



Baze de date relaționale

Noțiuni fundamentale

Ș.I. Pupezescu Valentin

Sisteme de baze de date

Definitie (in sens larg):

O baza de date (database) este o colecție de date corelate din punct de vedere logic, care reflectă un anumit aspect al lumii reale și este destinată unui anumit grup de utilizatori. În acest sens pot fi considerate ca fiind “baze de date”:

- Fișe de evidență (menținute manual)
- Fișiere de documente sau tabele
- Baze de date menținute computerizat

Sisteme de baze de date

Definiție (în sens restrâns):

O bază de date este o colecție de date creată și menținută computerizat, care permite operații de:

- Introducere (insert)
- Stergere (delete)
- Actualizare (update)
- Interogare (query)

Aceste operații se numesc operații CRUD (Create, Read, Update, Delete).

Componentele unui sistem de baze de date

Un **sistem de baze de date** este un sistem computerizat de menținere a evidenței unei anumite activități, folosind baze de date.

Componentele unui sistem de baze de date sunt: hardware, software, utilizatori și date persistente.

Hardware:

- Sistemele de baze de date sunt instalate pe calculatoare de uz general
- Bazele de date sunt memorate fizic ca fișiere hdd-uri, ssd-uri, etc.

Software:

- Sisteme de operare, biblioteci, instrumente de dezvoltare, interfețe
- Sistemul de Gestiune a Bazelor de Date (SGBD) (Database Management System – DBMS) - recepționează cererile utilizatorilor de acces la baza de date, le interpretează, execută operațiile corespunzătoare și returnează rezultatul
- Aplicații de baze de date: (Database Applications) – sunt programe care oferă anumite utilizări ale unei baze de date

Avantajele unui sistem de baze de date

- Compactitate ridicată a datelor
- Reprezentarea unor asocieri complexe între date
- Timp de dezvoltare a bazelor de date redus
- Viteza mare de actualizare și regasire a datelor
- Redundanta controlată a datelor (și cât mai scăzută)
- Flexibilitate, menținerea datelor actualizate la zi
- Independența datelor față de suportul hardware utilizat
- Securitatea datelor: autentificarea utilizatorilor și autorizarea accesului
- Impunerea de restricții (constrângeri) de integritate la introducerea și actualizarea datelor
- Menținerea integrității datelor în caz de defecte: salvare și refacere
- Posibilitatea de partajare a datelor între mai multe categorii de utilizatori
- Posibilitatea de introducere a standardelor

Clasificarea sistemelor de baze de date (1)

Clasificare după modelul de date:

- Modelul relațional (implementat de MySQL, MS SQL Server etc.)
- Modelul obiect-orientat (implementat de Objectivity/DB)
- Modelul obiect-relațional (implementat de Oracle, PostgreSQL)
- Modelul non-relațional / NoSQL (implementat de MongoDB, Apache Cassandra etc.)
- Modelul NewSQL (implementat de VoltDB, Clustrix, NuoDB)
- Modelul de date orientat pe grafuri - Graph (OrientDB, Profium Sense, Stardog)

Clasificare după numărul de utilizatori:

- Sisteme mono-utilizator
- Sisteme multi-utilizator

Clasificarea sistemelor de baze de date (2)

Clasificare după numărul de stații pe care este memorată baza de date:

- Baze de date centralizate
- Baze de date distribuite

Arhitectura client-server:

- Server (back-end): SGBD-ul și baza de date
- Client (front-end): program (programe) de aplicație

Modelul relațional (1)

Modelul relațional: E.F.Codd, 1970 – IBM

O **bază de date relațională** este compusă dintr-o mulțime finită de **relații**

- fiecare relație reprezintă o mulțime (tip) de entități sau o mulțime (tip) de asocieri
- fiecare relație este unică într-o bază de date
- o relație se definește prin intermediul atributelor sale

Atributele unei relații corespund atributelor tipului de entitate sau de asociere pe care îl reprezintă relația respectivă

- fiecare atribut are un nume (A_i) și un domeniu de definiție $D(A_i)$
- pentru o entitate dată, un atribut poate lua o singură valoare (scalar)

Atributele pot fi: simple (un element) sau compuse (o submulțime de attribute)

Domeniu: o mulțime de valori $D = \{d_i \mid i = 1, \dots, n\}$, definit printr-o specificare de tip, unde:

- D este numele domeniului
- d_i este un element al domeniului care satisface anumite constrângeri
- Elementele domeniilor sunt atomice (indivizibile)
- O valoare specială, null, poate aparține oricarui domeniu (înseamnă lipsă de informație sau valoare necunoscută)

Modelul relațional (2)

Schema relației: descriere a unei relații (tipul, intensiunea relației)

Schema relației: $R(A_1, A_2, \dots, A_i, \dots, A_n)$, unde:

- R este numele schemei relației
- lista ordonată a atributelor sale $A_1, A_2, \dots, A_i, \dots, A_n$
- fiecare atribut A_i definit pe domeniul său de definiție, $D(A_i)$
- Gradul relației: numărul de attribute ale schemei acelei relații (n)
- Exemplu: STUDENTI (Nume, Prenume, DataNasterii, Adresa, Facultatea)

O **relație** r definită prin schema $R(A_1, A_2, \dots, A_i, \dots, A_n)$ este:

- o mulțime finită de n-tupluri t
- tuplul t este o listă ordonată de n valori: $t = \langle v_1, v_2, \dots, v_i, \dots, v_n \rangle$, unde $1 \leq i \leq n$
- v_i este o valoare a atributului A_i , $v_i \in D(A_i)$

Relația r(R): r este variabila, instanta a schemei (tipului) R

- Valoarea variabilei: starea sau extensiunea relației
- Numărul de tupluri ale unei relații: cardinalitatea relației
- Fiecare tuplu este unic într-o relație (nu există tupluri duplicat)
- Corespondența: relație \rightarrow mulțime de entități (sau de asocieri); tuplu \rightarrow entitate

În mod curent: se folosește R atât pentru schemă cât și pentru relația însăși.

Reprezentarea relațiilor prin tabele

Un tabel (table) = reprezentarea grafică a unei relații; aceasta este compusă din:

- Numele tabelului - identic cu numele relației
- Coloanele corespund atributelor relației
- Capul tabelului - conține numele atributelor (coloanelor) -> schema relației
- O mulțime de linii, fiecare linie corespunzând unui tuplu -> starea relației
- Valori ale atributelor fiecarui tuplu
- Exemplu: Tabelul care reprezintă relația (starea relației) STUDENTI

The diagram illustrates the components of a table. Labels with arrows point to specific parts of the table:

- Numele** points to the table name **STUDENTI**.
- Coloane - Atribute** points to the column headers: **Nume**, **Prenume**, **DataNasterii**, **Adresa**, and **Facultatea**.
- Valori atribute** points to the data values within the table rows.
- Capul tabelului** points to the header row.
- Linii - tupluri** points to the data rows.

Nume	Prenume	DataNasterii	Adresa	Facultatea
Anghelescu	Octavian	1999	Bucuresti	ETTI
Beldiman	Cristina	1998	Bucuresti	ETTI
Boeru	Marius	1999	null	ETTI

Constrângeri de tuplu (1)

O **supercheie** (superkey) este o submulțime SK de attribute ale unei relații care prezintă proprietatea de unicitate (orice combinație de valori ale atributelor supercheii este unică pentru orice stare a relației).

Dacă se cunoaște valoarea (combinația de valori ale atributelor) supercheii, atunci acel tuplu poate fi identificat în mod unic.

Orice relație are cel puțin o supercheie, care este mulțimea tuturor atributelor sale.

O **cheie candidată** (candidate key) este o supercheie ireductibilă:

- Unicitate: nu există două tupluri diferite ale relației care să conțină aceeași combinație de valori ale atributelor cheii CK;
- Ireductibilitate: nu există nici o submulțime proprie, nevidă a cheii CK care să aibă proprietatea de unicitate.

Constrângeri de tuplu (2)

O relație = mulțime de tupluri -> tuplurile unei relații trebuie să fie distincte (nu pot exista două sau mai multe tupluri identice).

Pentru ca tuplurile unei relații să fie distincte se folosește câte o cheie primară (primary key) în fiecare relație.

O **cheie primară** PK a unei relații este un atribut (simplu sau compus) al acelei relații care are proprietatea de unicitate, adică fiecare valoare a cheii primare este unică în acea relație. Aceasta înseamnă că:

Nu există două tupluri distincte (diferite) care să aibă aceeași valoare a cheii primare (sau combinație de valori) pentru orice stare a relației, adică:

$t_i[PK] \neq t_j[PK]$ dacă $i \neq j$, unde t_i și t_j sunt 2 tupluri diferite ale relației

Proprietatea de unicitate a cheii primare este o constrângere de integritate a tuplurilor: fiecare tuplu poate fi identificat în mod precis și se păstrează integritatea acestuia, dacă se cunoaște valoarea cheii sale primare.

Cheia primară este o constrângere implicită: se definește de proiectant la crearea tabelului și este verificată de SGBD (să nu existe duplicate, etc)

Cheia primară mai are următoarele restricții:

- Este ireductibilă: nu există o submulțime proprie nevidă a cheii PK care să aibă proprietatea de unicitate
- Nici o valoare a atributelor cheii primare nu poate fi modificată prin operații de actualizare (UPDATE)
- Nu se admit valori de NULL pentru nici unul dintre attributele cheii primare

Constrângeri de tuplu (3)

Se pot defini chei primare naturale sau chei primare artificiale, cu condiția ca acestea să îndeplinească condiția de unicitate.

O **cheie primară naturală** este un atribut (simplu sau compus) al relației:

- reprezintă o proprietate a tipului de entitate (sau asociere) reprezentat de acea relație
- are în mod natural valori unice: nu există două tupluri cu aceeași valoare a cheii primare, deoarece nu există două entități cu aceeași valoare a proprietății respective.

O **cheie primară artificială** este un atribut (de obicei simplu) care nu reprezintă o proprietate a tipului de entitate sau asociere reprezentat de relație, ci se adaugă în schema relației special pentru identificarea unică a tuplurilor.

Cheia străină

Fie două relații R1 și R2, între care exista o asociere cu raportul 1: N.

O **cheie străină** (foreign key) este o submulțime FK de attribute ale relației R2

care referă cheia CK din relația R1 și satisface următoarele condiții:

- attributele cheii străine FK sunt definite pe domenii compatibile cu cele ale atributelor cheii candidate CK a relației R1.
- valorile atributelor FK într-un tuplu din relația R2, fie sunt identice cu valorile atributelor CK ale unui tuplu oarecare din starea curentă a relației R1, fie sunt NULL

Două domenii sunt compatibile dacă ele sunt compatibile din punct de vedere al tipului de date și compatibile semantic (are sens să fie comparate).

În limbajul SQL verificarea domeniilor se rezumă la verificarea tipurilor de date, iar compatibilitatea semantică trebuie să fie asigurată de proiectant.

Cheia străină reprezintă o constrângere referențială între cele 2 relații. Relația referită (R1) – relație părinte, relația care referă (R2) – relație fiu.

Menținerea integrității referențiale (1)

Integritatea referențială (referential integrity) este proprietatea bazei de date prin care orice cheie străină:

- fie are o valoare care se regăsește printre valorile cheii candidate referite;
- fie are valoarea NULL.

Pentru menținerea integrității referențiale trebuie să fie impuse restricții operațiilor de modificare a stării relațiilor (INSERT, DELETE, UPDATE).

Restricțiile care se impun operațiilor de modificare a relațiilor depind de rolul relației (relație care referă, relație referită, sau poate avea ambele roluri)

Operația INSERT:

- Într-o relație care nu referă altă relație, inserarea se poate face fără restricții
- Într-o relație care referă (care conține o cheie străină): SGBD-ul permite introducerea

unui tuplu nou numai dacă: (a) valoarea cheii străine a tuplului nou este NULL sau (b) există o valoare a cheii referite egală cu valoarea cheii străine a tuplului nou

Operația DELETE:

- Într-o relație care nu este referită ștergerea se poate face fără restricții;
- Într-o relație referită se admite: ștergere restricționată, ștergere în cascadă, anularea (SET NULL) a cheilor străine care refereau tuplul șters;

Ștergerea restricționată interzice ștergerea unui tuplu din relația referită dacă acesta este referit de un tuplu din relația care o referă

Menținerea integrității referențiale (2)

Ștergerea în cascadă permite ștergerea unui tuplu din relația referită; dacă tuplul șters era referit de unul sau mai multe tupluri, atunci se șterg și acestea din relația care o referă; dacă tuplurile șterse din relația care referă sunt, la rândul lor referite de alte tupluri din alte relații, atunci trebuie să fie șterse și acestea, ș.a.m.d.; se execută deci o ștergere în cascadă.

Operația UPDATE este o ștergere urmată de o inserare, deci restricțiile de actualizare reprezintă combinația restricțiilor de inserare și de ștergere.

În limbajul SQL se specifică opțiunile ON DELETE și ON UPDATE constrângerii de cheie străină; valorile posibile ale acestor opțiuni sunt:

- RESTRICT – ștergerea restricționată (este valoare implicită);
- CASCADE – ștergerea în cascadă;
- SET NULL – anularea cheilor străine care refereau tuplul șters;
- NO ACTION – se admit valori care nu respectă integritatea relațională.