Data Warehouse

Prep. Maria Deac
Universitatea "Tibiscus" din Timişoara

ABSTRACT. "A data warehouse is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management's decision making process." The following article contains a presentation of the Data Warehousing concept and an exemplification of a data warehouse throughout two special programs – DB2 Warehouse Manager and Red Brick Warehouse Manager.

1 Prezentare generală

Deoarece în lumea afacerilor de azi doar schimbarea rămâne constantă, dezvoltarea tehnologiei bazelor de date a condus la apariţia unor baze de date mari şi foarte mari. O astfel de bază de date, în care o organizaţie păstrează toate datele proprii privind producţia, datele financiar contabile, clienţii, furnizorii etc. a căpătat denumirea de „*data warehouse*", adică depozit de date.

Depozitele de informaţii reprezintă o cerinţă acută a organizaţiilor moderne (fie ele întreprinderi, bănci, administraţie, etc) şi, totodată, o realitate tehnologică pusă în practică din ce în ce mai frecvent.

Datele din *data warehouse* provin în principal din datele capturate din sistemul operaţional, dar mai pot proveni din datele de arhivă (în perioada de constituire a depozitului) precum şi din surse externe, cum ar fi baze de date publice.

Câteva exemple posibile: date demografice (obţinute în urma unui recensământ), date statistice (furnizate de institute specializate), date de prognoză economică (furnizate de instituţii orientate pe studiul pieţei), date obţinute în urma unor sondaje de opinie, etc. Aceste date pot fi cumpărate, pot fi preluate pe bază de abonament sau pot fi date publice gratuite.

2 De ce are nevoie o *data warehouse* ca să funcţioneze?

O *data warehouse* trebuie să fie în stare să suporte diferite tipuri de aplicaţii informaţionale. Suportul de procesare a deciziilor este principalul tip de aplicaţie informaţională într-o *data warehouse*, dar folosirea unei *data warehouse* nu trebuie să fie restricţionată la un sistem cu suport de decizii. Este posibil ca fiecare aplicaţie să aibă propriul său set de nevoi în termeni de date, un mod propriu în care datele sunt modelate şi felul în care sunt folosite.

O *data warehouse* trebuie să consolideze datele de bază şi să furnizeze toate facilităţile pentru a deriva informaţii din ele după cerinţele utilizatorilor. Datele de bază detaliate sunt de primă importanţă, dar volumele de date, care tind să fie mari, şi utilizatorii au nevoie de obicei de informaţii derivate din datele de bază. Datele din *data warehouse* trebuie să fie organizate astfel încât să poată fi analizate sau explorate din diferite puncte de vedere.

Din punct de vedere al implementării unui depozit de date există mai mult tendinţe: o tendinţă ar fi implementarea unui sistem distribuit, descentralizat unde datele sunt păstrate în unităţi independente, „*Independent Data Marts*”, fiecare conţinând date relevante pentru un anumit aspect al operaţiilor unei instituţii. O a doua posibilitate ar fi implementarea unei surse de date unice, centralizate, la care au acces utilizatorii din toate departamentele unei instituţii.

Există însă şi posibilitatea combinării celor două abordări prin implementarea unei surse de date centralizate la nivelul întregii instituţii cu existenţa unor unităţi dependente, „*Dependent Data Marts*” care reprezintă subseturi ale datelor din depozitul de date şi care au fost selectate şi organizate pentru utilizarea specifică anumitor aspecte ale operaţiilor unei instituţii (resurse umane, contabilitate, etc.). Unităţile dependente *data marts* pot fi implementate logic în cadrul aceluiaşi depozit de date sau pot fi implementări fizice separate.

Uneori apar greşeli frecvente sau concepţii greşite în legătură cu *data marts* şi *data warehouse*. O *data marts* sau un „*târg de date*” nu este un depozit de date propriu zis, este un depozit de date cu scopul de a sprijini o aplicaţie de decizii, care conţine date pertinente unui anumit compartiment al unei companii, în timp ce *data warehouse* consolidează toate datele pentru o analiză mai bună, pentru a răspunde oricărei întrebări privind afacerile unei companii.

3 Caracteristicile unui depozit de date

Un depozit de date este folosit pentru stocarea datelor financiare non-volatile, tranzacţii şi evenimente. El constituie o sursă de informaţii pentru toate suporturile decizionale şi aplicaţii, conţinând date integrate consolidate şi securizate. În spatele unui *data warehouse* stă un subsistem populat cu un depozit de date obişnuit, o *data warehouse* centrală şi un suport de decizii bine definit pentru aplicaţii informaţionale. O sursă pentru un depozit de date poate fi aproape orice sursă relaţională sau nerelaţională (tabel sau fişier) care este conectată la reţea. Este esenţială identificarea tabelelor şi fişierelor care vor furniza date în *data warehouse*.

Există mai multe soluţii pentru lucrul cu depozitele de date, dintre acestea vom vorbi mai pe larg despre *DB2 Warehouse Manager* din pachetul *DB2 Universal Database* şi *Red Brick Warehouse Manager*.

4 DB2 Warehouse Manager

DB2 Warehouse Manager reuneşte instrumente pentru a construi, administra şi accesa depozitele de date *DB2*. *DB2 Warehouse Manager* simplifică şi accelerează crearea prototipurilor de depozite de date, dezvoltarea şi implementarea acestora şi oferă centrelor de calcul controlul pentru generarea de interogări, analiza costurilor, gestionarea resurselor şi urmărirea utilizării. Ajută la satisfacerea cerinţelor utilizatorului pentru a găsi, accesa şi înţelege informaţia. Oferă instrumente şi tehnici flexibile pentru construcţia, administrarea şi accesul depozitului de date. *DB2 Warehouse Manager* răspunde celor mai comune cerinţe pentru realizarea de rapoarte în intreprinderi de orice mărime. *DB2 Warehouse Manager* se alătură funcţiilor de bază şi analitice ale depozitelor de date disponibile în *DB2 Universal Database* oferind:

- scalabilitate sporită a depozitelor de date prin agenţi colocaţi cu baza de date, aceşti agenţi administrează fluxul între sursele depozitelor de date şi destinaţii;
- transformări avansate, utilizând proceduri memorate Java si funcţii definite de utilizator, inclusiv filtrarea datelor, pivotarea tabelelor, generarea de chei, etc.;
- un catalog integrat de informaţii ale afacerii pentru a îndruma utilizatorii către informaţiile relevante pe care le pot utiliza în luarea deciziilor;
- guvernarea interogărilor sofisticate şi distribuţia sarcinilor;

- rapoarte ale interogărilor care satisfac cerințele obişnuite ale majorității întreprinderilor;

5 Red Brick

Red Brick este un server de baze de date destinat să facă față cerințelor specializate pentru analiza datelor. Furnizează o platformă robustă şi scalabilă pentru construcția unor suporturi de aplicație bazată pe decizii. Dă posibilitatea mai multor utilizatori să analizeze mai multe date şi să ia decizii mai bune în timp scurt. Procesează seturi mari de date, efficient cu timpi de răspuns foarte mici. Reduce timpul şi costul întreținerii unui depozit de date.

Administrarea *data warehouse* se confruntă adesea cu integrarea datelor într-un mediu de baze de date eterogen. Ar putea să existe date stocate în *DB2*, dar ar fi de preferat avantajul capabilității lui *IBM Red Brick Warehouse* de a analiza date. Astfel, ne putem cconfrunta cu problema transferării datelor din *DB2* în *Red Brick*.

Integrarea acestor două sisteme de baze de date se poate realiza prin *DB2 Warehouse Manager* şi aduce avantaje tehnice şi beneficii economice. Există anumite metode de configurare care pot să ajute la transferul automat de date foarte mari între *DB2* şi *Red Brick*.

Într-un proces de exportare, transformare şi încărcare, o bază de date poate fi o sursă de date sau o țintă înspre care merg datele. În *DB Warehouse Manager*, *Red Brick* poate fi configurat să exporte date (ca şi sursă) sau să primească date importate (ca şi destinație) din alte surse cum ar fi *DB2*.

În afară de a configura o sursă sau o țintă de date *Red Brick*, procesele *Warehouse Manager* care implică *Red Brick* nu sunt diferite de alte surse de date când se folosesc funcții *data warehouse*, cum ar fi transformări, controlul proceselor şi programare (schedule).

În cele ce urmează vom dicuta despre două lucruri: În primul rând vom vedea cum se pot exporta date din *Red Brick*. Vorbim despre opțiunile de exportare ale datelor *Red Brick Warehouse* folosind ODBC sau folosind comenzi SQL de export din *Red Brick*.

În al doilea rând vom vedea cum se importă datele în *Red Brick*, adică despre felul cum se importă datele în *Red Brick Warehouse* folosind ODBC sau folosind "*încărcătorul Red Brick*" numit Table Management Utility (TMU) printr-un program definit de utilizator (User Defined Program (UDP)).

6 Exportul datelor în Red Brick

Se pot exporta date din *Red Brick Warehouse* în două moduri:

- **Folosind ODBC** - O bază de date *Red Brick* poate fi definită ca şi o sursă generică ODBC în *DB2 Warehouse Manager* folosind următorii paşi:
  - mai întâi trebuie definit un nume al sursei de date ODBC (Data Source Name DSN) pentru a accesa baza de date *Red Brick*;
  - apoi în *DB2 Data Warehouse Center* se execută paşii: Warehouse Source -> Define ->Generic ODBC pentru a termina paşii de configurare standard;
  - după aceea se poate folosi un pas SQL în *Warehouse Manager* pentru a primi date din orice tabel *Red Brick*, prin interfața ODBC.
- **Folosind exportul SQL** - Red Brick :
  - se execută paşii descrişi anterior (pentru ODBC);
  - sintaxa pentru exporul detaliat poate fi găsită în ghidul de referință Red Brick SQL :

export to output_file ddl_file tmu_file format_type

Comanda de export este invocată prin ODBC la fel ca şi orice interogare normală, dar scrie direct rezultatul execuției interogării într-un fişier, director sau într-un pipe, trecând peste straturile ODBC. Formatul de fişiere suportate poate fi de lungime fixă şi binară. Suportul XML a fost adăugat în *Red Brick* versiunea 2. Toate fişierele de ieşire rezultate în urma exportului SQL trebuie să fie accesibile de pe stația locală.

Atunci când se specifică un format extern, fiecare coloană din rezultat va avea o lungime fixă.

Din cele două metode de export, metoda SQL este de preferat din cauza performanței mai ridicate. Este de reținut faptul că fişierul care rezultă dintr-un export SQL trebuie să fie accesibil de pe stația locală. Dacă utilizatorul trebuie să mute datele între stații, el poate folosi un pas ajutător ftp pentru a trimite fişierele către o stație aflată la distanță.

Metoda ODBC este mai lentă decât exportul SQL, dar poate muta date între stații fără a avea nevoie de paşi în plus.

7 Importul datelor în Red Brick

La fel ca şi exportul datelor se poate opta între două metode de import a datelor.

- **Folosind ODBC** – O bază de date *Red Brick* poate fi definită ca o țintă de date ODBC.
- **Folosind un program TMU- UDP** – utilizatorii pot configura un program UDP în *Warehouse Manager* pentu a folosi *Red Brick* TMU. Pentru a seta parametrii UDP, un utilizator specifică calea pâna la: un executabil TMU, un fişier de control TMU, o bază de date țintă, un user-name şi o parolă. După aceasta TMU UDP va fi disponibil pe panoul de instrumente al ferestrei modelelor de procese.

Din cele două metode se recomandă TMU UDP pentru că lucrează cu volume mari de date. Această metodă are nevoie ca fişierele de intrare şi fişierele TMU să fie accesibile de pe stația locală. Dacă accesul este o problemă se poate folosi clientul TMU din R*ed Brick 6.2* sau un pas adițional ftp.

Metoda ODBC poate muta date între stații, dar pentru că este mult mai lentă este folosită pentru lucrul cu volume mai mici de date.

În concluzie, din cauza performanței ridicate, se recomandă folosirea comenzilor de export SQL şi a programului TMU UDP pentru exportul/ importul în/din baze de date mari. Amândouă metode oferă instrumente pentru întreținerea magaziei de date .

Bibliografie

[Inm03] **Wiliam H. Inmon,** *Building the Data Warehouse*, third Edition, 2003

[QiJ03] **Qi Jin**, *IBM Red Brick @ Warehouse Development*, IBM Silicon Valley lab, 2003

[Sar98] **Mircea Sârbu**, *În căutarea definiției*, Computer Press Agora SRL, 1998