



Introducere

Noțiuni de bază

- Informații și date
- Sisteme de gestiune a bazelor de date
- Modele de date
- Limbaje și utilizatori
- Avantaje și dezavantaje ale sistemelor de gestiune a bazelor de date

- Disponibilitatea informației și capacitatea de a o gestiona în mod eficient - esențiale în orice activitate
- Organizație \Leftrightarrow sistem informațional - gestionează informația necesară funcționării
- Informația este înregistrată sub formă de date
 - datele necesită o interpretare pentru a furniza informații
- Nu se poate da o definiție exactă a conceptului de dată, precum și a diferenței dintre dată și informație
- **Data nu are nicio semnificație, dar odată interpretată și corelată convenabil, ea furnizează informații ce îmbogățesc cunoașterea**

Exemplu

Șirul de caractere	<i>Popescu Ion</i>
Numărul	<i>123456</i>

„Cine este șeful departamentului de cercetare și care este numărul său de telefon?”

„Persoana Popescu Ion este șeful departamentului de cercetare și numărul său de telefon este 123456”

Definiție. O *bază de date* este o colecție de date utilizată pentru a reprezenta informațiile de interes pentru un sistem informațional

Definiție. Un *sistem de gestiune a bazelor de date (SGBD)* este un sistem software capabil

- să gestioneze colecții de date *mari, partajate și persistente*
- să asigure *corectitudinea și securitatea* datelor
- să fie eficient
- să producă rezultatul scontat

Definiție. O *bază de date (BD)* este o colecție de date gestionate de un SGBD

Caracteristici ale bazelor de date

- BD sunt **mari**
 - în general depășesc memoria principală disponibilă
 - SGBD-ul trebuie să gestioneze datele în memoria secundară ⇒ este limitat doar de de capacitatea dispozitivelor de stocare
- BD sunt **partajate**
 - diverse aplicații și utilizatori trebuie să aibă posibilitatea de a obține accesul la datele de interes
 - “controlul concurenței” – permite accesul partajat la date al mai multor utilizatori ce operează simultan
- BD sunt **persistente**
 - BD au o durată medie de viață ce nu este limitată de o singură execuție a programelor utilizator
 - (datele gestionate de un program în memoria principală au o durată de viață cuprinsă între începutul și sfârșitul execuției programului, nefiind persistente)

Caracteristicile SGBD

- **corectitudinea datelor** - funcții specifice de backup
 - are capacitatea de a conserva conținutul BD (sau cel puțin permite reconstituirea acesteia) în cazul unor defecțiuni hardware sau erori software
- **securitatea datelor** - mecanisme de autorizare
 - fiecare utilizator, recunoscut prin intermediul unui nume de utilizator specific accesului la SGBD, poate efectua numai anumite operații asupra datelor

- **eficiența** - finalizează operațiile utilizând cantitatea adecvată de resurse (timp și spațiu) pentru fiecare utilizator
 - această caracteristică se bazează atât pe tehnicile utilizate în implementarea SGBD-ului, cât și pe modul de proiectare a produsului respectiv

- **productivitatea mărită**
 - *productivitatea* - capacitatea sistemului cu BD de a conduce la realizarea scopurilor utilizatorilor
 - această definiție este generică și nu corespunde unei anumite funcții a SGBD-urilor, dat fiind că un SGBD pune la dispoziția utilizatorilor diverse servicii și funcții
 - sarcina proiectării unei BD și a aplicațiilor ce o utilizează vizează garantarea unei bune productivități a întregului sistem

Model de date - combinație de elemente destinată organizării datelor

Fiecare model de date pune la dispoziția utilizatorilor **mecanisme de structurare**

- similare constructorilor *type* din limbajele de programare - permit definirea unor tipuri noi de date pe baza tipurilor elementare, predefinite

Tipuri de modele de date:

- Modelul de date **relațional**
- Modelul de date ierarhic (structuri de tip arbore și ierarhie)
- Modelul de date rețea (grafuri)
- Modelul de date obiect (paradigma programării orientate obiect)

- **Modelul de date relațional** - propus la începutul anilor '70
 - sistemele reale bazate pe modelul relațional au apărut la începutul anilor '80
 - cel mai răspândit model de date
 - pune la dispoziție constructorul **relație**
 - oferă astfel posibilitatea organizarea datelor sub forma unei colecții de înregistrări având structură fixă

Relația - se reprezintă sub forma unui tabel

- liniile coincid cu înregistrările
- coloanele corespund câmpurilor înregistrării
- ordinea în care apar liniile și coloanele nu este relevantă

Exemplu

Datele referitoare la cursurile universitare și titularii lor, precum și inserarea cursurilor în planul de învățământ al diverselor specializări pot fi organizate cu ajutorul a două relații (tabele) **CURSURI** și **PLAN**:

CURSURI

Curs	Titular
Baze de date	Popescu
Rețele de calculatoare	Ionescu
Tehnici de programare	Anton

OBSERVAȚII?

PLAN

Specializare	Curs	An
Automatică	Rețele de calculatoare	4
Automatică	Baze de date	3
Calculatoare	Tehnici de programare	3
Calculatoare	Rețele de calculatoare	4
Calculatoare	Baze de date	4

SUGESTII??

Modelele prezentate - „modele logice”

- ❑ deși structurile utilizate sunt abstracte, ele reflectă o organizare particulară (arbore, graf, tabel, obiect)

Modele conceptuale

- ❑ au fost introduse pentru a descrie datele într-o manieră independentă de modelul logic
- ❑ nu sunt disponibile în SGBD-urile comerciale
- ❑ sunt utilizate în faza preliminară a proiectării bazelor de date, pentru o analiză cât mai bună a aplicației, fără implicații de implementare
- ❑ exemplu de model conceptual - **modelul Entitate - Relație**

Scheme și instanțe

schema bazei de date - caracteristici ale datelor care nu se modifică în timp; conține numele relației (tabelului), urmat de numele atributelor (coloanelor)

Exemplu

schema { **CURSURI (Curs, Titular)** – schema relației (tabelului) CURSURI
bază date { **PLAN (Specializare, Curs, An)** – schema relației (tabelului) PLAN

instanța (starea) bazei de date - valorile actuale din baza de date care se modifică în timp; conține o colecție de linii ale relației respective, care variază în timp

Exemplu

Instanța relației CURSURI este dată de următoarele trei perechi:

Baze de date	Popescu
Rețele de calculatoare	Ionescu
Tehnici de programare	Anton

Nivele de abstractizare în SGBD-uri

Arhitectura unui SGBD este împărțită pe trei nivele, **logic**, **intern** și **extern**; fiecare nivel are asociată o schemă

Schema logică - o descriere a întregii baze de date prin prisma modelului logic adoptat de SGBD (relațional, ierarhic, rețea sau obiect)

Schema internă - descrie implementarea schemei logice prin prisma structurilor de stocare fizică

Schema externă - descrierea unei porțiuni a bazei de date prin prisma modelului logic

- o schemă externă poate oferi o organizare diferită a datelor cu scopul de a reflecta un punct de vedere al unui anumit utilizator sau grup de utilizatori
- este posibil ca unei singure scheme logice să-i fie asociate diverse scheme externe
 - fiecare schemă externă va furniza o anumită vedere asupra bazei de date
- în majoritatea sistemelor, nivelul extern nu apare explicit, dar este posibilă definirea unor relații derivate, numite **vederi**

CURSURI

Curs	Titular
Baze de date	Popescu
Rețele de calculatoare	Ionescu
Tehnici de programare	Anton

PLAN

Specializare	Curs	An
Automatică	Rețele de calculatoare	4
Automatică	Baze de date	3
Calculatoare	Tehnici de programare	3
Calculatoare	Rețele de calculatoare	4
Calculatoare	Baze de date	4

Exemplu de

schemă

externă?

Exemplu

Un student de la Automatică este interesat de cursurile oferite în cadrul acestei specializări. Această informație este prezentă în relația **AUTOMATICĂ**, derivată din relația PLAN

AUTOMATICĂ

Specializare	Curs	An
Automatică	Rețele de calculatoare	4
Automatică	Baze de date	3

- schemelor externe li se pot asocia **mecanisme pentru autorizarea accesului**
 - un utilizator poate fi autorizat să manipuleze doar datele descrise prin prisma schemei externe asociată



Introducere - Limbaje și utilizatori

SGBD-urile oferă o gamă largă de limbaje pentru gestiunea datelor și implică un spectru larg de utilizatori

Ținând cont de diferența dintre structura și instanța unui tabel, putem face o diferențiere similară între limbajele bazei de date:

- **limbajul de definire al datelor (LDD)**
 - definirea schemelor logice, externe și fizice
 - definirea autorizărilor de acces

- **limbajul de manipulare al datelor (LMD)**
 - interogarea instanțelor unei baze de date
 - modificarea instanțelor unei baze de date

Remarcă: Unele limbaje (SQL) oferă facilitățile ambelor limbaje (LDD și LMD) într-o formă integrată

În funcție de modul de exploatare a bazei de date, utilizatorii se pot împărți în:

- **administratorul bazei de date (DBA)** – persoana responsabilă de proiectarea, controlul și administrarea bazei de date
 - mediază între cerințele diverse, adesea conflictuale, exprimate de utilizatori, asigurând controlul partajat asupra datelor
 - este responsabil pentru garantarea serviciilor
 - asigură corectitudinea sistemului
 - gestionează autorizările de acces la date
- **programatorii de aplicații** – definesc și creează programele ce accesează baza de date
 - utilizează LMD sau alte unelte pentru generarea interfețelor cu baza de date

- **utilizatorii obișnuiți** – cei care utilizează efectiv baza de date
 - **utilizatori finali** – utilizează programe ce realizează activități frecvente și predefinite
 - **utilizatori ocazionali** – capabili să utilizeze limbaje interactive pentru a accesa baza de date, formulând interogări (sau actualizări) de diferite tipuri

pot fi specialiști în limbajul pe care îl utilizează și interacționează frecvent cu baza de date; termenul „ocasional” se referă la faptul că interogările nu sunt predefinite

Avantaje

- Datele - resurse comune ale organizației, disponibile tuturor membrilor autorizați
- Baza de date furnizează un model standardizat și precis
 - Modelul poate fi folosit, cu extensiile necesare, în aplicații viitoare
- Control centralizat al datelor
- Partajarea bazelor de date \Rightarrow reducerea redundanței și inconsistenței datelor
- Independența datelor \rightarrow dezvoltarea de aplicații mai flexibile și mai ușor de modificat

Dezavantaje

- SGBD - scumpe, complexe și foarte diferite de multe alte unelte software
- Introducerea SGBD necesită un efort financiar
 - direct - costul produsului
 - indirect - resurse hardware și software, pregătirea personalului etc.
- SGBD - furnizează, în forma standard, un set de servicii care au un anumit cost
 - în cazurile în care câteva dintre aceste servicii nu sunt neapărat necesare, e dificilă extragerea serviciilor necesare la un moment dat → ineficiență

Situații în care folosirea SGBD-urilor nu este necesară:

- există un singur utilizator
- există mai mulți utilizatori care nu au nevoie de acces concurent la date

Dezvoltarea actuală a SGBD-urilor a dus la sisteme mai eficiente și mai fezabile, și la arhitecturi din ce în ce mai ieftine ⇒ a crescut posibilitatea dezvoltării aplicațiilor cu SGBD-uri



■ Exemplu...

Schema bazei de date pentru o bibliotecă