

# Modelul relational

- Tabele, randuri, coloane
- SGBDR – Regulile lui Codd

## Proiectarea bazelor de date relationale

- Crearea schemei conceptuale
- Crearea design-ului logic al bazei de date
- Normalizarea bazei de date
  - Prima forma normala (1NF – First Normal Form)
  - A doua forma normala (2NF – Second Normal Form)
  - A treia forma normala (3NF – Third Normal Form)
  - Forma normala Boyce-Codd (BCNF – Boyce-Codd Normal Form)
  - A patra forma normala (4NF – Fourth Normal Form)

# Modelul relational

## Tabele, randuri, coloane

Relatii	↔	Tabele
Atribute	↔	Coloane
Tuple	↔	Inregistrari

## Regulile lui Codd

1. **Regula reprezentarii logice a datelor:** Intr-o baza de date relationala, toate datele sunt reprezentate la nivel logic intr-un singur mod, si anume sub forma de valori logice in tabele.
2. **Regula accesului la date:** Toate datele individuale din tabele trebuie sa fie accesibile prin furnizarea numelui tabelului, numelui coloanei si valorii cheii primare.
3. **Regula reprezentarii valorilor necunoscute:** Un sistem relational trebuie sa permita declararea si manipularea sistematica a valorilor Null, cu semnificatia unor valori necunoscute sau inaplicabile.

# Modelul relational

## Regulile lui Codd (continuare)

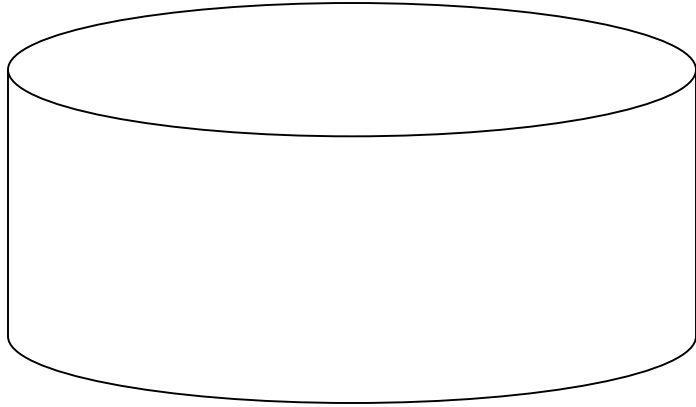
- 4. Regula dictionarului de date:** Descrierea bazei de date (dictionarul de date) trebuie sa fie reprezentata la nivelul logic tot sub forma de tabele, astfel incat asupra acesteia sa se poata aplica aceleasi operatii ca si asupra datelor propriu-zise.
- 5. Regula limbajului de acces:** Intr-un sistem relational trebuie sa existe cel putin un limbaj de accesare a datelor, care sa asigure urmatoarele operatii: definirea tabelor de baza si a tabelor virtuale (view-uri, vederi); manipularea si interogarea datelor (atat interactiv cat si prin program); definirea restrictiilor de integritate, autorizarea accesului la date, delimitarea tranzactiilor.
- 6. Regula de actualizare a tabelor virtuale:** Un SGBD trebuie sa poata determina daca o vedere poate sa fie actualizata sau nu.
- 7. Regula manipularii datelor:** Un sistem relational trebuie sa ofere posibilitatea procesarii tabelor nu numai in operatiile de interogare a datelor cat si in cele de inserare, actualizare si stergere.
- 8. Regula independentei fizice a datelor:** Aplicatiile (la nivel logic) nu trebuie sa depinda de modul de stocare si accesare fizica a datelor.

# Modelul relational

## Regulile lui Codd (continuare)

9. **Regula independenteii logice a datelor:** Programele de aplicatie nu trebuie sa fie afectate de nici o restructurare logica a tabelelor bazei de date care conserva datele.
10. **Regula independenteii datelor din punctul de vedere al integritatii:** Regulile de integritate a bazei de date trebuie sa fie definite in limbajul utilizat de sistem pentru definirea datelor si nu in cadrul aplicatiilor individuale; in plus, aceste reguli de integritate trebuie stocate in dictionarul de date.
11. **Regula independenteii datelor din punctul de vedere al distribuirii:** Programele de aplicatie nu trebuie sa fie afectate de distribuirea pe mai multe calculatoare a bazei de date.
12. **Regula privind prelucrarea datelor de catre un limbaj de nivel inferior:** Dacă sistemul de gestiunea bazelor de date pune la dispozitia utilizatorilor o interfață de nivel inferior (la nivel de înregistrare) atunci interfața nu poate să submineze sistemul, de exemplu, prin nerespectarea unei constrângeri de integritate sau a unei limitări impuse prin politica de securitate.
0. **Regula de baza:** Un SGBD relational trebuie sa fie capabil sa gestioneze baza de date exclusiv pe baza caracteristicilor sale relationale.

# Modelul relational



## Proiectarea bazelor de date relaționale

- Crearea schemei conceptuale
- Crearea design-ului logic al bazei de date
- Normalizarea bazei de date

# Proiectarea bazelor de date relationale

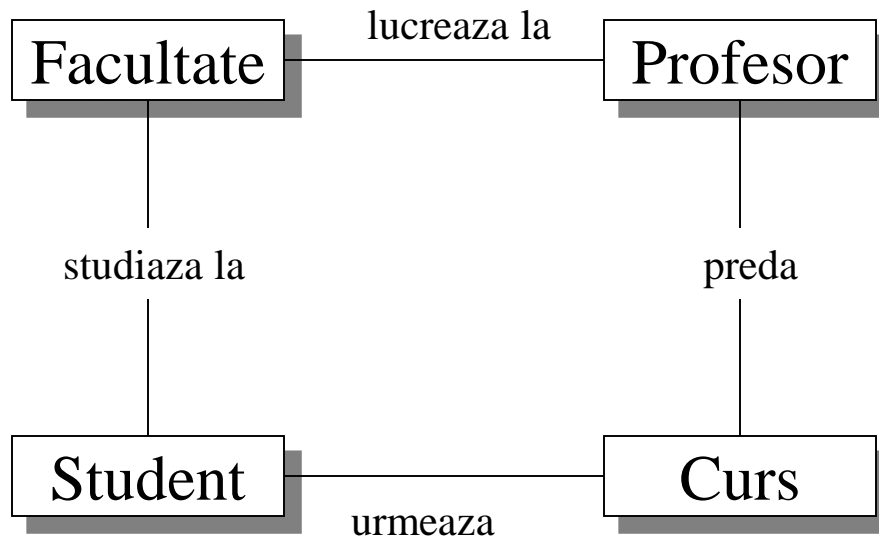
## Crearea schemei conceptuale a bazei de date

Modelul entitate-relatie

### Entitate

Obiect de interes pentru care trebuie sa existe date inregistrate

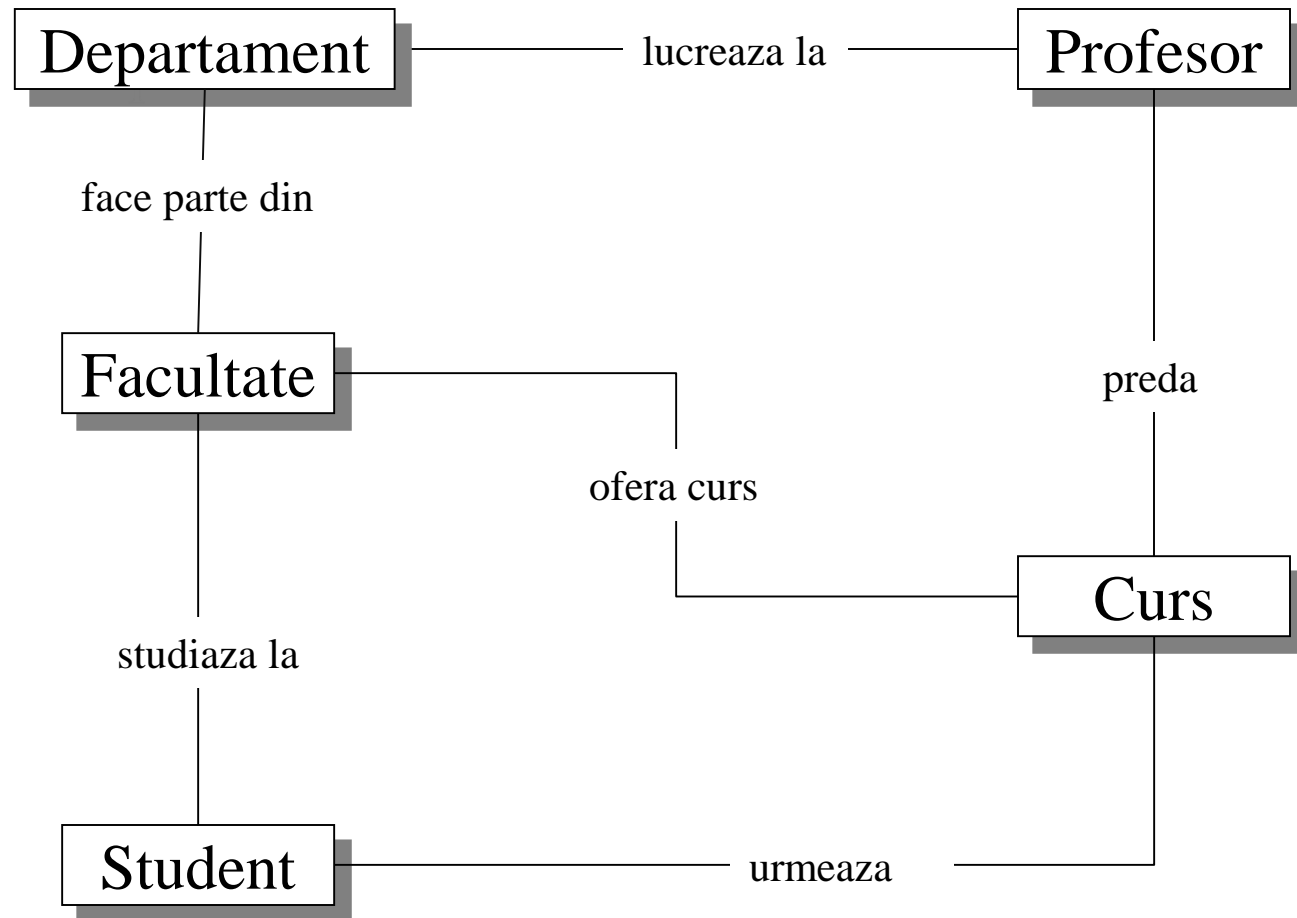
Exemplu:



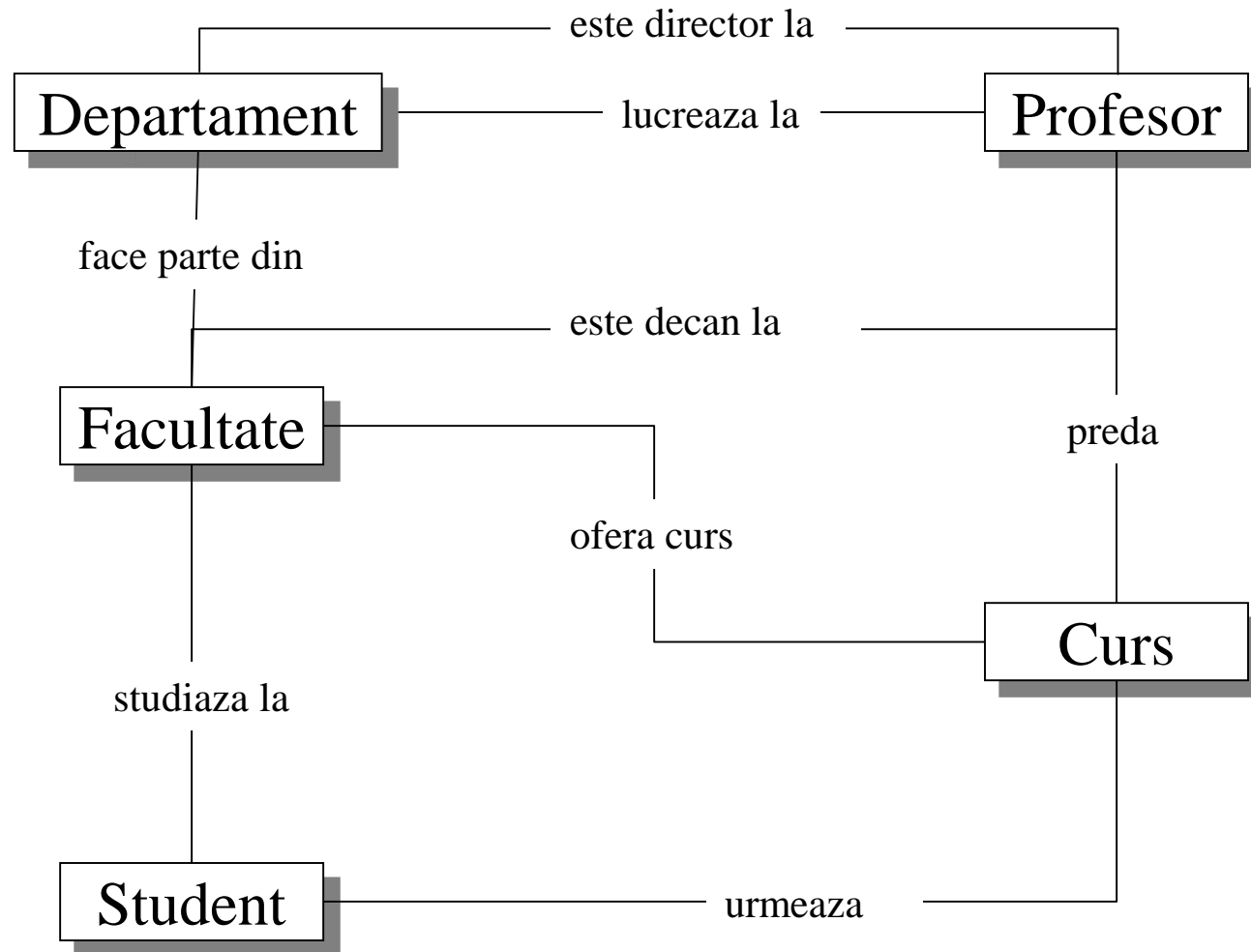
Fiecare entitate e denumita  
in mod unic

Pentru fiecare entitate se  
da o descriere detaliata

# Proiectarea bazelor de date relationale

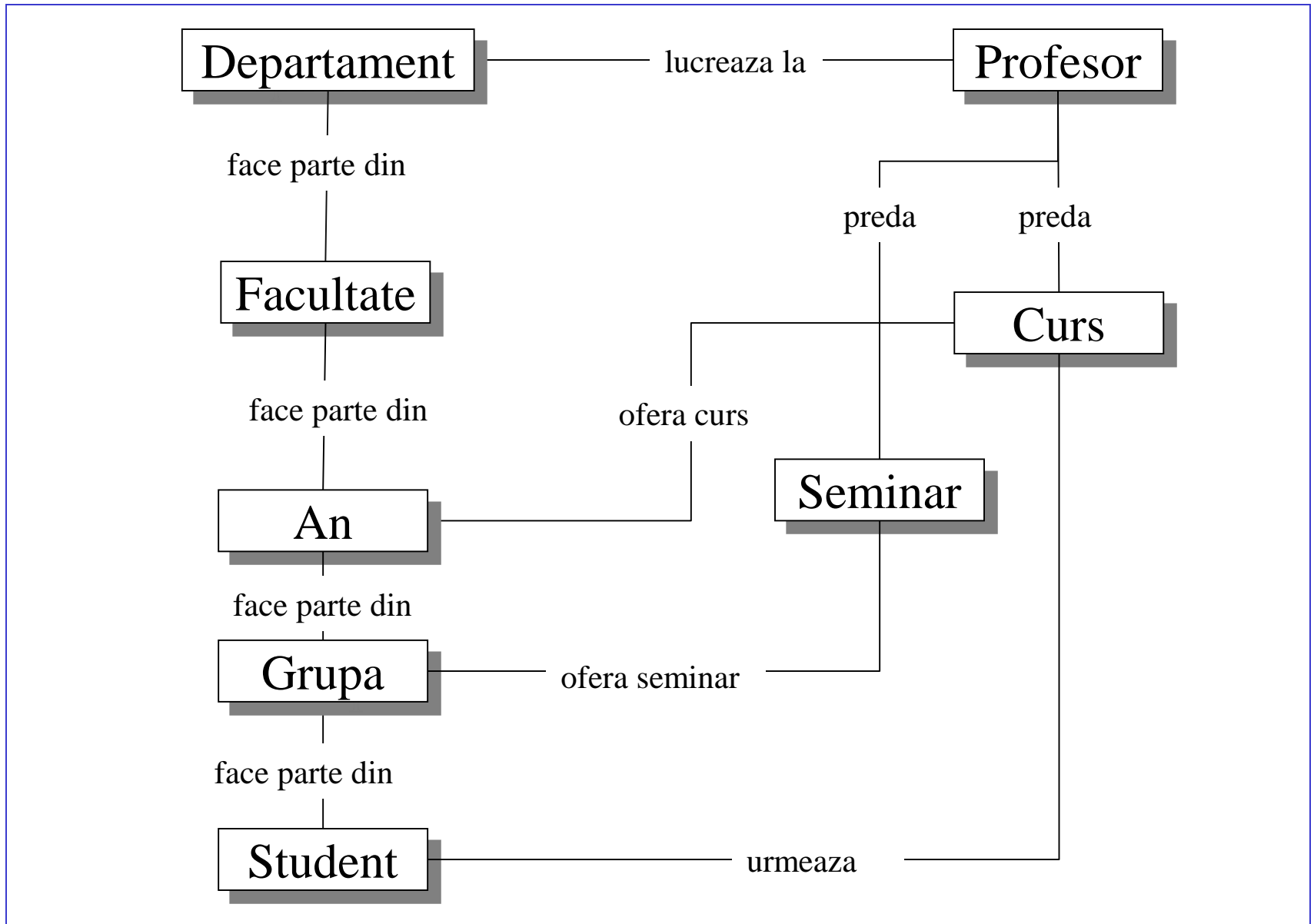


# Proiectarea bazelor de date relationale





# Proiectarea bazelor de date relationale



# Proiectarea bazelor de date relationale

## Relatie

Entitatile pot forma relatii intre ele

Relatiile sunt reprezentate prin verbe

Intre doua entitati poate exista mai mult decat o relatie (exemplu)

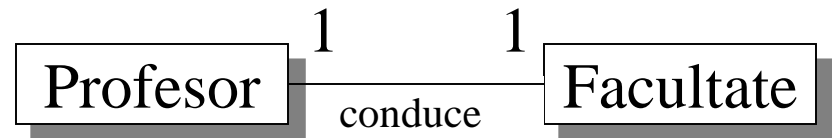
*Cardinalitatea* unei relatii: numarul maxim de entitati din fiecare entitate care participa la o relatie.

## Exemple:

Multi la unul (many-to-one N:1)



Unu la unu (one-to-one 1:1)



Multi la multi (many-to-many N:M)



# Proiectarea bazelor de date relationale

Discutie

Cardinalitate minima/maxima

Atribut al unei entitati, atribut al unei relatii

## **Model entitate-legatura si model relational**

O entitate devine un tabel

Un atribut al unei entitati devine o coloana a tabelului respectiv

O relatie va fi reprezentata fie printr-un tabel special (M:N), fie printr-o cheie straina intr-unul dintre cele doua tabele-entitate, care face referire la o cheie primara in celalalt tabel-entitate.

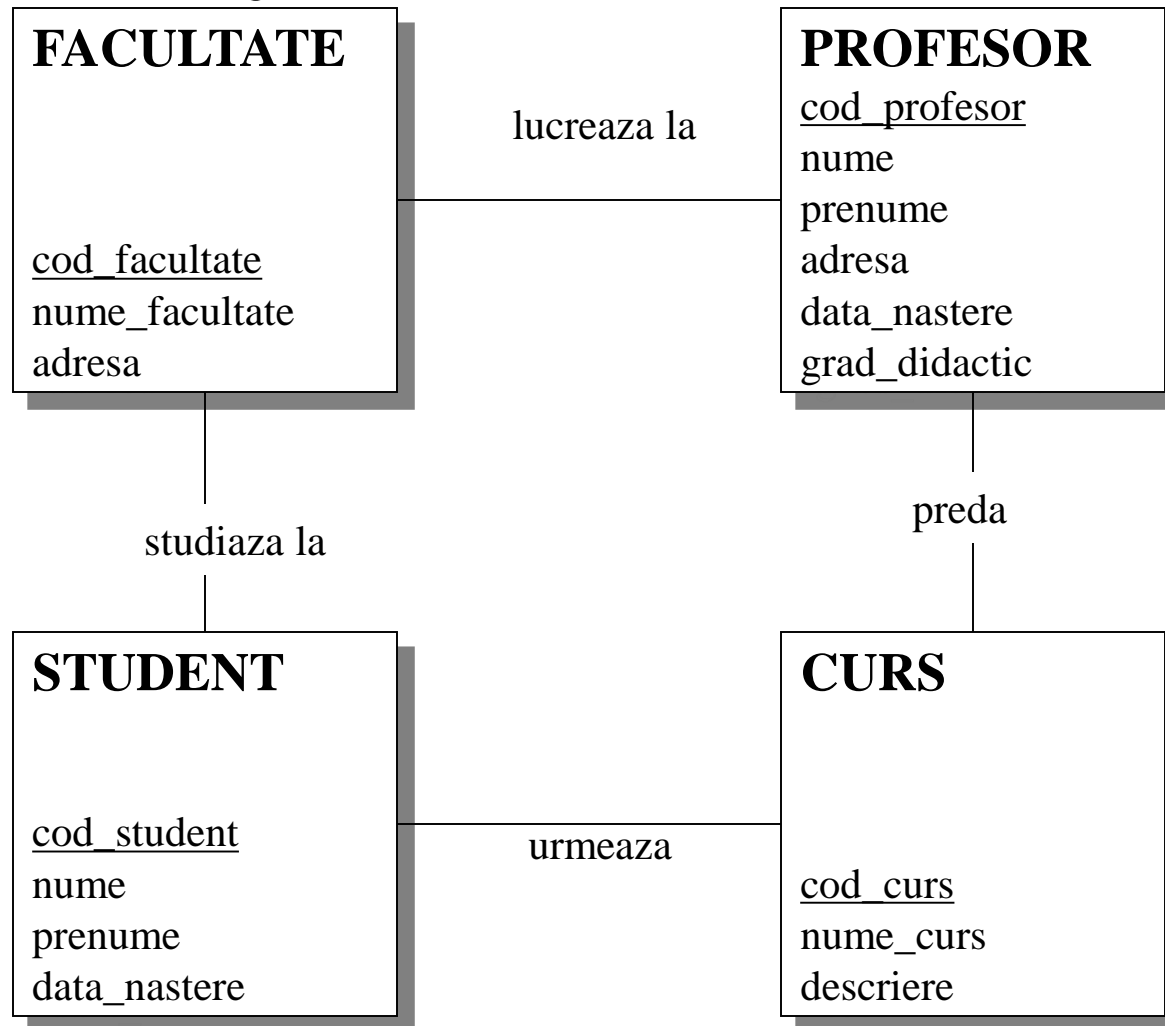
## **Chei primare. Chei naturale si chei artificiale**

Avantaje ale folosirii cheilor artificiale:

- Stabilitatea;
- Simplitatea;
- Nu prezinta ambiguitati;
- Elimina valorile Null (accidentale).

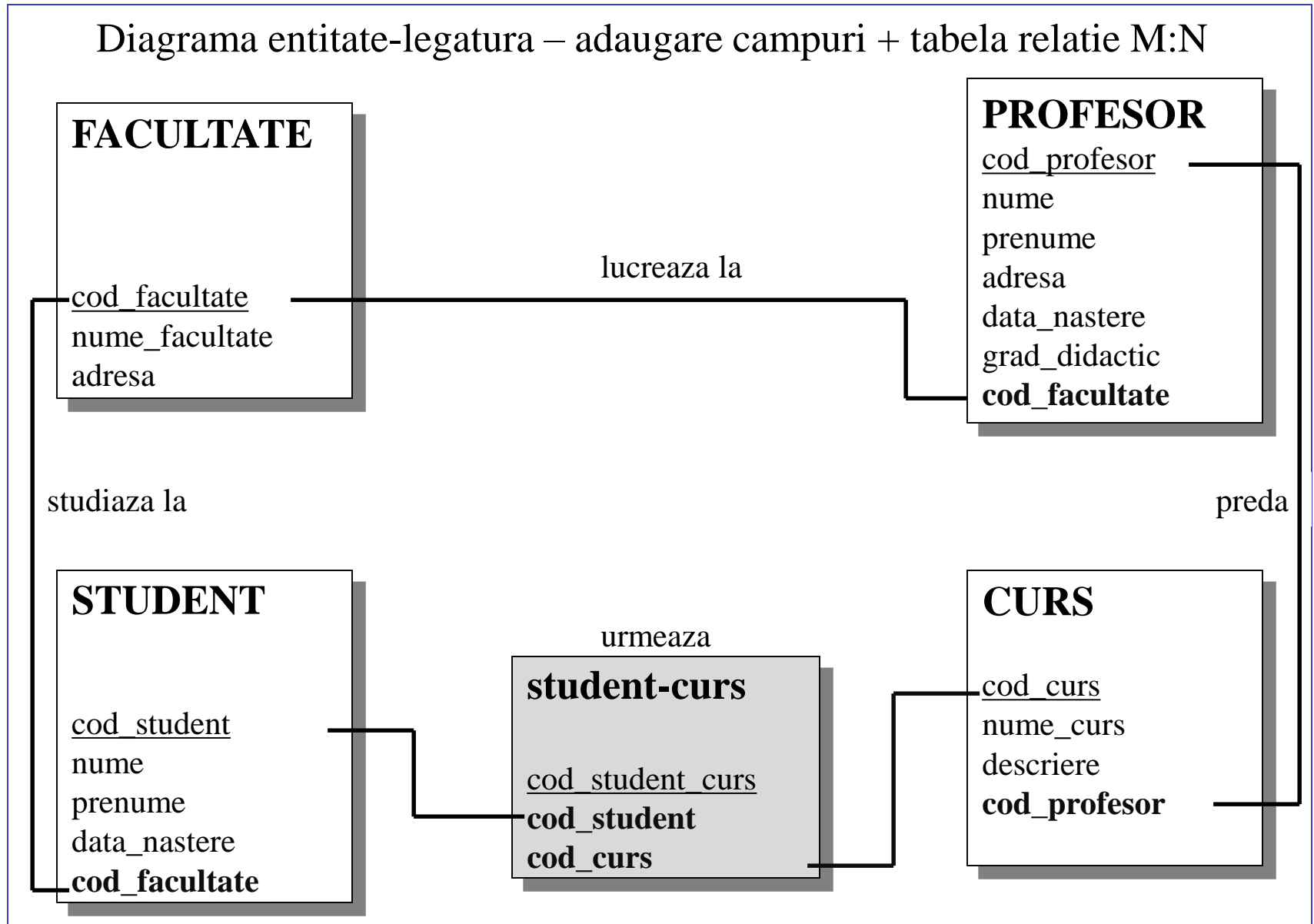
# Proiectarea bazelor de date relationale

Diagrama entitate-legatura



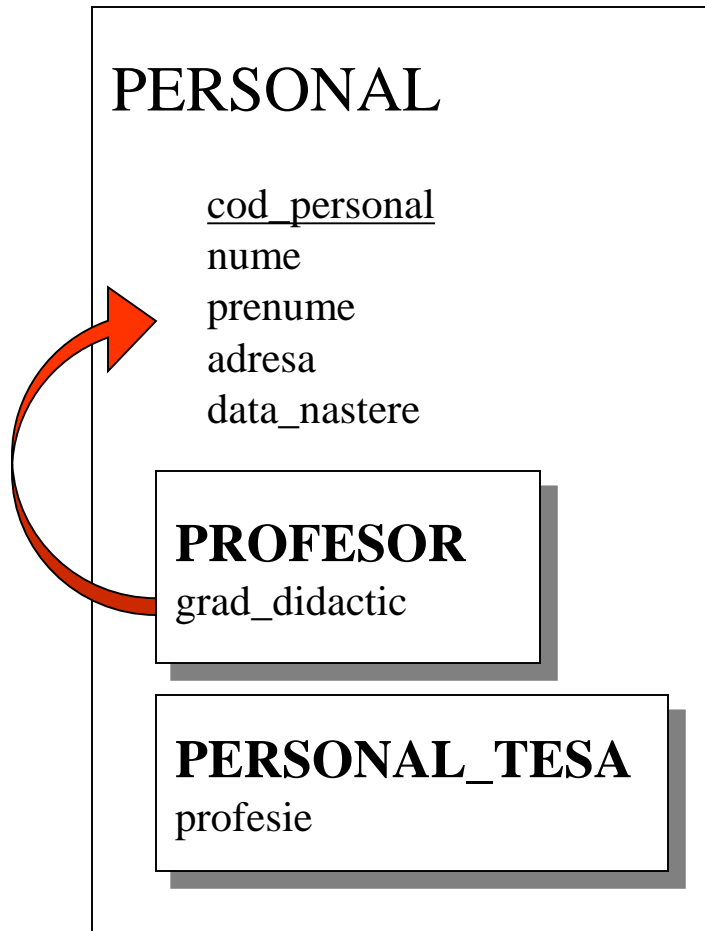
# Proiectarea bazelor de date relationale

Diagrama entitate-legatura – adaugare campuri + tabela relatie M:N



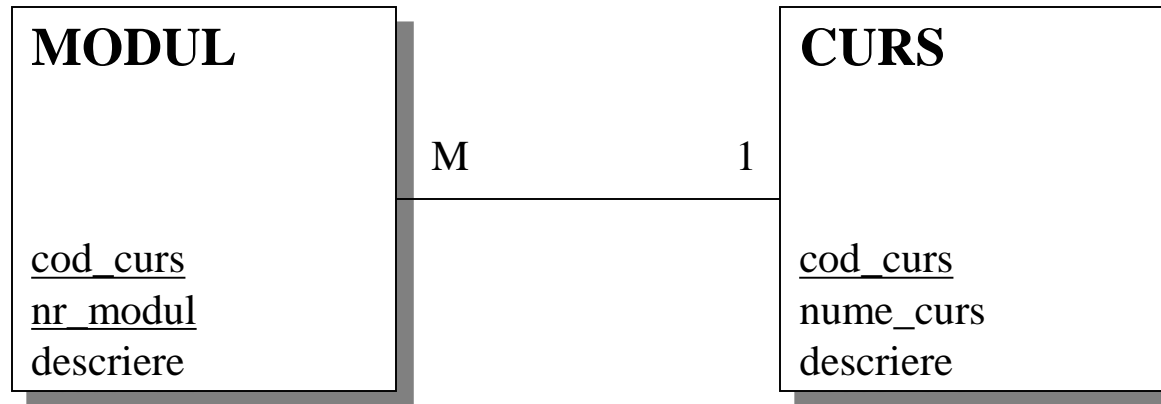
# Proiectarea bazelor de date relationale

Superentitate/subentitate



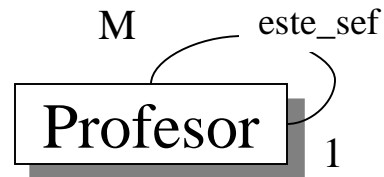
# Proiectarea bazelor de date relationale

Entitate dependenta/entitate master



Modul e o entitate dependenta de entitatea curs (un curs e format din mai multe module). Cardinalitate intre master si detaliu: 1:M, cel putin 1:0 (un curs nu are nici un modul).

Entitate recursiva



# Proiectarea bazelor de date relationale

## Crearea design-ului logic al bazei de date

### Transformarea entitatilor

- Entitatile devin tabele
- Entitatile dependente devin tabele dependente
- Subentitatile devin subtabele (tabele ale caror cheie primara contine cheie straina ce face referinta la cheia primara a tabelul superentitate).

### Transformarea relatiilor

- Relatiile 1:1 devin chei straine, cheia straina fiind plasata in tabelul cu mai putine inregistrari.
- Relatia N:1 devine cheie straina plasate in tabelul care se afla in partea “multi” a relatiei. Exemplu: o facultate are mai multi studenti, un student e la o singura facultate
- O relatie multi-multi N:M se transforma intr-un tabel asociativ, care are doua chei straine pentru cele doua tabele asociate. Exemplu: mai multi studenti-la mai multe cursuri

### Transformarea atributelor

- Atributele simple ale unei entitati devin coloane in tabelul provenit din relatie.
- Atributele repetitive (multivaloare) devin tabele dependente care contin o cheie straina ce face referinta la cheia primara a entitatii si atributul multi-valoare.
- Atributele simple ale unei relatii M:N vor deveni coloane ale tabelului asociativ.



# Proiectarea bazelor de date relationale

## Normalizarea bazei de date

cod_client	nume_client	nr_telefon	cod_comanda	data	cod_articol	nume_articol	cost_articol	cantitate
A1	Petrescu	2338291	C1	12.05.2008	P1	pantaloni	50	100
A1	Petrescu	2338291	C1	12.05.2008	P3	camasa	45	20
A2	Vasilescu	3485734	C2	13.05.2008	P1	pantaloni	50	200
A2	Vasilescu	3485734	C2	13.05.2008	P3	camasa	45	300
A2	Vasilescu	3485734	C2	13.05.2008	P2	bluza	35	10
A1	Petrescu	2338291	C3	13.05.2008	P3	camasa	45	20
A3	Ionescu	5409054	C4	13.05.2008	P1	pantaloni	50	30

cod_client	nume_client	nr_telefon	cod_comanda	data	cod_articol	nume_articol	cost_articol	cantitate
A1	Petrescu	2338291	C1	12.05.2008	P1	pantaloni	50	100
A1	Petrescu	2338291	C1	12.05.2008	P3	camasa	45	20
A2	Vasilescu	3485734	C2	13.05.2008	P1	pantaloni	50	200
A2	Vasilescu	3485734	C2	13.05.2008	P3	camasa	45	300
A2	Vasilescu	3485734	C2	13.05.2008	P2	bluza	35	10
A1	Petrescu	2338291	C3	13.05.2008	P3	camasa	45	20
A3	Ionescu	5409054	C4	13.05.2008	P1	pantaloni	50	30

Dependente functionale pentru care determinantul nu este cheie a tabelului:

cod\_client -> {nume\_client, numar\_telefon}

cod\_comanda -> {data, cod\_client, nume\_client, nr\_telefon}

cod\_articol -> {nume\_articol, cost\_articol}

# Proiectarea bazelor de date relationale

Datorita dependentelor prezente intre attributele relatiei, pot aparea urmatoarele anomalii:

- redundante in date;
- Anomalii de actualizare:
  - Anomalie la insertie;
  - Anomalie la stergere;
  - Anomalie la modificare;

Anomaliile apar datorita dependentelor din baza de date, dependente pentru care determinantul nu este cheie a tabelului.

Normalizarea: procesul reversibil de descompunere a unui tabel relational in tabele cu o structura mai simpla, proces care are ca scop evitarea redundantei datelor si evitarea anomaliilor de actualizare.

Reversibil: descompunerea se face fara pierdere de informatie.

# Proiectarea bazelor de date relationale

## Normalizarea bazei de date

cod_client	nume_client	nr_telefon	cod_comanda	data	cod_articol	nume_articol	cost_articol	cantitate
A1	Petrescu	2338291	C1	12.05.2008	P1	pantalon	50	100
A1	Petrescu	2338291	C1	12.05.2008	P3	camasa	45	20
A2	Vasilescu	3485734	C2	13.05.2008	P1	pantalon	50	200
A2	Vasilescu	3485734	C2	13.05.2008	P3	camasa	45	300
A2	Vasilescu	3485734	C2	13.05.2008	P2	bluza	35	10
A1	Petrescu	2338291	C3	13.05.2008	P3	camasa	45	20
A3	Ionescu	5409054	C4	13.05.2008	P1	pantalon	50	30

CLIENT		
cod_client	nume_client	nr_telefon
A1	Petrescu	2338291
A2	Vasilescu	3485734
A3	Ionescu	5409054

ARTICOL		
cod_articol	nume_articol	cost_articol
P1	pantalon	50
P2	bluza	35
P3	camasa	45

COMANDA		
cod_comanda	data	cod_client
C1	12.05.2008	A1
C2	13.05.2008	A2
C3	13.05.2008	A1
C4	13.05.2008	A3

VANZARI		
cod_comanda	cod_articol	cantitate
C1	P1	100
C1	P3	20
C2	P1	200
C2	P3	300
C2	P2	10
C3	P3	20
C4	P1	30