



# BAZE DE DATE

Proiectarea bazelor de date relaționale

Normalizare

Mihaela Elena Breabăn

© FII 2015-2016

# Normalizare

---

- ▶ Dependențe funcționale (revizitat)
- ▶ 1NF, 2NF, 3NF
- ▶ Forma normală Boyce-Codd (BCNF)
- ▶ Dependențe multivaluate (revizitat)
- ▶ Forma normală 4 (4NF)
- ▶ Denormalizare

# Proiectarea schemei

---

- ▶ De obicei mai multe variante de proiectare
  - ▶ Unele sunt (mult) mai bune decât altele
  - ▶ Cum alegem?
- 
- ▶ Teorie pentru proiectarea bazelor de date relaționale cu fundamente în algebra relațională, introdusă de Codd între '70-'74
    - ▶ Eliminarea anomaliilor la modificări în date
    - ▶ Minimizarea necesității reproiectării când sunt necesare extensii ale structurii
    - ▶ Evitarea avantajării anumitor interogări

# Exemplu

## Schemă cu anomalii

---

- ▶ Informații cu privire la candidatii la admitere
  - ▶ CNP și nume
  - ▶ Universitatea la care s-a candidat
  - ▶ Liceele de la care provin candidații (și orașele)
  - ▶ Hobby-urile candidaților

Candidati(CNP, aNume, uNume, lIceu, lOraș, hobby)

Ioana cu CNP-ul **2810605222111** a studiat la **Negruzzi** în **Iași**, candidează la **Cuza, Asachi** și la **Babes-Bolyai**, îi place să joace **tenis** și să cânte la **chitară**

Câte tuple sunt necesare a fi inserate în relația **Candidati** pentru a păstra toate informațiile despre Ioana?

# Anomalii de proiectare

---

- ▶ Redundanță
- ▶ Anomalii de actualizare
- ▶ Anomalii la ștergere

# Exemplu

## Schemă fără anomalii

---

- ▶ Informații cu privire la aplicațiile de admitere
  - ▶ CNP și nume
  - ▶ Universitatea la care s-a candidat
  - ▶ Liceele de la care provin candidații (și orașele)
  - ▶ Hobby-urile candidaților

`Abso1vent(CNP, aNume)`

`Candidat(CNP, uNume)`

`Liceu(CNP, codLiceu)`

`LocatieLiceu(codLiceu, lNume, lOraș)`

`Hobbies(CNP, hobby)`

???

- 
- ▶ Informații cu privire la cursurile luate de studenți
    - ▶ Studenții au id-uri unice și nume (nu sunt unice)
    - ▶ Cursurile au număr de identificare unic și titlu (nu unic)
    - ▶ Studenții iau un curs într-un anumit an și primesc o notă
  - ▶ Care e schema recomandată?
    - ▶ Studiază(sID, nume, cID, titlu, an, notă)
    - ▶ Curs(cID, titlu, an), Studiază(sID, cID, notă)
    - ▶ Student(sID, nume), Curs(cID, titlu), Studiază(sID, cID, an, notă)
    - ▶ Student(sID, nume), Curs(cID, titlu), Studiază(nume, titlu, an, notă)

# Proiectarea prin descompunere

---

- ▶ Se pleacă de la “mega-relații” ce conțin tot
- ▶ Se descompune în relații mai mici ce păstrează toate informațiile
- ▶ Se poate realiza automat
  - ▶ Mega-relații + *proprietăți ale datelor*
  - ▶ Descompunerea se realizează pe baza proprietăților
  - ▶ Setul final de relații satisface anumite *forme normale*
    - ▶ **Fără anomalii**
    - ▶ **Fără pierdere de informații**

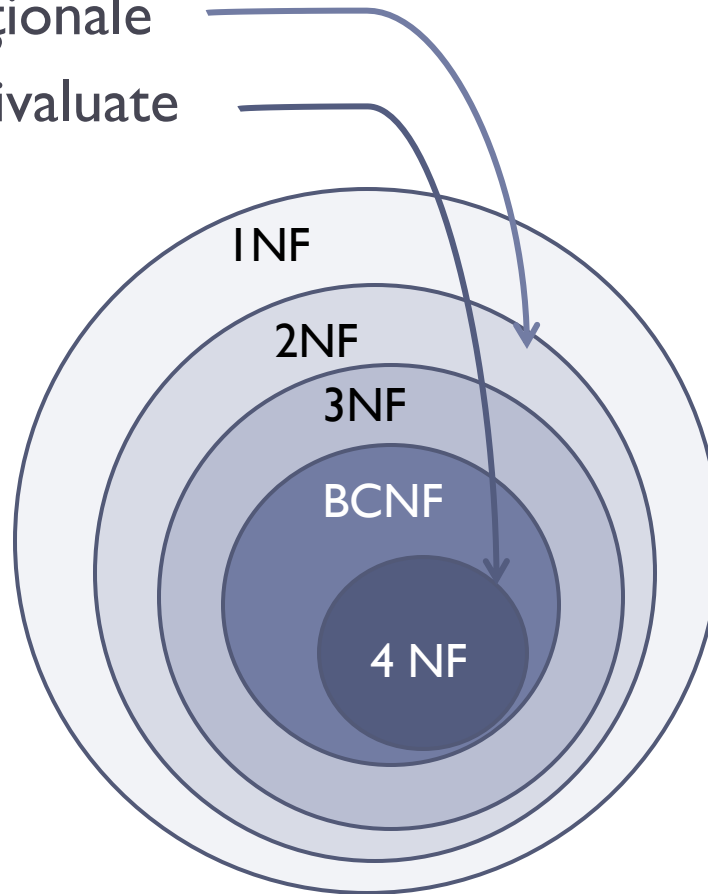


# Proprietăți și forme normale

- ▶ **Proprietăți**

- ▶ Dependențe funcționale
- ▶ Dependențe multivaluate

- ▶ **Forme normale**



# Dependențe funcționale

---

- ▶ Concepte folositoare pentru
  - ▶ Stocarea datelor – compresie
  - ▶ Optimizarea interogărilor

$X \rightarrow Y$  dacă

$$\forall t_1, t_2 \in r, t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

$r$  – relație peste mulțimea de attribute  $U$

$X, Y$  – submulțimi ale lui  $U$

*De ce funcțional?*

# Exemplu

## Dependențe funcționale

---

Absolvent(CNP, aNume, adresa,  
lCod, lNume, lOras, medie,  
prioritate)

Candidat(CNP, uNume, uOras, data,  
specializare)

- ▶ Valorile atributului **prioritate** sunt determinate de valorile atributului **medie**

*medie* → *prioritate*

Care sunt dependențele funcționale pentru relația Absolvent?

Dar pentru relația Candidat?

Ce constrângere este impusă de {CNP,uNume → data}

???

- 
- ▶  $R(A,B,C,D,E,F)$
  - ▶  $ABC \rightarrow D$
  - ▶  $DE \rightarrow F$
  - ▶ Fiecare din attributele  $A,B,C,E$  are cel mult 3 valori diferite.
  - ▶ Care este numărul maxim de valori diferite pe care îl poate lua  $E$ ? (3,9,27,81?)

# Dependențe funcționale

## Reguli de inferență

---

- ▶ Reflexivitatea (FD1)
  - ▶ (A1)
  - ▶ dependențe triviale
- ▶ Descompunerea (FD6)
  - ▶ A(21)
  - ▶ Se poate descompune și membrul stâng?
- ▶ Uniunea (FD5)
  - ▶ (A22)
- ▶ Tranzitivitatea (FD3)
  - ▶ (A3)
- ▶ Teorema de completitudine
  - ▶ o dependență funcțională este consecință a unei mulțimi de dependențe funcționale d.d. are demonstrație utilizând regulile de mai sus (Axiomele lui Armstrong)

# Dependențe funcționale și chei

---

## ▶ Dependențe funcționale (d.f.)

- ▶ Valorile unei submulțimi de attribute determină (identifică) valorile unei alte submulțimi de attribute
  - ▶ Formulate pe baza cunoașterii lumii reale
  - ▶ Toate instanțele relației trebuie să le satisfacă
  - ▶ *Se specifică un set minimal netrivial a.î. toate dependențele satisfăcute de relație se obțin ca și consecințe ale acestei mulțimi*

## ▶ Supercheie

- ▶ Valorile unei submulțimi de attribute identifică în mod unic un tuplu => valorile tuturor atributelor
- ▶ O cheie este submulțime minimală cu proprietatea de mai sus
- ▶ Garantează o relație fără duplicate

*Dependențele funcționale sunt o generalizare a noțiunii de cheie*

# Închideri

---

- ▶ *Închiderea unei mulțimi de d.f.  $\Sigma$  notată  $\Sigma^+$* 
  - ▶ Mulțimea d.f.  $\Sigma$  împreună cu toate d.f. consecințe din  $\Sigma$
- ▶ *Închiderea unei mulțimi de attribute  $X$  notată  $X^+$  relativ la un set de d.f.  $\Sigma$* 
  - ▶ Mulțimea tuturor atributelor  $B$  pentru care există  $X \rightarrow B \in \Sigma^+$
- ▶ O dependență funcțională  $X \rightarrow B$  este consecință a unei mulțimi de dependențe funcționale d.d.  $B \in X^+$

Algoritm de calcul a închiderii lui  $X$ ?

# Închideri și chei

---

- ▶ Date schema de relație  $R(U)$  și un set de d.f.  $\Sigma$  satisfăcute de  $R$ , submulțimea de attribute  $X$  este cheie pentru  $R$  d.d.  
 $X^+ = U$  și  $\forall X' \subset X, X'^+ \neq U$

- ▶ Exemplu

Abso1vent(CNP, aNume, adresa,  
1Cod, 1Nume, 1Oras, medie, prioritate)

CNP  $\rightarrow$  aNume, adresa, medie

medie  $\rightarrow$  prioritate

1Cod  $\rightarrow$  1Nume, 1Oras

Perechea {CNP, 1Cod} este cheie

Data o mulțime de d.f. cum putem determina toate cheile?



???

- 
- ▶  $R(A,B,C,D,E)$
  - ▶  $AB \rightarrow C$
  - ▶  $AE \rightarrow D$
  - ▶  $D \rightarrow B$
  - ▶ Care sunt cheile pt. R?

# Attribute (ne)prime

---

- ▶ **Atribut prim**

- ▶ Există o cheie care să-l conțină

- ▶ **Atribut neprim**

- ▶ Nu aparține nici unei chei

- ▶ **Exemplu**

- ▶  $R(A, B, C, D)$

- ▶  $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}.$

- ▶ *AB si BC sunt singurele chei cu privire la F, deci A, B, C sunt attribute prime*

- ▶ *D este atribut neprim.*

# Dependențe pline

---

- ▶ Fie dată o schemă de relație  $R$  cu multimea de atribute  $U$  și  $F$  o multime de dependente functionale. O dependență funcțională  $X \rightarrow B \in F^+$  ( $X \subset U$ ,  $B \in U$ ,  $B \notin X$ ) se numește o dependență plină a lui  $R$  (sau  $B$  este dependent plin de  $X$  sub  $F$ ), dacă nu există nici o submultime proprie  $X' \subset X$ , astfel încât  $X' \rightarrow B \in F^+$ .
- ▶ Exemplu
  - ▶  $R(A, B, C, D)$
  - ▶  $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$ .
  - ▶ *Toate dependențele din  $F$  sunt pline.*
  - ▶  *$AB \rightarrow D \in F^+$  nu este dependență plină*

# Atribut tranzitiv dependent

---

- ▶ Fie  $R$  o schema cu multimea de attribute  $U$  si  $F$  o multime de dependente functionale. Un atribut  $B$  din  $U$  se numeste *tranzitiv dependent* de  $X$  ( $X \subset U$ ,  $B \notin X$ ), daca exista  $Y \subset U$ , astfel incat:
  - ▶  $B \in U - Y$ ,
  - ▶  $X \rightarrow Y \in F^+$ ,
  - ▶  $Y \rightarrow B \in F^+$ ,
  - ▶  $Y \rightarrow X \notin F^+$ .

# 1NF

---

- ▶ O schemă de relație este în 1NF dacă domeniile de valori ale tuturor atributelor sunt elementare (indivizibile) deci diferite de multimi, de tuple de valori dintr-un anumit domeniu. În general numim valoare elementară o valoare pentru care în aplicații nu se utilizează părți ale ei

# 2NF

- ▶ O schema de relatie  $R$  situata in 1NF, impreuna cu o multime de dependente functionale  $F$  este in a doua forma normala daca orice atribut neprim din  $R$  este dependent plin de orice cheie a lui  $R$ .
- ▶ Obs: Orice relație ce nu are chei multivaluate este în 2NF.
- ▶ Exemplu
  - ▶  $R(A, B, C, D)$
  - ▶  $F = \{AB \rightarrow C, B \rightarrow D, BC \rightarrow A\}$ .
  - ▶  $AB$  si  $BC$  sunt singurele chei
  - ▶  $D$  este atribut neprim
  - ▶  $B \rightarrow D \in F^+$ , deci  $D$  nu este dependent plin nici de  $AB$ , nici de  $BC$ . In concluzie, aceasta schema impreuna cu  $F$  nu este in 2NF.

???

- 
- ▶ Absolventi(CNP, aNume, hobby)
  - ▶ Un Absolvent, identificat prin CNP, are un singur nume însă mai multe hobbyuri
  - ▶ Absolventi nu este în 2 NF. Ce anomalii apar din nerespectarea 2NF?
- 
- ▶ Olimpici(concurs,an,CNP,nume)
  - ▶ Într-un anumit an există un singur câștigător (olimpic) la un anumit concurs. Câștigătorul e identificat prin CNP și are asociat un singur nume.
  - ▶ Este Olimpici in 2NF?

# 3NF

- ▶ Schema de relatie  $R$  impreuna cu multimea de dependente functionale  $F$  este in forma a treia normala (notata 3NF) daca este in a doua forma normala si orice atribut neprim din  $R$  **NU** este tranzitiv dependent de nici o cheie a lui  $R$ .
- ▶ Exemplu
  - ▶  $R(O, S, C)$
  - ▶  $F = \{OS \rightarrow C, C \rightarrow O\}$
  - ▶  $OS$  si  $SC$  sunt chei.
  - ▶ toate attributele sunt prime, deci schema este in 2NF si 3NF.



???

- 
- ▶ Olimpici(concurs,an,CNP,nume)
  - ▶ Într-un anumit an există un singur câștigător (olimpic) la un anumit concurs. Câștigătorul e identificat prin CNP și are asociat un singur nume.
  - ▶ Olimpici nu este in 3NF. Ce probleme de inconsistență a datelor pot să apară?

???

- 
- ▶ Candidat(CNP,uNume,data,specializare)
  - ▶ Un absolvent identificat prin CNP poate candida la orice universitate (identificata prin uNume) la o singură specializare din cadrul acesteia, deci o singura data
  - ▶ Universitățile au date de aplicație care nu se suprapun
  - ▶ Este Candidat în 3NF relativ la regulile specificate mai sus?

# BCNF

---

- ▶ O schemă de relație  $R$  împreună cu o mulțime de dependențe funcționale  $F$  este în BCNF dacă este în 1NF și pentru orice dependență funcțională netrivială  $X \rightarrow A \in F^+$   $X$  este (super)cheie pentru  $R$
- ▶ Orice schemă de relație în BCNF este în 3NF
  - ▶ **Demonstrație?**
- ▶ Proiectarea unei scheme de BD în BCNF are la bază descompunerea:
  - ▶ Intrare: o mega-relație împreună cu un set de dependențe funcționale
  - ▶ ieșire: un set de relații în BCNF care în urma reasamblării produc informațiile originale

???

- 
- ▶ Candidat(CNP,uNume,data,specializare)
  - ▶ Un absolvent poate aplica la o universitate la o singură specializare
  - ▶ Universitățile au date de aplicație care nu se suprapun
  - ▶ Este Candidat în BCNF?

# Descompunerea schemelor de relatie

- ▶ Fie schema de relatie  $R[A_1, A_2, \dots, A_n]$ .
- ▶  $\rho = \{R_1, \dots, R_k\}$ , unde  $R_i[A_{i1}, \dots, A_{ihi}]$  este o *descompunere* a lui  $R$  dacă 
$$\bigcup_{i=1}^k \bigcup_{j=1}^{h_i} A_{ij} = \{A_1, \dots, A_n\}$$
- ▶  $\rho$  este o *descompunere de tip join fără pierdere* a lui  $R$  *cu privire la o mulțime de d.f.*  $F$ , dacă pentru orice relație  $r$  peste  $R$  ce satisface  $F$ , avem  $r = r[R_1] \bowtie \dots \bowtie r[R_k]$  – deci  $r$  se obține în urma joinului natural peste descompunerea  $\rho$ .

# Exemplu

## Descompunere

---

- ▶ Absolvent(CNP,aNume,adresa,ICod,INume,IOras,medie,prioritate)
- ▶  $\rho_1 = \{S_1(\text{CNP}, \text{aNume}, \text{adresa}, \underline{\text{ICod}}, \text{medie}, \text{prioritate}), S_2(\underline{\text{ICod}}, \text{INume}, \text{IOras})\}$
- ▶  $\rho_2 = \{S_1(\text{CNP}, \underline{\text{aNume}}, \text{adresa}, \text{ICod}, \underline{\text{INume}}, \text{IOras}), S_2(\underline{\text{aNume}}, \underline{\text{INume}}, \text{medie}, \text{prioritate})\}$
- ▶  $\rho_1$  - de tip join fără pierdere
- ▶  $\rho_2$  - NU e de tip join fără pierdere

# Descompuneri de tip join fără pierdere

## ► Teoremă

- Dacă  $\rho = (R_1, R_2)$  este o descompunere a lui  $R$  și  $F$  este o multime de d.f., atunci  $\rho$  este o descompunere join fără pierdere cu privire la  $F$  d.d.  $R_1 \cap R_2 \rightarrow R_1 - R_2 \in F^+$  sau  $R_1 \cap R_2 \rightarrow R_2 - R_1 \in F^+$ .

(prin intersectia schemelor se intelege intersectia multimilor de atribute)

## ► Exemplu

- $R(A, B, C)$
- $F = \{A \rightarrow B\}$ .
- $\rho_1 = (R_1(A, B), R_2(A, C))$
- $AB \cap AC = A, AB - AC = B, A \rightarrow B \in F^+$
- $\rho_1$  este de tip join fără pierdere
  
- $\rho_2 = (R_1(A, B), R_2(B, C))$ .
- $AB \cap BC = B, AB - BC = A, B \rightarrow A \notin F^+$ ,
- $AB \cap BC = B, BC - AB = C, B \rightarrow C \notin F^+$ ,
- $\rho_2$  nu este de tip join fără pierdere cu privire la  $F$ .

# Descompunere de tip join fara pierdere in BCNF

---

## ▶ Intrare:

- ▶ Schema de relatie R cu dependentele functionale F.

## ▶ Iesire:

- ▶ Descompunerea  $\rho = (R_1, \dots, R_k)$ , astfel incat  $\rho$  este de tip join fara pierdere cu privire la F si  $(R_i, F_i)$  este in BCNF  $\forall i = 1, k$ .

## ▶ Pasul 1.

- ▶  $\rho = R = R_1$
- ▶ Calculăm  $F^+$  și cheile necesare verificării formei BCNF

## ▶ Pasul 2.

- ▶ Fie  $R_i$  o schema de relatie din  $\rho$ , pentru care  $(R_i, F_i)$  nu este in BCNF.
- ▶ Exista  $X \rightarrow A \in F_i^+, A \notin X$  si  $X$  nu include o cheie.
- ▶ Construim  $S_1 = X \cup \{A\}, S_2 = R_i - A$
- ▶ Înlocuim  $R_i$  in  $\rho$  prin  $S_1, S_2$ .  $k = k + 1$ .
- ▶ Calculăm  $F_{S_1}^+ \cup F_{S_2}^+$  și cheile pt.  $S_1, S_2$  necesare verificării formei BCNF

## ▶ Pasul 3.

- ▶ Repetăm pasul 2, pana cand obținem toate  $(R_i, F_i), i = 1, k$  in BCNF.



# Exemplu

## Descompunere în BCNF

---

Abso1vent(CNP, aNume, adresa,  
1Cod, 1Nume, 1Oras, medie, prioritate)

CNP → aNume, adresa, medie

medie → prioritate

1Cod → 1Nume, 1Oras

{R1(1Cod, 1Nume, 1Oras),  
R2(medie, prioritate),  
R3(CNP, aNume, adresa, medie),  
R4(CNP, 1Cod)}

este o descompunere de tip join fără pierdere în BCNF

- Pentru o schemă de relație R pot exista mai multe descompuneri de tip join fără pierdere în BCNF?

# Garantează desc. în BCNF o schemă bună?

---

- ▶ Poate fi reconstruită relația originală?
- ▶ Elimină redundanța?
  - ▶ `Candidat(CNP, uNume, hobby)`
    - ▶ d.f.? NU
    - ▶ Chei? Toate attributele
    - ▶ BCNF? DA
    - ▶ Schemă bună? ...

# Dependențe multivaluate

---

## ► Reguli generatoare de tuple

$X \twoheadrightarrow Y$  dacă

$\forall t_1, t_2 \in r, t_1[X] = t_2[X], \text{ exista } t_3, t_4 \in r \text{ astfel încât}$

(i)  $t_3[X] = t_1[X], t_3[Y] = t_1[Y]$  și  $t_3[Z] = t_2[Z]$

(ii)  $t_4[X] = t_2[X], t_4[Y] = t_2[Y]$  și  $t_4[Z] = t_1[Z]$

$r$  – relație peste mulțimea de attribute  $U$

$X, Y$  – submulțimi ale lui  $U$

$Z = U - XY$

## ► Orice d.f. este d.mv.

# Exemplu

## Dependențe multivaluate

---

- ▶ `Candidat(CNP, uNume, hobby)`
- ▶ Cerințe:
  - ▶ Aceleași hobbyuri la toate univ
- ▶ **Regula corespunzătoare:**
  - ▶  **$CNP \twoheadrightarrow uNume$**

### ▶ Exemplu extins

- ▶ `Candidat(CNP, uNume, data, specializare, hobby)`
- ▶ Cerințe:
  - ▶ Hobbyurile sunt introduse selectiv în funcție de universitate
  - ▶ Un Absolvent candidează într-o singură zi la o anumită universitate
  - ▶ Un Absolvent poate aplica la mai multe specializări (hobbyurile declarate la o univ. trebuie să fie vizibile la specializarile de la acea univ.)
- ▶ **Regulile corespunzătoare:**
  - ▶  **$CNP, uNume \rightarrow data$**
  - ▶  **$CNP, uNume, data \twoheadrightarrow specializare$**

???

- 
- ▶ Fie  $R(A,B,C)$  și  $A \rightarrow \rightarrow B$
  - ▶ A ia cel puțin 3 valori diferite iar fiecare valoare a lui A este asociată cu cel puțin 4 valori diferite pentru B și cel puțin 5 valori diferite pentru C.
  - ▶ Care este numărul minim de tuple în R?

# Dependențe multivaluate

## Reguli

---

- ▶ Dependențe triviale
  - ▶ Reflexivitate (MVDI)
  - ▶  $X \twoheadrightarrow Y$  unde  $XY=U$
- ▶ Complementariere (MVD0)
- ▶ Tranzitivitatea (!=d.f.)
- ▶ Intersecția

# 4NF

---

- ▶ O schemă de relație  $R$  și o mulțime de dependențe multivaluate  $D$  este în 4NF dacă este în 1NF și pentru orice dependență multivaluată netrivială  $X \twoheadrightarrow A \in D^+$   $X$  este (super)cheie pentru  $R$
- ▶ Orice schemă de relație în 4NF este în BCNF
- ▶ Proiectarea unei scheme de BD în 4NF are la bază descompunerea:
  - ▶ Intrare: o mega-relație împreună cu un set de dependențe funcționale și multivaluate
  - ▶ Ieșire: un set de relații în 4NF care în urma reasamblării produc informațiile originale

# Descompunere de tip join fara pierdere in 4NF

---

## ▶ Intrare:

- ▶ Schema de relatie R cu dependentele functionale F și dependențele multivaluate MV

## ▶ Iesire:

- ▶ Descompunerea lui  $\rho = (R_1, \dots, R_k)$ , astfel incat  $\rho$  este de tip join fara pierdere cu privire la F si  $(R_i, F_i, MV_i)$  este in 4NF  $\forall i = 1, k$ .

## ▶ Pasul 1.

- ▶  $\rho = R = R_1$
- ▶ Calculăm  $M = \{F^+, MV^+\}$  și cheile necesare verificării formei 4NF

## ▶ Pasul 2.

- ▶ Fie  $R_i$  o schema de relatie din  $\rho$ , pentru care  $(R_i, F_i, MV_i)$  nu este in 4NF.
- ▶ Exista  $X \twoheadrightarrow A \in M$  netrivială si X nu include o cheie.
- ▶ Construim  $S_1 = X \cup \{A\}$ ,  $S_2 = R_i - A$
- ▶ Înlocuim  $R_i$  in  $\rho$  prin  $S_1, S_2$ .  $k = k + 1$ .
- ▶ Calculăm d.mv și cheile pt.  $S_1, S_2$  necesare verificării formei 4NF

## ▶ Pasul 3.

- ▶ Repetam pasul 2, pana cand obtinem toate  $(R_i, F_i), i = 1, k$  in NF.



# Exemplu

## Descompunere în 4NF

---

- ▶  $\text{Candidat}(\text{CNP}, \text{uNume}, \text{hobby})$
- ▶  $\text{CNP} \twoheadrightarrow \text{uNume}$
- ▶  $\rho = \{A_1(\text{CNP}, \text{uNume}), A_2(\text{CNP}, \text{hobby})\}$  este descompunere în 4NF de tip join fără pierdere
- ▶  $u * h, u + h$

### ▶ Exemplu extins

- ▶  $\text{Candidat}(\text{CNP}, \text{uNume}, \text{data}, \text{specializare}, \text{hobby})$
- ▶  $\text{CNP}, \text{uNume} \rightarrow \text{data}$
- ▶  $\text{CNP}, \text{uNume}, \text{data} \twoheadrightarrow \text{specializare}$
- ▶  $\rho = \{A_1(\text{CNP}, \text{uNume}, \text{data}), A_2(\text{CNP}, \text{uNume}, \text{specializare}), A_3(\text{CNP}, \text{uNume}, \text{hobby})\}$   
este în 4NF de tip join fără pierdere

# Neajunsuri ale normalizării

## Exemplu 1

---

- ▶  $\text{Candidat}(\text{CNP}, \text{uNume}, \text{data}, \text{specializare})$
- ▶  $\text{CNP}, \text{uNume} \rightarrow \text{data}, \text{specializare}$
- ▶  $\text{data} \rightarrow \text{uNume}$
  
- ▶  $\{\text{A1}(\text{data}, \text{uNume}), \text{A2}(\text{CNP}, \text{data}, \text{specializare})\}$  în 4NF este o schemă mai bună?

# Neajunsuri ale normalizării

## Exemplu 2

---

- ▶  $\text{Absolvent}(\text{CNP}, \text{INume}, \text{medie}, \text{prioritate})$
- ▶  $\text{CNP} \rightarrow \text{medie}$
- ▶  $\text{medie} \rightarrow \text{prioritate}$
- ▶  $\text{CNP} \rightarrow \text{prioritate}$
  
- ▶  $\{S1(\text{CNP}, \text{prioritate}), S2(\text{CNP}, \text{medie}), S3(\text{CNP}, \text{INume})\}$  în 4NF este o schemă bună?

# Neajunsuri ale normalizării

---

- ▶ Supra-descompunere
- ▶ Interogări supra-încărcate
  
- ▶ Ca soluție se poate aplica denormalizarea

# Bibliografie

---

- ▶ Victor Felea: *Baze de date relationale. Dependente*. Editura Universitatii “Al.I.Cuza” Iasi, 1996
- ▶ Hector Garcia-Molina, Jeff Ullman, **Jennifer Widom**: *Database Systems: The Complete Book*, Prentice Hall; 2nd edition (June 15, 2008)
- ▶ Unele exemple sunt **adaptate** dupa cursul public de Baze de date de la Stanford (autor Jennifer Widom)