

VIOREL STOIAN

**INTEROGĂRI COMPLEXE
CU SQL ORACLE**



**EDITURA UNIVERSITARIA
Craiova, 2017**

Referenți științifici:

Conf. univ. dr. ing. AUGUSTIN IONESCU

Prof. univ. dr. ing. LIANA STĂNESCU

Copyright © 2016 Universitaria

Toate drepturile sunt rezervate Editurii Universitaria

Descrierea CIP a Bibliotecii Naționale a României

STOIAN, VIOREL

Interogări complexe cu SQL Oracle / Viorel Stoian. - Craiova :
Universitaria, 2017

Bibliogr.

ISBN 978-606-14-1091-0

004.43 SQL

Prefață

Cartea “*Interogări complexe cu SQL Oracle*” este rezultatul experienței mai multor ani de activitate specifică la disciplina *Baze de date* cu studenți ai Facultății de Automatică, Calculatoare și Electronică din Craiova. Ședințele de curs, de proiect sau aplicațiile practice în laborator ofereau excelente ocazii de pătrundere în tainele acestui interesant domeniu. De aici decurge caracterul de manual al lucrării.

Lucrarea pornește de la premisa că o bază de date a fost deja proiectată pe baza unui scenariu sau a unui set de specificații, parcurgându-se toate etapele specifice: analiza atentă a cerințelor, stabilirea modelelor conceptual, logic și fizic, efectuarea normalizării. Procesul este finalizat cu un set de tabele cărora li s-a stabilit structura relațională. Pe măsură ce parcurge capitolele, cititorul se familiarizează rând pe rând cu implementarea setului de tabele în baza de date, cu modificarea structurii acestora, cu popularea cu date sau cu eliminarea lor, dacă este necesar. Apoi, după mai multe exemple și aplicații, va ști să modifice datele din tabele, să le ștergă când este cazul și să interogheze baza de date pentru extragerea informațiilor dorite.

Instrumentul ideal pentru efectuarea acestor acțiuni este limbajul SQL (Structured Query Language) care este un limbaj standard, structurat și universal pentru interogarea bazelor de date, recunoscut de organisme importante pe plan mondial precum ANSI (American National Standards Institute) și ISO (International Standards Organization). Deși este un limbaj simplu, cu un număr relativ mic de instrucțiuni, este foarte puternic și permite efectuarea unor operații complexe cu bazele de date, devenind util atât pentru proiectanții și dezvoltatorii de aplicații cât și pentru utilizatorii obișnuiți.

SQL stă la baza tuturor Sistemelor de Gestiune a Bazelor de Date Relaționale (SGBDR). Trebuie precizat faptul că majoritatea companiilor de software care au dezvoltat SGBD-uri (Oracle, Microsoft etc.) au utilizat varianta SQL standard (numit și ANSI SQL) căreia i-au adăugat diverse extensii în scopul îmbunătățirii performanțelor programării și simplificării unor operații specifice.

Cei interesați vor găsi în paginile lucrării de față o multitudine de exemple de interogări bazate pe limbajul SQL - varianta Oracle care își propun să scoată în evidență bogăția și diversitatea aspectelor dialogului dintre o bază de date și utilizatorii acesteia.

Autorul

Capitolul I

Introducere

1.1. Noțiuni fundamentale despre bazele de date relaționale

Printr-o **bază de date** înțelegem o colecție de date stocate și accesate într-un mod organizat în așa 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 și de o mare diversitate, informații referitoare la unele macro-sisteme cum ar fi organizațiile foarte mari (exemplu: stocarea datelor referitoare la structura organizațională și funcționarea corporațiilor) sau referitoare la procese naturale complexe (exemplu: stocarea datelor pe o perioadă lungă de timp și de pe o arie extinsă a parametrilor specifici din domeniul meteorologiei și hidrologiei) etc.

Datorită faptului că o bază de date se referă la structura și 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 și 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 și 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 și 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* [5], iar în 1985 același autor a publicat un set de 13 reguli în raport cu care o baza de date poate fi considerată relațională [6]. Ținând cont de caracteristicile bazelor de date relaționale, marile companii și 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 creșterea performanței în interacțiunea utilizatorului cu baza de date. Iată, pe scurt, aceste reguli [6], [9],[11]:

R1. Regula reprezentării logice a datelor

Într-o bază de date relațională toate datele sunt reprezentate la nivel logic într-un singur mod și anume sub formă de valori atomice în tabele. Valoarea stocată la intersecția dintre un rând și 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 și valorii cheii primare.

R3. Regula reprezentării valorilor necunoscute

Un sistem relațional trebuie să permită declararea și 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 aceleași operații ca și 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 tabelor de bază și a tabelor virtuale (vederilor), manipularea și interogarea datelor (atât interactiv cât și prin program), definirea restricțiilor de integritate, autorizarea accesului la date, delimitarea tranzacțiilor.

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

R6. Regula de actualizare a tabelor virtuale (vederilor)

Un SGBD trebuie să poată determina dacă o vedere poate fi actualizată sau nu. Deoarece tabelele virtuale sunt obținute pe baza tabelor 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 tabelor (de bază sau virtuale) nu numai în operațiile de interogare a datelor cât și în cele de inserare, actualizare și ștergere.

R8. Regula independenței fizice a datelor

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

R9. Regula independenței logice a datelor

Programele de aplicație nu trebuie să fie afectate de nici o restructurare logică a tabelor 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 și 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 distribuției

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 aceleași condiții de integritate ca și 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 și anume: reguli de bază (R0 și R12), reguli structurale (R1 și R6), reguli privind integritatea datelor (R3 și R10), reguli privind manipularea datelor (R2, R4, R5 și R7), reguli privind independența datelor (R8, R9 și R11).

O bază de date care respectă în totalitate cele 13 reguli ale lui Codd se numește 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 și 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 interacționăm cu o bază de date.

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

Cele afirmate mai sus impun trecerea 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 și coloane care constituie elementele de stocare folosite [16].

Observație: nu trebuie făcută 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, fișiere. 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 și, ulterior, interacțiunea cu aceasta [14].

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 și manipulării acestora, să fie asigurată independența datelor (separarea aspectelor logice de cele fizice) [1], [2], [3], [12].

Există mai multe metode de proiectare a BD relaționale, dar niciuna nu s-a impus cu desăvârșire 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 și anume: scenariu, set de specificații, protocoale de funcționare a sistemului, modul cum sunt folosite datele, modul de gestionare a acestora etc [6], [7], [9].

Proiectarea bazelor de date cuprinde 4 etape principale [3], [10], [11]:

- 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: entitățile (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) și 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 și 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 fișiere 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 întâlnește și 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 și denumirea de *tupluri*. În locații 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 conventională, 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 să 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 existența 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 și 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 și care are cheia primară *cod_angajat* și tabelul SECTII - unde sunt date despre departamentele (secțiile) organizației și care are cheia primară *cod_sectie*. În primul tabel există o coloană, *cod_sectie*, care ne informează despre apartenența la o anumită secție a fiecărui angajat. Atributul (coloana) *cod_sectie* din tabelul ANGAJATI se numește *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 străine 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 satisfăcute de orice bază de date ce utilizează modelul relațional) și *constrângeri de comportament* (care sunt specifice fiecărei baze de date, în particular) [11].

1.2. Obiectele bazei de date

O bază de date conține mai multe structuri de date care se mai numesc și obiecte ale bazei de date. De exemplu, o bază de date Oracle conține următoarele obiecte [21], [22]:

- *tabele* – structuri de bază compuse din linii și coloane și 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
- *indecși* – elemente ce îmbunătățesc performanțele anumitor interogări
- *sinonime* – denumiri alternative pentru anumite obiecte ale bazei de date

Aceste structuri pot fi modificate on-line. Setul de obiecte ale bazei de date ce aparțin unui anumit utilizator se numește *schema* aceluși utilizator și poartă același nume cu utilizatorul.

1.3. SQL - un limbaj structurat, universal pentru interogarea bazelor de date. Componente

Limbajul SQL (Structured Query Language) oferă posibilitatea utilizatorului să interacționeze cu baza de date. Organisme importante pe plan mondial precum ANSI (American National Standards Organization) și ISO (International Standards Organization) au acceptat propunerea ca limbajul SQL să fie considerat un limbaj standard pentru bazele de date relaționale. Este un limbaj simplu având cuvinte descriptive și “frazе” (instrucțiuni) ușor de înțeles. Deși conține un număr relativ mic de instrucțiuni, este foarte puternic și permite efectuarea unor operații complexe cu bazele de date .

Limbajul SQL este un instrument extrem de util atât pentru administratorii bazelor de date, pentru proiectanții și dezvoltatorii de aplicații cât și pentru utilizatorii obișnuiți. Stă la baza tuturor SGBDR-urilor (Sisteme de Gestiune a Bazelor de Date Relaționale) [14], [23] ... [26], [29]. Însă, trebuie precizat faptul că majoritatea companiilor de software care au în portofoliul lor lucrul cu baze de date utilizează varianta SQL standard (numit și ANSI SQL) căreia i-au mai adăugat diverse extensii în scopul îmbunătățirii performanțelor și simplificării unor operații specifice (exemple: PL/SQL, Transact-SQL etc.) [13], [20], [28].

Instrucțiunile SQL pot fi grupate în seturi de instrucțiuni specifice, în funcție de acțiunile pe care le realizează, seturi de instrucțiuni care mai poartă numele de sub-limbaje SQL [11], [21], [22], [27], [28], [30]:

- instrucțiuni care au rolul de a crea, modifica, desființa structuri ale bazelor de date (Data Definition Language – DDL): CREATE, ALTER, DROP, RENAME, TRUNCATE. Efectul acestor instrucțiuni este consemnat în *dicționarul de date*.
- instrucțiuni care au rolul de a introduce noi date, a șterge date, a modifica date (Data Manipulation Language – DML): INSERT, UPDATE, DELETE, TRUNCATE, MERGE.
- instrucțiuni care au rolul de a administra modificările realizate prin DML în urma așanumitelor *tranzacții logice* (controlul tranzacțiilor): COMMIT, ROLLBACK, SAVEPOINT.
- instrucțiuni prin intermediul cărora sunt administrate diferite drepturi ale diverșilor utilizatori în scopul protecției datelor (Data Control Language-DCL): GRANT, REVOKE.
- instrucțiuni pentru interogarea, regăsirea datelor din tabele (DATA QUERY LANGUAGE – DQL): instrucțiunea SELECT cu un bogat set de clauze.