

Proiectarea Bazelor de Date

Curs

Specializarea Ingineria Informației - anul IV-A

**Facultatea de Electronica, Telecomunicatii si
Tehnologia Informatiei**

Prof. Felicia Ionescu

Conținutul cursului

- Introducere – definiții, clasificări,
- Modelarea datelor
- Baze de date relaționale. Limbajul SQL
- Interogarea bazelor de date relaționale
- Dezvoltarea sistemelor de baze de date relaționale
 - Dezvoltarea bazelor de date
 - Dezvoltarea aplicațiilor de baze de date
- Gestiunea tranzacțiilor
- Baze de date obiect relaționale

Bibliografie

- Felicia Ionescu, “*Baze de date relationale si aplicatii*”, Editura Tehnica, București, 2004
- C. J. Date, “*An Introduction to Database Systems*”, 8th Edition, 2004
- R. Elmasri and S. B. Navathe, “*Fundamentals of Database Systems*”, Fourth Edition, 2004
- J. Ullman, J. Windom, “*A First Course In Database Systems*”, Prentice Hall, 1997
- Kevin Kline, Daniel Kline, “ *SQL In A Nutshell*”, O'Reilly, 2001
- Oracle 11g Documentation - <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>
- MySQL Documentation - <http://dev.mysql.com/doc/>
- PostgreSQL Documentation - <http://www.postgresql.org/docs/manuals/>
- Microsoft SQL Server Books Online – MSDN - Academic Alliance

Capitolul 1: Introducere

- Definiții – baze de date, sisteme de baze de date
- Componentele sistemelor de baze de date
- Arhitectura internă a sistemelor de baze de date
- Avantajele oferite de sistemele de baze de date
- Clasificări ale sistemelor de baze de date
 - Clasificare după modelul de date
 - Clasificare după numărul de utilizatori
 - Clasificare după numărul de stații pe care este memorată baza de date
- Modelarea datelor
 - Modele conceptuale de nivel înalt
 - Modele specifice de date
- Evoluția sistemelor de baze de date

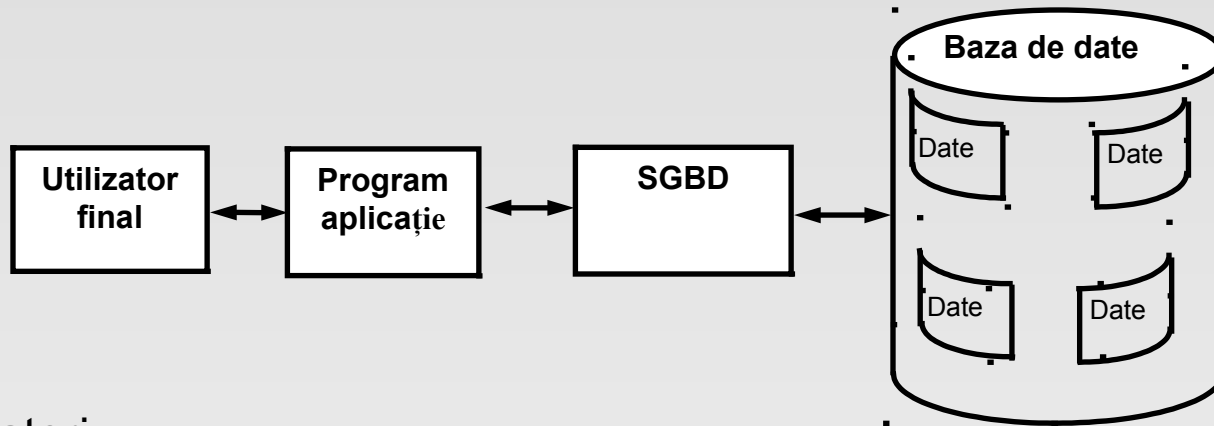
Sisteme de baze de date

- Bazele de date se folosesc in aproape toate domeniile de activitate actuale:
 - Activitati bancare si comerciale (depozite bancare, vanzari produse etc.)
 - Productie (gestiunea stocurilor, gestiunea financiar-contabila, salarizare etc.)
 - Evidenta populatiei, taxe si impozite
 - Servicii (servicii medicale, rezervari bilete de calatorie etc.)
- Definitie (in sens larg): **O baza de date** (*database*) este o colecție de date corelate din punct de vedere logic, care reflecta (modelează) un anumit aspect al lumii reale și este destinata unui anumit grup de utilizatori. In acest sens pot fi considerate ca fiind “baze de date”:
 - Fise de evidenta (mentinute manual)
 - Fisiere de documente sau foi de calcul tabelar (Microsoft Word, Microsoft Excel)
 - Baze de date mentinute computerizat
- Definitie (în sens restrans, actual): **O bază de date** este o colecție de date creată și menținută computerizat, care permite operații de:
 - Introducere (insert)
 - Stergere (delete)
 - Actualizare (update)
 - Interogare (query)

Componentele unui Sistem de Baze de Date (1)

- **Un sistem de baze de date** (*Database System*) este un sistem computerizat de menținere a evidenței unei anumite activități, folosind baze de date
- Componentele unui sistem de baze de date sunt: hardware, software, utilizatori și date persistente
- Hardware:
 - Sistemele de baze de date sunt instalate pe calculatoare de uz general
 - Bazele de date sunt memorate fizic ca fișiere pe discuri magnetice (hard-discuri)
 - Dimensiunea și performanțele bazei de date depind de sistemul de calcul pe care este instalată
- Software:
 - **Sisteme de operare**, biblioteci, instrumente de dezvoltare, interfețe
 - **Sistemul de Gestiune a Bazelor de Date** (SGBD) (*Database Management System – DBMS*) - recepționează cererile utilizatorilor de acces la baza de date, le interpretează, execută operațiile corespunzătoare și returnează rezultatul
 - **Aplicații de baze de date**: (*Database Applications*) – sunt programe care creează și utilizează baze de date

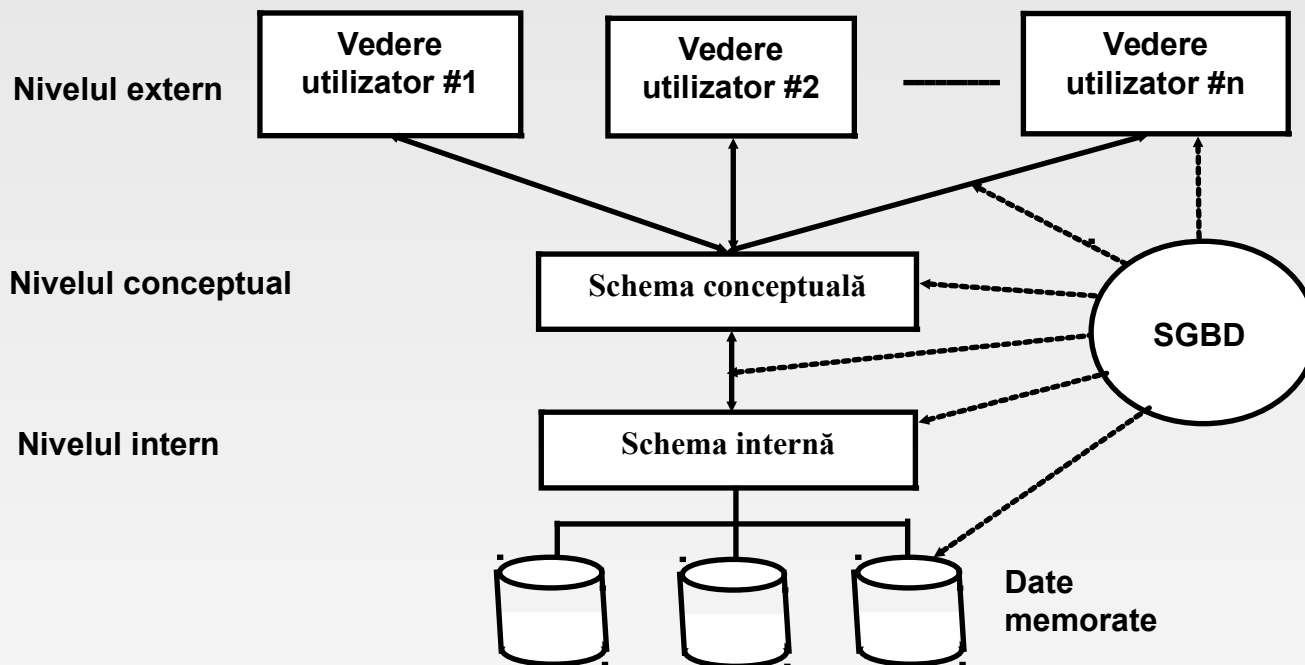
Componentele unui Sistem de Baze de Date (2)



- Utilizatori:
 - Administratorul bazei de date
 - Analisti si proiectanti ai bazelor de date
 - Programatori de aplicații
 - Utilizatori finali
- Datele persistente – sunt memorate in fisiere pe hard-disk
- Limbaje conceptuale pentru lucrul cu bazele de date:
 - Limbaje pentru Definirea Datelor(LDD) (*Data Definition Languages – DDL*)
 - Limbaje pentru Manipularea Datelor (LMD) (*Data Manipulation Languages – DML*)

Arhitectura interna a unui Sistem de BD

- Arhitectura pe 3 niveluri relativ independente: nivelul intern, nivelul conceptual și nivelul extern (Standard ANSI/X3/SPARC -1975)
- Schema → descrierea datelor pe un anumit nivel: schema internă, conceptuală (logica) și scheme externe (vedere utilizator)
- Corespondente între niveluri (*mappings*)



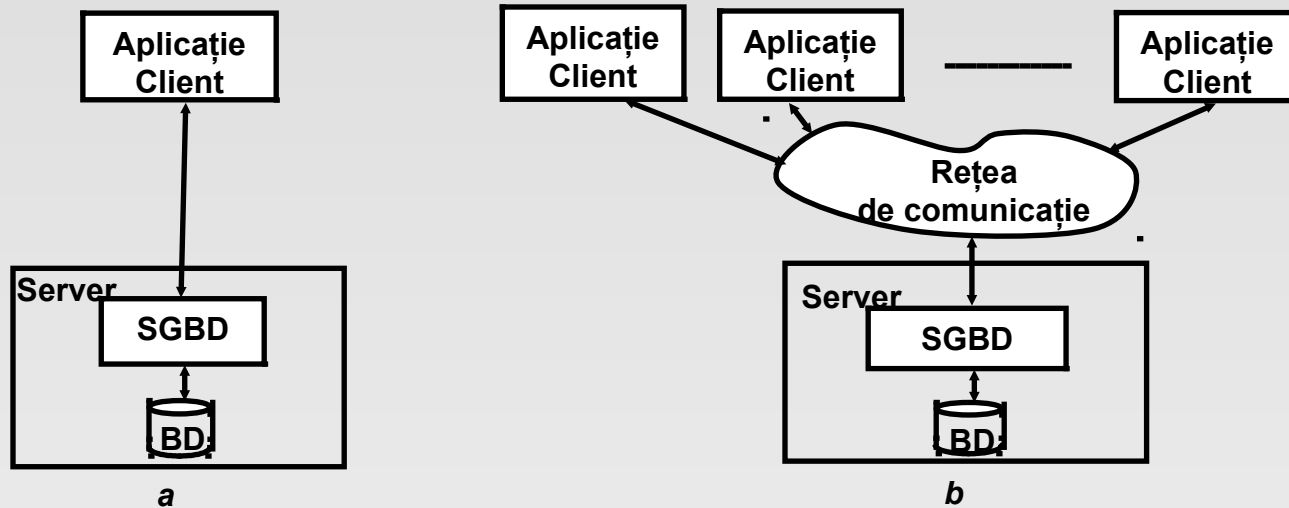
Avantaje oferite de Sistemele de BD

- Avantaje evidente:
 - Compactitate ridicată a datelor
 - Viteza mare de inserare, actualizare si regasire a datelor
 - Timp de dezvoltare și întreținere a bazelor de date redus
- Alte avantaje:
 - Reprezentarea unor asocieri complexe intre date
 - Redundanta controlata (si cat mai scazuta) a datelor
 - Flexibilitate, mentinerea datelor actualizate la zi
 - Independenta datelor fata de suportul hardware utilizat
 - Securitatea datelor: autentificarea utilizatorilor si autorizarea accesului
 - Impunerea de restrictii (constrangeri) de integritate la introducerea si actualizarea datelor
 - Mentinerea integritatii datelor in caz de defecte: salvare si refacere
 - Posibilitatea de partajare a datelor intre mai multe categorii de utilizatori, acces concurent la date
 - Posibilitatea de introducere a standardelor

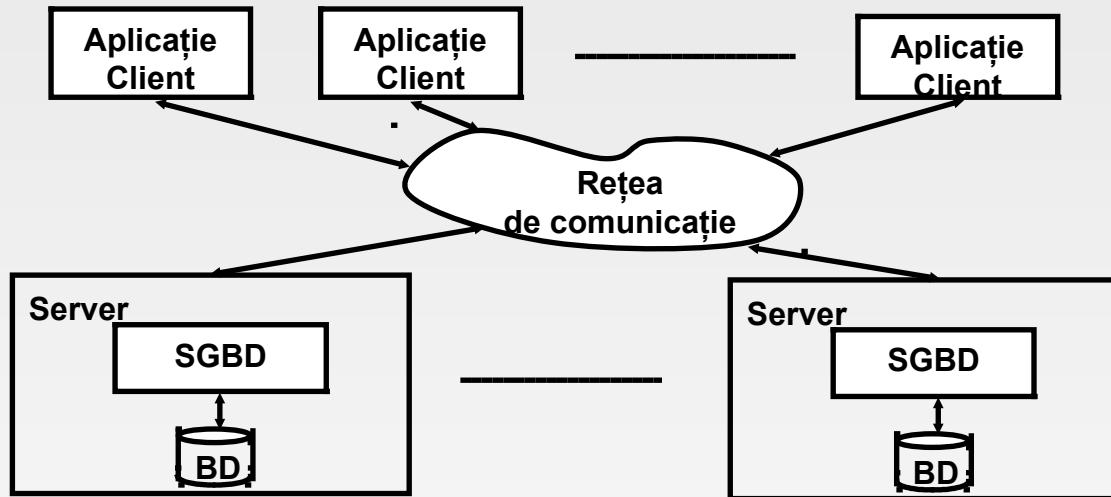
Clasificarea Sistemelor de Baze de Date (1)

- Arhitectura client-server:
 - Server (back-end): SGBD-ul si baza de date
 - Client (front-end): program (programe) de aplicatie
- Clasificare dupa modelul de date:
 - Modelul ierarhic de date
 - Modelul de date retea
 - **Modelul relational**
 - Modelul obiect-orientat
 - Modelul obiect-relational
- Clasificare dupa numarul de utilizatori
 - Sisteme mono-utilizator
 - Sisteme multi-utilizator
- Clasificare dupa numarul de statii pe care este memorata baza de date:
 - Baze de date centralizate
 - Baze de date distribuite

Clasificarea Sistemelor de Baze de Date (2)



Sisteme de baze de date centralizate: *a*- mono-utilizator; *b*- multi-utilizator



Sistem de baze de date distribuit

Modelarea datelor

- **Un model** este o abstractizare a unui sistem:
 - captează cele mai importante trăsături caracteristice ale sistemului (concepte)
 - conceptele trebuie sa fie relevante din punct de vedere al scopului pentru care se definește modelul respectiv
- Tehnica de identificare a trăsăturilor caracteristice esențiale ale unui sistem se numește **abstractizare**
- **Un model de date** stabilește regulile de organizare și interpretare a unei colecții de date și se reprezintă prin:
 - mulțimi de entități
 - asocieri între mulțimi de entități
- În proiectarea bazelor de date se folosesc 2 categorii de modele:
 - **Modele conceptuale de nivel înalt** (*modelul Entitate-Asociere, modelul Entitate-Asociere Extins*) – descriu concis mulțimile de entități și a asocierile dintre acestea, fără să specifice modul de structurare sau de prelucrare a datelor; se reprezintă printr-o - **schemă conceptuală de nivel înalt**
 - **Modele de structurare** a datelor (modelul ierarhic, modelul rețea, modelul relațional, etc.) - descriu reprezentarea mulțimilor de entități și a asocierilor dintre acestea prin anumite **structuri de date** (arbori, grafuri, tabele) - se reprezintă printr-o **schemă conceptuală (logică)**

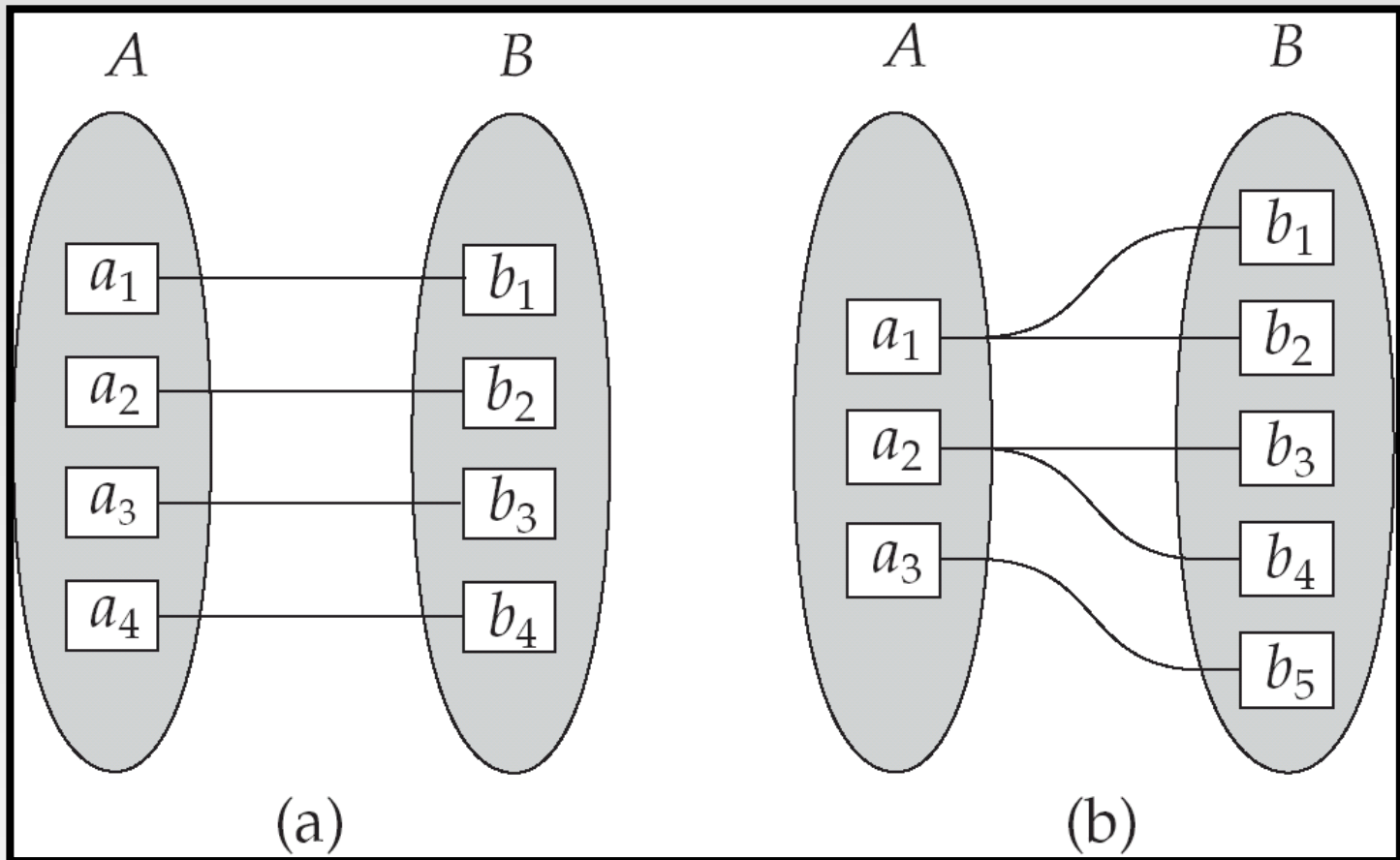
Modelul Entitate-Asociere

- **Modelul Entitate-Asociere** (*Entity-Relationship Model*) definește multimile de entități și asocierile dintre ele, dar nu impune nici un mod specific de structurare și prelucrare (gestiune) a datelor; Introdus în 1976 de P.S. Chen
- **O entitate** (*entity*) este „*orice exista în realitatea obiectivă și poate fi identificat în mod distinctiv*”
 - Exemple: o persoană, o plantă, o activitate, un concept etc.
- **Un atribut** (*attribute*) este o proprietate care descrie un anumit aspect al unei entități
 - Exemple: persoanele au nume, prenume, adresa etc.
- **Tip de entitate** (*entity type*): se referă la entitățile similare, care pot fi descrise prin aceleași atribute
 - Exemple: tipul persoană, tipul plantă
- **Multime de entități** (*entities set*): colecția tuturor entităților de același tip dintr-o bază de date constituie o mulțime de entități
 - Exemple: multimea tuturor persoanelor, multimea tuturor plantelor
- O entitate este o instanță a unui tip de entitate și un element al multimii de entități de acel tip
- În exprimarea curentă, adeseori nu se face diferențierea dintre entitate, tip de entitate și multime de entități, dar diferența este evidentă
- Asemănare cu modelul obiect: tip de entitate - clasă; entitate - obiect

Asocieri

- **O asociere** (*relationship*) este o legătură (corespondență) între entități din două sau mai multe mulțimi de entități; asocierile pot avea atribute
- **Tipul asocierii** (*relationship type*) – se refera la asocierile similare, care pot fi definite între entități din două sau mai multe mulțimi de entități
- **Multime de asocieri** (*relationship set*): multimea asocierilor de același tip
- O asociere este o instanță a unui tip de asociere și un element al mulțimii de asocieri de acel tip
- În exprimarea curentă, adeseori nu se face diferențierea dintre asociere, tip de asociere și multime de asocieri, dar diferența este evidentă
- **Gradul** unui (tip de) asociere (*degree*): numărul de mulțimi de entități asociate; după grad, asocierile pot fi:
 - **binare** (de gradul 2, între 2 mulțimi de entități) – majoritatea asocierilor
 - **multiple** (între k mulțimi de entități, $k > 2$)
- Categoriile (tipuri) de asocieri binare - după numărul elementelor din fiecare dintre cele două mulțimi puse în corespondență:
 - “unul-la-unul” (one-to-one) – 1:1; exemplu: sot-sotie
 - “unul-la-multe” (one-to-many) – 1:N; exemplu: parinte-fii
 - “multe-la-unul” (many-to-one) – N:1; exemplu: fii-parinte
 - “multe-la-multe” (many-to-many) – M:N; exemplu: profesori-studenti

Categorii de asocieri binare (1)

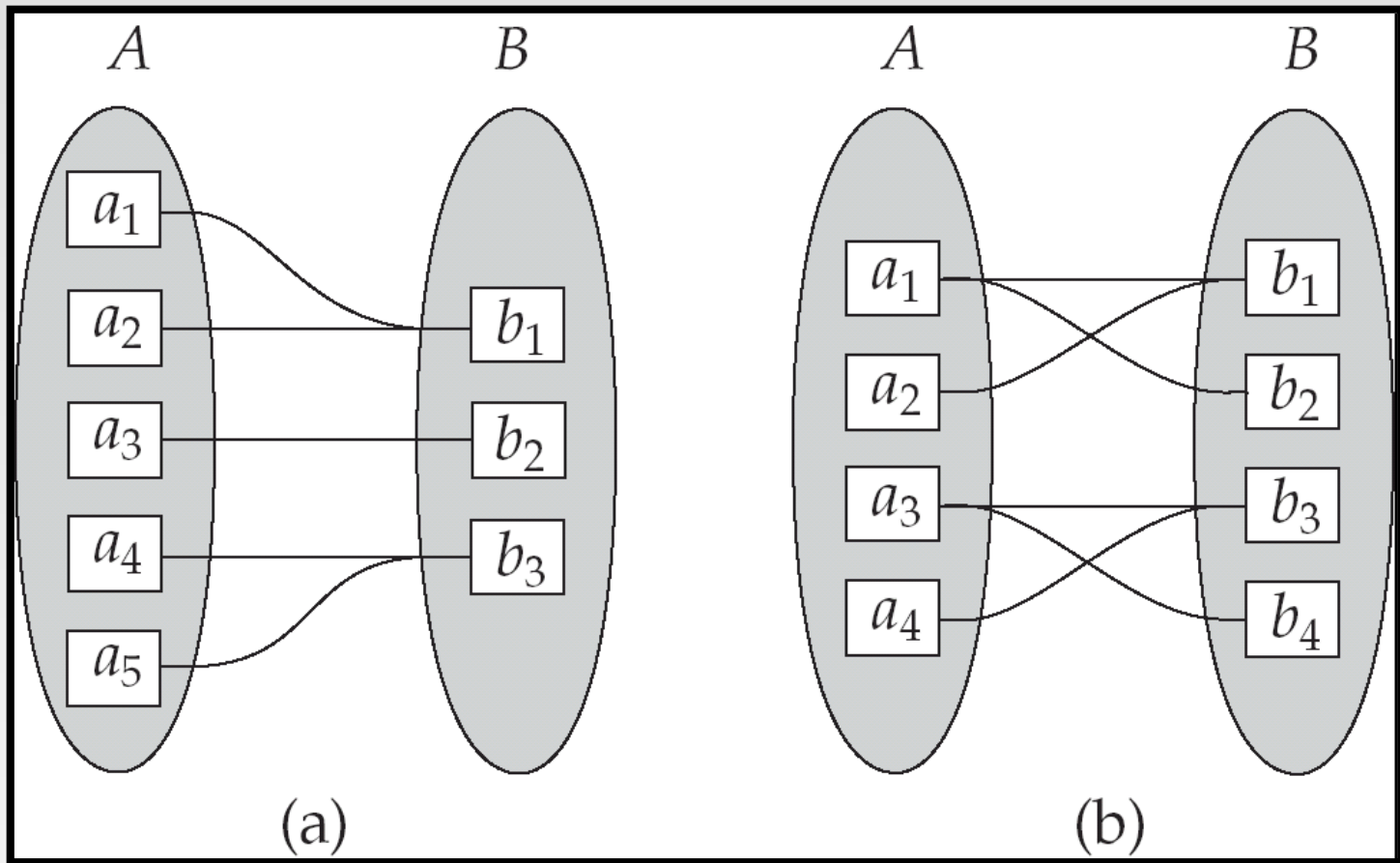


“unul-la-unul” – 1:1

“unul-la-multe”- 1:N

Asocieri binare intre multimile de entitati A si B

Categorii de asocieri binare (2)



"multe-la-unul"- N:1

"multe-la-multe"- M:N

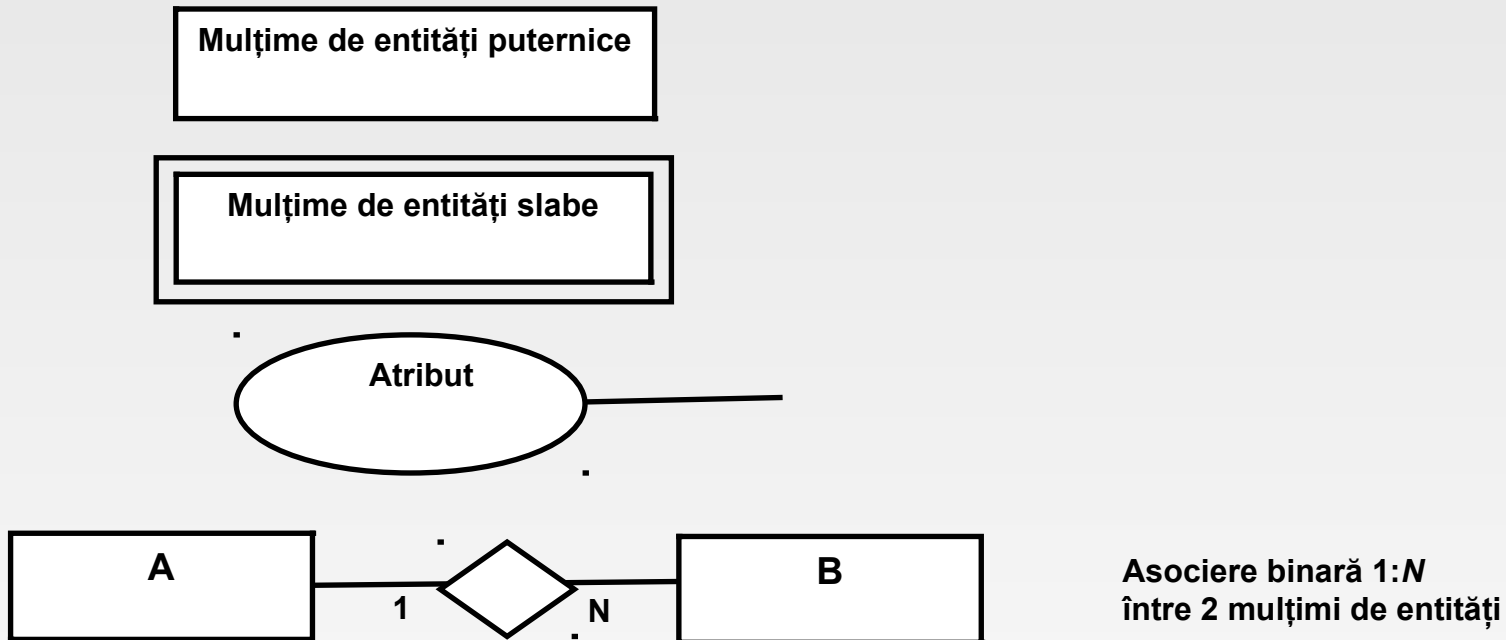
Asocieri binare intre multimile de entitati A si B

Cardinalitatea asocierilor

- **Cardinalitatea** (multiplicitatea) unei asocieri față de o mulțime de entități (*cardinality, multiplicity*) este *numărul maxim de elemente din acea mulțime care pot fi asociate cu un element din altă mulțime a asocierii*
 - Exemplu: asocierea “unul-la-multe” dintre mulțimile A și B prezintă multiplicitatea 1 față de mulțimea A și multiplicitatea N (se înțelege o valoare oarecare $N > 1$) față de mulțimea B
- **Raport de cardinalitate** (*cardinality ratio*): raportul dintre valorile cardinalităților unei asocieri față de două din mulțimile de entități asociate
 - Exemple pentru asocieri binare: 1:1, 1:N, N:1, M:N
 - Asocierile multiple (k-are, $k > 2$) prezintă câte un raport de cardinalitate pentru fiecare pereche de mulțimi de entități pe care le asociază.

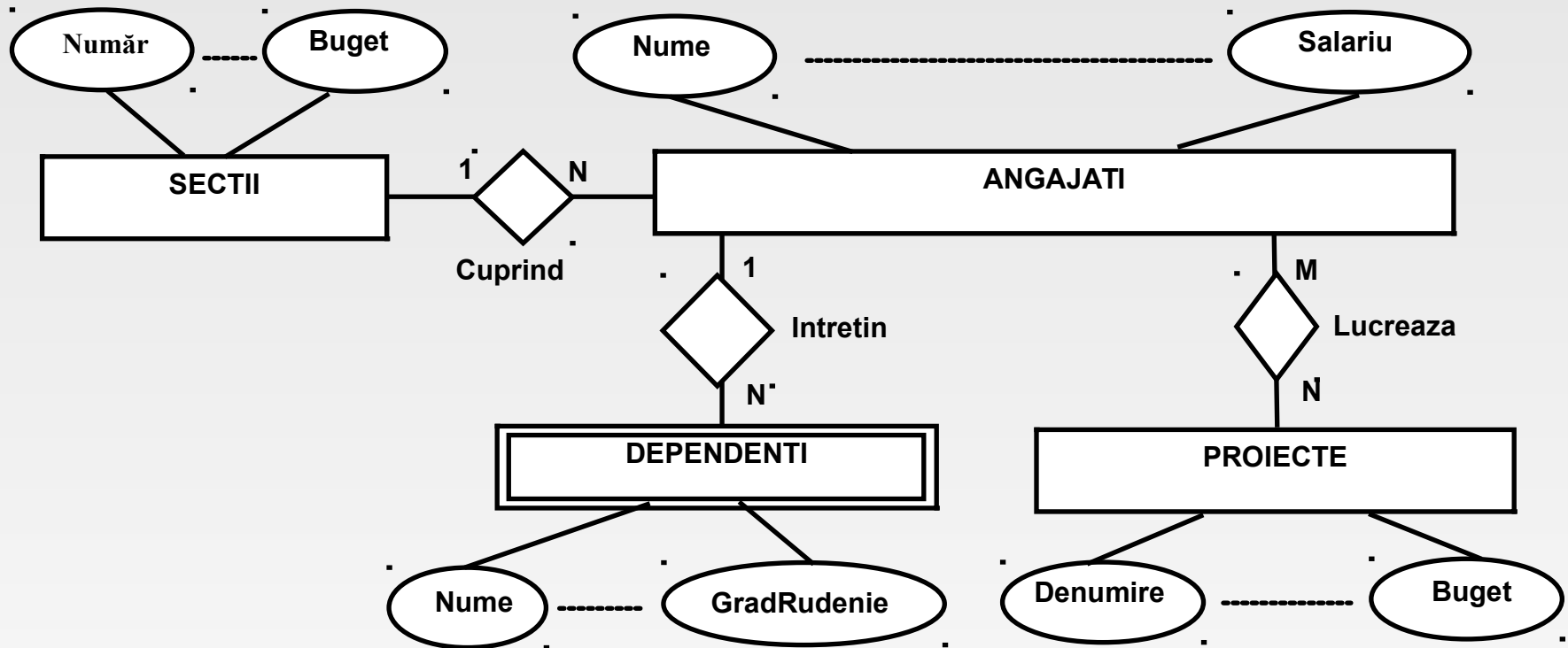
Diagrama Entitate-Asociere

- **Diagrama Entitate-Asociere** (*Entity-Relationship Diagram*) reprezintă grafic modelul Entitate-Asociere prin mulțimile de entități și asocierile dintre acestea
- Multimi de entitati de același tip:
 - Puternice (de sine statatoare)
 - Slabe (depind de alte multimi de entitati)
- Notatii:



Exemplu de diagrama Entitate-Asociere (1)

- Multimi de entitati puternice:
 - SECTII (Numar, Nume, Buget)
 - ANGAJATI (Nume, Prenume, DataNasterii, Adresa, Functie, Salariu)
 - PROIECTE (Denumire, DataInceperii, Termen, Buget)
- Multimi de entitati slabe: DEPENDENTI (Nume, Prenume, DataNasterii, GradRud)

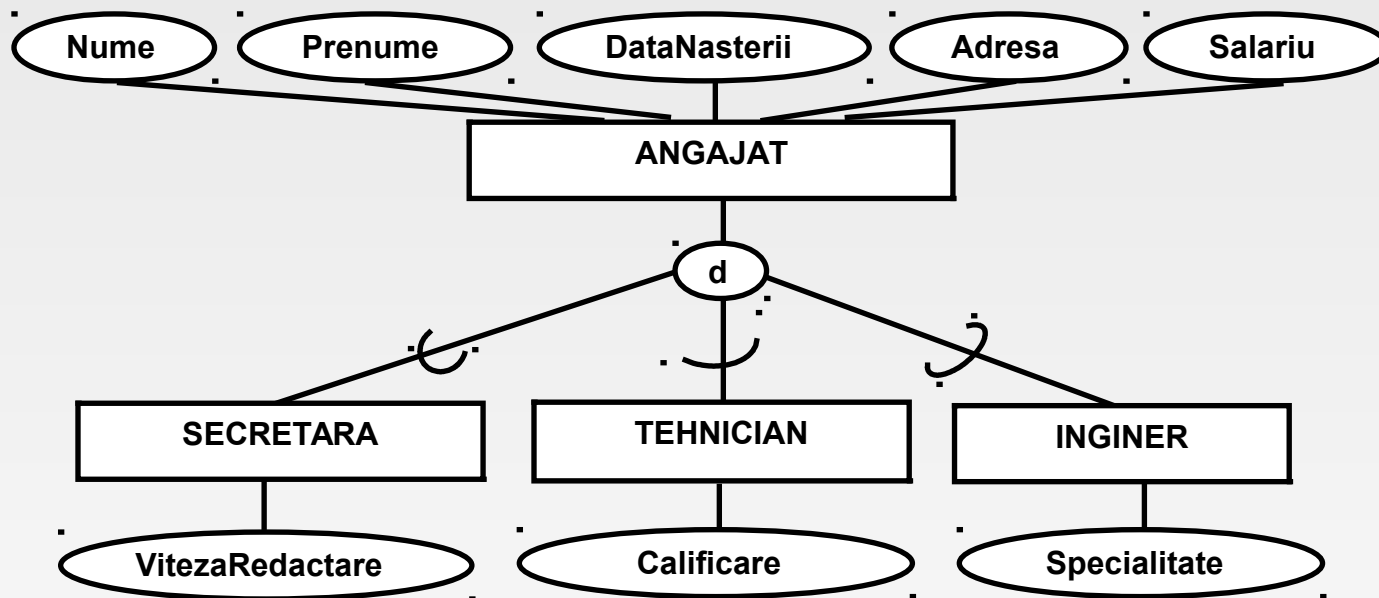


Exemplu de diagrama Entitate-Asociere (2)

- Asocieri:
 - Asocierea SECTII - ANGAJATI - 1:N
 - Asocierea ANGAJATI - PROIECTE - M:N
 - Asocierea ANGAJATI - DEPENDENTI - 1:N
- Raportul de cardinalitate al unei asocieri *este stabilit de proiectant* astfel încât să reflecte cât mai corect modul de organizare a activității modelate
- Modul de stabilire a tipurilor de entități și a asocierilor nu este unic: aceeași funcționalitate se poate obține printr-o varietate de diagrame E-A
- O mulțime de entități se denumeste printr-un substantiv, iar o asociere se denumeste (de regulă) printr-un verb, deoarece o asociere reprezintă o interacțiune între entități
- Modelul E-A nu precizează modul în care sunt realizate asocierile între mulțimile de entități: acest aspect depinde de modelul de structurare a datelor utilizat pentru baza de date:
 - în modelul ierarhic și rețea asocierile sunt realizate prin pointeri de la o entitate la entitățile asociate;
 - în modelul relațional asocierile se realizează prin egalitatea valorilor unor attribute comune ale multimilor de entități (chei);
 - în modelul obiect-relațional asocierile se reprezintă prin referințe care folosesc indentificatori unici ai obiectelor.

Modelul Entitate-Asociere Extins

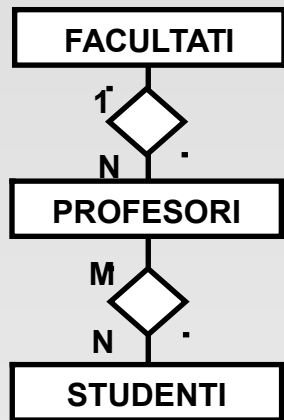
- Modelul Entitate-Asociere Extins (*Enhanced Entity-Relationship Model*) permite definirea de subtipuri ale unui tip de entităţi, care moştenesc atribute de la tipul de entitate respectiv
- Crearea ierarhiilor: *specializare* si *generalizare*
- Tipurile şi a subtipurile formeaza ierarhii de tipuri de entităţi complexe, organizate pe mai multe niveluri
- Diagrama Entitate-Asociere Extinsa



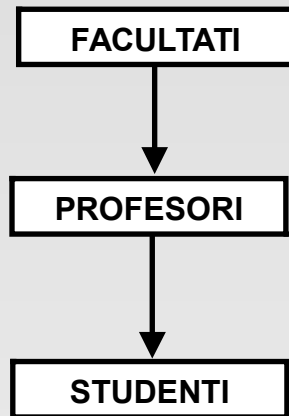
Modelul de date ierarhic

- **Modelul ierarhic** (*Hierarchical Model*): baza de date se reprezinta printr-un arbore: structură ierarhică de înregistrări (*records*) conectate prin legături (*links*)
 - A fost primul model folosit pentru dezvoltarea bazelor de date
 - Cel mai cunoscut SGBD ierarhic: sistemul IMS (*Information Management System*) dezvoltat de IBM în programul de cercetări Apollo, în 1960
- **O înregistrare de date** este o instanță a unui tip de înregistrare (*record type*) și constă dintr-o colecție de câmpuri (*fields*), fiecare câmp conținând valoarea unui atribut
- **O legătură (link)** este o instanță a unui tip de legătură între o înregistrare părinte și o înregistrare fiu
- Se pot reprezenta numai legături de tipul părinte-fiu, care corespund asocierilor 1:1 și 1:N din modelul E-A
- Schema conceptuală a unei baze de date în modelul ierarhic se reprezintă printr-un număr oarecare de scheme ierarhice
- O schemă ierarhică este un arbore direcționat, reprezentat pe mai multe niveluri, în care nodurile sunt tipuri de înregistrări (*records*), iar arcele sunt tipuri de legături (*link-uri*)

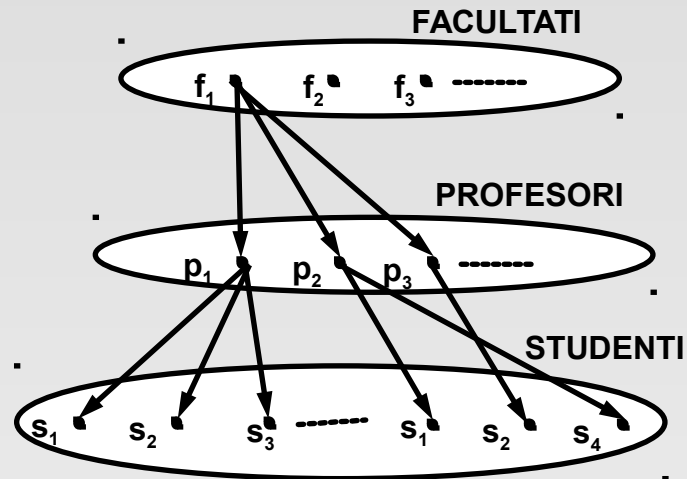
Baze de date ierarhice



(a) Diagrama E-A



(b) Schema ierarhica



(c) Arbori de
instantiere a datelor

- Asocierile M:N se pot reprezenta prin multiplicarea înregistrărilor de tip fiu, atunci când sunt referite de mai multe înregistrări de tip părinte
 - Această multiplicare produce redundanța datelor
- Bazele de date ierarhice
 - Sunt ineficiente pentru aplicații de uz general (de ex. financiar-contabile)
 - Sunt eficiente în acele aplicații specializate în care asocierile sunt de tipul arbore
- Utilizarea bazelor de date ierarhice
 - Initial (1960) s-au utilizat în diferite aplicații, fiind singurul model la acea vreme
 - După apariția modelului relațional (1970) au fost abandonate
 - Actualmente (2015) s-a reluat acest model și se folosește în aplicații specializate, baze de date XML etc.

Modelul de date retea

- **Modelul rețea** (*Network Model*) folosește o structură de graf pentru definirea schemei conceptuale a bazei de date
- Modelele ierarhic si retea → modele pre-relationale
- Standardizat în 1971, de o comisie DBTG (*Database Task Group*).
- Sisteme de gestiune comerciale in modelul retea: IDS II (Honeywell), UNISYS (Burroughs), IDMS (Computer Associates)
- Nodurile grafului sunt tipuri de entități (*înregistrări - records*), iar muchiile reprezintă asocierile (*legăturile-links*) dintre tipurile de entități
- Asocierile M:N se reprezintă fără duplicarea înregistrărilor, fiecare înregistrare putând fi referită de mai multe înregistrări, ceea ce elimină redundanța datelor
- La fel ca bazele de date ierarhice, bazele de date rețea (graf)
 - Sunt ineficiente pentru aplicații de uz general (financiar-contabile)
 - Sunt eficiente în acele aplicații specializate în care asocierile sunt de tipul graf
- Utilizarea bazelor de date rețea:
 - S-au utilizat initial, după definirea lor, dar au fost abandonate după extinderea modelului relațional
 - Actualmente (2015) se folosesc în aplicații specializate, în care asocierile între tipurile de înregistrări sunt complexe (grafuri); de ex: baze de date grafice (scene virtuale), baze de date pentru rețele sociale, analiza volumelor mari de date etc.
 - Produse comerciale actuale: InfiniteGraph (compania Objectivity Inc)

Modelul de date relational

- **Modelul relațional** (*Relational Model*) se bazează pe noțiunea de **relație** (*relation*) din matematică, care corespunde unei mulțimi de entități
- Fundamentat de E.F. Codd (IBM), prin lucrarea "Un Model Relațional de Date pentru Bănci Mari de Date Partajate" (1970)
- Dezvoltare extraordinară a sistemelor de gestiune a bazelor de date relationale, datorită simplității și a fundamentării matematice a modelului
- Alte lucrări ale cercetătorilor C.J. Date, P. Chen, R. Boyce, J.D. Ullman, R. Fagin, W. Armstrong, M. Stonebraker etc. au perfecționat modelul relațional
- Primul Sistem de Gestiune a Bazelor de Date Relaționale (SGBDR) a fost prototipul System R (IBM, 1970)
- După aceasta numeroase companii au realizat sisteme de gestiune relaționale: Oracle, Microsoft, Ingres, Sybase, IBM, Informix
- SGBDR folosesc limbajul SQL (*Structured Query Language*), pentru care au fost emise mai multe standarde ANSI (American National Standardization Institute) și ISO (International Standardization Office)
- Majoritatea SGBD-urilor relaționale actuale implementează versiunea SQL2 (SQL92) sau versiuni ulterioare (SQL-1999, SQL-2003, SQL-2006)

Modelul obiect-orientat

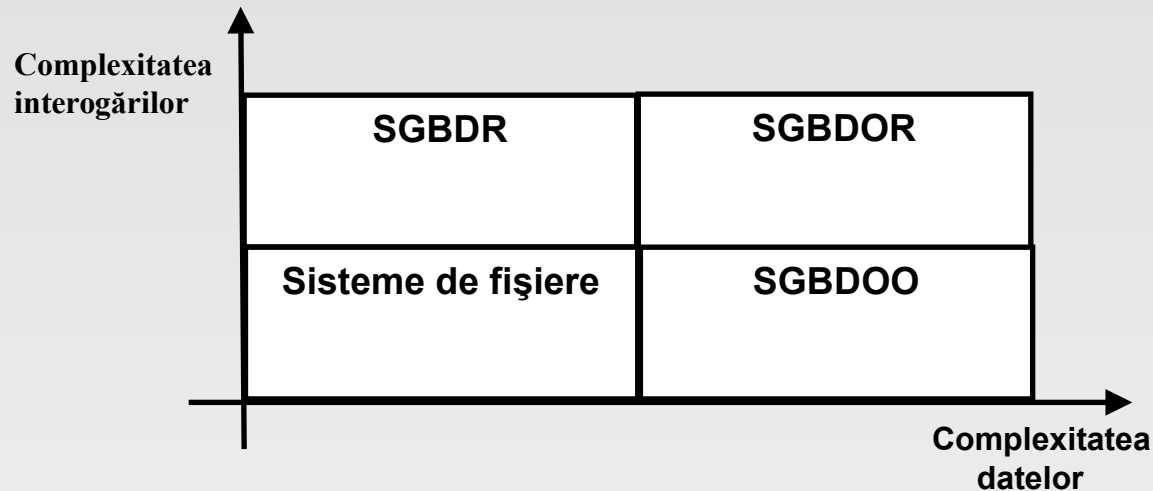
- **Modelul obiect** (*Object Model*) este un concept unificator
- Necesari în domenii în care se manipulează date de tipuri complexe:
 - proiectarea sistemelor de calcul: programare, hardware, interfețe
 - proiectarea asistată de calculator (CAD-CAM)
 - sisteme de informații geografice
 - fizică, biologie, medicină și altele
- Bazele de date obiect-orientate trebuie să integreze limbajele obiect-orientate cu sistemele de baze de date (cu date persistente)
- Este necesară extinderea limbajelor de progr. obiect-orientate pentru a oferi:
 - suport pentru persistența obiectelor → limbaje de programare persistente
 - suport pentru reprezentarea asocierilor între obiecte persistente
 - suport pentru interogarea bazelor de date și tranzacții
- Standardul ODMG (*Object Data Management Group* -ODMG 3.0 - 1999):
 - Modelul obiect pentru baze de date (*ODMG Object model*)
 - ODL (*Object Definition Language*)
 - OQL (*Object Query Language*)
 - Corespondența cu limbaje de programare obiect-orientate de implementare (binding) (C++, Java, Smalltalk)
- Utilizare SGBDOO: cam 5% din sistemele de gestiune actuale
- Sisteme comerciale: ObjectivityDB, ObjectStore, db4o etc.

Modelul obiect-relational

- **Modelul obiect-relațional** (*Object-Relational Model*) reprezintă extinderea modelului relațional cu caracteristici ale modelului obiect
- Modelul obiect-relațional păstrează structurarea datelor în relații, și, în plus:
 - permite definirea unor noi tipuri de date, ca domenii ale atributelor
 - permite definirea unor operatori pentru tipurile de date definite de utilizator
 - permite extinderea tipurilor de date prin moștenire
- Sistemele de gestiune a bazelor de date obiect-relaționale (SGBDOR) se realizează prin extinderea sistemelor relaționale, de regula în mod gradat, adăugându-se de la o versiune la alta cât mai multe caracteristici posibile ale modelului obiect
- Aceasta abordare asigură rularea în continuare a aplicațiilor relaționale existente în noile versiuni de sisteme SGBDOR, ceea ce permite producătorilor să-și păstreze clienții și domeniile de utilizare
 - Așa s-au extins SGBD-urile de la Oracle, IBM, Microsoft etc.
- Limbajele de programare pentru SGBDOR sunt definite în standardele mai recente ale limbajului SQL: SQL3 (SQL-1999), SQL-2003, SQL-2006

Complexitatea datelor si a interogărilor

- Clasificare propusa de M. Stonebraker (1996)

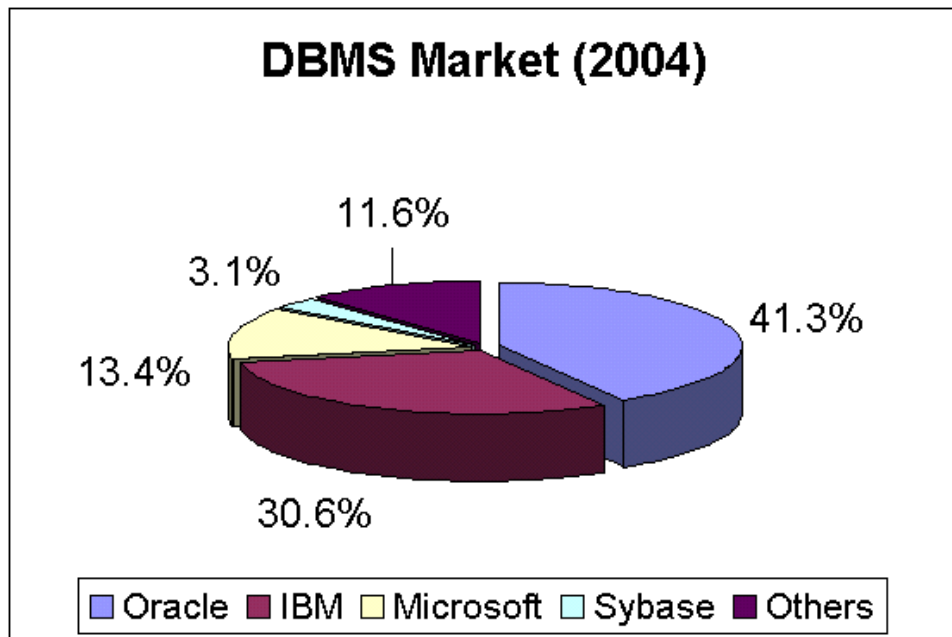


- SGBDR prelucrează tipuri simple de date, dar permit interogări complexe
- SGBDOO prelucrează tipuri de date complexe, dar rezolvarea interogărilor complexe este destul de dificilă
- SGBDOR permit prelucrarea datelor complexe și rezolvarea interogărilor complexe; sistemele de baze de date obiect-relaționale sunt considerate sisteme de **baze de date universale**

Evolutia sistemelor de baze de date

- 1960 Modele prerelationale: ierarhic si retea
 - Primele produse de baze de date (DBOM, IMS, IDS, Total, IDMS)
 - Standarde Codasyl
- 1970 Modelul relational – prototipuri de SGBDR
 - Lucrari teoretice asupra modelului relational
 - Modelul Entitate-Asociere
- 1980 Dezvoltarea SGBDR comerciale
 - Primul standard SQL (1986 - ANSI, ISO)
- 1990 Arhitectura client/server a sistemelor de baze de date
 - Baze de date obiect-orientate
 - Baze de date obiect-relationale
 - Baze de date distribuite
 - Standarde SQL: SQL 92, SQL 99
- 2000 Arhitectura pe 3 niveluri a aplicațiilor de baze de date (three-tier arch.)
 - Baze de date in sistemul WWW (World Wide Web)
 - Baze de date XML, multimedia
- 2010 Baze de date în sisteme Cloud Computing

Sisteme de Gestiune a Bazelor de date



Sisteme Comerciale

Oracle (\$\$\$\$)

DB2 (IBM) (\$\$\$)

SQL Server (Microsoft) (\$\$)

Sisteme Open Source

PostgreSQL

MySQL