimensiuni pe un ecran bidimensional, pentru vizualizare şi manipulare. De exemplu, în cazul foilor de calcul tabelar tridimensionale, setul de date tridimensional din figura 2.1 este afişat pe ecran, pe rânduri, coloane şi pagini. Este uşor de a vizualiza relaţia între datele prezentate pe ecran şi întregul set de date stocat în calculator. Tot ce trebuie să facem este să ne imaginam un cub de date tridimensional şi un ecran ce afişează o felie din acel cub.

**Figura 2.1** *Reprezentarea sub forma unui cub a modelului multidimensional*

***Dimensiunile logice nu sunt identice cu dimensiunile fizice.*** Cuburile studiate la geometrie sunt implicit fizice, deoarece se bazează pe noţiuni fizice ca lungime, lăţime şi înălţime. Cele trei axe

perpendiculare (x, y, z) se transformă perfect în dimensiuni fizice ca lungime, lăţime şi înălţime. Cubul fizic este o reprezentare intuitivă a unui eveniment, deoarece toate dimensiunile coexistă pentru orice punct din cub şi sunt independente între ele. Într-un spaţiu tridimensional, orice punct (orice vânzare) este identificat de coordonatele sale x, y şi z (sau de produs, timp şi locaţie). Totuşi fizic sunt numai trei dimensiuni independente. Chiar dacă nu este greşit de a folosi un cub cu unghiuri drepte pentru reprezentarea multidimensională a unui eveniment, definiţia bazată pe unghi drept a unei dimensiuni nu este necesară pentru reprezentarea evenimentului. O reprezentare corectă cere dimensiuni independente [THOM96].

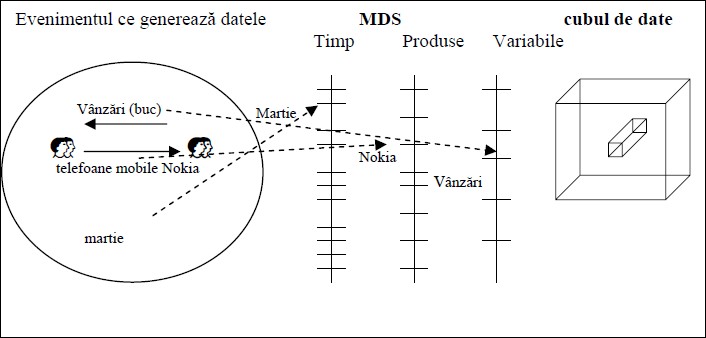
Erik Thomsen \*THOM96+ utilizează pentru reprezentarea evenimentelor un nou concept: “*structură de domeniu multidimensională*” (multidimensional domain structure-MDS) (figura 2.2). Fiecare dimensiune este reprezentată printr-un segment vertical. Orice membru dintr-o dimensiune este reprezentat de un interval din segment. Pentru exemplu tridimensional anterior, se identifică patru segmente: unul pentru timp, unul pentru produse, unul pentru locaţie şi unul pentru variabile. Fiecărui element din eveniment şi din cub îi corespunde un interval din fiecare din cele patru segmente. De exemplu, în figura 2.2, MDS prezintă vânzările de telefoane mobile Nokia din luna martie. Un MDS este mai descriptiv decât un cub fizic, totuşi nu arată punctele de date curente, ci combinaţii posibile ale membrilor dimensiunilor. Utilizând un MDS este uşor de a adăuga alte dimensiuni la model.

Un MDS nu este o reprezentare fotografică a evenimentului ce a generat datele, dar nu este nici cub fizic. Un MDS:

* arată numărul de puncte de date extrase din eveniment şi organizarea lor logică;
* prezintă toate dimensiunile care se pot vizualiza;
* arată mai multe informaţii structurale decât un cub fizic;
* şi poate fi realizat pentru orice număr de dimensiuni.

-

***Dimensiunile logice pot fi combinate.*** Cum se pot reprezenta patru sau mai multe dimensiuni logice în trei dimensiuni fizice (rând, coloană, pagină)? Răspunsul este de a combina multiple dimensiuni logice în aceeaşi dimensiune fizică. Maparea a două dimensiuni logice într-o singură dimensiune fizică înseamnă crearea unei versiuni unidimensionale. Metoda tipică este de a include o dimensiune în alta. Două lucruri se modifică ca rezultat al combinării dimensiunilor : forma datelor prezentate şi vecinii.



**Figura 2.2** *Exemplu de utilizare a MDS-ului pentru reprezentarea evenimentelor*

Într-o matrice bidimensională, fiecare punct are patru vecini. Când dimensiunile sunt combinate fiecare punct într-o listă unidimensională are numai doi vecini. Un lucru important nu se modifică în timpul procesului de combinare a dimensiunilor logice: nu are importanţă cum se

combină dimensiunile, rezultatele sunt aceleaşi. Abilitatea de a modifica uşor prezentările aceloraşi date, prin reconfigurarea a cum sunt afişate dimensiunile, este una din cele mai mari avantaje ale sistemelor OLAP. Se separă structura datelor reprezentată în MDS, de afişarea datelor. Orice combinaţie de dimensiuni logice poate fi mapată la orice combinaţie de rânduri, coloane şi pagini ale unui ecran.

# Conceptul de dimensiune

Conceptul de *dimensiune* este definit de Consiliul OLAP ca “*un atribut structural al unui cub ce constă dintr-o listă de membrii, pe care utilizatorii îi percepe ca fiind de acelaşi tip. De exemplu, toate lunile, trimestrele, anii formează dimensiunea Timp….. O dimensiune acţionează ca un index pentru identificarea valorilor dintr-o matrice multidimensională Dimensiunile oferă un mod foarte*

*concis, intuitiv de organizare şi selectare a datelor pentru explorare şi analiză.*” [OLAP97]. Dimensiunile sunt atribute de identificare a evenimentelor măsurabile sau a lucrurilor pe care le analizăm. Spre deosebire de dimensiunile fizice care sunt bazate pe unghiuri şi sunt limitate la trei, dimensiunile logice nu au astfel de limite.

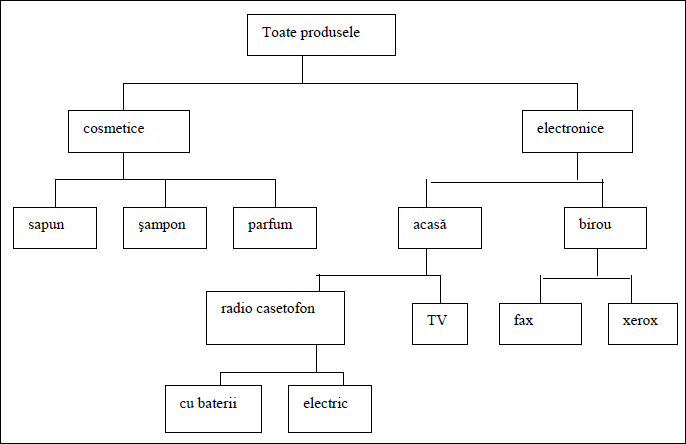
Frecvent numărul de dimensiuni într-un set de date depăşeşte cele trei dimensiuni fizice (rând, coloană, pagină) ale ecranului de afişare. Abilitatea instrumentului OLAP de a modela multiple dimensiuni de informaţii îl face mult mai potrivit pentru a lucra cu seturi complexe de date, decât bazele de date relaţionale şi foile de calcul tabelar. Dimensiunile au următoarele caracteristici:

* furnizează informaţii descriptive despre fiecare indicator (variabilă);
* conţin în general date statice şi sunt esenţiale pentru analiză. Un model multidimensional ce oferă un număr mare de atribute dimensionale permite analize cât mai complexe şi mai variate;
* într-un *cub n-dimensional* o dimensiune este reprezentată printr-o axă;
* într-o schemă stea sunt tabelele care se dispun radial în jurul tabelei de fapte şi se mai numesc tabele de dimensiuni.

De exemplu, într-o bază de date pentru analiza vânzărilor se identifică următoarele dimensiuni: *Timp, Regiune/locaţie, Client, Agent de vânzare, Produs*. O dimensiune conţine mai mulţi *membri*. Un membru este “*un nume distinct sau un identificator folosit pentru a determina poziţia unui element de dată* (în schema stea apare sub denumirea de atribut dimensional)”. De exemplu, toate lunile, trimestrele şi anii formează dimensiunea *Timp* şi toate oraşele, regiunile şi ţările dimensiunea *Locaţie*. Un membru poate aparţine la una sau mai multe ierarhii sau poate să nu fie inclus într-o ierarhie (independent). De exemplu, în dimensiunea *Produs* membru “culoare” nu este inclus în nici o ierarhie.

# Conceptul de ierarhie

O ierarhie este un atribut al unei dimensiuni. Cele mai multe dimensiuni au o structură multi- nivel sau ierarhică. Timpul este o dimensiune ierarhică multi-nivel (ore, zile, săptămâni, luni, trimestre şi ani), *Locaţia geografică* este o dimensiune ierarhică (vecini, oraşe, state şi ţări). În cele mai multe activităţi ale unei firme, ierarhiile sunt o necesitate. Ar fi imposibil de a funcţiona o firmă, dacă toate datele sale ar fi limitate la nivel tranzacţional. De exemplu, este necesar de a păstra informaţii despre volumul vânzărilor lunare, pe trimestru, pe an, pentru a vedea care produse se vând mai bine şi care mai prost. Criteriile după care datele sunt agregate pentru analiză şi raportare trebuie să fie aceleaşi cu factorii folosiţi în procesul decizional. În figura 2.3 este prezentată o ierarhie de produse.

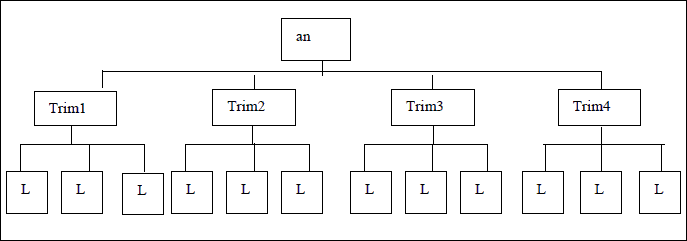


**Figura 2.3** *O ierarhie de produse*

Elementele individuale sau nodurile sunt numite *membri* (de exemplu: cosmetice, echipamente de birou etc). Cei mai mulţi membri au conexiuni în sus şi jos, în ierarhie. Conexiunile în sus sunt de tip (m:1) şi sunt numite *asocieri părinte*. Conexiunile în jos sunt de tip (1:m) şi sunt numite *asocieri copil*. De exemplu, echipamentele electronice sunt părinţi pentru echipamentele de birou. Faxurile şi xerox-urile sunt copii pentru echipamentele de birou. În general, un membru poate avea un singur părinte. Un membru ce nu are părinte se numeşte *membru rădăcină* (root). În figura

* 1. membru *Toate Produsele* este rădacină în ierarhia de produse. Membrii ce nu au copii sunt numiţi *frunză* (de exemplu radio casetofoane cu baterii etc). Pentru a identifica poziţia unui membru într-o dimensiune se folosesc conceptele de *înălţime* şi *adâncime* în ierarhie. *Înălţimea* se stabileşte de jos în sus. Din acest motiv nivelul (L0) al ierarhiei reprezintă nodurile frunză ale ierarhiei (înălţimea cea mai mică). *Adâncimea* în ierarhie este stabilită de sus în jos. Există posibilitatea ca doi membri, care au aceeaşi înălţime, să aibă adâncimi diferite şi invers (structuri arborescente neechilibrate). Structurile arborescente neechilibrate sunt implementate cu succes în bazele de date multidimensionale, bazele de date relaţionale fiind necorespunzătoare. Ierarhiile dimensionale pot fi simetrice sau asimetrice (figura 2.4). În dimensiunea *Timp* există o ierarhie simetrică. Cu ierarhiile simetrice se pot referi membrii prin nivelul lor ierarhic.

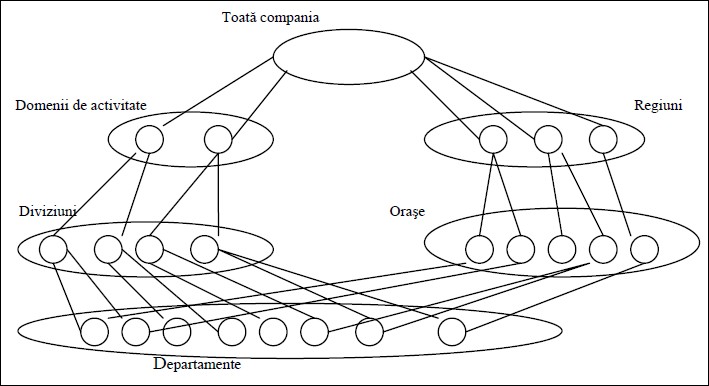
În glosarul de termeni OLAP se specifică că “ *membrii dimensiunilor pot f iorganizaţi pe baza relaţiilor de tip părinte-copil, unde un membru părinte reprezintă consolidarea (agregarea) membrilor copil. Rezultatul este o ierarhie şi relaţiile părinte/copil sunt relaţii ierarhice”.* Produsele se pot grupa şi în alte moduri decât cosmetice şi electronice. De exemplu, preţul reprezintă un alt criteriu pentru organizarea dimensiunii *Produs* (de exemplu: solduri, specifice, de lux). Cu alte cuvinte, într-o dimensiune pot exista mai multe ierarhii. Alegerea nivelurilor intermediare de grupare sau agregare este importantă pentru înţelegerea şi abilitatea de a lua decizii, deoarece datele au numai un nivel de detaliu, un nivel de agregare complet, dar multe niveluri intermediare. Aşa cum arată figura 2.5, fiecare direcţie de agregare scoate în evidenţă unii factori şi-i ascunde pe alţii \*THOM96+. Numărul de niveluri intermediare, cum ar fi grupuri de produse după preţ, grupuri de produse după profit, după tipul produsului şi producător, permite să experimentăm diferite moduri de a privi şi înţelege datele. Aceasta este una din ariile de convergenţă între structurile multidimensionale şi analiza statistică.



**Figura 2.4** *Ierarhie simetrica*

Ierarhiile sunt fundamentul pentru agregarea datelor şi pentru navigarea între nivelurile de detaliu dintr-un *cub n-dimensional*. Deşi nu toate dimensiunile conţin ierarhii, toate aplicaţiile din lumea reală implică dimensiuni ierarhice. Numărul de ierarhii distincte într-o structură multidimensională este egal cu produsul dintre numărul de ierarhii din fiecare dimensiune. De exemplu, dacă sunt trei ierarhii în dimensiunea *Timp*, trei ierarhii în dimensiunea *Locaţie*, patru ierarhii în dimensiunea *Produs* şi o singură variabilă ar putea fi 3\*3\*4\*1=36 ierarhii distincte.

Unele instrumente OLAP permit numai o singură ierarhie pe dimensiune. În acest caz, fiecare ierarhie este tratată ca o dimensiune separată. *Combinaţia de multiple dimensiuni şi multiple niveluri pe dimensiune constituie esenţa unui cub n-dimensional sau hypercub*.

O *celulă* într-un *cub n-dimensional* este definită de intersecţia unui membru din fiecare dimensiune. Unele celule conţin date de intrare cum ar fi vânzările zilnice, altele conţin date derivate. Valorile datelor derivate sunt definite cu ajutorul formulelor. Cu cât sunt mai multe dimensiuni şi ierarhii în cub, cu atât este mai complexă vecinătatea din jurul unei celule şi există mai multe direcţii după care se pot vizualiza datele. Într-un *cub n-dimensional* (cu un nivel ierahic pe dimensiune), fiecare celulă are 2n vecini imediaţi sau direcţii de vizualizare. De exemplu, o celulă într-o foaie de calcul tabelar are patru vecini (4 celule adiacente), iar o celulă într-un cub tridimensional are şase. Când se adaugă ierarhii, numărul de direcţii de vizualizare creşte.

**Figura 2.5** *Agregarea datelor dintr-o dimensiune*

Ce tip de date pot fi stocate într-un *cub n-dimensional*? Deşi majoritatea datelor stocate în *cuburile n-dimensionale* sunt numerice, orice tip de date de la text la grafice şi chiar sunete pot fi multidimensionale. Multe instrumente OLAP oferă abilitatea de a popula *cuburile n-dimensionale* cu date text. Datele numerice sunt mai potrivite pentru aplicaţiile OLAP, deoarece au o organizare multidimensională şi se pot agrega. Alte tipuri de date pot beneficia de organizarea multidimensională, dar apar probleme datorate agregării lor. Totuşi cele mai multe date sursă, la

nivel de întreprindere, sunt de tip caracter. Din acest motiv, datele de tip şir de caractere vor deveni mult mai importante pentru instrumentele OLAP. Date cum ar fi: culoarea, adresa, tipul de ambalaj, tipul de client sunt factori esenţiali pentru analiză.

Conceptul de *nivel într-o ierarhie* este foarte important pentru a determina ce tipuri de navigări se pot executa în dimensiuni şi ce tipuri de calcule suportă dimensiunile. În glosarul OLAP se specifică că “ *doi membrii ai unei ierarhii sunt de aceeaşi generaţie dacă ei au acelaşi număr de strămoşi….. Termenii de generaţie şi nivel sunt necesari pentru a descrie subgrupuri de membrii întrucât, de exemplu, deşi doi fraţi membri au acelaşi părinte şi sunt de aceeaşi generaţie, ei ar putea să nu fie la acelaşi nivel, dacă unul din fraţi are un copil şi celălalt nu.”.*

# Conceptul de măsură

O *măsură* (variabilă) este un indicator de performanţă prin care se poate analiza performanţa activităţii modelate (de regulă un atribut numeric al unui element din colecţia de fapte). Valorile unui indicator se modifică continuu. Pentru fiecare combinaţie posibilă între dimensiuni, există sau nu o valoare corespunzătoare a indicatorilor. Nu orice atribut numeric este un indicator. De exemplu, “dimensiunea ambalajului” este un atribut numeric şi totuşi nu este un indicator ci un atribut dimensional. Dacă valoarea atributului numeric variază continuu (de exemplu: costul de livrare, cantitatea vândută, volumul vânzărilor) atunci atributul este un indicator, iar dacă atributul este perceput mai mult ca o constantă (de exemplu: dimensiunea ambalajului, descriere produs, culoare) atunci este un atribut dimensional. Indicatorii pot fi clasificaţi în:

* + - indicatori de *bază* (de exemplu: volumul vânzărilor, cantitatea vândută, costurile, numărul de clienţi);
    - indicatori *derivaţi* care se obţin prin combinarea indicatorilor de bază (de exemplu profitul).

O altă clasificare este dată de Ralph Kimball în cartea sa “*The Data Warehouse Toolkit*” şi anume după posibilitatea indicatorilor de a se însuma după dimensiuni:

* + - indicatori *aditivi* care se pot însuma după toate dimensiunile. De exemplu, indicatorul “volumul vânzărilor” se poate calcula pentru o categorie de produse sau pentru o anumită regiune sau pentru a anumită perioadă de timp. Volumul vânzărilor*,* cantitatea vândută şi costurile sunt aditive. Aceasta înseamnă că are sens de a aduna lei sau cantităţi de-a lungul oricărei combinaţii timp, produs, magazin.
    - indicatori *semiaditivi* care se pot însuma numai după unele dimensiuni. De exemplu, indicatorul “numărul de clienţi” este semiaditiv, deoarece se poate calcula numărul de clienţi într-o anumită perioadă de timp sau pentru o anumită regiune, dar nu este aditiv de-a lungul dimensiunii *Produs.* Dacă pentru fiecare produs (dintr-o categorie de produse) se cunoaşte numărul de clienţi şi dorim să aflăm numărul de clienţi pentru categoria de produse, nu se pot aduna aceste numere. Rezultatul poate fi eronat, deoarece pot exista clienţi, care au cumpărat mai multe tipuri de produse, din categoria respectivă.
    - indicatori *neaditivi* care nu se pot însuma după nici o dimensiune (de exemplu profitul marginal). Variabilele neaditive pot fi combinate cu alte variabile pentru a deveni aditive.

# Conceptul de multicub

Un *cub n-dimensional* este de fapt un set de variabile ce utilizează aceleaşi dimensiuni de identificare. Întrebarea care apare este dacă se poate adăuga un număr nelimitat de dimensiuni la un *cub n-dimensional* şi dacă pentru modelarea unei activităţi complexe este suficient un singur *cub n- dimensional*. De exemplu, se consideră *cubul n-dimensional* care modelează activitatea de vânzări cu dimensiunile iniţiale: *Timp*, *Locaţie, Produs* şi dimensiunea de variabile \*THOM96+. Se adaugă dimensiunea *Angajat* şi variabila “numărul de ore lucrate/tip de angajat”. În modelul modificat, variabilele iniţiale (de exemplu cantitatea vândută, volumul vânzărilor) continuă să fie identificate de locaţie, timp şi produs la care se adaugă tipul de angajat. În cubul modificat, variabilele pentru vânzări sunt acum în funcţie de locaţie, timp, produs şi angajat. Dar în realitate, vânzările nu depind de tipul angajatului. Într-un caz simplificat (fără dimensiunile *Locaţie* şi *Timp*) se consideră dimensiunea *Angajat* cu 10 membri, dimensiunea *Produs* cu 10 membri. Ca variabile se consideră numai “volumul vânzărilor” pentru fiecare produs şi “totalul vânzărilor” pentru toate produsele şi “numărul de ore lucrate” pentru fiecare tip de angajat, precum şi “total număr de ore lucrate” pentru toţi angajaţii. Cele două seturi de date au câte 11 puncte de date fiecare. Ierarhiile au fost eliminate din cub. Membrii “Toti angajaţii” şi “Toate produsele” au fost adăugaţi la fiecare din dimensiunile respective, pe un singur nivel. Cubul rezultat este format din 11\*11\*2 = 242 de intersecţii (celule). Din acestea numai 22 de celule au date, celelalte sunt goale. Cu alte cuvinte rezultă un *cub ndimensional* foarte împrăştiat. Apare necesitatea de a defini un nou *cub ndimensional* logic şi un mod de a determina când o dimensiune nouă aparţine sau nu unui nou *cub n-dimensional*. Dacă două seturi de date aparţin aceluiaşi *cub ndimensional* logic, densitatea combinaţiilor lor va fi egală cu media aritmetică a densităţilor lor înainte de a fi combinate. De exemplu, un cub perfect dens ce conţine dimensiunea *Locaţie* (cu 100 de membri), dimensiunea *Timp* (cu 10 perioade de timp), dimensiunea *Produs* (cu 100 de produse) şi dimensiunea de variabile (cu 5 variabile) va avea 500000 puncte de date. Un cub ce conţine dimensiunea *Locaţie* (cu 100 de magazine), dimensiunea *Timp* (cu 10 perioade de timp), dimensiunea *Produs* (cu 100 de produse), dimensiunea de variabile (cu 5 variabile) şi dimensiunea *Scenariu* (planificat, curent şi variaţia), iar datele există numai pentru scenariu planificat, va conţine 500000 celule de date, dar este numai 33 % dens. Combinaţia celor două cuburi va conţine 1000000 celule de date şi este 67% dens.

În multe aplicaţii multidimensionale se utilizează date din mai multe domenii de activitate (de exemplu activitatea financiară şi activitatea de marketing sau activitatea de producţie şi activitatea de desfacere). În aceste cazuri, se utilizează mai multe *cuburi n-dimensionale* logice sau *structuri multicub*. Problema care apare în proiectarea unei *structuri multicub* este legată de modul cum se realizează legătura între *cuburile n-dimensionale*. Se consideră cele două cuburi din exemplu anterior: cubul cu date despre vânzări şi cubul cu date despre angajaţi. Cele două cuburi au două dimensiuni comune: *Locaţie* şi *Timp*. Cubul pentru vânzări are în plus dimensiunile *Produs, Scenariu* şi dimensiunea de variabile. Niciuna din aceste dimensiuni nu este folosită de cubul pentru angajaţi. Cubul pentru angajaţi are dimensiunea *Angajat* şi dimensiunea de variabile legate de activitatea angajaţilor (de exemplu numărul total de ore lucrate/tip angajat) care nu sunt folosite de cubul pentru vânzări. Se doreşte să se analizeze dacă există o relaţie între variabila “numărul de ore lucrate/tip de angajat” şi variabila “cantitatea de produse vândute”. Pentru a compara valorile celor două variabile, care se găsesc în două *cuburi ndimensionale* separate, trebuie mai întâi să se definească un cadru analitic sau un numitor comun \*THOM96+. În cazul celor două *cuburi n- dimensionale*, numitorul comun sunt cele două dimensiuni comune: *Locaţie* şi *Timp*. În acest fel se pot compara valori individuale, serii dimensionale sau volume dimensionale. Figura 2.6 arată o schemă pentru acest model multicub. Se observă că modelul conţine dimensiunile globale *Locaţie* şi *Timp* şi fiecare *cub n-dimensional* din model este o ramificaţie a dimensiunilor globale. Prin utilizarea structurii multicub este posibil de a integra seturi de date eterogene.

În concluzie, cel mai simplu mod de reprezentare a datelor unei aplicaţii multidimensionale este cel al unui spaţiu cartezian definit de toate dimensiunile aplicaţiei (spaţiul datelor). Totuşi datele

multidimensionale sunt împrăştiate şi celulele de date nu sunt distribuite în mod egal în spaţiul multidimensional.

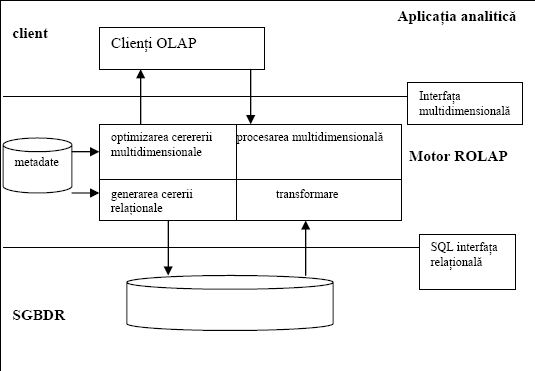
Proiectanţii de instrumente OLAP au adoptat o varietate de strategii pentru a trata fenomenul de împrăştiere dar şi gruparea datelor. Instrumente ca Essbase, Power Play utilizează structura *hypercub*, o structură logică de cub simplu, dar cu un model sofisticat de compresie a datelor. Cealaltă structură, *multicubul,* este mai des întâlnită. În aplicaţiile multicub, proiectanţii descompun baza de date într-un set de structuri multidimensionale, fiecare fiind compusă dintr-un subset de dimensiuni ale bazei de date. Aceste structuri multidimensionale au diferite denumiri (variabile – Oracle Express, Pilot; structuri – Holos; cuburi – Microsoft OLAP).

# Arhitectura sistemelor OLAP Sisteme ROLAP

In ultimii ani, un numar mare de instrumente OLAP permit stocarea datelor multidimensionale in baze de date relationale (depozite de date). Chiar daca o aplicatie OLAP stocheaza toate datele sale intr-o baza de date relationala, totusi ea va lucra separat, pe o copie a datelor (depozite de date/centre de date), nu pe baza de date tranzactionala. Aceasta permite sa fie structurata optim pentru OLAP, mai bine decat pentru alte aplicatii. Datele multidimensionale pot fi stocate in depozite de date/centre de date, utilizand schema *stea* sau *fulg de zapada*. Problema este de a oferi acces rapid si flexibilitate in manipularea multidimensionala. Totusi *tabela de fapte* este un mod ineficient de a stoca volume mari de date. Daca sunt multe variabile, exista posibilitatea de a nu le putea stoca pe toate intr-o singura *tabela de fapte* (gradul de imprastiere poate diferi mult intre grupurile de variabile). Datele pot fi incarcate din multiple surse, astfel actualizarile unei singure *tabele de fapte* foarte mare sunt ineficiente. De aceea, in aplicatiile complexe, *tabela de fapte* este partitionata in grupuri de variabile dupa gradul de imprastiere, stabilindu-se, de asemenea, dimensiunile corespunzatoare si sursele de date. In aplicatiile OLAP complexe pot exista 10 - 20 de *tabele de fapte*. Pentru o buna performanta la interogare, unele agregari trebuie antecalculate si stocate in tabele de agregate. In aplicatiile complexe pot exista mii de tabele de agregate.

In figura 2.4.1, este prezentata arhitectura unui sistem ROLAP. Un motor ROLAP face trecerea dinamica intre modelul de date multidimensional logic *M* si modelul de stocare relational *R*. Tehnic vorbind acest motor implementeaza o transformare a unei cereri multidimensionale *m* pe modelul de date multidimensional logic *M,* intr-o cerere relationala *r* pe modelul de stocare *R.* Eficienta la interogare este factorul dominat asupra performantei globale a sistemului si scalabilitatii. Strategiile de optimizare sunt factorii principali in diferentierea sistemelor ROLAP existente. Preocuparea majora este gasirea celei mai eficiente reprezentari a cererii pentru un SGBDR dat si a schemei date, precum si incarcarea dinamica echilibrata intre SGBDR si motorul ROLAP. Intrucat optimizatorul de cereri al SGBDR ar putea sa nu fie capabil sa aleaga planul de executie optim pentru cereri, unele instrumente (de exemplu DSS Server/Microstrategy) impart cererea complexa in mai multe subcereri, care sunt executate secvential (SQL in mai multi pasi). Sa consideram QR(m) setul tuturor reprezentarilor relationale ale cererii multidimensionale *m* pentru modelul relational *R* (incluzand toate variantele in mai multi pasi). Pentru fiecare cerere, motorul ROLAP trebuie sa aleaga o reprezentare optima pe baza unui criteriu (de exemplu performanta cererii).

Unele operatii multidimensionale nu pot fi exprimate usor folosind limbajul SQL. Este necesar pentru motorul ROLAP sa implementeze aceste operatii folosind structuri de date multidimensionale. Aceasta complica problema de optimizare prin extinderea spatiului de cautare QR(m). Incarcarea echilibrata intre serverul de baza de date si motorul ROLAP este un rezultat al strategiei utilizate pentru a alege cererea optima. Evaluarea operatiilor multidimensionale de catre motorul ROLAP poate reduce timpii de procesare a unei cereri, dar si scalabilitatea globala a sistemului.



***Figura 2.4.1*** *Arhitectura unui sistem ROLAP*

Factorii ce influenteaza decizia de optimizare pot fi impartiti in *statici si dinamici* (ce pot fi evaluati numai la momentul executiei). *Factorii statici* sunt: natura operatiei multidimensionale, complexitatea operatiei executate de motorul ROLAP fata de complexitatea implementarii relationale. *Factorii dinamici* includ caracteristicile de incarcare ale serverului de baza de date si ale motorului ROLAP, volumul datelor curente ce trebuie sa fie transferate de la sistemul relational la motorul ROLAP. De exemplu, Decision Suite/Microstrategy permite administratorului sa determine locatia procesarii prin metadate.

Alte probleme ale instrumentelor ROLAP sunt metamodelele (pentru o schema *stea* sau *fulg de zapada* metamodelul este foarte important), seturile de functii complexe etc. Limbajul SQL standard nu permite operatii multidimensionale. Specialistii ofera trei solutii pentru a adauga functionalitate multidimensionala la datele stocate in tabelele SQL:

* integrarea procesarii multidimensionale in sistemul de gestiune a bazei de date relationale, fie prin extinderea limbajului SQL sau prin adaugarea functionalitatii multidimensionale in nucleul SGBD-ului;
* executarea in mai multi pasi a comenzilor SQL (multipass SQL). Instrumentul OLAP realizeaza o serie de comenzi SELECT, in care iesirile comenzilor anterioare sunt stocate in tabele temporale, care sunt apoi utilizate de comenzile urmatoare;
* incarcarea datelor relevante din tabelele corespunzatoare pe un server de aplicatie intermediar, unde este realizata procesarea multidimensionala.

Datorita modului complicat in care datele sunt stocate pe disc, instrumentele OLAP ce folosesc baze de date relationale permit numai citirea datelor. Alte instrumente trebuie sa fie utilizate pentru actualizarea datelor de baza si a tabelele de agregate. Aceste instrumente ROLAP nu pot fi folosite pentru analize de tip “*what-if*”.

In concluzie, stocarea datelor multidimensionale se face intr-o baza de date relationala atunci

cand:

* Volumul de date este prea mare pentru a fi duplicat. Datele atomice nu sunt copiate in baza de date multidimensionala ci sunt stocate in baze de date relationale sursa (depozite de date/centre de date) si citite cand este nevoie;
* Datele sursa se modifica frecvent si este mai bine de a citi in timp real decat din copii;
* Integrarea cu alte sisteme informatice relationale existente;
* Firma are o politica de neduplicare a datelor in alte structuri de fisiere, pentru securitate sau alte motive, chiar daca aceasta conduce la aplicatii mai putin eficiente;

Cateva avantaje ale sistemelor ROLAP sunt:

* se integreaza cu tehnologia si standardele existente;
* sistemele MOLAP nu permit cereri ad-hoc eficace, deoarece sunt optimizate pentru operatii multidimensionale;
* actualizarea sistemelor MOLAP este dificila;
* sistemele ROLAP sunt adecvate pentru a stoca volume mari de date, prin utilizarea procesarii paralele si a tehnologiilor de partitionare etc;
* sistemele MOLAP sunt limitate la 5-10 dimensiuni si sunt adecvate pentru aplicatii departamentale cu volume mici de date si dimensionalitate limitata, iar sistemele ROLAP au o arhitectura flexibila ce ofera support pentru o varietate mare de SSD-uri si cerinte analitice ;
* volatilitatea descrie gradul la care datele si structurile de date se modifica in timp. Datele cu un nivel de volatilitate scazut raman relativ constante. De exemplu, datele despre timp au un nivel scazut de volatilitate (zilele sunt grupate intotdeauna in luni si lunile sunt intotdeauna grupate in ani). Datele despre produse, angajati, clienti se modifica frecvent. Volatilitatea datelor afecteaza cerintele de procesare in lot ale sistemului. Ori de cate ori datele atomice se modifica, datele agregate, ce au fost antecalculate, trebuie sa fie recalculate. De aceea, datele volatile au un impact mai mare asupra unui sistem, care contine un volum mare de informatii agregate, decat asupra unui sistem care calculeaza valorile agregate la momentul executiei. Atat sistemele ROLAP cat si cele MOLAP sunt optime pentru aplicatii cu volatilitate scazuta a datelor. Pentru aplicatiile cu volatilitate ridicata a datelor, numai sistemele ROLAP sunt optime.

# Sisteme MOLAP

Cel mai evident mod de a stoca datele multidimensionale este ca o simpla matrice. Spatiul este rezervat pentru fiecare combinatie posibila intre membrii dimensiunilor. In unele instrumente, numai valorile *nenule* sunt stocate. Numai matricile ce contin date sunt stocate fizic. Fiecare dimensiune a matricii reprezinta o dimensiune a cubului. Continutul matricii sunt *masurile* cubului. O astfel de matrice are multe avantaje: se identifica rapid locatia oricarei celule, iar celulele pot fi actualizate fara a afecta locatia fizica a celorlalte celule. Totusi stocarea datelor pe disc folosind o simpla matrice, ce reflecta direct viziunea utilizatorului, este foarte rar eficienta. Chiar si atunci cand datele sunt stocate in memorie, este adesea necesar de a pastra datele intr-un format mai complex decat o simpla matrice. Aceasta are legatura cu fenomenul de imprastiere si cu cerinta de a modifica dimensiunile, fara sa fie nevoie sa se reconstruiasca din nou intreaga matrice. In cazul stocarii pe disc a datelor imprastiate, datele sunt citite in blocuri si daca fiecare bloc are un grad de imprastiere mare, multe blocuri goale sau aproape goale vor fi stocate in memorie, iar memoria va fi utilizata de mult mai multe ori decat este necesar.

Sistemele MOLAP au pus accentul pe *flexibilitatea si optimizarea tehnicilor de stocare si pe conceptul de tranzactie*. Sistemele MOLAP nu au inca o tehnologie pentru stocarea si gestionarea datelor unanim acceptata. Stocarea fizica a datelor multidimensionale, precum si fenomenul de imprastiere sunt preocupari majore, in domeniul bazelor de date multidimensionale. O tehnica de stocare a datelor optima ar trebui sa tina cont de multi factori dinamici si anume:

* profilul datelor si volumul lor (numarul de dimensiuni si membrii ai dimensiunilor, tipuri de date etc);
* fenomenul de imprastiere (in care dimensiuni sau combinatii de dimensiuni, tipul de imprastiere);
* frecventa de modificare in sursele de date (cat de des vor fi actualizate bazele de date multidimensionale);
* frecventa de modificare in datele multidimensionale;
* frecventa de modificare in modelul multidimensional;
* accesul concurent.

Ideal un SGBD multidimensional (SGBDMD) ar trebui sa aleaga structura de date optima in functie de acesti factori. In cele mai multe SGBDMD comerciale se utilizeaza o tehnica de stocare pe doua niveluri. La nivelul inferior sunt stocate toate dimensiunile dense. Nivelul superior contine dimensiunile imprastiate ca o structura index, care contine pointeri la cuburile de date dense din nivelul inferior.

Unele instrumente OLAP ofera administratorului un numar foarte limitat de optiuni de optimizare. De exemplu, Arbor Essbase are o metoda proprie pentru stocarea si incarcarea datelor multidimensionale in memorie. Aceasta metoda utilizeaza o structura multinivel (cu un numar arbitrar de niveluri pentru diferitele grade de imprastiere). Administratorul poate specifica dimensiunile dense si imprastiate, care construiesc cele doua niveluri. Oracle Express suporta, de asemenea, o structura pe doua niveluri. Pilot Decision Support Suite (Pilot Software) suporta asa numitele *multicuburi*. Timpul este tratat ca o dimensiune densa, iar toate celelalte dimensiunile sunt considerate imprastiate.

O alta problema este transferul conceptului de tranzactie asa cum este cunoscut si acceptat in lumea relationala la SGBDMD. Putine instrumente MOLAP (de exemplu Arbor Essbase ) permit acces multiutilizator concurent atat la citire cat si la scriere. Majoritatea instrumentelor MOLAP permit acces multiutilizator la citire si monoutilizator la scriere. SGBDMD blocheaza intreaga baza de date in timpul actualizarilor (o forma foarte simpla de acces concurent). De asemenea, multe instrumente MOLAP au o notiune foarte vaga a conceptului de tranzactie. Modificarile in cuburile de date pot fi executate ca adaugari in cub sau in timpul analizei de tip “*what if”.* Adesea ele cer o actualizare incrementala a agregatelor sau masurilor, care sunt calculate pe baza de formula. Astfel de dependente fac actualizarile mult mai complicate. Conceptul de tranzactie este legat de multe alte probleme cum ar fi:

* *proprietatile ACID (atomic, consistency, isolation, durability).* Pentru a realiza proprietatile ACID, toti termenii (in special consistenta si izolarea) trebuie reanalizati. De exemplu, dependentele intre datele de detaliu, datele agregate si masuri complica notiunea de consistenta;
* *mecanismul de blocare.* Daca controlul concurential este realizat printr-o tehnica de blocare, trebuie sa fie definite modurile de blocare si nivelul la care se face blocarea. Blocarea intregii baze de date nu este o solutie foarte potrivita. Interdependentele intre date fac ca definirea nivelurilor de blocare sa fie o problema complexa;
* *strategia de propagare a modificarilor.* Datele (agregate si masuri) trebuie modificate in conformitate cu modificarile din datele de detaliu sau din modelul de date.

Sunt si alte probleme importante, deja rezolvate in SGBDR, dar care sunt nerezolvate sau numai partial rezolvate in SGBDMD cum ar fi: restaurarea bazei de date, conceptul de tabela virtuala etc.

Avantajul sistemelor MOLAP este ca ofera o viziune multidimensionala directa a datelor, in timp ce sistemele ROLAP sunt o “interfata multidimensionala” la datele relationale. SGBDMD cer antecalcularea tuturor agregatelor posibile, astfel sunt adesea mai performante decat SGBDR traditionale, dar mai dificil de actualizat si administrat. Deoarece, bazele de date multidimensionale folosesc acelasi motor atat pentru stocare cat si pentru procesarea datelor si acest motor are

informatii complete despre structurile de date multidimensionale si manipularile multidimensionale, este usor pentru instrument de a manipula datele multidimensionale si de a face calcule corecte si complexe. Totusi multe baze de date multidimensionale nu ofera facilitatea de recuperare a erorilor si alte facilitati specifice bazelor de date relationale.

Cateva avantaje ale sistemelor MOLAP sunt:

* tabelele relationale nu sunt potrivite pentru date multidimensionale;
* matricile multidimensionale permit stocarea eficienta a datelor multidimensionale;
* limbajul SQL nu este corespunzator pentru operatii multidimensionale.

Tabelul 2.4.1 prezinta o analiza comparativa intre sistemele ROLAP si sistemele MOLAP.

|  |  |  |
| --- | --- | --- |
| **Criterii** | **MOLAP/Baze de date**  **multidimensionale** | **ROLAP/Baze de date**  **relationale** |
| Spatiul de disc | ocupat mare, daca agregatele sunt antecalculate (explozia datelor derivate si fenomenul de  imprastiere) | posibil zero, daca sunt folosite baze de date existente  nemodificate  (depozite de date virtuale), dar poate fi mare daca noi  structuri  sunt create |
| Performanta la  incarcarea datelor | moderata | scazuta |
| Viteza de calcul | mare | mica |
| Volumul datelor  atomice | mediu la mare (Mb-  Gb) | extrem de mare (Gb-  Tb) |
| Posibilitatea de acces  la  date de catre alte aplicatii (integrare cu alte sisteme existente) | limitata | excelenta in principiu, dar poate fi limitata  daca  este folosita o schema complexa |
| Accesul la date | incet si adesea limitat,  citire/scriere | performanta  moderata |
| Dimensionalitate | mica (modele multidimensionale  simple, 5-10  dimensiuni) | mare (modele multidimensionale complexe) |
| Modificarea  dimensiunilor | buna dar baza de date  trebuie sa fie off-line | buna |
| Volatilitatea datelor | adecvate pentru  volatilitate scazuta | adecvate pentru  volatilitate ridicata |
| Facilitati de  administrare  a SGBD-ului | putine | foarte puternice |
| Usurinta de a construi aplicatii de catre  utilizatorii finali | moderata | aproape imposibila |
| Arhitectura | client/server pe doua  sau trei  niveluri, lipsa | client/server pe doua  sau  trei niveluri, |

|  |  |  |
| --- | --- | --- |
|  | standardelor | arhitectura  deschisa, standarde |
| Managementul  metadatelor | nu | da |
| Limbaj de interogare | specific fiecarui  instrument | SQL |
| Facilitati de calcul | foarte complexe, in  toate  dimensiunile | limitate |
| Jonctiuni | nu | da |
| Agregari dinamice | nu | da |
| Partitionarea datelor | nu | da |
| Cereri paralele | nu | da |
| Algoritmi hash | da | da |
| Indexare | da | da |
| Masuri derivate | da | da |
| Comparatii ale  perioadelor de timp | da | da |
| Analiza valutelor | nu | da |
| Previziuni | da | da |
| Consolidari financiare | da | da |
| Tipul aplicatiilor | la nivel departamental | la nivelul  intreprinderii |

***Tabelul 2.4.1*** *Analiza comparativa intre sistemele ROLAP si sistemele MOLAP*

# Sisteme HOLAP

Un sistem OLAP hibrid (HOLAP) este un sistem care utilizeaza pentru stocarea datelor atat baze de date relationale cat si baze de date multidimensionale, in scopul de a beneficia de caracteristicile corespunzatoare si de tehnicile de optimizare. Un sistem HOLAP trebuie sa aiba urmatoarele **caracteristici**:

* *transparenta locatiei si a accesului* (locatia datelor utilizate trebuie sa fie ascunsa pentru utilizator). Transparenta accesului permite utilizatorului sa acceseze datele la fel, indiferent daca sunt stocate in BDR sau BDMD;
* *transparenta fragmentarii.* Fragmentarea datelor trebuie sa fie invizibila pentru utilizator (cererile utilizatorului trebuie sa fie descompuse in cereri partiale in functie de sistemul de stocare);
* *transparenta performantei.* Trebuie furnizate tehnici de optimizare pentru ambele sisteme MOLAP si ROLAP;
* *un model de date comun* utilizat atat de datele relationale cat si de datele multidimensionale;
* *alocarea optima in sistemele de stocare*. Sistemele hibride ar trebui sa beneficieze de strategiile de alocare specifice sistemelor distribuite.

Sunt diferite arhitecturi pentru un sistem hibrid OLAP:

* integrarea sistemelor MOLAP si ROLAP printr-o interfata comuna. Clientul OLAP poate fi luat in considerare intr-o astfel de solutie, intrucat ofera o interfata comuna pentru SGBDMD si motoarele ROLAP. Totusi multe din cerintele listate mai sus nu sunt satisfacute. De exemplu,

Seagate Holos utilizeaza aceasta arhitectura, permite tehnici de stocare relationale si multidimensionale integrate in asa numita arhitectura OLAP compusa;

* integrarea mutuala a sistemelor ROLAP si MOLAP. Aceasta arhitectura poate fi gasita la Arbor Essbase, care ofera acces la datele relationale (IBM DB2 OLAP Server);
* extensii la SGBDR sau SGBDOR prin utilizarea tehnologiei *datablade* (de exemplu Informix cu optiunea Metacube). Totusi acesta nu este un system OLAP hibrid (Metacube este un motor ROLAP, deci nu este inca implicat un SGBDMD).

Aplicatiile complexe si cu grad mare de imprastiere vor folosi o combinatie a acestor moduri de stocare. Datele care sunt utilizate cel mai des vor fi stocate in memoria RAM. Datele care sunt utilizate regulat, dar nu frecvent pot fi stocate in baze de date multidimensionale optimizate. In final, volumele mari de date detaliate sunt stocate in baze de date relationale sursa.

Pentru aplicatii foarte imprastiate, o solutie hibrida este probabil cea mai buna. Aria graficului, in care sistemele MOLAP sunt recomandate, reflecta abilitatea lor de a stoca cel mai eficient volume medii pana la mari de date. Pentru date foarte imprastiate sau pentru baze de date foarte mari, o strategie de stocare de tip baza de date relationala poate fi singura optiune fezabila. In general, daca se doreste implementarea unei singure aplicatii, este mai eficient din punct de vedere al costului de a selecta un instrument mai simplu, decat unul proiectat special pentru acea aplicatie. Pentru scopuri strategice si aplicatii complexe poate fi necesar de a achizitiona un instrument complex. In functie de tipul bazei de date, se poate allege tehnica de indexare folosita. Cele mai multe baze de date multidimensionale stabilesc automat tehnica de indexare.

# Sisteme informatice pentru inteligenta afacerilor (Business Intelligence)

In pagina Web a firmei IBM, conceptul de *Business Intelligence* (BI) (inteligenta afacerii) este definit astfel:

*“BI inseamna utilizarea tuturor datelor de care dispune o firma, pentru a imbunatati procesul decizional. BI presupune accesul la date, analiza lor si descoperirea de noi oportunitati de utilizare a lor.”*

In climatul competitional al zilelor noastre, este vital pentru organizatii sa ofere acces rapid la informatii, la costuri mici, pentru un numar cat mai mare si mai variat de utilizatori. Solutia la aceasta problema este un sistem BI care ofera un set de tehnologii si produse software ce livreaza utilizatorilor informatiile necesare pentru a raspunde la intrebarile ce apar in rezolvarea problemelor de afaceri:

## Nevoia de a creste veniturile si de a reduce costurile.

S-au dus zilele cand utilizatorii finali puteau gestiona si planifica activitatile utilizand rapoarte lunare si organizatiile IT aveau mult timp la dispozitie pentru a implementa noi aplicatii. Astazi firmele trebuie sa dispuna rapid de aplicatii, sa ofere utilizatorilor acces rapid si usor la informatiile, ce reflecta schimbarile mediului de afaceri. Sistemele BI pun accentul pe accesul si livrarea rapida a informatiilor la utilizatori.

## Nevoia de a gestiona si modela complexitatea mediului de afaceri curent.

A intelege si gestiona un mediu de afaceri complex si a maximiza investitiile devine mult mai dificil. Sistemele BI ofera mai mult decat mecanisme de interogare si raportare, ele ofera instrumente de analiza a informatiilor complexe si de *data mining*.

## Nevoia de a reduce costurile IT.

Astazi, investitia in sistemele informatice este un procent semnificativ din cheltuielile firmelor si nu este necesar numai sa se reduca aceste cheltuieli, dar de asemenea, sa se obtina beneficii maxime de la informatiile gestionate de sistemele IT. Noile tehnologii informatice ca

Intranetul si arhitectura pe trei niveluri, reduc costul de utilizare a sistemelor BI de catre o mare varietate de utilizatori, in special manageri.

In cele ce urmeaza, se va prezenta evolutia *sistemelor informatice pentru inteligenta afacerii:*

Primele *sisteme informatice pentru afaceri* foloseau aplicatii a caror iesiri implicau volume uriase de hartie, pe care utilizatorii trebuiau sa le citeasca, pentru a obtine raspunsurile dorite. Aplicatiile client/server cu clienti de tip terminal permiteau un acces mai rapid la date, dar erau totusi greu de utilizat si cereau acces la baze de date operationale complexe. Aceste *sisteme informatice pentru afaceri* erau folosite numai de analisti. Managerii si directorii executivi puteau foarte rar sa utilizeze aceste sisteme.

A doua generatie de *sisteme informatice pentru afaceri* a aparut odata cu depozitele de date, care au o serie de avantaje fata de sistemele din prima generatie:

* depozitele de date sunt proiectate pentru a satisface nevoile managerilor si nu a aplicatiilor tranzactionale;
* informatia din depozitele de date este curata, consistenta si este stocata intr-o forma pe care managerii o inteleg;
* spre deosebire de sistemele operationale, care contin numai date de detaliu curente, depozitele pot furniza atat informatii istorice cat si agregate;
* utilizarea unei arhitecturi client/server ofera utilizatorilor de depozite de date interfete imbunatatite si instrumente suport de decizie mai puternice.

*A treia generatie: BI.* Un depozit de date nu este totusi o solutie completa pentru nevoile managerilor. Un punct slab al solutiilor ce folosesc depozitele de date este ca specialistii pun accentul pe tehnologie si mai putin pe solutii manageriale (business solutions). Desi producatorii de depozite de date ofera instrumente puternice pentru construirea si accesarea unui depozit de date, aceste instrumente cer un volum semnificativ de munca pentru implementare. De asemenea, se pune prea mult accent pe procesul de construire a depozitului si mai putin pe accesul la datele din depozit. Multe organizatii considera ca daca construiesc un depozit de date si ofera utilizatorilor instrumente corecte, problema este rezolvata. De fapt este tocmai inceputul. Desi informatia din depozit este complet documentata si usor de accesat, complexitatea va limita utilizarea depozitului de catre manageri, principalii beneficiari*.* Sistemele pentru inteligenta afacerii pun accentul pe imbunatatirea accesului si livrarii de informatii utile atat la consumatorii de informatii cat si la cei care ofera informatii. Un sistem informatic pentru inteligenta afacerii trebuie sa ofere scalabilitate si sa fie capabil sa suporte si sa integreze instrumente software de la mai multi fabricanti. Un depozit de date este una din sursele de date ale unui sistem BI. De asemenea, exista un volum mare de informatii pe serverele de Web ale Intranetului, pe Internet si in format de hartie. Sistemele informatice pentru inteligenta afacerilor sunt proiectate pentru a permite acces la toate formele de informatii, nu numai cele stocate intr-un deposit de date.

Intr-o firma se colecteaza volume mari de date in tranzactiile zilnice: date despre comenzi, stocuri, facturi, vanzari, clienti etc. De asemenea, firmele au nevoie si de informatii externe (de exemplu informatii demografice). A fi capabil sa consolidezi si sa analizezi aceste date poate conduce adesea la un avantaj competitional (cresterea vanzarilor, reducerea costurilor de productie, imbunatatirea activitatii de desfacere, descoperirea unor noi surse de venit etc). Toate acestea sunt posibile daca exista aplicatii corespunzatoare si instrumente necesare pentru a analiza datele si daca datele sunt intr-un format corespunzator pentru analiza.

Astfel, un sistem BI are trei avantaje cheie:

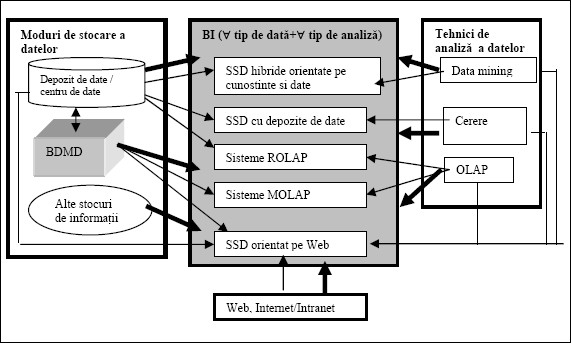
* include in arhitectura sa cele mai avansate tehnologii informatice;
* pune accentul pe accesul si livrarea de informatii la utilizatorii finali si ofera suport atat pentru specialisti cat si pentru utilizatorii finali;
* permite acces la toate formele de informatii, nu numai cele stocate intr-un depozit de date.

Principalele obiective ale unui sistem BI sunt:

* sa permita solutii cu costuri scazute ce ofera avantaje firmei;
* sa permita acces rapid si usor la informatiile firmei pentru un numar mare si variat de utilizatori ;
* sa ofere suport pentru tehnologiile moderne (tehnici de analiza complexe, instrumente OLAP, instrumente de tip data mining etc);
* sa ofere un mediu de operare deschis si scalabil.

Observam ca sistemele informatice pentru inteligenta afacerilor sunt de fapt *sisteme suport de decizie moderne la nivel de organizatie,* care utilizeaza noile tehnologii informatice. Termenul de *sistem informatic pentru inteligenta afacerilor* este de fapt un termen “umbrela” utilizat de specialisti pentru o categorie mai vasta de sisteme suport de decizie, ce integreaza toate facilitatile oferite de depozitele de date, instrumentele OLAP, instrumentele data mining, Web-ul etc. In functie de complexitatea procesului decizional la nivel de organizatie, de numarul de utilizatori, de cerintele organizatiei, de volumul de informatii necesare procesului decizional si de alti multi factori, sistemele suport de decizie moderne vor utiliza si vor integra una sau mai multe din noile tehnologii informatice actuale. Daca se utilizeaza depozite de date/centre de date si instrumente de interogare si raportare atunci avem un sistem suport de decizie cu depozite de date. Daca se integreaza depozitele de date cu instrumentele OLAP se obtin asa numitele sisteme ROLAP, iar daca se utilizeaza si facilitatile oferite de Web se obtin sistemele suport de decizie orientate pe Web (figura 2.5.1).

Multe firme prefera sa construiasca un sistem separat pentru aplicatii BI fie din motive de securitate, fie din motive de performanta ale sistemelor operationale.



***Figura 2.5.1*** *Influenta noilor tehnologii informatice in evolutia sistemelor suport de decizie moderne*

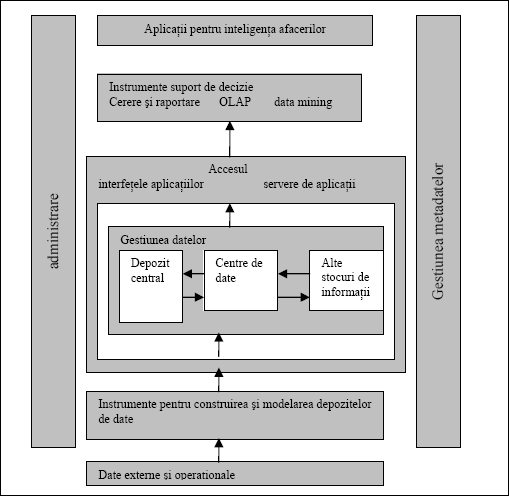
La ora actuala exista o gama variata de instrumente si metodologii valabile pentru a dezvolta solutii BI :

* Aplicatii cum ar fi IBM’s DecisionEdge pentru managementul relatiilor cu

clientii, Oracle Sales Analyzer pentru analiza activitatii de marketing, Oracle Financial Analyzer pentru analiza activitatii financiare etc;

* Instrumente pentru interogari cum ar fi Power Play-Cognos, Business Objects- Business Objects, IBM’s Query Management Facility etc;
* Instrumente OLAP cum ar fi Essbase-Arbor Software, Express Analyzer, Express Objects-Oracle etc;
* Instrumente pentru analiza statistica cum ar fi SAS System-SAS Institute, etc;
* Instrumente pentru data mining cum ar fi IBM’s Intelligent Miner.

Multe din aceste aplicatii si instrumente au facilitati Web. In figura 2.5.2 este prezentata arhitectura unui *sistem informatic pentru inteligenta afacerilor* sau a unui *sistem suport de decizie modern la nivel de organizatie*.



***Figura 2.5.2*** *Arhitectura unui sistem informatic pentru inteligenta afacerilor*

# Arhitectura ORACLE Business Intelligence

**Arhitectura Serverului si a Depozitului de date**

Arhitectura Serverului de Business Intelligence si a depozitului de date prevede un nivel abstract ce permite utilizatorilor sa transmita interogari logice SQL catre sursa de date asociata.

# Serverul de Business Intelligence

Serverul de Business Intelligence este o componenta a platformei de BI ce proceseaza cererile si interogarile asupra sursei de date. Serverul de BI mentine modelul logic de date si permite accesul clientului asupra acestui model prin serverul de baze de date. Serverul de BI foloseste metadata in cadrul depozitului de date pentru a realiza urmatoarele taskuri:

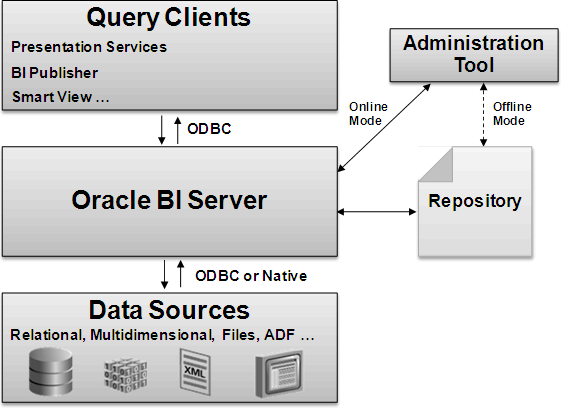
* Interpretarea interogarilor SQL si scrierea interogarilor fizice corespunzatoare aupra celei mai apropiate surse de date;
* Transformarea si combinarea rezultatelor fizice realizate si actualizarea calculelor

finale.

Serverul de BI se conecteaza la sursa de date fie prin intermediul serverului de baze de date,

fie prin API.

Tool-ul client de administrare este o aplicatie Windows si este folosit pentru crearea si editarea depozitelor de date BI. Acest tool se poate conecta la depozitul de date in modul offline, sau se poate conecta prin Serverul de BI,online. Unele configurari se pot face numai in modul online. Figura 3.1 prezinta interactiunea dintre Serverul de BI si interogarile clientilor, sursele de date, depozitele de date si tool-ul de administrare.



**Figure 3.1** *BI Server Architecture*

Sa presupunem ca Serverul de BI primeste urmatoarea cerere: SELECT

"D0 Time"."T02 Per Name Month" saw\_0, "D4 Product"."P01 Product" saw\_1,

"F2 Units"."2-01 Billed Qty (Sum All)" saw\_2

FROM "Sample Sales" ORDER BY saw\_0, saw\_1

Serverul de BI va converti aceasta interogare SQL intr-o interogare fizica, dupa cum urmeaza: WITH SAWITH0 AS (

select T986.Per\_Name\_Month as c1, T879.Prod\_Dsc as c2, sum(T835.Units) as c3, T879.Prod\_Key as c4

from

Product T879 /\* A05 Product \*/ , Time\_Mth T986 /\* A08 Time Mth \*/ ,

FactsRev T835 /\* A11 Revenue (Billed Time Join) \*/

where ( T835.Prod\_Key = T879.Prod\_Key and T835.Bill\_Mth = T986.Row\_Wid) group by T879.Prod\_Dsc, T879.Prod\_Key, T986.Per\_Name\_Month

)

select SAWITH0.c1 as c1, SAWITH0.c2 as c2, SAWITH0.c3 as c3 from SAWITH0

order by c1, c2

# Tipuri de Layere din Depozitul de Date

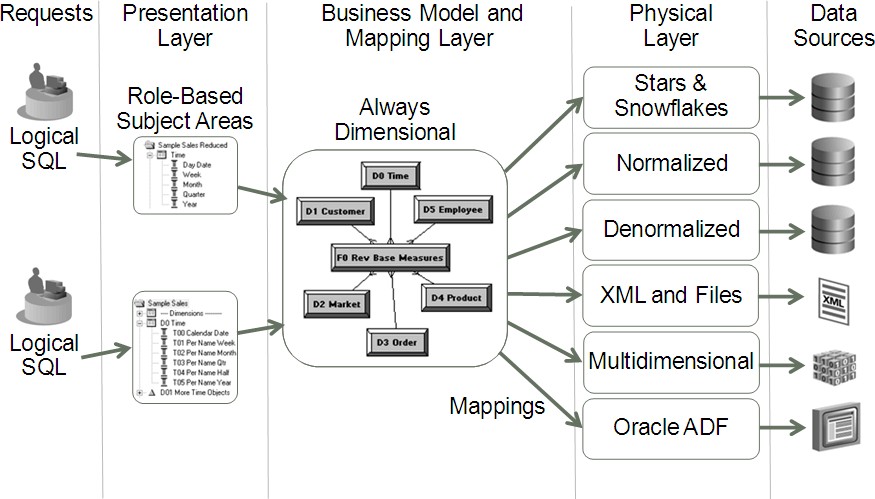
Depozitul de date BI contine urmatoarele layere:

* Layerul fizic. Acest layer contine definitiile obiectelor si relatiile necesare Serverului de BI pentru a scrie interogarile native asupra fiecarei surse fizice de date. Acest layer se poate crea importand tabele,cuburi de date, si fisiere flat. Separand comportamentul logic al aplicatiei de modelul fizic, se poate realiza asocierea mai multor surse fizice unui singur obiect logic, permitand navigarea agregata si partitionata, cat si izolarea dimensiunii , indiferent de modificarile aparute in sursa fizica de date. Aceasta separare creaza posibilitatea portarii aplicatiilor de Business Intelligence.
* Layerul de Modelare si Mapare Business contine definitii business sau modele logice de date si specifica maparea dintre modelul business si schema fizica. Acest layer determina comportamentul analitic valabil pentru utilizatori. Aici se ascund complexitatile modelului sursei de date.
* Layerul de Prezentare. Acest layer ofera o posibilitate de prezentare customizata, sigura, in functie de rolurile de vizualizare a modelului business a utilizatorilor. Astfel se adauga un nivel abstract layerului de Modelare si Mapare Business si ofera posibilitatea de vizualizare a datelor utilizatorilor construind cereri in cadrul Serviciului de Prezentare.

Inainte de construirea depozitului de date in cadrul toolului de administrare, este important sa se creze un design de nivel inalt in cadrul layerului de Modelare si Mapare Business bazat pe necesitatile analitice ale utilizatorului. Dupa realizarea designului la nivel conceptual, se pot crea apoi obiectele de metadata.

Ordinea tipica este de creare a obiectelor in cadrul layerului Fizic, apoi obiectele layerului de Modelare si Mapare Business, iar la sfarsit obiectele layerului de Prezentare. Dupa completarea celor trei layere, se seteaza securitatea obiectelor inainte de a incepe testarea depozitului de date business.

Figura 3.2 prezinta modalitatea in care interogarile SQL traverseaza layerele din cadrul depozitului de date business.



**Figure 3.2** *Modalitatea de parcurgere a layerelor de catre interogari SQL in cadrul unui*

*depozit de date BI*

# Planificarea unui Model Business

Planificarea unui model business este primul pas in dezvoltarea unui model de date pentru suportul decizional. Dupa crearea acestui plan decizional se poate trece la crearea depozitului de date.

# Analiza cerintelor unui model business

Primul task consta in intelegerea cerintelor unui model de business. Trebuie determinat ce model business trebuie construit inainte de analizarea cerintelor layerului fizic.

Intr-un mediu de suport decizional, obiectivul modelarii datelor consta in crearea unui concept capabil sa prezinte informatiile business intr-o maniera in care sa detaseze cunostintele analistilor business de structura business. Un model de succes contine procese naturale, permitand analistilor business sa structureze interogarile intr-un mod intuitiv, ca si cum ar formula intrebari business. Acest model trebuie sa fie inteles de analisti si sa prezinte raspunsul corect al interogarilor!

Pentru construirea acestui model business trebuie o structura pe diverse componente care sa raspunda la urmatoarele intrebari:

* La ce fel de intrebari incearca analistii business sa raspunda?
* Care sunt masurile necesare pentru atingerea performantelor business?
* Sub ce dimensiuni opereaza platforma business? In alte cuvinte, care ar fii dimensiunile necesare pentru a crea un raport?
* Exista elemente ierarhice in fiecare dimensiune, daca da, ce tip de relatie defineste fiecare ierarhie?

Raspunsul la aceste intrebari te ajuta in identificarea si definirea elementelor modelului business necesar.

# Identificarea Structurii Modelului Business

Pentru a crea un model business este necesar, in primul rand, maparea logica a datelor intr- un model business. Serverul de BI foloseste modele dimensionale in acest sens.

Aria business este analizata si impartita pe criterii, iar modelul business este dezvoltat in jurul acestor criterii. Aceste dimensiuni (criteriile) formeaza baza unui model valid. Datorita acestui fapt, sunt interpretate unele fapte in diferite modalitati ( dimensiuni, atribute).

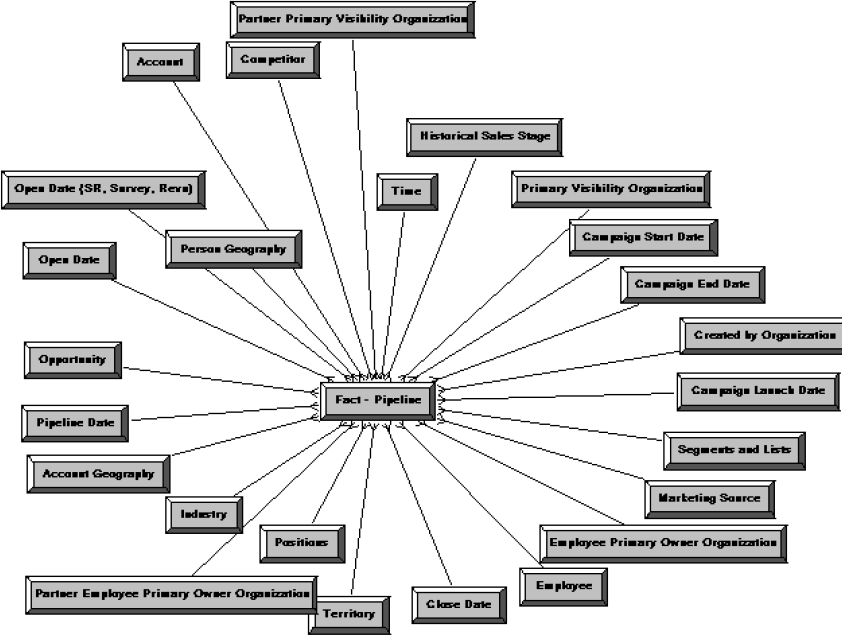
Dupa analizarea cerintelor modelului business, se specifica tabelele logice si ierarhiile necesare in cadrul modelului.

Un tabel faptic este un tabel compus din masuri. Aceste masuri se definesc intr-un tabel de fapte logic. Pentru masuri sau fapte, se calculeaza de obicei valoarea in dolari sau cantitatea vanduta pentru un produs, iar acest lucru trebuie specificat in termeni de dimensiuni. Spre exemplu, sa presupunem ca trebuie sa determinam suma de dolari pentru un produs de la un anumit super- market, pentru o anumita perioada.

Fiecare dimensiune are regula sa de agregare precum SUM,MIN, sau MAX. S-ar putea dorii o regula de comparare a valorilor si este necesara o astfel de formula. De asemenea, regulile de agregare pot fi specificate pe anumite dimensiuni. Serverul BI ofera posibilitatea crearii unor reguli complexe in functie de posibilitatea de agregare a unei dimensiuni.

Serverul BI recunoaste legatura unui tabel de tip many-to-one din layerul de Mapare si Modelare Business; este vorba de fapt de un tabel faptic.

Figura 3.3 prezinta legatura many-to-one a unui tabel faptic din cadrul unei digrame business.



**Figura 3.3** *Diagrama tabelelor de fapte*

O regula business foloseste fapte pentru a masura performanta prin intermediul unor dimensiuni precise, spre exemplu, prin timp, produs, piata de desfasurare. Fiecare dimensiune are un

set de atribute. Tabelele de dimensiuni contin atribute ce descriu entitatile business (precum Nume\_Client, Regiune, Adresa, Tara, etc). Tabelele de dimensiuni contin si o cheie primara pentru identificarea fiecarui membru.

Cea mai buna metoda de a identifica dimensiunile si atributele aferente este de a discuta cu analistul ce urmeaza a folosi pentru rapoartele respective. Este important ca terminologiile folosite de analisti sa fie foarte bine intelese. Atunci cand este nevoie de un model relational many-to-many intre tabelele dimensionale si cele faptice, se pot crea tabele de legatura.

Pentru a intelege ierarhia intr-un business este esentiala crearea de metadata ce permite Serverului BI sa determine daca o cerere particulara poate raspunde printr-o functie agregata ce este deja formulata.Spre exemplu, daca se ajunge la o luna dintr-un anumit an si exista un tabel agregat la nivel de luna, acel tabel poate fi folosit pentru a raspunde la intrebari referitoare la nivelul anului, adaugand toate datele lunilor din acel an.

O ierarhie este un set de relatii de tipul sus-jos dintre anumite atribute si o dimensiune. Aceste atribute ale ierarhiei, numite nivele, ajung de la nivel mic la un nivel mare. Aceste schimbari se intampla la nivelul elementelor din ierarhie si largesc aria relationala.

Dimensiunile sunt impartite in functie de atribute business definite. Dimensiunile comune sunt perioadele de timp, produsele, pietele, cumparatorii, conditiile de promotii, etc. Pentru o dimensiune pot exista mai multe atribute. Spre exemplu, pentru dimensiunea timp pot exista urmatoarele atribute: zi,saptamana, luna, semestru, an. Atributele se seteaza in functie de cerintele unui analist business. O dimensiune ierarhica prezinta o legatura one-to-many intre atribute.

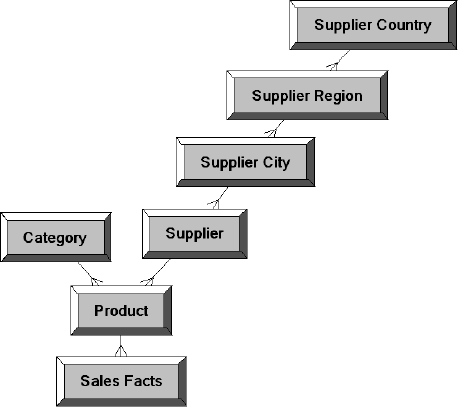
Definitiile unei ierarhii trebuie sa fie specificate in functie de modelul business. Serverul BI ofera posibilitatea crearii de ierarhii particulare, complexe, in functie de business. Trebuie identificate cat mai multe ierarhii naturale cu putinta. Asemenea intrebarilor business, unele ierarhii sunt evidente, dar unele nu sunt la fel de usor de identificat, si sunt stiute doar de utilizatorul ce interactioneaza cu ele. In primul rand este necesara intelegerea si definirea ierarhiilor sistemului inainte de a fi construite in toolul de administrare.

Unele relatii business au loc intre membrii diferiti aflati intr-un arbore ierarhic, precum managerul-angajatul.Aceasta ierarhie speciala este cunoscuta ca ierarhie parinte-copil, si nu prezinta nume speciale pe nivele. Nu exista limita a numarului implicit de nivele intr-o astfel de ierarhie.

Trebuie identificate cazurile in care relationarile membrilor de acelas nivel sunt importante pentru analistul business, astfel sunt setate dimensiunile corespunzatoare in ierarhiile parinte-copil.

Fiecare ierarhie intr-o dimensiune cu multiple ierarhii permite analiza pe diferite seturi de atribute si poate avea mai multe sau mai putine nivele. Figura 3.4 prezinta un exemplu unde dimensiunea Product contine urmatoarele doua ierarhie:

* Supplier Country - Supplier Region - Supplier City- Supplier - Product
* Category - Product



**Figura 3.4** *Exemplu de dimensiune multi-ierarhica*

# Identificarea continutului bazei de date pentru modelul business

Pentru a descrie în detaliu modelele de date multidimensionale (extensii ale modelului relaţional sau orientate pe cub) şi a putea fi înţelese, s-a considerat necesar a se prezenta o serie de concepte utilizate şi anume: *hypercubul (cub ndimensional), multicubul, dimensiunile*, *ierarhiile*, *măsurile* şi *fenomenul de împrăştiere*. Aceste concepte apar în majoritatea modelelor OLAP, deşi modul lor de definire diferă uneori foarte mult. De asemenea, consiliul OLAP a propus un glosar de termeni care se doreşte standardizat. De aceea, în prezentarea acestor concepte de bază s-au utilizat şi definiţiile date de Consiliul OLAP.

Serverul BI prevede o interfata ce permite maparea depozitului de date business intr-o schema de baze de date. Uneori se poate controla designul schemei fizice a bazei de date folosite. Dar uneori baza de date exista deja in sistem si trebuie lucrat cu ea exact in stadiul respectiv. In orice caz, este necesara intelegerea structurii si a continutului bazei de date.

Exista trei tipuri tipuri de scheme fizice (modele): scheme dimensionale, scheme normalizate si scheme complet denormalizate.

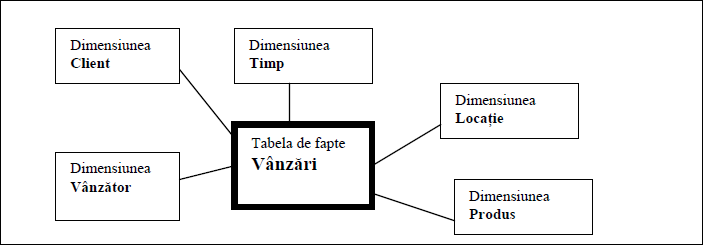
* **Scheme dimensionale.** Scheme dimensionale sunt create pentru a prelua continutul si pentru a optimiza interogarile de performanta. O schema dimensionala este o schema denormalizata ce urmareste un model business. Schemele dimensionale contin tabele dimensionale si tabele faptice. Dimensiunile tabelelor contin atribute ale businessului, iar tabelele faptice contin inregistrari individuale si chei straine pentru fiecare dimensiune din tabele. Schemele dimensionale sunt folosite special pentru analistii business si ofera doua avantaje asupra schemelor normalizate pentru suport decizonal:

1. Interogari mai bune pentru performanta
2. Mai usor de inteles

Schemele dimensionale nu sunt atat de eficiente ca schemele normalizate pentru updatarea unor inregistrari discrete, dar ele functioneaza foarte bine pentru interogarile ce analizeaza asupra mai multor dimensiuni.

**Schema stea**. Schema stea este o reprezentare intuitivă a cubului de date multidimensional într-un mediu relaţional. Schema *stea* conţine o tabelă centrală şi un set de *tabele de dimensiuni* aranjate într-o manieră radială, în jurul tabelei centrale. Fiecare *tabelă de dimensiuni* reprezintă o dimensiune a activităţii analizate, în timp ce tabela centrală conţine conţinutul cubului de date.

Schema este foarte asimetrică. Există o tabelă dominantă în centrul schemei, singura tabelă din schemă cu multiple joncţiuni, prin care se conectează la celelalte tabele. Această tabelă se numeşte *tabela de fapte* (fact table). Celelalte tabele au numai o singură joncţiune, prin care se leagă la tabela centrală şi se numesc *tabele de dimensiuni* sau *dimensiuni* (dimension table). Schema *stea* reduce numărul de joncţiuni între tabele, ca urmare a procesului de denormalizare. Nu există nici o informaţie despre nivelurile ierarhice stocate în structura schemei *stea*. În plus, reprezentarea unei dimensiuni printr-o singură tabelă conduce la date redundante, denormalizate şi inconsistente. În mediile cu depozite de date, aceasta nu cauzează nici o problemă, deoarece datele sunt în cele mai multe cazuri interogate. *Tabelă de fapte* are următoarele caracteristici:

* Conţine un număr foarte mare de tupluri (posibil milioane). Numărul de tupluri din tabelă reprezintă de fapt produsul cartezian al dimensiunilor. De exemplu, pentru analiza activităţii de desfacere a unei firme de comerţ cu 500 de magazine de distribuţie în întreaga lume şi care comercializează 50000 de produse în fiecare magazin, *tabela de fapte* va conţine 500\*50000\*2\*365 de tupluri. Baza de date va stoca informaţiile referitoare la tranzacţiile zilnice, pe o perioadă de doi ani.
* Dimensiunea ei creşte dinamic, în funcţie de cantitatea de date încărcate la fiecare ciclu de actualizare a bazei de date, precum şi în funcţie de volumul de date istorice stocate în baza de date.
* Este tabela care reflectă performanţa activităţii (procesului) analizate.
* Cheia primară a tabelei este o cheie compusă, formată din cheile primare ale tabelelor de dimensiuni. Fiecare *tabelă de fapte* are o cheie compusă şi invers fiecare tabelă care are o cheie compusă este o *tabelă de fapte*.
* Este normalizată şi realizează de fapt o legătură indirectă între dimensiuni.
* Figura 3.5 prezintă un exemplu de schemă *stea.* Tabela centrală este tabela Vânzări înconjurată de dimensiunile *Timp, Client, Locaţie, Produs, Vânzător***.** Deci schema *stea* are următoarele caracteristici: *tabela de fapte* se leagă de dimensiuni prin joncţiuni de egalitate;
* fiecare atribut din cheie primară a *tabelei de fapte* reprezintă cheia primara a unei dimensiuni;
* atributele non cheie pot fi agregate. *Tabelele de fapte* conţin numai atribute numerice;
* tabelele de dimensiuni sunt denormalizate.

**Figura 3.5** *Modelul stea*

Avantajele schemei stea:

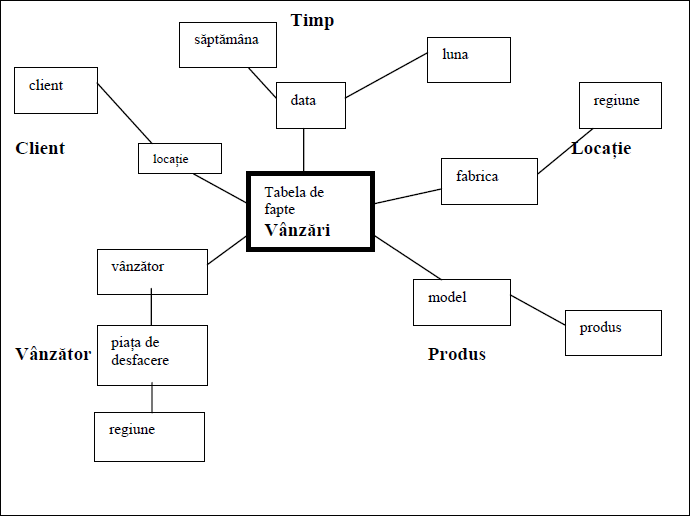
* este simplu de construit;
* reflectă cu exactitate modul cum înţeleg utilizatorii activitatea modelată;
* furnizează o performanţă ridicată pentru cereri analitice, prin reducerea numărului de joncţiuni;
* permite modelarea unor structuri multidimensionale complexe;
* este o schemă uşor de înţeles de către utilizatori, deoarece datele sunt aranjate într-un mod uşor de înţeles, iar relaţiile între entităţi sunt foarte

clare;

* toţi indicatorii de performanţă ai activităţii modelate sunt stocaţi în *tabela de fapte*.

Dezavantajele schemei stea:

* este o schemă inflexibilă. Adăugarea unei noi dimensiuni în schemă, poate duce la modificarea granulaţiei *tabelei de fapte;*
* conţine date redundante ce conduc la creşterea riscului de inconsistenţă a datelor;
* pentru a permite analize complexe, trebuie să includă multe tabele de agregate;
* are o scalabilitate limitată;
* face dificilă joncţiunea între *tabelele de fapte*.

Schema fulg de zapada.Schema *fulg de zăpadă* este o variantă a schemei *stea*. Este rezultatul descompunerii unei dimensiuni sau a mai multor dimensiuni care au ierarhii. Relaţiile de tip (m:1) între membrii unei dimensiuni se descompun în tabele separate formând o ierarhie de tabele. De exemplu, dimensiunea *Produs* este descompusă în subdimensiunile *Model* şi *Produs* (figura 3.6). Schema *fulg de zăpadă* vizualizează structura ierarhică a dimensiunilor. Diferenţa esenţială faţă de schema *stea* este că dimensiunile sunt normalizate. Principala motivaţie a normalizării este spaţiul de stocare. Dacă ierarhiile se păstrează în tabele separate, se consideră că spaţiul de stocare se reduce consistent. Totuşi Ralph Kimball în lucrarea sa “*Practical Techniques for Building Dimensional Data Warehouse*”(1996) încearcă să demonstreze contrariul. El afirmă că eforturile de a normaliza tabelele, în scopul de a reduce spaţiul de stocare, sunt inutile şi consumatoare de timp. Normalizarea dimensiunilor reduce cu mai puţin de 1% spaţiul total de stocare, iar vizualizarea datelor este mai dificilă, implicând un număr mai mare de joncţiuni între tabele. Mulţi proiectanţi nu consideră că salvarea spaţiului ar fi o cerinţă majoră în selectarea tehnicii de modelare.

**Figura 3.6** *Modelul fulg de zapada*

Avantajele schemei *fulg de zăpadă*:

* este mai flexibilă şi se pot defini cerinţele utilizatorilor mult mai bine. De exemplu, cu schema *stea* nu se pot folosi tabele istorice pentru modificarea dimensiunilor;
* structura normalizată a dimensiunilor este mai uşor de modificat;
* reduce redundanţa datelor;
* îmbunătăţeşte performanţa operaţiilor de *drill down* şi *roll up.*

Dezavantajele schemei *fulg de zăpadă*:

* este mai complexă decât schema *stea;*
* scade performanţa la interogare, deoarece sunt folosite multe joncţiuni;
* schema este mai dificil de înţeles de către utilizatori, fiind mai apropiată de modelul entitate-asociere.

-

Modelele *stea* şi *fulg de zăpadă,* precum şi variantele lor au devenit o reprezentare logică bine cunoscută a structurilor de date multidimensionale, în sistemele relaţionale. Selectarea unei scheme potrivite ţine cont de raportul între costul de stocare şi performanţa interogării. Schema *stea* oferă performanţe mai bune la interogare, datorită unui număr mic de operaţii de joncţiune între *tabela de fapte* şi *tabelele de dimensiuni,* dar costul de stocare este mai ridicat. Schema *fulg de zăpadă* implică mai multe operaţii de joncţiune, dar presupune o capacitate de stocare mai mică. Pentru ambele modele există o mulţime de variante. Schema *stea de zăpadă* (starflake) sau schema *fulg de zăpadă degenerată* (degenerated snowflake scheme) este o combinaţie de schemă *stea* şi schemă *fulg de zăpadă,* în care o parte a *tabelelor de dimensiuni* sunt denormalizate.

**Scheme normalizate**. Schemele normalizate ajuta la redundanta datelor stocate si optimizeaza updateul de date. Schema normalizata este schema clasica relationala folosita in multe sisteme de tranzactii online (OLTP - online transaction processing). Relatiile dintre tabele reprezinta relatiile intre date, nu neaparat intre relatiile de business.

Schemele normalizate in general nu functioneaza foarte bine cu interogarile referitoare la analiza istorica din cauza a doua probleme: performanta redusa si impodibilitate de transpunere a intrebarii in SQL.

* Problemele de performanta persista cu normalizarea interogarilor istorice deoarece aceste interogari presupun reluarea datelor si intregirea lor, iar acesta este un proces complex,incet.
* Un administrator de baze de date familiarizat cu datele, poate scrie o interogare SQL pe fundalul unei scheme normalizate care teoretic poate raspunde unei intrebari business, dar nu poate oferii o analiza detaliata. Chiar daca interogarea SQL este detaliata,complexa, apare deseori refuzarea raspunsului in schimbul returnarii rezultatului.

**Scheme complet denormalizate.** Acest tip de scheme combina faptele si dimensiunile intr-un singur tabel (sau fisier flat) si se mapeaza diferit fata de celelelate scheme. Maparea unei scheme complet denormalizate intr-o schema stea presupune maparea coloanelor faptice in tabele logice si maparea coloanelor de dimensiune in tabele logice de dimensiuni.

# Sursa de date multidimensionala

Cuburile sunt formate din masuri si organizate in dimensiuni. Deoarece ele sunt deja dimensionate, fiecare cub mapeaza faptele logice si tabelele de dimensiuni in modelul business.

* Masurile in cuburile multidimensionale si relatiile cu coloanele faptice, amandoua sunt mapate in masuri logice in layerul de mapare si modelare business. Masuratorile din cadrul cuburilor multidimensionale contin deja formule si agregari, spre deosebire de relatiile coloanelor faptice ce necesita implementarea unor formule si agregari pentru a fi aplicate in modelul business. Mai exact decat prelucrarea cuburilor, Serverul BI prezinta avantajul de a putea folosi date deja agregate si poate realiza calcule extrem de puternice in cadrul cubului.
* Obiectele fizice multidimensionale si relatiile obiectelor fizice sunt ambele mapate in dimensiuni logice in layerul de mapare si modelare business. Semanticile dimensionale si ierarhice sunt deja construite in sursele de date multidimensionale, spre deosebire de sursele relationale. Serverul BI prezinta avantajul detinerii unei ierarhii complete si suport decizional in cadrul cubului, atat in timpul importului cat si in timpul cererilor.

# 3.7 Unelte BI specializate

Numim aceste unelte ca « specializate » deoarece acestea sunt disponibile numai împreuna cu baza de date OLAP. Acestea nu sunt uneltele de raportare generale. Ele sunt numite unelte deoarece reprezintă încă o piesa în trusa de unelte pentru a rezolva problemele de afaceri întâlnite in zilele noastre. Uneltele specializate BI nu reprezintă un scop in sine, ci sunt doar nişte mijloace in plus pentru a aduna informaţia. Un rol important al acestor unelte este mecanismul de livrare. Exista doua mecanisme de livrare oferite de către furnizorii de unelte BI :

1 Aplicaţii pentru Desktop 2 Livrări (pe) web

În mod specific, aplicaţiile pentru Desktop, oferă o interfaţa bogata clientului. Aici un număr mare de caracteristici sunt supuse analizei, informării, explorării, editării şi a răspândirii informaţiei. Acesta este un set ce caracteristici foarte important pentru analişti si pentru editorii de informaţii. Ei sunt persoanele care pregătesc/ modifica/ aranjează informaţia pentru nu număr mare de oameni/ o audienta mare ce nu au/are timp sau nevoie sa facă o analiza.

Livrări( pe) web reprezintă informaţia ce este disponibila într-un browser web al unui website. Aceasta nu înseamnă internet, si nu este in mod special internet, ci izolată în reţeaua internă a unei companii VPN( virtual private network), RPV(reţea privata virtuala) . Informaţia expusa pentru uneltele BI si pentru tehnologia OLAP este precisa de obicei şi nu este disponibila pe world wide web. Cu toate acestea exista organizaţii/asociaţii ce oferă informaţia clienţilor prin internet si altele(organizaţii) care oferă persoanelor din companie informaţia cu ajutorul unor site-uri securizate. Livrarea pe web este un mecanism cheie pentru aceste asociaţii. Partea nefavorabila a livrării pe web o reprezintă faptul ca interfaţa si experienţa utilizatorului nu sunt la fel de bogate precum aplicaţiile pentru Desktop. De fapt unele livrări web servesc numai ca pagini cu un format static în timp ce altele au un nivel înalt de interactivitate. In ceea ce priveşte uneltelesi progresul tehnologiei, livrarea pe web va intra in norme/ vor deveni norme cu ajutorul funcţionalităţii disponibile în aplicaţiile pe desktop mutându-se la livrarea informaţiilor pe web.

Aplicaţiile pe desktop şi livrarile web nu sunt aceleaşi mecanisme elementare ca acum cinci ani,si probabil nici nu vor mai fi acelasi nici de acum in cinci ani. Spaţiul de unelte BI evoluează şi exploatează schimbarea rapidă a tehnologiei

# Caracteristici şi Funcţionalitate : O perspectiva tehnica

Caracteristicile cele mai importante ale uneltelor BI, menţionate în capitolul 3 ar trebuie examinate îndeaproape în ceea ce priveşte semnificaţia cuvântului « feature » si bineînţeles din perspectiva consumatorului. Software-ul BI devine din ce in ce mai puternic si multe din soluţiile cunoscute împart o funcţionalitate comuna. Lista ce urmează explica câteva din caracteristicile cheie.

***Drill-down*** (de la general la detaliat)

Aceasta este o acţiune ce permite utilizatorilor să navigheze în jos în cadrul unei ierarhii dimensionale. Ierarhiile sunt adaptarea structurala a unei dimensiuni. De exemplu , timpul poate avea o structura ierarhica a anului, trimestru,luna si zi. Daca ne uitam la o data din anul 1998 şi mergem, de la o informaţie generală ajungem la una detaliata, adică ajungem sa vedem trimestrele anului 1998. Trecerea de la general la detaliat ne permite sa observam ce tipuri de informaţie aduce o singura colecţie de valori.

Diagrame şi grafice standard. Sprijinirea diagramelor standard ale afacerilor reprezintă un rol important în livrarea informaţiei în BI. Enunţul, « O fotografie valorează o mie de cuvinte » conferă si mai multa greutate informaţiilor OLAP. O unealta care sprijină diagramele standard ale afacerilor (ex. bara, bara orizontala, domeniu, aliniere,punct, etc.) este mai valoroasa decât o unealta care pune numere într-o grila, forţând consumatorul de informaţie să sorteze printr-un tabel de numere pentru a accesa informaţia.

## Vizualizare avansată

Aceasta caracteristica este definita ca avansata de către furnizor. Ea se refera in special la o parte a uneltei care are un brevet/privilegiu/certificat al furnizorului si îşi fac reclama căci este o cale mai buna de a primi informaţia. De exemplu, Copacul de descompunere de la corporaţia ProClarity

unde consumatorului de informaţie ii este permis să caute de la general la detaliat(drill down) în orice dimensiune dintr-o măsurare tipizata(data point) din structura de copac.

## Evidenţierea Excepţiei

Aceasta este numita câteodată şi « alerte », prin intermediul cărora consumatorilor de informaţii le este permis sa-si specifice parametrii pentru valorile evidenţiate şi de asemeni dispunerea acelor valori. Aceasta caracteristica este folositoare pentru că atunci când ne uităm la un tabel mare cu numere si valori intr-un fel ar trebui sa iasă în evidenţă. Un exemplu , ar putea fi colorarea tuturor profiturilor mai mici de $100.000 cu roşu ca o dovada a scăderii sub $100.000 este un semn nesănătos. Acesta este acelaşi concept precum caracterul de editare condiţional pe care îl întâlnim în Excel.

## Proiectii combo

Acestea se refera la capacitatea de a face o combinaţie grafică în care o diagrama a liniei şi a barei poate fi pe acelaşi grafic şi are capacitatea de a arata doua axe Y diferite, cu scări de valori diferite. Unul din cele mai cunoscute exemple este o diagrama care arata o valoarea absolută pentru profit şi una valorile procentuale pentru creşterea profitului. Valorile pentru profit şi pentru procentaj, nu vor balansa dacă sunt arătate pe aceeaşi axa deoarece profitul va opri creşterea valorilor procentuale. Soluţia este de a plasa profitul de o parte a diagramei(pe axa unu y) şi creşterea procentajului în partea cealaltă a axei(axa y doi). Apoi aplicaţi o diagrama diferita pentru fiecare valoare o linie grafica pentru creştere şi o bara grafica pentru profit.

## Pivotarea rândurilor si a coloanelor.

Aceasta este o caracteristica fundamentala pentru majoritatea uneltelor BI. Aceasta permite ca valorile prezente pe rânduri şi coloane să se schimbe. Valorile de pe rânduri pot fi mutate pe coloane şi invers, valorile de pe coloane sunt mutate pe rânduri.

***Dimensiuni drag si drop*** (trage şi lasă/ia/trage şi muta)

Acesta este un element important pentru a simplifica interacţiunea unui consumator de informaţie cu unealta BI. Acţiunea aşteptata de către consumatorul de informaţie este de a da un click pe un item si apoi sa-l mute intr-ul loc dorit. Ce uşor ar fi sa iei dimensiunea dorita şi să o muţi în poziţia dorita pe rânduri,coloane şi să le aranjezi ordinea, totul numai cu ajutorul unui click şi prin tragerea mouse-ului. Nu este necesar să forţezi consumatorul de informaţie să deschidă o fereastra nouă şi să specifice poziţionarea dimensionala utilizând alte tehnici .

## Calcularea taxelor dezvoltate de către utilizator/ user-defined.

Aceasta înseamnă sa plasezi abilitatea de a crea metrice importante pentru consumatorii de informaţii. Oricine a utilizat Excel-ul si a creat formule înţelege puterea acestei caracteristici. Persoana creează metrice provenite de la alte valori ce oferă mai multe informaţii. Aceasta este o caracteristica puternică a uneltei BI , de a plasa aceasta abilitate în mâinile consumatorilor de informaţii. Din alt punct de vedere, să creezi metrice ce nu se găsesc în mod normal în setul de informaţii va reprezenta o îndatorire administrativa a directorilor/responsabililor cu structura OLAP.

## Expunerea chestionarelor

Aceasta se refera la capacitatea de a modifica de mână limbajul sau script-ul ce defineşte un chestionar într-o unealta BI. Unele unelte expun aceasta şi folosirea sa va fi făcuta de către utilizatorii ce au nevoie de un şiretlic, de un chestionar, deja existent pentru a obţine cu exactitate ceea ce vor si nu pot de la utilizatorul interfeţei acelei unelte. Astfel de exemple ar fi script-urile descrise de către Analizatorul de vânzări Oracle (oracle sales analyzer) sau de către MDX expus de către Platforma ProClarity, de la corporaţia ProClarity.

## Comentarii calitative.

Augmentarea unei informaţii cu comentarii calitative poate conferi credibilitatea necesara pentru valorile informaţionale reale. Gândiţi-vă la asta ca la o raportare a unui analist/specialist. Uitându-ne la numere este un început bun, dar daca un specialist poate adăuga cuvinte explicative pentru acele cifre, valoarea informaţiei creste. Comentariile calitative ar trebui introduse de către utilizatori ca o privire asupra informaţiei/analiza asupra informaţiei pe care o construiesc iar

comentariile să fie salvate ca parte a acelei analize. Obiectivul principal este de a răspândi/distribui/împărţi comentariile şi altor persoane interesate de aceeaşi informaţie.

# Dashboard (Tablou de bord).

Acest termen este folosit pentru a defini unele metrici corespunzătoare sănătăţii afacerii, consumatorului informaţional care monitorizează aceste metrici, asigurându-i-se accesul într-un mod facil. Un exemplu de astfel de locaţie, accesibilă, îl poate reprezenta Microsoft Outlook, indivizii având acces la inbox, pentru a-şi vizualiza mesajele, calendar, si lista de sarcini. O astfel de locaţie ar fi ideală pentru plasarea metricilor cu importanţă în cadrul unui tablou de bord. Alte exemple, implică un site web intern, pentru o organizaţie care se comportă ca şi “portal” informaţional. Plasarea unui tablou de bord în această locaţie, îl face accesibil în cadrul organizaţional, prin intermediul unui browser Internet. Implementarea unui tablou de bord, poate diferi, de la un grid cu câteva numere pentru câteva metrici cheie, până la un grafic de tip vitezometru, care se schimbă rapid, sub influenţa schimbărilor informaţionale. Alegerea tipului de implementare depinde de nevoile consumatorului informaţional, şi de ceea ce poate fi produs de instrumentele BI.

# Distribuţia cuburilor/rapoartelor.

Aceasta este critica pentru distribuţia informaţiei. Cei doi termeni, cuburi şi rapoarte, sunt corelate, deoarece prin intermediul acestei combinaţii, consumatorul informaţional poate face mai mult decât a vedea un raport static. Majoritatea dintre noi suntem obişnuiţi să vedem o imagine statică a informaţiilor conţinute într-un raport. Raportul este separat de setul de date. Pentru a obţine mai multe informaţii, trebuie accesat un alt raport. Acest ciclu continuă, până când obţinem informaţiile dorite. În cazul unui raport legat de un cub care derivă informaţia. Pentru a obţine mai multe detalii consumatorul informaţional trebuie să folosească tehnica de “drill-down”, în cadrul cubului informaţional – în cadrul aceluiaşi raport. Beneficiile unei astfel de soluţii le reprezintă economisirea timpului, relevanţă pentru consumatorul informaţional, iar informaţia este derivată mai rapid.

# Filtrarea şi sortarea

Filtrarea datelor reprezintă o metodă uşoară pentru a limita înregistrările vizualizate la un moment dat. Această metodă este folosită în principal pentru a elimina aria superioară, inferioară, sau o gamă specificată de valori. Acest lucru facilitează eliminarea, de exemplu, celor 50 de clienţi ce contează pentru 2% din afacere, şi concentrarea pe cei 20 de clienţi ce contează pentru 98% din afacere. Prin filtrarea datelor, poţi lucra cu un set mai mic de membri şi reduce timpul necesar unei interogări pentru a returna un răspuns. Sortarea este deseori necesară pentru a analiza datele sortate după metricile observate. Dacă se vizualizează vânzările, o persoană poate dori să analizeze termenii sortaţi în ordinea descrescătoare a vânzărilor.

***Analize în timp real: mit sau realitate*** (Perspective tehnologice)

Analiza în timp “real” de referă la o implementare ROLAP în cadrul căreia cubul de date vine direct dintr-un sistem tranzacţional. Datele încorporate în definiţia ROLAP reprezintă acelaşi set de date reprezentat de sistemul OLTP(online transaction processing). Aceasta reprezintă o contradicţie faţă de implementarea tradiţională OLAP, în cadrul căreia informaţiile sunt filtrate şi planificate, anterior incluziunii în cub.

Beneficiul important al utilizării implementării OLAP în timp real, este reprezentat de faptul că informaţia este la fel de nouă ca şi cea din cadrul sistemului OLTP. Nu există nici o întârziere între momentul în care data a fost introdusă în sistemul OLTP şi momentul din care este disponibilă din cadrul cuburilor OLAP. Cu acest beneficiu, de ce fiecare implementare OLAP nu este una în timp real? Sunt două motive majore de ce OLAP-ul în timp real nu este implementat în mod tipic:

1. Acurateţea informaţiilor. Credibilitatea informaţiilor din cadrul unui cub OLAP este la fel de solidă ca procesele de filtrare şi planificare, care apar în mod normal în cadrul extracţiei, transformării, şi încărcării(ETL). Există un motiv bun pentru care acele faze sunt denumire filtrare şi planificare. Un procentaj din datele introduse sa menţinute în cadrul unui sistem OLAP sunt inexacte, nu sunt relaţionate corect cu celelalte tabele, sau pur şi simplu incorecte. De aceea o implementare tipică OLAP îşi va lua datele dintr-un depozit de date, sau o piaţă de date, unde datele au fost supuse

proceselor de filtrare şi planificare, furnizând astfel credibilitate. Când o implementare OLAP în timp real extrage direct din tabelele OLTP, nu are loc nici un proces ETL pentru a filtra şi planifica datele. Nu a avut loc nici un audit sau verificare a acurateţei, de aceea exactitatea informaţiei ar trebui chestionată.

1. Performanţă. Un aspect important al implementării OLAP este de a furniza un acces rapid la informaţie. Acest lucru este realizat prin agregarea informaţiei. Agregările sunt centralizatoare precalculate ale datelor, care îmbunătăţesc răspunsul interogărilor prin a avea răspunsurile pregătite înaintea formulării. Cu o implementare OLAP în timp real, agregările sunt eliminate. Informaţia cubului este extrasă direct din tabelele care compun sistemul OLTP. Agregarea datelor în cadrul acelor tabele, pur şi simplu, nu este realizată. Performanţa interogărilor OLAP va suferi. Gradul de afectare a performanţei este influenţat de mai mulţi factori:

* Mărimea tabelelor OLTP. Dacă tabelele sunt mari, şi implementarea OLAP cere centralizarea datelor de peste timp, de exemplu pe trimestre financiare, atunci putem ca centralizarea acelor date să consume mult timp. Această întârziere în returnarea rezultatelor reprezintă un efect direct al lipsei agregării în cadrul implementării OLAP.
* Numărul de utilizatori concurenţi. Nu reprezintă doar numărul utilizatorilor concurenţi OLAP, cu şi cel al utilizatorilor care introduc, modifică, şi administrează datele OLTP. Aceştia includ persoanele care rulează rapoarte şi care adaugă, modifică, sau şterg înregistrări existente pentru ca informaţia sa fie actualizată.
* Scalabilitatea sistemului care administrează tabelele OLTP. Sistemul care administrează tabelele OLTP trebuie să fie suficient de scalabil pentru a face faţă acestui nivel de utilizatori concurenţi. Utilizatorii OLAP nu cer pur şi simplu o înregistrare, îi schimbă conţinutul şi o introduc înapoi în sistem (o acţiune tipică OLTP), dar emit înregistrări către sistemul OLTP care va strânge cantităţi mari de date, le centralizează, şi prezintă rezultatele înapoi sistemului OLAP. Această acţiune întreprinsă de un număr de utilizatori concurenţi, va taxa scalabilitatea sistemului.

Implementările OLAP în timp real nu sunt tipice, în primul rând pentru motive de acurateţe a informaţiei şi performanţă, listate mai sus. Aceste restricţii sunt destul de prohibitive încât neagă beneficiile originale pentru care sistemul OLAP a fost abordat. Realitatea este că sistemele OLAP în timp real nu sunt implementate aproape niciodată. Totuşi, în viitor sistemele OLAP în timp real ar putea deveni instrumente şi tehnologii comune.

***Add-ins-uri Excel***

Dintr-un alt punct de vedere, există şi un alt set de instrumente folosite la extragerea informaţiei din sursele de date OLAP şi plasarea lor intr-o foaie de lucru Excel. Acestea reprezintă add-ins-uri Excel, care sunt proiectate pentru funcţionarea în cadrul Excel şi nu ca funcţii cu soluţii de sine stătătoare, sau integrate într-un alt mediu precum un site web sau o aplicaţie realizate de utilizator.

Transferul informaţiei în Excel dintr-o sursă de date OLAP reprezintă un avantaj puternic, în special pentru consumatorii de informaţii, care efectuează deja analize în Excel. Beneficiul real al unui add-in Excel care permite accesul la datele OLAP, îl reprezintă crearea unei legături între Excel şi datele OLAP. Este mai mult decât o simplă extragere a datelor dintr-o sursă OLAP şi plasarea unui set static de valori într-o foaie de lucru. Astfel, apare o “legătură” între Excel şi datele OLAP, astfel încât datele OLAP sunt actualizate în cadrul foii de lucru. Actualizarea are loc la cerinţele consumatorului informaţional, însă ei trebuie să definească legătura o singură dată, iar apoi să reîmprospăteze datele când doresc, de exemplu când sursa datelor OLAP este actualizată.

Un alt beneficiu cheie de a aduce datele OLAP în Excel este de a profita de capacităţile de formatare. Multe organizaţii folosesc Excel ca un instrument de raportare, deoarece pot formata informaţia astfel încât să întâlnească în totalitate nevoile lor de raportare. Unele agende Excel implică mult timp pentru crearea şabloanelor complexe, care formatează şi prezintă informaţia. Aducerea datelor OLAP în cadrul Excel ajută la protejarea investiţiei.

## Metodele tradiţionale de raportare versus metode de analiză OLAP

Există multe discuţii în lumea BI din zilele noastre, conform cărora rapoartele vor dispărea pur şi simplu, odată ce informaţia este disponibilă online. Rapoartele continuă să joace un rol important în cadrul afacerilor actuale, chiar dacă există o deplasare către analiză online. Rapoartele afacerilor standard, privitoare la balanţă, sau la pierdere şi profit furnizează, multor analişti, o vizualizare sumară a ceea ce au nevoie să înţeleagă sănătatea financiară a afacerii.

Totuşi, există un alt grup de rapoarte care sunt, în mod continuu, îmbunătăţite şi customizate, pentru a furniza celor ce iau deciziile informaţii necesare pentru a lua decizii optime. Obiectivul acestor rapoarte este de a furniza o granulaţie mai fină a informaţiei pentru a identifica trend-urile, arăta relaţiile, sau instanţele de tip cauză-efect în cadrul informaţiei. Acest grup de rapoarte este destinat învechirii, încât uneltele BI devin mai comune şi plasează această ”explorare” a informaţiei în mâinile celui care ia deciziile.

Analiza online depaseste acest ciclu ce ia timp plasand responsabilul cu deciziile online cu informatia. Cum intrebarile sunt puse responsabilii cu deciziile pot explora lucrurile comanadand informatii rezumate. Ei pot identifica orientari/directii , explora realtii so sa observe instantele cauza- efect in informare. Rapoartele ofera functionalitatea necesara de a rezuma dtaliile si instantanelle informatiilor. Analiza online pune responsabilii cu deciziile in legatura cu informatiile pe care le pot explora si edita in folosul lor, eliminand ciclul de risipa de timp la cererea rapoartelor.

Uneltele de scriere a rapoartelor sunt axate pe o capcaitate de editare ridicata. Cei ce determina rapoartele pot specifica elementele ca titlurile, configuratia coloanelor, latimea coloanelor, impartirile sumei partiale, fonturile(fonts), marimea fonturilor, si o lista lunga de configuratii/ caracteristici de continut. Functionalitatile principale ale unei scrieri bune de rapoarte sunt.

*Flexibilitatea in configuraţia raportului.*

Flexibilitatea se traduce ca adaptare la cerinţele personale. Poate formatul raportului să fie adaptat la cerinţele cerute de nevoile afacerii ? Majoritatea uneltelor de scriere a rapoartelor sunt uşor de modificat pentru capacitatea de editare. Aceasta funcţionalitate este « biletul de intrare » , daca o unealta de scriere a raportului nu este destul de uşor de adaptat, atunci un evaluator de produs nici măcar nu se va uita la ea.

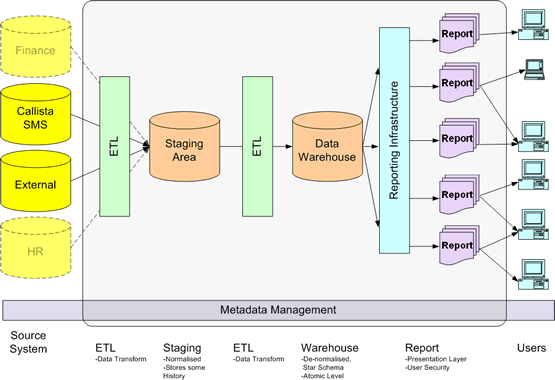
*Adaptabil pentru a îndeplini nevoile afacerii.*

Aceasta reprezintă mai mult decât flexibilitatea în configuraţia raportului. O enunţare a profitului si a pierderilor pentru o afacere. « A » va fi diferit de cel al afacerii « B ». O unealta bună de scriere a rapoartelor va permite atât afacerii să adapteze profitul şi pierderile pentru a reflecta nevoile afacerii. O enunţare a profitului şi a pierderilor este doar un exemplu. Capacitatea de a modifica, ar trebuie să acopere un număr mare de rapoarte în unealta de scriere a rapoartelor.

*Automatizarea*

Ultima caracteristica de automatizare a unei unelte de scriere a rapoartelor este dubla ( 1) de a genera raportul si (2) de a-l distribui. Distribuţia automatizată are loc atunci când nişte unelte de scriere a rapoartelor lipsesc. În acest punct, câteva dezvoltări de bază sau alte unelte trebuiesc folosite pentru distribuţie. O unealtă să poată efectua o generare, o distribuţie automată ce este uşor menţinută şi poate fi foarte valoroasă.

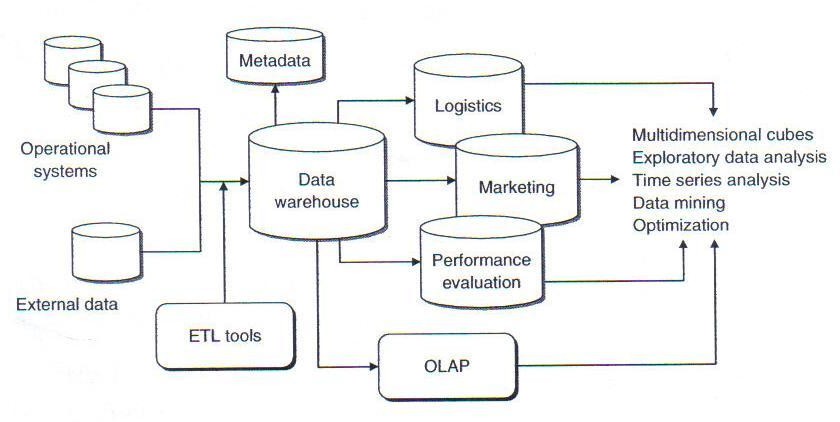
# Depozite de date (Data warehouse)



# Arhitectura depozitului de date

Arhitectura depozitului de date, arătată în figura 4.1, include următoarele componente funcţionale majore.

* Însuşi depozitul de date împreună cu datele marts adiţionale, care conţin date şi funcţii ce permit ca datele să fie accesate, vizualizate şi poate modificate.



**Figura 4.1** *Arhitectura şi funcţiile unui depozit de date*

* Aplicaţiile tip achiziţionare date, de asemenea cunoscute ca extragere, transformare şi încărcare unelte back-end, care permit ca datele să fie extrase, transformate şi încărcate în depozitul de date.
* Business Intelligence şi aplicaţiile pentru sprijinirea deciziilor reprezintă front-end-ul şi permit celor care lucrează pe aplicaţie să analizeze şi să vizualizeze rezultatele.

Distincţia celor trei nivele se aplică la arhitectura prezentată în Figura 4.1 chiar şi dintr-o perspectivă tehnologică.

* Nivelul surselor de date şi instrumentelor legate de ETL sunt de obicei instalate pe una sau mai multe servere.
* Nivelul depozitelor de date şi a oricăror date mart, eventual disponibile pe una sau mai multe servere, de asemenea, şi separate de cele care conţin surse de date. Acest al doilea nivel, de asemenea, include originea documentării metadatelor şi sensul înregistrărilor stocate în depozitul de date.
* Nivelul de analize care creşte valoarea a informaţiilor conţinute într-un depozit de date prin interogare, raportare şi, posibil, instrumente sofisticate de suport al deciziei. Cererile de informaţii de afaceri şi de decizie de analiză de sprijin sunt de obicei găsite pe servere separate, sau direct de pe PC-ul client utilizate de către analişti şi lucrători de cunoştinţe.

Aceeaşi baza de date a platformelor sistemului de management utilizate pentru a dezvolta sisteme tranzacţionale sunt, de asemenea, adoptate pentru implementarea depozitelor de date şi datelor mart. Datorită cerinţelor ridicate de răspunsuri la întrebările complexe adresate depozitului de date, sistemul de management al bazelor de date utilizate pentru stocarea de date sunt supuse la diferite structurări şi parametrizări cu privire la sistemele de tranzacţionare.

Un depozit de date poate fi implementat în conformitate cu diferite abordări privind designul: de sus în jos, de jos în sus şi mixt.

**De sus în jos (Top-down).** Metodologia Top-down se bazează pe proiectarea de ansamblu a depozitului de date, şi este, prin urmare, mai sistematică. Cu toate acestea, ea implică o dezvoltare mai lunga în timp şi riscuri mai mari de a nu fi finalizată pe întregul curs de dezvoltare şi în cadrul programului de stocare a datelor.

**De jos în sus (Bottom-up).** Metoda Bottom-up se bazează pe utilizarea de prototipuri şi, prin urmare, extensiile de sistem sunt realizate pas cu pas în conformitate cu o schemă. Această abordare este de obicei mai rapidă, oferă mai multe rezultate tangibile, dar îi lipseşte o viziune de ansamblu a întregului sistem care urmează să fie dezvoltat.

**Mixte.** Metodologia mixtă se bazează pe proiectarea de ansamblu a depozitului de date, apoi a veniturilor cu abordare de prototipuri, prin implementarea secvenţială a diferitelor părţi ale întregului sistem. Această abordare este foarte practică şi de obicei, de preferat, deoarece permite micşorarea şi controlarea paşilor care urmează să fie realizaţi având în vedere imaginea de ansamblu.

Paşii pentru dezvoltarea unui depozit de date sau a unui data mart pot fi rezumaţi după cum urmează.

* Unul sau mai multe procese din cadrul organizaţiei, care urmează să fie reprezentate în depozitul de date sunt identificate, cum ar fi vânzări, logistică sau contabilitate.
* Granularitatea, pentru a reprezenta procesele selectate, este identificată şi nivelul atomic al datelor este definit.
* Măsurile relevante care urmează să fie exprimate de fapt în tabelele pentru analiza multidimensională sunt apoi alese, aşa cum este descris în secţiunea 3.3.
* În cele din urmă, dimensiunile din tabele sunt determinate.

# Instrumente ETL

ETL se referă la instrumentele software care sunt dedicate să efectueze într-un mod automat trei funcţii principale: de extracţie, transformare şi încărcare a datelor în depozitul de date.

**Extracţia.** În timpul primei faze, datele sunt extrase din sursele disponibile interne şi externe. O distincţie logică poate fi făcută între extracţia iniţială, în cazul în care datele disponibile în raport cu toate perioadele din trecut, sunt încorporate în depozitul de date gol, şi extracţii ulterioare elementare, care actualizează depozitul de date utilizând date noi, şi devin valabile de-a lungul timpului. Selecţia datelor care urmează să fie importate, se bazează pe proiectarea depozitelor de date, care, la rândul lor, depind de informaţiile necesare pentru analiza de business intelligence şi de sistemele suport decizie de operare într-un domeniu specific al aplicaţiei.

**Transformarea.** Scopul de curăţare şi faza de transformare este de a îmbunătăţi calitatea datelor extrase din diferite surse, prin corectarea unor neconcordanţe, inexactităţi şi valori lipsă. Unele din deficienţele majore, care sunt eliminate în cursul etapei de curăţare a datelor sunt:

* neconcordanţele între valorile înregistrate în diferitele atribute să aibă acelaşi înţeles
* duplicarea de date;
* datele care lipsesc;
* existenţa unor valori inadmisibile.

În timpul fazei de curăţare, sunt aplicate reguli automate presetate pentru a corecta greşelile cele mai recurente. În multe cazuri, sunt folosite dicţionare cu termeni valabili pentru a înlocui presupuşii termeni incorecţi, bazate pe nivelul de similitudine. Mai mult decât atât, în timpul fazei de transformare, apar date de conversie suplimentare în scopul de a garanta omogenitatea şi integrarea cu privire la diferite surse de date. Mai mult, datele de agregare şi de consolidare sunt realizate în scopul de a obţine rezumate, care vor reduce timpul de răspuns cerut de întrebări ulterioare şi analize, pentru care este destinat depozitul de date.

**Încărcarea.** În cele din urmă, după ce au fost extrase şi transformate, datele sunt încărcate în tabelele de stocare date pentru a le face disponibile analiştilor şi aplicaţiilor de suport decizii.

# Metadate

Cu scopul de a da sens datelor conţinute într-un depozit de date, se recomandă să se înfiinţeze o structură specifică de informaţii, ştiute ca metadate, sau date care descriu date. Metadatele indică, pentru fiecare atribut dintr-un depozit de date, sursa originală a datelor, semnificaţia lor şi transformările la care acestea au fost supuse. Documentaţia furnizată de metadate ar trebui să fie permanent actualizată, pentru a reflecta orice modificare în structura depozitului de date. Documentaţia ar trebui să fie direct accesibilă pentru utilizatorii depozitelor de date, în mod ideal, printr-un browser de web, în funcţie de drepturile de acces referitoare la rolul fiecărui analist.

În special, metadatele trebuie să desfăşoare următoarele sarcini informative:

* o documentare a structurii depozitului de date: aspecte (layout-uri), vizualizări logice, dimensiuni, ierarhii, derivate de date, localizarea oricăror date Mart;
* o documentare de genealogie de date, obţinute prin etichetarea surselor de date de la care au fost extrase şi prin descrierea oricăror transformări efectuate pe datele propriu-zise;
* păstrarea unei liste a statisticilor utilizate a depozitului de date, indicând cât de multe accesări au fost efectuate la un câmp sau la o vizualizare logică;
* o documentare în sensul general al depozitului de date cu privire la domeniul de aplicare, prin furnizarea definiţiei termenilor utilizaţi, şi descrierea în totalitate a proprietăţilor de date, a dreptului de proprietate asupra datelor şi a politicilor de încărcare.

# Cuburile şi analiza multidimensională

Conceperea şi modelarea depozitelor de date şi a centrelor de date este bazată pe o abordare multidimensională pentru reprezentarea datelor, abordare care oferă cel puţin două avantaje: din punct de vedere funcţional garantează un timp de răspuns mic chiar şi la interogări complexe, iar din punct de vedere logic - dimensiunile se potrivesc natural la criteriile urmate de knowledge workers pentru a realiza analiza.

Reprezentarea multidimensională este bazată pe o schema sub formă de stea care conţine două tipuri de tabele de date: tabele de dimensiuni şi tabele de fapte (tranzacţii).

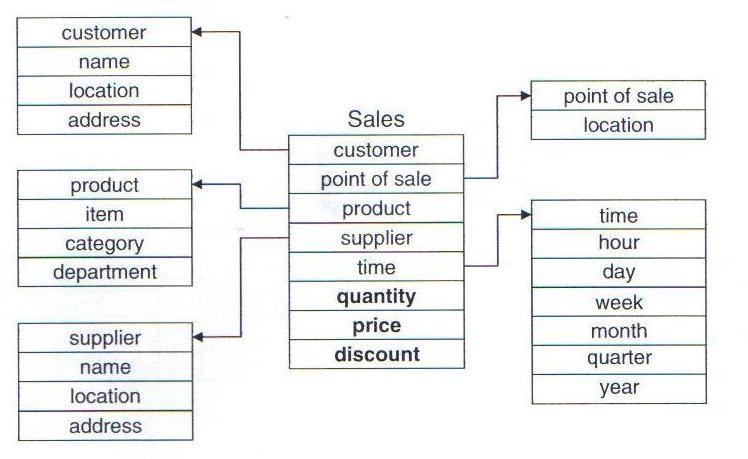
**Tabelele de dimensiuni.** În general, *dimensiunile* sunt asociate entităţilor în jurul cărora procesele din cadrul unei organizaţii se desfăşoară. Astfel tabelele de dimensiuni corespund entităţilor primare din cadrul depozitelor de date şi în cele mai multe cazuri derivă din tabelele master (tabelele principale) stocate în sistemele OLTP cum ar fi clienţii, produsele, vânzările, locaţiile şi datarea. De cele mai multe ori tabelele de dimensiuni au o structură internă la baza căreia stă organizarea ierarhică. De exemplu dimensiunea temporală este bazată pe două mari ierarhii: ,zi, săptămână, an- şi ,zi, lună, trimestru, an-. Similar, dimensiunea spaţială poate fi reprezentată ierarhic ca ,stradă, cod poştal, oraş, provincie, regiune, ţară, arie-. La rândul lor, produsele au structuri ierarhice precum ,produs, familie, tip- în sectorul producţiei şi ,produs, categorie, departament- în sectorul distribuţiei. Într-un fel, dimensiunile predetermină calea pe care analiza OLAP va putea fi dezvoltată ulterior.

**Tabelele de fapte .** Faptele se referă la tranzacţii şi cuprind două tipuri de date:

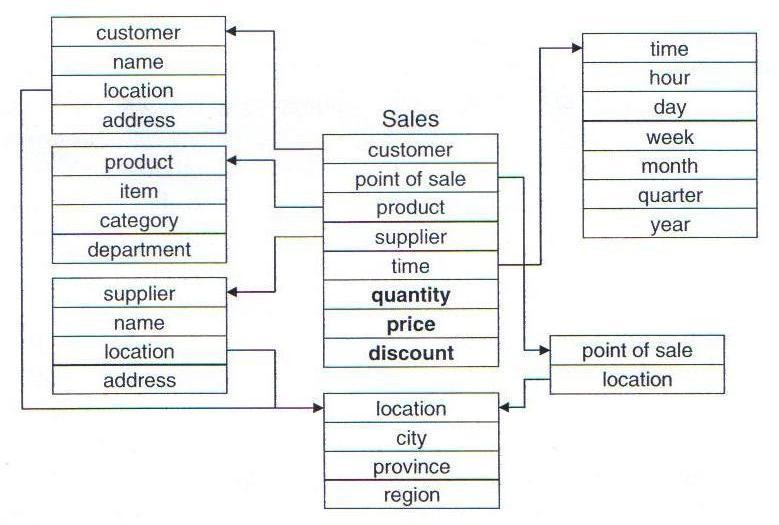
* legături către tabelele de dimensiuni, care sunt necesare pentru a face referire la informaţiile cuprinse în tabelele de fapte;
* valori numerice ale atributelor care caracterizează tranzacţiile şi care reprezintă scopul analizei OLAP care va urma.

De exemplu, o tabelă de fapte poate conţine tranzacţii legate de vânzări şi face referire la mai multe tabele de dimensiuni cum ar fi clienţi, puncte de lucru la care se face vânzarea, produsele vândute, furnizorii de produse, localizarea temporală. Următoarele puncte de interes sunt atribute precum cantitatea de produs vândută, preţul de vânzare al produsului şi discounturile oferite. În

acest exemplu tabelele de fapte permit analiştilor să evalueze trendul vânzărilor de-a lungul timpului, fie la nivelul tuturor clienţilor, la nivelul fiecărui client în parte sau la nivelul unui grup de clienţi care pot fi identificaţi uşor datorită organizării ierarhice a tabelului de dimensiuni asociate clienţilor. De asemenea, analistul poate evalua trendul vânzărilor aferente clienţilor localizaţi într-o anumită regiune.



**Figura 4.2** *Exemplu de schemă Stea*

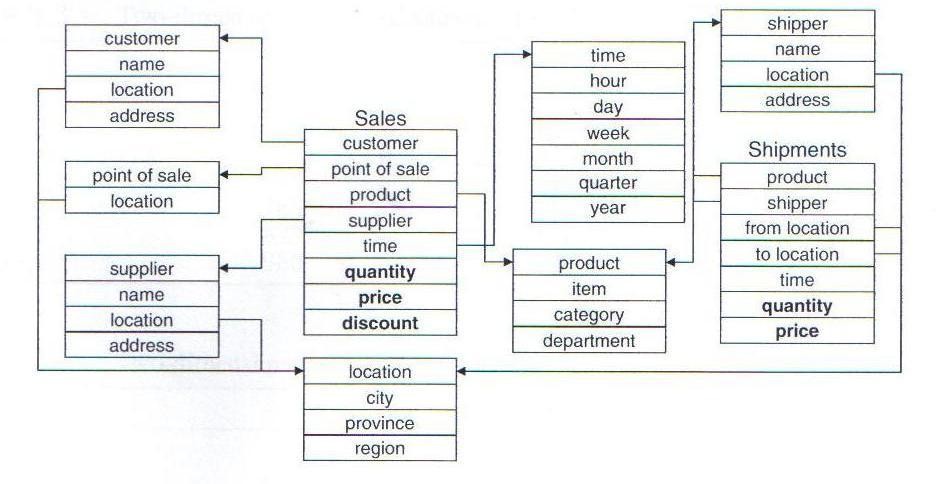


**Figura 4.3** *Exemplu de schemă Fulg de nea*

Figura 4.2 afişează schema stea asociată cu tabelul de fapte reprezentând tranzacţii de vânzare. Tabelul de fapte este plasat în mijlocul schemei şi este legat de tabelele dimensiune prin referinţe adecvate. Măsurile în tabelul de fapte apar cu caractere reliefate.

Uneori, dimensiunea tabelelor sunt conectate la rândul lor, la alte dimensiuni ale tabelelor, aşa cum se arată în Figura 4.3, printr-un proces de standardizare parţială a datelor în scopul de a reduce consumul de memorie. În exemplul dat, dimensiunea tabelului, referindu-se la locaţie, este la rândul său ierarhic în legătură cu tabelul de dimensiune care conţine informaţii geografice. Acesta aduce cu schema “ fulg de zăpadă”.

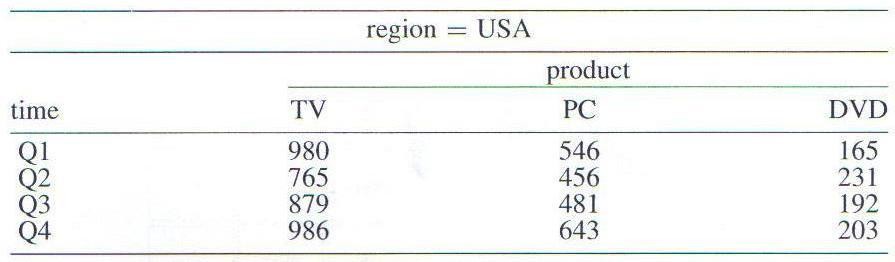
Un depozit de date cuprinde mai multe tabele de fapte, interconectate cu tabele dimensionale, legate la rândul lor, cu alte tabele dimensionale. Ultimul tip de schemă, din Figura 4.4 este numită *schema galaxie.*



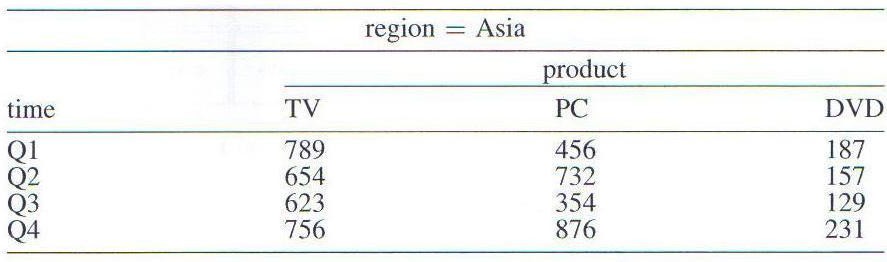
**Figura 4.4** Exemplu de schemă galaxie

O tabelă de fapte conectată cu n tabele dimensiuni poate fi reprezentată de un n dimensional cub de date unde fiecare axă corespunde unei dimensiuni. Cuburile multidimensionale au o extensie naturală a celor mai populare două foi de calcul dimensionale, care pot fi interpretate ca două cuburi dimensionale. De exemplu, considerăm o tabelă de fapte de vânzări dezvoltată de-a lungul celor trei dimensiuni (timp, produs, regiune). Presupunem că selectăm doar două dimensiuni pentru analiză, ca (timp, produs), având presetat atributul regiune de-a lungul a trei valori (USA, Asia, Europa). În acest fel obţinem trei – două tabele dimensionale în care fiecare rând corespunde trimestrelor unui an şi coloanele produselor (vezi Tabelul 4.5 – 4.6 ). Cubul din Figura 4.8 este o ilustraţie tri-dimensională a aceleaşi tabele fapte vânzări. Data atomică este reprezentată de 36 de celule care pot fi obţinute prin trecerea tuturor valorilor posibile prin cele trei dimensiuni: timp (Q1,Q2,Q3,Q4), regiune (USA, Asia,Europa) şi produse (TV, PC, DVD). Aceste celule atomice pot fi suplimentate de 44 de celule corespunzătoare valorii finale obţinută prin consolidarea uneia sau mai multe dimensiuni, ca în cubul din figură.

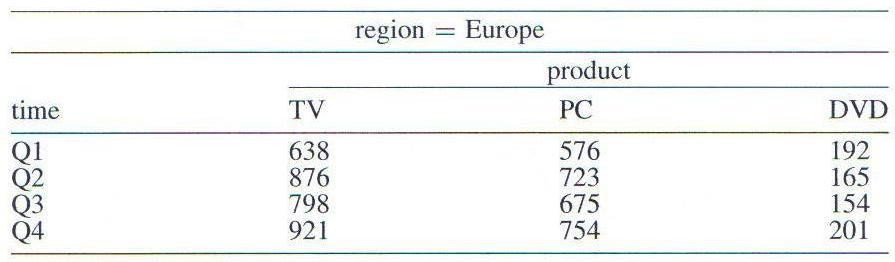
**Tabelul 4.5** *Două dimensiuni văzute la datele de vânzări în SUA*



**Tabelul 4.6** *Două dimensiuni văzute la datele de vânzări în Asia*

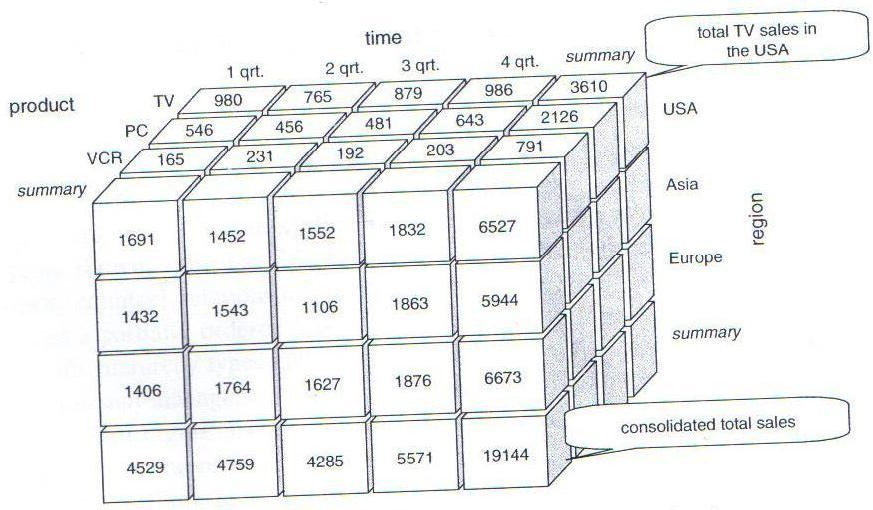


**Tabelul 4.7** *Două dimensiuni văzute la datele de vânzări în Europa*

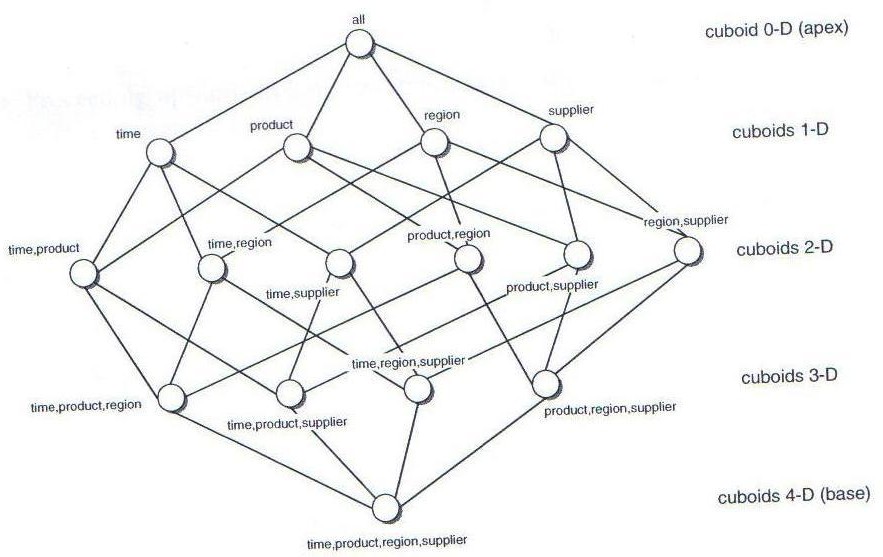


Presupunem că tabelele fapte vânzări conţin patru dimensiuni reprezentate de furnizori. Data corespunzătoare cubului constituie o structură în spaţiul cvasi-dimensional şi, prin urmare, nu pot fi reprezentate grafic. Cu toate acestea, putem obţine patru vizualizări logice compuse din trei cuburi tri-dimensionale, numite cuboiduri, în interiorul cubului cvasi-dimensionale, prin stabilirea valorilor de o singură dimensiune.

Mai general, pornind de la o tabelă de fapte conectată cu n tabele dimensiuni, este posibil să se obţină un grilaj de cuboiduri, fiecare dintre ele corespunzându-i un nivel diferit de consolidare de- a lungul uneia sau mai multor dimensiuni. Acest tip de agregare este echivalent în limbajul de interogare structurat (SQL), la o sumă de interogare derivat dintr-un grup de condiţii. Figura 4.9 ilustrează grilajele compuse de cuboiduri obţinute dintr-un cub de date definite de-a lungul celor patru dimensiuni (timp, produs, regiune, furnizor).



**Figura 4.8** *Exemplu de cub tri-dimensional*



**Figura 4.9** *Grilajele cuboidului derivat din cubul cvasi-dimensional*

Cuboidele asociate cu datele atomice, care, prin urmare, nu implică nici un tip de consolidare, sunt numite cuboide de bază. La cealaltă extremă, cuboidul apex este definit ca fiind cuboidul corespunzător consolidării de-a lungul tuturor dimensiunilor, prin urmare, asociat cu totalul general al măsurii de interes.

# Ierarhii de concepte şi operaţiuni OLAP

În multe cazuri, analizele OLAP sunt bazate pe ierarhii de concepte pentru a consolida datele şi pentru a crea vizualizări logice de-a lungul dimensiunii unui depozit de date. O ierarhie concept defineşte un set de hărţi de la un nivel mai scăzut de concepte la un nivel superior.

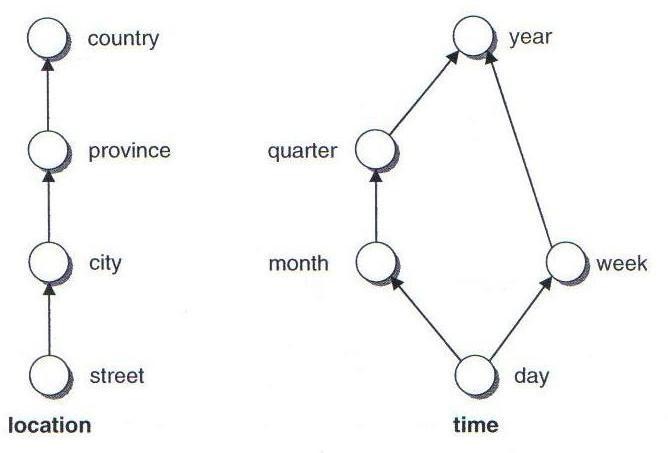
De exemplu, dimensiunea (locaţia) poate proveni dintr-o ierarhie ordonată în totalitate, aşa cum se arată în Figura 4.10, dezvoltându-se de-a lungul relaţiei (adresă, municipalitate, provincie, ţară). Dimensiunea temporală, pe de altă parte, provine dintr-o ierarhie parţial ordonată, de asemenea, arătată în Figura 4.10.

Specifice tipuri ierarhice pot fi predefinite în platforma software-ului folosit pentru crearea şi gestionarea unui depozit de date, ca şi în cazul dimensiunilor prezentate în Figura 4.10. Pentru celelalte ierarhii, este necesar pentru analişti să definească în mod explicit relaţiile dintre concepte.

Ierarhiile de concepte sunt, de asemenea, folosite pentru a efectua operaţiunile de vizualizare corespunzătoare care se ocupă cu cuburile de date într-un depozit de date.

**Roll-up.** Operaţiunea roll-up, numită şi drill-up, constă într-o agregare a datelor în cub, care pot fi obţinute, alternativ, în următoarele două moduri.

* Se porneşte în sus spre un nivel mai ridicat de-a lungul unei singure dimensiuni definite peste conceptele ierarhiei. De exemplu, pentru dimensiunea locaţie, este posibilă deplasarea spre partea superioară de la nivelul oraş către nivelul provincie şi pentru a consolida măsurile de interes prin suma condiţionată group-by peste toate înregistrările prin care oraşul aparţine aceleaşi provincii.



**Figura 4.10** *Ierarhii de concepte*

* Reducerea unei dimensiuni. De exemplu, eliminarea dimensiunii temporale duce la măsuri consolidate printre care suma perioadelor de timp existente în cubul de date.

**Roll-down.** O operaţiune roll-down, denumită şi drill-down, este operaţiunea opusă roll-up. Aceasta permite navigarea printr-un cub de date de la informaţii agregate şi consolidate la informaţii mai detaliate. Efectul este de a inversa rezultatul obţinut printr-o operaţiune roll-up. O operaţiune drill-down poate fi efectuată în două moduri.

* Trecerea la un nivel mai mic de-a lungul unei singure dimensiuni ierarhice. De exemplu, în cazul dimensiunii locaţie, este posibil să se treacă de la nivelul provincie la nivelul oraş şi de a se dezagrega măsuri de interes peste toate înregistrările prin care oraşul aparţine aceleaşi provincii.
* Adăugarea unei dimensiuni. De exemplu, introducerea dimensiunii temporale duce la măsuri dezagregate de interes faţă de toate perioadele de timp existente, intr-un cub de date.

**Slice şi dice.** Prin operaţiunea slice (felie) valoarea unui atribut este selectat şi fixat de-a lungul unei singure dimensiuni. De exemplu, tabelul 3.3 a fost obţinut prin stabilirea unei regiuni la valoarea de SUA. Operaţiunea dice obţine un cub într-un subspaţiu, prin selectarea mai multor dimensiuni simultane.

**Pivot.** Operaţiunea pivot, denumită şi de rotaţie, produce o rotaţie a axelor, pompează unele dimensiuni pentru a obţine viziuni diferite ale unui cub de date.

# Materializarea cuburilor de date

Analizele OLAP dezvoltate de către cei care lucrează la aplicaţie pot fi necesare pentru a accesa informaţiile asociate cu mai multe cuboiduri, pe baza interogărilor specificate şi a analizelor efectuate. În scopul de a garanta un timp suficient de răspuns, ar putea fi utilă proiectarea unui depozit de date în care toate (sau cel puţin o mare parte a) valorile de măsuri de interes asociate cu toate cuboidurile posibile sunt pre-calculate. Această abordare este denumită materializarea completă de informaţii în raport cu cuburile de date.

Se observă că, în cazul în care ierarhiile de concepte lipsesc, este posibil sa se formeze 2 *n* cuboiduri distincte din toate combinaţiile posibile de n dimensiuni. Existenţa unei ierarhii de-a lungul diferitelor dimensiuni face ca numărul de cuboiduri distincte să fie şi mai mare. Dacă L *i* reprezintă numărul de niveluri ierarhice asociate cu dimensiunea i, pentru un cub de date cu n dimensiuni, este posibil să se calculeze numărul total de cuboiduri, aceasta redată prin ecuaţia:

*n*

*T*  (*Li*  1).

*i*1

De exemplu, dacă un cub de date include 5 dimensiuni, şi dacă fiecare din aceste dimensiuni

include 3 nivele ierarhice, numărul de cuboiduri este egală cu

45  210  103 . Este clar că

materializarea completă a cuboidelor pentru toate cuburile asociate cu tabelele de fapt al unui depozit de date ar impune cerinţe de stocare, care ar putea fi cu greu susţinute în timp, având în vedere rata la care noi recorduri sunt colectate.

Pentru toate motivele de mai sus, este necesar să se găsească un echilibru între nevoia de acces rapid la informaţii, care ar sugera materializarea deplină a cuboidelor, precum şi nevoia de a păstra utilizarea memoriei în limite rezonabile. În consecinţă, materializarea preventivă ar trebui să fie efectuată numai pentru acele cuboiduri care sunt cele mai frecvent accesate, în timp ce celelalte calcule ar trebui să fie efectuate la cerere, numai atunci sunt efectuate întrebări reale care solicită informaţiile asociate. Această abordare din urmă este denumită în continuare materializarea parţială a informaţiilor în raport cu cuburile de date.

# Integrarea tehnologiei relationale cu tehnologia OLAP

Instrumentele OLAP utilizeaza tehnici analitice simple (analiza multidimensionala a datelor) pentru a analiza seturi mari de date. Majoritatea specialistilor sunt de acord ca depozitele de date împreuna cu instrumentele OLAP ofera posibilitatea de a transforma cantitatile uriase de date din firme, în informatii utile procesului decizional. De asemenea, folosirea tehnicilor analitice oferite de instrumentele OLAP împreuna cu depozitele de date si facilitatile oferite de Web, permit un acces mai usor si mai rapid la informatiile necesare procesului decizional modern. Cele mai multe aplicatii BI construite cu instrumente OLAP (de exemplu, pentru analiza costurilor, analiza profitului, analiza vânzar ilor, analize de marketing etc .) utilizeaza baze de date multidimensionale care permit cereri multidimensionale complexe (de exemplu „Care a fost procentul de crestere a volumului vânzarilor anul acesta fata de anul trecut, pent ru fiecare din primele 10 produse vândute?”). Utilizarea acestor baze de date este costisitoare si sunt destinate de regula managerilor si analistilor (deci unui numar restrâns de utilizatori).

La ora actuala exista doua tendinte în aplicatiile BI:

1. *O corelare cât mai buna între activitatea de analiza si cea de planificare*. Aplicatiile BI trebuie:
   1. sa furnizeze acces rapid unui numar mare de utilizatori, distribuiti geografic, la un stoc de date centralizat;
   2. sa pe rmita agregarea complexa a datelor;
   3. sa utilizeze facilitatile oferite de Internet. Existenta unui catalog de metadate centralizat si a unui stoc de date centralizat poate sa le ofere utilizatorilor

o viziune comuna asupra datelor.

Toti utilizatorii trebuie sa acceseze acelasi model de date si aceleasi date.

1. *Convergenta aplicatiilor BI cu aplicatiile operationale (tranzactionale).* Activitatile operationale si cele de analiza constituie nucleul activitatii unei firme, independent de marimea ei, domeniul de activitate, forma legala. Informatiile despre vânzari, productie si costuri pot fi înregistrate si gestionate în una sau mai multe baze de date, folosite pentru scopuri operationale. Activitatile operationale se executa la un interval relativ constant. Datele sunt citite si actualizate frecvent si reprezinta o fotografie curenta a ceea ce se întâmpla în firma. Fiecare cerere foloseste un volum mic de informatii, iar natura ei este în general previzibila. Monitorizarea, evaluarea, compararea, planificarea si alocarea strategica a resurselor sunt exemple de activitati de analiza. Informatia generata prin activitatile de analiza este orientata pe decizie, deoarece este într-o forma care o face imediat utilizabila în procesul decizional. Datele sunt mai mult citite decât actualizate în aceste activitati. Cererile analitice folosesc date derivate si natura lor nu este întotdeauna previzibila. Ca urmare a acestor diferente, majoritatea firmelor folosesc instrumente diferite pentru cele doua tipuri de activitati:
2. pentru a asigura eficienta maxima în ambele activitati;
3. pentru a realiza actualizare rapida în activitatiletranzactionale si calcul rapid în activitatile de analiza.

De asemenea, aplicatiile BI pot utiliza pentru stocarea datelor fie baze de date relationale (depozit de date/centru de date) sau baze de date specializate (de regula multidimensionale). Stocarea datelor se face într-o baza de date relationala atunci când:

1. volumul de date este prea mare pentru a fi duplicat;
2. datele sursa se modifica frecvent si este mai bine de a citi în timp real decât din copii;
3. integrarea cu alte sisteme informatice relationale existente;
4. organizatia are o politica de neduplicare a datelor în alte structuri de fisiere, pentru securitate sau alte motive, chiar daca aceasta conduce la aplicatii mai putin eficiente.

De asemenea, depozitele de date pot fi accesate de o mare varietate de instrumente de raportare si interogare ce utilizeaza limbajul SQL. Totusi limbajul SQL standard nu permite

operatii multidimensionale. Specialistii ofera trei solutii pentru a adauga functionalitate multidimensionala la datele stocate în baze de date relationale:

1. integrarea procesarii multidimensionale în sistemul de gestiune a bazei de date relationale, fie prin extinderea limbajului SQL sau prin adaugarea functionalitatii multidimensionale în nucleul

SGBD-ului (de exemplu Oracle 9i cu optiunea OLAP);

1. executarea în mai multi pasi a comenzilor SQL (de exemplu Microstrategy); instrumentul OLAP realizeaza o serie de comenzi SELECT, în care iesirile comenzilor anterioare sunt stocate în tabele temporale,

care sunt apoi utilizate de comenzile urmatoare;

1. încarcarea datelor relevante din tabelele corespunzatoare pe un server de aplicatie intermediar, unde este realizata procesarea multidimensionala.

Bazele de date multidimensionale ofera facilitati analitice complexe, o performanta mai buna la interogare, dar întretinerea unei astfel de baze de date este costisitoare. O baza de date multidimensionala cere replicarea datelor, un proces costisitor care poate cauza si o întârziere semnificativa în utilizarea datelor de catre analisti. De asemenea, se cere modelarea separata a datelor (modelarea multidimensionala) si multe baze de date multidimensionale nu ofera facilitatea de recuperare a erorilor si alte facilitati specifice bazelor de date relationale. SGBDMD cer antecalcularea tuturor agregatelor posibile, astfel sunt adesea mai performante decât SGBDR traditionale, dar mai dificil de actualizat si administrat. Stocarea fizica a datelor multidimensionale,

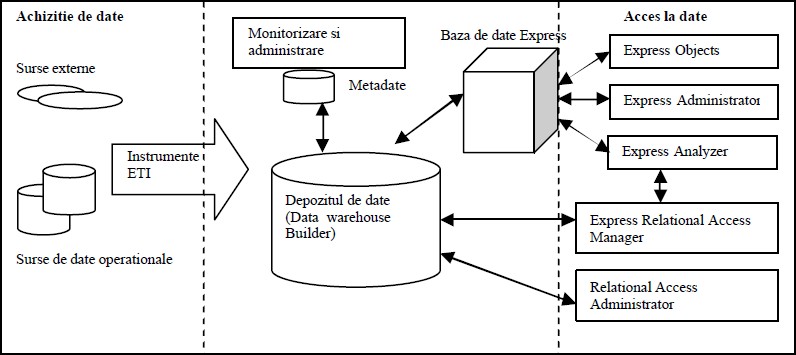
precum si fenomenul de împrastiere sunt preocupari majore, în domeniul bazelor de date multidimensionale. O tehnica de stocare a datelor optima ar trebui sa tina cont de multi factori dinamici si anume: profilul datelor si volumul lor (numarul de dimensiuni si membrii ai dimensiunilor, tipuri de date etc), fenomenul de împrastiere, frecventa de modificare în sursele de date (cât de des vor fi actualizate bazele de date multidimensionale), frecventa de modificare în datele multidimensionale (de exemplu pentru analiza de tip „*what if”*), frecventa de modificare în modelul multidimensional , accesul concurent etc. În concluzie, utilizarea celor doua tehnologii separat implica:

1. doua baze de date gestionate separat;
2. dictionare de metadate separate;
3. modele de securitate separate pentru BDR si BDMD;
4. instrumente de administrare separate;
5. mai mult personal pentru administrare;
6. o interfata OLAP API special proiectata pentru a accesa baze de date multidimensionale ;
7. de regula, aplicatiile care utilizeaza limbajul SQL nu pot interoga datele din bazele de date multidimensionale;
8. replicarea datelor din baza de date relationala în baza de date multidimensionala;
9. actualizarea datelor se face în doua etape, mai întâi depozitul de date, apoi baza de date multidimensionala ; (acest proces este consumator de resurse si timp); de asemenea, creste perioada de timp între momentul actualizarii surselor de date si momentul actualizarii bazei de date multid imensionale ;
10. din perspectiva unui administrator de baza de date se utilizeaza trei tehnologii diferite (instrumentul de integrare a datelor, tehnologia relationala si tehnologia OLAP), trebuie gestionate trei dictionare de metadate si exista

doua stocuri de date (depozitul de date si baza de date multidimens ionala). De exemplu, aplicatiile BI, construite pâna acum cu produse Oracle, utilizau:

1. *Oracle Express Server/Oracle Personal Express,* un server OLAP (datele sunt stocate într-o baza de date multidimensionala);
2. *Oracle Express Administrator* un utilitar grafic pentru definirea, întretinerea si popularea bazelor de date multidimensionale Express;
3. *Oracle Express Relational Access Manager* (RAM) care permite ca *Express Analyzer* sa acceseze datele stocate într-o baza de date relationala (configurata cu *Oracle Relational Access Administrator*-*RAA*);
4. *Oracle Warehouse Builder* pentru a crea depozitul de date si pentru a se încarca date în depozitul de date.

*Oracle Warehouse Builder* include si instrumentele ETI (figura 4.4.1) .



**Fig.4.4.1***. Arhitectura pe componente a unei aplicatii BI construita cu produse Oracle*

# Avantajele unui mediu integrat relational-multidimensional

Integrarea tehnologiei relationale (utilizata frecvent în aplicatiile operationale) cu tehnologia OLAP (utilizata frecvent în aplicatiile BI) ar permite realizarea de aplicatii cu o functionalitate

ridicata si mai putin costisitoare, dar care ar oferi un suport adecvat atât pentru activitatea de analiza cât si pentru activitatile operationale. Va exista un singur stoc de date de gestionat si un singur dictionar de metadate. De asemenea, trebuie implementata o singura politica de securitate. De exemplu, *Oracle 9i cu optiunea OLAP* realizeaza integrarea tehnologiei relationale cu tehnologia OLAP (figura 4.4.2). Oracle 9i utilizeaza baza de date relationala atât pentru aplicatii operationale cât si pentru aplicatii BI. Instrumentul *Oracle Enterprise Manager* si limbajul PL/SQL sunt utilizate pentru administrarea bazei de date Oracle. Datele necesare analizelor sunt valabile 24 de ore din 24*.* Toti utilizatorii vad aceeasi versiune a datelor imediat dupa actualizare.

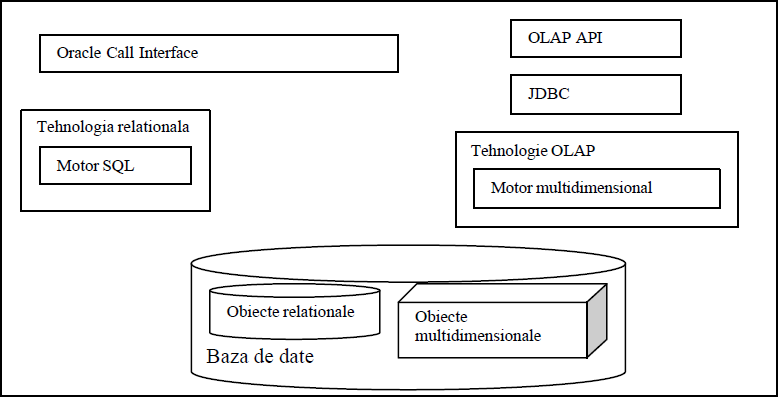
Datele pot fi stocate în tabele relationale (tabele de fapte si tabele de dimensiuni) sub forma de schema stea sau fulg de zapada sau în variabile multidimensionale. Dezvoltatorii de aplicatii pot utiliza OLAP API sau limbajul SQL pentru a accesa datele (stocate în tabele relationale sau variabile multidimensionale). De exemplu *Oracle Reports* si *Oracle Discoverer* acceseaza datele prin SQL, iar *BI Beans* sau *JDeveloper* prin *OLAP API*. De asemenea, datele (stocate în tabele relationale sau variabile multidimensionale) sunt stocate fizic în fisiere de date Oracle.

*Motorul multidimensional* poate executa calcule multidimensionale (previziuni, scenarii

„what-if”, manipularea seriilor de timp etc.) pe datele stocate în *spatiul de lucru analitic* (*analytic workspaces-AW*). Limbajul de manipulare a datelor OLAP (OLAP DML) este echivalentul

„multidimensional” al lui PL/SQL si extinde facilitatile analitice ale limbajului SQL. Atât OLAP API cât si PL/SQL pot include comenzi din limbajul de manipulare OLAP.

Un *spatiu de lucru analitic* contine un numar de obiecte multidimensionale (dimensiuni, variabile, relatii, programe, formule, modele etc.) si poate fi gândit ca o schema multidimensionala. Într-o baza de date se pot crea mai multe *spatii de lucru analitice*. Un *spatiu de lucru analitic* apartine unui utilizator si alti utilizatori pot avea acces la el. De asemenea, *un spatiu de lucru analitic* poate fi t mporar (pe durata sesiunii de lucru) sau poate fi persistent (stocat ca LOB în tabelele bazei de date). *Spatiile de lucru analitice* sunt o alternativa pentru stocarea datelor agregate ca si viziunile materializate.



**Fig. 4.4.2*.*** *Integrarea tehnologiei relationale cu tehnologia OLAP în Oracle9i*

Datele nu trebuie replicate în doua stocuri de date, astfel se reduce intervalul de timp între momentul actualizarii surselor de date si momentul când datele pot fi analizate. De asemenea, toti utilizatorii au acces la aceeasi versiune de date.

*Oracle OLAP API* este o interfata Java API ce permite conexiunea la baza de date, operatii de navigare (drill down, pivot etc .), selectia datelor, facilitati de analiza a datelor etc. *OLAP API* realizeaza conexiunea la baza de date prin *JDBC*. Pentru a accesa datele, *OLAP API* utilizeaza dictionarul OLAP. *Oracle 9i OLAP* transforma toate cererile OLAP API în cereri SQL. *Clientii SQL* se pot conecta la baza de date prin *Oracle Call Interface* (OCI) sau prin *JDBC*. *Oracle 9i OLAP* utilizeaza un model de date multidimensional. Elementele modelului multidimensional sunt: masurile, dimensiunile si atributele. Metadatele ce descriu modelul multidimensional utilizat de *Oracle 9i OLAP* sunt salvate în dictionarul OLAP si sunt create cu instrumentul *OLAP Management* integrat în *Oracle Enterprise Manager.* Aplicatiile pot accesa optiunea OLAP prin:

*Java OLAP API,* limbajul SQL sau comenzi din limbajul de manipulare OLAP.

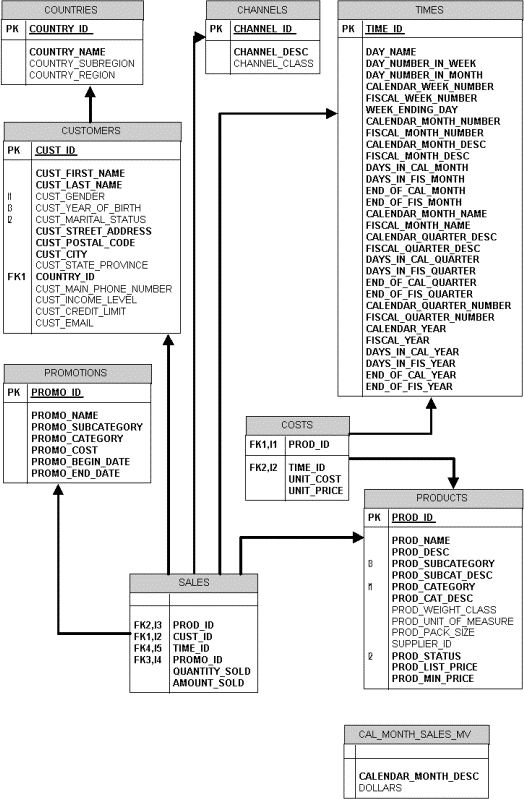
# Concluzii

Depozitele de date sunt concepute special pentru sprijinirea luării deciziilor. Ele au ca obiectiv regruparea datelor, agregarea şi sintetizarea lor, organizarea şi coordonarea datelor provenind din surse diferite, integrarea şi stocarea acestora pentru a da decidenţilor o imagine adecvată care să permită regăsirea şi analiza eficace a informaţiilor necesare. Interogările obişnuite într-un depozit de date sunt mai complexe şi mai variate decât cele din sistemele de gestiune a bazelor de date. Tehnologia OLAP este cea care are de a face cu toţi aceşti factori critici (volumul mare de date, complexitatea calculelor şi timpul de răspuns), transformând volumul imens de date stocate şi gestionate în depozite, în informaţii utile, procesul de decizie just-in-time.

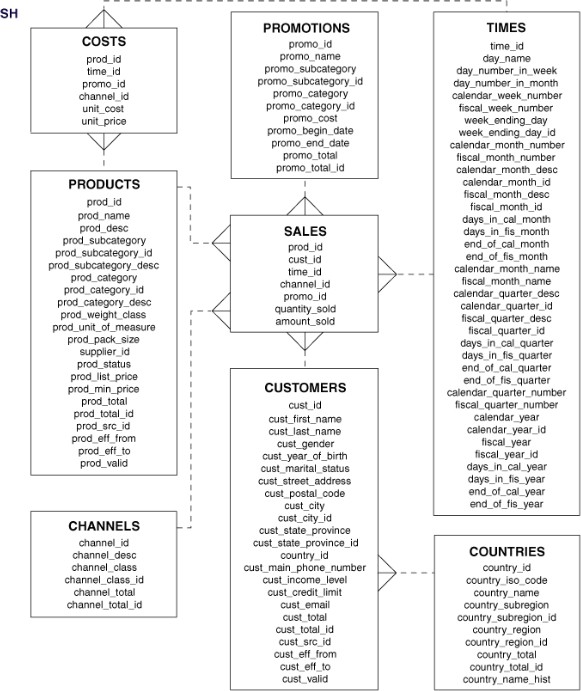
Cuburile de date sunt construite din dimensiuni. Ele sunt cele care oferă posibilităţile de vizualizare a datelor. O dimensiune poate conţine un singur criteriu de vizualizare a datelor, de exemplu după provincii, dar în mod obişnuit o dimensiune defineşte mai multe criterii organizate ierarhic, după care se pot vizualiza datele. Deci dimensiunile sunt cele care permit analiza „în adâncime” (drill-down) a datelor plecând de la nivel scăzut de detaliere şi înaintând progresiv spre niveluri din ce în ce mai ridicate de detaliere.

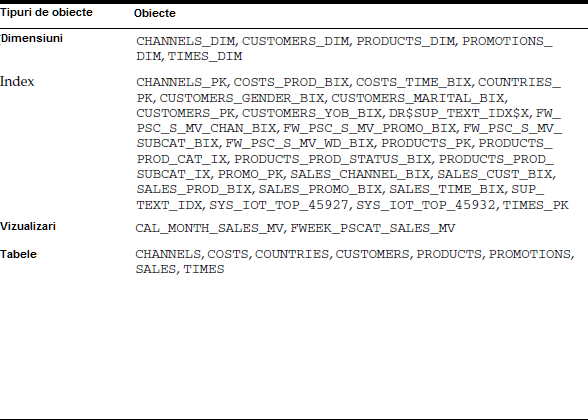
# Structura bazei de date

Baze de date SH (Sales History) contine un volum mare de date business ale unei companii model, avand posibilitatea de a dezvolta rapoarte statistice. Aceste rapoarte pot ajuta in luarea deciziilor asupra companiei de catre manageri. Majoritatea rapoartelor sunt in functie de timp si sunt nevolatile, analizand tendinta vanzarilor. Compania incarca informatii in mod constant in depozitul de date pentru analizarea noilor statistici.



Schema Sales History





Obiecte din cadrul depozitului de metadata ce sunt folosite la construirea rapoartelor

# Dezvoltarea aplicatiei de Business Intelligence (Sales Trend)

Spre deosebire de alte aplicaţii software, cum ar fi sistemele informaţionale şi instrumentele de automatizare a activităţii de birou, sistemele BI sunt disponibile de obicei ca programe standard. Mediul multidimensional de analiza au facilitat şi standardizat accesul la funcţii pasive de inteligenţă pentru afaceri. Oricum, pentru a putea dezvolta majoritatea sistemelor BI, este nevoie de un proiect specific. Figura 2.10 arată paşii majori de dezvoltare. Fluxul logic al activităţilor desfăşurate este ilustrat prin săgeţile complete. Săgeţile întrerupte, desenate în direcţia opusă indică revizia în una sau mai multe faze care ar putea fi necesară în timpul dezvoltării sistemului, printr-un mecanism de feedback.

Analiză

Planificare

Cerinţe

Implementare

Modelare

*Figura 2.10 Fazele dezvoltării unui sistem BI*

Livrare

Scopul principal al PLANIFICARII este acela de a înţelege nevoile şi oportunităţile, caracterizate adeseori drept *semnale slabe*, şi transformarea acestora într-un proiect iar mai târziu într-un sistem BI de succes.

In cadrul proiectului Sales Trends, am pornit de la dorinta de a analiza istoricul de vanzari din cadrul unei companii pentru a putea analiza datele intr-un mod mult mai usor si mai fiabil, care sa permita ulterior luarea unor decizii cu un risc cat mai mic pentru companie, si mai mult decat atat, sa ofere posibilitatea de crestere a profitul in functie de fluxul de vanzari, in functie de analiza comparative cu alte companii similar, etc.

În faza de ANALIZĂ, este necesară definirea în detaliu a funcţiilor ce trebuie dezvoltate în sistemul BI, prin dezvoltarea ulterioară şi elaborarea concluziilor preliminare dobândite pe parcursul dezvoltării studiului de fezabilitate. Ar trebuie prin urmare să se răspundă la întrebarea : Ce ar trebui să se atingă printr-o aplicatie BI, cine va folosi, când şi cum? . În final este necesară exploatarea datelor în aşa fel încât să se înţeleagă cât de multe şi ce fel de tip de informaţii există deja şi care informaţii pot fi sustrase din surse externe.

In faza de analiza pentru proiectul Sales Trend, se doreste urmarirea fluxului de date din trecut de catre manager, acesta avand posibilitatea sa analizeze unele fluctuatii din cadrul vanzarilor, sa determine profitul minim, maxim, sa determine un target sigur pentru vanzari,etc.

În timpul etapei de MODELARE întrebarea principală este: Cum va funcţiona aplicatia? Întreaga arhitectură a sistemului este definită în această etapă prin identificarea platformelor tehnologice hardware , structura reţelei, instrumentelor software pentru dezvoltarea aplicaţiei şi

baza de date ce va fi folosită. Este de asemenea necesară definirea în detaliu a interacţiunilor cu utilizatorii, prin mijloace de introducere a unor măşti, vizualizări grafice pe ecran şi rapoarte printate.

Modelarea proiectului Sales Trend se imparte in doua directii:

* + 1. Modelarea arhtecturii generale – s-au construit cele 3 layere de admistrare a aplicatiei: fizic, business si cel de prezentare. S-au construit tabele agregate, s-au imbinat mai multe baze de date si s-au implementat cuburi olap.
    2. Modelarea cunostintelor in cadrul interfetei grafice – Pentru crearea rapoartelor s-a plecat de la cerintele managerilor, spre examplu de la urmarirea profitului in ultimile luni, si costurile implicate in tot acest timp. Pentru un astfel de raport s-au construit diverse filtre, agregari de tabele, masti, etc.

Odată ce au fost elaborate specificaţiile, este timpul pentru IMPLEMENTAREA propriu-zisă, testarea şi instalarea, când sistemul de BI va fi pus la treabă.

Aplicatia a fost realizata cu ajutorul urmatoarelor tehnologii: Oracle 11g R2, Oracle Analytic Workspace Manager, Oracle Business Intelligence Enterprise Edition 10.1.3.2, OBIEE Plug-in for Analytic Workspace Manager.

Oracle 11g R2 - Oracle Database 11***g*** doboară toate recordurile în materie de performanţe şi scalabilitate pe servere Windows, Linux, şi UNIX. Furnizează un ROI rapid, permiţându-le utilizatorilor să treacă de la un singur server la Grid Computing, fără a modifica nicio linie de cod.Oracle Database 11***g*** permite obţinerea unor rezultate mai bune prin automatizarea sarcinilor administrative şi prin oferirea unor caracteristici de securitate şi de conformitate fără egal pe piaţă.

Oracle Analytic Workspace Manager – ofera posibilitatea de a construi intr-un mod extrem de usor, fiabil si sigur modele de date dimensionale, cuburi OLAP, crearea unui sistem business intr- un mod iterativ. Siguranta datelor este la un nivel ridicat in cadrul acestei platforme.

Oracle Business Intelligence Enterprise Edition – Acest produs aduce un pachet mare de tooluri ce permit o dezvoltare puternica de solutii analitice si creare de rapoarte complexe.

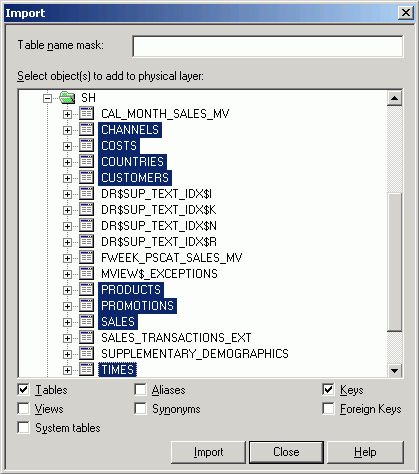
# Crearea layerului fizic

Dupa cum am precizat si in capitolele precedente, toolul de administrare a depozitului de metadate din cadrul platformei de Business Intelligence este impartit in 3 layere. Se porneste cu construirea layerului fizic.

Layerul fizic se defineste ca fiind sursa de date ce contine serverul BI care realizeaza inteorgari si legaturile dintre sursa da date fizica existenta pe server si multiple alte surse de date externe. O modalitate sigura si usoara de creare a layerului fizic este de a importa date, iar apoi se pot crea manual diverse obiecte. Prin importul de date se pot configura automat proprietatile, legaturile, care existau in cadrul sursei de date.

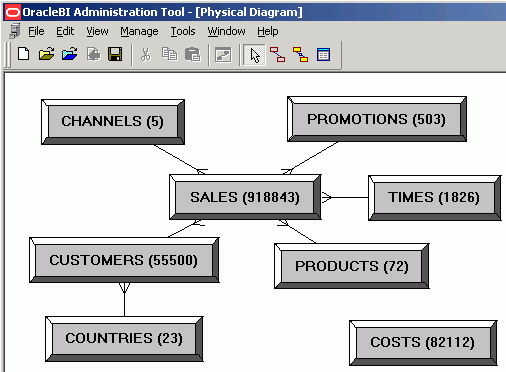
Pentru inceput, se pornesc toate serviciile bazei de date si ale platformei de business intelligence.

*Pentru a realiza importul de date este necesara crearea unei conexiuni la baza de date. Am folosit Data Sources (ODBC)* ***(****Start > Programs > Administrative Tools > Data Sources (ODBC)****) .*** *In momentul importarii datelor in cadrul depozitului de metadate am selectat doar o parte din tabele.*



Avem tabele, continutul, acum este nevoie de lagurile dintre ele. Am creat chei straine astfel: promotions.promo\_id - >sales.promo\_id

times.time\_id - > sales.time\_id products.prod\_id - >sales.prod\_id customers.cust\_id - > sales.cust\_id countries.country\_id - > customers.country\_id



In finalul acestei etape, am verificat consistenta acestei structuri folosind Global Consistansy Check. Acesta ma ajuta in a determina diverse erori, sau pot vizualiza eventuale avertismente.

# Crearea Layerului de Modelare si Mapare Business

Layerul de Modelare si Mapare Business reprezinta un model business sau un model logic de date si realizeaza maparea intre modelul business si cel fizic. In cadrul acestui layer se simplifica datele fizice pentru a putea crea un format de baza a datelor si pentru a permite utilizatorilor sa vizulizeze informatiile intr-un mod sugestiv. Rolul de baza al acestui layer consta in crearea unui model reprezentativ pentru felul in care utilizatorii isi percep businessul folosind vocabularul lor propriu, economic. Astfel, Layerul Business este format din colectii de coloane, tabele logice. Fiecare coloana logica contine unul sau mai multe obiecte fizice.

Exista doua categorii de tabele logice: dimensiuni si fapte. Tabelele de fapte contin masuri prin care orice companie isi poate masura performantele si operatiunile pe care le realizeaza. Tabelele de dimeniuni contin date pentru a valorifica faptele.

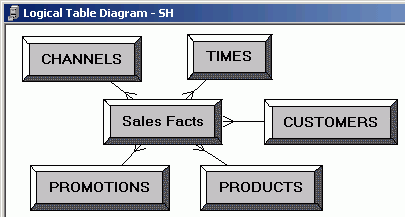
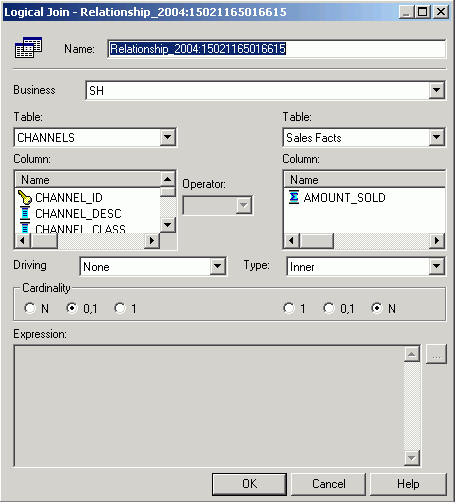
Pentru a crea Layerul de Modelare si Mapare Business trebuie indepliniti urmatorii pasi

majori:

* crearea unui model business
* crearea tabelelor logice
* crearea coloanelor logice
* crearea legaturilor logice
* redenumirea obiectelor business
* stergerea obiectelor ce nu sunt necesare
* construirea ierarhiilor dimensionale

Am inceput prin creerea unui model business pe care l-am numit SH (Sales History). Am creat

un prim tabel logic numai Sales Facts, iar in cadrul acestuia am creat o coloana logica numita AMOUNT\_SOLD cu referire la AMOUNT\_SOLD din tabela SALES din cadrul layerului fizic. Acestei coloane i-am aplicat functia de agregare SUM, pentru a putea realiza volumul de vanzari. Pe langa tabela de Sales Facts, am adaugat urmatoarele tabele logice din cadrul layerului fizic: CHANNELS, CUSTOMERS, PRODUCTS, PROMOTIONS, TIMES. Urmeaza crearea legaturilor intre tabele.

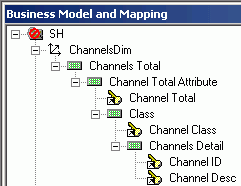


Astfel, am creat o schema stea ce contine un tabel de fapte (Sales Facts) si 5 tabele de dimensiuni (Channels, Customers, Products, Promotions, and Times).

Introducerea de ierarhii dimensionale permite serverului BI calcularea unor masuri intr-un mod precis, si permite utilizatorilor sa vizualizeze datele cat mai in detaliu. Cele mai des intalnite ierarhii de dimensiuni sunt cele de timp, produse, client, furnizori, etc.

Ierarhiile dimensionale nu sunt vizualizate de catre utilizatorii finali in tooluri precum Oracle BI Answer sau Interactive Dashboad. In orice ierarhie de dimensiuni, se stailesc atributele dimensiunilor pe nivele. Aceste nivele reprezinta regulile organizationale si necesitatile de raportare specific businessului. Ele ofera o structura capabila de a vizualiza toate datale necesare, parcurgerea realizandu-se asemanator ca la arbori, atat pe adancine, cat si pe latime.

Am creat tabelul Channels de dimensiuni si am aplicatele atributele specifice.



Pentru dimensiunea Channels Detail am atribuit lui Channel Desc proprietatea de *Use for Drilldown* , astfel se vor putea parcurge toate datele pe adancime din cadrul coloanei.

Am creat de asemenea tabele de dimensiuni pentru Customers, Products, Promotions,

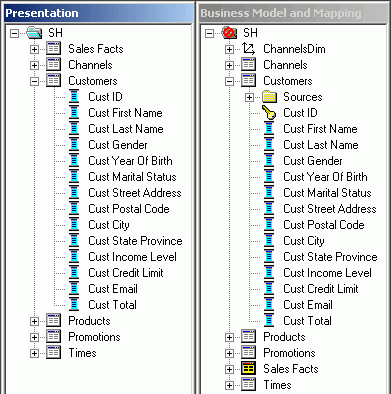
Times.



# Crearea Layerului de Prezentare

Layerul de Prezentare este construit dupa modelul business si cel fizic, iar pe langa acestea se adauga o nota abstracta asupra layerului de modelare si mapare business. Reprezinta partea de vizualizare a datelor de catre utilizatorii finali prin intermediul toolurilor precum Oracle BI Answer. Simplificand modalitatea de vizualizare a datelor se pot face intoragari foarte usor in functie de necesitatile business, afisand doar informatiile necesare utilizatorului si organizandu-le in functie de modalitatea de gandire a acestuia. Crearea le obiecte in acest layer se poate face prin drag and drop din layerul business. Obiectele corespunzatoare sunt create automat in layerul de prezentare, iar acestea pot fi modificate in functie de necesitati.

Pentru inceput am creat un layer de prezentare SH identic cu layerul business folosind drag and drop, iar obiectele au fost create automat.



# Dezvoltarea arhitecturii de business

*Cum se pot crea masuri pentru analiza businessului*

Masura este o functie numerica ce poate fi evaluata în orice punct din spatiul cubului de date reprezintand valorile centrale ce sunt analizate.

In cele mai multe cazuri, intr-un business sunt necesari termini de comparatie, iar masurile ajuta la evidentierea diferentelor. Masurile permit utilizatorilor finali sa puna intrebari de genul : “Arata-mi conturile si balanta pe trimestrul 3” sau “Prezinta diferenta dintre unitatile comandate si unitatile vandute”,etc. Tooul pentru crearea de expresii te ajuta sa creezi interogari similare ca cele create cu SQL. Cu ajutorul acestui tool am creat diferse masuri care apar ca niste coloane pentru a le putea folosi mai usor in Oracle Answer. Utilizatorii pot sa le foloseasca foarte usor pentru a crea interogari.

Exista mai multe modalitati de a crea masuri in cadrul componentei Administration Tool. Se pot folosi: coloane logice, coloane fizice, sau wizardul de calcul, pentru a crea formule special pentru masuri.:

* + Se folosesc coloane fizice pentru calcularea formulei, aceasta necesitand reguli de agregare ce se aplica dupa de calculul propriu-zis.
  + Se folosesc coloane logice pentru calcularea formulei, aceasta necesitand reguli de agregare ce se aplica inainte de calculul propriu-zis.
  + Se pot construe masuri in cadrul platformei Oracle Answer.

Un avantaj major al construirii masurilor in cadrul depozitului de metadata este construirea lor o singura data si refolosirea masurilor de catre utilizatorii finali in orice moment. Avantajul definirii formulei pe baza unei coloane logice consta in crearea calcului o singura data, pe cand crearea forumelei pe baza coloanelor fizice presupune si maparea pe fiecare sursa de date fizica din care deriva datele.

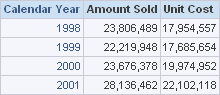
*Crearea unei masuri*

Din cadrul layerului fizic am selectat UNIT\_COST din tabele COSTS. Aceasta coloana am adaugatat-o si in layerul business folosind drag and drop, si am asezat-o in tabela SALES FACTS. Am redenumit coloana in Unit Costs, si acesteia am aplicat o functie de agregare, mai precis, functia SUM. Intr-un final am adaugat coloana si in layerul de prezentare. Am verificat depozitul de date de erori si atentionari folosind Check in Changes. Pentru o mai buna testare, am verificat coloana in Oracle Answer.

Am adaugat urmatoarele tabele:



Rezultatul este:



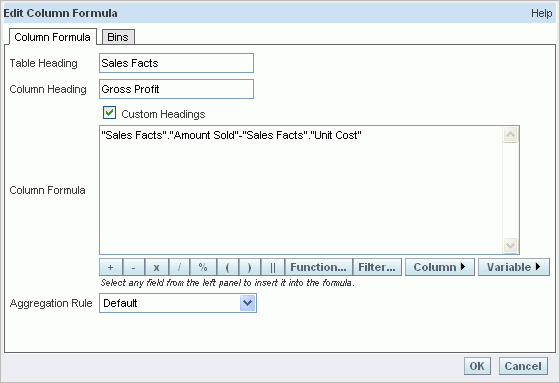
*Crearea unei masuri folosind Oracle Answer*

Inainte de crearea unei masuri in cadrul depozitului de metadata, trebuie mai intai create masura folosind Oracle Answers. Acest lucru ajuta la compararea rezultatelor ulterioare atunci cand creezi masuri similare in depozitul de metadata.

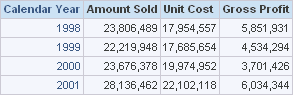
Pentru a crea o masura pentru “Profitul Brut” (Gross Profit), am creat in primul rand urmatorul tabel, ce contine datele necare pentru calcularea profitului.



Profitul Brut consta in venitul total al unei companii dupa ce a fost scazut costul bunurilor vandute. Astfel formula arata in felul urmator:

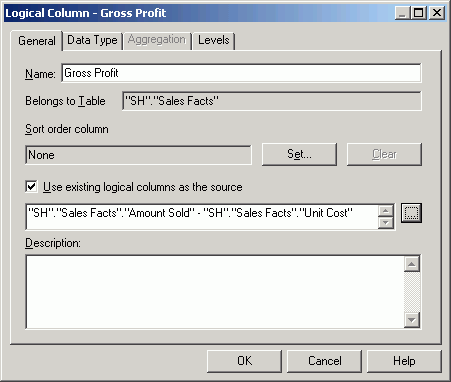


Iar rezultatul este:

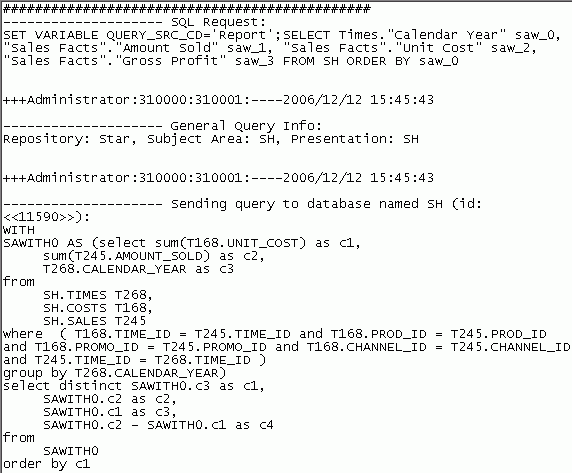


*Crearea unei masuri folosind coloane logice*

Am creat deasemenea masura pentru Profitul Brut folosind coloane logice in cadrul formulei. Astfel, folosim din nou toolul de administrare pentru depozitul de metadata SH. In layerul business, in cadrul tabelei logeice Sales Facts, am creat o noua coloana numit Gross Profit. Acestei coloane am aplicat formula Profitului Brut astfel:

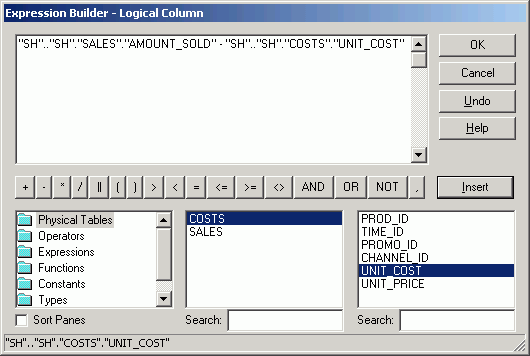


Am adaugat coloana Gross Profit si in layerul de prezentare, iar apoi am testat in Answer valorile. Rezultatul a fost acelas ca si cel prezentat la pasul anterior. Diferenta dintre AMOUNT\_SOLD si UNIT\_COST, este calculate in exteriorul blocului de interogari. Datorita faptului ca am folosit coloane logice in calcularea formulei pentru Gross Profit, coloanele sunt mai intai insumate si apoi se aplica diferenta. La urmatorul pas vom observa o alta modalitate de interogare a bazei de date.



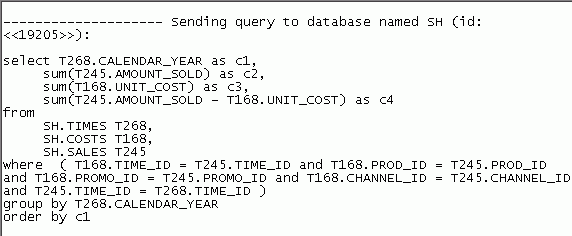
*Crearea unei masuri folosind coloane fizice*

In tabela logica Sales Facts, am creat o noua coloana logica Gross Profit Physical si i-am atasat functia de agregare SUM. Apoi, urmeaza maparea aceste coloane cu sursa de date fizica. Am creat formula pentru Profitul brut selectand coloanele fizice:



Am adaugat coloana si in layerul de prezentare. Rezultatul este acelas ca si la pasii anteriori.

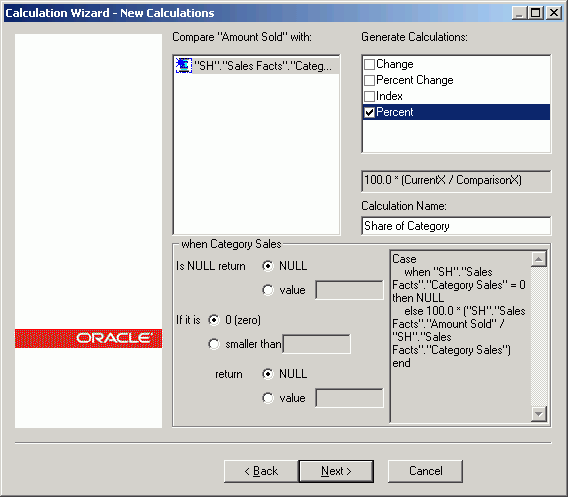
Formula pentru Profitul Brut folosind coloane logice arata in felul urmator “sum(Amount Sold) – sum(Unit Cost)” , iar formula ce contine coloane fizice este sum(AMOUNT\_SOLD – UNIT\_COST). Interogarea arata in felul urmator:



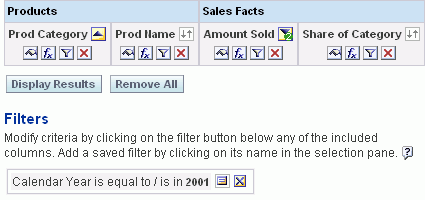
*Crearea unei masuri folosind wizardul de calcul*

In cadrul layerului business, am duplicat coloana Amount Sold din tabela Sales Facts, iar cea de-a doua coloana am redenumit-o Category Sales. Acestei coloane i-am atribuit nivelul logic pentru ProductsDim. Astfel, Category Sales este o masura de nivel ce ajuta la calcularea vanzarilor totale atunci cand este folosita in cadrul interogarilor. Masurile de nivel ajuta la crearea unor masuri de actiuni pentru urmatorii pasi.

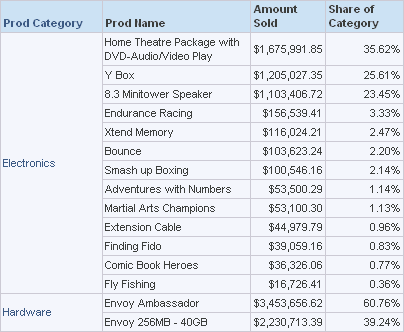
Pentru a crea masura Share of Category am folosit wizardul de calcul. Pentru a ajunge la acest wizard, am folosit “Calculation Wizard”. Am folosit urmatoarele proprietati pentru aceasta masura:



Pentru a verifica rezultatul am folosit interogarile din Oracle Answer:



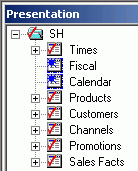
Acestea sunt doar o parte din rezultat:



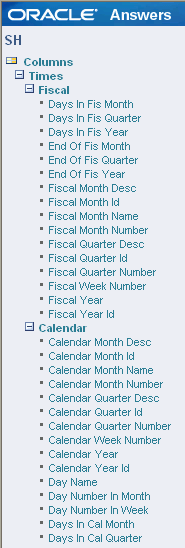
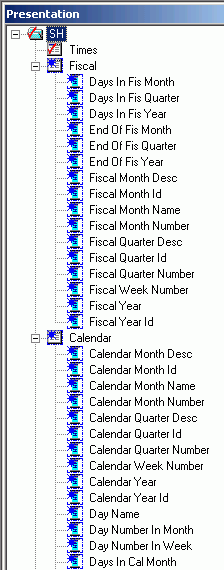
*Reorganizarea layerului de prezentare*

Catalogul de prezentare permite vizualizarea a diferite modele business pentru diverse seturi de utilizatori. Catalogul de prezentare trebuie sa fie populat cu obiecte referitoare la un singur model business si contine tabele de prezentare. Aceste tabele contin o multitudine de proprietati ce pot si modificate oricand.

In cadrul layerului de prezentare, am deschis catalogul de prezentare SH si am adaugat si alte tabele si le-am rearanjat.



Am mutat coloanele din tabela de prezentare Times in tabela de prezentare Fiscal, iar coloanele din Times in tabela Calendar. In imaginea urmatoare putem observa cum arata layerul de prezentare si rezultatul acestuia in cadrul platformei Answers:



*Initializarea de variabile si blocuri de variabile*

Se pot folosi variabile intr-un deposit de date pentru a rationaliza taskurile administrative si pentru a putea modifica continutul metadatelor in mod dinamic. O variabila isi pastreaza valoare in orice moment. Variabilele pot fi folosite asemenea unor constant in cadrul toolului de creare a expresiilor.

Exista doua tipuri de variabile:

1. variabile pentru depozitul de metadata
2. variabile pentru sesiuni.

Variabilele pentru depozitul de date detine aceeasi valoare in permanenta. Exista doua astfel de variabile: statice si dinamice. Variabilele statice contin aceeasi valoare tot timpul, nu se modifica niciodata, pe cand variabilele dinamice prezinta valori ce sunt updatate de catre datele returnate de interogari.

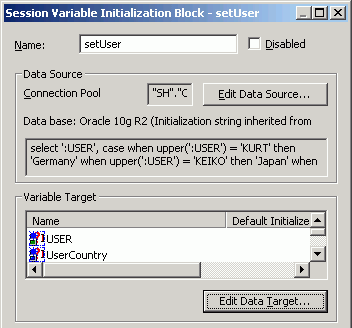
Variabilele de sesiune sunt create si le sunt asignate o valoare in momentul in care un utilizator se loggeaza in sistem. Exista doua tipuri de astfel de variabile: de sistem si non-sistem. Varibilele de sistem sunt folosite in anumite scopuri de catre serverul BI, precum autentificarea utilizatorilor. Varibilele de non-sistem sunt create de catre administrator.

*Crearea si initializarea de blocuri pentru variabile de sesiune*

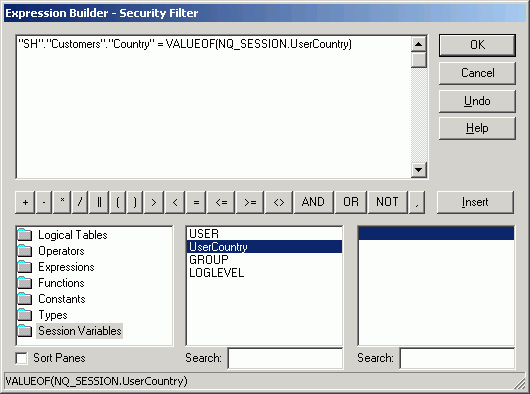
Atunci cand un utilizator incepe o sesiune, serverul BI creaza o noua instant de variabile de sesiune si le initializeaza. Spre deosebire de varibilele depozitelor de date, exista la fel de multe instante de variabile de sesiune cat si sesiuni active pe serverul BI.

Pentru a crea un bloc pentru variabilele de sesiune, se selecteaza depozitul de date din toolul de administrare, click Manage->Variables, pentru a deschide managerul de variabile. Am creat un nou bloc pentru utilizatori pe care l-am denumit setUser. Pentru acest bloc am specificat sursa de date si am initializat varibilele in felul urmator: “**select ':USER', case when upper(':USER') = 'KURT' then 'Germany' when upper(':USER') = 'KEIKO' then 'Japan' when upper(':USER')= 'CHARLES' then 'United Kingdom' when upper(':USER') = 'KAREN' then 'United States of America' end,**

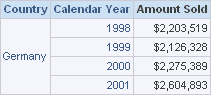
**'CountryManagers', 2 from Dual**”. Apoi am adaugat numele variabilelor in sistem : **USER, UserCountry**, **GROUP**, si **LOGLEVEL.**



Pentru testarea varibilelor am adaugat un filtru.



Apoi am folosit Answer pentru a vizualiza concret rezultatul. Am creat urmatoarea interogare, iar rezultatul de afla in tabela imediat urmatoare.



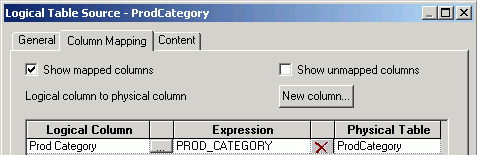
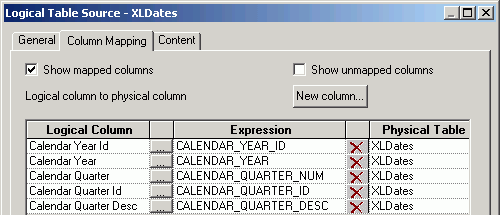
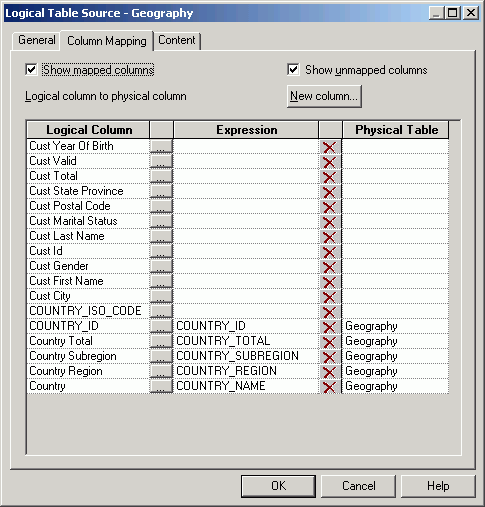
*Adaugarea de surse de date multiple*

Intr-un model business, de cele mai multe ori, datele sunt partitionate in mai multe surse de date fizice pentru un singur tabel logic. Atunci cand un tabel logic nu contine intregul set de date a unui nivel, trebuie specificat setul de date ce il contine. Idea de baza consta in oferirea unui acces eficient si unitar a mai multe surse de date pentru a satisface cerintele utilizatorului.

Baza de date a cotelor de vanzare este stocata intr-un fisier excel. Aceste date sunt inserate in layerul business pentru a putea crea masurile acestora, iar ulterior pentru a se putea face variatia de la valoarea propriu-zisa la procente.

In primul rand am creat o noua conexiune cu **Data Sources (ODBC) (Start > Programs > Administrative Tools > Data Sources (ODBC) )** speciala pentru fisierele Excel, am incarcat fisierul ce contine datele necesare si am importat datele in layerul fizic. In layerul business am adaugat

urmatoarele coloane din layerul fizic: **COUNTRY\_REGION, COUNTRY\_SUBREGION**, **COUNTRY\_NAME, COUNTRY\_ID, COUNTRY\_TOTAL, CALENDAR\_YEAR, PROD\_CATEGORY** . Maparea am realizat-o in felul urmator:

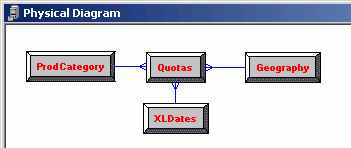


In layerul fizic am creat chei straine pentru a face urmatoarele legaturi: XLDates.CALENDAR\_QUARTER\_NUMBER = Quotas."Calendar Quarter Number" AND XLDates.CALENDAR\_YEAR = Quotas."Year" AND

XLDates.CALENDAR\_QUARTER\_DESC = Quotas."Calendar Quarter Desc" Geography.COUNTRY\_NAME = Quotas.Country AND

Geography.COUNTRY\_SUBREGION = Quotas."Country Subregion" AND Geography.COUNTRY\_REGION = Quotas."Country Region"

ProdCategory.PROD\_CATEGORY = Quotas."Prod Category".

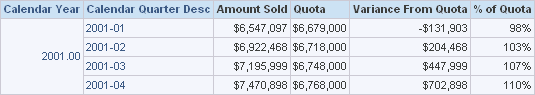


*Crearea masurilor pentru cote de vanzari*

Am adaugat coloana de cote de vanzari din layerul fizic in layerul logic, in tabela Sales Facts. La aceasta coloana am adaugat functia de agregare SUM si am folosit urmatoarea formula: **1000\*"SH"."Sales Facts"."Quota (000)".**

De asemenea, am mai creat doua noi masuri pentru cote de vanzari cu ajutorul carora putem vizualiza valoarea exprimata in mod procentual, si variatia acestora. Aceaste masuri sunt numite **% of Quota** si **Variance from Quota.**

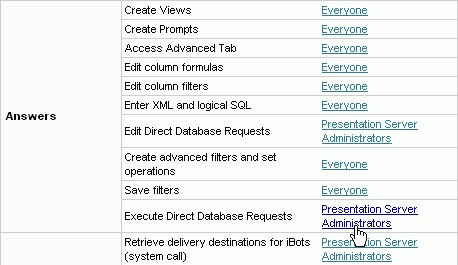
Pentru a testa aceste masuri am folosit din nou Oracle Answer. Am creat o noua interogare cu toate masurile create pentru cotele de vanzari.



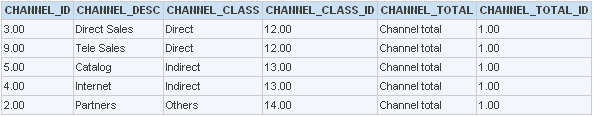
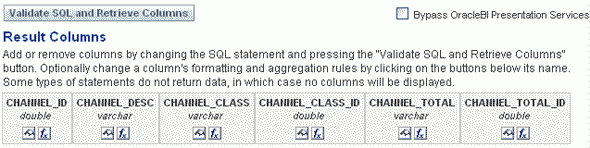
*Executarea directa a cererilor in baza de date*

Utilizatorii cu permisiunea de editare si/sau executare directa a bazei de date, pot crea cereri si/sau accesa direct baza de date . Rezultatul cererilor poate fi vizualizat si manipulate direct in Oracle Answer si apoi incorporate in Oracle BI Interactive Dashboards.

Pentru a seta privilegiile utilizatorilor, administratorul foloseste Oracle Answer. Din Administration-> Manage Priviledges, se pot seta toate privilegiile pentru fiecare tip de utilizator.



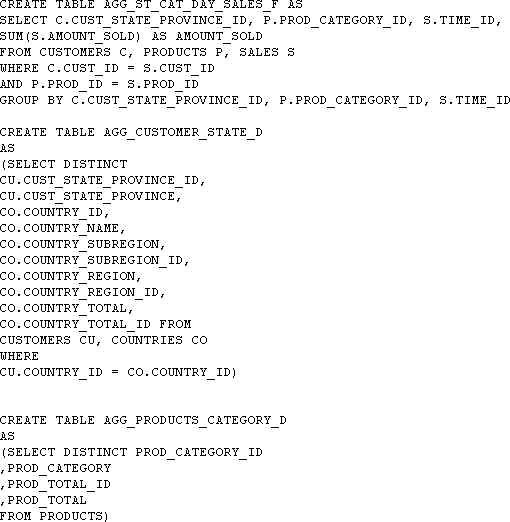
Pentru a crea cereri direct catre baza de date, utilizatorul ce detine privilegiile necesare, acceseaza Create Direct Request din Oracle Answer. Aici poate implementa interogarile dorite. Spre exemplu **SELECT \* FROM Channels,** si vor aparea coloanele cerute, si rezultatul dat de acestea.

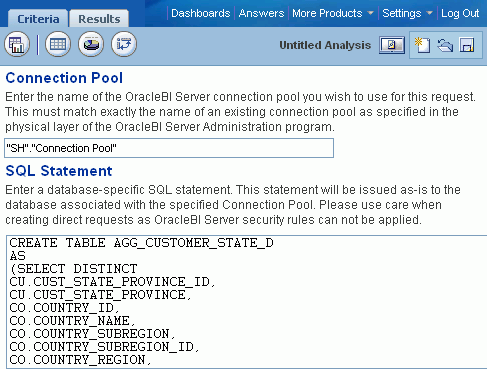


*Folosirea tabelelor agregate*

Tabelele agregate contin rezultate ce au fost calculate anterior, acestea sunt masuri ce au fost agregate asupra unui set de atribute dimensionale. Aceste tabele sunt foarte des folosite deoarece timpul de raspuns al interogarilor este foarte mic. Calculele sunt realizate anterior si stocate in tabele. Tabelele agregate contin mult mai putine randuri decat cele neagregate, iar procesarea lor devine foarte rapida.

Este foarte important ca utilizatorii sa formuleze intrebarile corecte asupra businessului, iar serverul BI va analiza cea mai rapida si mai buna solutie de a face combinatii intre tabelele agregate si de a oferii un raspuns cat mai rapid.

In primul rand, se pot crea tabele aggregate accesand direct baza de date. Spre exemplu, in acest proiect am folosit urmatorul script:



Dupa crearea tabelelor am realizat legurile intre acestea , in felul urmator:

# AGG\_CUSTOMER\_STATE\_D.CUST\_STATE\_PROVINCE\_ID = AGG\_ST\_CAT\_DAY\_SALES\_F.CUST\_STATE\_PROVINCE\_ID

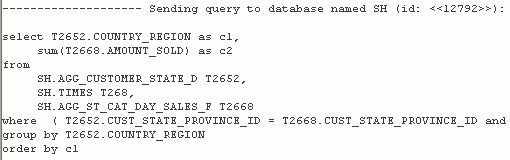
**AGG\_PRODUCTS\_CATEGORY\_D.PROD\_CATEGORY\_ID = AGG\_ST\_CAT\_DAY\_SALES\_F.PROD\_CATEGORY\_ID**

# TIMES.TIME\_ID = AGG\_ST\_CAT\_DAY\_SALES\_F.TIME\_ID

Pasul urmator consta in maparea coloanelor agregate.

|  |  |
| --- | --- |
| **Coloane logice** | **Coloane fizice** |
| Sales Facts.Amount Sold | AGG\_ST\_CAT\_DAY\_SALES\_F.AMOUNT\_SOLD |
| Customers.Cust State Province | AGG\_CUSTOMER\_STATE\_D.CUST\_STATE\_PROVINCE |
| Customers.Country | AGG\_CUSTOMER\_STATE\_D.COUNTRY\_NAME |
| Customers.Country Subregion | AGG\_CUSTOMER\_STATE\_D.COUNTRY\_SUBREGION |
| Customers.Country Region | AGG\_CUSTOMER\_STATE\_D.COUNTRY\_REGION |
| Customers.Country Total | AGG\_CUSTOMER\_STATE\_D.COUNTRY\_TOTAL |
| Products.Prod Category | AGG\_PRODUCTS\_CATEGORY\_D.PROD\_CATEGORY |
| Products.Prod Total | AGG\_PRODUCTS\_CATEGORY\_D.PROD\_TOTAL |

In final, testam tabele pentru a ne asigura ca nu exista nici o eroare. Pentru a vedea si mai bine ce interogari au fost apelate, deschidem fiesierul ViewLog . **AGG\_CUSTOMER\_STATE\_D** si **AGG\_ST\_CAT\_DAY\_SALES\_F** au fost accesate.



*Crearea masurilor de timp*

Abilitatea de a compara performantele unui business cu perioade de timp anterioare este fundamentala pentru intelegerea businessului. Solutia consta in modelarea datelor de timp in cadrul depozitului de metadate. Acest lucru permite utilizatorilor sa realizeze o singura cerere pentru rezultatul dorit. Oracle Server BI desfasoara multiple interogari in paralel pentru a ajunge la rezultat.

Oracle BI Server ofera 2 functii pentru compararea seriilor de timp:

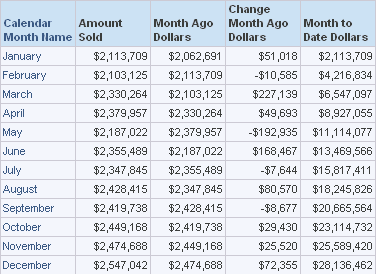
1. Ago – calculeaza valoarea agregata ca si cum o valoare de timp a fost shiftata de la timpul present. Spre exemplu, functia Ago poate realize calculul pentru vanzarile din fiecare luna a unui quarter,comparative cu vanzarile din anii anteriori ale aceluiasi quarter.
2. ToDate – functia este folosita pentru agregarea atributelor unei masuri de la inceputul unei date mentionate pana in prezent.

Am creat o noua coloana logica in tabela Sales Facts numita Month Ago Dollars. Aceasta coloana va rezulta valoarea vandura pana in ultima luna. Pentru aceasta am folosit functia Ago, expresia fiind urmatoarea:

Ago(“SH”.”Sales Facts”.”Amount Sold”, “SH”.”TimesDim”.”Month”, 1).

Am folosit si functia ToDate pentru o noua coloana logica din Sales Facts numita Month to Date Dollars. Pentru aceasta coloana am dorit sa calculez valoara totala din vanzari pe ultimul an.

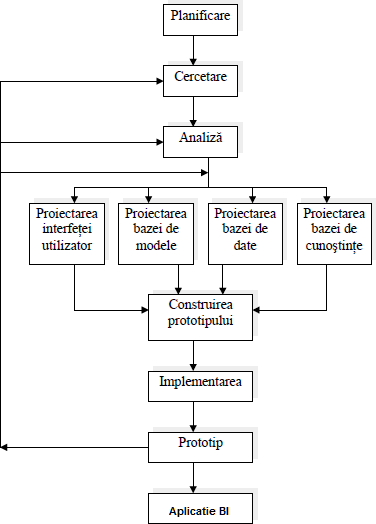
Am testat si aceste coloane in Oracle Answer, iar rezultatul este urmatorul:



# Diagrame UML

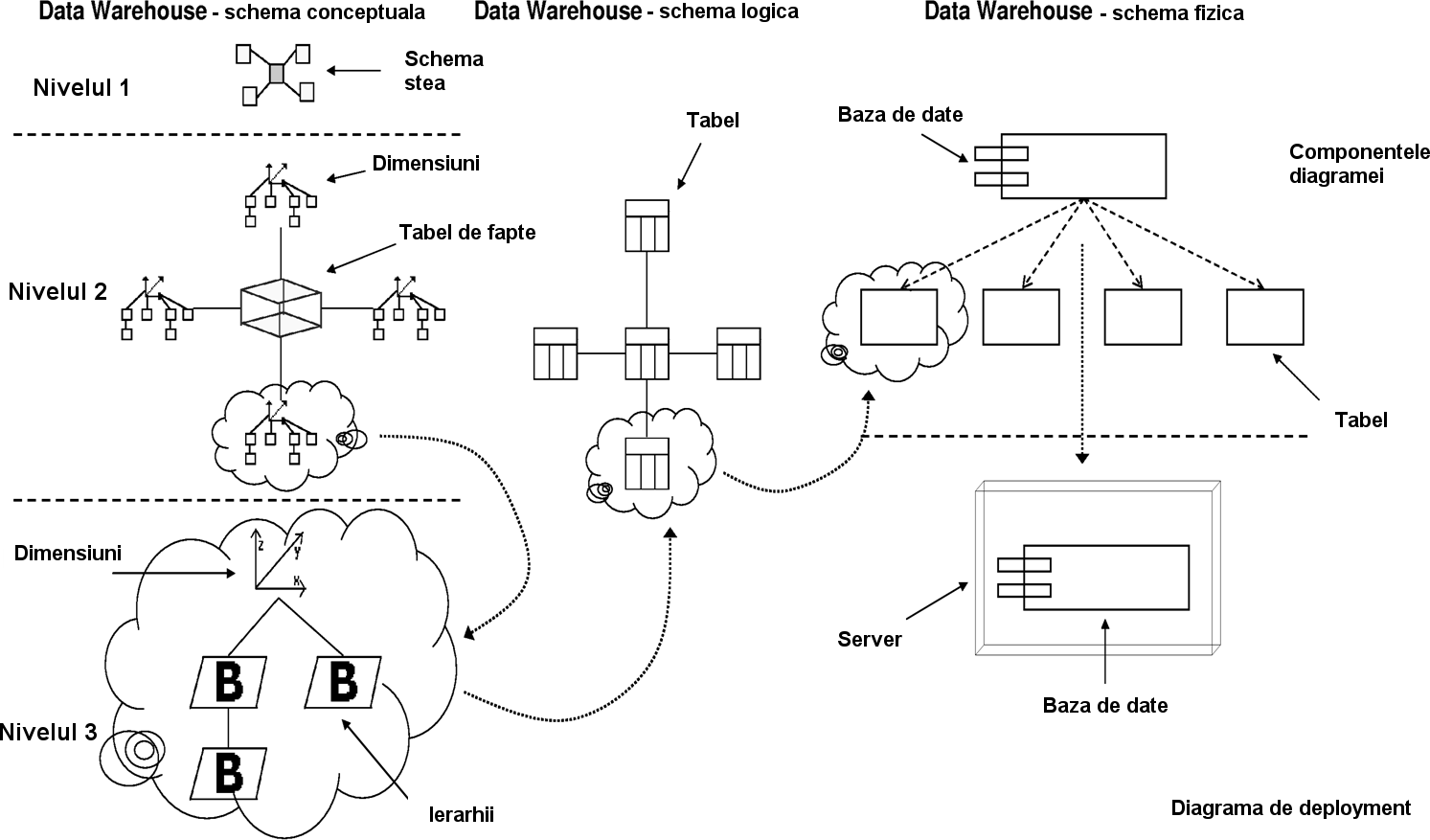
Procesul de elaborare al aplicatiei este structurat in mai multe cicluri, cate unul pentru fiecare dezvoltare de sub-sistem, si consta in urmatorii pasi:

* Proiectantul si utilizatorul definesc impreuna o sub-problema care va reprezenta inceputul dezvoltarii sub-sistemului. Aceasta sub-problema trebuie sa fie mai putin importanta ca dimensiune, delimitata clar, dar suficient de importanta ca utilitate pentru decident;
* In acelas timp, problema este analizata si un prototip este elaborat cu usurinta. Acest prototip trebuie sa includa functionalitatile principale ale sistemului;
* Sub-sistemul este utilizat si evaluat adaugand noi reprezentari, modele si structuri de control dupa fiecare ciclu de dezvoltare.



**Figura7.1.** *Dezvoltarea iterative a aplicatiei BI*

In figura urmatoare este prezentata schema fizica de implementare a arhitecturii aplicatiei de business intelligence.



**Figura7.2.** *Schema de constructie a depozitului de date*

Depozitul de date contine urmatoarele componenete la cel mai inalt nivel al arhitecturii: Administrarea rapoartelor – include toate functionalitatile utilizatorilor finali si interogarile ce pot fi create/editate.

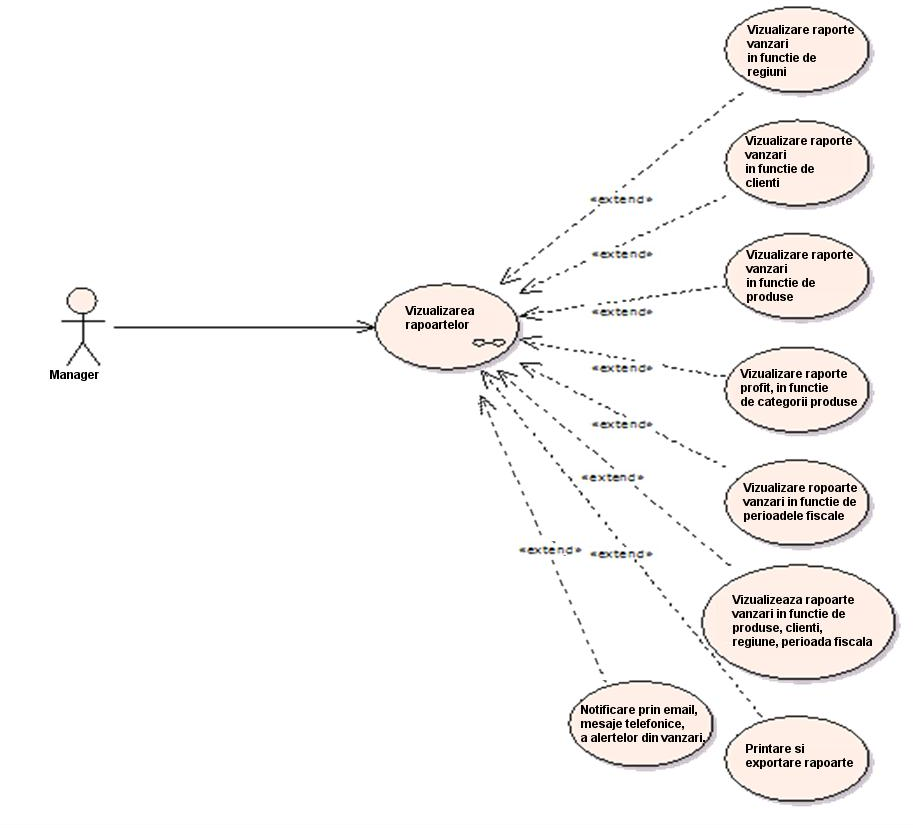
Mentenanta depozitului de date – se refera la incarcarea informatiilor, updateul datelor, din depozitul de date.

Administrare depozitului de date – include toate taskurile necesare unui administrator sa administreze depozitul de date.



**Figura 7.3.** *Diagrama de pachete a depozitului de date*

In figura este prezentata diagrama de use case-uri a aplicatiei in care sunt prezentare principalele functionalitati ale aplicatiei si accesul managerului la vizualizarea rapoartelor.



**Figura 7.4*.*** *Diagrama de use-caseuri a aplicatiei de Business Intelligence*

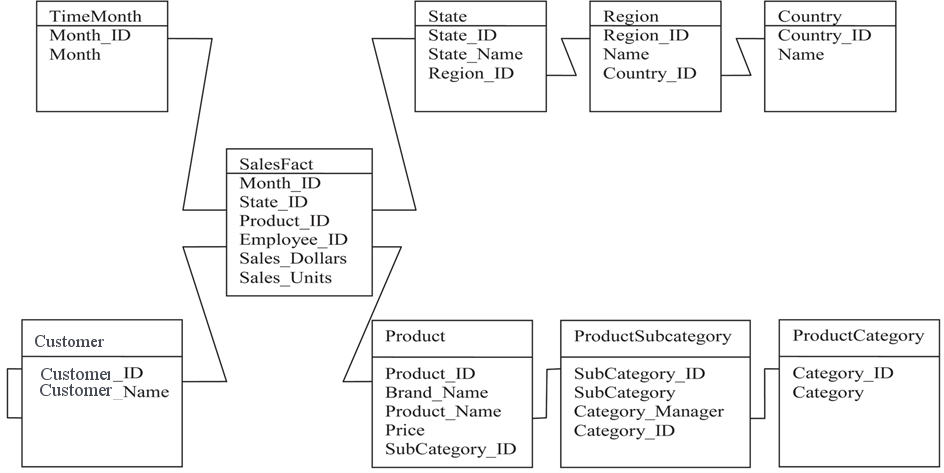
Dupa realizarea diagramei de use-caseuri ale aplicatiei se poate implementa diagram de clase cu atributele si operatiile necesare.



**Figura 7.5.** *Diagrama de clase*

Din diagram de clase putem observa o schema stea asociată cu tabelul de fapte reprezentând tranzacţii de vânzare. Tabelul de fapte este plasat în mijlocul schemei şi este legat de tabelele

dimensiune prin referinţe adecvate.



**Figura 7.6.** *Schema stea a aplicatiei*



**Figura7.7.** *Administrare utilizatori/date/rapoarte*



***Fig7.8.*** *Adaugare rapoarte*

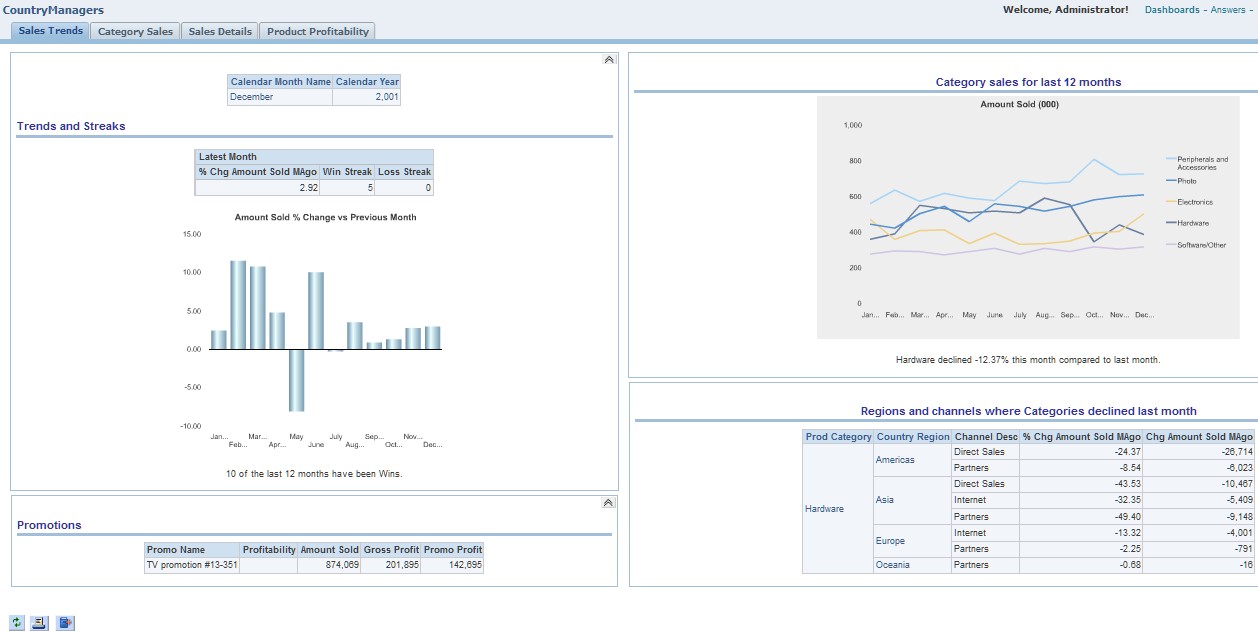
# Prezentarea aplicatiei Sales Trends

In prima pagina a aplicatiei, Sales Trends, este prezentata evolutia generala a vanzarilor pentru toate categoriile de produse. Din primul grafic, stanga sus, se pot observa lunile care au avut profitul cel mai bun si lunile in care au scazut vanzarile.

In tabelul din stanga jos, Promotions, sunt afisate produsele aflate la promotii si profitul adus de acestea.

In graficul „Category sales for last 12 months” se observa tendinta generala a vanzarilor din ultimul an pe fiecare categorie generala de produse.

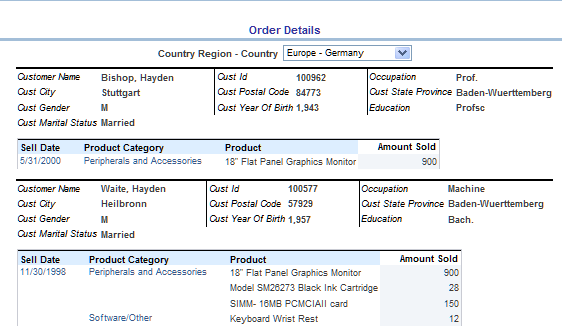
In tabelul din dreapta,jos, „Regions and channels where Categories declined last month” sunt specificate regiunile si modalitatile de vanzari pentru categoriile de produse la care vanzarile au scazut.



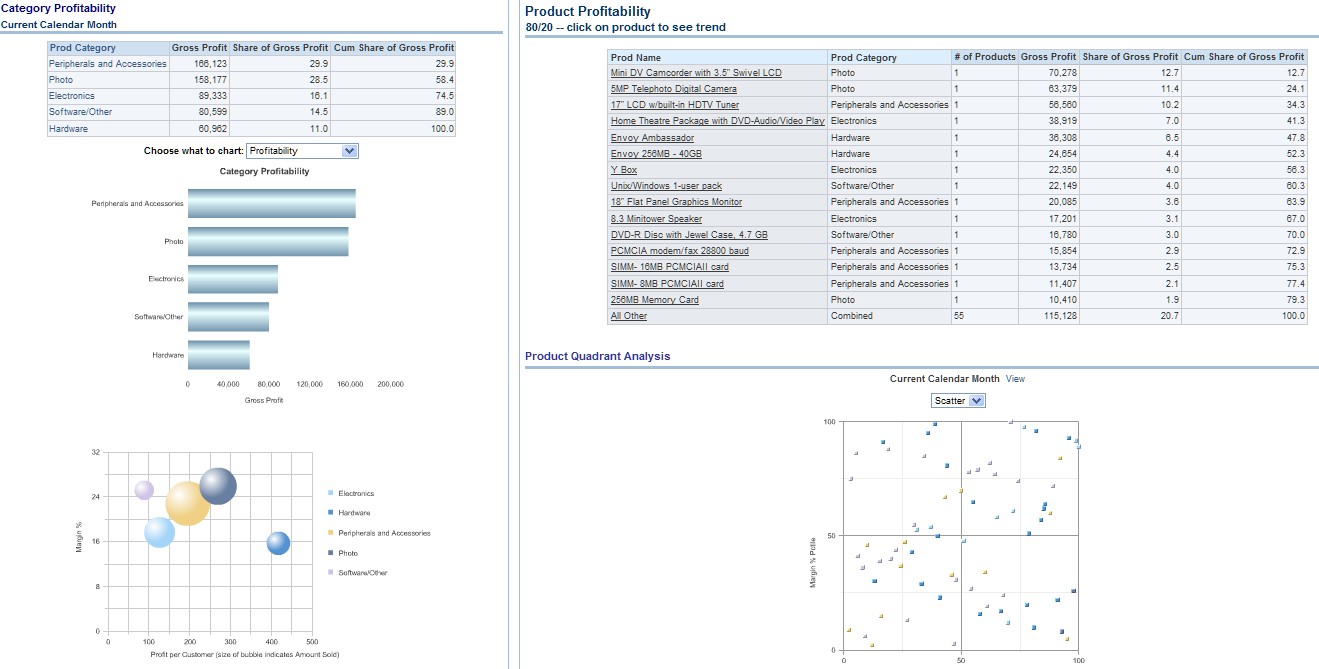
In cea de-a doua pagina „Category Sales” – Categoriile de vanzari, exista mai multe modalitati de selectie a produselor, in functie de categorie generala, de sub-categorie, de nume al produsului propiru-zis, si se poate specifica o anumita data pentru a putea vizualiza diferite masuri ale produsului selectat. Spre exemplu se pot vizualiza: protiful brut, volumul de vanzari, clasamanetul vanzarilor in fct de categoriile generale, costul unitar, procentul volumului de vanzari in comparatie cu cel aflat cu o luna in urma,etc. Toate aceste date se pot observa atat prin vizualizarea lor in tabel, cat si prin intermediul graficului.



In cadrul paginii de Sales Details, apar date amantuntite despre vanzarile realizate in diverse tari, despre clienti, produsele cumparate, etc.



In ultima pagina, „Product Profitability” – Profitul in functie de produse, se pot observa diverse tabele si grafice,charturi, pentru a putea analiza rata de profitabilitate in functie de categoriile generale de produse si de produsele propriu-zise, si evolutia lor in ultimul an.



# Avantajele oferite de Business Intelligence. Concluzii

Principala valoare a Business Intelligence este capacitatea de a transforma datele in informatii. Acest lucru permite companiilor sa elaboreze un mecanism eficient de luare a deciziilor, cu alte cuvinte de a lua decizii mai bune, mai repede. Beneficiile pot fi la fel de importante pentru conducerea companiei, in luarea unor decizii strategice, dar si pentru conducatorii de departamente sau echipe, analisti sau orice membru al echipei confruntat cu necesitatea de a lua decizii tactice.

O solutie de Business Intelligence trebuie sa integreze datele curente ale afacerii dar si date prealabile, provenind din mai multe programe si aplicatii si sa le consolideze intr-o singura baza de date optimizata pentru regasirea si analiza informatiei. Pentru a asigura ca utilizatorii vor beneficia de solutie pe o perioada lunga de timp, implementarea de Business Intelligence trebuie sa ofere posibilitatea de a efectua analize complexe in mod flexibil, fara o curba de invatare prea abrupta.

# Avantaje pentru procese

*Business Intelligence asigura o intelegere asupra bunastarii afacerii.*

* Promoveaza responsabilitatea
* Ofera managementului o viziune completa asupra afacerii
* Permite corectii asupra strategiilor de dezvoltare in timp
* Asigura o raportare si analiza consistente in intreaga companie
* Furnizeaza aplicatii web simple, flexibile pentru o analiza elaborata

# Avantaje pentru relatii

*Business Intelligence imbunatateste atentia acordata clientilor.*

* Intelegerea actiunilor clientilor
  + Asigurarea consistentei in relatiile cu clientii imbunatateste loialitatea acestora
  + Ajuta la determinarea portofoliului de produse care asigura profitul maxim
  + Aliniaza tendintele in vanzari sau ciclurile de cumparare ale clientilor cu promotiile si campanile de marketing
* Reduce semnificativ durata de lansare a unui produs
* Creste valoarea adaugata pentru client

# Avantaje pentru oameni

*Business Intelligence creste productivitatea.*

* Resursele umane sunt alocate mai eficient
* Permite analize realizate chiar de catre beneficiari
  + Raportare la nivel de management
  + Analiza in timp real
  + Se folosesc unelte cu care utilizatorii sunt familiari
* Imbunatateste curba de invatare atat pentru dezvoltatori cat si pentru utilizator

# Beneficii pentru IT

*Business Intelligence maximizeaza profitul din investitiile in IT.*

* Tehnologie facila la nivel de companie
* Consolideaza “muntii de date” existenti
* Ofera programe de raportare si unelte de analiza eficiente chiar in zone non-IT
  + Se elimina gatuirile in fluxul informatiei
  + Se dezvolta rapoartele mai repede
* Reduce costurile totale (TCO)
  + Simplificarea administrarii
  + Utilizarea de unelte si programe existente si familiare
  + Cresterea capitalizarii infrastructurii existente
  + Intrebuintarea expertizei profesionale existente si recrutarea usurata prin existenta fortei de munca calificate

Analiza datelor inteligente a fost mereu importantă deoarece prin această analiză se generează înţelegerea. Importanţa înţelegerii „concurenţei” a fost recunoscută de strategul militar Sun Tzu încă din sec. VI I.H.. Această necesitate de a-şi cunoaşte adversarul era afirmată prin folosirea spionilor: „*cunoaşterea aprofundată nu poate fi obţinută prin fantome şi spirite, dedusă din fenomene, sau prin raportare la standardele Raiului, ci trebuie dobândită de la oameni deoarece reprezintă cunoaştere adevăratei situaţii a duşmanului*”. („Advance knowledge cannot be gained from ghosts and spirits, inferred from phenomena, or projected from the measures of Heaven, but must be gained from men for it is the knowledge of the enemy's true situation”). Business intelligence este asemenea unei poveşti fără sfârşit, fiind un domeniu aplicabil în audit, în managementul riscurilor, diplomaţie sau activităţi de lobby. Am văzut câteva din tendinţele în acest domeniu. Iar alte tendinţe abia se conturează. Aşa ar fi situaţia fuziunii dintre business intelligence şi inteligenţa artificială, fuziune ce ar duce la naşterea unui nou concept, artificial business intelligence (ABI). În astfel de condiţii trebuie să admitem că aceste tehnologii se află la începutul unei lungi călătorii, într-o lume în care cheia succesului stă în capacitatea de a lua decizii mai bune şi într-un timp mult mai scurt decât competiţia. De altfel, viaţa unei companii depinde din ce în ce mai mult de astfel de decizii, fapt ce face imposibil de negat beneficiul adus de business intelligence.

# Dezvoltari ulterioare

Exista numeroase imbunatatiri care pot fi aduse aplicatiei. Una dintre ele ar fi posibilitatea utilizatorilor de a cauta un anumit raport, dupa diverse criterii: denumire, dimensiuni si masuri folosite, etc. Aceasta optiune ar reduce timpul de cautare al unui raport existent, in conditiile in care un utilizator ar avea drepturi pe foarte multe rapoarte.

O alta imbunatatire ar fii conectarea la un provider de servicii, oferind capacitatea de conectare la doua surse de date. Folosind Data Source Manager, implimcit si Smart View, se pot manipula datele analitice prin intermediul utilitatilor MS Office, in special MS Excel, MS Word si MS PowerPoint. Smart View Data Source Manager este o utilitate multifunctionalata ce permite conectarea la un server central printr-un URL, acceseaza sursele de date si le afiseaza in cadrul interfatei.

In cadrul platformei de Business Intelligence se poate integra si utilitatea de urmarire a performantelor sistemului (Usage Tracking). Aceasta functionalitate este folosita pentru monitorizarea sistemului si urmarirea performantelor interogarilor, timpul de raspuns, fregventa de accesare,etc. Usage traking detine un catalog de tablouri de board precalculate pentru a oferii informatii despre analizele realizate.

In viitorul apropiat, Oracle va lansa o noua versiune HYPERION cu platforma de Business Intelligence integrata. Acest lucra va putea oferi o putere mare de analiza a datelor si de planificare a acestora.

# Bibliografie

* Almeida, M. şi alţii, *Getting Started with DataWarehouse, and Business Intelligence*, IBMCorporation, 1999.
* Bernstein, A. şi alţii, *Business Intelligence: The Next Frontier for Information SystemsResearch?*, [www.technologyevaluation.com.](http://www.technologyevaluation.com/)
* Gilad, B., Gilad, T., *The Business Intelligence System, a New Tool for Competitive Advantage*, AMACOM.
* Giovinazzo, W., *A., Internet - Enabled Business Intelligence*, Prentice Hall PTR, 2002.
* Hyperion Solutions Corporation, *Transforming Business Intelligence Into Business Performance Management: Competitive Advantage in the Information Economy*, [http://www.technologyevaluation.com,](http://www.technologyevaluation.com/) November 12, 2005, Vol.1, No.7.
* Moss, L. T., Atre, S., *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*, Addison Wesley, 2003.
* Raisinghani, M., *Business Intelligence in the Digital Economy* – *Opportunities, Limitations and Risks*, Idea Group INC., 2004.
* Silaşi, G., Stanc, A., Sava, V., *Inteligenţa Economică, armă a războiului global*, Ed. Orizonturi Universitare, 2000.
* Waltz, E., *Knowledge Management in the Intelligence Enterprise*, Artech House, 2003
* Zaman, Mukhles *Predictive Analytics: the Future of Business Intelligence*, [www.technologyevaluation.com,](http://www.technologyevaluation.com/) November 8, 2005.
* Zaman, M., *Business Intelligence: Its Ins and Outs*, [www.technologyevaluation.com](http://www.technologyevaluation.com/) , January 10, 2005.
* \*\*\*, *Corporate performance management: One Step Further Than Business Intelligence*, [www.sap.info,](http://www.sap.info/) 31 octombrie 2005.
* \*\*\*, *Interception Capabilities 2000*, [www.iptvreports.mcmail.com.](http://www.iptvreports.mcmail.com/)
* \*\*\*, Wikipedia – The free encyclopedia, [http://en.wikipedia.org.](http://en.wikipedia.org/)
* Benchimol, Guy; Levine, Pierre. Sisteme expert în întreprindere. Bucureşti: Editura Tehnică, 1998;
* Candea, Rodica M.; Candea, Dan. Comunicarea managerială aplicată. Bucureşti: Expert, 1998;
* Candea, Rodica M.; Candea, Dan. Comunicarea managerială: concepte, deprinderi, strategie. Bucureşti: Expert, 1996;
* Choffray, Jean-Marie. Sisteme inteligente de management: diagnostic, analiza şi asistenţa deciziei. Bucureşti: Ştiinţă şi Tehnică, 1997;
* Cochina, Ion; Nicolescu, Ovidiu, cond. st. Managementul firmei asistat de calculator:

\*teză de doctorat+. Bucureşti: ASE, 1998;

* Coroescu, Tiberiu. Sisteme informatice pentru management. Bucureşti: Lumina Lex,

2002;

* El Habashi, Abdel Halim; Zaharie, Dorin, cond. st. The model of a decision support

system for the strategic high level management: \*teză de doctorat+. Bucureşti: ASE, 2005;

* + Filip, Florin Gh. Decizie asistată de calculator: decizii, decidenţi, metode şi instrumente informatice asociate. Ed. a 2-a rev. şi ad. Bucureşti: Editura Tehnică, 2005;
  + Gaf-Deac, Maria. Teoria deciziei. Bucureşti: Editura Fundatiei România de Mâine,

2004;

* Georgescu, Vasile. Asistarea deciziei manageriale, între procedural şi logic: teză de

doctorat. Bucureşti: ASE, 1996;

* + Larissa Terpeluk Moss,Shaku Atre. Business intelligence roadmap: the complete project lifecycle for decision ...
  + Carlo Vercellis. Business Intelligence: Data Mining and Optimization for Decision

Making

* Mark Whitehorn. Business intelligence: the IBM solution
* Rajiv Sabherwal,Irma Becerra-Fernandez. Business Intelligence
  + Cindi Howson. Successful Business Intelligence: Secrets to Making BI a Killer App Zaharia, Răzvan; Cruceru, Anca, *Gestiunea forţelor de vânzare,* Editura
  + Uranus, Bucureşti, 2002;

Rackham, Neil, *Succesul în vânzări. Strategia SPIN,* Editura Teora, Bucureşti,

* + 2001;
  + Choffray, Jean-Marie. Sisteme inteligente de management: diagnostic, analiza şi asistenţa deciziei. Bucureşti: Ştiinţă şi Tehnică, 1997;
  + Cochina, Ion; Nicolescu, Ovidiu, cond. st. Managementul firmei asistat de calculator:

\*teză de doctorat+. Bucureşti: ASE, 1998;

* + Coroescu, Tiberiu. Sisteme informatice pentru management. Bucureşti: Lumina Lex,

2002;

* El Habashi, Abdel Halim; Zaharie, Dorin, cond. st. The model of a decision support

system for the strategic high level management: \*teză de doctorat+. Bucureşti: ASE, 2005;

* + Filip, Florin Gh. Decizie asistată de calculator: decizii, decidenţi, metode şi instrumente informatice asociate. Ed. a 2-a rev. şi ad. Bucureşti: Editura Tehnică, 2005;
  + Gaf-Deac, Maria. Teoria deciziei. Bucureşti: Editura Fundatiei România de Mâine,

2004;

* Georgescu, Vasile. Asistarea deciziei manageriale, între procedural şi logic: teză de

doctorat. Bucureşti: ASE, 1996;