

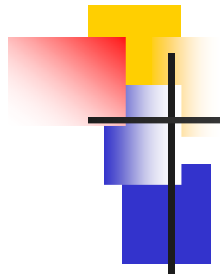


TEORIA SI IMPLEMENTAREA BAZELOR DE DATE

ș.l.dr.ing.ec. Mirela Danubianu

mdanub@eed.usv.ro

C208



Scopul cursului

- Recapitularea noțiunilor fundamentale de baze de date
- Baze de date avansate.
 - Baze de date distribuite
 - Baze de date spațiale
 - Baze de date temporale
 - Baze de date active



Bibliografie

- **Clasic si modern in teoria si practica bazelor de date relationale**, *Mirela DANUBIANU*, Ed. InfoData, Cluj-Napoca, 2009
- Baze de date. Proiectare. Gestiune. Implementare. *Thomas Connolly, Carolyn Begg, Anne Strachan ; Ed. Teora 2001*
- Baze de date relationale – M. Fotache
- Baze de date si gestiunea tranzactiilor – R. Dollinger, Ed. Albastră

Necesitatea studierii bazelor de date...

Trecerea de la stadiul efectuării de calcule către nevoia de informații (ex. sisteme suport pentru decizii);

- Creșterea volumului și a diversității datelor de prelucrat;
- Limitele posibilităților de analiză ale factorului uman;
- Evoluția tehnologiilor informatice;
- Extinderea domeniilor de utilizare a aplicațiilor de baze de date
- Extinderea cercetărilor fundamentate pe domeniul bazelor de date către depozite de date (Data Warehousing), analiza datelor (OLAP), explorarea datelor (Data Mining) și descoperirea cunoștințelor din bazele de date (KDD)





Scurt istoric al organizării și prelucrării datelor

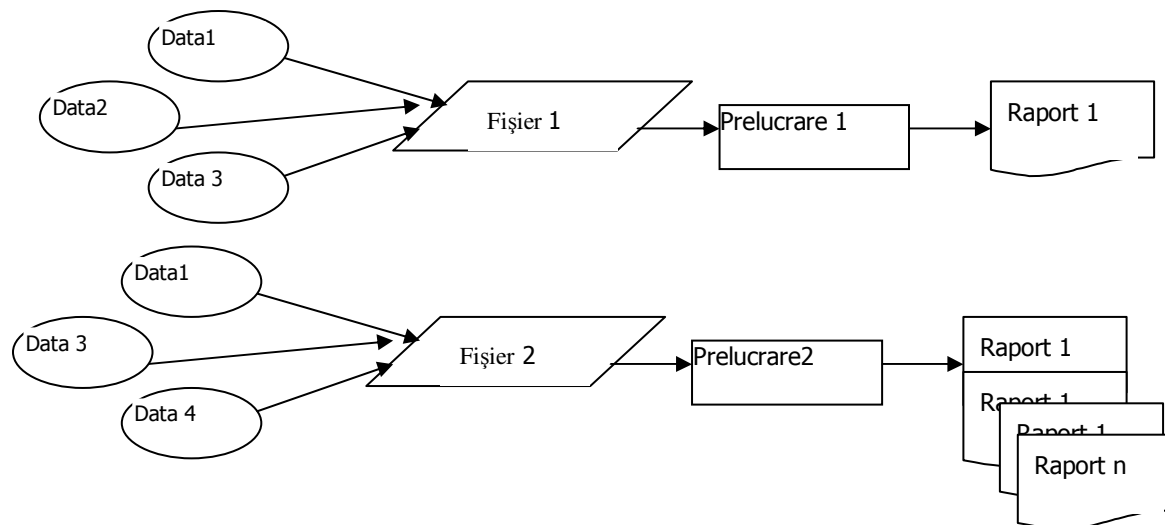
- Sisteme tradiționale bazate pe fișiere (1950-1960)
- SGBD bazate pe modelul de date ierarhic sau rețea (1970)
- SGBD relaționale
 - Apariția modelului relațional (1970)
 - Dezvoltarea SGBD relaționale (1970)
 - Apariția SGBDR comerciale (1980)
 - Maturizarea tehnologiei relationale pentru SGBD (1990)
- Sisteme de baze de date obiect-relaționale
 - Sisteme de baze de date deductive și sisteme de baze de date orientate obiect.
- Sisteme de baze de date orientate spre aplicații
 - Baze de date spațiale, temporale, multimedia, Web...
- Sisteme de depozitare a datelor (Data warehousing), sisteme de analiză a datelor și sisteme de explorare a datelor (data mining)

Evolutia activității de organizare și prelucrare a datelor

Sisteme tradiționale bazate pe fișiere

Colecție de programe aplicație, care efectuează servicii pentru utilizatorii finali, cum ar fi producerea de rapoarte. Fiecare program definește și gestionează propriile sale date.

- **Fișierul** = principalul tip de organizare a datelor.
- fiecare dată este descrisă independent în toate fișierele în care apare
- între fișiere nu există o relație definită explicit





Ce probleme apar la sistemele bazate pe fișiere?

- **Caracteristicile sistemelor de prelucrare bazate pe fișiere**
 - Răspunsul la o nouă problemă implică scrierea unei noi aplicații care creează fișierele de date corespunzătoare
 - Pentru o organizație fișierele de date au formate diferite iar aplicațiile pot fi scrise în limbaje diferite
- **Probleme**
 - Redundanță și inconsistență a datelor (cum pot fi actualizate datele din toate fișierele?)
 - Dificultăți în accesarea datelor (necesitatea unor noi programe de aplicație?)
 - Izolarea datelor (diferite formate, diferite fișiere)
 - Imposibilitatea controlului concurenței (utilizatori multipli)
 - Probleme de securitate
 - Probleme de integritate a datelor (satisfacerea constrângerilor?)

Evolutia activitatii de organizare si prelucrare a datelor

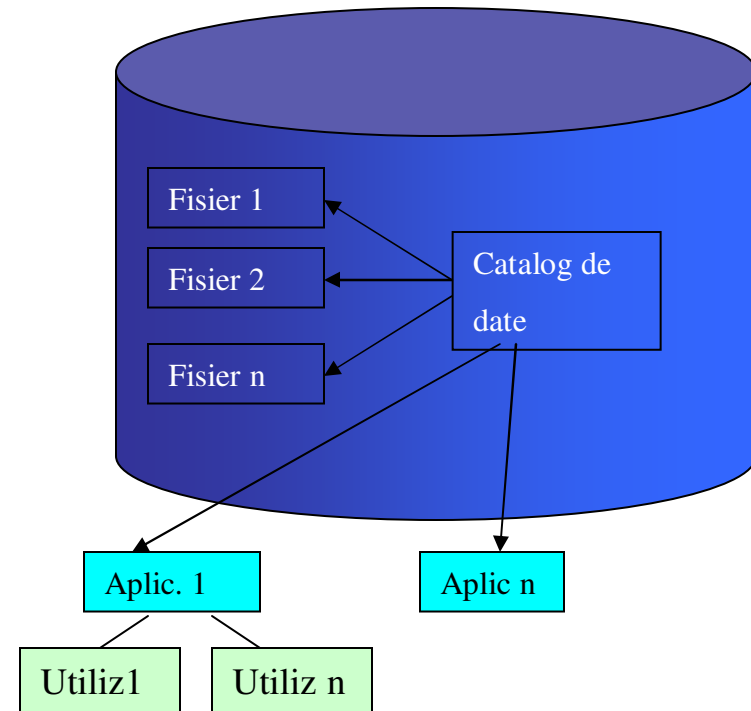
Tratarea prin baze de date

Cauze ale limitării tratării anterioare:

- Definitia datelor este încorporată în programele aplicatie
- Nu exista controlul accesului si manipularii datelor

■ Baza de date –

- ✓ Colectie partajata de date, între care exista relatii logice si o descriere a acestor date, proiectata pentru a satisface necesitatile informationale ale unei organizatii.
- ✓ Colectie autodescrisa de înregistrari integrate.
- Acumulări de date aflate în interdependență, cu o anumită organizare, în vederea obținerii operative a unor informații utile despre un anumit subiect.
- Ansamblu de date interconectate, împreună cu descrierea lor, care răspunde calităților de centralizare, coordonare, integrare și difuzie a informațiilor și care asigură satisfacerea tuturor necesităților de prelucrare ale utilizatorilor din sistem.



- ✓ **Concluzie:** este un depozit de date unic, de volum mare care este definit o singura data si este utilizat simultan de mai multe departamente și utilizatori.

M. Danubianu - Teoria și implementarea bazelor de date



Tratarea prin baze de date

- **Schema** bazei de date consta in descrierea generala a bazei de date.
 - Este specificată în procesul de proiectare și se modifică foarte rar
- **Instanta** bazei de date este data de setul de date operationale din baza la orice moment dat
 - se modifica frecvent
- **Natura autodescriptiva** a datelor duce la **independenta program-date**
- analiza necesitatilor informationale ale unei organizatii --> identificarea **entitatilor, a atributelor si a legaturilor** dintre ele

Tratarea prin baze de date

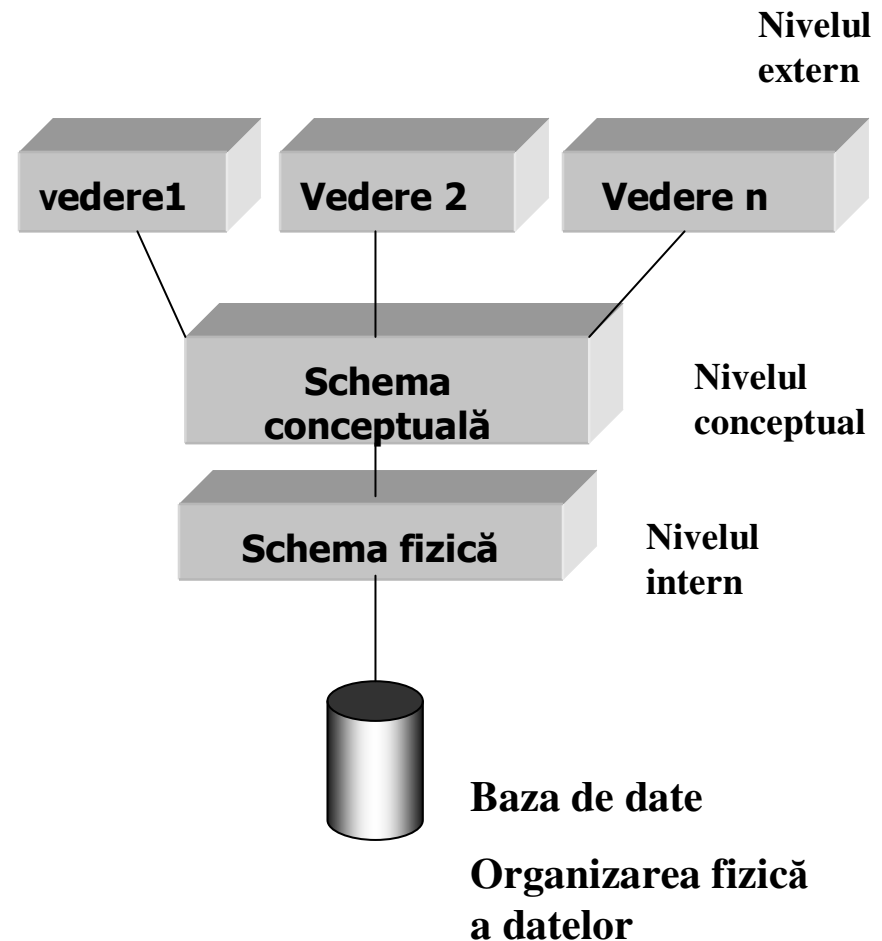
■ Exemplu

- baza de date Universitate
 - entitati : Studenti, Profesori, Cursuri, Sectii,...
 - attribute (proprietati): Nr. Matricol, Nume, Prenume, Data de nastere, adresa, etc.
 - legături: Studentii asistă la cursuri
 - Profesorii susțin cursuri
 - Cursurile se bazează pe cunoștințele altor cursuri, etc
- separarea definiției datelor de programele de aplicație presupune existența unei **definiții interne** și a unei **definiții externe** a datelor => **abstractizare a datelor**
 - **avantaj**: permite modificarea definiției interne a unui obiect fără a afecta utilizatorii acestuia dacă definiția externă rămâne aceeași
- **modelul extern** - descrierile care se referă strict la datele unei aplicații
- **avantaje**:
 - reducerea redundanței
 - evitarea inconsistenței
 - posibilitatea partajării datelor
 - încurajarea introducerii standardelor
 - posibilitatea aplicării restricțiilor de securitate
 - menținerea integrității datelor

Arhitectura bazelor de date

Arhitectura ANSI/SPARC cu trei niveluri

- 1971 grupul DBTG (**Data Base Task Group**) propune o terminologie standard si o arhitectură pe **doua niveluri**:
 - **schema** - abordare d.p.d.v. al sistemului si
 - **subschemă** - abordare d.p.d.v. al utilizatorului
- 1975 comitetul ANSI/SPaRC - abordare pe **trei niveluri**, cu un **catalog de sistem**.
- *Scop*: separarea fiecărei vederi a utilizatorilor bazei de date de modul în care aceasta este reprezentată fizic
- *Presupunere*: necesitatea unei zone independente de implementare care să izoleze programele de considerațiile de reprezentare de bază



Vedere 1

nrmatr	Nume	Prenume	Vârsta	adresa
--------	------	---------	--------	--------

Vedere 2

nrmatr	Nume	grupa
--------	------	-------

Nivel Conceptual



nrmatr	Nume	Prenume	DDN	adresa	grupa
--------	------	---------	-----	--------	-------

Nivel Intern

```
struct STUDENT {  
    int nrmatr;  
    int grupa;  
    char nume [15];  
    char prenume [15];  
    struct date DDN;  
    char adresa [20];  
    struct STUDENT *next; /* pointer către următoarea înregistrare */  
};  
index nrmatr; index grupa; /* definește indecșii
```



Independența datelor

- \exists unei delimitări nete între reprezentarea fizică a datelor și imaginea pe care o are utilizatorul despre aceste date
- Independența datelor este de două feluri:
 - fizică
 - logică
- **Independența fizică** dă măsura imunității aplicațiilor față de modificările în structura fizică de memorare a datelor
 - presupune că aplicațiile nu conțin nici o referire explicită la tipul fișierelor în care sunt memorate datele, la tipul dispozitivului de memorare sau la strategia de acces la date



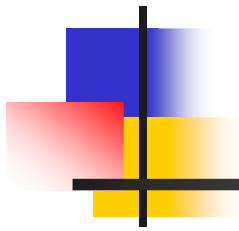
Independența datelor

- **Independența logică** a datelor se referă la imunitatea modelului propriu al fiecărui utilizator față de modificările în structura logică globală a bazei de date.
 - Se referă în special la problema adăugării de noi unități logice (câmpuri) la structura bazei de date și/sau la modificarea acestora și a relațiilor dintre ele.
 - Permite:
 - dezvoltarea bazei de date fără a afecta utilizatorii care nu au nevoie de noile date
 - reorganizarea bazei de date - regruparea câmpurilor în înregistrări, și definirea de noi câmpuri pe baza celor existente
 - problema delicată - eliminarea unei entități logice din baza de date - afectează utilizatorii care fac referire la entitatea eliminată
- d.p.d.v. al utilizatorului, problema independenței logice se manifestă legat de operațiile pe care sistemul îi permite să le efectueze asupra datelor din modelul propriu astfel încât să nu afecteze modelul altor utilizatori care folosesc parțial sau total aceleași date.



Avantajele utilizării bazelor de date

- Memorarea convenabilă și eficientă a datelor și posibilitatea generării de informații din volume mari de date
- Beneficii majore:
 - ◆ Eficiență și scalabilitate în ceea ce privește accesul la date și regăsirea acestora
 - ◆ Independența datelor
 - ◆ Redundanță controlată
 - ◆ Posibilitatea de a asigura integritatea și securitatea datelor
 - ◆ Facilități de acces concurent asupra datelor și posibilitatea recuperării acestora în cazul unui defect
 - ◆ Încurajarea introducerii standardelor
 - ◆ Administrarea uniformă a datelor
 - ◆ Analiza datelor (tehnici OLAP și Data Mining)



Sisteme de gestiune a bazelor de date



SISTEME DE GESTIUNE A BAZELOR DE DATE (SGBD)

Un SGBD reprezintă un sistem de programe care permit utilizatorului definirea, crearea, întreținerea bazei de date și accesul controlat la aceasta.

Oferă:

- **facilități de descriere a datelor** - prin intermediul limbajului de descriere a datelor(DDL)
 - specificarea tipurilor de date și a structurilor
 - specificarea constrângerilor asupra datelor
- **facilități de manipulare a datelor** - prin limbajul de manipulare a datelor (DML)
 - inserare de date
 - ștergerea de date
 - extragerea și interogarea datelor
- ∃ două tipuri de limbaje de manipulare a datelor:
 - *limbaje procedurale* - tratează bazele de date înregistrare cu înregistrare și specifică **cum** se va obține rezultatul dorit
 - *limbaje neprocedurale* - operează asupra unor seturi de înregistrări și descriu numai **ce date** vor fi obținute (SQL)



SISTEME DE GESTIUNE A BAZELOR DE DATE (SGBD)

■ ... mai oferă:

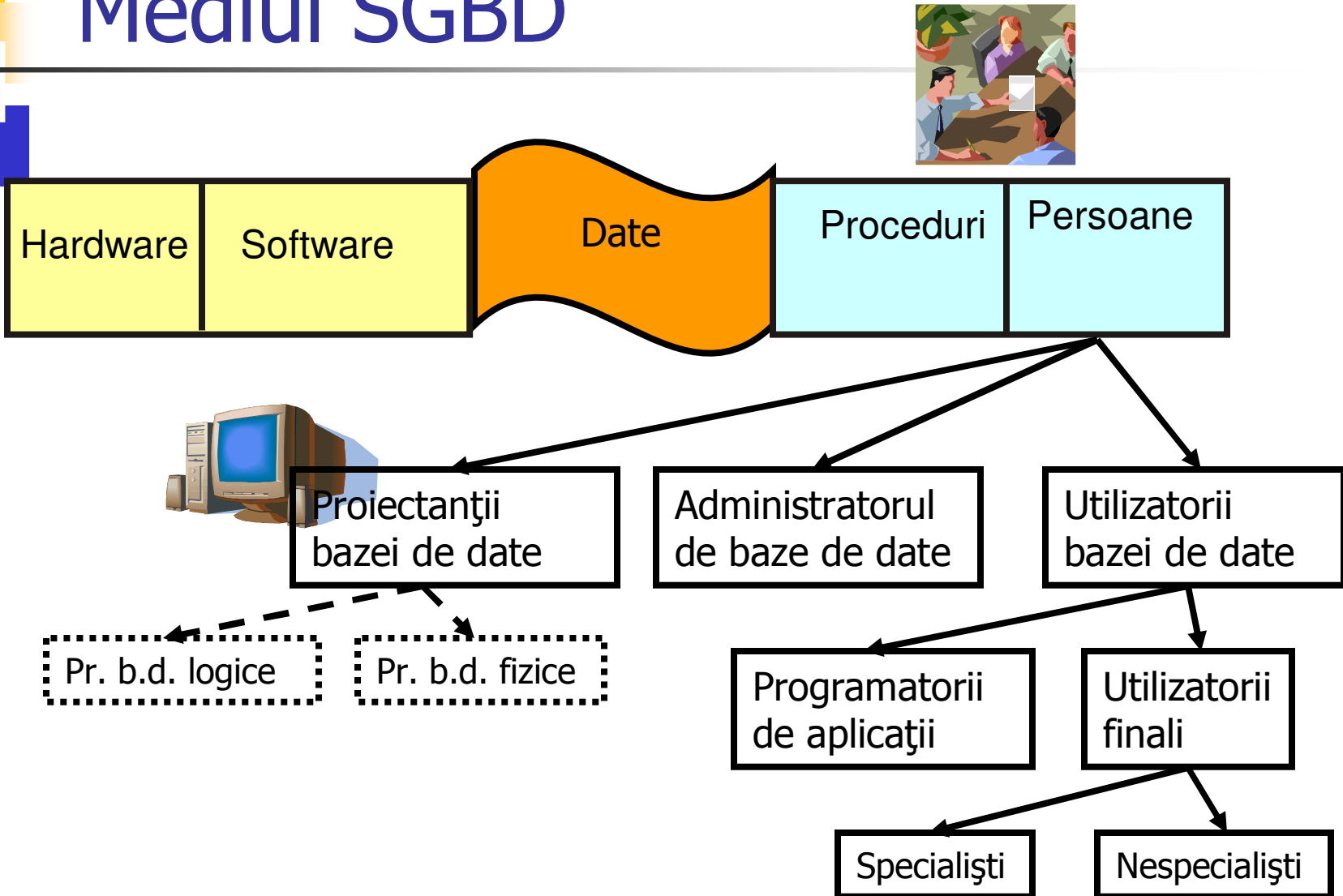
- **accesul controlat la baza de date** - presupune existența:
 - *unui sistem de securitate* - previne accesarea bazei de date de către utilizatori neautorizați
 - *unui sistem de integritate* - menține concordanța datelor stocate
 - *unui sistem de control al concurenței* - permite accesul partajat la baza de date
 - *unui sistem de control al refacerii* - restaurează baza de date într-o stare precedentă concordantă, în cazul unei defecțiuni hard sau soft
 - *unui catalog accesibil utilizatorilor* - care conține descrierea datelor din bază
- **un mecanism de vizualizare** - permite fiecărui utilizator să-și definească propriul mod de vizualizare a bazei de date;
- **o colecție de utilitare**: editoare de rapoarte, generatoare de aplicații, programe asistent, module de proiectare, posibilități de dezvoltare a unor aplicații de tip CASE, etc.



DDL și DML

- **Limbajul de definire a datelor (Data Definition Language - DDL)**
 - Specifică schema unei baze de date ca un set de definiții
 - Specifică structurile de memorare și metodele de acces
 - Instrucțiunile DDL sunt compilate, rezultând un set de tabele despre care sunt memorate date (metadate) într-un fișier special : dictionarul de date
- **Limbajul de manipulare a datelor (Data Manipulation Language - DML)**
 - **Regăsirea** informațiilor din bazele de date (**QL Query language**)
 - **Inserarea/ștergerea/actualizarea** informațiilor in bazele de date
- Un **limbaj de interogare** este o parte a DML care implică numai regăsirea de informații.
- **Limbajul de definire a vederilor (VDL View Definition Language)**

Mediul SGBD





SGBD

Funcțiile unui SGBD

1982 -Codd a enumerat 8 servicii care trebuie realizate de către un **SGBD complet**:

1. stocarea, regăsirea și reactualizarea datelor
2. un catalog accesibil utilizatorului care să conțină descrierile articolelor de date (caracteristică a arhitecturii ANSI/SPARC)
 - conține meta-date (date despre date)
3. asigurarea tranzacțiilor
 - **tranzacția** constă într-o serie de acțiuni realizate de un singur utilizator sau un program aplicație prin care se accesează sau se schimbă conținutul bazei de date.
 - SGBD furnizează un mecanism care garantează ca sunt efectuate toate reactualizările corespunzătoare unei anumite tranzacții sau că nu se efectuează nici una
4. servicii de control concurente
 - mecanism care garantează că baza de date este corect reactualizată atunci când mai mulți utilizatori efectuează simultan astfel de operații



SGBD

Funcțiile unui SGBD

- 5. servicii de reconstituire
 - mecanism de reconstituire a unei baze de date în cazul în care aceasta este deteriorată într-un fel oarecare
- 6. servicii de autorizare
 - se garantează accesul la date numai pentru utilizatorii autorizați => securitatea datelor
- 7. suport pentru comunicarea datelor
- 8. servicii de integritate
 - mijloace care asigură că atât datele din baza de date cât și modificările acestora respectă anumite reguli
- servicii suplimentare:
 - servicii pentru promovarea independenței de date
 - servicii utilitare

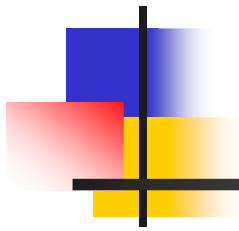
Avantajele si dezavantajele utilizarii SGBD-urilor

Avantaje

- controlul redundanței datelor
- asigurarea coerenței datelor
- mai multe informații obținute din aceeași cantitate de date
- posibilitatea partajării datelor
- integritate crescută a datelor
- securitate crescută
- concurență îmbunătățită
- posibilitatea aplicării standardelor
- economie de scală
- productivitate crescută
- servicii de salvare de siguranță și refacere

■ Dezavantaje:

- complexitate
- dimensiune
- costul sistemelor SGBD
- costuri adiționale pentru elemente hardware
- costul conversiei
- performanța
- impactul crescut al unei defecțiuni



Modele de date și modelare conceptuală



Abstractizarea

- constă în neglijarea aspectelor nerelevante și concentrarea asupra proprietăților care prezintă interes dintr-un anumit punct de vedere.
- este legată de criteriul de relevanță \Rightarrow prin procesul de abstractizare se rețin acele aspecte care sunt relevante pentru un anumit scop.
- se folosește pentru a obține categorii de date sau pentru a combina categorii de date în categorii mai generale.
- poate fi făcută pe mai multe nivele, tipurile de pe un nivel constituind obiecte pentru nivelul superior de abstractizare
 - \Rightarrow ierarhii de tipuri.
- \exists două tipuri de abstractizare:
 - generalizarea
 - agregarea.

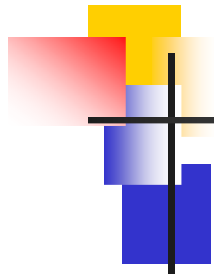


Modele de date și modelarea conceptuală

Modelul de date reprezintă o colecție integrată de concepte, necesare descrierii datelor, relațiilor dintre date și constrângerilor asupra datelor dintr-o organizație.

- ***Scop:***
 - *să fie reprezentată cât mai fidel situația din lumea reală;*
 - *datele să fie adaptate reprezentării și prelucrării pe calculator.*

- ***un model de date se compune din trei părți:***
 - *o parte structurală*, care constă dintr-un set de reguli conform căreia sunt constituite bazele de date ;
 - *o parte de manipulare*, care definește tipurile de operații care sunt permise asupra datelor (operații de reactualizare sau regăsire a datelor din bazele de date, operații de modificare a structurii bazei de date, etc.) ;
 - *un set de reguli de integritate* care garantează că datele sunt corecte.



Modele de date - clasificare

- *strict tipizate* : fiecare dată trebuie să aparțină unei categorii.
 - Dezavantaj: lipsa de flexibilitate, \Rightarrow dificultatea reprezentării deosebirilor semantice mai subtile.
 - Avantaj: proprietățile datelor pot fi abstractizate și cercetate în termenii categoriilor cărora le aparțin \Rightarrow bazat pe aceste categorii se poate formula o teorie care cuprinde proprietățile datelor.
 - stau la baza proiectării SGBD-urilor.
- *slab tipizate*: nu fac nici o presupunere legată de categorii.
 - categoriile sunt permise în măsura în care sunt utile, dar datele individuale pot exista prin ele însele și pot fi legate de alte date.



Tipuri de modele de date

- modele de date bazate pe obiecte
 - utilizează concepte cum ar fi : entitățile, attributele și relațiile.
 - modelul ER
- modele de date bazate pe înregistrări:
 - baza de date constă dintr-un număr de înregistrări cu format fix, posibil de tipuri diferite. Fiecare tip de înregistrare definește un număr fix de câmpuri, fiecare câmp având de obicei o lungime fixă.
 - modelul ierarhic
 - modelul rețea
 - modelul relațional
- modele de date fizice.



Modelul de date ierarhic

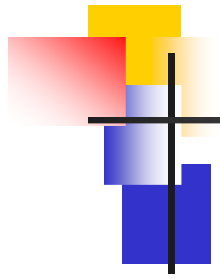
- Datele sunt reprezentate ca o colecție de înregistrări
- Legaturile sunt reprezentate ca multimi
- Se prezinta ca o structură arborescentă in care fiecare nod poate avea un singur nod parinte

Modelul de date rețea

- Datele sunt reprezentate ca o colecție de înregistrări
- Legaturile sunt reprezentate ca multimi
- Foloseste teoria grafurilor pentru reprezentarea datelor

Modele de date fizice

- Descriu modul de memorare al datelor pe baza structurilor înregistrărilor, modul de ordonare al înregistrărilor, căile de acces...



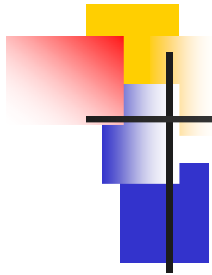
Model conceptual

- Un *model conceptual* sau *schemă*, cuprinde :
 - numele categoriilor
 - proprietățile acestora și
 - legăturile dintre categorii.
- Orice model conceptual este o **descriere a unei structuri de date**
- Regulile după care este construit un *model conceptual* sunt definite în cadrul unui *model de date*.



Entități, legături între entități

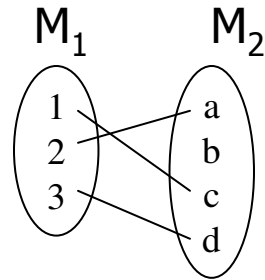
- Modelul conceptual cuprinde descrierea tuturor entităților unei baze de date împreună cu toate legăturile existente între ele.
- **entitate** este un conținut de sine stătător, o realitate obiectivă care există prin ea însăși.
 - este caracterizată prin proprietățile sale, reprezentate prin atribute.
- În cadrul diferitelor modele de date intervin două forme de structurare a datelor:
 - **prima formă de structurare a datelor** se referă la modalitatea de asociere a atributelor pentru a forma descrierile tipurilor de entități cărora le aparțin.
- Între mulțimile de entități dintr-o bază de date există diferite legături (relații), care trebuie descrise în modelul conceptual al bazei de date.



- ***A doua formă de structurare a datelor*** se referă la modul în care sunt reprezentate legăturile existente între diferitele mulțimi de entități ale bazei de date.
- Modelele de date diferă între ele prin modul în care se poate realiza reprezentarea legăturilor dintre mulțimi și entități.

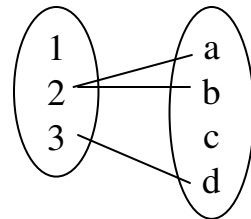
Tipurile de legături

■ 1:1



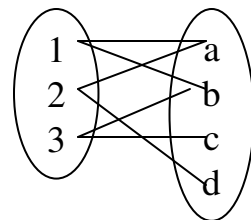
- unei entități din M_1 îi corespunde o singură entitate din M_2 și reciproc (rel. de tip soț-soție);

■ 1:N

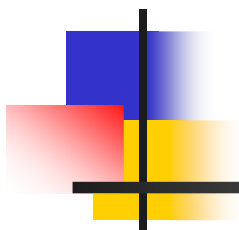


- unei entități din M_1 îi corespund una sau mai multe entități din M_2 , dar fiecărei entități din M_2 îi corespunde o singură entitate din M_1 (relație de tip tată-fiu)

■ M:N



relație M:N când unei entități din M_1 îi corespund una sau mai multe entități din M_2 și reciproc (relație de tip prieten-prieten)



Modelul relațional



De ce se studiază modelul relațional

Cel mai utilizat model de date

- IBM (DB2), Microsoft (ACCESS și SQLServer), Oracle (9i), Sybase, MySQL, etc.
- Sisteme superioare celor implementate pe baza modelelor mai vechi
 - IMS de la IBM (modelul ierarhic) și CODASYL (modelul rețea)
- Competitori actuali: modelul orientat pe obiecte și modelul obiect relațional
 - ObjectStore, Versant, Ontos
 - Informix Universal Server, UniSQL, O₂



Modelul de date relațional

1970 F.E.Codd - “ Un model relational de date pentru bănci de date partajate de dimensiuni mari”

- Obiective:
 - să permită un grad înalt de independență a datelor
 - să furnizeze baze solide pentru tratarea coerenței și a problemelor de redundanță a datelor --> este introdus conceptul de relații normalizate = relații care nu conțin grupuri ce se repetă
 - să permită expansiunea limbajelor de manipulare a datelor orientate spre seturi
- Datele se prezintă sub forma de relații (fie că se referă la entități, fie că se referă la legăturile dintre acestea)
- **Caracteristici :**
 - este orientat spre multimi
 - a permis introducerea unor limbaje neprocedurale de manipulare a datelor
 - nu este orientat spre sistemul de calcul.



Definiii

- I. Se dă o colecție de mulțimi D_1, D_2, \dots, D_n (nu neapărat distincte). **R este o relație** pe aceste mulțimi, dacă este o mulțime de n-tuple ordonate (d_1, d_2, \dots, d_n) astfel încât $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$
- II. Se definește produsul cartezian $D_1 \times D_2 \times \dots \times D_n$ al mulțimilor D_1, D_2, \dots, D_n ca fiind mulțimea tuturor n-tupelor ordonate (d_1, d_2, \dots, d_n) astfel încât $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$. O relație R pe mulțimile D_1, D_2, \dots, D_n este o submulțime a produsului cartezian $D_1 \times D_2 \times \dots \times D_n$.
 - $n = \text{gradul (aritatea) relației}$
 - numărul n-tupelor din relație la un moment dat = *cardinalitatea relației*
 - $D_1, D_2, \dots, D_n = \text{domeniile relației R}$

Orice relație este o *mulțime* ale cărei elemente sunt n-tuple. \Rightarrow următoarele proprietăți:

- într-o relație nu există două n-tuple identice
 - stă la baza definirii conceptului de cheie a unei relații
- ordinea n-tupelor într-o relație este arbitrară
- Baza de date relațională = *un set de relații normalizate.*

Domenii și attribute



Domeniul reprezintă ansamblul de valori admisibile pentru o componentă a unei relații

- domeniul numelor de orașe, domeniul numelor de persoane, domeniul numerelor întregi, etc

Două domenii sunt **compatibile** dacă ele sunt comparabile din punct de vedere semantic (dacă mulțimile de valori care le definesc nu sunt disjuncte)

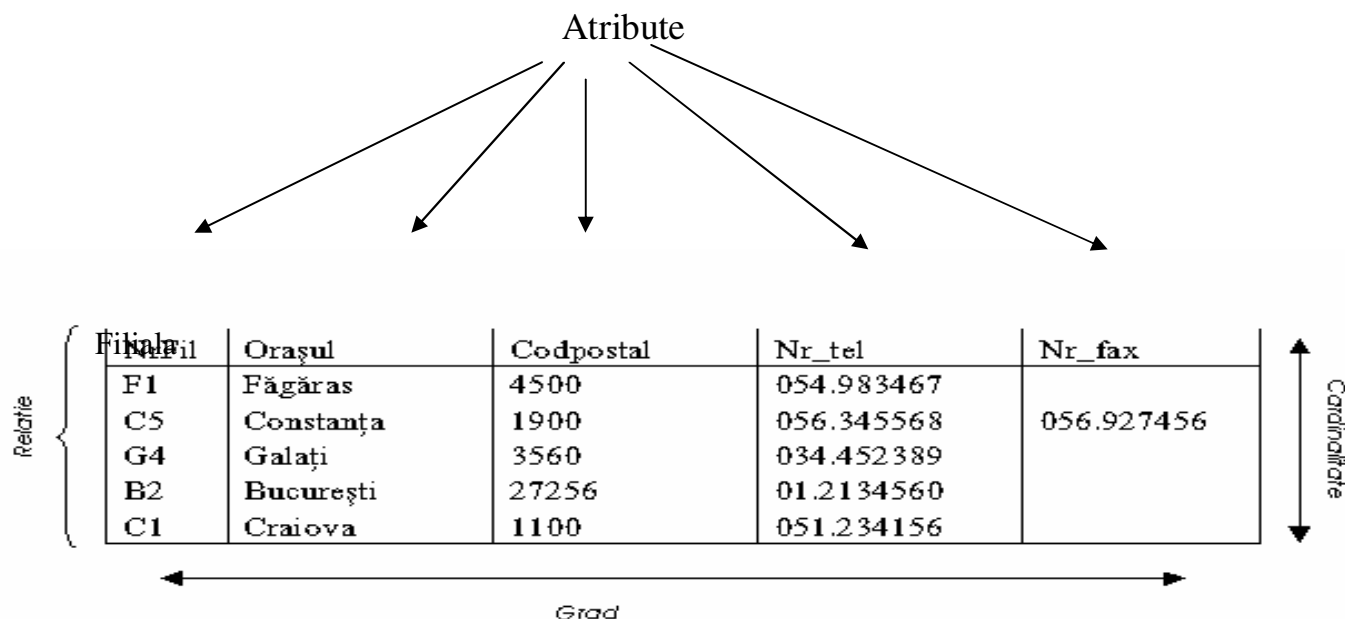
- noțiunea de **domeniu interpretat** restricționează problema compatibilității domeniilor chiar în cazul în care, conform definiției, două domenii ar fi compatibile

Atributul este o utilizare sub un nume oarecare a unui domeniu

- într-o relație pot exista mai multe attribute derivate din același domeniu
- în termenii abstractizărilor, un domeniu este o generalizare a unor attribute

Modul de prezentare al unei relații

Filiala (NrFil, Orasul, Codpostal, Nr_tel, Nr_fax)

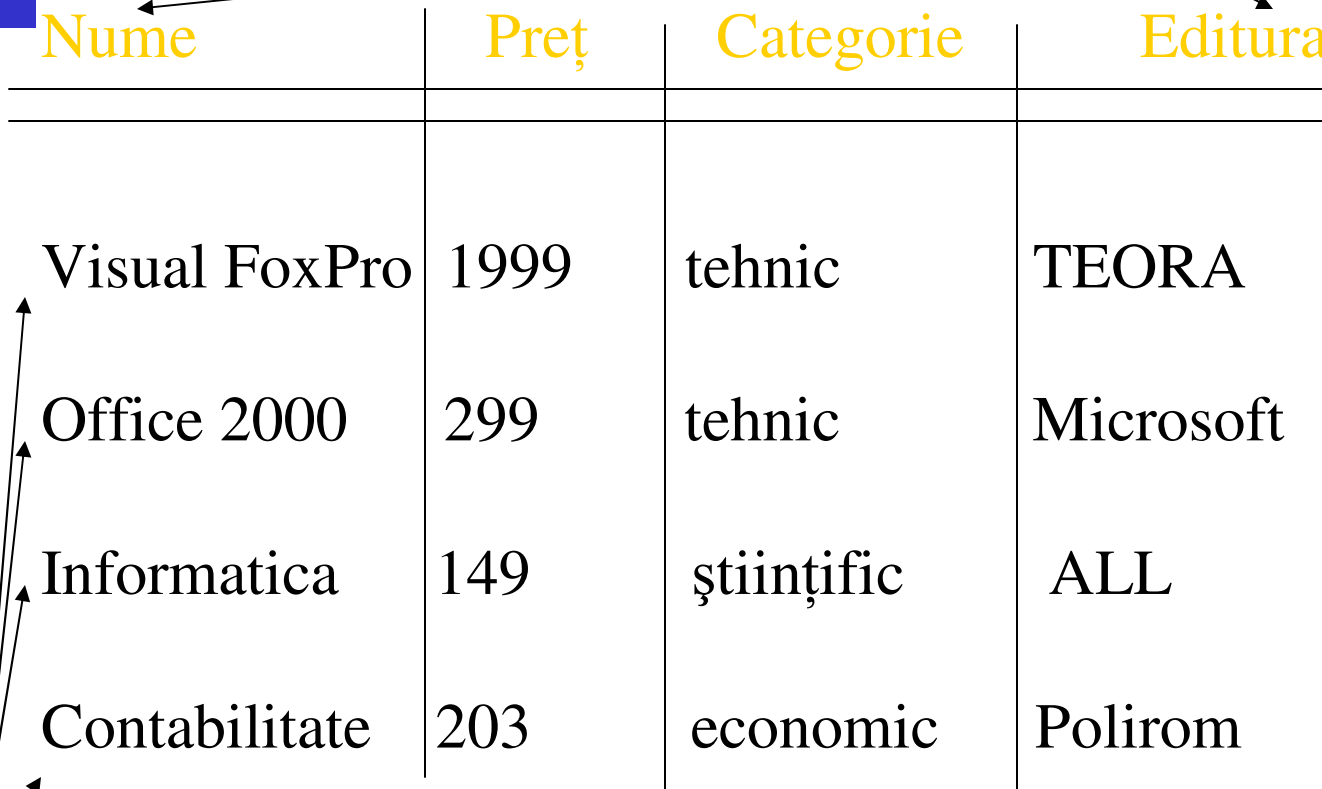


Este reprezentată ca un tabel.

Terminologie

Tabel sau relație Produse:

Atribute sau câmpuri



	Nume	Preț	Categorie	Editura
Visual FoxPro		1999	tehnic	TEORA
Office 2000		299	tehnic	Microsoft
Informatica		149	științific	ALL
Contabilitate		203	economic	Polirom

Tupluri sau rânduri sau înregistrări



Structuri de reprezentare

- \exists o singură formă de structurare a datelor – relațiile - se utilizează atât pentru reprezentarea tipurilor de entități cât și a legăturilor dintre acestea.
- Descrierea unei baze de date relaționale - printr-o **schemă relatională**, care constă din una sau mai multe **scheme de relație**
- **Schema de relație** este formată din denumirea unei relații urmată de un set de atribute:
 - **Facultate** (Codf, Nume, Adresa)
 - **Personal** (Codpers, Nume, Funcție, Salariu, Codf)
 - **Profesori** (Codp, Nume, Funcție, Disciplină, Codf)
 - **Student** (Matr, Nume, Incadrare, Sit_scolară, An, Sex)
 - **Nota** (Matr, Codp, Nota)
- reprezentarea legăturilor dintre tipurile de entități folosește două tehnici:
 - propagarea cheilor dintr-o schemă de relație în alta - pentru reprezentarea legăturilor de tip 1: 1 sau 1: N
 - crearea unei scheme de relație separate - de joncțiune - pentru reprezentarea oricărui tip de legătură, inclusiv a celor de tip N:M
- **instanța unei baze de date** = totalitatea datelor operaționale conținute în baza de date la orice moment de timp. (un tabel conținând valorile curente pentru atribute)



Exemplu de instanță a relației STUDENT

sid	nume	login	specializare	adresa
53666	Axinte	ax@cs	cs	S V
53688	Drăgan	dr@ee	ee	NT
53650	Croitoru	cr@mat	mat	BT

❖ Cardinalitate = 3, grad = 5 , toate rândurile sunt distincte

Chei relaționale

Supercheie - un atribut sau un set de attribute care identifică în mod unic un tuplu al unei relații.

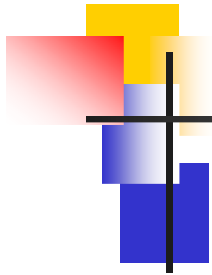
- Poate conține attribute adiționale care nu sunt strict necesare identificării unui tuplu
- **Cheie candidat (K)** a unei relații R - o supercheie pentru care nici o submulțime nu este supercheie în cadrul relației respective.

Proprietățile cheii candidat K: **unicitate** - valorile cheii identifica în mod unic orice tuplu al unei relații

■ **ireductibilitate**

Tipuri de chei candidat:

- simple
- compuse - attributele care o compun sunt **attribute prime** (celelalte attribute sunt neprime)
- **Cheie primară** este cheia candidat care este selectată pentru a identifica în mod unic tuplurile unei relații.
 - Restricții:
 - nu sunt admise valori nedefinite pentru nici un atribut al cheii primare
 - nici o valoare a unui atribut dintr-o cheie primara nu poate fi modificat printr-o operație de actualizare



Chei alternative sunt cheile candidat care nu sunt selectate drept chei primare.

Cheia străină este un atribut sau o multime de attribute din cadrul unei relații provenite prin propagarea unei chei candidat a unei alte relații.

- Valoarea unei chei străine = o referință la tuplul a cărei valoare este identică cu cea a cheii străine.
 - Relația care conține cheia străină - **relație de referință**
 - relația care conține cheia care s-a propagat ca și cheie străină - **relație referită**



Integritatea relațională

■ **Integritatea relațională** \Rightarrow set de reguli cu rolul de a garanta că datele sunt corecte.

1. Constrângeri de domeniu
2. Integritatea entităților
3. Integritatea referențială
4. Constrângeri de întreprindere

■ Constrângerile de integritate se bazează pe semantica realității descrise prin relațiile bazei de date.

■ Se poate verifica dacă o instanță a bazei de date violează o constrângere de integritate, dar nu se poate NICIODATĂ afirma că o constrângere este adevărată prin verificarea unei instanțe.

■ O constrângere de integritate este o stare referitoare la *toate instanțele posibile* ale unei baze de date!

■ Integritatea entităților (constrângere de cheie primară) și integritatea referențială (constrângere de chei străine) sunt cele mai generale constrângeri de integritate.



Constrângeri de domeniu și constrângeri de întreprindere

Constrângeri de domeniu- restrictii asupra mulțimii de valori permise pentru attributele relațiilor

- expresie a *principiului integrității domeniului* - posibilitatea SGBD de a verifica din punct de vedere *sintactic si semantic* orice valoare din baza de date sau orice operație efectuată asupra acesteia,
- folosesc definiția domeniului. => imposibilitatea înregistrării unor valori din afara domeniului corespunzător unui atribut și imposibilitatea efectuării unor operații între valori din domenii incompatibile.

Constrângeri de întreprindere - specificate de către utilizatorii sau administratorii unei baze de date



Integritatea entitatilor (constrângeri asupra cheii primare)

Se aplică cheilor primare ale *relațiilor de bază*.

Relație de baza = relație ce corespunde unei entități în schema conceptuală, și ale cărei tupluri sunt stocate fizic în baza de date

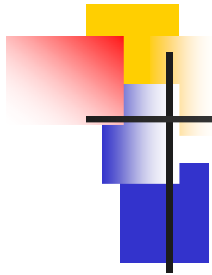
- **Într-o relație de bază nici un atribut al unei chei primare nu poate fi null.**
- **Null** - valoare a unui atribut care este în mod curent necunoscută sau care nu este aplicabilă unui anumit tuplu.
 - modalitate de tratare a datelor incomplete sau deosebite
 - nu trebuie confundat cu o valoare numerică egală cu zero sau cu un șir de caractere completat cu spații.
 - este tratată diferit față de alte valori=> pot să apară probleme de implementare, ca urmare a faptului că trebuie renunțat la logica booleană (specifică calculului predictiv de ordinul întâi) și trebuie lucrat cu o logică polivalentă (tri sau cvadrivalentă)




Integritatea referențială (constrângeri de cheie străină)

◆ Principiul integrității referinței - enunțat de Codd în 1979 sub următoarea formă:

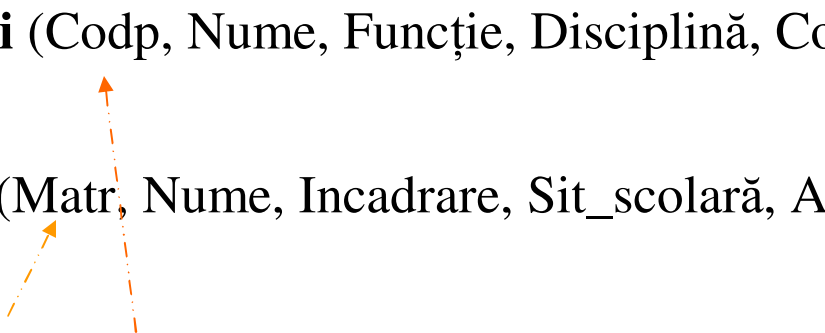
- ◆ *Dacă A este o cheie primară monoatribut în relația $R1$ și B este o componentă a unei chei primare multiatribut în relația $R2$, B fiind definită pe același domeniu cu atributul A , atunci mulțimea valorilor lui B în $R2$ trebuie să fie inclusă în mulțimea valorilor lui A în $R1$.*
- ◆ *Dacă B este o cheie străină în $R2$ rezultată prin propagarea unei chei primare A din $R1$, atunci orice valoare a lui B din $R2$ trebuie să se regăsească printre valorile lui A din $R1$.*
- ◆ *Dacă într-o relație există o cheie străină, valoarea acesteia trebuie ori să coincidă cu valoarea unei chei candidat a unui tuplu în relația sa de bază, ori să fie în întregime null.*



- ♦ => constrângeri referențiale -> pot fi reprezentate prin diagrame referențiale = grafuri ale căror noduri reprezintă relații, iar arcele reprezintă constrângeri referențiale. Sensul arcului este de la relația de referință către relația referită.
 - ♦ pot exista cicluri, numite cicluri referențiale și chiar cicluri de lungime unitară, caz în care relația referită este una și aceeași cu cea de referință. => relație autoreferită.



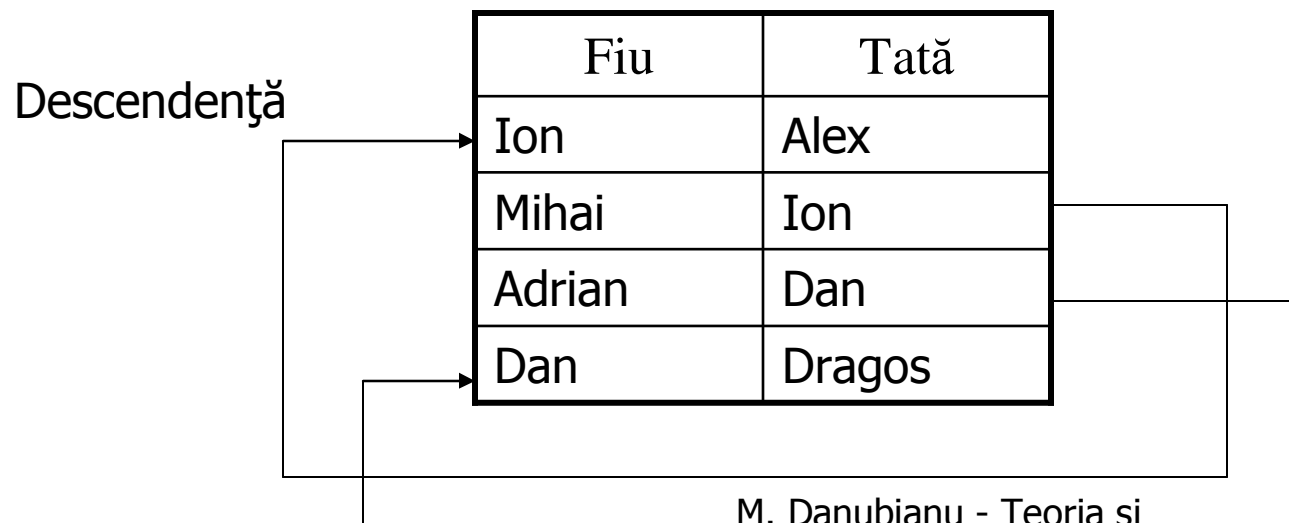
Exemple: Facultate (Codf, Nume, Adresa)

- **Profesori** (Codp, Nume, Funcție, Disciplină, Codf)
 - **Student** (Matr, Nume, Incadrare, Sit_scolară, An, Sex)
 - **Nota** (Matr, Codp, Nota)
- 

♦ FACULTATE ← PROFESORI ← NOTA → STUDENT

♦ Fie schema de relație : **Descendență (Tată, Fiu).**

- \forall fiu are un singur tată $\Rightarrow \forall$ tuplu al relației *Descendență* este univoc determinat de valoarea atributului *Fiu* (cheie a relației).
- \forall tată este , la rândul său, fiul cuiva $\Rightarrow \forall$ valoare a atributului *Tată* se regăsește printre valorile atributului *Fiu* \Rightarrow atributul *Tată* este cheie străină în relația *Descendență* și reprezintă o referință la aceeași relație.
- *relația Descendență este autoreferită*



Integritatea referențială - continuare

Poate fi afectată de operațiile de **adăugare, ștergere și actualizare**.

- ♦ **Relații de referință**

- ♦ **Operația de adăugare** - valorile tuturor atributelor care fac parte dintr-o cheie străină trebuie să se regăsească printre valorile atributelor pe care le referă în relațiile referite.
- ♦ **Operația de ștergere** - fără nici un fel de restricții d.p.d.v. al condițiilor de integritate referențială.
- ♦ **Operația de actualizare** = o operație de ștergere urmată de una de adăugare..

- ♦ **Relațiile referite**

- ♦ **Operația de adăugare** - fără restricții
- ♦ **Operația de ștergere** - posibil ca în relațiile de referință să existe tuple care fac referire la tuplul care se șterge. => una dintre următoarele două opțiuni: **ștergerea restricționată sau ștergerea cascadată**.
 - ♦ **Ștergerea restricționată**- nu se acceptă ștergerea unui tuplu din relația referită dacă acesta este referit de cel puțin un tuplu din relația de referință.
 - ♦ **Ștergerea cascadată** - ștergerea unui tuplu din relația referită va fi urmată de ștergerea tuturor tuplurilor din relațiile de referință care fac referire la tuplul șters. Dacă tuplurile șterse din relația de referință sunt, la rândul lor referite de alte tuple, atunci ștergerea se propagă, în cascadă, asupra tuplurilor care fac referire la cele din urmă ș.a.m.d.
- ♦ **Operația de actualizare** Actualizarea unei chei dintr-o relație referită se poate face în două moduri:
 - ♦ **actualizare restricționată**
 - ♦ **actualizare cascadată** -
- ♦ problema integrității - abordată în două variante: **declarativ sau procedural**.

M. Danubianu - Teoria și
implementarea bazelor de date



Proprietățile, avantajele și limitele modelului relațional

Proprietăți:

- nu exista tupluri identice
- ordinea liniilor si a coloanelor este arbitrară
- articolele unui domeniu sunt omogene
- fiecare atribut definește un domeniu și nu se poate repeta în cadrul aceleiași relații
- toate valorile unui domeniu sunt considerate atomice

Avantaje

- independența datelor
- redundanță minimă
- suplețe în comunicarea cu utilizatorul

Limite:

necesita volum mare de spațiu

- nu lucrează cu obiecte complexe
- nu realizează gestiunea datelor distribuite
- nu realizează gestiunea cunoștințelor
- pot să apară fenomene de inconsistență



Concluzii

- Reprezentare tabelară a datelor
 - Simplu și intuitiv
 - Cel mai utilizat pentru implementarea SGBD-urilor comerciale
 - Constrângerile de integritate pot fi specificate de DBA pe baza semnificațiilor aplicațiilor. SGBD-ul verifică violarea acestor constrângeri
 - A permis dezvoltarea limbajelor de interogare neprocedurale