

# BAZE DE DATE

Proiectarea bazelor de date relaționale

Modelul entitate-asociere

Mihaela Elena Breabăn

© FII 2013-2014

# Modelul entitate-asociere (Entity/Relationship)

## Obiective

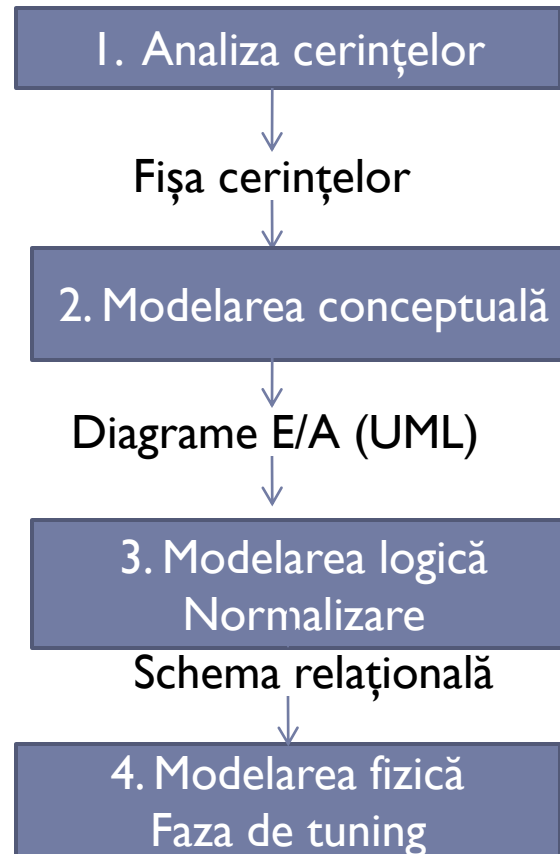
---

- ▶ Concepte E/A de bază
- ▶ Modelarea constrângerilor
- ▶ Capcane de conectare
- ▶ Modelarea E/A înUML
- ▶ De la diagrame E/A (UML) la schema relațională

# Proiectarea unei BD

## Metodologie

---



# Concepte E/A clasice (Chen 1976)

---

## ▶ Entitate

- ▶ Un concept de sine stătător ce corespunde unui grup de obiecte de același tip
- ▶ ○ instanță - entitate
  - ▶ ○ instanță unic identificabilă

## ▶ Asociere (Relationship)

- ▶ Conexiune/asociere între două sau mai multe entități (sau chiar a unei entități cu ea însăși)
- ▶ Gradul asocierii = nr. de entități participante
  - ▶ unare, binare, ternare...

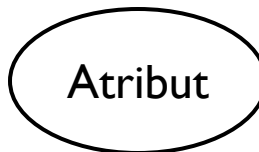
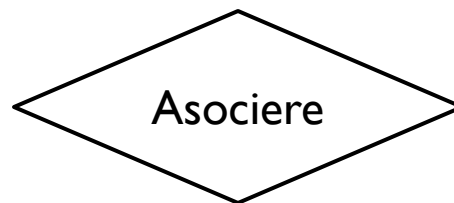
## ▶ Atribut

- ▶ Proprietate a unei entități
- ▶ În asociere
  - ▶ Atribute ale entităților referențiate
  - ▶ Noi atribute

# Diagrame E/A

---

- ▶ Reprezentare grafică a conceptelor E/A
  - ▶ Există mai multe standarde grafice, aici varianta Chen

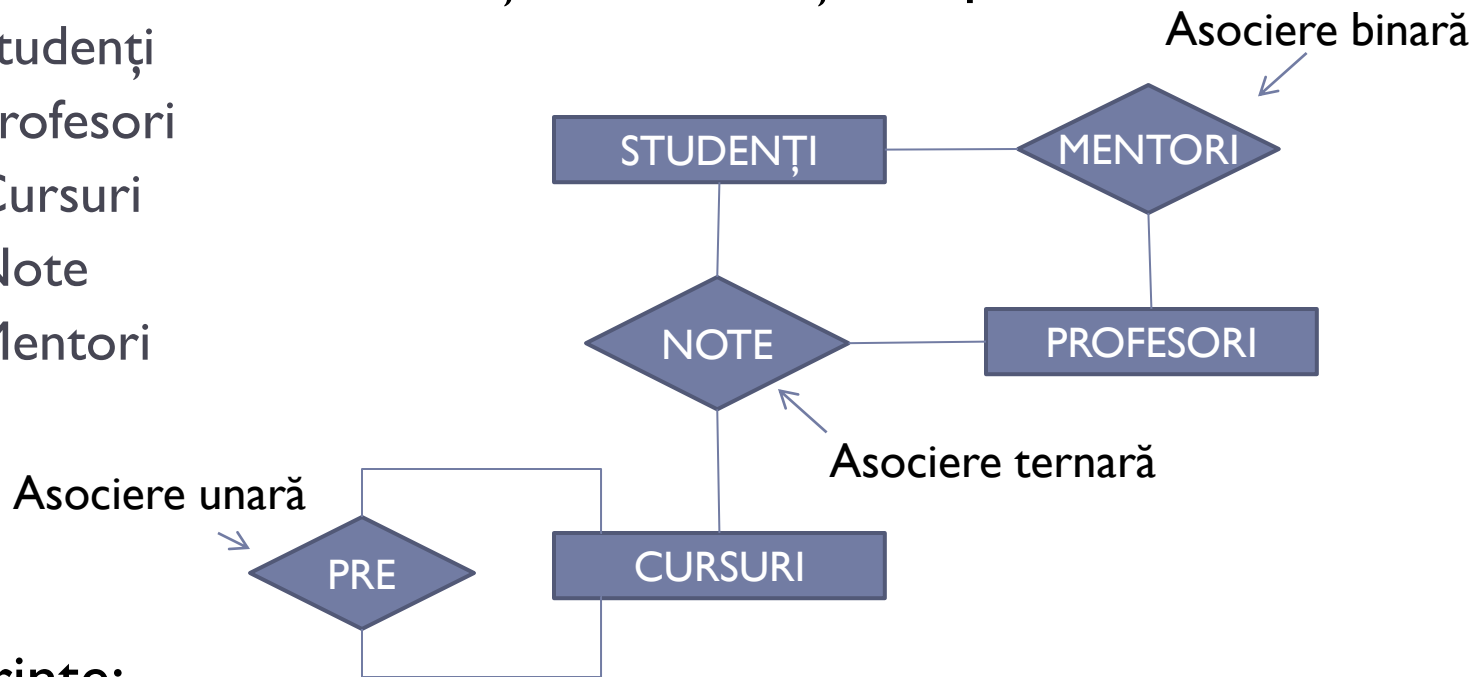


- ▶ Un graf
  - ▶ Entitățile, asocierile și attributele sunt noduri
  - ▶ Există muchii doar între
    - ▶ noduri-entitate și noduri-asociere
    - ▶ noduri-entitate și noduri-atribute
    - ▶ noduri-asociere și noduri-atribute

# Exemplu

## ► O bază de date ce conține informații despre:

- Studenți
- Profesori
- Cursuri
- Note
- Mentori



## ► Cerințe:

- Putem determina notele obținute, cursurile pe care le-a finalizat și creditele obținute de orice student
- Putem determina mentorul oricărui student

# Alte concepte E/R

## ▶ Rol

- ▶ Explică semnificația entităților în asocieri



## ▶ Cheie primară

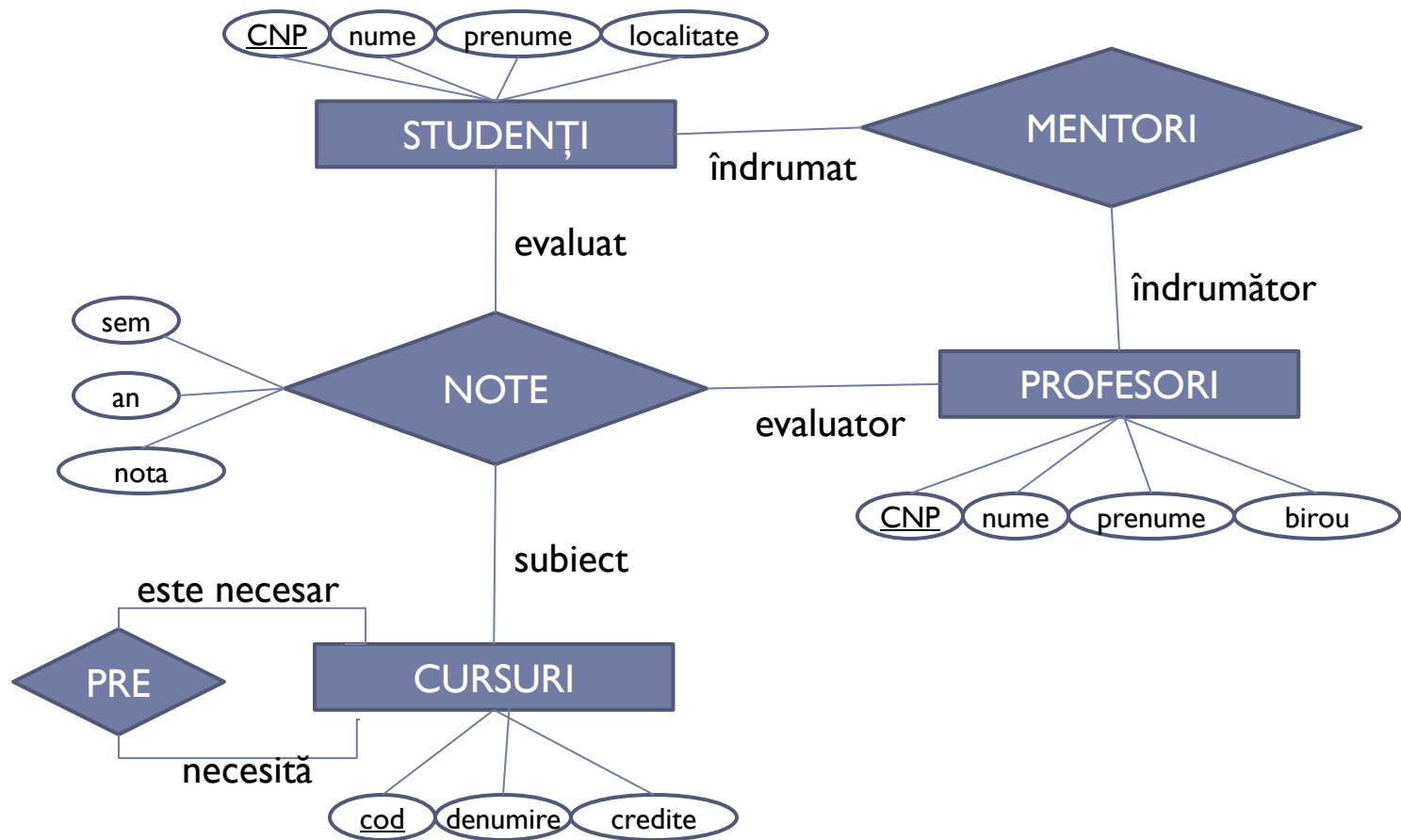
- ▶ Un atribut sau o submulțime minimală de attribute ce identifică unic o instanță a unei entități sau a unei asocieri
- ▶ Obligatorie pentru entități, pentru a indica care instanțe participă în asocieri

Cheie primară

## ▶ Cheie străină pentru o asocierie

- ▶ Un atribut sau o mulțime de attribute care constituie cheie primară pentru entitățile implicate

# Exemplu

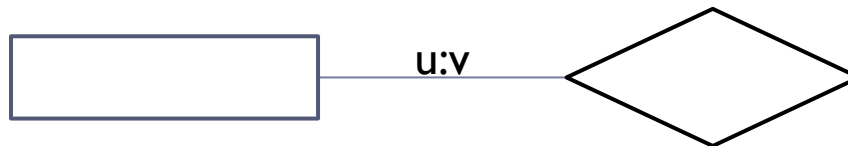


Care sunt cheile străine pentru cele trei asocieri?



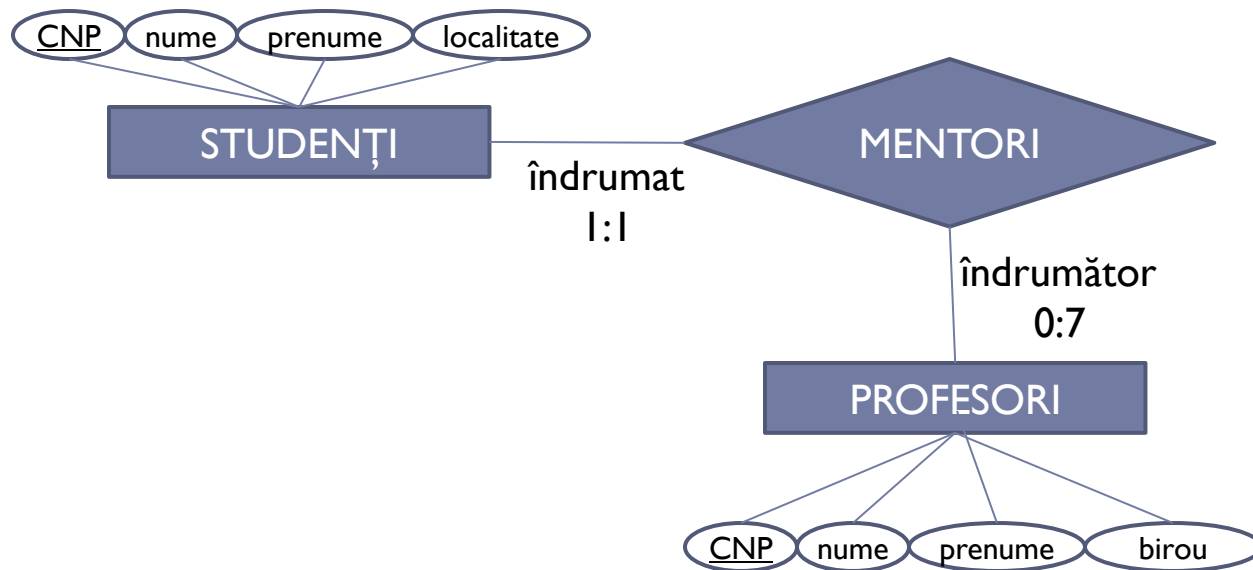
# Constrângeri de conectivitate/multiplicitate

- ▶ Modelul E/A permite declararea de constrângeri asupra numărului de instanțe ale unei asocieri în care o instanță a unei entități participă
- ▶ Fie  $R$  o asociere între  $n$  entități  $E_i$ ,  $i=1..n$ . Baza de date satisface constrângerea  $(E_i, u,v,R)$  dacă fiecare instanță a entității  $E_i$  participă în cel puțin  $u$  și cel mult  $v$  instanțe ale asocierii  $R$ .

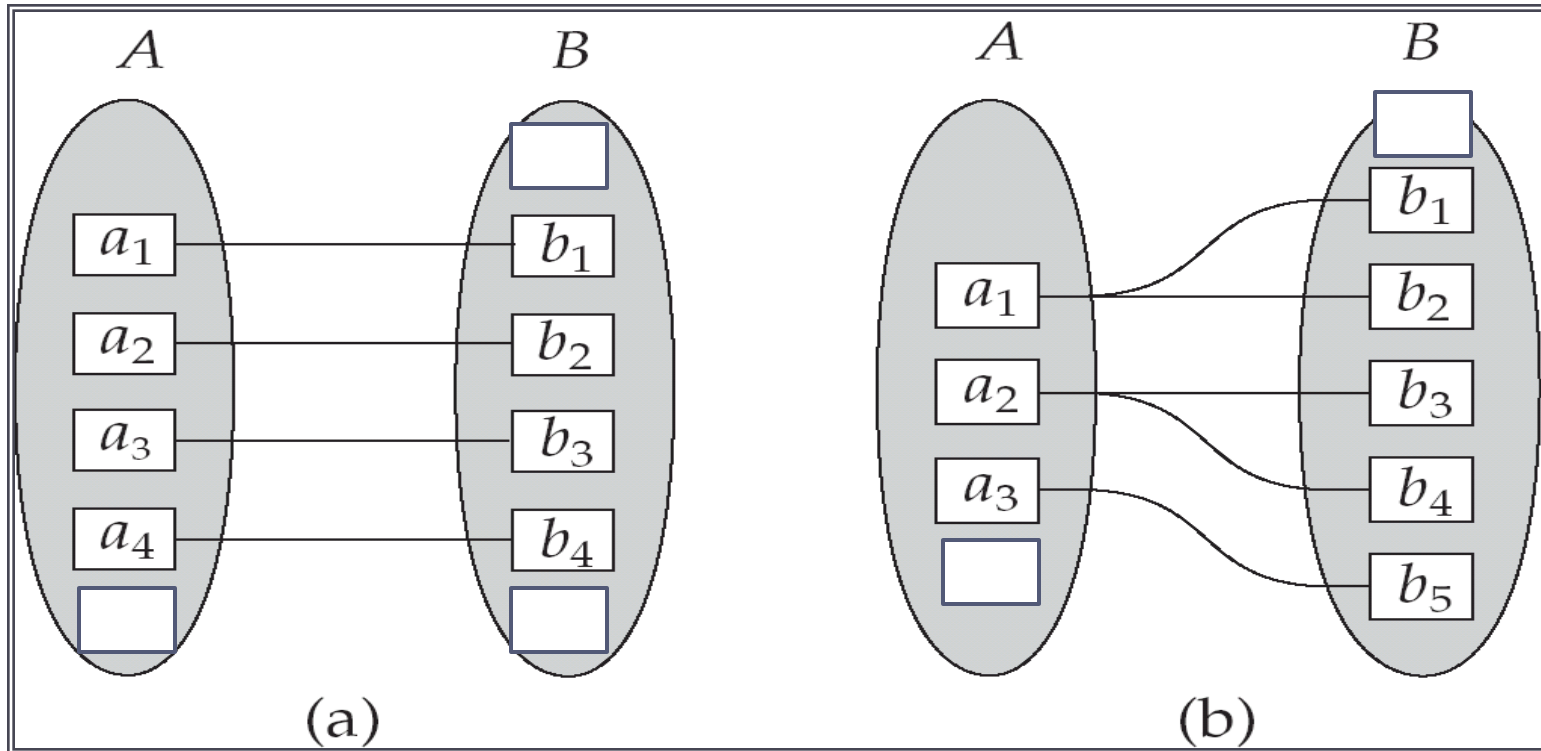


# Exemplu

- ▶ (Studenti, 1, 1 Mentori)
- ▶ (Profesori, 0, 7, Mentori)
- ▶ Fiecare student are un singur profesor drept mentor iar un profesor poate fi mentor pentru cel mult 7 studenti



# Constrângeri de conectivitate pentru asocieri binare (1)



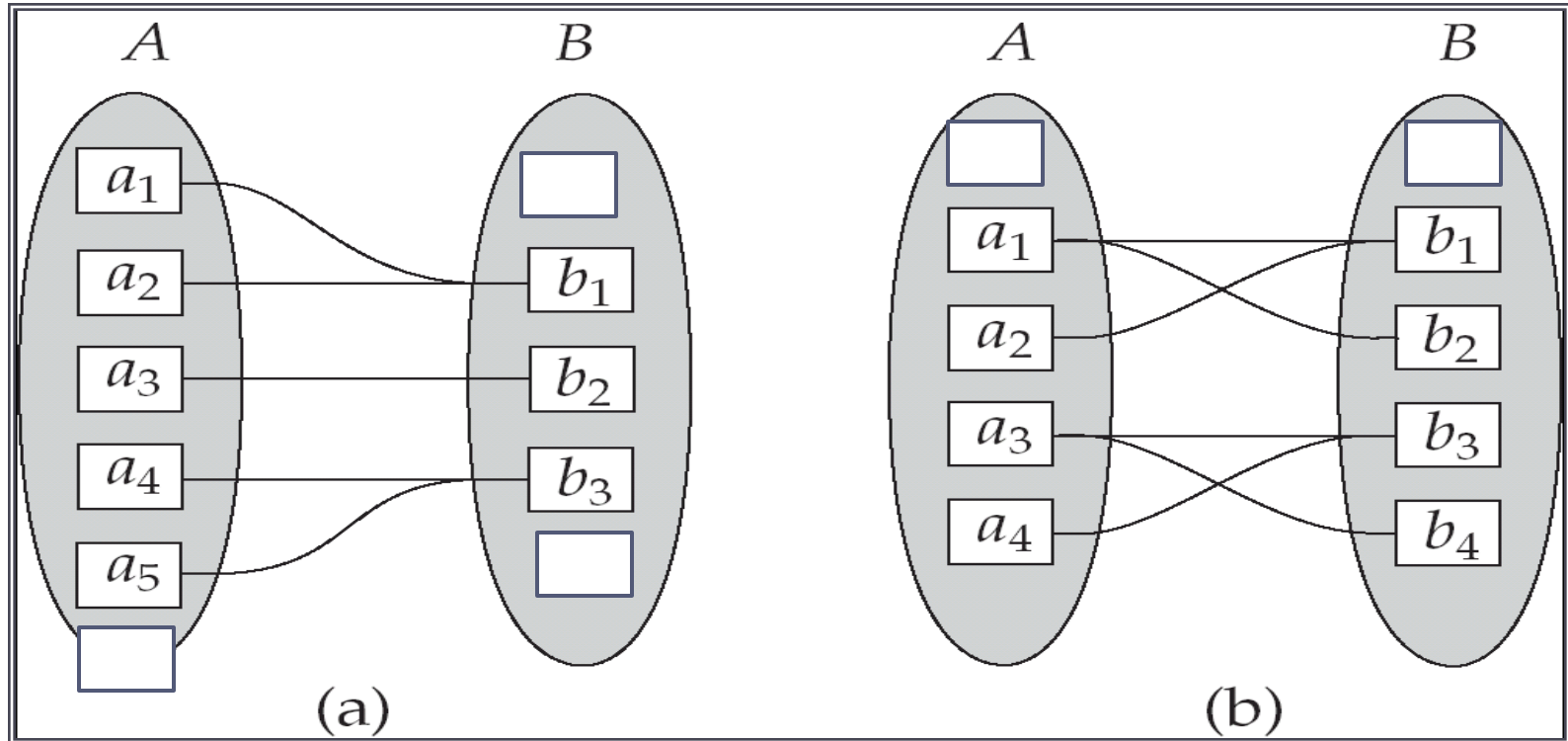
a) Asocierie unu la unu  
(A,0,1,R) (B,0,1,R)



b) Asocierie unu la mulți  
(A,0,n,R) (B,0,1,R),  $n > 1$



# Constrângeri de conectivitate pentru asocieri binare (2)



a) Asocierie mulți la unu  
 $(A, 0, 1, R) (B, 0, n, R)$

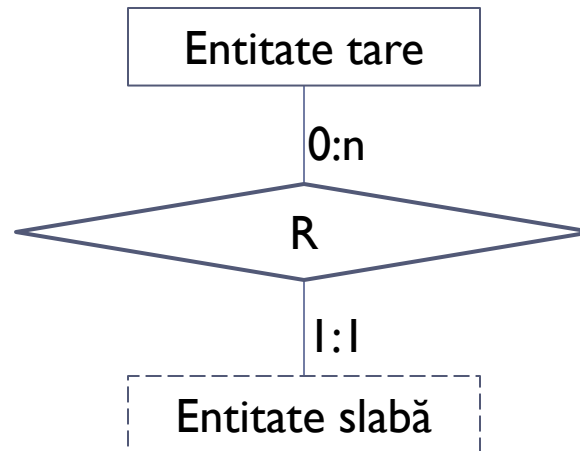


b) Asocierie mulți la mulți  
 $(A, 0, m, R) (B, 0, n, R), m, n > 1$



# Entitate slabă

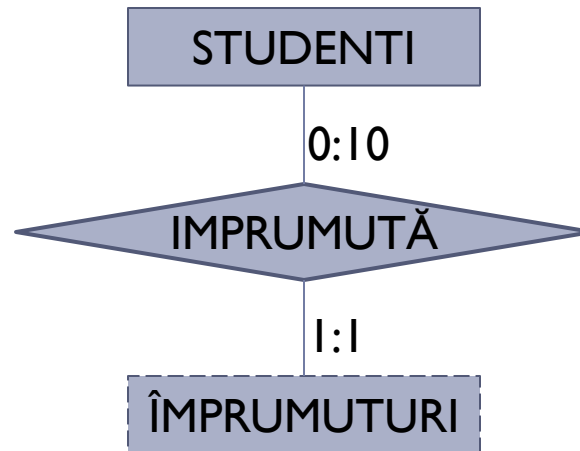
- ▶ O entitate este slabă dacă existența instanțelor sale depinde de existența instanțelor altei entități (dependență existențială)



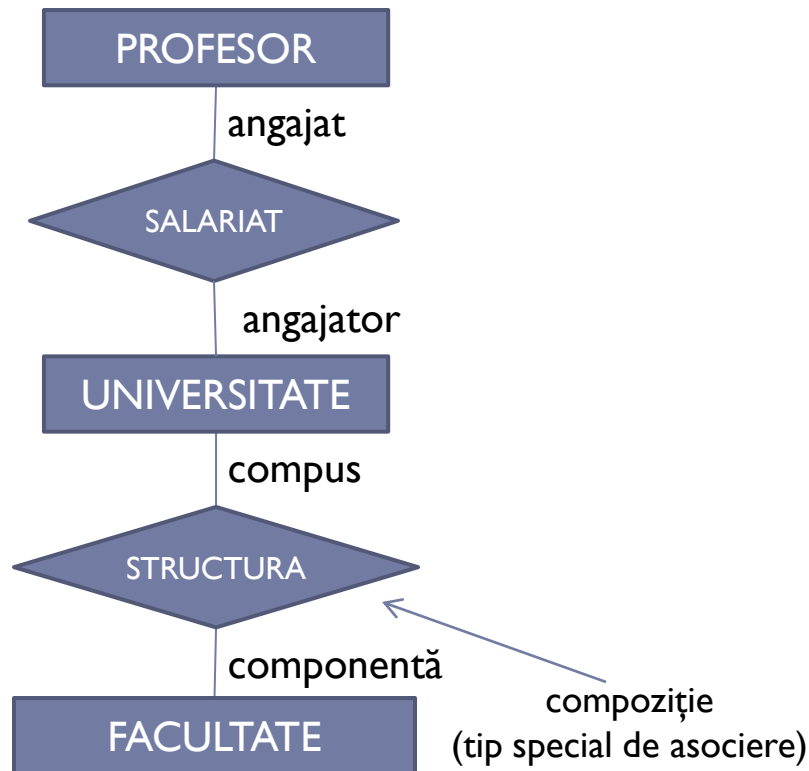
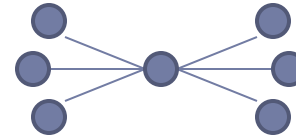
- ▶ Nu are cheie
- ▶ Satisface constrângerea de conectivitate (Entitate\_slaba, 1, 1, R), deci participă într-o asocierie de tip unu la mulți relativ la entitatea tare

# Exemplu

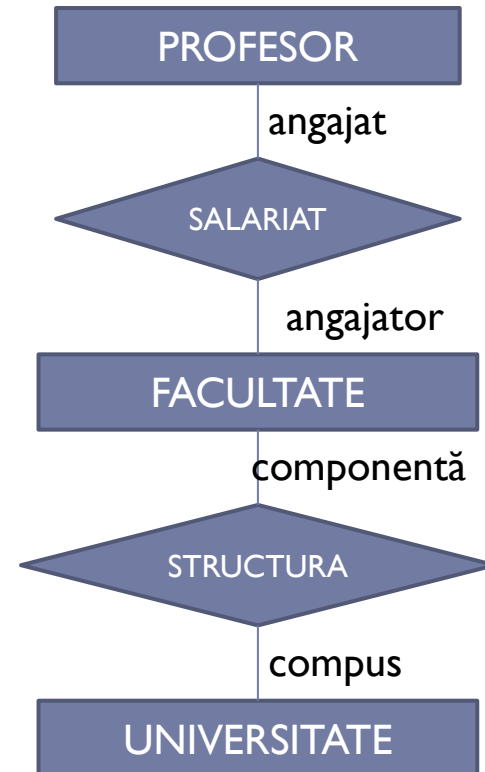
---



# Capcane de conectare (Fan traps)

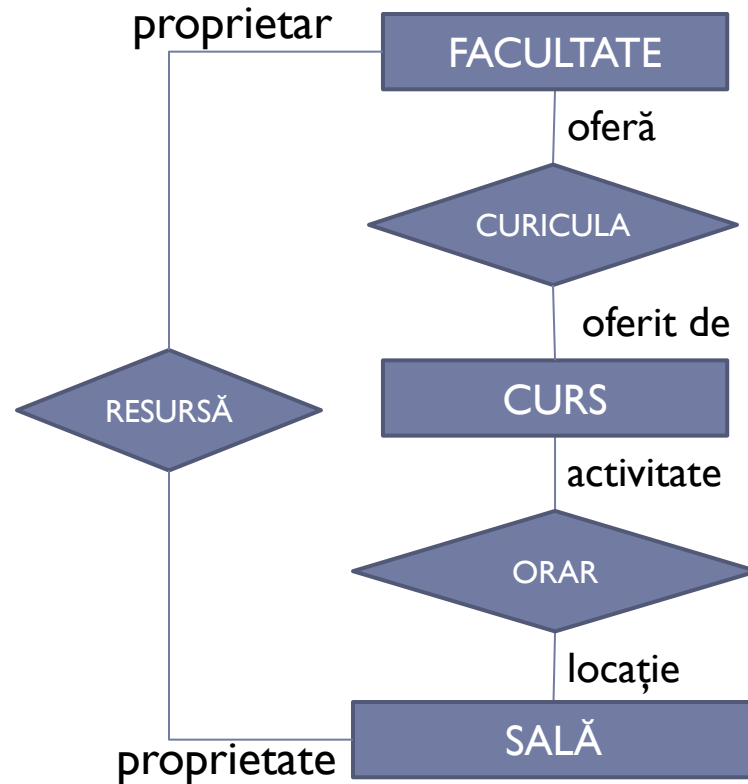
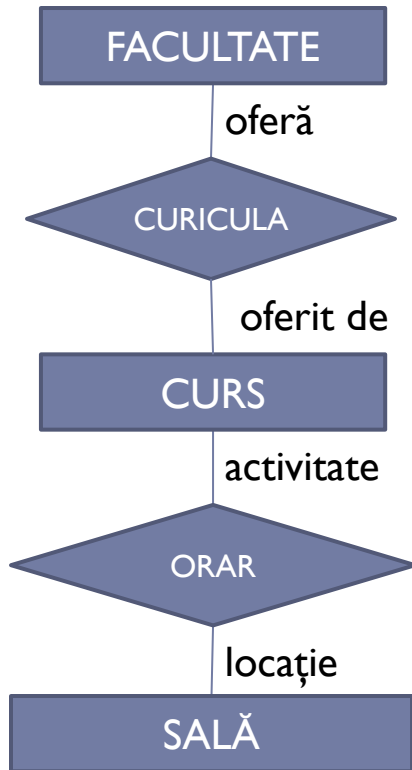
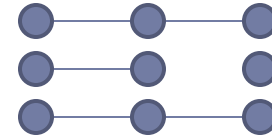


**Problema:**  
La ce departament aparține profesorul X?



**Soluția:**  
Model restructurat

# Capcane de conectare (Chasm traps)



**Problema:**  
Care sunt toate sălile ce aparțin unei facultăți?

**Soluția:**  
Noi asocieri

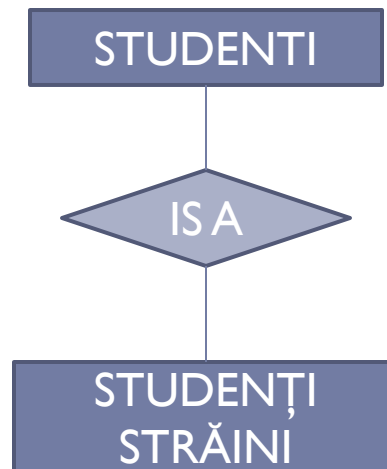


# Modelul E/A extins

## Specializare

---

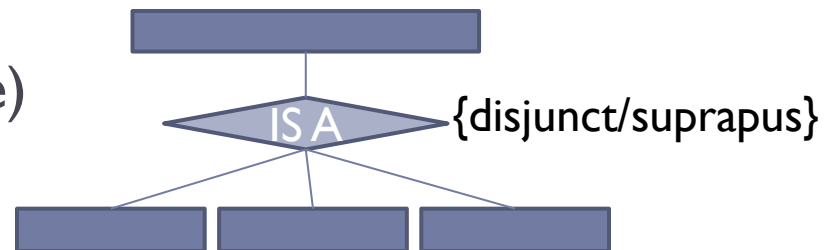
- ▶ Subgrupuri distinctive de instanțe ale unei entități
  - ▶ Au în plus anumite atribute
  - ▶ Participă în asocieri la care nu participă toate instanțele entității
  - ▶ Corespund unei entități specializate care se află într-o asociere de tip IS-A cu entitatea de bază



# Constrângeri specifice specializării

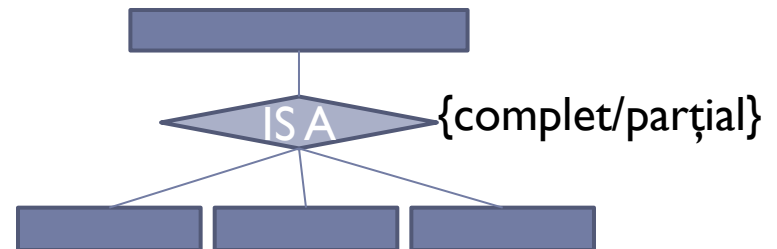
- ▶ Instanțele specializării moștenesc toate atributele și asocierile entității de bază, inclusiv cheia
- ▶ O instanță a unei entități poate aparține la una sau la mai multe specializări

- ▶ Specializări disjuncte (exclusive)
- ▶ Specializări cu suprapunere



- ▶ O instanță a unei entități trebuie sau nu să aparțină la cel puțin o specializare

- ▶ Complet
- ▶ Incomplet (parțial)



# Modelare UML

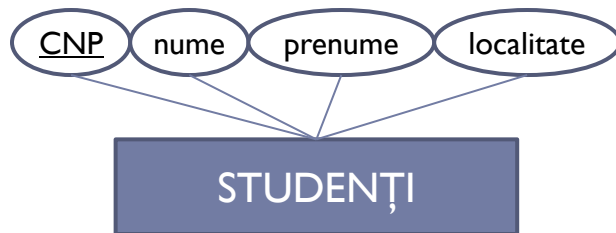
- ▶ **Unified Modeling Language**
  - ▶ Utilizat în ingineria software
  - ▶ Bazat pe concepte orientate obiect
  - ▶ Unealtă de comunicare cu clientul în termenii utilizați în companie
  - ▶ Un limbaj foarte mare, utilizăm un set restrâns de elemente (diagrama de clase) pentru a modela o bază de date.

# Mapare E/R – UML

E/R	UML
Entitate + atribut	Clasă
Asociere fără atribut proprii	Asociere
Asociere cu atribut proprii	Clasă de asociere
Specializare	Subclasă
	Compoziție și agregare

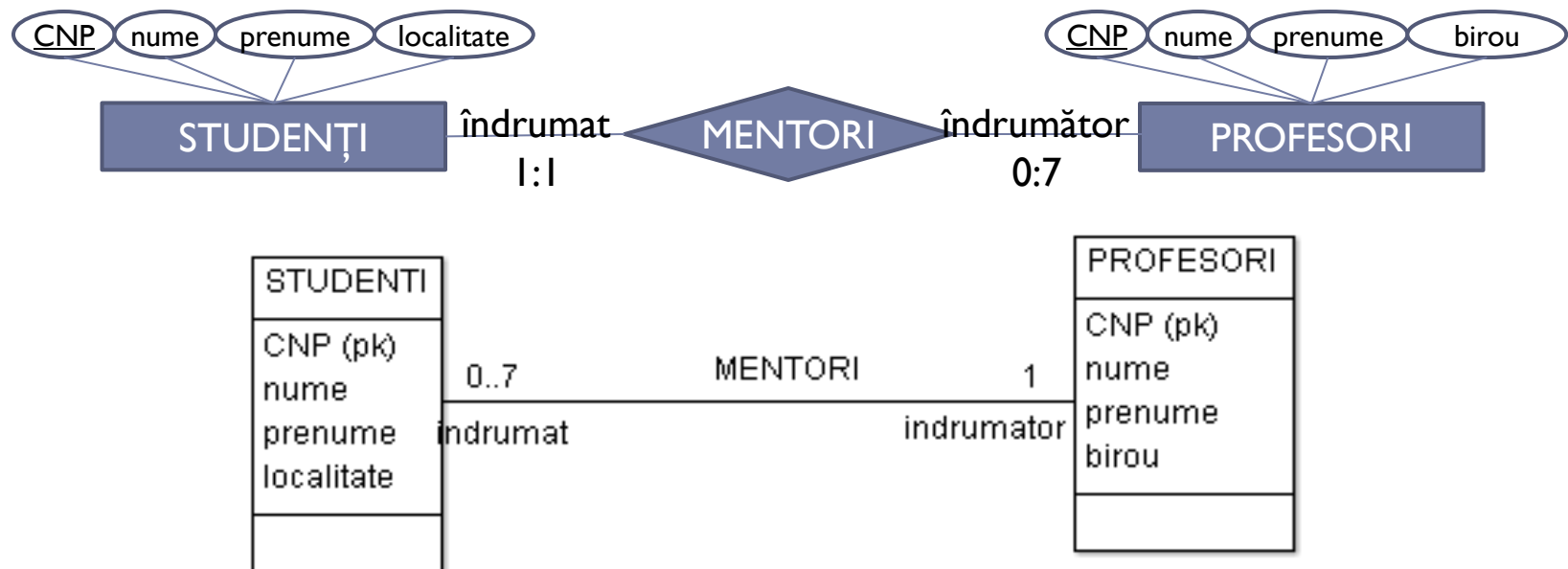
# Clase

- ▶ Componente: nume, attribute, metode
- ▶ BD: nume, attribute (cheia primară)



# Asocieri

- ▶ Exprimă asocierea dintre obiectele aparținând la **2** clase
- ▶ BD: asocierea dintre instanțele a două entități



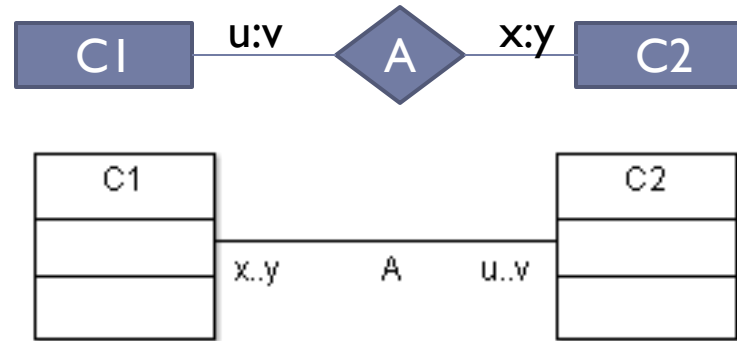
- ▶ Obs: constrângerile de cardinalitate se specifică invers decât în diagramele E/A

# Asocieri

## Constrângeri de conectivitate/multiplicitate

### ► Restricții

- $(C1, u, v, A)$
- $(C2, x, y, A)$



- Fiecare obiect din (instanță a entității) C1 este asociat cu cel puțin u și cel mult v obiecte din (instanțe ale entității) C2
- Fiecare obiect din (instanță a entității) C2 este asociat cu cel puțin x și cel mult y obiecte din (instanțe ale entității) C2

<b>x..y</b>	<b>u..v</b>	<b>Tip asociere</b>
0..1	0..1	unu la unu incompletă
1..1 (1)	1..1 (1)	unu la unu completă (implicită)
0..1	0..* (*)	unu la multi incompletă

...

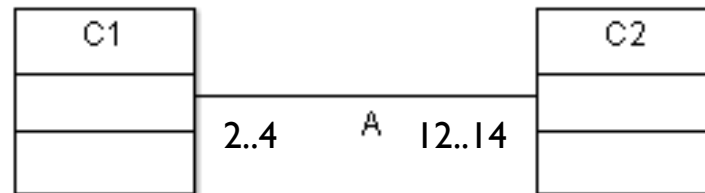
...

...

???

- ▶ Modelați asocierea dintre STUDENTI și UNIVERSITĂȚI. Un student poate aplica la cel mult 5 universități și e necesar să aplice la cel puțin una. O universitate primește cel mult 10.000 aplicații.

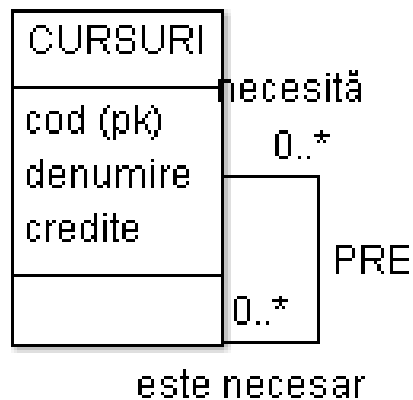
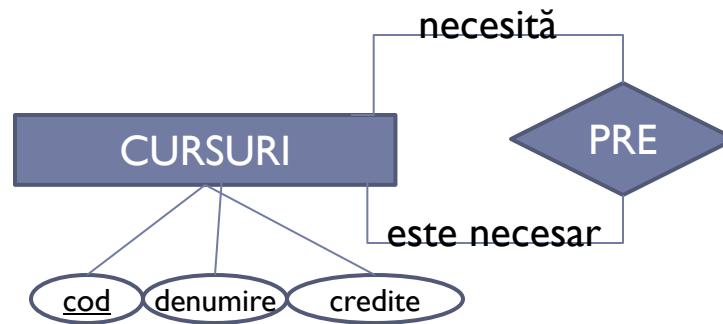
- ▶ Fie asocierea



Care e numărul minim de instanțe pentru entitatea C1 și pentru C2?

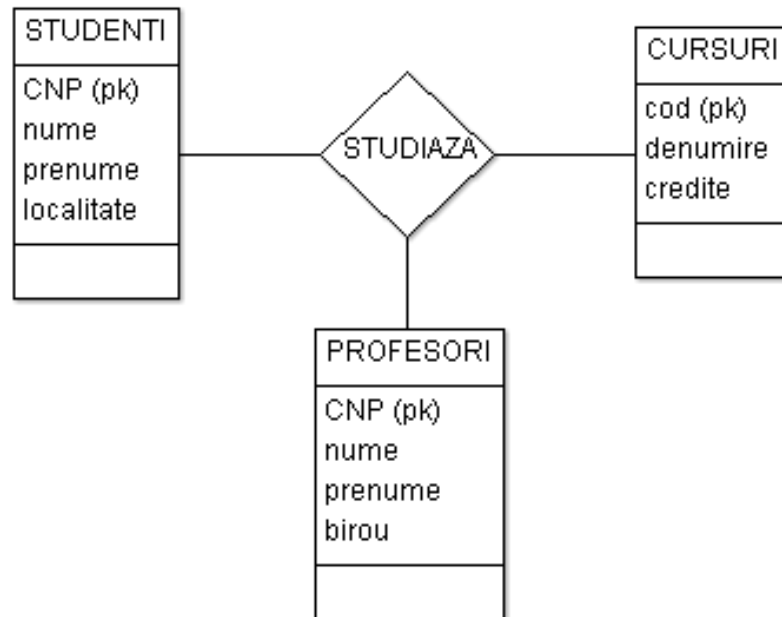


# Asocieri recursive

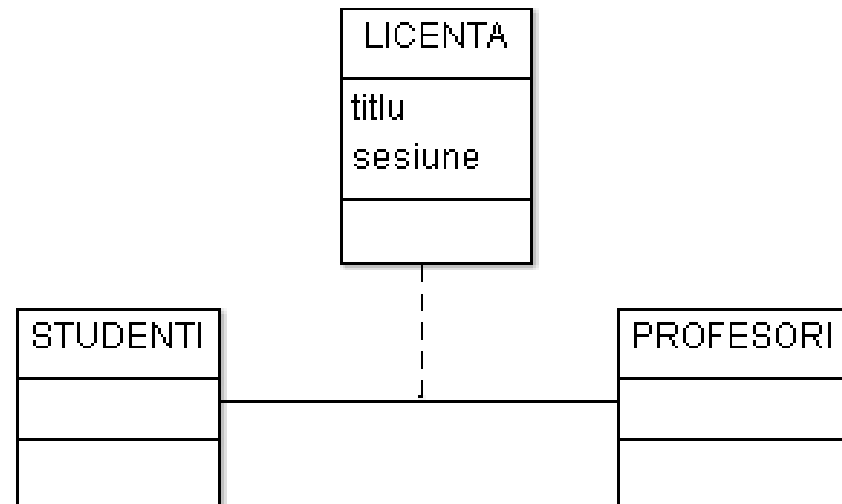
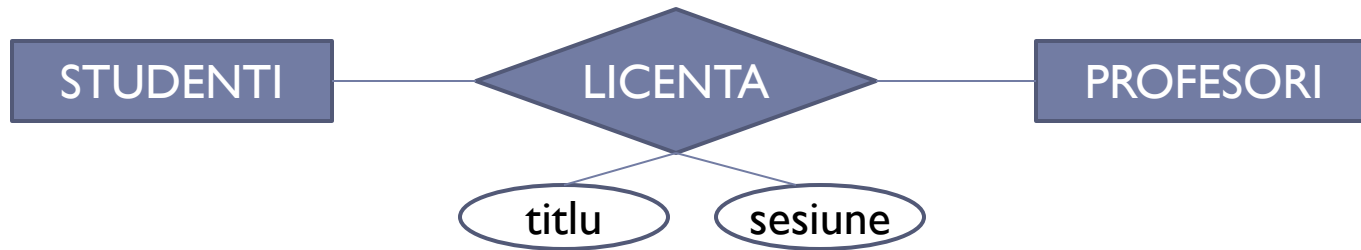


# Asocieri n-are

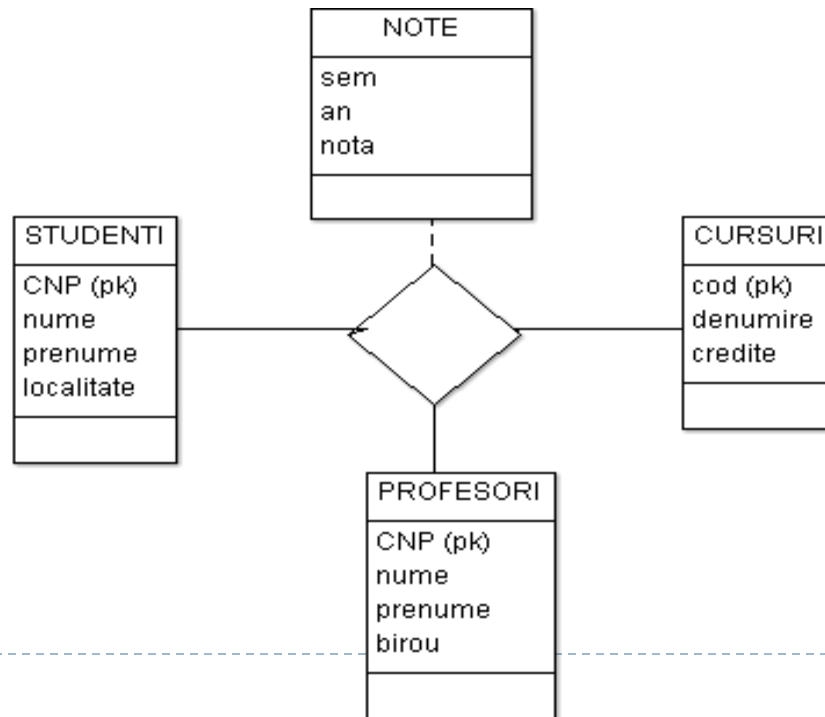
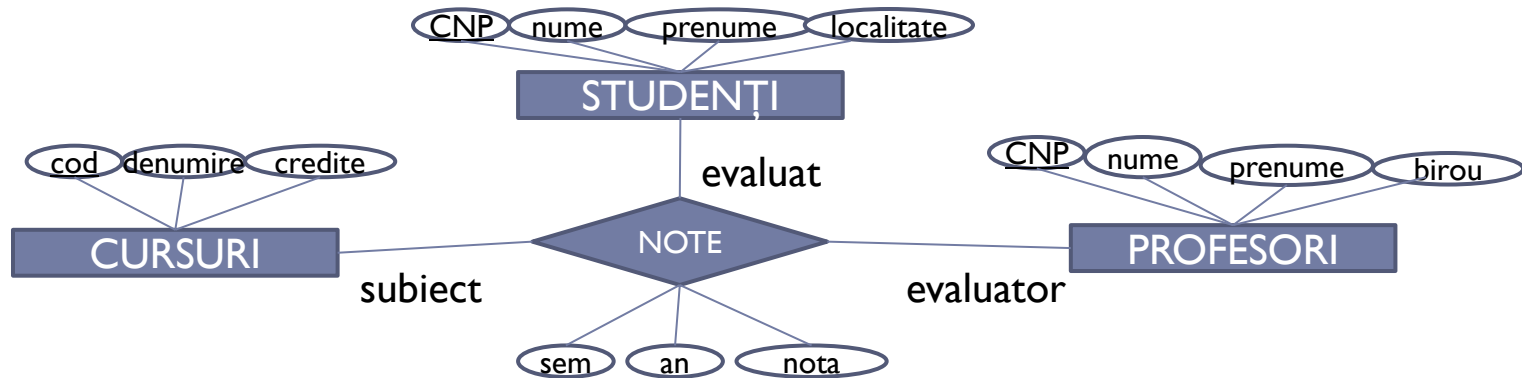
---



# Clase de asociere

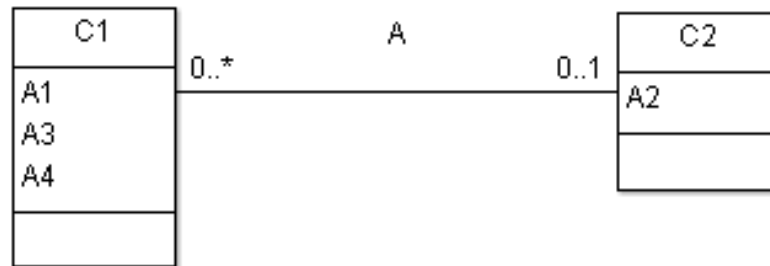
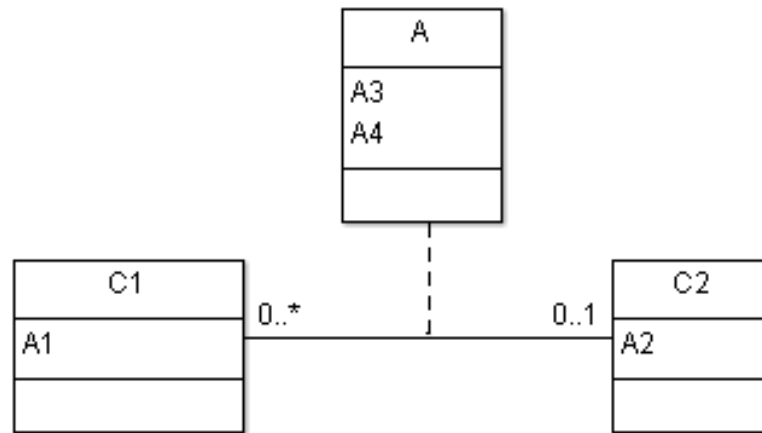


# Clase de asociere

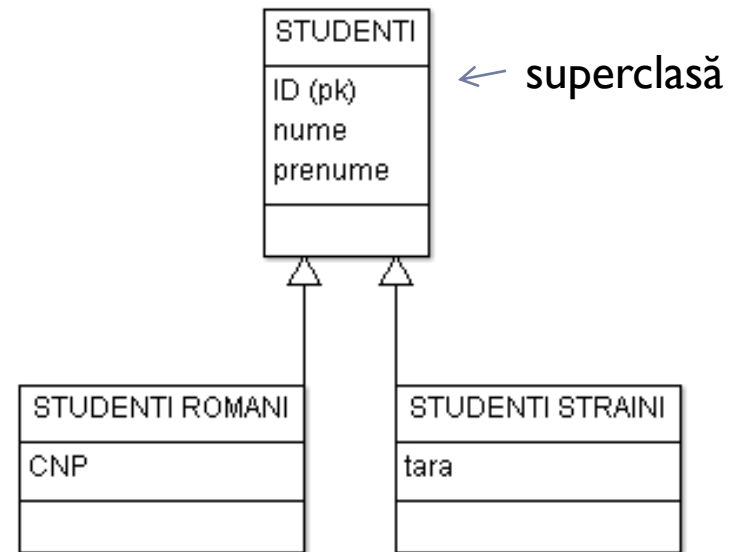
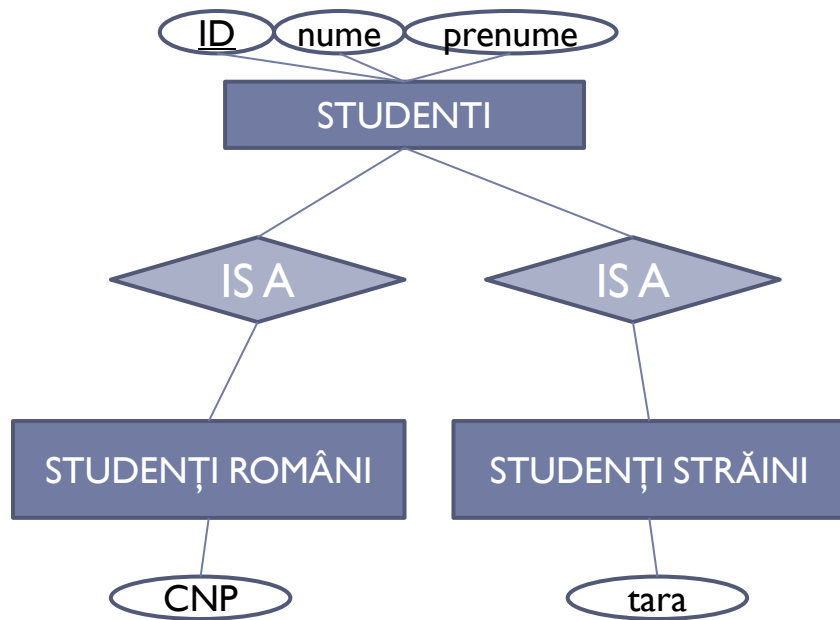


# Eliminarea claselor de asociere

- Atunci când avem multiplicitate 0..1 sau 1..1

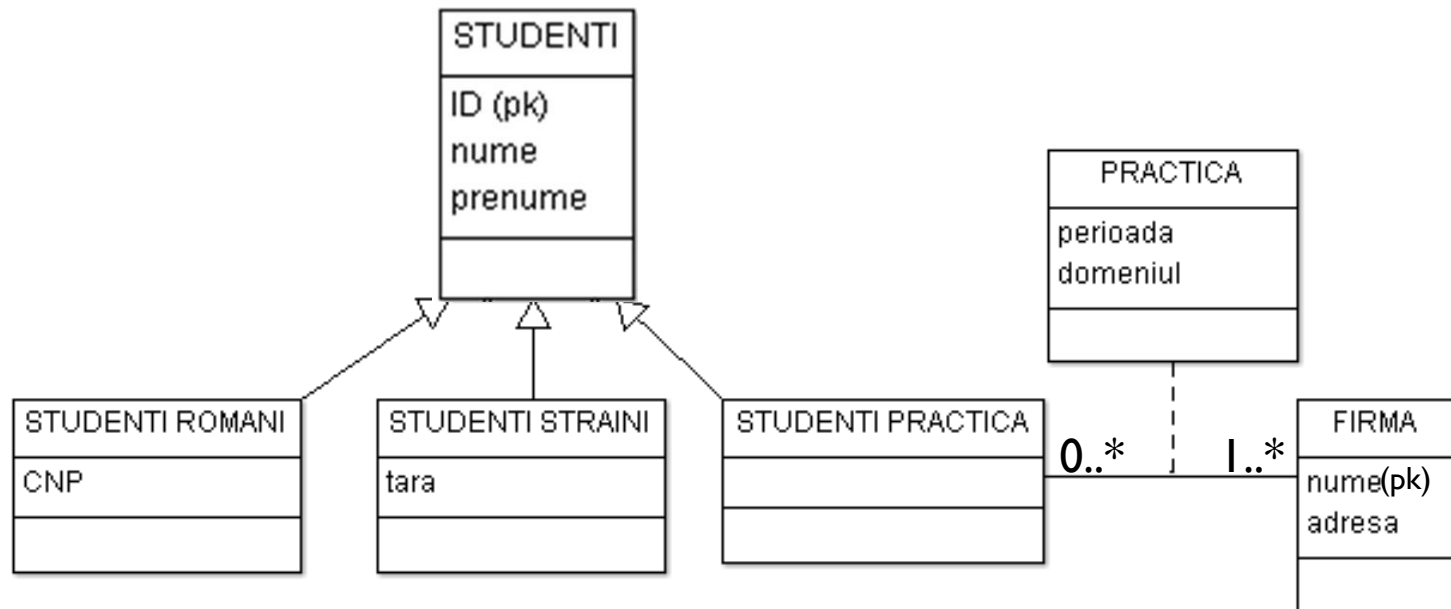


# Subclasă (1)



Specializare completă, disjunctă

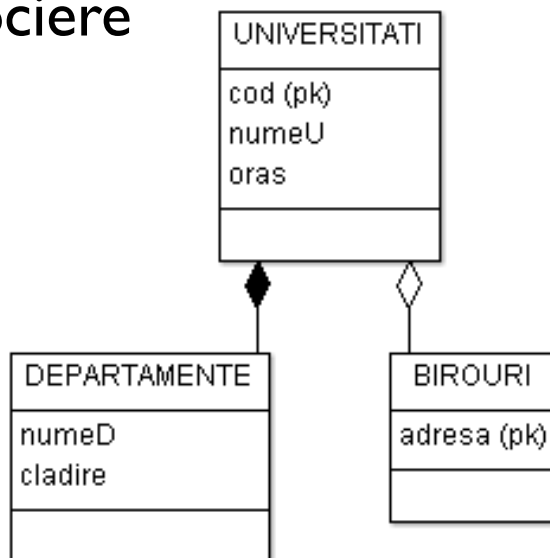
# Subclasă (2)



Specializare completă, cu suprapunere

# Compoziție și agregare

- ▶ Obiecte dintr-o clasă aparțin obiectelor din altă clasă
- ▶ Tipuri speciale de asociere



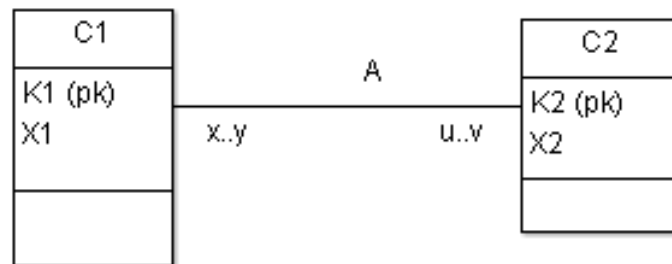
- ▶ Compoziția: **toate** obiectele unei clase *părți* aparțin obiectelor dintr-o clasă *compusă*; clasei *părți* îi corespunde de obicei o entitate slabă (multiplicitate 1..1; fără cheie primară);
- ▶ Agregarea: **unele** obiecte dintr-o clasă aparțin obiectelor din altă clasă (multiplicitate 0..1)



# Mapare E/A, UML -> schema BD relațională

E/A	UML	Schema relațională
Entitate + atribut	Clasă	Relație cu cheie primară
Asociere fără atribut proprii	Asociere	Relație cu chei străine
Asociere cu atribut proprii	Clasă de asociere	Relație cu chei străine și alte atribute
Specializare	Subclasă	Relație cu cheie primară (cea a superclasei) și atribute particulare/specializate
	Compoziție și agregare	Relație cu cheie străină și atribute particulare

# Entități/clase și asocieri



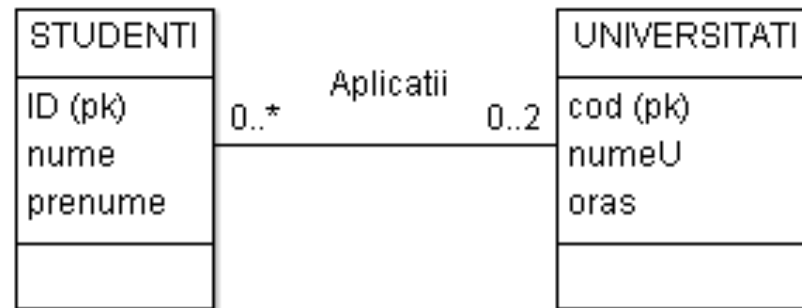
$\{C1(\underline{K1}, X1), C2(\underline{K2}, X2), A(K1, K2)\}$

## ► Cheia primară pentru asociere depinde de multiplicitate

x..y	u..v	Cheia primară pt A	Observații
0..1 1..1	*	K2	Nu e necesară relația A $\{C1(\underline{K1}, X1), C2(\underline{K2}, X2, K1)\}$
*	0..1 1..1	K1	Nu e necesară relația A $\{C1(\underline{K1}, X1, K2), C2(\underline{K2}, X2)\}$
*	*	(K1, K2)	

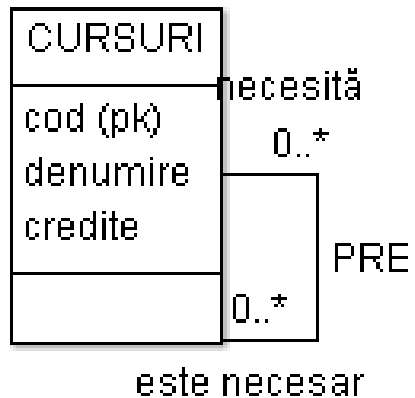
???

► Fie diagrama

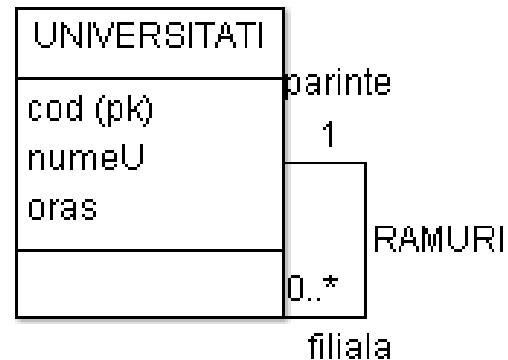


► Mai este posibilă renunțarea la relația corespunzătoare asocierii?

# Asocieri recursive



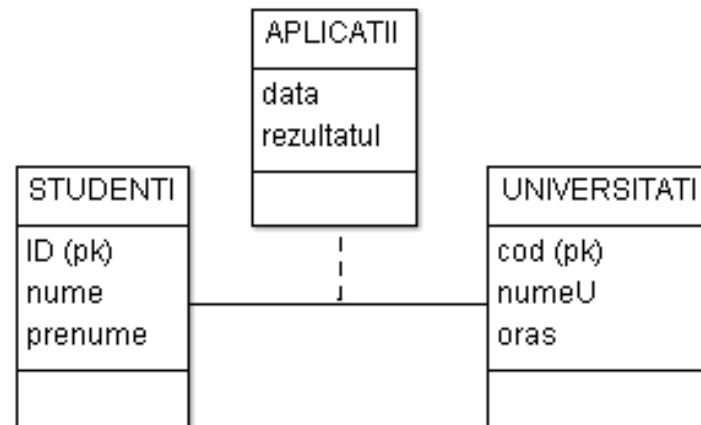
{CURSURI (cod, denumire, credite)  
PRE (cod1, cod2)}



{UNIVERSITATI (cod, numeU, oras)  
RAMURI (codFiliala, codParinte)}

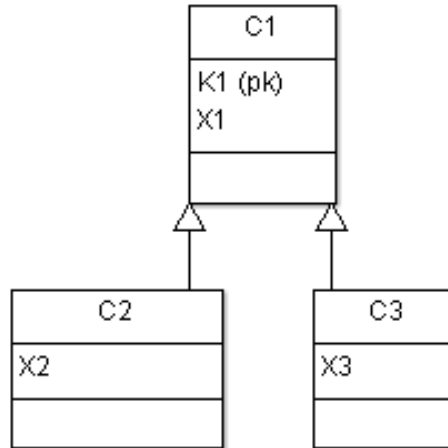
# Clase de asociere

---



{STUDENTI (ID, nume, prenume)  
UNIVERSITATI (cod, numeU, oras)  
APLICATII (ID, cod, data, rezultatul)}

# Specializare / Subclase



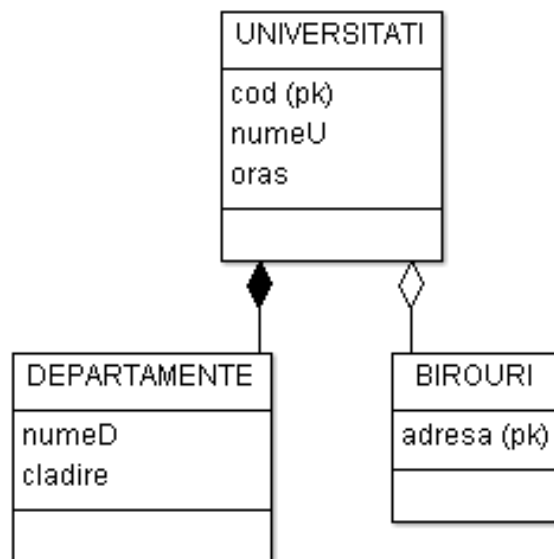
## ► Posibilități

- Relații subclasă ce conțin cheia superclasei și attributele specializate  
 $C1(\underline{K1}, X1), C2(\underline{K1}, X2), C3(\underline{K1}, X3)$
- Relații subclasă ce conțin attributele superclasei (inclusiv atributul cheie) și attributele specializate; superclasa conține doar ule nespecializate  
 $C1(\underline{K1}, X1), C2(\underline{K2}, X1, X2), C3(\underline{K2}, X1, X3)$
- O singură relație ce conține attributele din superclasă și subclasă  
 $C(\underline{K1}, X1, X2, X3)$

???

- 
- Fie superclasa  $S$  cu un număr de subclase. Considerăm că relația de specializare este incompletă și cu suprapunere. Dacă  $n_1$ ,  $n_2$  și  $n_3$  reprezintă numărul total de tuple necesare fiecărei scheme de decodificare din cele 3 date anterior (în ordinea dată), care este relația dintre cele 3 valori?
- $n_1 < n_2 < n_3$
  - $n_1 \leq n_2 \leq n_3$
  - $n_3 < n_2 < n_1$
  - $n_3 \leq n_2 \leq n_1$

# Compoziție și agregare



{ UNIVERSITATI(cod, numeU, oras)  
DEPARTAMENTE(codU, numeD, cladire)  
BIROURI (codU, adresa)}

← NU acceptă NULL  
← acceptă NULL



# Modelare EA/UML

## Sumar

---

### ► PROS

- Tehnică populară de modelare conceptuală
- Construcții expresive, descriu punctul de vedere personal asupra aplicației
- Permite exprimarea unor tipuri de constrângeri (chei primare, străine, multiplicitate, exclusivitate...)

### ► CONS

- Tehnică subiectivă (entitate sau atribut, entitate sau asociere, subclasare sau nu, compoziție sau nu)
- Nu permite modelarea tuturor dependentelor
- Necesită utilizarea ulterioară a normalizării

# Bibliografie

---

- ▶ Hector Garcia-Molina, Jeff Ullman, Jennifer Widom: *Database Systems: The Complete Book (2nd edition)*, Prentice Hall; (June 15, 2008)
- ▶ Thomas Connolly, Carolyn Begg: *Database Systems: A Practical Approach to Design, Implementation and Management*, (5<sup>th</sup> edition) Addison Wesley, 2009
  
- ▶ Unelte:
  - ▶ <https://createy.com> (diagrame EA, diagrame UML de clasă)
  - ▶ <http://diagramo.com/> (diagrame EA)
  - ▶ <http://argouml-downloads.tigris.org/nonav/argouml-0.32.2/ArgoUML-0.32.2.zip> (diagrame UML de clasă)