Cuprins

VIOREL STOIAN

SQL. APLICAȚII

EDITURA UNIVERSITARIA
Craiova, 2012

Capitolul I

Introducere

1.1. Noțiuni fundamentale despre bazele de date relaționale

Printr-o **bază de date** înțelegem o colecție de date stocate şi accesate într-un mod organizat în aşa fel încât să prezinte avantaje pentru utilizator.

Apariția bazelor de date în accepțiunea acestei definiții a fost cauzată de necesitatea obținerii de informații utile dintr-o cantitate imensă de date şi de o mare diversitate, informații referitoare la unele macro-sisteme cum ar fi organizațiile foarte mari (exemplu: stocarea datelor referitoare la structura organizațională şi funcționarea corporațiilor) sau referitoare la procese naturale complexe (exemplu: stocarea datelor pe o perioadă lungă de timp şi de pe o arie extinsă a parametrilor specifici din domeniul meteorologiei şi hidrologiei) etc.

Datorită faptului că o bază de date se referă la structura şi funcționarea unui anumit sistem putem afirma că aceasta creează un *model* informațional al acelui sistem. În timp au evoluat mai multe tipuri de modele ale bazelor de date, principale fiind următoarele:

- Modelul *ierarhic*. Datele sunt stocate în structuri de tip arbore.
- Modelul *rețea*. Datele sunt stocate sub formă de înregistrări cu legături multiple şi complexe între ele. Este o extindere a celui ierarhic.
- Modelul *relațional*. Reprezintă cea mai simplă structură pe care o poate avea o bază de date. Datele sunt organizate în tabele formate din linii şi coloane.
- Modelul *relațional orientat pe obiect*. Este introdus conceptul de *obiect*. Integrează principiile programării orientate pe obiect cu cele ale bazelor de date.

În prezent, cel mai utilizat dintre acestea este modelul relațional datorită simplității sale, dar şi fundamentării matematice riguroase. Acest model a fost propus în iunie 1970 de către Edgar F. Codd în lucrarea sa intitulată „*A Relational Model of Data for Large Shared Data Banks* [4], iar în 1985 acelaşi autor a publicat un set de 13 reguli în raport cu care o baza de date poate fi considerată relațională [5]. Ținând cont de caracteristicile bazelor de date relaționale, marile companii şi corporații de software au creat sisteme de administrare a acestor structuri de date numite SGBDR (Sisteme de Gestiune a Bazelor de Date Relaționale) sau RDBMS (Relational Data Base Management System) care

încearcă să implementeze în totalitate aceste reguli pentru creşterea performanței în interacțiunea utilizatorului cu baza de date. Iată, pe scurt, aceste reguli [5], [8],[10]:

R1. *Regula reprezentării logice a datelor*

Într-o bază de date relațională toate datele sunt reprezentate la nivel logic într-un singur mod şi anume sub formă de valori atomice în tabele. Valoarea stocată la intersecția dintre un rând şi o coloană ale unui tabel trebuie să fie atomică, adică să nu mai poată fi descompusă din punct de vedere logic.

R2. *Regula accesului la date*

Toate datele individuale din tabele trebuie să fie accesibile prin furnizarea numelui tabelului, numelui coloanei şi valorii cheii primare.

R3. *Regula reprezentării valorilor necunoscute*

Un sistem relațional trebuie să permită declararea şi manipularea sistematică a valorilor Null cu semnificația unor valori necunoscute, absente sau inaplicabile.

R4. *Regula dicționarului de date*

Descrierea bazei de date (informațiile despre obiectele din baza de date-dicționarul de date) trebuie să fie reprezentată la nivel logic tot sub formă de tabele, astfel încât asupra acesteia să se poată aplica aceleaşi operații ca şi asupra datelor propriu-zise.

R5. *Regula limbajului de acces*

Într-un model relațional trebuie să existe cel puțin un limbaj de accesare a datelor, care să asigure următoarele operații: definirea tabelelor de bază şi a tabelelor virtuale (vederilor), manipularea şi interogarea datelor (atât interactiv cât şi prin program), definirea restricțiilor de integritate, autorizarea accesului la date, delimitarea tranzacțiilor.

Limbajul care s-a impus pentru realizarea interacțiunii dintre utilizator şi baza de date este limbajul SQL (Structured Query Language) care va fi prezentat în capitolele urmatoare.

R6. *Regula de actualizare a tabelelor virtuale (vederilor)*

Un SGBD trebuie să poată determina dacă o vedere poate fi actualizată sau nu. Deoarece tabelele virtuale sunt obținute pe baza tabelelor din baza de date actualizarea acestora presupune propagarea actualizării la tabelele din care provin. Nu întotdeauna această propagare este unică. Un SGBDR trebuie să dispună de un set de reguli care să stabilească dacă o coloană a unei vederi poate sau nu să fie actualizată.

R7. *Regula manipulării datelor*

Un sistem relațional trebuie să ofere posibilitatea procesării tabelelor (de bază sau virtuale) nu numai în operațiile de interogare a datelor cât şi în cele de inserare, actualizare şi ştergere.

R8. *Regula independenței fizice a datelor*

Programele de aplicație nu trebuie să depindă de modul de stocare şi accesare fizică a datelor.

R9. *Regula independenței logice a datelor*

Programele de aplicație nu trebuie să fie afectate de nici o restructurare logică a tabelelor bazei de date care conservă datele.

R10. *Regula independenței datelor din punctul de vedere al integrității*

Regulile de integritate a bazei de date trebuie să fie definite în limbajul utilizat de sistem pentru definirea datelor şi nu în cadrul aplicațiilor individuale: în plus, aceste reguli de integritate trebuie stocate în dicționarul de date.

R11. *Regula independenței datelor din punctul de vedere al distribuirii*

Programele de aplicație nu trebuie să fie afectate de distribuirea pe mai multe calculatoare a bazei de date.

R12. *Regula privind prelucrarea datelor de către un limbaj de nivel inferior*

Orice limbaj nerelațional folosit pentru accesarea datelor trebuie să respecte aceleaşi condiții de integritate ca şi limbajul relațional de acces.

R0. *Regula de bază*

Un SGBD relațional trebuie să fie capabil să gestioneze o bază de date exclusiv pe baza caracteristicilor sale relaționale.

Această regulă are rolul de a rezuma concluziile desprinse din celelalte reguli.

Regulile de mai sus sunt grupate în 5 categorii şi anume: reguli de bază (R0 şi R12), reguli structurale (R1 şi R6), reguli privind integritatea datelor (R3 şi R10), reguli privind manipularea datelor (R2, R4, R5 şi R7), reguli privind independența datelor (R8, R9 şi R11).

O bază de date care respectă în totalitate cele 13 reguli ale lui Codd se numeşte bază de date relațională ideală, iar SGBDul care o administrează, un SGBDR ideal.

Un astfel de sistem ar trebui să realizeze următoarele funcțiuni: stocarea datelor, definirea structurilor de date, manipularea datelor, interogarea (extragerea şi prelucrarea) datelor, asigurarea securității datelor, asigurarea integrității datelor, accesul concurent la date cu păstrarea consistenței acestora, asigurarea unui mecanism de recuperare a datelor, asigurarea unui mecanism de indexare care să permită accesul rapid la date.

În această lucrare vom discuta despre limbajul SQL (Structured Query Language) care este un limbaj cu ajutorul căruia interacționam cu o bază de date.

În capitolele următoare vom prezenta diferite exemple şi aplicații care să scoată în evidență capabilitățile sale. Interacțiunea despre care aminteam mai sus se referă la crearea de structuri specifice bazelor de date, introducerea de date în baza de date, ştergerea datelor, modificarea lor, vizualizarea datelor în vederea utilizării lor în diferite scopuri, dar şi multe alte operații.

Cele afirmate mai sus impun să facem o trecere în revistă a câtorva noțiuni elementare, fundamentale despre aceste structuri. Ne vom referi la bazele de date relaționale fiindcă acest model este folosit pe scară largă în momentul actual.

O bază de date relațională este o bază de date în care datele sunt organizate logic în structuri bidimensionale de date, asemănătoare tabelelor, formate din linii şi coloane care constituie elementele de stocare folosite [14].

Observație: nu trebuie facută confuzia între noțiuni precum baza de date propriu-zisă, date, sisteme de gestiune a bazelor de date. Baza de date propriu-zisă este doar ansamblul de "recipiente" în care sunt depozitate datele. Aceste recipiente

sunt, din punct de vedere logic, tabele sau, din punct de vedere fizic, fişiere. De cele mai multe ori, printr-o bază de date înțelegem o bază de date propriu-zisă populată cu date. SGBD-ul este un program software ce ne permite implementarea bazei de date pe sistemul de calcul şi, ulterior, interacțiunea cu aceasta [13].

O bază de date este proiectată stabilindu-se un set de tabele cu relații între ele ținând cont de câteva principii: să fie stocate toate datele ce ne interesează, să fie evitată redundanța datelor, să nu existe ambiguitate în identificarea obiectelor bazei de date, să fie asigurată persistența datelor, simplitatea stocării şi manipulării acestora, să fie asigurată independența datelor (separarea aspectelor logice de cele fizice) [1], [2], [3], [11].

Există mai multe metode de proiectare a BD relaționale, dar niciuna nu s-a impus cu desăvârşire până în prezent. Fiecare dintre ele însă se bazează pe considerații practice referitoare la sistemele modelate, considerații care stau la baza activităților de procesare a datelor şi anume: scenariu, set de specificații, protocoale de funcționare a sistemului, modul cum sunt folosite datele, modul de gestionare a acestora etc [6], [7], [8].

Proiectarea bazelor de date cuprinde 4 etape principale [3], [9], [10]:

- Analiza scenariului/setului de specificații impuse/protocoalelor de funcționare a sistemului/caietului de sarcini etc., discuții cu beneficiarul, ajustări, adaptări.
- Realizarea schemei conceptuale a bazei de date sau a schemei entitate-legatură (ERD–Entity Relationship Diagram). Sunt stabilite următoarele elemente: entitațile (obiecte de interes din sistem pentru care trebuie să existe date înregistrate), relațiile/legăturile (asocieri nedirecționate între 2 entități), cardinalitățile relațiilor (numarul de instanțe din fiecare entitate care poate participa la relație) şi atributele (caracteristici ale entităților sau ale unor relații).
- Realizarea modelului logic al bazei de date (schemei logice a bazei de date). Aici se realizează: prelucrarea sau eliminarea relațiilor non-conforme, transformarea entităților în tabele, a relațiilor în chei straine şi a atributelor în coloane.
- Realizarea proiectului fizic al bazei de date (schemei fizice a bazei de date) Structurile logice sub formă de tabele se regăsesc fizic în fişiere organizate în memoria sistemului de calcul.

Structurile de date folosite sunt:

- *tabelul* = structura de bază, bidimensională pentru stocarea datelor. În literatura de specialitate se mai întâlneşte şi denumirea de *relație*. Un tabel conține datele referitoare strict la o entitate.
- *atribute* = numele *coloanelor* tabelelor
- *aritatea relației* = numărul de coloane ale tabelului
- *domeniu* = mulțimea în care un atribut poate lua valori
- *înregistrări = liniile*, *rândurile* tabelului. Este întâlnită şi denumirea de *tupluri*. În locațtii sau câmpuri se găsesc valorile înregistrărilor pentru anumite atribute.
- *câmpuri, locații* = se găsesc la intersecțiile coloanelor cu liniile. Acestea ar trebui să conțină informație *atomică*, ce nu mai poate fi divizată. Atunci când o valoare dintr-o locație e necunoscută sau neaplicabilă, în acea locație se va înregistra o valoare convențională, specială în bazele de date, denumită NULL.

Câteva considerații:
- Numele tabelelor trebuie să fie unice în cadrul schemei unui utilizator.
- De asemenea, numele coloanelor trebuie sa fie unice în cadrul unui tabel.
- Ordinea în care se găsesc dispuse coloanele într-un tabel nu are relevanță.
- De asemenea, nu are relevanță ordinea în care sunt returnate înregistrările.
- Într-un tabel poate exista o coloană sau un set de coloane care au valori unice, nu permit duplicatele, însă permit existenta valorilor NULL. Această coloană sau set de coloane poartă denumirea de *cheie unică*. Însă trebuie să existe neapărat o coloană sau un set de coloane (cheie) cu ajutorul căreia să putem identifica în mod unic orice linie a tabelului. Datorită funcției pe care trebuie să o îndeplinească, o astfel de cheie nu poate conține duplicate şi nici valori NULL. Ea poartă denumirea de *cheie primară*.
- Există noțiunea de *cheie străină*. De exemplu, să presupunem că avem 2 tabele: tabelul AGAJATI - unde sunt înregistrate date despre membrii (angajații) unei organizații oarecare şi care are cheia primară *cod_angajat* şi tabelul SECTII - unde sunt date despre departamentele (secțiile) organizației şi care are cheia primară *cod_sectie*. În primul tabel exista o coloana, *cod_sectie*, care ne informează despre apartenența la o anumită secție a fiecărui angajat. Atributul (coloana) *cod_sectie* din tabelul ANGAJATI se numeste *cheie străină* în acest tabel deoarece face referire la atributul (coloana) *cod_sectie* din tabelul SECTII unde acesta are statutul de *cheie primară*. O cheie străină poate face referire la una dintre cheile dintr-un alt tabel sau din tabelul propriu. Valorile sale trebuie să se regasească printre valorile cheii la care face referire sau să fie valori NULL. Cheile straine stabilesc legăturile dintre tabele.

Pentru asigurarea integrității datelor, o bază de date trebuie să satisfacă un număr de constrângeri, numite *constrângeri de integritate* care sunt de două tipuri: *constrângeri structurale* (care trebuie satisfacute de orice bază de date ce utilizează modelul relațional) şi *constrângeri de comportament* (care sunt specifice fiecarei baze de date, în particular) [10].

1.2. Obiectele bazei de date

O bază de date conține mai multe structuri de date care se mai numesc şi obiecte ale bazei de date. De exemplu, o bază de date Oracle conține următoarele obiecte [16], [17]:
- *tabele* – structuri de bază compuse din linii şi coloane şi care au drept scop să stocheze date
- *vederi* – structuri logice de date, asemănătoare tabelelor, obținute din unul sau mai multe tabele
- *secvențe* – generatoare de valori numerice
- *indecşi* – elemente ce îmbunătățesc performanțele anumitor interogări
- *sinonime* – denumiri alternative pentru anumite obiecte ale bazei de date