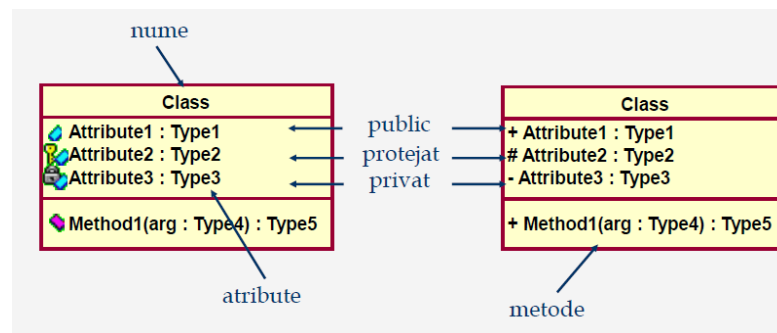


# PROIECTAREA BAZELOR DE DATE

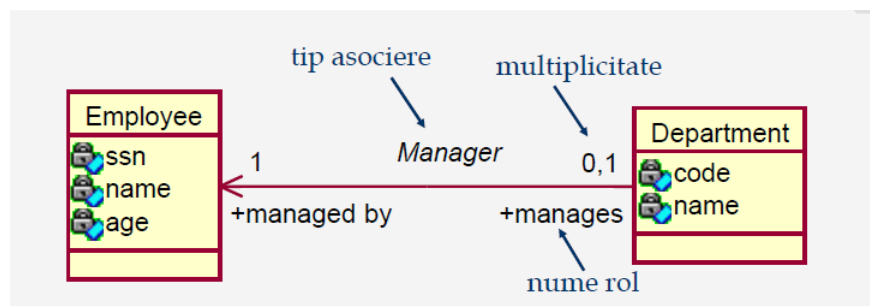
- Proiectare conceptuală (diagrama de clase)
- Proiectare logică (model conceptual => structură de baze de date)
- Rafinarea bazei de date (eliminarea redundanțelor și a problemelor conexe)
- Proiectare fizică și eficientizare (indexare, de-normalizare)

## Diagrama de clase UML

### - Clase



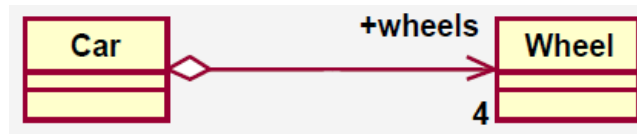
### - Asocieri



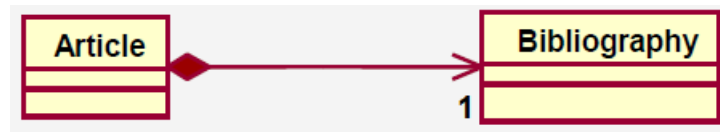
- Multiplicități:
  - valori: 4, 5
  - intervale: 1..10
  - nedefinit: \*
- Navigabilitatea asocierii:
  - un sens
  - bidirecțional

## Tipuri de asocieri:

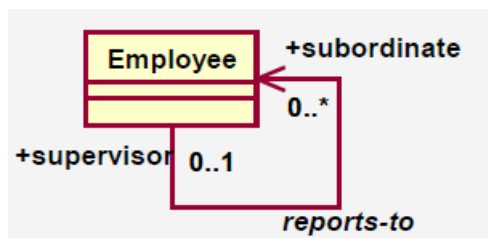
- *Agregare* (asociere parte-întreg) – entitățile pot exista una fără cealaltă



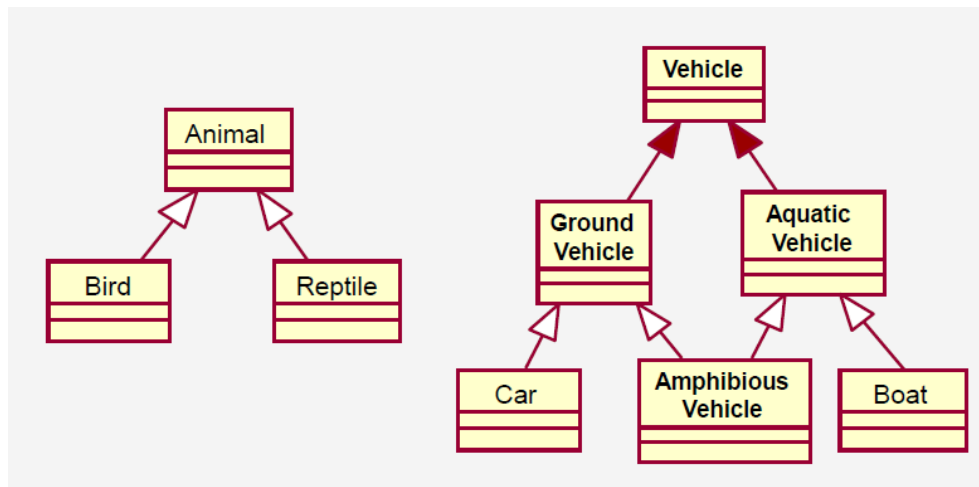
- *Compunere* (“weak entities”) – dacă o entitate este ștearsă, cealaltă entitate nu are sens



- *Asociere reflexivă*



## - Moștenire



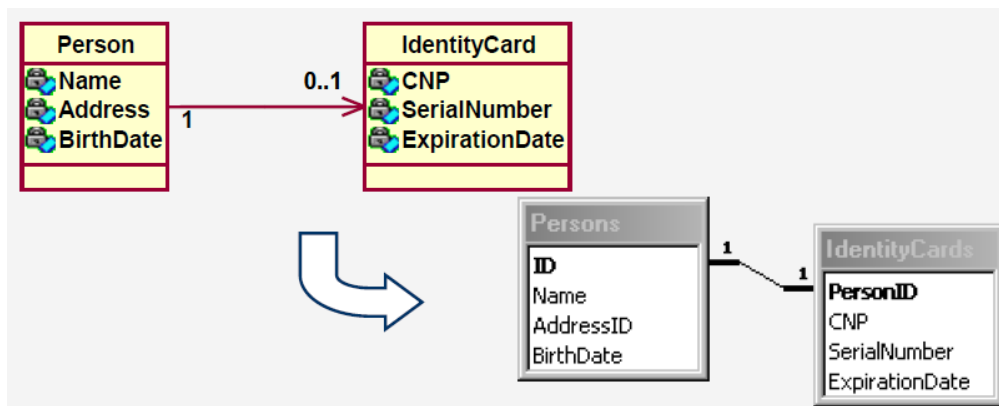
## Transformarea claselor în tabele

- numele tabelii reprezintă *pluralul* numelui clasei
- toate atributele simple sunt transformate în câmpuri
- atributele compuse devin tabele de sine stătătoare
- atributele derivate nu vor avea niciun corespondent în tabelă

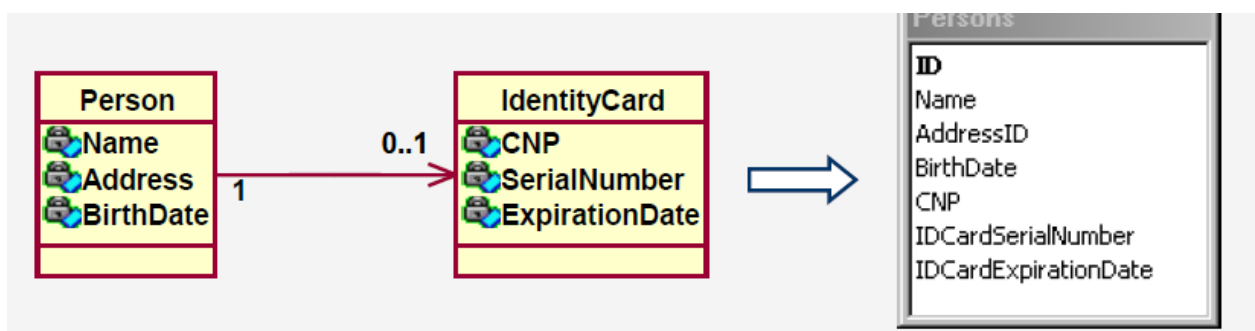
**Chei surogat** = chei care nu sunt obținute din domeniul problemei modelate (chei generate automat de SGBD).

## Transformarea asocierilor simple

- 1: 0,1
  - se creează câte o tabelă corespunzătoare fiecărei clase implicate în asociere
  - cheia tabelii corespunzătoare multiplicității “0, 1” este cheia străină în cea de-a doua tabelă
  - o singură cheie va fi generată automat (de obicei cea corespunzătoare multiplicității “1”)

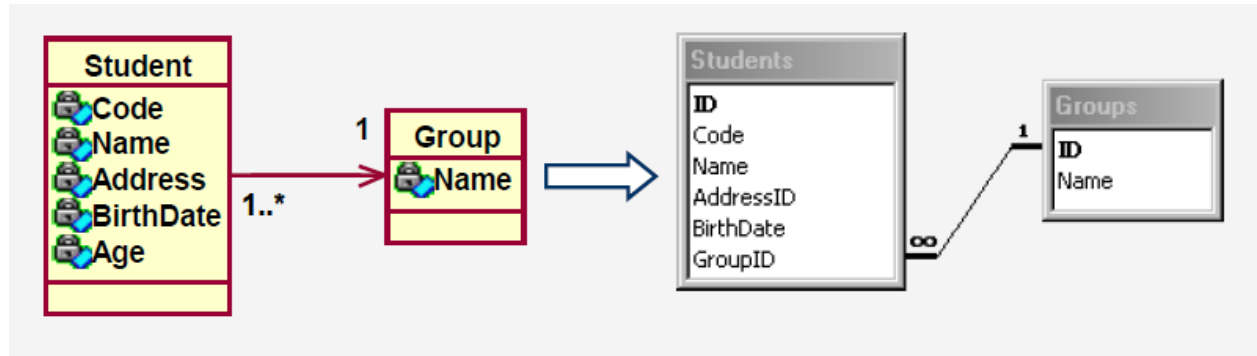


- 1: 1
  - se creează o singură tabelă ce conține attributele ambelor clase asociate

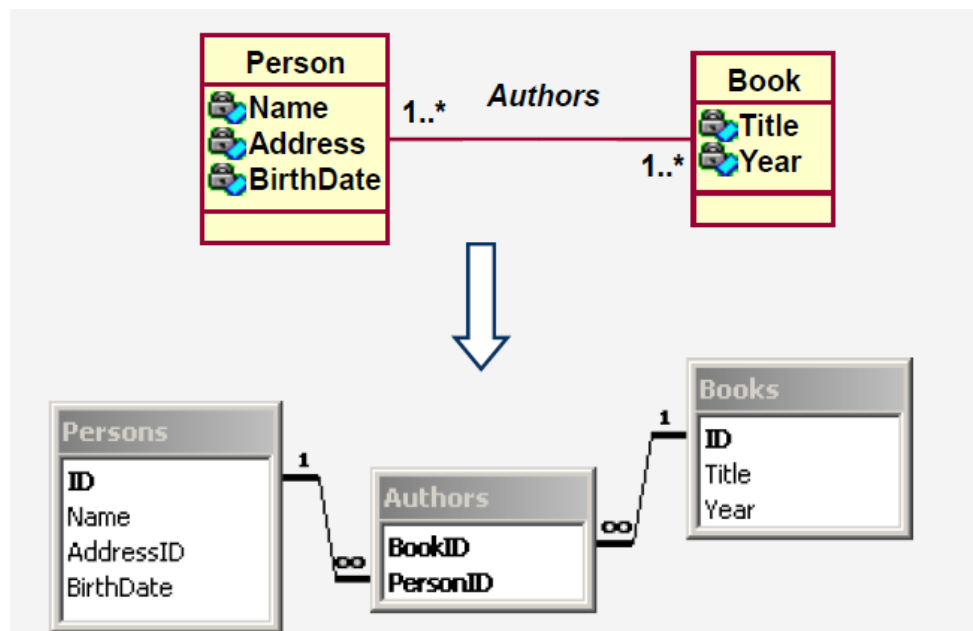


- 1: 1..\*
  - se creează câte o tabelă corespunzătoare fiecărei clase implicate în asociere

- cheia tabelii corespunzătoare multiplicității “1” este cheia străină în cea de-a doua tabelă, corespunzătoare multiplicității “1..\*”



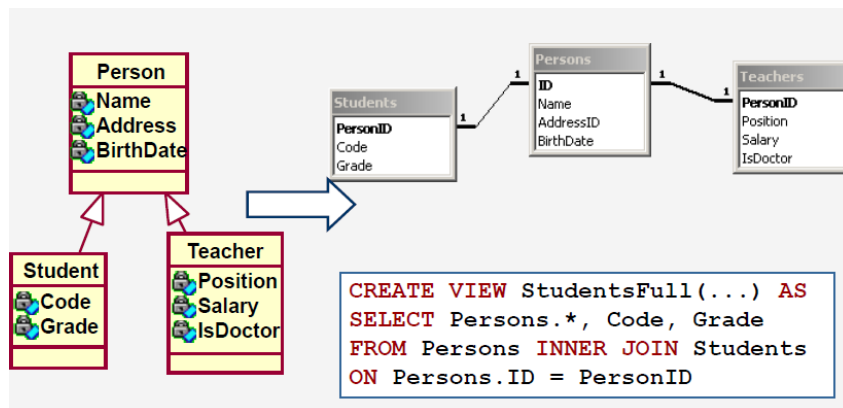
- 1..\* : 1..\*
  - se creează câte o tabelă corespunzătoare fiecărei clase implicate în asociere
  - se creează o tabelă adițională numită tabelă de intersecție (*cross table*)
  - cheile primare corespunzătoare tabelelor inițiale sunt definite ca și *cheie străină* în tabela de intersecție
  - *cheia primară* a tabelii de intersecție este, de obicei, compusă din cele două chei străine spre celelalte tabele; sunt cazuri în care se utilizează și aici cheie surogat



## Transformarea moștenirii

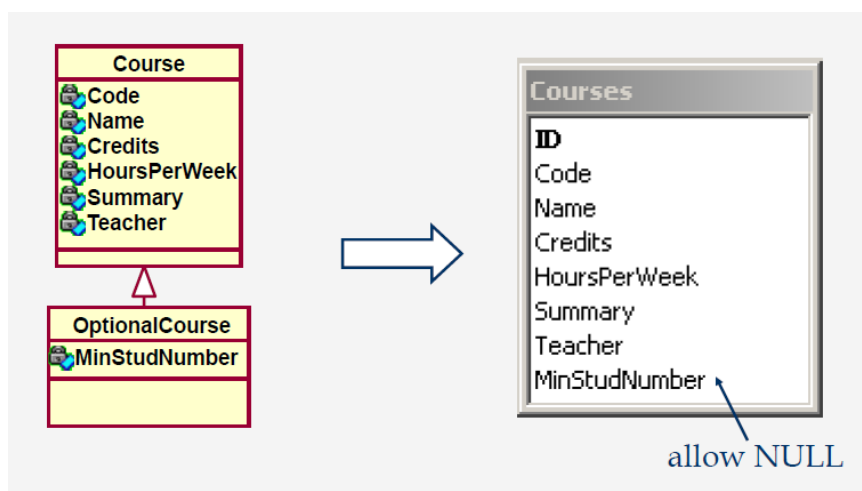
**Metoda 1** – crearea a câte unei table corespunzător fiecărei clase și a câte unui view pentru fiecare pereche super-clasă/subclasă

- ➔ flexibilitate – permite adăugarea viitoarelor subclase fără impact asupra tabelelor/view-urilor deja existente
- ➔ implică crearea celor mai multe table/view-uri
- ➔ posibile problem de performanță deoarece fiecare acces va implica execuția unui *join*



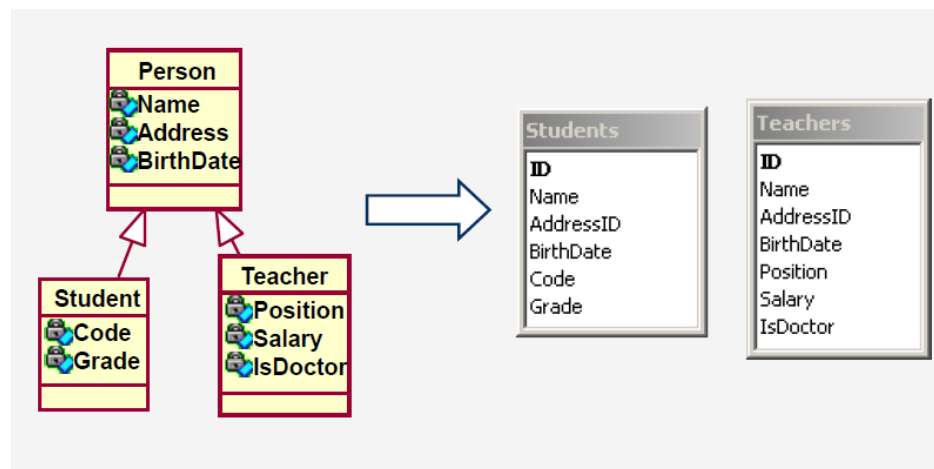
**Metoda 2** – se crează o singură tabelă (corespunzătoare superclasei) și se denormalizează toate attributele subclaselor acesteia

- ➔ implică crearea celor mai puține table/view-uri
- ➔ se obține, de obicei, cea mai mare performanță
- ➔ adăugarea unei noi subclase implică modificări structurale
- ➔ creștere “artificială” a spațiului utilizat



**Metoda 3** – se crează câte un tabel corespunzător fiecărei sub-clase și se de-normalizează attributele super-clasei în fiecare dintre tabelele create

- ➔ performanța obținută este satisfăcătoare
- ➔ adăugarea unei noi subclase nu implică modificări structurale
- ➔ posibile modificări structurale la nivelul superclasei afectează toate tabelele definite



**Care este metoda potrivită?**

Metoda 1 – dacă numărul înregistrărilor stocate în tabel este redus

Metoda 2 – dacă subclasele au instanțe puține

Metoda 3 – dacă superclasa are un număr restrâns de attribute

### **Transformarea agregării/compunerii**

- agregarea și compunerea sunt modelate în mod asemănător modelării asocierilor
- compunere: se utilizează o singură tabelă – deoarece compunerea implică mai multe relații 1:1
- numărul fix de “părți” dintr-un “întreg” presupune introducerea unui număr egal de chei străine în tabela “întreg”
- în cazul implementării compunerii în tabele separate este necesară setarea “ștergerii în cascadă” (în cazul agregării acest lucru nu este necesar)

