Multidimensional Data Analysis Using OLAP Technology (2)

Asist. Gianina RIZESCU
Catedra de Contabilitate şi Informatică Economică, Universitatea "Dunărea de Jos" Galaţi

In this paper we present the main steps in creating an application using OLAP technology. The main goals of this application are: the quality analysis of the teaching process through the results obtained by the students of a university and the structure and dynamic analysis of the students in a university ("Dunărea de Jos" of Galaţi was taken as example). Thus, after a brief introduction, we will present de general architecture of the application followed by the description of the analysis, design, and populating stages of the data warehouse and also by the multidimensional data analysis using decision cubes.
Keywords: *OLAP, OLTP, data warehouse, data mart, decision cub.*

În articolul anterior au fost prezentate etapele de analiză şi proiectare a depozitului de date. In continuare vor fi prezentate popularea depozitului de date şi analiza datelor utilizând cuburile de decizie.

III. Popularea depozitului de date

Popularea depozitului de date se face cu date preluate din sistemele tranzacţionale care trebuie supuse unor procese de transformare pentru a se încadra în structura prefigurată a depozitului. Aceasta etapă va fi repetată periodic pentru a adăuga datele noi. SQL Server 7.0 oferă un instrument puternic pentru transformarea datelor, Data Transformation Services (DTS), care face această etapă mult mai uşoară. Încărcarea datelor în depozitul de date se face în câteva etape: extragerea datelor din sursele de date, transformarea datelor şi popularea depozitului.

III.1 Planificarea transformărilor

Transformarea datelor implică câteva procese. Aceste procese trebuie proiectate pentru a asigura validitatea şi integritatea, utilitatea şi în anumite cazuri agregarea şi calcularea unor noi valori bazate pe datele sursă. Aceste procese sunt clasificate şi cunoscute ca: validarea, curăţirea şi transformarea datelor.

Validarea datelor asigură integritatea şi validitatea. De exemplu, asigură faptul că oraşul menţionat se află în judeţul sau ţara corespunzătoare. Validarea asigură de asemenea integritatea referenţială a bazei de date în termeni de chei primare şi străine.

Curăţirea datelor analizează şi integrează datele provenite din surse diferite pentru ca în depozitul de date să ajungă o versiune unitară a acestora. Prin curăţire se elimină discrepanţele sau conflictele între diferitele surse de date.

Transformarea datelor este procesul prin care datele extrase din sursele de date sunt convertite in formatul corespunzător încărcării lor în depozitul de date. Metadatele sunt în mod obişnuit utilizate pentru a memora caracteristicile mapărilor utilizate pentru a transforma datele sursă în Microsoft Repository. Metadatele definesc orice schimbare care este cerută înaintea încărcării datelor în depozitul de date. Transformarea datelor va înlătura anomaliile din sursele de date şi va oferi date corespunzătoare pentru depozitul de date. Transformarea datelor poate interveni la nivel de înregistrare sau la nivel de câmp. Există trei modalităţi de bază utilizate în transformarea datelor: transformare structurală, transformare de conţinut, transformare funcţională.

Transformarea structurală schimbă structura înregistrărilor sursă pentru a fi similară cu cea din baza de date destinaţie. Această modalitate transformă datele la nivel de înregistrare. Ceea ce înseamnă că întreaga înregistrare este transformată pentru ca structura acesteia să fie similară cu structura înregistrării din baza de date destinaţie.

Transformarea de conţinut schimbă valorile înregistrărilor. Acest tip de transformare se aplică la nivel de câmp şi schimbă valorile

câmpurilor utilizând algoritmi sau tabele pentru transformarea datelor (Lookup Table).

Transformarea funcțională creează noi valori pentru datele destinație pornind de la datele sursă. Această tehnică transformă datele la nivel de câmp. Aceste transformări pot fi folosite şi în agregarea sau îmbogățirea datelor. Agregarea reprezintă calcularea unor valori derivate cum ar fi: totaluri sau medii ale valorilor câmpurilor din înregistrări diferite. Îmbogățirea combină două sau mai multe valori şi creează unul sau mai multe noi atribute din una sau mai multe înregistrări care pot proveni din aceeaşi sursă sau surse diferite.

Microsoft SQL Server oferă câteva instrumente pentru toate operațiile prezentate mai sus. Aceste instrumente sunt: Data Transformation Services, Import/Export Wizard, programe sau script-uri ActiveX, DTS Lookup.

Transformarea datelor utilizând serviciul DTS implică în general planificarea şi proiectarea transformărilor precum şi crearea şi executarea pachetului DTS ce va efectua aceste transformări în mod automat. DTS Designer permite definirea unor transformări complexe cu multe task-uri bazate pe constrângeri de precedență.

III.2 Pachetul DTS pentru popularea automată a depozitului de date

Pachetul realizat pentru popularea automată a depozitului de date are următoarele caracteristici:

✓ Are ca sursă o singură bază de date şi anume baza de date SQL Server, *Disertatie.mdf*

✓ Destinația o reprezintă depozitul de date care este tot o bază de date de tip SQL Server, *Diseratiemart.mdf*

✓ Extrage datele din baza de date sursă, le transformă şi le încarcă în baza de date *destinație*

Pentru realizarea transformărilor, pachetul DTS creat are următoarea structură:

Conexiuni de date. Pentru realizarea transformărilor în pachet au fost necesare următoarele conexiuni:

✓ Conexiuni de date destinație: sunt cele trei conexiuni *disertație OLAP* care fac conexiunea cu depozitul de date *diseratiemart.mdf*. Prima conexiune începând de la stânga a fost creată pentru realizarea transformărilor necesare popularii tabelelor de dimensiunii, a doua pentru popularea tabelei de fapte *rezultate* iar a treia pentru popularea tabelei de fapte *structură_studenți*.

✓ Conexiuni de date sursă realizează conectarea la baza de date sursă, *disertatie.mdf*. Acestea sunt:

- DW_tipexamen – creată pentru sprijinirea populării tabelei de dimensiuni *tip_examen*
- DW_profesor - creată pentru sprijinirea populării tabelei de dimensiuni profesor
- DW_timp - creată pentru sprijinirea populării tabelei de dimensiuni *timp*
- DW_disciplină - creată pentru sprijinirea populării tabelei de dimensiuni *disciplina*
- DW_student - creată pentru sprijinirea populării tabelei de dimensiuni *student*
- DW_zonă - creată pentru sprijinirea populării tabelei de dimensiuni *zona_geografică*
- *DW_structura studenti* - creată pentru sprijinirea populării tabelei de dimensiuni *structura student*
- *DW_timp_stud* - creată pentru sprijinirea populării tabelei de dimensiuni *timp_stud*
- DW_rezultate - creată pentru sprijinirea populării tabelei de fapte *rezultate*
- DW_structură_studenți - creată pentru sprijinirea populării tabelei de fapte structură_studenți

Taskuri. În cadrul pachetului a fost folosit un singur task *actualizarea tabelelor* de tipul Execute SQL Task, care actualizează (şterge înregistrările) tabelele de dimensiuni şi tabelele de fapte înaintea începerii procesului de transformare şi încărcare a depozitului de date.

Fluxuri (Workflows). Au fost necesare următoarele fluxuri:

✓ Fluxuri pentru transformarea datelor (Data Transformation).

- Fluxul DW_tipexamen - disertatieOALP – extrage datele necesare, le transformă şi le încarcă în tabela de dimensiuni *tip_examen*
- DW_profesor - extrage datele necesare, le transformă şi le încarcă în tabela de dimensiuni *profesor*
- DW_timp - extrage datele necesare, le

transformă şi le încarcă în tabela de dimensiuni *timp*

- DW_disciplină - extrage datele necesare, le transformă şi le încarcă în tabela de dimensiuni d*isciplina*
- DW_student - extrage datele necesare, le transformă şi le încarcă în tabela de dimensiuni *student*
- DW_zonă - extrage datele necesare, le transformă şi le încarcă în tabela de dimensiuni *zona_geografică*
- DW_rezultate - extrage datele necesare, le transformă şi le încarcă în tabela de fapte *rezultate*
- DW_structură_studenţi - extrage datele necesare, le transformă şi le încarcă în tabela de fapte *structură_studenţi*

✓ **Constrângeri de precedenţă** au fost necesare următoarele constrângeri:

- Constrângerile de precedenţă On Succes între task-ul *actualizarea tabelelor* şi conexiunile de date sursă pentru ca procesul de încărcare a tabelelor de dimensiuni sa nu înceapă decât după actualizarea cu succes a acestora.
- Constrângerea de precedenţă On succes între conexiunea destinaţie *disertaţie OLAP* şi conexiunea sursă *DW_rezultate* pentru ca procesul de populare a tabelei de fapte sa nu înceapă până când nu se încheie cu succes popularea tabelelor de dimensiuni.
- Constrângerea de precedenţă On Succes între conexiunea destinaţie disertaţie OLAP şi conexiunea sursă DW_structură_studenţi pentru ca popularea tabelei de fapte *structură_studenţi* să nu înceapă până când popularea tabelei de fapte *rezultate* nu se încheie cu succes.

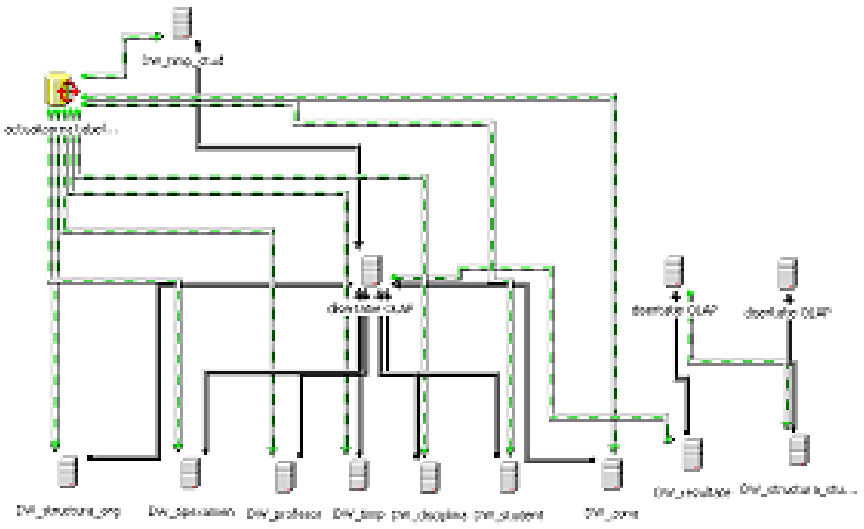

Fig.1. Pachetul DTS pentru popularea depozitului de date

IV Analiza datelor utilizând cuburile de decizie

IV.1 Proiectarea cuburilor: dimensiuni, măsuri, fapte

Proiectarea unui cub presupune identificarea şi proiectarea pe rând a dimensiunilor, faptelor şi a măsurilor.

Dimensiunile sunt elemente de bază în definirea unui cub. 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 student, 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.

Există două tipuri de dimensiuni care trebuie analizate atunci când se definesc acestea, şi anume: dimensiuni private (Private Dimensions) şi dimensiuni publice (Shared Dimensions). Dimensiunile Publice sunt

create independent de orice cub iar dimensiunile private sunt create odată cu cubul de decizie şi sunt salvate în librăria cubului de decizie respectiv

Se optează pentru dimensiunile private în cazul în care dimensiunea respectivă implementează o logică de vizualizare a datelor valabilă doar în contextul cubului respectiv. În toate celelalte cazuri este bine să se definească dimensiuni publice care să asigure viziuni unitare şi consistente asupra datelor.

Măsurile sunt valori numerice care dau măsura analizelor efectuate asupra datelor. Obligatoriu orice tabelă de fapte din depozitul de date trebuie să conţină măcar o măsură. Asupra măsurilor din tabela de fapte se pot aplica următoarele funcţii de agregare: sum (efectuează suma elementelor), min (calculează valoarea minimă), max (calculează valoarea maximă), count (calculează numărul de elemente), avg (care calculează media aritmetică).

Aceste măsuri se pot defini atât pentru vizualizarea lor în cub cât şi pentru obţinerea unor membri calculaţi.

Membri calculaţi pot fi definiţi pe baza datelor stocate deja în cub. Aceasta înseamnă că valorile membrilor calculaţi sunt calculate la cerere ele nefiind salvate fizic în cubul de decizie.

Agregarea datelor. Agregările sunt date precalculate care permit scăderea timpului de răspuns la interogări. Prin memorarea valorilor precalculate, server-ul nu trebuie să acceseze toate datele sursă necesare efectuării calculului respectiv. Utilizarea agregărilor este vitală pentru scăderea timpului de răspuns al sistemelor OLAP. În sistemele OLAP cuburile sunt cele care memorează datele agregate. Dimensiunile sunt coordonatele după care se interoghează cubul de decizie. Agregările sunt memorate, stocate la intersecţia dintre dimensiuni. Fiecare intersecţie, denumită celulă stochează o valoare agregată.

In mod normal agregările îmbunătăţesc performanţele unui cub cu toate acestea la definirea lor trebuie avuţi în vedere următorii factori:

- ✓ Spaţiul pe disc. De la un anumit punct spaţiul pe disc ocupat suplimentar de către agregări nu se justifică în raport cu performanţele obţinute. De aceea se determina şi se creează mai întâi cele mai importante agregări şi apoi în timp se adaugă noi agregări care devin necesare sau se pot elimina cele care sunt utilizate mai rar
- ✓ Timpul necesar pentru procesarea cubului. Cu cât mai multe agregări sunt adăugate cubului cu atât timpul necesar procesării acestuia va fi mai mare
- ✓ „Explozia datelor” (Data Explosion). Explozia volumului de date este o problemă atunci când se definesc prea multe agregări. Dacă toate agregările posibile ar fi calculate, spaţiul necesar stocării cubului ar fi enorm. OLAP Services dispune de un algoritm avansat pentru determinarea optimului între performanţă şi spaţiu necesar pentru stocare.

Ţinând cont de consideraţiile prezentate anterior, pentru analiza calităţii procesului de învăţământ s-a creat cubul de analiză *rezultate* care are următoarele caracteristici:

Dimensiuni: pentru proiectarea cubului de decizie au fost identificate şi definite următoarele dimensiuni:

Profesor cu următoarea ierarhie: grad didactic, sex, nume profesor;

Disciplină cu ierarhia: denumire categorie, denumire disciplină

Structură organizatorică: facultate, secţie, an, grupa, subgrupa

Examen: tip examen, examen

Timp: an, sesiune

Regim: regim

Student: sex, stare civilă, student

Zona geografică: ţara, regiune, judeţ, oraş

Ca **măsuri** au fost definite următoarele:

- ✓ *Nota, note peste 5, note peste 8, credite* care sunt preluate din depozitul de date
- ✓ *nr note şi suma* utilizate pentru obţinerea membrului calculat *media*
- ✓ *minim, maxim* utilizate pentru obţinerea notelor minimă, respectiv maximă
- ✓ *note5 note8* utilizate pentru obţinerea membrilor calculaţi *promovabilitate*, respectiv *rezultate peste 8*.

Ca schemă conceptuală a depozitului de date

a fost ales modelul de tip stea aşa după cum se poate vedea în figura 2.

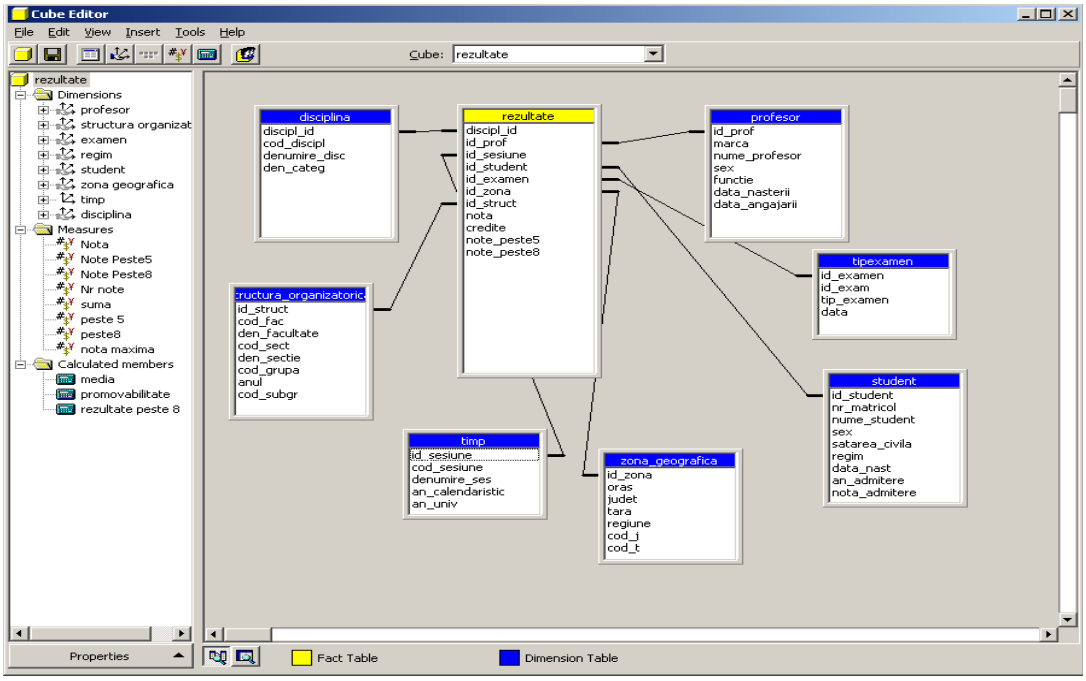

Fig 2. Schema conceptuală a cubului de date *rezultate*

Pentru analiza numărului şi structurii studenţilor universităţii s-a creat cubul de analiză *structură studenţi* care este definit de următoarele elemente:

Dimensiuni: pentru proiectarea cubului de decizie au fost identificate şi definite următoarele dimensiuni:

✓ *Structură organizatorică cu următoarea ierarhie: facultate, secţie, an, grupa, subgrupa*

✓ *Regim:* regim

✓ *Zona geografic* cu ierarhia*:* ţara, regiune, judeţ, oraş

✓ *An:* anul calendaristic şi anul universitar

✓ *Stare civilă:* starea civilă

✓ *Sex:* sex

Ca **măsuri** au fost definite următoarele:

✓ *Total studenţi* calculează numărul total de studenţi

Membri calculaţi:

✓ *Pondere fete* care calculează ponderea fetelor în numărul total de studenţi

✓ *Pondere băieţi* care calculează ponderea băieţilor în numărul total de studenţi

✓ *Pondere buget* care calculează ponderea studenţilor la buget în numărul total de studenţi

✓ *Pondere taxă* care calculează ponderea studenţilor cu taxă în numărul total de studenţi

Ca schemă conceptuală a depozitului de date a fost ales de asemenea modelul de tip stea aşa după cum se poate vedea în figura 3.

IV.2 Alegerea unei metode de stocare: MOLAP, ROLAP sau HOLAP

Datele din cuburile de analiză pot fi stocate în trei modalităţi: multidimensional, relaţional sau hibrid. Modalitatea de stocarea trebuie aleasă cu grijă deoarece fiecare dintre acestea prezintă atât avantaje cât şi dezavantaje.

MOLAP (Multidimensional OLAP) stochează atât datele cât şi agregările în structuri multidimensionale, numite cuburi. Aceste structuri sunt deci stocate în afara depozitului de date. Datele stocate în cuburile OLAP au următoarele caracteristici:

✓ Toate măsurile sunt memorate în aceeaşi înregistrare ceea ce reduce timpul de acces la date

✓ Câmpurile cu valori nule nu sunt stocate, se elimină astfel problema „împrăştierii” (sparsity) datelor

✓ Row segments pot avea dimensiuni de 64 KB
✓ Foloseşte indecşi speciali, de tip bitmap ce îmbunătăţesc performanţele cubului

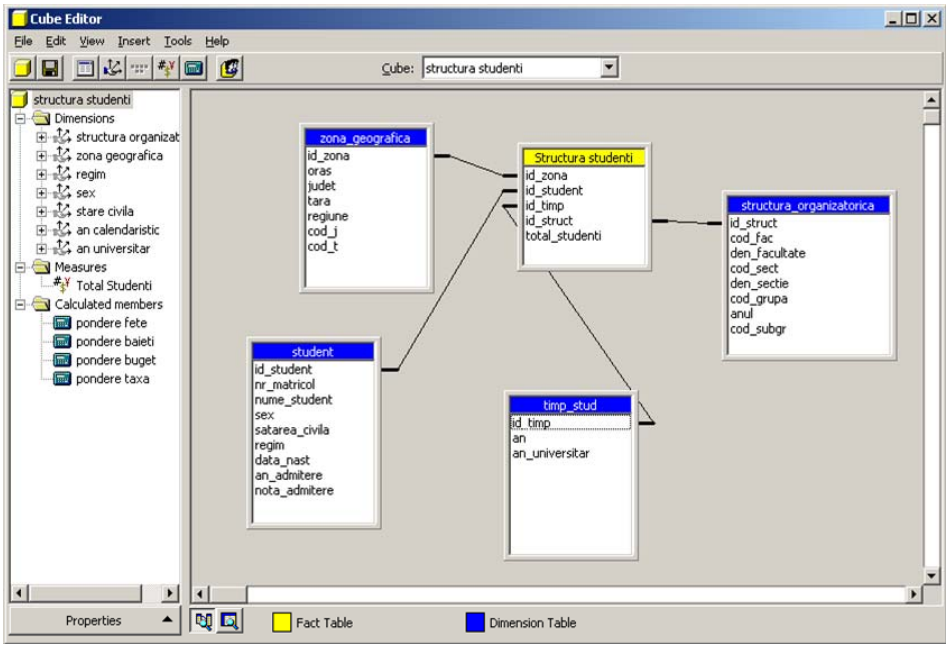

Fig.3. Schema conceptuală a cubului de date *statistici studenţi*

Stocarea cuburilor în structuri MOLAP este cea mai potrivită pentru cele care sunt interogate frecvent şi care necesită un timp de răspuns redus

ROLAP (Relaţional OLAP) stochează datele şi agregările în tabele relaţionale. Aceste tabele sunt memorate în aceeaşi bază de date în care sunt memorate şi tabelele depozitului de date. Deoarece datele sunt stocare în depozitul de date, cuburile ROLAP nu necesită un alt spaţiu de stocare. Având in vedere această modalitate de stocare, se pot utiliza şi comenzi standard Transact-SQL în afara instrumentului OLAP Services, pentru interogarea cubului deşi, numele tabelelor şi a coloanelor sunt greu de utilizat fiind generate automat de către sistem.

Structura datelor ROLAP este formată din tabele şi indecşi. Dimensiunile sunt stocate în tabele iar măsurile în coloane. Fiecare nivel al unei dimensiuni este indexat.

Nu se poate utiliza această modalitate de stocare a datelor dacă sursa de date pentru cub este de tipul OLE DB Provider for OLAP Services (adică este multidimensională) deoarece ROLAP necesită o bază de date relaţională ca sursă.

Stocarea datelor cubului într-o structură ROLAP se recomandă pentru cuburile a căror date nu sunt interogate frecvent. De exemplu dacă 80% dintre utilizatori solicită datele doar din ultimul an în interogările lor, datele istorice pot fi mutate într-o structură ROLAP pentru a micşora spaţiul general de stocare.

HOLAP (Hybrid OLAP) este o combinaţie între MOLAP şi ROLAP. Astfel, HOLAP stochează datele în tabele relaţionale în aceeaşi bază de date ca şi depozitul de date iar agregările în structuri multidimensionale în afara depozitului de date.

Stocarea datelor cubului într-o structură de tip HOLAP este recomandată în cazul interogărilor frecvente asupra datelor agregate ce au la bază un volum mare de date. De exemplu, rezultatele anuale ale studenţilor pot fi stocate într-o structură MOLAP iar rezultatele fiecărei sesiuni pot fi stocate într-o structură ROLAP.

După alegerea modalităţii de stocare şi procesarea cubului, datele devin disponibile şi pot fi interogate. Desigur că operaţia de procesare a cubului trebuie repetată periodic pentru actualizarea datelor din cub. Vizuali-

zarea datelor se face cu instrumentul Cube Browser care permite interogarea şi vizualizarea rapidă şi facilă a rezultatelor interogărilor într-un format standard.

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. Ele se aplică asupra unor volume foarte mari de date şi presupun calcule complexe (analiza tendinţei, medii, dispersii etc.) care necesită adesea agregări (group by). Tehnologia OLAP este cea care „se descurcă" 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, just-in-time procesului de decizie

Trebuie să subliniem că această aplicaţie nu se doreşte un sistem deoarece la realizarea ei nu s-au luat în calcul factorii care ţin de implementare şi problemele ce deriva de aici. A fost creat mai mult pentru a demonstra aplicabilitatea unui astfel de instrument de analiza a datelor (OLAP) în domeniu. Limita este dată în primul rând de faptul că nu a fost dezvoltată partea de interfaţă cu utilizatorul iar modulul pentru popularea depozitului de date a fost creat luând în considerare doar datele din baza de date tranzacţională a unei singure facultăţi (Facultatea de Ştiinţe Economice şi Administrative). Stadiul actual de dezvoltare al sistemelor tranzacţionale aferente facultăţilor este insular (nu au nici un fel de legătură unele cu altele) şi primitiv. Sunt facultăţi la care aplicaţiile de evidenţă ale studenţilor folosesc ca structuri de date, fişiere independente. In aceste condiţii crearea şi chiar proiectarea unui sistem de populare cu date din toate sistemele tranzacţionale ar fi atins o întindere şi o complexitate apreciabile şi ar fi depăşit preocupările autorului. Aplicaţia se vrea un punct de plecare în dezvoltări ulterioare dacă o astfel de soluţie de genul depozit de date se consideră utilă.

Bibliografie

1. Albescu F., Bojian I. „Management information systems and decizion support systems" Ed,. DualTech, Bucureşti 2001
2. Airinei D, „Sisteme informatice de asistare a deciziei", curs on-line
3. Airinei D. „Depozite de date", curs on-line
4. Bain T. &co, „Professiona SQLServer 2000 Datawarehousing with Analysis Services", Wrox Press, Londra 2000
5. Connolly T., „Baze de date", Teora, Bucureşti 2001
6. Tanrikorur Tuturu, "Enterprise DSS arhitecture: a hybrid approach", DM Review, February 1998
7. William McKnight, "Way a data warehouse?", McKnight Associates, Inc, April 2002
8. Zaharie D &co, „Sisteme informatice pentru asistarea deciziei", Ed. DualTech, Bucureşti 2001
9. ***"Microsoft SQL Server 7.0 Data Warehousing Training Kit", MS Press
10. ***"Microsoft SQL Server 7.0 Data warehousing strategy", MS Press
11. ***"Microsoft SQL Server 7.0 OLAP Services", MS Press
12. ***"MCSE Training Kit-SQL 7.0 Data Warehousing", MS Press
13. http://www.billinmon.com
14. http://www.datawarehouse.com
15. http://www.datawarehousingonline.com
16. http://www.dmreview.com
17. http://www.dw-institute.com
18. http://www.olapconcil.com
19. http://www.intelligententreprise.com
20. http://www.olapreports.com
21. http://www.bijonline.com