

Software profesional pentru aplicatii cu baze de date

Structura curs

- **Introducere**
 - Modelul relational
 - Proiectarea bazelor de date relationale, arhitectura unui SGBDR, organizarea logica a bazei de date, acces concurent la date si pastrarea consistentei datelor
- **SQL**
 - Limbajul de de definire a (structurii) datelor; limbajul de modificare a datelor; limbajul de interogare a datelor;
 - Securitatea unei baze de date; asigurarea integritatii si consistentei datelor; marirea performantei.
 - Functii si proceduri stocate, tranzactii, declansatoare, contoare.
- **Proiectarea unei aplicatii cu baze de date**
 - Specificatii produs (interfata utilizator, fluxuri de functionare, conditii functionale);
 - Documentatie de proiectare (de nivel inalt, detaliata);
 - Implementarea solutiei software; validarea si asigurarea calitatii;
 - Mentenabilitate, robustețe, flexibilitate

Bibliografie

- [1] Rick van der Lans, Introduction to SQL, Addison Wesley, 2000.
- [2] Florentin Eugen Ipate, Monica Popescu, Dezvoltarea aplicatiilor de baze de date in Oracle 8[...], All, 2000.
- [3] Ramalho Jose A., Learn SQL in 3 days, Wordware Publishing Inc., 2001.
- [4] Larry Ullman, PHP si MySQL pentru site-uri web dinamice, Teora, 2006.
- [5] MySQL 5.5 Reference Manual: http://docs.oracle.com/cd/E17952_01/refman-5.5-en/index.html
- [6] MySQL 5.6 Reference Manual: <http://dev.mysql.com/doc/refman/5.6/en/index.html>

Introducere

- Scurt istoric, scop si utilizare
- Arhitectura unui sistem de baze de date
 - Nivel conceptual
 - Nivel extern
 - Nivel logic
 - Nivel intern
- Sisteme de gestiune a bazelor de date
SGBD

Modelul de date, Comunicarea cu baza de date, Securitatea datelor, Mentinerea si constrangerea integritatii, Asigurarea accesului concurent la date, Baze de date distribuite

Introducere • Scurt istoric, scop si utilizare

- **Definitie:** Colectie de date organizate care serveste unui anume scop
- **Exemplu:** agenda telefonica
- **Diverse aplicatii:** aplicatii economice (contabilitate, financiare, bancare, personal, asigurarea calitatii, controlul fluxului productiei, specifice fiecarui domeniu), stiintifice, militare
- **Tipuri de date:** text/numeric dar si imagini, sunete, multimedia
- **Origine:**
 - Sisteme istorice (ne-electronice) de stocare, indexare si accesare a informatiei (biblioteci, administratie publica, dosare medicale);
 - Nevoia de a asigura persistenta datelor de la o rulare la alta a unei aplicatii informatice;
 - Nevoia de a asigura simplitatea stocarii si manipularii datelor;

Introducere • Scurt istoric, scop si utilizare

• **Cronologie:**

- **1960:** primele aplicatii de baze de date comerciale (companii private): model de tip retea (CODASYL) si de tip ierarhic (IMS); acces prin intermediul pointerilor low-level; detalii de stocare depind de tipul de date ce trebuie stocate; succes comercial: SABRE – IBM si American Airlines
- **1970-1972:** E.F. Codd propune modelul relational de organizare a datelor; deconecteaza schema (tipul de organizare logica a datelor) de tipul de stocare fizica.
- **1970-1980:** apare Ingres, din care se vor dezvolta apoi Ingres Corp., Sybase, MS SQL Server; apare System R (IBM), din care se vor dezvolta Oracle, IBM DB2
- **1976:** P. Chen propune modelul entitate-relatie; aceasta abordare va permite concentrarea pe proiectarea functionala si nu pe organizarea tabelor
- **1980-1990:** revolutia PC face posibila dezvoltarea PARADOX, DBASE III, IV, FoxPro; DB2 devine un produs important al IBM; SQL devine standardul general acceptat pentru SGBDR.
- **1990 (inceputul decadei):** Apar si se dezvolta aplicatii dedicate dezvoltarii de aplicatii de baze de date: VB, PowerBuilder (Sybase), Oracle Developer; modelele client-server se impun treptat pe piata;
- **1990 (mijlocul decadei):** Aparitia Internetului; regandirea in perspectiva a modelelor de aplicatii de baze de date;
- **1990 (sfarsitul decadei):** Investitiile masive in Internet stimuleaza aparitia si dezvoltarea de produse ce faciliteaza interconectarea si manipularea datelor pentru aplicatii online (ASP, FrontPage, Java Servlets, JDBC, Java Beans, ColdFusion, Dream Weaver, Oracle Developer 2000)
- **2000:** Trei mari companii domina: Oracle, IBM (cumpara Informix) si Microsoft
- **2009:** Open source MySQL/Apache server inlocuieste sisteme scumpe/istorice/comerciale (dar ...Oracle cumpara Sun)
- **now:** noSQL (BigTable, MapReduce, MongoDB), XML

Tendinte

- Baze de date de dimensiuni uriase (TB) impun regandirea metodelor de acces si analiza a datelor;
- Au aparut succesori ai SQL (noSQL, incluzand XML): tabele nenormalizate, scalare horizontala, stocare cheie-valoare;
- Probleme etice/de securitate/utilizare devin, periodic, de mare importanta

Introducere • Arhitectura unui sistem de baze de date

- Nivelul conceptual
- Nivelul extern
- Nivelul logic
- Nivelul intern

Nivelul conceptual

Descrierea sistemului utilizand limbajul natural, diagrame intuitive, exemple;

Se descrie informatia utilizand cele mai simple/mici unitati posibile (un fapt la un moment dat)

La acest nivel va fi realizata schema conceptuala a unui sistem, designul general al sistemului

Nivelul extern

Descrierea sistemului asa cum este el percept de un utilizator sau grup de utilizatori

Schema externa: specificarea informatiilor asa cum pot fi ele vazute de un utilizator/grup de utilizatori (si in functie de nivelele de securitate implementate)

Moduri de prezentare diferite a aceleiasi informatii pentru utilizatori diferiti

La acest nivel se proiecteaza diferitele interfete utilizator

Introducere • Arhitectura unui sistem de baze de date

Nivelul logic

Pentru realizarea implementarii eficiente a sistemului, schema conceptuala trebuie convertita intr-o structura de nivel inferior

Se alege, pentru o aplicatie data, un model logic adecvat (relational, ierarhic, retea etc.)

Exemplu: pentru un model relational, faptele sunt depozitate in tabele, constrangerile sunt realizate cu ajutorul cheilor primare sau straine etc.

Nivelul intern

Dupa proiectarea schemei logice, aceasta trebuie sa fie implementata conform unei scheme interne intr-un anumit SGBD

Exemplu: schema relationala poate fi implementata in DB2, Sybase, Oracle, Access, MySQL

Schema interna include toate detaliile despre stocarea fizica si structurile de acces utilizate in sistemul respectiv (indecsi, clustere etc.). Pentru aceeasi schema logica pot fi alese diferite scheme interne (depind de SGBD-ul folosit)

Sisteme de gestiune a bazelor de date SGBD

- Modelul de date
- Comunicarea cu baza de date
- Securitatea datelor
- Mentinerea si constrangerea integritatii
- Asigurarea accesului concurent la date
- Baze de date distribuite

Sisteme de gestiune a bazelor de date SGBD

Modelul de date

- **Ierarhic**
- **Retea**
- **Relational**

Modelul de date descrie modul de organizare a datelor in baza de date.

SGBD ierarhic: datele sunt stocate in structuri de tip arbore; intre date exista relatie de tip parinte-copil

Exemple:

- IMS de la IBM; proiectat initial pentru gestiunea BOM (lista de materiale) pentru programul Apollo; este si un sistem de management al informatiei si un mediu foarte puternic tranzactional (TM); implementarea foloseste blocuri de date (segmente); ex: clientul intr-un prim segment, sub care pot sa fie plasate segmente pentru fiecare comanda; sub fiecare astfel de segment ar putea fi plasate segmente pentru fiecare element dintr-o comanda; se modeleaza cu usurinta o relatie de tip 1:N;
- Windows Registry de la Microsoft (stocheaza setarile aplicatiilor instalate in Windows); registrii contin chei si valori; cheile sunt asemenea folderelor;

Sisteme de gestiune a bazelor de date SGBD

SGBD retea: datele sunt stocate sub forma de inregistrari si legaturi intre acestea; o inregistrare copil are mai multi parinti

1969 – Consorțiul CODASYL standardizeaza modelul de baze de date retea

Exemple: CODASYL;

Structura (schema) bazei de date este un graf generalizat de tipuri de inregistrari conectate de tipuri de relatii;

baza de date propriuzisa este un graf generalizat de inregistrari conectate de relatii.

datele sunt organizate in inregistrari si seturi (multimi); seturile definesc relatii unu-la-multi intre inregistrari: multipli membri, un singur owner; implementarea relatiilor utilizeaza pointeri, rezultand intr-o performanta deosebita in accesarea datelor, in detrimentul operatiunilor de reorganizare a bazei de date.

SGBDR: datele sunt organizate in tabele; tabelele sunt formate din inregistrari; inregistrările sunt formate din campuri; doua sau mai multe tabele sunt legate prin intermediul unuia sau mai multor campuri.

Exemple: Oracle, Informix, Sybase, MS SQL Server (Microsoft), DB2 (IBM)

OO SGBD: Oracle 8 + (relational obiectual);

Sisteme de gestiune a bazelor de date SGBD

Comunicarea cu baza de date

Definirea structurilor de date

Limbaj de definire a structurilor de date (data definition language – DDL) – permite definirea schemei bazei de date (structurile de date folosite si a legaturilor dintre acestea).

Schema – metadata – nu tb. confundata cu datele (instante ale schemei bazei de date).

Dictionar de date (data dictionary) stocheaza date despre toate obiectele pe care le detine

Manipularea datelor

Limbaj de manipulare a datelor (data manipulation language – DML); permite inserarea, actualizarea, stergerea datelor

Interogarea datelor

Limbaj de interogare a datelor (data query language – DQL)

Controlul datelor

Limbaj de control al datelor (data control language – DCL)

Asigurarea confidentialitatii si securitatii datelor, salvarea datelor si realizarea fizica a modificarilor in baza de date, garantarea integritatii si consistentei datelor in caz de acces concurential

SQL (Structured Query Language)

Limbaj de interogare dar si de definirea, manipularea si controlul datelor

Sisteme de gestiune a bazelor de date SGBD

Securitatea datelor

Mai multe nivele de securitate

Conturi pentru utilizatori, cu parola folosita pentru autentificare

Altele: grupuri, roluri, privilegiile si profilurile; acestea permit nu numai constrangeri ci si stabilirea unei politici de securitate.

Exemplu: o agentie de turism se poate abona la baza de date a unui tour-operator; fiecare angajat al agentiei poate capata, dupa rolul sau in agentia sa, acces la una sau mai multe functii (consultare, rezervare, comanda ferma).

In cele mai multe SGBD-uri, fiecare obiect al bazei de date este detinut de un utilizator, schema proprie utilizatorului continand toate obiectele sale; pentru a accesa un obiect, un utilizator trebuie sa aiba privilegiile necesare. Privilegiile pot fi acordate direct unui utilizator sau pot fi grupate in roluri, care la randul lor pot fi acordate utilizatorului.

Exemplu: un forum de discutii are utilizatori grupati pe roluri ca: administrator, moderator, membru. Fiecare rol poate avea privilegii diferite: administratorul poate configura baza de date (modifica schema, adauga tabele, configureaza interfata), moderatorul poate valida, modifica, sterge postarile membrilor, membrii pot adauga inregistrari.

Sisteme de gestiune a bazelor de date SGBD

Mentinerea si constrangerea integritatii

Integritatea se refera la consistenta si corectitudinea datelor.

Consistenta: datele sunt modelate si implementate la fel pentru toate circumstantele.

Corectitudinea: datele sunt exacte si au inteles si utilitate.

Conditii de integritate:

- Structurale (ex. orice SGBDR trebuie sa impuna mentinerea integritatii relationale);
- Comportament (ex. un camp gen tb. sa contina numai masculin, feminin, neutru).

Cum se mentine integritatea in bazele de date relationale?

Se utilizeaza:

- Constrangeri (constraints);
- Declansatori (triggers).

Sisteme de gestiune a bazelor de date SGBD

Asigurarea accesului concurent la date

SGBD-ul asigura accesul concurent al mai multor utilizatori la baza de date;

Fiecare utilizator trebuie sa aiba o vedere valida si consistenta asupra bazei de date, incluzand si modificarile facute de alti utilizatori; in acelasi timp, procesarea incorecta a datelor trebuie evitata, pentru a nu afecta consistenta datelor sau integritatea acestora.

In functie de complexitatea operatiei de acces concurent, problema gestionarii concurentei se complica:

- Acces concurent a mai multor utilizatori numai pentru consultarea datelor;
- Acces concurent a mai multor utilizatori cu unul dintre ei modificand datele;
- Acces concurent a mai multor utilizatori cu mai multi dintre ei modificand datele;

Pentru ultimele doua, se utilizeaza blocarea datelor (primul utilizator care le acceseaza, le blocheaza); cu cat dimensiunea datelor blocate e mai mica, cu atat gestionarea accesului concurential e mai eficienta.

Sisteme de gestiune a bazelor de date SGBD

Asigurarea accesului concurent la date

Tranzactii

Toate tranzactiile sunt atomice (adica nu poate fi executata partial).

O tranzactie este fie committed (finalizata) fie rolledback (derulata inapoi).

Transaction manager: gestioneaza concurenta si se asigura de integritatea tranzactiilor.

SGBD-ul pastreaza o istorie a tranzactiilor (transaction log) pentru a putea fie sa le finalizeze, fie sa le deruleze inapoi.

Decizia de a face roll back pentru o tranzactie poate fi luata fie de operator, fie automat, de SGBD, daca, de ex., rezultatul operarii datelor in cadrul tranzactiei nu a putut fi salvat pe disc sau daca intervine o alta eroare.

Exemplu: o operatie in care un cont al unui client al unei banci este debitat si un altul este creditat cu aceeasi suma (transfer) trebuie sa fie conceputa ca o operatie atomica (tranzactie); limitari in sistemele deschise;

Sisteme de gestiune a bazelor de date SGBD

Baze de date distribuite

Multime de date corelate logic, dar distribuita pe mai multe servere, interconectate printr-o retea de comunicatie. Din punctul de vedere al utilizatorului, o baza de date distribuita reprezinta o singura baza de date.

Exemple: sistemul de source control pentru o companie de dezvoltare software cu mai multe site-uri; baza de date integrata a bibliotecilor universitare; bazele de date ale unei mari companii comerciale cu sedii si filiale in mai multe locatii; baza de date de stocuri ale unui magazin virtual global.

Avantajele utilizarii bazelor de date distribuite: fiecare locatie isi administreaza propria baza de date, pe care o acceseaza in mod frecvent, avand acces si la datele din celelalte locatii; deconectarea uneia dintre locatii nu afecteaza celelalte date din retea;

Observatie: aplicatiile moderne cu baze de date de mari dimensiuni distribuite (zeci de PB) nu mai folosesc un model relational. Se prefera sisteme bazate pe BigTable, MapReduce (Hadoop).

Modelul relational

- Tabele, randuri, coloane
- SGBDR – Regulile lui Codd

Proiectarea bazelor de date relationale

- Crearea schemei conceptuale
- Crearea design-ului logic al bazei de date
- Normalizarea bazei de date
 - Prima forma normala (1NF – First Normal Form)
 - A doua forma normala (2NF – Second Normal Form)
 - A treia forma normala (3NF – Third Normal Form)
 - Forma normala Boyce-Codd (BCNF – Boyce-Codd Normal Form)
 - A patra forma normala (4NF – Fourth Normal Form)

Modelul relational

Tabele, randuri, coloane

Relatii	↔	Tabele
Atribute	↔	Coloane
Tuple	↔	Inregistrari

Regulile lui Codd

1. **Regula reprezentarii logice a datelor:** Intr-o baza de date relationala, toate datele sunt reprezentate la nivel logic intr-un singur mod, si anume sub forma de valori logice in tabele.
2. **Regula accesului la date:** Toate datele individuale din tabele trebuie sa fie accesibile prin furnizarea numelui tabelului, numelui coloanei si valorii cheii primare.
3. **Regula reprezentarii valorilor necunoscute:** Un sistem relational trebuie sa permita declararea si manipularea sistematica a valorilor Null, cu semnificatia unor valori necunoscute sau inaplicabile.

Modelul relational

Regulile lui Codd (continuare)

- 4. Regula dictionarului de date:** Descrierea bazei de date (dictionarul de date) trebuie sa fie reprezentata la nivelul logic tot sub forma de tabele, astfel incat asupra acesteia sa se poata aplica aceleasi operatii ca si asupra datelor propriu-zise.
- 5. Regula limbajului de acces:** Intr-un sistem relational trebuie sa existe cel putin un limbaj de accesare a datelor, care sa asigure urmatoarele operatii: definirea tabelor de baza si a tabelor virtuale (view-uri, vederi); manipularea si interogarea datelor (atat interactiv cat si prin program); definirea restrictiilor de integritate, autorizarea accesului la date, delimitarea tranzactiilor.
- 6. Regula de actualizare a tabelor virtuale:** Un SGBD trebuie sa poata determina daca o vedere poate sa fie actualizata sau nu.
- 7. Regula manipularii datelor:** Un sistem relational trebuie sa ofere posibilitatea procesarii tabelor nu numai in operatiile de interogare a datelor cat si in cele de inserare, actualizare si stergere.

Modelul relational

Regulile lui Codd (continuare)

8. **Regula independentei fizice a datelor:** Programele de aplicatie nu trebuie sa depinda de modul de stocare si accesare fizica a datelor.
9. **Regula independentei logice a datelor:** Programele de aplicatie nu trebuie sa fie afectate de nici o restructurare logica a tabelor bazei de date care conserva datele.
10. **Regula independentei datelor din punctul de vedere al integritatii:** Regulile de integritate a bazei de date trebuie sa fie definite in limbajul utilizat de sistem pentru definirea datelor si nu in cadrul aplicatiilor individuale; in plus, aceste reguli de integritate trebuie stocate in dictionarul de date.
11. **Regula independentei datelor din punctul de vedere al distribuirii:** Programele de aplicatie nu trebuie sa fie afectate de distribuirea pe mai multe calculatoare a bazei de date.
12. **Regula privind prelucrarea datelor de catre un limbaj de nivel inferior:** Orice limbaj nerelational folosit pentru accesarea datelor trebuie sa respecte aceleasi conditii de integritate ca si limbajul relational de acces.
0. **Regula de baza:** Un SGBD relational trebuie sa fie capabil sa gestioneze baza de date exclusiv pe baza caracteristicilor sale relationale.