

Baze de date

Curs - anul III B

Sl.Dr.Ing. Valentin Pupezescu

Conținutul cursului

- Introducere: definiții și clasificări ale bazelor de date
- Modelarea datelor
- Baze de date relaționale. Limbajul SQL
- Interogarea bazelor de date
 - Dezvoltarea bazelor de date (proiectare și implementare)
 - Dezvoltarea aplicațiilor de baze de date
- Gestiunea tranzacțiilor
- Normalizarea relațiilor

Capitolul 1 - introducere

- Definiții - baze de date , sisteme de baze de date
- Componentele 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 sistemelor de calcul utilizate
- Modelarea datelor
 - Modele conceptuale de nivel înalt
 - Modele specifice de baze de date
- Evoluția sistemelor de gestiune a bazelor de date

Sisteme de baze de date

- Domeniile de aplicabilitate ale bazelor de date:
 - Activități bancare și comerciale
 - Producție
 - Machine learning, inteligență artificială
 - Internet of Things
 - Servicii (medicale, rezervări bilete de tren/avion/film etc.)
- **Definitie (în sens larg):** O bază de date (database) este o colecție de date corelate din punct de vedere logic, care reflectă (modelează) un anumit aspect al lumii reale și este destinată unui anumit grup de utilizatori. În acest sens pot fi considerate ca fiind “baze de date”: fișe de evidență (menținute manual), fișiere de documente (Word, Excel etc.), baze de date menținute computerizat.
- **Definitie (în sens restrâns, actual):** O bază de date este o colecție de date creată și menținută computerizat, care permite operații de tip creare, citire, modificare, ștergere - CRUD (Create, Read, Update, Delete).

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.
- **Componenta hardware:**
 - Sistemele de baze de date sunt instalate pe calculatoare de uz general.
 - Bazele de date sunt memorate fizic ca fișiere pe hdd-uri, ssd-uri sau direct în memorie (pentru creșterea performanțelor la interogare).
 - Dimensiunea și performanțele bazei de date depind de sistemul de calcul pe care este instalată
 - Pentru creșterea disponibilității datelor și pentru a spori rezistența la defecte se poate opta pentru o implementare distribuită a datelor pe mai multe sisteme de calcul.
- **Componenta software:**
 - Sisteme de operare, biblioteci, instrumente de dezvoltare și interfețe.
 - Sistemul de gestiune a bazelor de date (SGBD) - Database Management System (DBMS). Acesta recepționează cererile de acces la bazele de date lansate de utilizatori, le interpretează, execută operațiunile corespunzătoare și returnează rezultatele.
 - Aplicații de baze de date: (Database Applications) – sunt programe care creează și utilizează bazele de date.

Componentele unui sistem de baze de date (2)

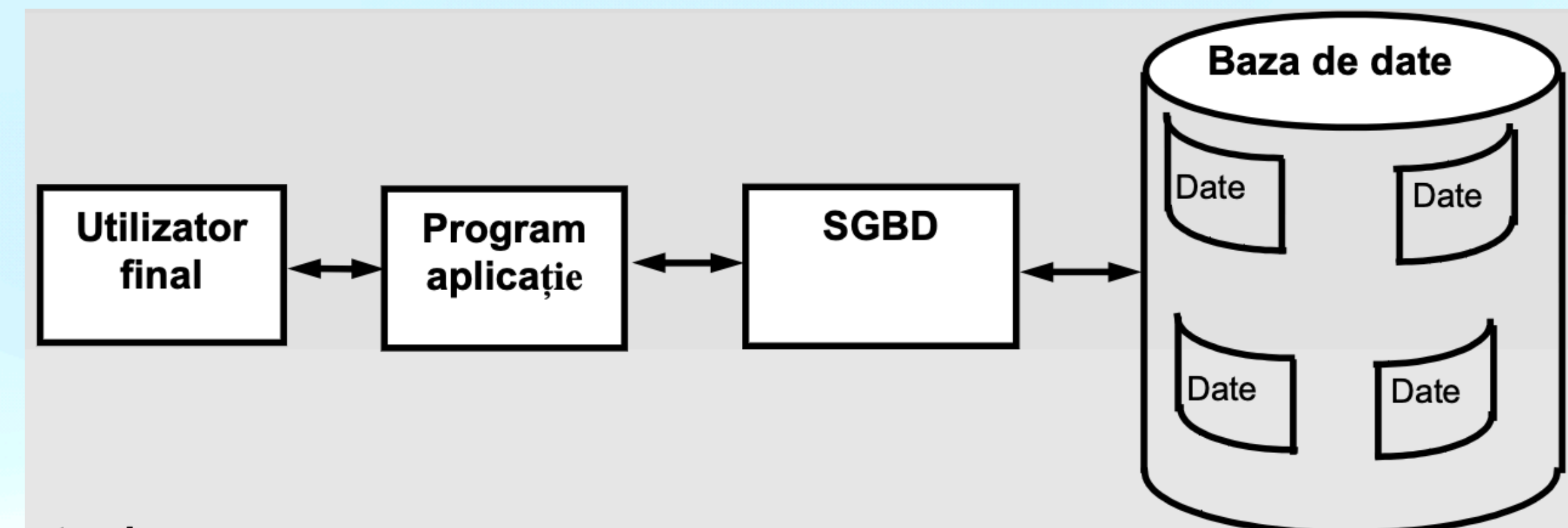
- **Componenta - utilizatori:**

- Administratorul bazei de date
- Analisti (data scientists) și proiectanți ai bazelor de date
- Programatori de aplicații: web / mobile / stand-alone

- **Componenta - date persistente:** datele sunt stocate pe hdd / ssd (se pot utiliza Configurații RAID) / in-memory

- 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)



Componentele unui sistem de baze de date

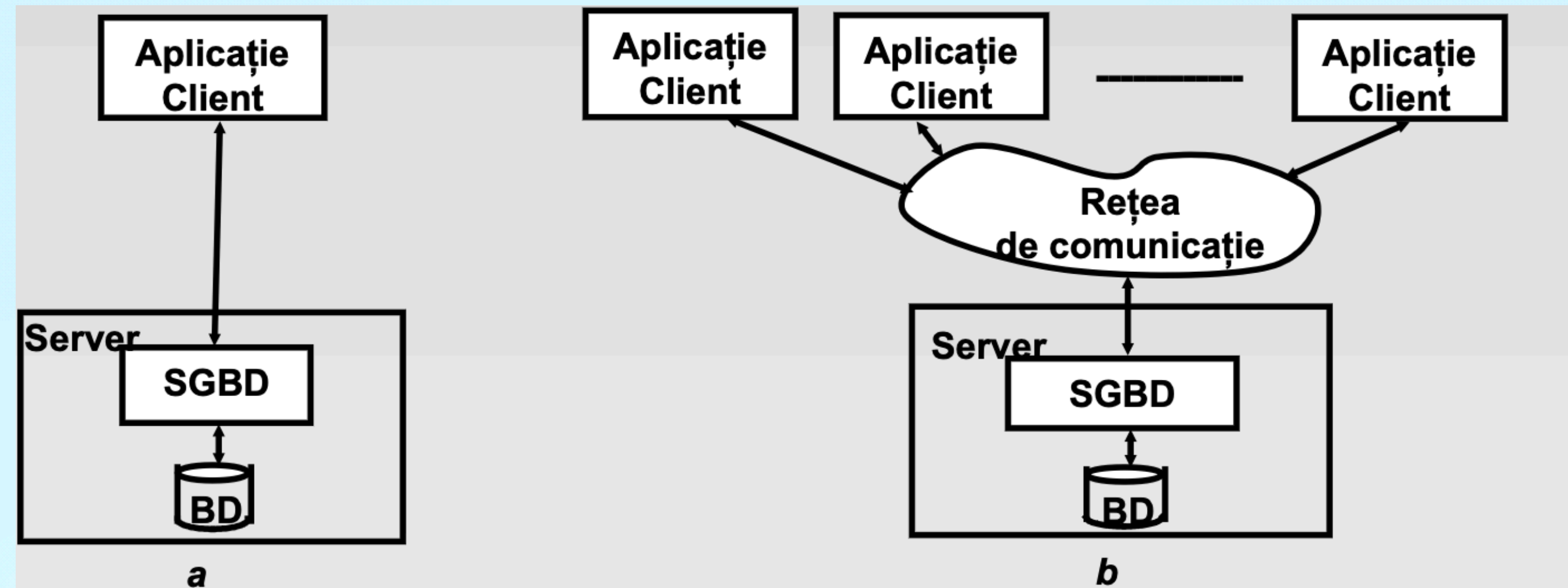
Avantajele sistemelor de baze de date

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

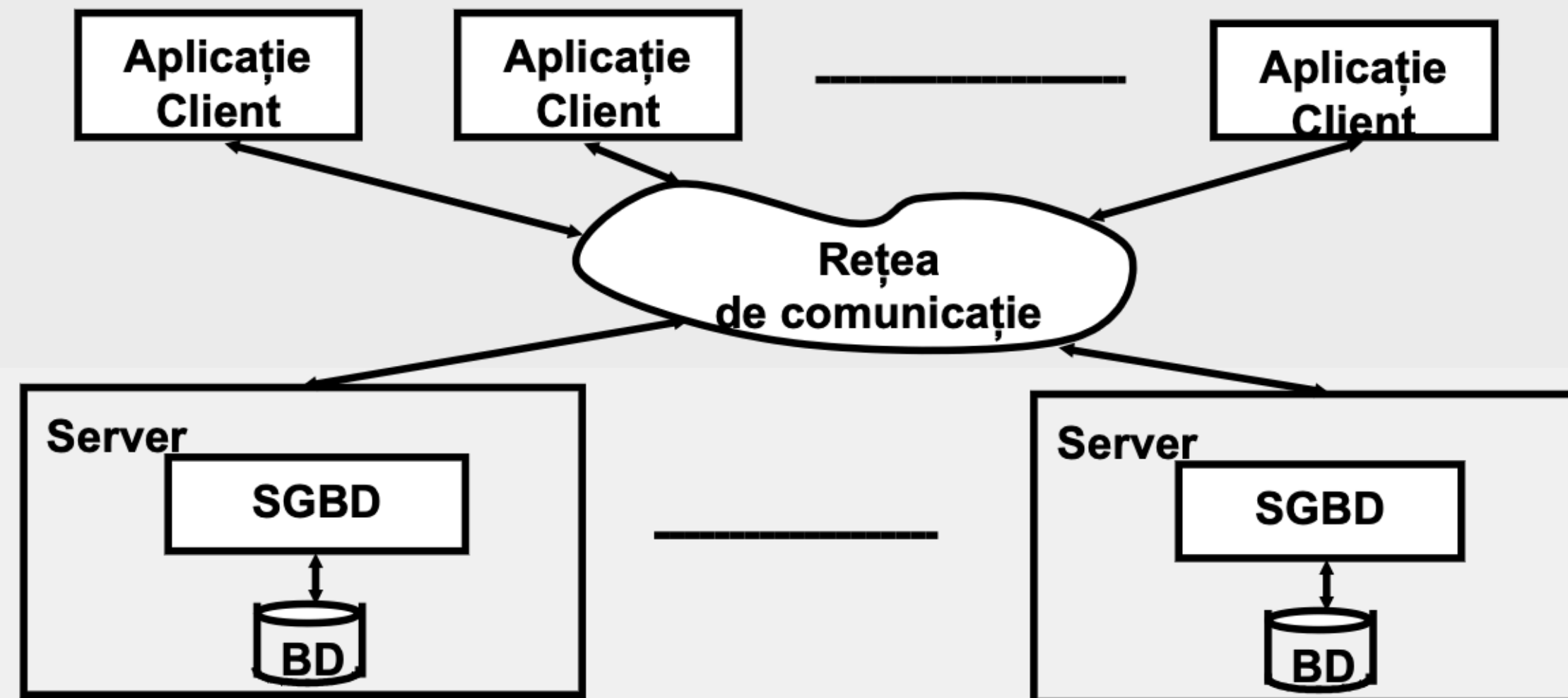
Clasificarea sistemelor de baze de date (1)

- Arhitectura client-server:
 - Server (back-end) - SGBD si baza de date
 - Client (front-end) - aplicația
- Clasificare după modelul de date:
 - Model ierarhic de date
 - Model de date rețea (orientat pe grafuri)
 - Model de date relațional
 - Model de date obiect-orientat
 - Model de date obiect-relațional
 - Model de date orientat pe colecții de documente
- Clasificare după numărul de utilizatori
 - Sisteme mono-utilizator
 - Sisteme multi-utilizator
- Clasificare după numărul de stații de calcul pe care este stocată 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

Clasificarea sistemelor de baze de date

Modelarea datelor

- Un **model** este o abstractizare a unui sistem:
 - Captează trăsăturile principale ale unui sistem (concepte) - tehnica de identificare a acestor trăsături se numește abstractizare.
 - Conceptele trebuie să fie relevante din punct de vedere al scopului pentru care se definește modelul respectiv
- 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 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 asocierile dintre acestea prin anumite structuri de date (arbori, grafuri, tabele) - se reprezintă printr-o schemă conceptuală (logică).

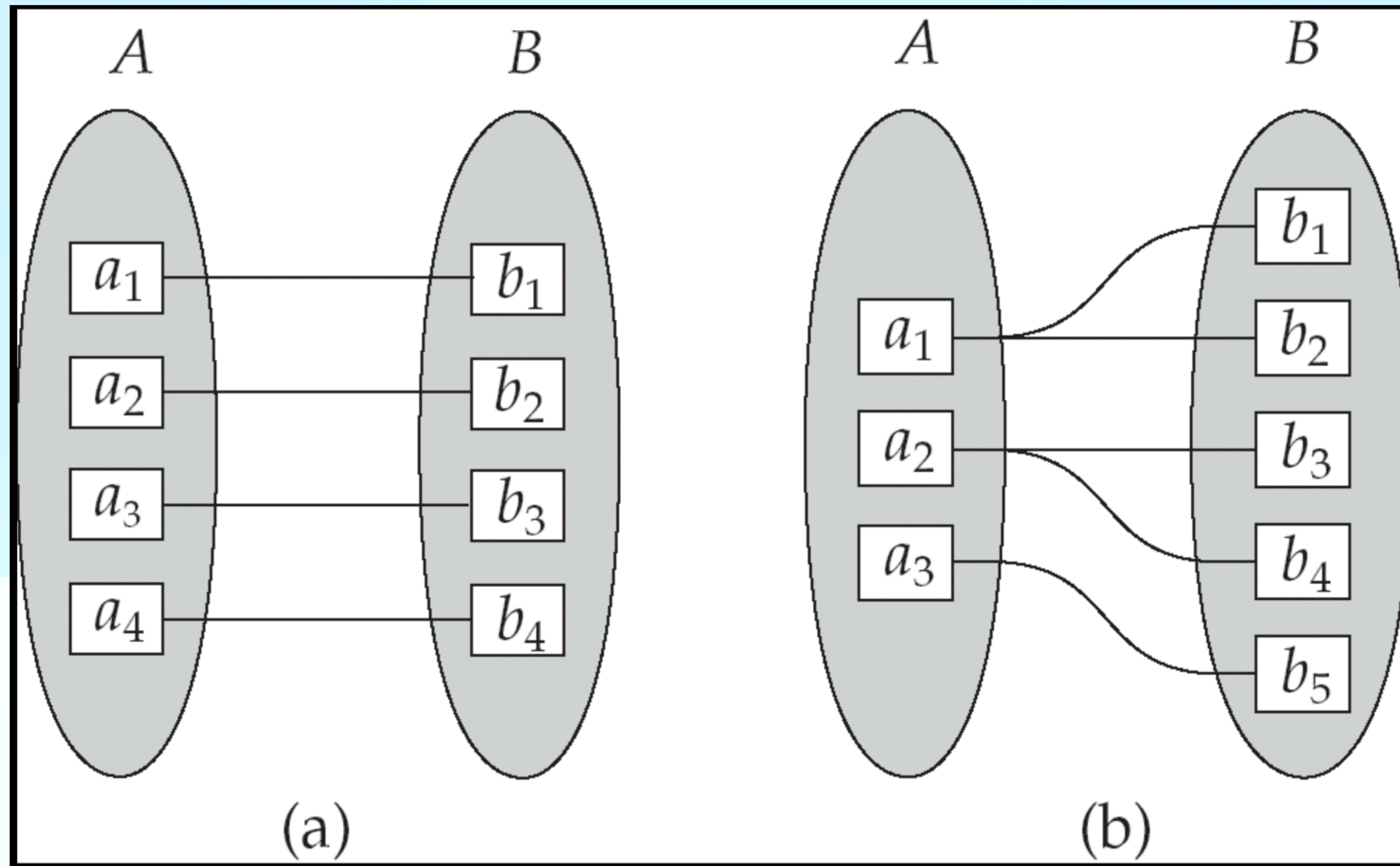
Modelul entitate-asociere

- **Modelul Entitate-Asociere** (Entity-Relationship Model) definește mulțimile 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 există 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, adresă etc.
- **Tip de entitate** (entity type): se referă la entitățile similare, care pot fi descrise prin aceleași atribute - exemple: tipul persoană, tipul plantă.
- **Mulțime de entități** (entities set): colecția tuturor entităților de același tip dintr-o bază de date constituie o mulțime de entități - exemple: mulțimea tuturor persoanelor, mulțimea tuturor plantelor.
- O entitate este o instanță a unui tip de entitate și un element al mulțimii de entități de acel tip. În exprimarea curentă, adeseori nu se face diferențierea dintre entitate, tip de entitate și mulțime de entități, dar diferența este evidentă. A se vedea asemănarea 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 attribute.
- **Tipul asocierii** (relationship type) – se referă la asocierile similare, care pot fi definite între entități din două sau mai multe mulțimi de entități.
- **Mulțime 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 mulțime 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: soț-soție;
 - “unul-la-multe” (one-to-many) – 1:N; exemplu: părinte-fii;
 - “multe-la-unul” (many-to-one) – N:1; exemplu: fii-părinte;
 - “multe-la-multe” (many-to-many) – M:N; exemplu: profesori-studenți;

Categorii de asocieri binare (1)

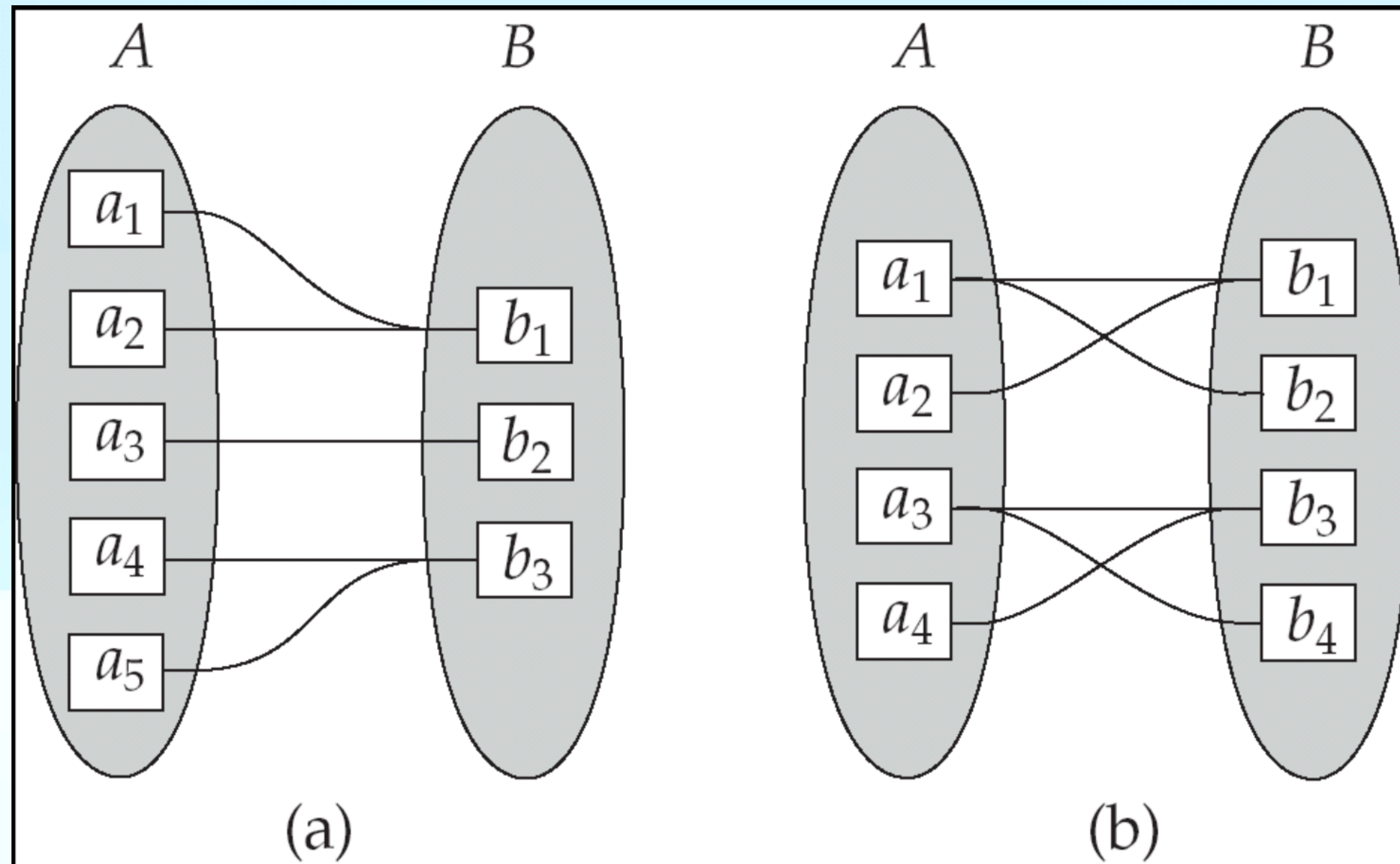


“unul-la-unul” – 1:1

“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

Tipuri de asocieri binare

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 entități de același tip:
 - Puternice (de sine stătătoare);
 - Slabe (depind de alte mulțimi de entități).
- Notatii:

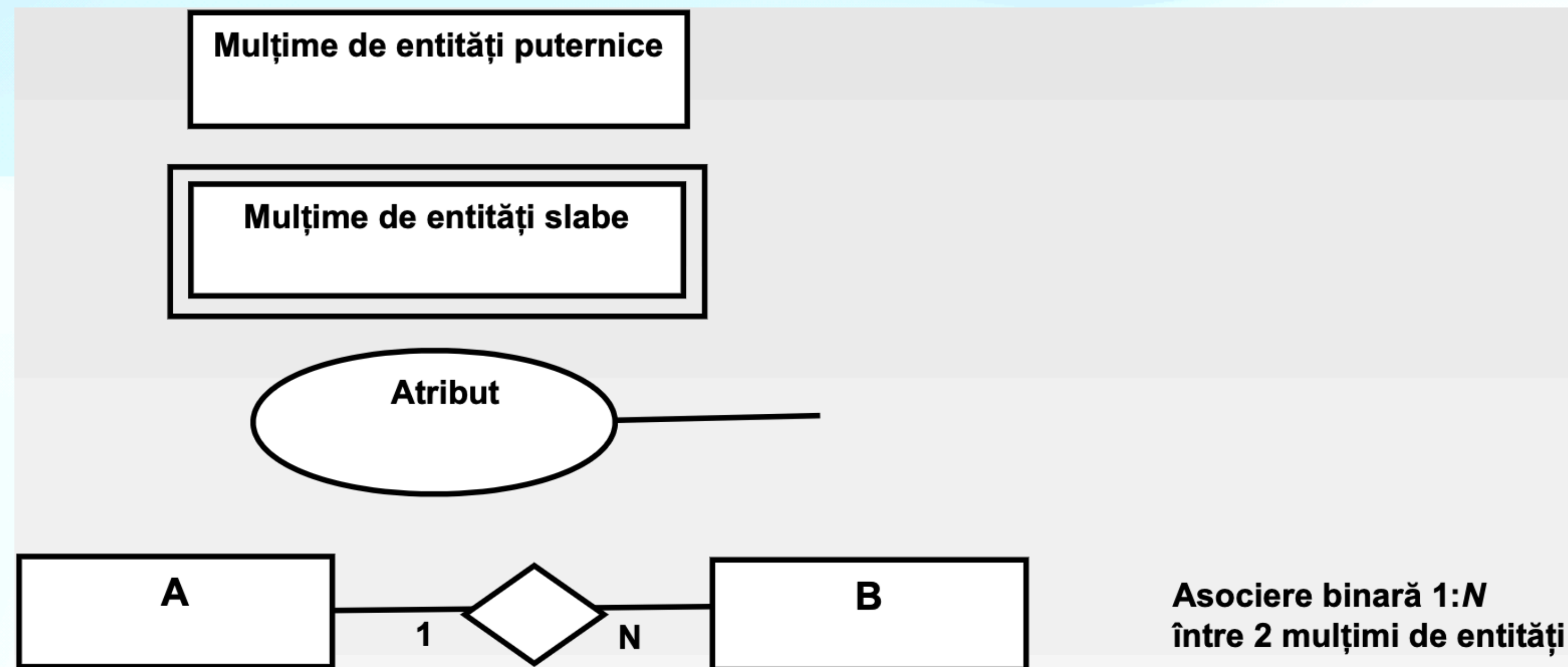
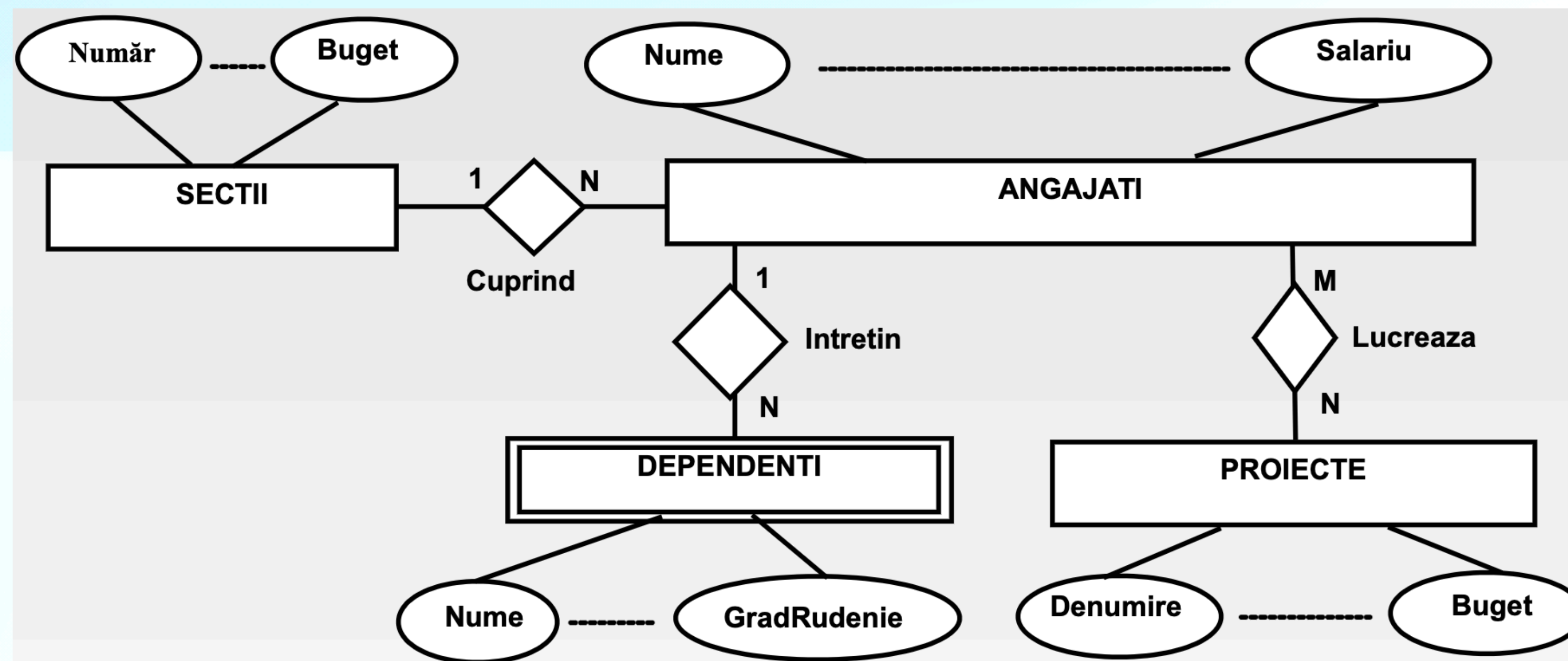


Diagrama E-A

Exemplu de diagramă entitate-asociere (1)

- Mulțimi de entități puternice:
 - SECTII(Numar, Nume, Buget)
 - ANGAJATI(Nume, Prenume, DataNasterii, Adresa, Functie, Salariu)
 - PROIECTE(Denumire, DataInceperii, Termen, Buget) etc.
- Multimi de entități slabe: DEPENDENTI(Nume, Prenume, DataNasterii, GradRudenie).



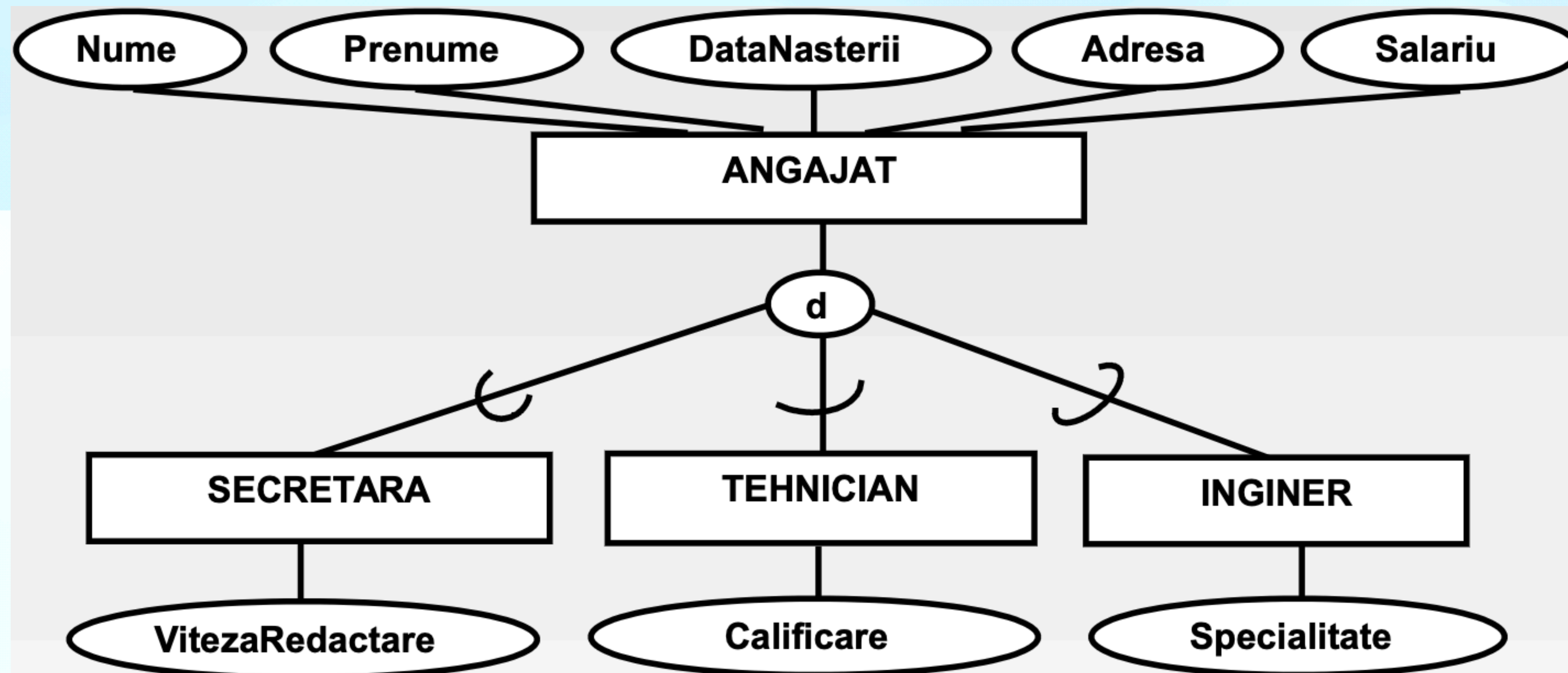
Exemplu de diagrama E-A

Exemplu de diagramă 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 denumește 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 mulțimilor 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 attribute de la tipul de entitate respectiv.
- Crearea ierarhiilor: specializare si generalizare.
- Tipurile şi a subtipurile formează ierarhii de tipuri de entităţi complexe, organizate pe mai multe niveluri.



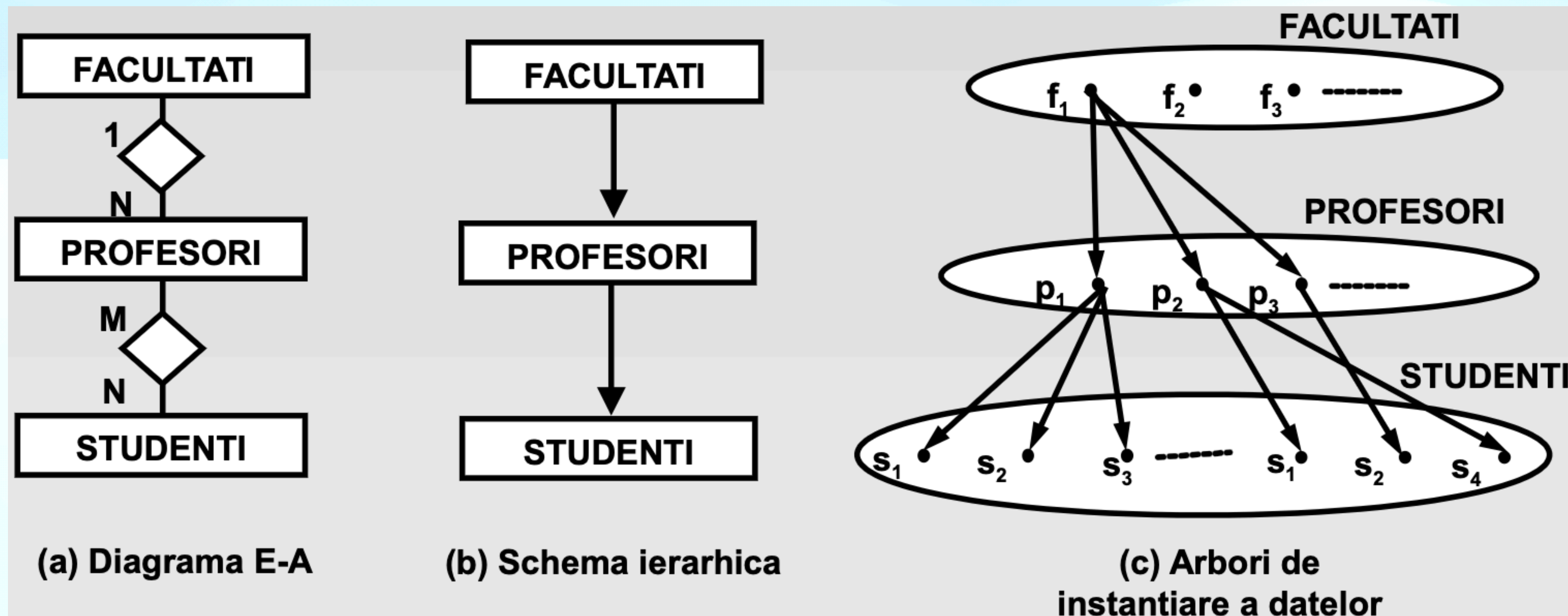
Exemplu de diagrama entitate-asociere extinsă

Modelul de date ierarhic

- **Modelul ierarhic** (Hierarchical Model): baza de date se reprezintă 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.
 - În prezent avem SGBD-uri precum Oracle care are și posibilitatea de lucru cu XML-uri (OracleXML). Alt SGBD care permite manipularea datelor stocate în fișiere XML este BaseX.
- **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

- 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 eficiente în acele aplicații specializate în care asocierile sunt de tipul arbore;
 - Sunt ineficiente pentru aplicații de uz general.
- 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.



Reprezentarea bazelor de date ierarhice

Modelul de date rețea (orientat pe grafuri)

- Modelul rețea (Network Model) folosește o structură de graf pentru definirea schemei conceptuale a bazei de date.
- Modelele ierarhic si rețea sunt modele pre-relationale
- Standardizat în 1971, de o comisie DBTG (Database Task Group).
- 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 eficiente în acele aplicații specializate în care asocierile sunt de tipul graf (în special pentru identificarea fraudelor din domeniul bancar, aplicații de tip social-media etc.);
 - Nu sunt întotdeauna eficiente pentru aplicații de uz general.
- Utilizarea bazelor de date rețea:
 - S-au utilizat initial, după definirea lor, dar au fost abandonate după extinderea modelului relațional;
 - Actualmente 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: Neo4j, InfiniteGraph (compania Objectivity Inc).

Modelul de date relațional

- **Modelul relațional** (Relational Model) se bazează pe noțiunea de relație (relation) din matematică, care corespunde unei mulțimi de entități.
- Acesta a fost 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 relaționale, 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, MySql, PostgreSQL, 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 de date obiect-orientat

- **Modelul obiect** (Object Model) este un concept unificator, necesar 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). Astfel, este necesară extinderea limbajelor de programare 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 de date obiect-relațional

- **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 regulă în mod gradual, 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, PostgreSQL etc. Limbajele de programare pentru SGBDOR sunt definite în standardele mai recente ale limbajului SQL: SQL3 (SQL-1999), SQL-2003, SQL-2006

Modelul de date orientat pe colecții de documente

- **Modelul orientat pe colecții de documente** este bazat pe colecții de documente non-relaționale de tip JSON (JavaScript Object Notation).
- Acesta este non-relațional (NoSql) - nu se aplică operații de algebră relațională pe colecțiile de documente (exemplu - SGBD-ul MongoDB).
- Formatul acestora este următorul:
- ```
{ 'Nume' : 'Ionescu', 'Prenume' : 'Ion', 'Locuri de munca' : ['Luxsoft' , 'Ubisoft' , 'EA'], 'Adrese' : [{ 'Oras' : 'Pitesti', 'Strada' : 'Cuza Nr.1', 'Bloc' : 'Z3'}, { 'Oras' : 'Bucuresti', 'Strada' : 'Barajul Rovinari', 'Bloc' : 'T5'}] }
```
- Aceste SGBD-uri nu o structură uniformă de stocare a datelor (schemaless).
- Caracteristicile acestor SGBD-uri:
  - Model de date orientat pe documente cu scheme dinamice;
  - Suport complet pentru indecși , flexibil și interogări bogate;
  - Auto-Sharding pentru scalabilitate orizontală;
  - Replicare încorporată pentru disponibilitate ridicată a datelor;
  - Funcții de căutare - text;
  - Securitate avansată;
  - Framework de agregare și MapReduce;
  - Depozitare mare de conținut media cu GridFS.



# Complexitatea datelor și a interogărilor

- Clasificare propusă de M. Stonebraker (1996)



Clasificarea SGBD-urilor

- 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.



# Evoluția 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



# Topul sistemelor de gestiune a bazelor de date

410 systems in ranking, February 2023

| Rank     |          |          | DBMS                         | Database Model               | Score    |          |          |
|----------|----------|----------|------------------------------|------------------------------|----------|----------|----------|
| Feb 2023 | Jan 2023 | Feb 2022 |                              |                              | Feb 2023 | Jan 2023 | Feb 2022 |
| 1.       | 1.       | 1.       | Oracle +                     | Relational, Multi-model ⓘ    | 1247.52  | +2.35    | -9.31    |
| 2.       | 2.       | 2.       | MySQL +                      | Relational, Multi-model ⓘ    | 1195.45  | -16.51   | -19.23   |
| 3.       | 3.       | 3.       | Microsoft SQL Server +       | Relational, Multi-model ⓘ    | 929.09   | +9.70    | -19.96   |
| 4.       | 4.       | 4.       | PostgreSQL +                 | Relational, Multi-model ⓘ    | 616.50   | +1.65    | +7.12    |
| 5.       | 5.       | 5.       | MongoDB +                    | Document, Multi-model ⓘ      | 452.77   | -2.42    | -35.88   |
| 6.       | 6.       | 6.       | Redis +                      | Key-value, Multi-model ⓘ     | 173.83   | -3.72    | -1.96    |
| 7.       | 7.       | 7.       | IBM Db2                      | Relational, Multi-model ⓘ    | 142.97   | -0.60    | -19.91   |
| 8.       | 8.       | 8.       | Elasticsearch                | Search engine, Multi-model ⓘ | 138.60   | -2.56    | -23.70   |
| 9.       | ↑ 10.    | ↑ 10.    | SQLite +                     | Relational                   | 132.67   | +1.17    | +4.30    |
| 10.      | ↓ 9.     | ↓ 9.     | Microsoft Access             | Relational                   | 131.03   | -2.33    | -0.23    |
| 11.      | ↑ 12.    | 11.      | Cassandra +                  | Wide column                  | 116.22   | -0.09    | -7.76    |
| 12.      | ↓ 11.    | ↑ 15.    | Snowflake +                  | Relational                   | 115.65   | -1.60    | +32.47   |
| 13.      | 13.      | ↓ 12.    | MariaDB +                    | Relational, Multi-model ⓘ    | 96.81    | -2.55    | -10.30   |
| 14.      | 14.      | ↓ 13.    | Splunk                       | Search engine                | 87.08    | -1.32    | -3.73    |
| 15.      | 15.      | ↑ 17.    | Amazon DynamoDB +            | Multi-model ⓘ                | 79.69    | -1.87    | -0.67    |
| 16.      | 16.      | ↓ 14.    | Microsoft Azure SQL Database | Relational, Multi-model ⓘ    | 78.75    | -1.62    | -6.20    |
| 17.      | 17.      | ↓ 16.    | Hive                         | Relational                   | 72.12    | -2.22    | -9.76    |
| 18.      | 18.      | 18.      | Teradata                     | Relational, Multi-model ⓘ    | 63.03    | -2.40    | -5.54    |
| 19.      | 19.      |          | Databricks                   | Multi-model ⓘ                | 60.33    | -0.49    |          |
| 20.      | 20.      | 20.      | Neo4j +                      | Graph                        | 55.43    | -0.41    | -2.81    |

Topul SGBD-urilor - februarie 2023