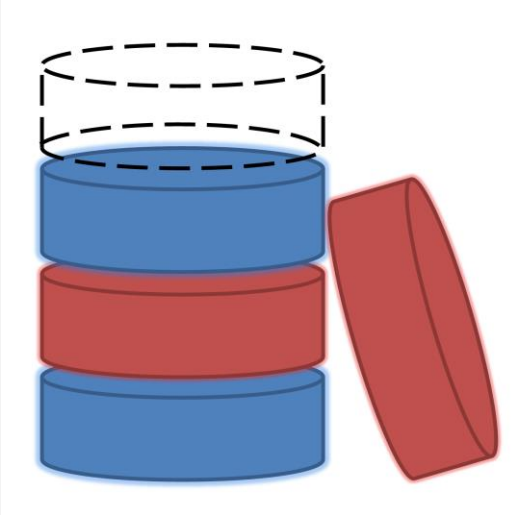
Forme normale

continuare



Descompunerea în 3NF

Evident, procedeul descompunerii din BCNF poate fi utilizat şi pentru descompunerea 3NF.

- Cum asigurăm păstrarea dependențelor?
  - Dacă X→Y nu se păstrează, adăugăm XY.
  - Problema este că XY e posibil să nu respecte 3NF! (pp. că adăugăm CJP pt `păstrarea' JP→C. Dacă însă are loc şi J→C atunci nu e corect.)

- *Rafinare*: În loc de a utiliza mulțimea inițială F, folosim o *acoperire minimală a lui F*.

Redundanța in DF

- Un atribut $A \in \alpha$ e redundant în DF $\alpha \rightarrow B$ dacă

$$(F - \{\alpha \rightarrow B\}) \cup \{\alpha\text{-}A \rightarrow B\} \quad \equiv \quad F$$

- Pentru a verifica dacă $A \in \alpha$ e redundant în $\alpha \rightarrow B$, calculăm $(\alpha\text{-}A)^+$. Apoi $A \in \alpha$ e redundant în $\alpha \rightarrow B$ dacă $B \in (\alpha\text{-}A)^+$

- *Exercițiu*: Care sunt atributele redundante în $AB \rightarrow C$ având: $\{AB \rightarrow C, A \rightarrow B, B \rightarrow A\}$?

Redundanța in DF

- O DF $f \in$ F e redundantă dacă F -{f} e echivalent cu F

- Verificăm că $\alpha \rightarrow A$ e redundantă in F, calculând α^+ pe baza F-{$\alpha \rightarrow A$}. Atunci $\alpha \rightarrow A$ e redundantă în F dacă $A \in \alpha^+$

- *Exercițiu*: Care sunt dependențele funcționale redundante în:
  {$A \rightarrow C$, $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$, $C \rightarrow A$}?

Acoperire minimală

- O acoperire minimală pentru mulțimea F de dependențe funcționale este o mulțime G de dependențe funcționale pentru care:
  1. Fiecare DF din G e de forma $\alpha \rightarrow A$
  2. Pt fiecare DF $\alpha \rightarrow A$ din G, α nu are atribute redundante
  3. Nu sunt DF redundante in G
  4. G şi F sunt echivalente

Fiecare multime de DF are cel putin o acoperire minimala!

Algoritm de calcul al acoperirii minimale pt F:

1. Folosim descompunerea pentru a obține DF cu 1 atribut în partea dreaptă.
2. Se elimină atributele redundante
3. Se elimină dependențele funcționale redundante

Calcul Acoperire Minimală

Fie F = {ABCD →E, E →D, A →B, AC →D}

Atributele BD din ABCD → E sunt redundante:

⇨ F = {AC→E, E→D, A→B, AC→D}

AC → D este redundanta

⇨ F = {AC → E, E → D, A → B}

care este o acoperire minimala

Acoperirile minimale nu sunt unice
(depind de ordinea de alegere a DF/atr. redundante)

Decompunere în 3NF

Input: Schema R with *F* which is a minimal cover
Output: A dependency-preserving, lossless-join 3NF decomposition of R

Initialize D = $\varnothing$
Apply *union rule* to combine FDs in F with same L.H.S. into a single FD

For each FD $\alpha \rightarrow \beta$ in F do
Insert the relation schema $\alpha\beta$ into D

Insert δ into D, where δ is some key of R

Remove redundant relation schema from D as follows:
delete R_i from D if $R_i \subseteq R_j$, where $R_j \in$ D

return D

Exemplu

Fie R(A,B,C,D,E) cu dependentele functionale:

F = {ABCD $\rightarrow$E, E $\rightarrow$D, A $\rightarrow$B, AC $\rightarrow$D}

- Acoperirea minimală a F este {AC $\rightarrow$E, E $\rightarrow$D, A $\rightarrow$B}
- Unica cheie: AC
- R nu e in 3NF deoarece A $\rightarrow$B nu respectă 3NF
- descompunerea 3NF a R:
  - Relații pentru fiecare DF: R_1(A, C, E), R_2(E,D), si R_3(A,B)
  - Relație pentru cheia lui R: R_4(A, C)
  - Eliminare relație redundanta: R_4 (deoarece $R_4 \subseteq R_1$)
  - $\Rightarrow$ descompunerea 3NF este {R_1(A,C,E),R_2(E,D),R_3(A,B)}

- Descompunerea 3NF nu este unică. Depinde de:
  - Alegerea *acoperirii minimale* sau
  - Alegerea *relatiei redundante care va fi eliminata*

Din nou despre... descompunere

- Descompunerea este ultima solutie de rezolvare a problemelor generate de redundanțe & anomalii

- Excesul poate fi nociv!

Exemplu:

R = (Teacher,Dept, Phone, Office)

cu DF F = {Teacher → Dept Phone Office}

- Uneori, din motive de performanță se practica de-normalizarea

Dependențe multivaloare

course	teacher	book
alg101	Green	Alg Basics
alg101	Green	Alg Theory
alg101	Brown	Alg Basics
alg101	Brown	Alg Theory
logic203	Green	Logic B.
logic203	Green	Logic F.
logic203	Green	Logic intro.

relația e în BCNF

Dependențe multivaloare

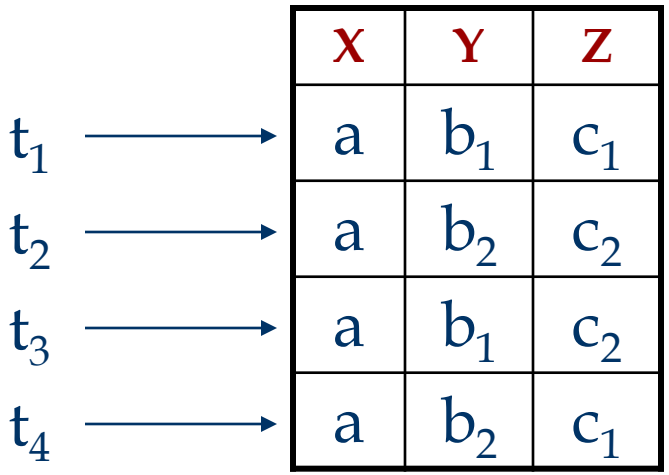
■ Fie α, β două submulțimi de atribute din R. Dependența multivaloare $\alpha \rightarrow\rightarrow \beta$ este respectată de R dacă, pentru fiecare instanță validă r a lui R, fiecare valoare α este asociată cu o mulțime de valori pentru β și această mulțime valori este independentă de valorile altor atribute.

■ Formal: dacă $\alpha \rightarrow\rightarrow \beta$ e respectată de R şi γ=R-$\alpha\beta$, următoarea afirmație e adevărată pentru orice instanță validă r a lui R:

$$t_1, t_2 \in r \text{ şi } \pi_\alpha(t_1) = \pi_\alpha(t_2) \Rightarrow$$
$$\exists t_3 \in r \text{ a.î. } \pi_{\alpha\beta}(t_1) = \pi_{\alpha\beta}(t_3) \text{ şi } \pi_\gamma(t_2) = \pi_\gamma(t_3)$$

Ca şi consecință, pentru t_2 şi t_1 se poate deduce că există şi $t_4 \in r$ a.î. $\pi_{\alpha\beta}(t_2) = \pi_{\alpha\beta}(t_4)$ şi $\pi_\gamma(t_1) = \pi_\gamma(t_4)$

Dependențe multivaloare



	X	Y	Z
t_1	a	b_1	c_1
t_2	a	b_2	c_2
t_3	a	b_1	c_2
t_4	a	b_2	c_1

$\forall t_1, t_2 \in r$ şi $\pi_X(t_1) = \pi_X(t_2) \Rightarrow$
$\exists\, t_3 \in r$ astfel încât
$\pi_{XY}(t_1) = \pi_{XY}(t_3)$,
$\pi_Z(t_2) = \pi_Z(t_3)$

Reguli adiționale:

Complementare: $X \rightarrow\rightarrow Y \Rightarrow X \rightarrow\rightarrow R - XY$

Augumentare: $X \rightarrow\rightarrow Y, Z \subseteq W \Rightarrow WX \rightarrow\rightarrow YZ$

Tranzitivitate: $X \rightarrow\rightarrow Y, Y \rightarrow\rightarrow Z \Rightarrow X \rightarrow\rightarrow Z - Y$

Replicare: $X \rightarrow Y \Rightarrow X \rightarrow\rightarrow Y$

Fuzionare: $X \rightarrow\rightarrow Y, W \cap Y = \emptyset, W \rightarrow Z, Z \subseteq Y \Rightarrow X \rightarrow Z$

A patra formă normală (4NF)

Definiție. Fie R o schemă relațională şi F o mulțime de dependențe funcționale şi multivaloare pe R.

Spunem că R este în a patra forma normală NF4 dacă este în 3NF și pentru orice dependență multivaloare $X \twoheadrightarrow Y$:

- $Y \subseteq X$ sau
- $XY = R$ sau
- X e super-cheie

A patra formă normală (4NF)

course	teacher	book
alg101	Green	Alg Basics
alg101	Green	Alg Theory
alg101	Brown	Alg Basics
alg101	Brown	Alg Theory
logic203	Green	Logic B.
logic203	Green	Logic F.
logic203	Green	Logic intro.

course→→teacher

Relatia se poate descompune in:
(Course,Teacher)
si (Course,Book)

course	teacher
alg101	Green
alg101	Brown
logic203	Green

course	book
alg101	Alg Basics
alg101	Alg Theory
logic203	Logic B.
logic203	Logic F.
logic203	Logic intro.

Dependența *Join*

- Spunem ca R satisface dependența *join* $\otimes\{R_1,\ldots,R_n\}$ dacă $R_1, R_2,\ldots,R_n$ este o descompunere cu joncțiuni fără pierderi a lui R.

O dependență multivaloare $X \rightarrow\rightarrow Y$ poate fi exprimată ca o dependență join: $\otimes\{XY, X(R\text{-}Y)\}$.

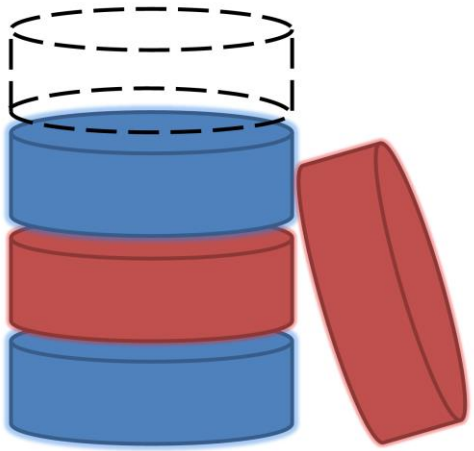
A cincea formă normală (5NF)

O relație R este în NF5 dacă şi numai dacă pentru orice dependență *join* a lui R:

- R_i=R pentru un *i* oarecare, sau
- dependența este implicată de o mulțime de dependențe functionale din R în care partea stângă e o cheie pentru R

Proiectarea bazelor de date

Partea 1



5

Proiectarea bazelor de date

- Proiectare conceptuală (ex. diagrama de clase)
  - Identificarea entităților şi a relațiilor dintre ele
- Proiectarea logică
  - Transformarea modelului conceptual într-o structură de baze de date (relațională sau nu)
- Rafinarea bazei de date (normalizare)
  - Eliminarea redundanțelor şi a problemelor conexe
- Proiectare fizică şi eficientizare
  - Indexare
  - De-normalizare!

Diagrama de clase UML - Clase

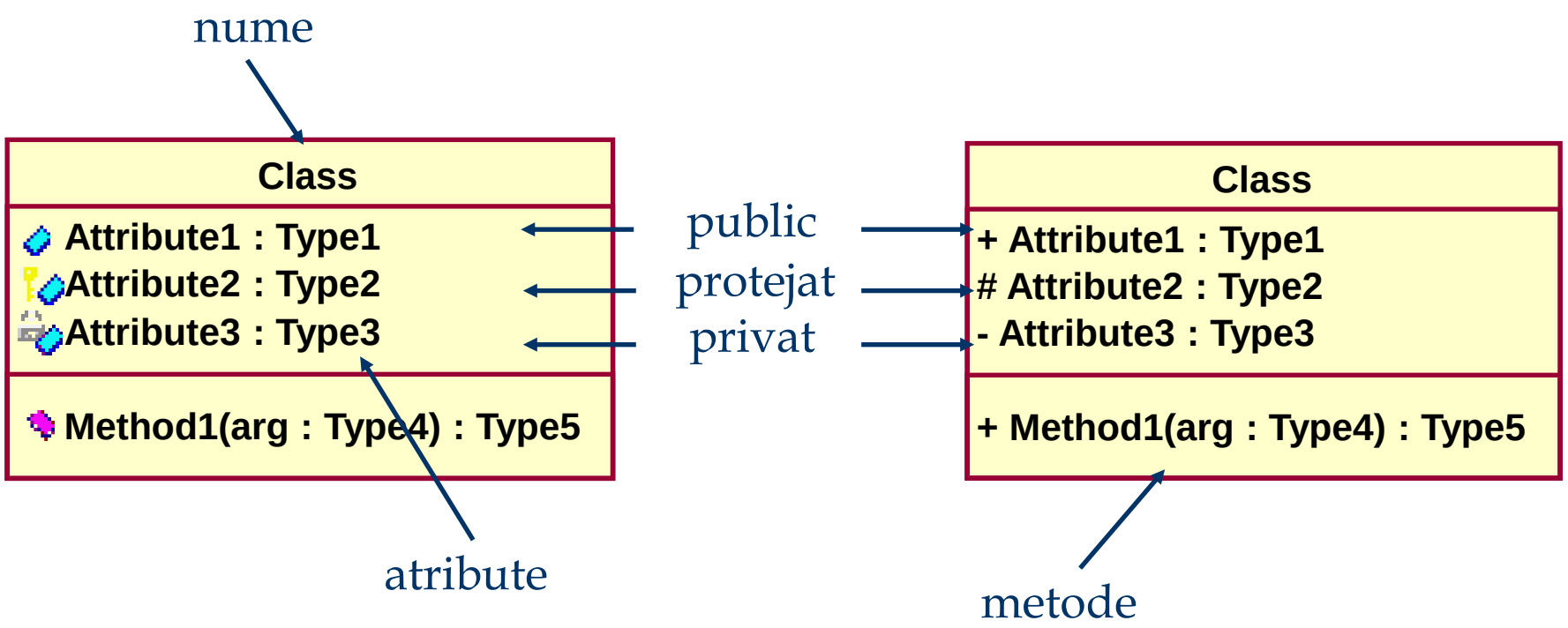
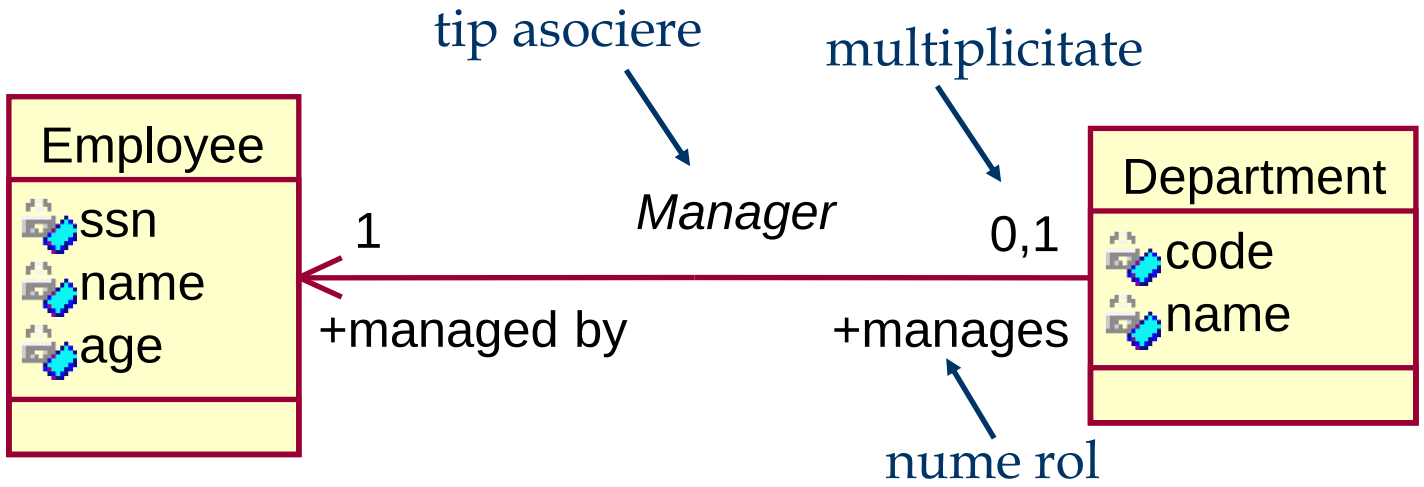


Diagrama de clase UML - Asocieri



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

Diagrama de clase UML - Asocieri

- Citirea numelor de rol

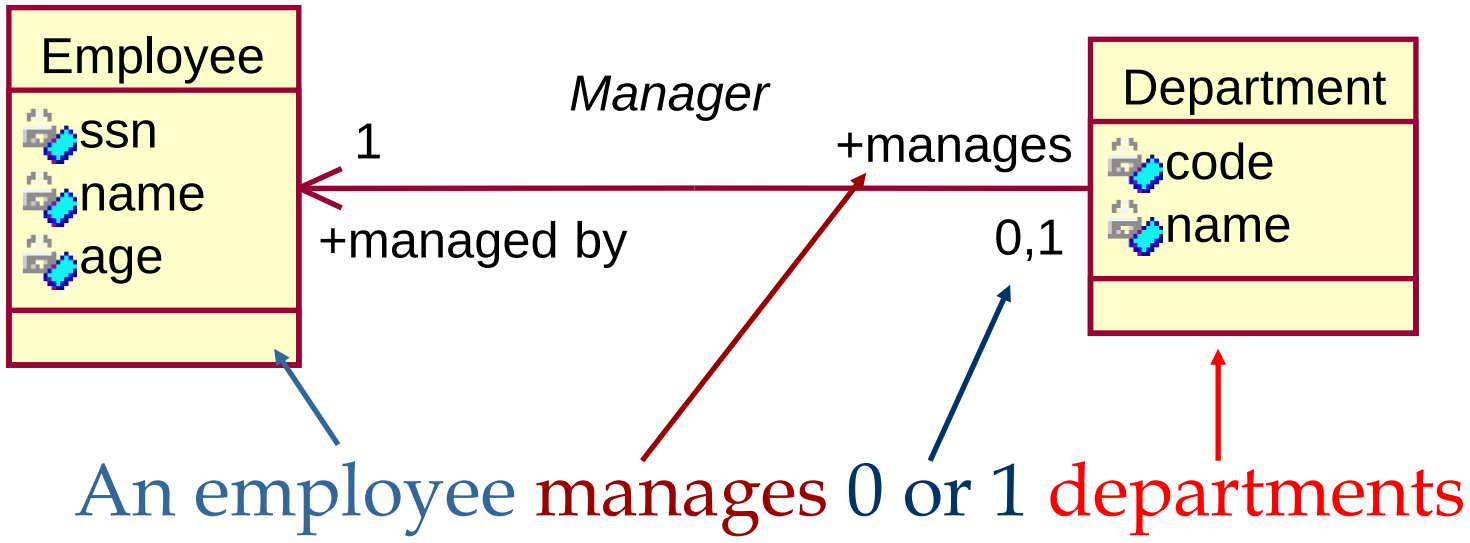
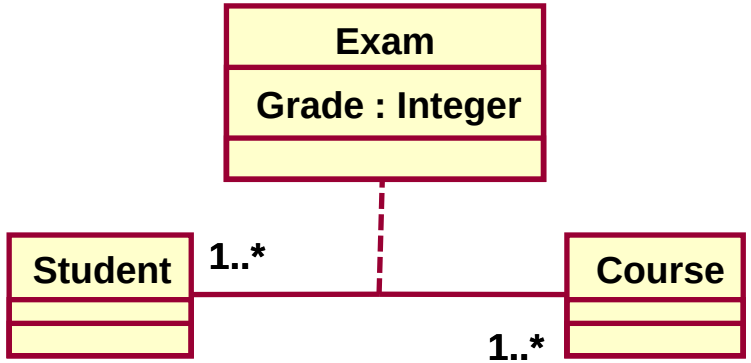


Diagrama de clase UML - Asocieri

- Agregare
  - asociere parte-întreg

- Compunere
  - "weak entities"

- Clasă asociere



- Asociere reflexivă

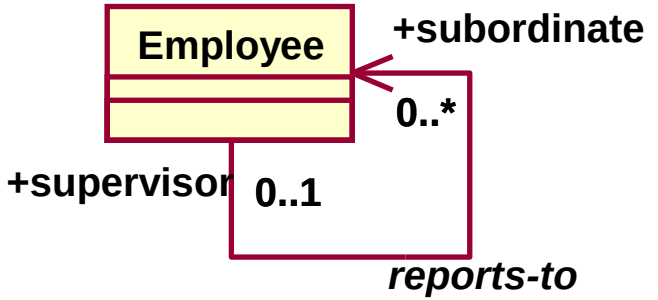
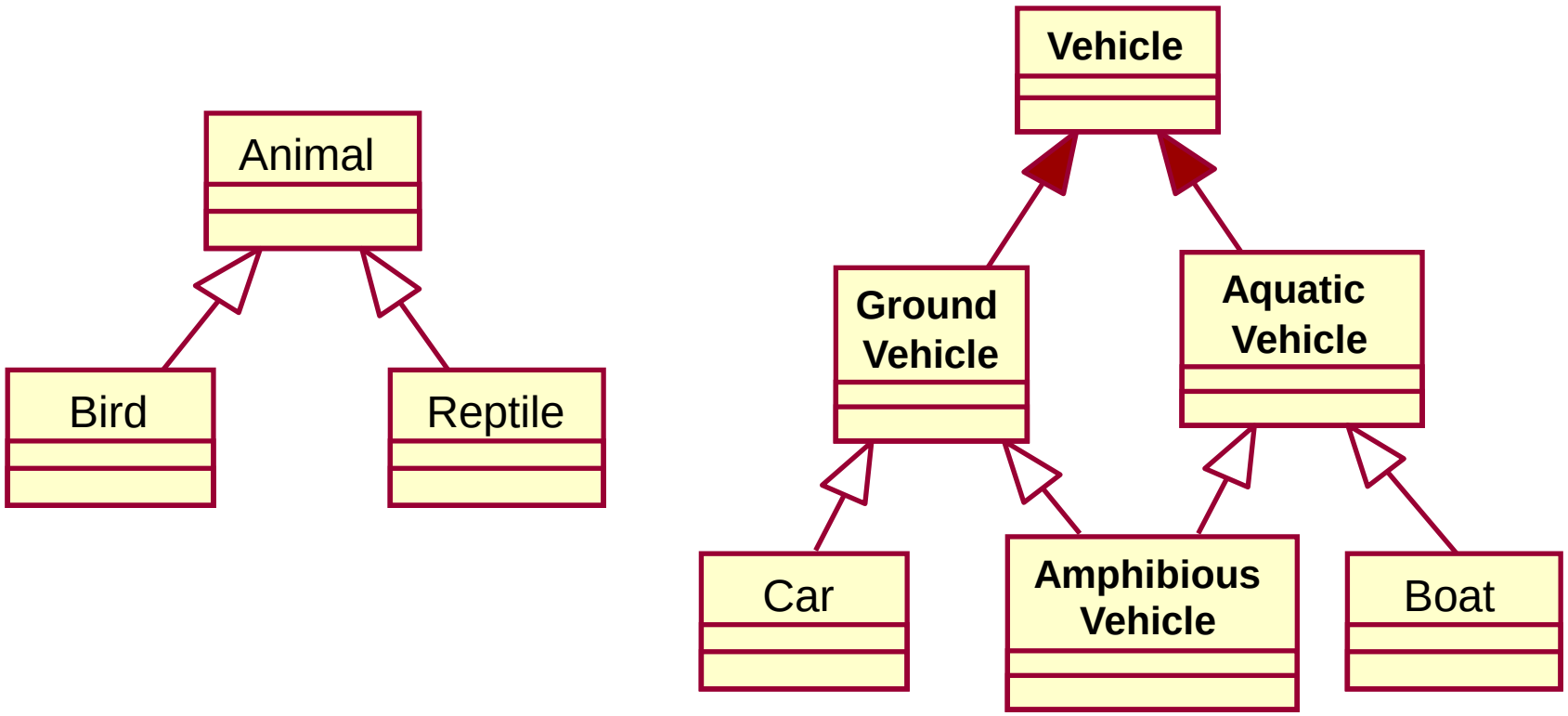


Diagrama de clase UML – Moștenire

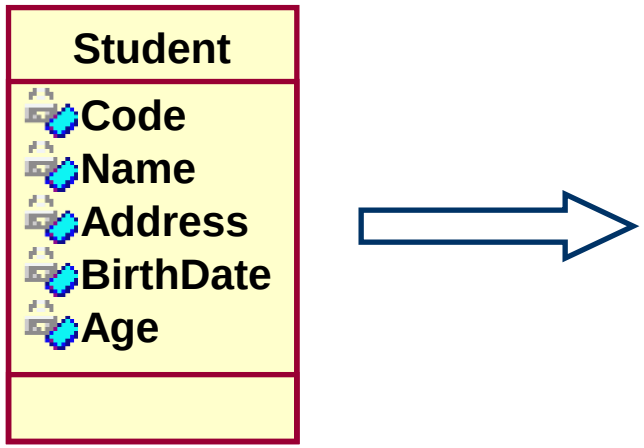


Modelul conceptual ⇒ bază de date relațională

- Transformare 1:1 a claselor în tabele:
  - Prea multe tabele – pot rezulta mai multe tabele decât este necesar
  - Prea multe op. *join* – consecință imediată a faptului că se obțin prea multe tabele
  - Tabele lipsă – asocierile m:n între clase implică utilizarea unei tabele speciale (*cross table*)
  - Tratarea necorespunzătoare a moştenirii
  - Denormalizarea datelor – anumite date se regăsesc în mai multe tabele

Transformarea claselor în tabele

- Numele tabelei reprezintă pluralul numelui clasei
- Toate atributele simple sunt transformate în câmpuri
- Atributele compuse devin tabele de sine stătătoare
- Atributele derivate nu vor avea nici un corespondent în tabelă



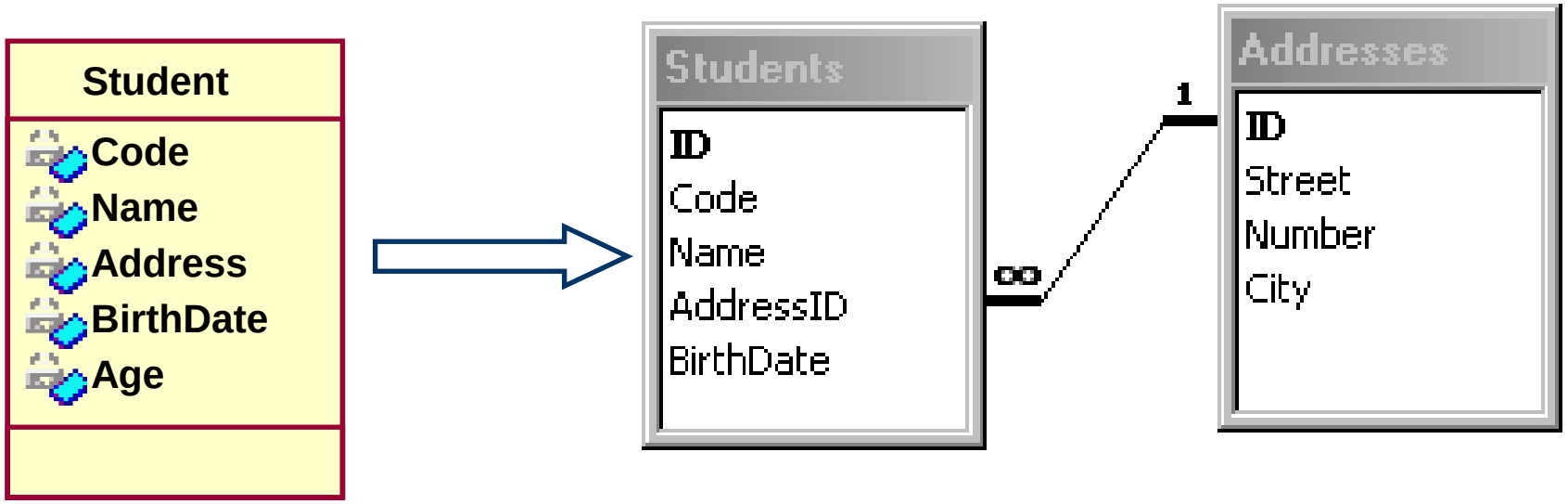
Students (Code, Name, BirthDate)

Addresses (Street, Number, City)

Transformarea claselor în tabele

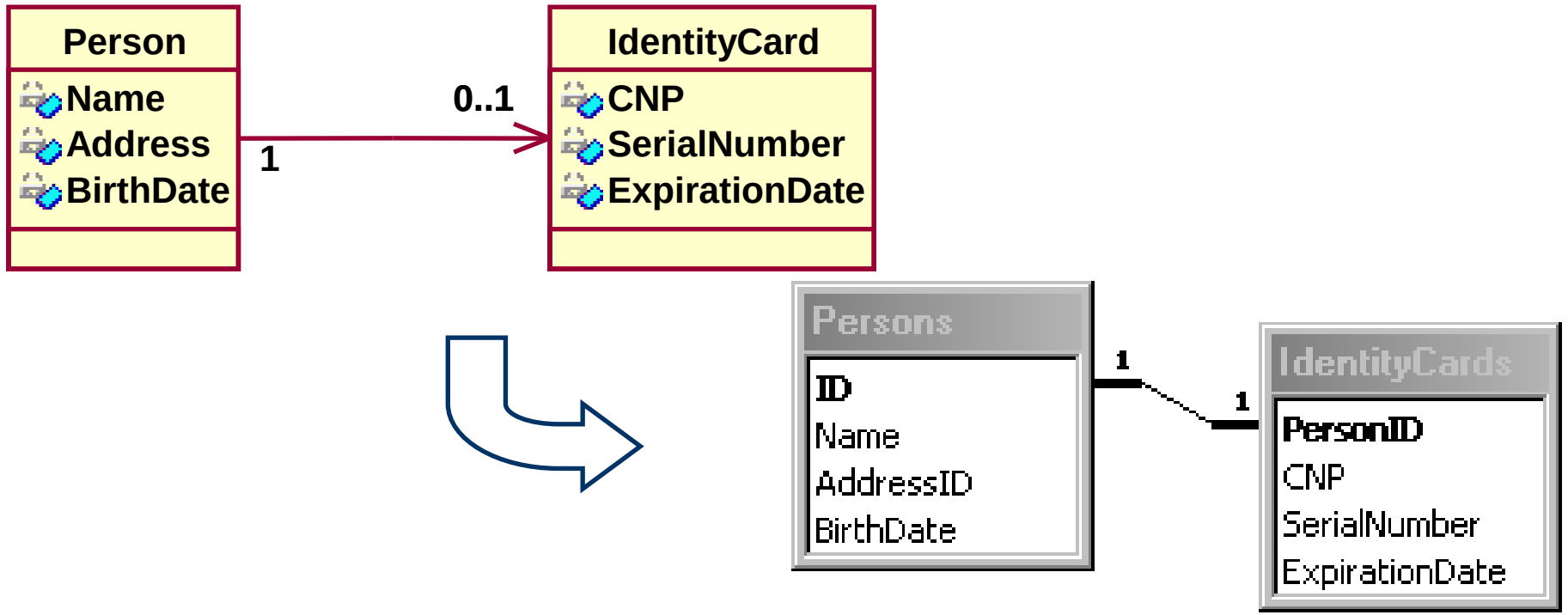
- Chei surogat – chei care nu sunt obținute din domeniul problemei modelate
- Conceptul de cheie nu este definit în cadrul claselor UML
- O *bună practică*: utilizarea (atunci când este posibil) a cheilor de tip întreg generate automat de SGBD:
  - uşor de întreținut (responsabilitatea sistemului)
  - eficient (interogări rapide)
  - simplifică definire cheilor străine
- Disciplină de proiectare a BD:
  - toate cheile surogat vor fi numite **ID**
  - toate cheile străine se numesc **<NumeTabel>ID**

Transformarea claselor în tabele (cont)



Transformarea asocierilor simple

- 1 : 0,1
  - se crează câte o tabelă corespunzătoare fiecărei clase implicate în asociere
  - cheia tabelei 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”)



Transformarea asocierilor simple (cont)

- 1 : 1
  - se crează o singură tabelă ce conține atributele ambelor clase asociate
  - aceasta variantă de transformare se aplică şi asocierilor *"1 : 0,1"* atunci când este vorba de un număr relativ mic de cazuri in care obiectele primei clase nu sunt legate de obiectele celei de-a doua clase