

Baze de date

Litera A

active database = bază de date activă

Un sistem de gestiune a bazelor de date care, pe lângă facilitățile obișnuite (asigurarea persistenței datelor, limbaje de interogare și definire a datelor, gestiune tranzacțională a datelor, etc), permite specificarea și declanșarea unor operații atunci când se îndeplinesc anumite condiții. De exemplu, condițiile pot fi: inserare, ștergere sau modificarea unei înregistrări într-o tabelă a bazei de date. Operațiile care se pot efectua sunt programate de către utilizator, și sunt folosite în general pentru testarea și asigurarea îndeplinirii unor constrângeri de către datele stocate.

aggregate = colecție

Gruparea mai multor entități, în general de același tip, într-o singură entitate (de tip colecție). Colecțiile sunt folosite în sistemele de gestiune a bazelor de date ce pot stoca date complexe sau obiecte; în aceste sisteme, colecțiile sunt de obicei constructori de bază, ce permit realizarea de noi tipuri definite de utilizator (de obicei ca o colecție de tipuri simple și/sau alte tipuri complexe definite de utilizator). Modul de implementare fizică a unei colecții este lăsat la latitudinea programatorului sau a mediului de dezvoltare a aplicației.

Exemplu un sistem de gestiune a bazelor de date orientate spre obiecte permite definirea unei clase "Persoană" și a unei variabile colecție de persoane, "studenți":

```
class Persoana
```

```
type tuple (nume: string, cetatenie: string, an_nastere:integer);
```

```
name studenti: set (Persoana);
```

alphabetize (to) = a ordona alfabetic

Operație prin care datele primare stocate în baza de date sunt reordonate, astfel încât acestea să poată fi parcurse în ordine alfabetică. Ordonarea alfabetică se poate face numai pentru câmpuri de tip șir de caractere (de exemplu, după numele sau prenumele persoanelor). Un termen echivalent în limba engleză, ce desemnează aceeași operație, este și termenul "to alphasort".

alphanumeric = alfanumeric

Tip de date ce poate conține atât litere, cât și cifre. Folosirea datelor alfanumerice în baze de date simplifică operațiile în care apar date mixte (litere și cifre), ca de exemplu indexarea după o cheie multiplă ce conține atât câmpuri numerice, cât și șiruri de caractere.

Exemplu: identificarea unei persoane pe baza buletinului său de identitate se face specificând seria (2 litere) și numărul (6 cifre) buletinului. Astfel, rezultă o cheie alfanumerică după care se poate căuta orice persoană.

append (to) = a adăuga

Operație prin care, la datele stocate într-o bază de date, se adaugă o nouă înregistrare cu informații. Adăugarea înregistrării nu se face neapărat la sfârșit, ci eventual în locul unei înregistrări șterse sau în prima poziție disponibilă. La efectuarea unei interogări, datele nu sunt obținute neapărat în ordinea în care au fost adăugate (ordinea poate depinde și de folosirea unui index).

archived copy = copie de arhivă

O copie a unei baze de date, efectuate periodic, în scopul de a permite refacerea datelor valabile la un anumit moment de timp (în scopul de a extrage acele informații care au dispărut ulterior din baza de date). O copie de arhivă se poate utiliza și pentru refacerea (măcar parțială) a bazei de date în cazul unei erori irecuperabile (datorate în general a mediului de stocare a datelor

- discul magnetic). Copia de arhivă se poate efectua pe o bandă, pe un alt disc magnetic de rezervă, pe un disc optic, etc.

Armstrong's axioms = axiomele lui Armstrong

Un set de axiome care precizează modul în care, pornind de la o mulțime de dependențe funcționale date, se pot obține noi dependențe funcționale.

Exemplu

R1. (reflexivitate) dacă $X \twoheadrightarrow Y$, atunci $X \twoheadrightarrow X$
 R2. (augmentare) $\{X \twoheadrightarrow Y\} \models XZ \twoheadrightarrow YZ$
 R3. (tranzitivitate) $\{X \twoheadrightarrow Y, Y \twoheadrightarrow Z\} \models X \twoheadrightarrow Z$

atomic = atomic

O tranzacție efectuată într-o bază de date este atomică dacă se garantează că aceasta odată începută, ea nu va mai fi întreruptă de o altă tranzacție. Atomicitatea este o proprietate esențială în asigurarea consistenței unei baze de date exploatate de mai mulți utilizatori. În cazul bazelor de date orientate spre obiecte, datorită apariției unor tranzacții foarte lungi, atomicitatea poate conduce la blocaje pe durate mari de timp, de aceea se folosesc modele tranzacționale bazate pe tranzacții imbricate (nested transactions).

attribute = atribut

O proprietate a unei entități (memorate într-o bază de date), care are o anumită valoare la un moment dat. Atributele pot fi atomice (au o structură simplă, predefinită: întregi, șiruri de caractere, etc) sau compuse (sunt alcătuite la rândul lor din alte atribute). Atributele sunt folosite în cazul bazelor de date orientate spre obiecte pentru a specifica diverse proprietăți ale obiectelor memorate (inclusiv faptul că obiectul este format din alte obiecte). De asemenea, în modelul entitate- legătură, atributele sunt asociate entităților pentru a descrie structura acestora și a permite extragerea schemei de relație necesare construirii bazei de date.

authorization = autorizație

Permisiune acordată unui utilizator de a accesa anumite informații dintr-o bază de date. Un sistem de gestiune a bazelor de date poate permite sau interzice unui utilizator să citească sau să modifice informații dintr-o anumită tabelă a bazei de date, cu scopul de a preveni și împiedica folosirea neautorizată a datelor. De exemplu, un angajat al serviciului financiar al unei companii nu ar trebui să poată să modifice salariul unui angajat, sau să poată accesa informații confidențiale despre firma la care lucrează.

Litera B

back up (to) = a salva

Operație care constă în duplicarea unor informații pe un alt suport de date decât cel de lucru curent. Salvarea se face în scopul păstrării unei baze de date de rezervă, care să poată fi utilizată în cazul în care apare o eroare ireparabilă (de exemplu, defectarea discului magnetic). Prin salvare se asigură păstrarea informațiilor valabile la o anumită dată, informațiile introduse între momentul salvării și momentul apariției defecțiunii se pierd, dar astfel se păstrează cea mai mare parte a datelor. Pentru minimizarea pierderilor accidentale de date, este necesară elaborarea unei politici de salvare periodică a bazei de date (de exemplu, la sfârșitul unei zile de lucru, la sfârșitul săptămânii și după ce s-au efectuat operațiile de "închidere" a lunii curente).

binary search = căutare binară

Metodă de căutare a datelor într-un fișier ordonat crescător în funcție de valoarea unei chei. Metoda constă în împărțirea fișierului în două jumătăți și compararea valorii căutate cu valoarea cheii din înregistrarea aflată în poziția mediană. În funcție de relația de inegalitate rezultată, se va continua căutarea în subintervalul din stânga sau în cel din dreapta, metoda aplicându-se apoi succesiv și împărțind în două subintervalul rămas.

block = bloc (de date)

Unitatea de organizare fizică a transferului de informații între un fișier și memoria principală. Modelul fizic de organizare a unei baze de date permite un bloc de date să conțină mai multe înregistrări, accesul la o înregistrare făcându-se numai prin citirea întregului bloc în memorie; din motive de eficiență a accesului la informații, devine important ca informațiile prelucrate împreună să fie stocate în același bloc de date, citirea unui bloc însemnând că majoritatea înregistrărilor stocate în acesta trebuie să fie prelucrate.

Boyce-Codd normal form = forma normală Boyce-Codd

Formă canonică a unei scheme de relație. O schemă de relație (sau o relație) este în forma normală Boyce-Codd dacă toate dependențele funcționale netriviabile valabile pe mulțimea atributelor sale sunt de forma $X \rightarrow A$, unde X este o cheie primară, iar A este un atribut oarecare.

browse = răsfoire

Metodă de acces la informațiile dintr-o bază de date, care presupune afișarea și parcurgerea lor consecutivă. Pe ecran sunt afișate mai multe înregistrări consecutive, într-un format tabelar, utilizatorul având posibilitatea să navigheze înainte și înapoi prin baza de date. Uneori, este permisă și modificarea unor informații din baza de date răsfoită.

B-trees = arbori B

Metodă de organizare a datelor, folosită în majoritatea sistemelor de gestiune a bazelor de date, și care constă în gruparea cheilor sub forma unui arbore multicăi de căutare, în care fiecare nod trebuie să conțină minim jumătate din numărul maxim permis de chei. Această metodă permite un timp de acces foarte bun la o înregistrare oarecare, dar cu prețul unei sporiri a complexității algoritmilor de căutare și mai ales de inserare și ștergere a înregistrărilor.

Litera C**candidate key = cheie-candidat**

Chei ale unei relații, care nu sunt selectate drept cheie primară. Pentru cheile-candidat, în relațiile asociate pot exista valori necunoscute, sau "valori nule".

catalogue = catalog

O colecție de informații referitoare la o aplicație de baze de date: fișierele de date necesare, programele și formularele ce alcătuiesc aplicațiile. De exemplu, un catalog dBASE conține numele tuturor fișierelor bazei de date, împreună cu o scurtă descriere a fiecărui fișier, programele dBASE ce exploatează aceste fișiere, formularele ecran folosite pentru vizualizarea și introducerea datelor, formularele de tipărire la imprimantă, etc. Catalogele permit gruparea aplicațiilor pe categorii de prelucrări (poate exista astfel un catalog pentru aplicația de salarizare a personalului, un altul pentru aplicația de gestiune a stocurilor, un altul pentru evidența personalului unei întreprinderi, etc).

chained record = înregistrare înlănțuită

Tehnică de memorare a înregistrărilor de date, în care se memorează, pe lângă datele utile, și un indicator către următoarea înregistrare din aceeași categorie, ceea ce permite parcurgerea secvențială rapidă a tuturor înregistrărilor dintr-un astfel de lanț.

change file = fișier de actualizări

Un fișier special al unei baze de date, în care se înscriu informațiile care au fost actualizate în baza de date. Folosirea unui fișier de actualizări, împreună cu o tehnică de salvare periodică a bazei de date, permite refacerea automată a acesteia în cazul unor erori ireparabile ce au distrus informațiile. În principiu, fiecare înregistrare a unui fișier de actualizări conține: data și ora actualizării, identificatorul utilizatorului care a efectuat-o, numele câmpului actualizat, vechea valoare existentă în baza de date și valoarea nouă, actualizată.

conceptual schema = schema conceptuală

O specificare a structurii bazei de date, rezultată în urma analizei problemei reale, care se interpune între schema fizică (ce dă detalii de stocare fizică a datelor pe disc) și schema externă (modul în care utilizatorul vede datele și cum i se prezintă aplicația). Schema conceptuală este o descriere la nivel logic al organizării datelor, specificând relațiile dintre înregistrările fizice și regulile de integritate ale bazei de date.

concurrency = concurență

Modalitate de procesare a interogărilor într-o bază de date, care asigură execuția cvasi-simultană a acestora. Cererile sunt descompuse în unități atomice de prelucrare, numite tranzacții, care se execută întrețesut, astfel încât rezultatele se pot obține în paralel. Execuția concurentă a cererilor (îndeosebi a actualizărilor) pune probleme deosebite de asigurare a consistenței bazei de date, fiind necesară o serializare a tranzacțiilor (adică execuția lor într-o anumită ordine, care să garanteze efectuarea corectă a modificărilor).

consistency = consistență

Proprietate a unei baze de date care se referă la faptul că datele conținute nu sunt contradictorii. Consistența unei baze de date este asigurată prin constrângerile de integritate (asertiuni care se aplică informațiilor din baza de date, și care trebuie să fie întotdeauna adevărate).

consistent database = bază de date coerentă

Bază de date pentru care mulțimea restricțiilor de integritate este respectată de către datele conținute în bază.

constraint = constrângere

Un predicat logic care este adevărat pentru toate informațiile din baza de date. Există o varietate de constrângeri: simpla specificare a tipurilor datelor sau a subdomeniilor de valori, unicitatea valorilor dintr-un anumit câmp al înregistrărilor, "legarea" unei valori dintr-o înregistrare de o valoare dintr-o altă înregistrare (de obicei o cheie), etc.

corrupted = eronat, alterat

Stare a bazei de date în care informațiile stocate în aceasta sunt inutilizabile. Alterarea unei baze de date poate apare în general datorită defecțiunilor hardware, dar și datorită unor pene ascunse în sistemul de gestiune a bazei de date sau chiar a sistemului de operare. Refacerea unei baze de date alterate se poate realiza folosind programe speciale, furnizate de producătorul sistemului de gestiune, și nu garantează refacerea în totalitate a informațiilor stocate (uneori se pot recupera numai anumite tabele ale bazei de date).

Litera D**data abstraction = abstractizare a datelor**

Un principiu folosit pentru a ascunde utilizatorului detaliile legate de modul de memorare fizică a datelor. Abstractizarea datelor se realizează în principal printr-un model al datelor pus la dispoziția utilizatorului de către sistemul de gestiune a bazei de date; acest model trebuie să fie suficient de puternic pentru a permite reprezentarea tuturor conceptelor și suficient de simplu pentru a permite gestiunea ușoară a întregii aplicații.

data analysis = analiza datelor

Fază inițială a proiectării unei aplicații de baze de date, în care se urmărește izolarea entităților reprezentative aplicației, a relațiilor dintre acestea și a caracteristicilor semantice (cum ar fi constrângerile la care sunt supuse datele). În urma analizei datelor, rezultă de obicei un set de specificații de proiectare constând în: diagrame (de exemplu, diagrama entitate-legătură), asertiuni asupra datelor, un dicționar de date, etc.

data description language (DDL) = limbaj de descriere a datelor

Un limbaj utilizat pentru descrierea structurii unei baze de date; uneori, descrierea se referă atât la schema conceptuală cât și la schema fizică a bazei de date.

Exemplu: crearea unei tabele cu informații despre o secție se poate realiza astfel:

```
CREATE TABLE SECTII (NUME_SECT CHAR(15), NR_ANGAJATI, SALARIU_TOTAL REAL);
```

data entry = introducere de date

Una din operațiile frecvente ce se execută asupra unei baze de date, și care constă în adăugarea de noi informații. Introducerea datelor necesită pe de o parte existența unor formulare de introducere a datelor (proiectate cu grijă pentru a corespunde modelului mental pe care utilizatorul îl are în privința aplicației), iar pe de altă parte implică verificarea cât mai bună a corectitudinii datelor introduse și a consistenței cu informațiile deja stocate în baza de date.

data field = câmp de date

Un câmp al unui formular de introducere a datelor în care utilizatorul poate introduce o informație. Valoarea din câmpul de date poate să corespundă unui câmp dintr-o tabelă a bazei de date (caz în care va fi stocată ca atare), sau poate servi la calcularea valorilor care se vor memora în alte câmpuri din baza de date.

data independence = independență față de date

O caracteristică a unui sistem de gestiune a bazelor de date, care permite schimbarea unor caracteristici ale bazei de date fără a fi necesară și o schimbare în arhitectura acesteia sau în programele de aplicație. Independența față de date poate să fie logică (ceea ce permite schimbarea schemei conceptuale fără să fie necesară schimbarea schemei externe sau a programelor ce folosesc schema externă), sau poate să fie fizică (ceea ce permite schimbarea schemei interne fără să fie necesară și schimbarea schemei conceptuale)

data integrity = integritatea datelor

Reprezintă acuratețea datelor și menținerea semanticii acestora, în special după datele au fost mutate sau prelucrate. Menținerea integrității datelor în sistemele de gestiune a bazelor de date se realizează prin: validarea conținutului câmpurilor individuale și în corelație cu alte câmpuri din aceeași tabelă sau dintr-o altă tabelă și verificarea faptului că actualizarea bazei de date s-a făcut cu succes și corect din punctul de vedere al integrității.

data manipulation language (DML) = limbaj de manipulare a datelor

Un limbaj pe care utilizatorii unei baze de date îl folosesc pentru a interoga informațiile din aceasta, precum și pentru a adăuga noi informații, a șterge informații existente sau a modifica unele din informațiile din baza de date.

Exemplu: creșterea salariului cu 10% a angajaților unei secții se face după 10 ani de vechime în munca, utilizând următoarea cerere SQL:

```
UPDATE ANGAJATI  
SET ANGAJATI.SALARIU = ANGAJATI.SALARIU * 1.1  
WHERE ANGAJATI.VECHIME > 10;
```

data model = model al datelor

Constă într-o mulțime de concepte care pot fi utilizate pentru a descrie structura unei baze de date (adică tipurile de date, relațiile între entitățile bazei de date și constrângerile care se aplică acestora). Majoritatea modelelor de date includ și o mulțime de operații ce se pot efectua asupra bazei de date pentru regăsirea și actualizarea informațiilor.

data record = înregistrare de date

O structură de date compusă din mai multe entități numite câmpuri. Fiecare câmp are un nume și un tip (care dă domeniul de

valori ce pot fi stocate în acel câmp). În cadrul unui program se pot referi atât câmpuri individuale dintr-o înregistrare, cât și întreaga înregistrare. Înregistrările reprezintă unitatea de memorare fundamentală a informațiilor într-o bază de date.

data reliability = siguranța datelor

Un concept care se referă la faptul că informațiile stocate într-o bază de date trebuie să fie protejate împotriva unor accidente cum ar fi: căderea suportului magnetic (a discului magnetic), căderea tensiunii de alimentare, etc. Siguranța datelor este realizată în principal prin: copii de siguranță ale bazei de date, jurnalizarea extensivă a operațiilor efectuate asupra bazei de date, "ogîndirea" discului magnetic pe un alt disc, etc.

data retrieval = regăsirea datelor

Operație asupra unei baze de date, care constă în selectarea anumitor informații ce corespund cerințelor utilizatorului. Regăsirea datelor este operația cea mai frecventă care se face în exploatarea bazei de date. Datele care trebuie selectate din baza de date sunt specificate folosind expresii logice ce conțin denumiri de tabele din baza de date și câmpuri ale acestor tabele, operatori de comparație între aceste câmpuri și operatori asupra mulțimilor de tupluri. Un sistem de gestiune a bazelor de date folosește mecanisme speciale de optimizare a interogărilor pentru a regăsi în mod eficient datele cerute.

Exemplu: o interogare în baza de date pentru aflarea numelui și prenumelui angajaților care lucrează într-o secție mică (cu mai puțin de 20 de angajați) este exprimată prin:

```
SELECT NUME, PRENUME FROM ANGAJATI, SECTII  
WHERE (ANGAJATI.NUME_SECT = SECTII.NUME_SECT) AND (SECTII.NR_ANGAJATI <= 20);
```

data security = securitatea datelor

Un concept care se referă la protecția datelor împotriva acceselor neautorizate. Protecția se poate realiza la nivelul unei tabele (prin definirea așa-numitor "vederi" - views în limba engleză), sau la nivelul câmpurilor unei tabele (folosind mecanismele speciale de acces puse la dispoziția programatorului de către sistemul de gestiune a bazei de date). În lipsa unor mecanisme de asigurare a securității datelor, acestea ar putea fi modificate de către orice utilizator, ceea ce ar conduce la inconsistențe în baza de date (de exemplu, un angajat ar putea să-și mărească singur salariul, ceea ce ar duce la pierderi financiare pentru întreprindere).

data transaction = tranzacție de date

Unitatea de bază a efectuării tuturor operațiilor asupra unei baze de date. O tranzacție este executată atomic (adică tranzacția se execută în totalitate sau deloc). Tranzacțiile care sosesc simultan de la mai mulți utilizatori sunt serializate, astfel încât execuția acestora să se facă într-o ordine care să garanteze consistența bazei de date.

database = bază de date

O colecție de informații funcționale, memorate pe un suport persistent, împreună cu o mulțime de operații care permit exploatarea acestora. Operațiile cele mai frecvente ce se execută asupra datelor sunt: căutarea informațiilor, inserarea de noi informații și ștergerea de informații existente în baza de date, modificarea unor informații din baza de date, recombinația datelor după anumite reguli (de exemplu, sortarea informațiilor sau combinarea datelor memorate în mai multe tabele).

database administrator (DBA) = administrator al bazei de date

Funcția de administrator al bazei de date poate fi îndeplinită de o persoană sau grup de persoane responsabile pentru crearea și întreținerea unei baze de date. Uzual, o astfel de persoană proiectează schemele conceptuale și fizice a bazei de date, stabilește drepturile de acces a diferiților utilizatori la informațiile din baza de date, stabilește constrângerile de integritate ce se aplică bazei de date și monitorizează accesul la aceasta în scopul îmbunătățirii performanțelor.

database engine = motor de bază de date

Un program sau colecție de programe proiectate pentru a facilita gestiunea și accesul la o bază de date. Un motor de baze de date trebuie să pună la dispoziția utilizatorilor limbaje de definire și de manipulare a datelor, mecanisme de definire a

schemelor conceptuală și fizică, posibilități de conectare la diverse baze de date și mecanisme de control și recuperare a erorilor.

Exemplu: pentru calculatoarele personale există mai multe motoare de baze de date, dintre care amintim: JET (proiectat de Microsoft și folosit de sistemul Access de gestiune a bazelor de date), BDE - Borland Database Engine (realizat de Borland și folosit de produsele Paradox și Delphi).

database machine = mașină pentru bază de date

Un calculator special conceput pentru a ușura accesul la o bază de date și a mări viteza cererilor.

database management system (DBMS) = sistem de gestiune a bazelor de date

Un set de programe care se interpun între baza de date fizică și utilizatori, permițându-le acestora să o folosească la nivel conceptual. Utilizatorul adresează cereri sistemului de gestiune a bazelor de date (fără să fie interesat de localizarea fizică a fișierelor de date, de existența indecșilor sau de formatul fizic al datelor), iar acesta le translatează în operații de nivel scăzut, gestionând în același timp într-un mod centralizat cererile simultane sosite de la mai mulți utilizatori și permițând autorizarea și validarea accesului diverșilor utilizatori la baza de date.

database manager = gestionar al bazei de date.

În funcție de context, poate desemna fie un sistem de gestiune a bazelor de date, fie un administrator al bazei de date.

database schema = schema bazei de date

O descriere a unei baze de date, care specifică exact structura bazei de date (structura conceptuală sau structura fizică), tipurile de date ce vor fi memorate în baza de date, restricțiile la care aceste date trebuie să se supună, etc. Trebuie făcută distincția între schema bazei de date, ca o descriere teoretică a structurii unei baze de date, și baza de date propriu-zisă, care conține datele propriu-zise.

decomposition = descompunere

Înlocuirea unei relații cu schema R, cu o colecție de relații cu sub-schemele R1, R2, ..., Rn, astfel încât reuniunea acestor sub-scheme are ca rezultat R.

dependencies preserving decomposition = descompunere care conservă dependențele.

Descompunerea (R1, R2, ..., Rn) a unei scheme de relație R, astfel încât închiderea tranzitivă a dependențelor funcționale valabile pe R este aceeași cu rezultatul reuniunii dependențelor funcționale valabile pe subschemele descompunerii.

dependency = dependență

O relație logică între atributele unei tabele dintr-o bază de date. Noțiunea de dependență funcțională este utilizată ca o modalitate de asigurare a consistenței informațiilor stocate într-o bază de date, iar interpretarea dependențelor funcționale asociate unei baze de date reprezintă tocmai semantica informațiilor stocate în aceasta. Pe baza dependențelor au fost propuse mai multe "forme normale" ale unei baze de date, care sunt folosite în procesul de "normalizare" a unei baze de date, adică aducerea acestora într-o formă care respectă una din formele normale și, în plus, permite refacerea informațiilor inițiale cu păstrarea dependențelor și a altor proprietăți (ca de exemplu, filtrarea fără pierderi).

domain = domeniu

O mulțime de valori pe care le poate lua un câmp al unei tabele a bazei de date. Stabilirea corectă a domeniilor este importantă atât pentru definirea schemei bazei de date, cât și pentru exprimarea cererilor (în cazul utilizării unui limbaj de tipul calculului orientat pe domenii).

Litera F

Fifth normal form = forma normală 5 (a cincea formă normală)

Formă canonică a unei scheme de relație. O schemă de relație (sau o relație) este în forma normală 5, în raport cu o mulțime D de dependențe funcționale, plurivalente și joncționale, dacă pentru orice dependență joncțională $*(R_1, R_2, \dots, R_n)$ din închiderea D+, unde !! Formula 1 !! iar !! Formula 2 !! , fie $*(R_1, R_2, \dots, R_n)$ este trivială, fie fiecare R_i este cheie primară pentru R.

First normal form = forma normală 1 (prima formă normală)

Formă canonică a unei scheme de relație. O schemă de relație (sau o relație) este în forma normală 1 dacă atributele sale au numai valori atomice.

file = fișier

În modelul fizic de organizare a unei baze de date, noțiunea de fișier reprezintă o colecție de înregistrări care au toate același format. Un fișier constituie deci reprezentarea fizică a unei relații din modelul relațional. Înregistrările unui fișier pot fi accesate în orice ordine, fișierul nefiind văzut ca un șir de elemente (ca de exemplu un fișier text, care este un șir de caractere, accesul la un caracter sau la o linie făcându-se numai prin citirea caracterelor anterioare).

form = formular

Desemnează un element folosit pentru introducerea și modificarea informațiilor dintr-o bază de date. Formularele sunt proiectate odată cu programele de exploatare a bazei de date (deseori se folosesc instrumente speciale pentru crearea de formulare), și sunt afișate pe ecran pe parcursul execuției programului. Un formular trebuie să satisfacă niște cerințe: să fie explicit, să aibă o formă familiară utilizatorului, să nu fie foarte încărcat cu informație, să facă o validare a datelor introduse și să semnaleze utilizatorului eventualele erori, etc.

fourth generation language (4GL) = limbaj de programare din generația a patra

Un concept ce desemnează o categorie de limbaje orientate spre descrierea mai degrabă a relației dintre datele de ieșire și cele de intrare, și nu pe descrierea modului în care se obțin datele de ieșire. Limbajele de generația a patra sunt limbaje declarative, și permit o mare flexibilitate în scrierea programelor.

fourth normal form = forma normală 4 (a patra formă normală)

O schemă de relație R (sau o relație), care satisface mulțimea dependențelor funcționale și dependențelor plurivalente valabile pentru atributele din R, este în forma normală 4, dacă orice dependență plurivalentă $X \twoheadrightarrow Y$ este trivială (X, Y sunt incluse în R), sau dacă X este o cheie primară pentru R.

functional dependency = dependență funcțională

O clasă de dependențe care se referă la relațiile existente între atributele unei tabelă din baza de date, relații privite prin prisma semnificativității informațiilor memorate în acea tabelă. Dependențele funcționale sunt specifice schemei de relație a bazei de date și nu unei baze de date particulare, ceea ce implică faptul că orice bază de date ce se încadrează în schema respectivă va respecta acele dependențe funcționale. Formal, dacă X și Y sunt submulțimi ale mulțimii de atribute care constituie schema de relație R, dependența funcțională "Y depinde de X" sau "X determină Y" se reprezintă prin "X \rightarrow Y".

Litera H**hashed file = fișier organizat ca hash**

Un fișier cu structură specială, în care înregistrările sunt grupate în funcție de valoarea cheii fiecărei înregistrări și de valoarea unei funcții de dispersie aplicată asupra valorii cheii. În principiu, funcția de dispersie trebuie să asigure clasificarea fiecărei înregistrări într-un grup cu cât mai puține înregistrări, numărul mare al acestora crescând timpul de căutare.

hierarchical database = bază de date ierarhică

O bază de date în care relațiile între înregistrări sunt organizate sub forma unui arbore. Bazele de date ierarhice sunt descrise formal de modelul ierarhic; un exemplu de sistem de gestiune a bazelor de date ierarhice este sistemul IMS (Information Management System), realizat de IBM.

Litera I

incremental data = date incrementale

Date care sunt construite prin adăugarea de noi date (calculate) la cele existente. Datele incrementale sunt utilizate pentru calculul vederilor materializate (materialised views) și în diferite sisteme de gestiune a bazelor de date care pot calcula închiderea unei relații.

indexed file = fișier indexat

Structură de memorare a datelor pe disc, care constă în alăturarea unui fișier de date și a unui index primar. Denumită uneori și "isam" (indexed sequential access method - metodă de acces secvențial indexat), metoda permite parcurgerea înregistrărilor unui fișier în mai multe moduri: prin acces direct (căutarea înregistrării ce conține o anumită cheie) și prin acces secvențial (parcurgerea înregistrărilor în ordinea crescătoare sau descrescătoare a valorii cheii, indiferent de ordinea fizică a înregistrărilor în fișier).

insert (to) = a insera

Operație prin care se adaugă noi informații într-o bază de date, într-o anumită poziție a unei tabeli. Deși inserarea fizică se poate face într-o poziție nespecificată, din punct de vedere logic datele inserate vor apărea în poziția corectă (de exemplu, inserarea unei noi înregistrări se va face în funcție de valoarea conținută în câmpul cheie, astfel încât să se păstreze ordinea înregistrărilor).

integrity constraint = restricții de integritate

Aserțiune care trebuie să fie verificată (respectată) de către date, la momente de timp determinate.

Litera J

join = joncțiune (filtrare)

Operație care se efectuează între două relații (tabele) ale unei baze de date, rezultând o nouă relație ce conține numai acele tupluri care îndeplinesc o condiție specificată. Joncțiunea este un operator al algebrei relaționale și este folosită în interogările care se referă la mai multe tabele ale bazei de date. Se realizează efectuând mai întâi produsul cartezian al celor două relații; din relația rezultată se extrag tuplurile care satisfac condițiile menționate, prin operația algebrică de selecție.

join dependency = dependență joncțională

Fie R o schemă de relație și (R_1, R_2, \dots, R_n) o descompunere cu joncțiune conservativă pentru R , astfel încât

$$R = \bigcup_{i=1}^n R_i$$

Formal, o relație r cu schema R , satisface dependența joncțională $*(R_1, R_2, \dots, R_n)$ dacă joncțiunea naturală a proiecțiilor relației r pe subschemele R_i este r , deci:

$$\bigotimes_{i=1}^n (\pi_{R_i}(r)) = r$$

O dependență joncțională este trivială dacă una din subschemele R_i este chiar R .

Litera K**key = cheie**

Un identificator folosit pentru a referi o înregistrare a unei tabele dintr-o bază de date. De cele mai multe ori, o cheie este formată din conținutul unui singur câmp, dar există și chei compuse, formate dintr-o expresie ce conține mai multe câmpuri (rezultatul fiind de obicei convertit la un șir de caractere, în cazul câmpurilor de tipuri diferite). Cheile sunt folosite nu numai pentru identificarea înregistrărilor, ci și pentru construirea index-urilor (utili pentru mărirea vitezei de căutare).

key field = câmp cheie

Un câmp folosit drept cheie a unei înregistrări. Modificarea acestui câmp va duce automat și la schimbarea cheii, și deci și a poziției relative a înregistrării în indexul cheii respective.

Litera L**log file = fișier de jurnalizare**

Un fișier având un format special, folosit de către sistemul de gestiune a bazelor de date pentru a ține evidența modificărilor efectuate asupra informațiilor din baza de date. În fișierul de jurnalizare sunt scrise toate operațiile care alterează datele (inserări, modificări, ștergeri de înregistrări), astfel încât în cazul unei distrugerii accidentale a bazei de date, aceasta să se poată refăce pe baza ultimei copii de siguranță și a fișierului de jurnalizare.

lossless join decomposition = descompunere conservativă (fără pierderi)

Descompunerea (R_1, R_2, \dots, R_n) a unei scheme de relație R , astfel încât orice relație r cu schema R , se poate reconstitui prin joncțiunea naturală a relațiilor r_i cu schemele R_i ($i = 1, 2, \dots, n$).

Litera M**multivalued dependency = dependență plurivalentă**

Fie X și Y două submulțimi ale mulțimii de attribute care constituie schema de relație R . Atunci, Y depinde plurivalent de X (sau X determină plurivalent Y), dependență notată $X \twoheadrightarrow Y$, dacă fiind date valori ale atributelor din X , există - asociate cu acestea - o mulțime de valori ale atributelor din Y , iar această mulțime este independentă de alte attribute din mulțimea $Z = R - X - Y$.

Litera N**natural join = joncțiune (filtrare) naturală**

Operație efectuată pentru relații ale căror scheme conțin unele attribute comune. După efectuarea produsului cartezian, se extrag numai acele tupluri pentru care valorile atributelor comune coincid; din relația obținută astfel, se elimină una din fiecare pereche de coloane cu același atribut.

nested transactions = tranzacții imbricate

Tranzacții compuse la rândul lor din alte tranzacții. Modelul tranzacțiilor imbricate a apărut ca o necesitate pentru implementarea bazelor de date orientate spre obiecte și a bazelor de date colaborative, în care o entitate poate fi blocată un timp inadmisibil de lung (mult mai mare decât în modelele clasice de gestiune a bazelor de date). Conform acestui model, o tranzacție nu se poate încheia până când toate subtranzacțiile sale nu s-au încheiat.

network model = modelul rețea (de baze de date)

Un model de descriere a bazelor de date, similar modelului ierarhic, dar în care o înregistrare poate avea mai multe înregistrări părinte și mai multe înregistrări fii. Modelul rețea a stat la baza propunerilor grupului de lucru CODASYL DBTG

(Conference On Data SYstems Languages Data Base Task Group) pentru accesul la o bază de date rețea.

Litera O

object = obiect

Un concept ce desemnează o entitate de program caracterizată prin proprietăți (analoge câmpurilor unei înregistrări) și metode (proceduri ce pot fi apelate de către alte obiecte pentru a transmite mesaje obiectului în cauză). Obiectele sunt descrise din mai multe puncte de vedere: al comportării în timp (modelul dinamic), al structurii și relațiilor cu alte obiecte (modelul informațional), al prelucrărilor pe care alesta le efectuează (modelul funcțional), etc. în ultimii ani, obiectele au fost integrate în sistemele de gestiune a bazelor de date, rezultând sisteme ce pot memora persistent colecții foarte mari de obiecte, păstrând toate celelalte caracteristici ale paradigmei orientate spre obiecte (moștenire, agregare, încapsulare, etc).

object-oriented database = bază de date orientată spre obiecte

O bază de date destinată stocării obiectelor, și care poate prelucra eficient obiecte complexe. O astfel de bază de date trebuie să permită stocarea unor date mari (de exemplu, imagini, sunete, etc), fără ca programele de aplicație să sufere penalități de viteză.

OID (Object ID) = identificator de obiect

Un identificator special care identifică în mod unic un obiect stocat într-o bază de date orientată spre obiecte. Identificatoarele de obiecte sunt folosite pentru a individualiza obiectele, în condițiile în care mai multe obiecte pot avea valori egale. De asemenea, mai sunt folosite și pentru gestiunea relației de agregare între obiecte.

on-line database = bază de date interactivă

O bază de date capabilă în orice moment să răspundă cererilor utilizatorului. O astfel de bază de date trebuie să fie localizată pe un server ce funcționează în permanență și eventual conectat într-o rețea de calculatoare (majoritatea bazelor de date interactive sunt accesate de pe alte calculatoare conectate în rețea).

Litera P

physical structure (of a database) = structura fizică (a unei baze de date)

O descriere a unei baze de date din punctul de vedere al localizării fișierelor de date, formatului fizic al înregistrărilor, fișierelor index asociate, etc. Structura fizică nu trebuie folosită pentru proiectarea de aplicații, întrucât acestea vor fi dependente de orice schimbare în structura bazei de date. De asemenea, structura fizică este în general diferită de modul în care proiectantul de aplicații și utilizatorul final văd baza de date (schema conceptuală).

pinned record = înregistrare ancorată

O înregistrare către care există în locuri necunoscute pointeri către aceasta. O înregistrare ancorată, specificată printr-un pointer absolut, nu poate fi mutată, deoarece s-ar pierde legăturile către aceasta. Prin specificarea pointerilor sub forma perechilor (bloc, cheie), înregistrările ancorate pot fi mutate în cadrul aceluiași bloc. În plus, înregistrările ancorate nu pot fi șterse, căci referințele ar indica o nouă înregistrare ce a fost inserată în locul devenit liber, și deci referințele ar fi incorecte.

primary index = index primar

O structură de date asociată unui fișier de înregistrări, care permite regăsirea rapidă a informațiilor. Indexul primar poate fi stocat într-un fișier separat de fișierul de date sau chiar în cadrul acestuia. Pentru organizarea informațiilor se folosește o cheie, indexul primar conținând valorile cheilor și pointeri la înregistrarea care conține fiecare cheie.

primary key = cheie primară

Una dintre cheile existente pentru o relație, aleasă ca bază pentru identificarea și extragerea tuplurilor relației. Pentru o cheie

primară, în relația asociată trebuie să avem întotdeauna valori cunoscute, sau definite ("ne-nule").

prime attribute = atribut prim

Atribut care face parte din orice cheie oarecare a unei scheme de relație (sau relație).

Litera Q

QUEL = limbajul QUEL

Un limbaj de interogare și actualizare a unei baze de date, utilizat de sistemul de gestiune a bazelor de date Ingres.

Exemplu: numele și adresa angajaților laboratorului de cercetare se pot găsi folosind interogarea

```
QUEL:
RANGE OF A IS ANGAJATI,
D IS DEPARTMENT
RETRIEVE (A.NUME, A.ADRESA)
WHERE D.NUME='Cercetare'
```

Litera R

record counter = contor de înregistrări

O valoare asociată unei tabele dintr-o bază de date, care arată câte înregistrări sunt memorate efectiv în tabela respectivă. Contorul de înregistrări este gestionat automat de către sistemul de gestiune a bazei de date, fiind actualizat la fiecare inserare sau ștergere de înregistrări din baza de date.

record format = formatul înregistrării

Reprezintă o descriere a înregistrărilor memorate într-o bază de date, și constă dintr-o listă a câmpurilor din înregistrare, împreună cu domeniile de valori pe care le poate lua fiecare câmp.

record pointer = pointer al înregistrării

Un identificator al unei înregistrări, suficient pentru a localiza o înregistrare căutată într-un fișier. Pointerul înregistrării poate să fie o adresă absolută (ceea ce împiedică mutarea înregistrărilor), sau o pereche de tipul (b,k), unde b identifică un bloc, iar k este o cheie folosită pentru localizarea înregistrării în cadrul blocului b.

recover = refacere

Operație care constă în reconstituirea unei baze de date după un incident care a provocat distrugerea acesteia. Refacerea pornește de la o copie de siguranță și folosește fișierul jurnal pentru a aduce la zi conținutul acesteia.

referential integrity = integritate referențială

O constrângere aplicată asupra a două relații dintr-o bază de date, și care este folosită pentru menținerea integrității datelor stocate în cele două relații. În principiu, integritatea referențială se asigură prin impunerea condiției ca un tuplu dintr-o relație care se referă la un tuplu dintr-o altă relație să aibă un corespondent în cea de-a doua relație (adică să se refere la un tuplu existent).

relation = relație

Desemnează o structură compusă din attribute și tupluri. Relațiile sunt implementate în modelul relațional ca tabele, coloanele tabelului corespunzând atributelor relației, iar liniile acestei tabele constituind tuplurile). În general, se pun diferite restricții asupra structurii relațiilor (de exemplu, câmpurile nu trebuie să se repete, trebuie să fie atomice).

relation scheme = schemă de relație

Colecție de attribute, ale căror valori fac parte din domeniul fiecărui atribut.

relational algebra = algebră relațională

Un formalism ce permite descrierea manipulării tabelor unei baze de date. Algebra relațională se bazează pe o serie de operatori de selecție, proiecție, reuniune, intersecție și filtrare pentru a crea noi relații pe baza celor existente.

relational calculus = calculul relațional

O metodă neprocedurală pentru manipularea relațiilor într-o bază de date. Există două variante, calculul relațional pe tupluri și calculul relațional pe domenii, echivalente între ele și ambele echivalente cu algebra relațională.

relational data base = bază de date relațională

O bază de date construită și exploatată conform modelului relațional. În prezent, cele mai răspândite baze de date sunt cele relaționale.

relational model = model (de date) relațional

Un model de date în care relațiile sunt implementate ca tabele. Sistemul de gestiune a bazei de date relaționale este responsabil pentru căutarea informațiilor cerute, prin inspectarea valorilor din coloanele unei tabele și, eventual, corelarea acestor valori cu valorile din alte coloane ale altor tabele.

relational query = cerere relațională

O interogare a unei baze de date relaționale, făcută într-un limbaj relațional.

Exemplu numele și adresa angajaților laboratorului de cercetare se pot găsi folosind interogarea relațională:

```
SELECT ANGAJATI.NUME, ANGAJATI.ADRESA FROM ANGAJATI, DEPARTAMENT  
WHERE (DEPARTAMENT.NUME='Cercetare') AND (ANGAJATI.DEPT_COD=DEPARTAMENT.COD);
```

retrive (to) = a regăsi (date)

Operație efectuată asupra unei baze de date în scopul localizării anumitor informații. Regăsirea se face folosind un limbaj de interogare a bazei de date (DML), iar timpul de răspuns este influențat de existența unor chei în relațiile bazei de date.

Litera S

second normal form = forma normală 2 (a doua formă normală)

Formă canonică a unei scheme de relație. O schemă de relație (sau o relație) este în a doua formă normală dacă și numai dacă: 1) ea este în prima formă normală; 2) orice atribut care nu aparține unei chei (deci este ne-prim) nu depinde decât de o parte a acestei chei.

secondary indices = indexuri secundare

Structuri de date folosite pentru organizarea informațiilor pentru care există un index primar. Indexurile secundare permit parcurgerea fișierului în altă ordine decât cea dată de indexul primar (indexul secundar conține valorile dintr-un alt câmp al înregistrării decât indexul primar). De obicei, cheia indexului primar trebuie să fie unică, dar pentru indexul primar nu se mai pune aceeași condiție, acesta putând conține chei cu valori multiple.

SQL (Structured Query Language) = limbajul SQL

Un limbaj de gestiune a bazelor de date relaționale, având la bază un alt limbaj (SEQUEL) elaborat de către IBM. SQL conține componente pentru formularea cererilor interactive și pentru definirea, modificarea, controlul și asigurarea securității datelor într-o bază de date relațională. SQL poate fi folosit fie de sine-stătător, fie poate fi înglobat în aplicații (dezvoltate într-un alt limbaj de programare) care necesită accesul la o bază de date.

Exemplu: interogarea unei baze de date pentru găsirea tuturor studenților care urmează cursul de Programarea Calculatoarelor se exprimă astfel:

```
SELECT NUME, PRENUME FROM STUDENTII, CURSURI  
WHERE (CURSURI.NUME = 'Programarea Calculatoarelor') AND (STUDENTII.COD_CURS = CURSURI.COD_CURS)
```

superkey = super-cheie

Mulțime de attribute care include o cheie.

Litera T

table = tabelă

O structură de date folosită în modelul relațional pentru implementarea unei relații. O tabelă este caracterizată prin coloane (corespunzând atributelor) și prin linii (corespunzând tuplurilor).

third normal form = forma normală 3 (a treia formă normală)

Formă canonică a unei scheme de relație. O schemă de relație (sau o relație) este în a treia formă normală dacă și numai dacă:
1) ea este în a doua formă normală; 2) orice atribut ne-prim al schemei de relație nu depinde de un atribut ne-prim al acesteia.

transitive closure = închidere tranzitivă

Menținerea dependențelor funcționale elementare valabile pentru o schemă de relație, obținută prin aplicarea proprietății de tranzitivitate asupra unei mulțimi inițiale de dependențe funcționale elementare.

trigger = declanșator

Un proces care monitorizează îndeplinirea unei anumite condiții de către informațiile dintr-o bază de date. Condiția este specificată ca un predicat logic, iar acțiunea ce trebuie executată la îndeplinirea condiției este specificată în limbajul de manipulare a datelor. Această noțiune a apărut odată cu dezvoltarea bazelor de date active, care permit datelor să capete "inteligență", adică să reacționeze la anumite evenimente.

Litera V

variable-length record = înregistrare cu lungime variabilă

Tip de înregistrare în care numărul de octeți ocupați efectiv este diferit de la înregistrare la înregistrare. O înregistrare cu lungime variabilă trebuie să conțină deci, ca informație de sistem, și numărul de octeți pe care îi ocupă efectiv. Astfel de înregistrări conduc la o ocupare mai judicioasă a spațiului ocupat pe disc de o bază de date, dar pun probleme speciale la regăsirea informației și măresc timpul de căutare.