

BAZE DE DATE

Proiectarea bazelor de date relaționale

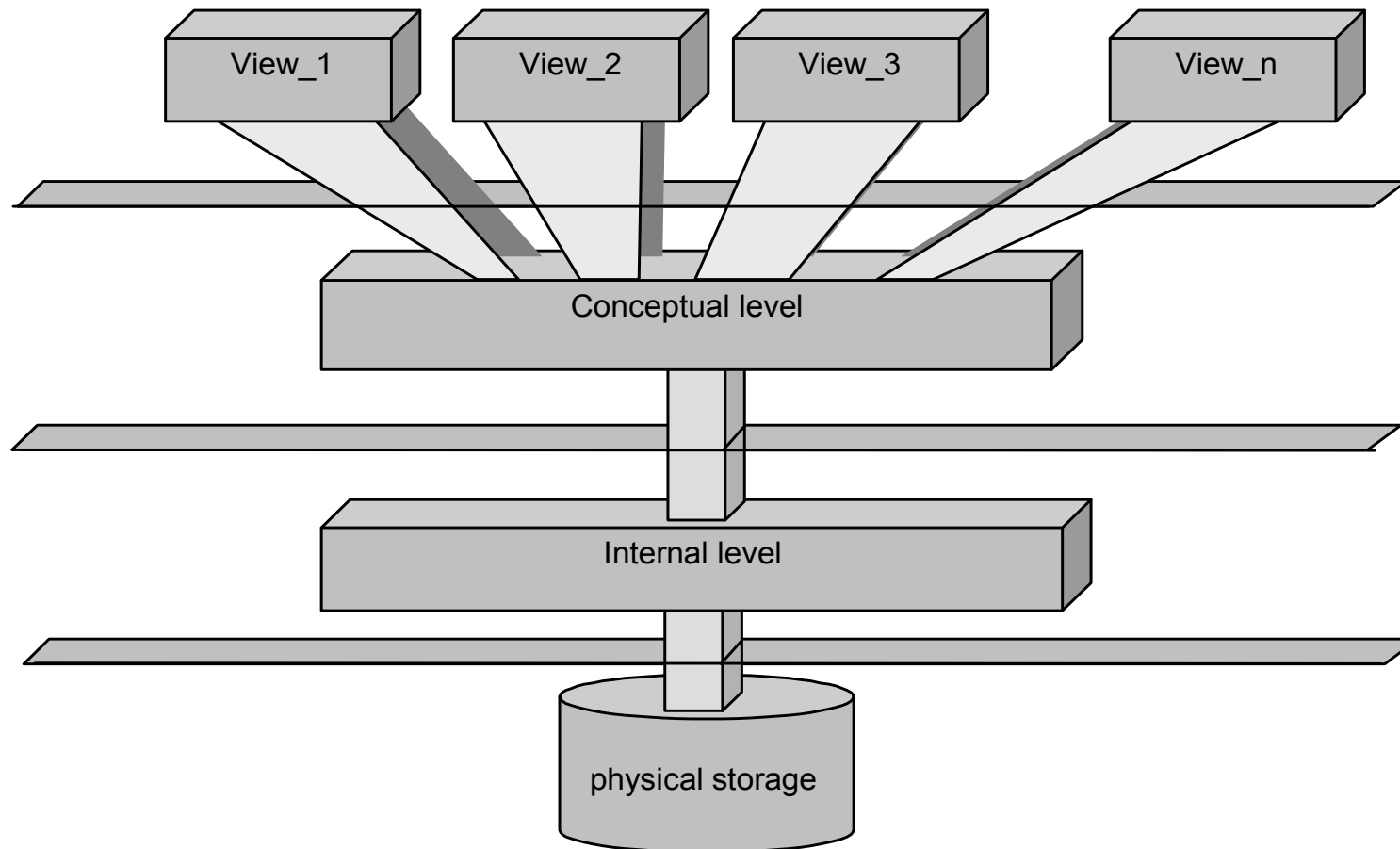
Normalizare
@FII (2011-2012)

prezentat de Mihaela Elena Breabăn

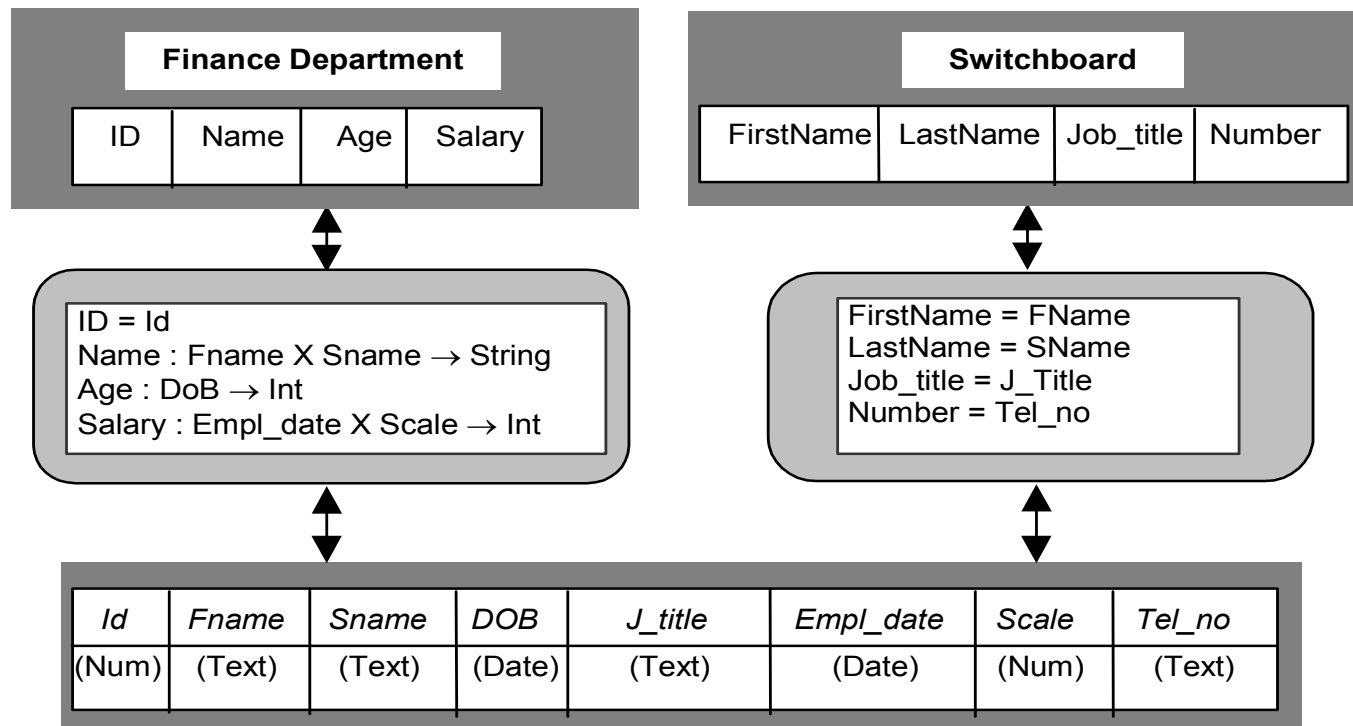
Tematică curs

- ▶ Proiectarea bazelor de date relaționale
 - ▶ Normalizare și denormalizare
 - ▶ Modelul entitate-asociere, diagrame UML
 - ▶ Constrângeri și declanșatoare
 - ▶ Indecși
- ▶ View-uri
- ▶ Procesarea interogărilor
- ▶ Managementul tranzacțiilor
- ▶ OLAP, Baze de date distribuite, NoSQL, Data Mining

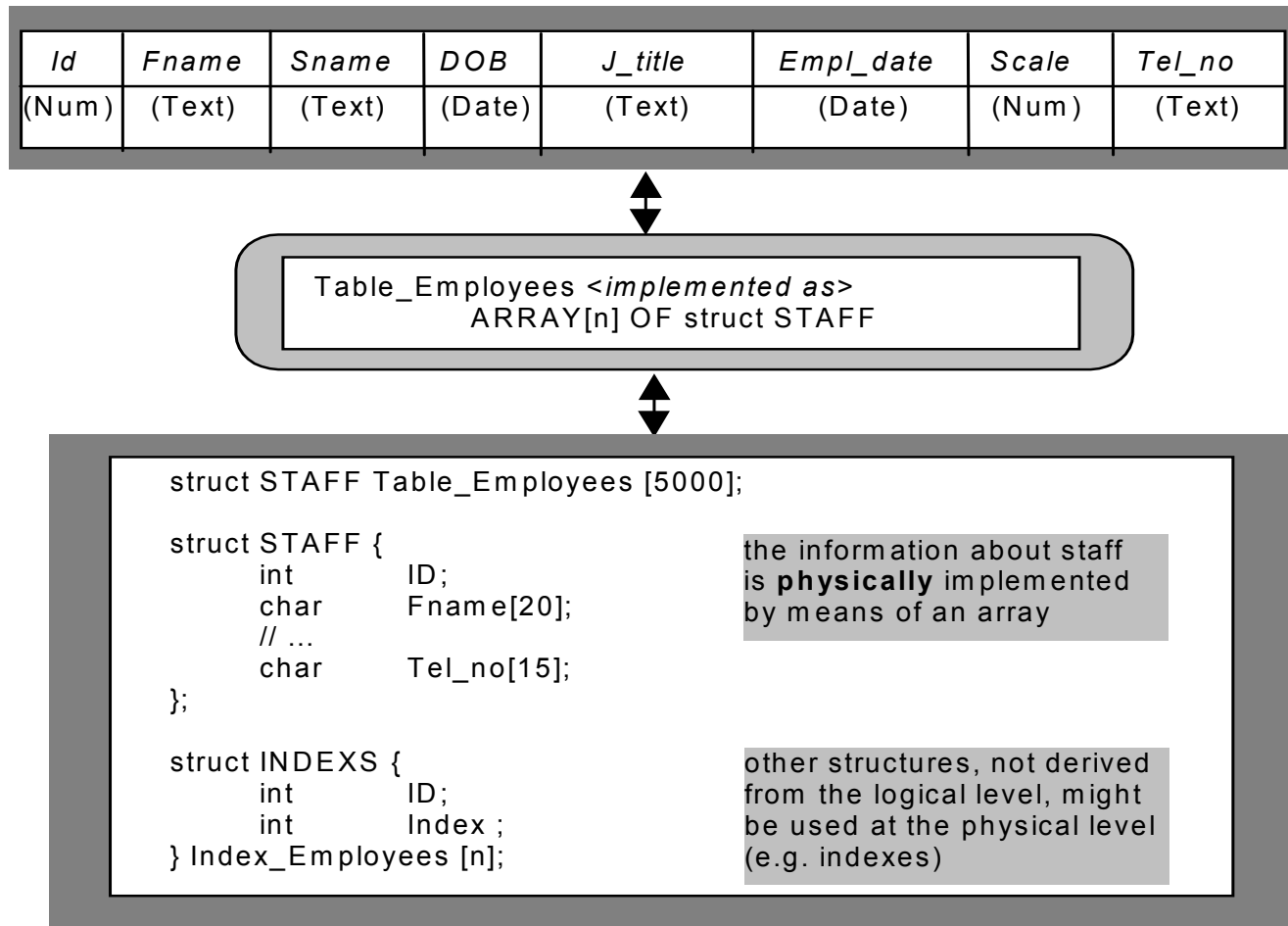
Arhitectura pe 3 nivele ANSI-SPARC



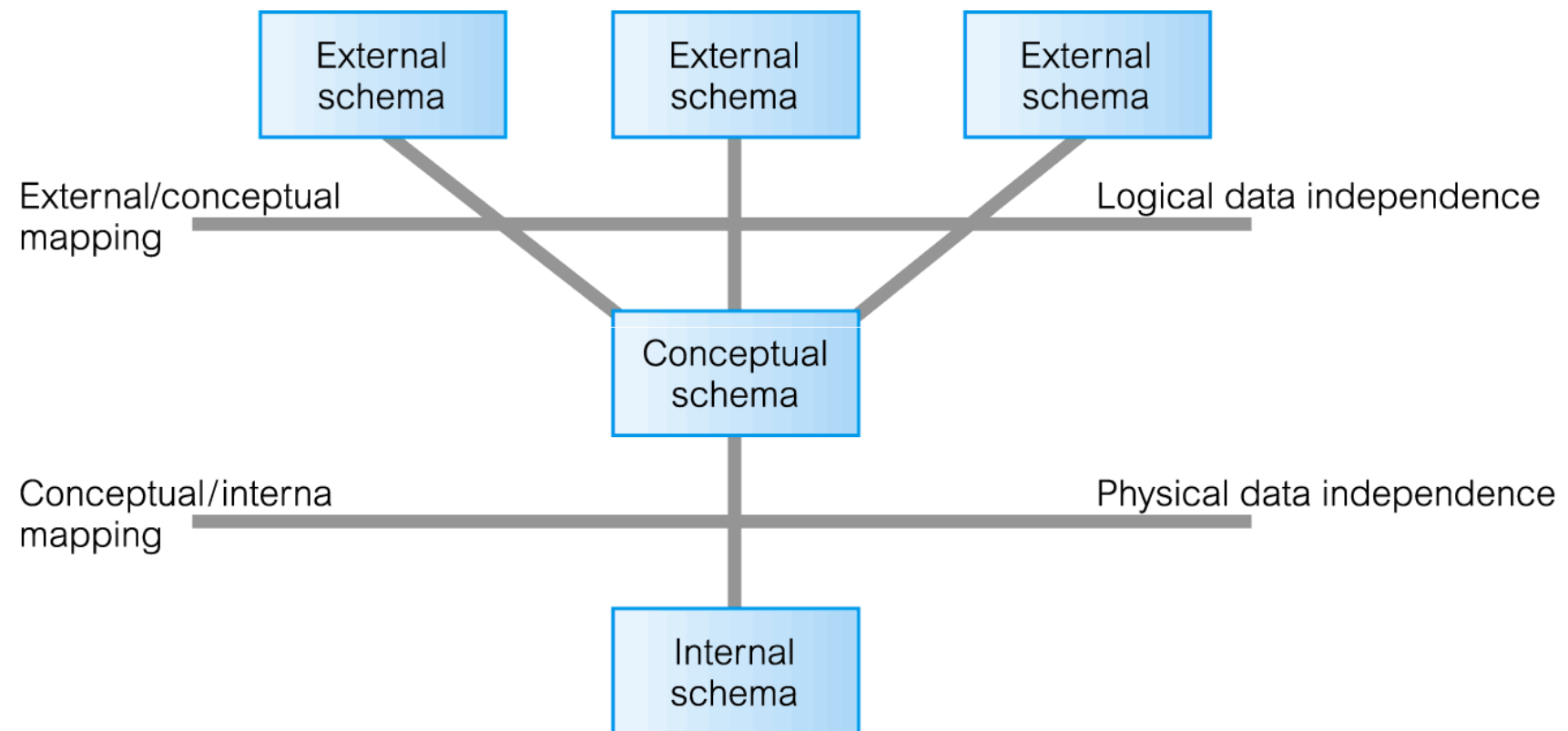
Mapare nivel extern/conceptual



Mapare nivel conceptual/intern

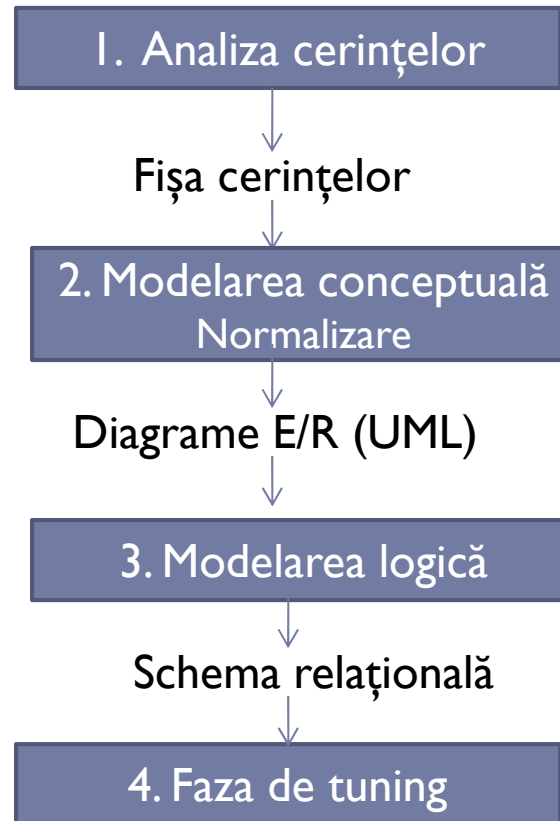


Arhitectura pe 3 nivele Scheme



Proiectarea unei BD

Metodologie



Normalizare

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

Proiectarea schemei

- ▶ De obicei mai multe variante de proiectare
 - ▶ Unele sunt (mult) mai bune decât altele
 - ▶ Cum alegem?
-
- ▶ Teorie pentru proiectarea bazelor de data relaționale cu fundamente în algebra relațională

Exemplu

Schemă cu anomalii

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

Aplicatie(CNP, sNume, uNume, lIiceu, 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 uuple sunt necesare a fi inserate în relația Aplicatie 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 aplicat
 - ▶ Liceele de la care provin candidații (și orașele)
 - ▶ Hobby-urile candidaților

Student(CNP, sNume)

Aplicatie(CNP, uNume)

Liceu(CNP, codLiceu)

LocatieLiceu(codLiceu, lNume, lOraș)

Hobbies(CNP, hobby)

Quiz

- ▶ Informații cu privire la cursurile luate de studenti
 - ▶ 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

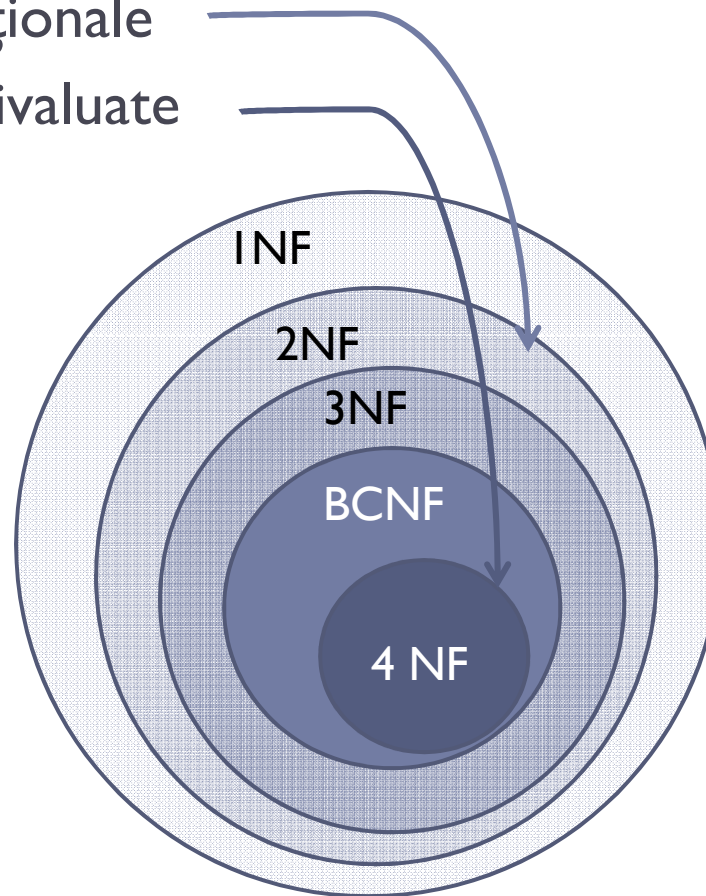
- ▶ Se pleacă de la “mega-relații” ce conțin tot
- ▶ Se descompune în relații mai mici ce păstrează toate informațiile
- ▶ Se poate realiza automat
 - ▶ Mega-relații + *proprietăți ale datelor*
 - ▶ Descompunerea se realizează pe baza proprietăților
 - ▶ Setul final de relații satisface anumite *forme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

Proprietăți și forme normale

- ▶ **Proprietăți**

- ▶ Dependențe funcționale
- ▶ Dependențe multivaluate

- ▶ **Forme normale**



Dependențe funcționale

- ▶ Concepte folosite 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

Student(CNP, sNume, adresa,
lCod, lNume, lOras, medie,
prioritate)

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

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

medie → *prioritate*

Care sunt dependențele funcționale pentru relația Student?

Dar pentru relația Aplicatie?

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

Quiz

- ▶ $R(A,B,C,D,E)$
- ▶ $AB \rightarrow C$
- ▶ $CD \rightarrow E$
- ▶ Fiecare din attributele A,B,D 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 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 a acestei mulțimi*
- ▶ **Chei**
 - ▶ Valorile unei submulțimi de attribute determină valorile tuturor atributelor - supercheie
 - ▶ O cheie este submulțime minimală cu proprietatea de mai sus
 - ▶ Relație fără duplicate
- ▶ **Dependențele funcționale sunt o generalizare a noțiunii de cheie**

Închideri

- ▶ Închiderea unei mulțimi de d.f. Σ notată Σ^+
 - ▶ Mulțimea d.f. Σ împreună cu toate d.f. consecințe din Σ
- ▶ Închiderea unei mulțimi de atribute X notată X^+ relativ la un set de d.f. Σ
 - ▶ 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 relația R definită peste mulțimea de attribute U și un set de d.f. Σ satisfăcute de R , submulțimea de attribute X este cheie pentru R d.d. $X^+ = U$ și $\forall X' \subset X, X'^+ \neq U$

- ▶ Exemplu

Student(CNP, sNume, adresa,
lCod, lNume, lOras, medie, prioritate)

CNP \rightarrow sNume, adresa, medie

medie \rightarrow prioritate

lCod \rightarrow lNume, lOras

Perechea {CNP, lCod} este cheie

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

Quiz

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

Atribute (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 și BC sunt singurele chei cu privire la F, deci A, B, C sunt atribute 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 mulțimi, de mulțimi 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.

Quiz

- ▶ `Studenti(CNP, sNume, hobby)`
- ▶ Un student are un singur nume însă mai multe hobbyuri
- ▶ `Studenti` 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 numele.
- ▶ Este `Olimpici` în 2NF?

3NF

- ▶ Schema de relatie R impreuna cu F se spune ca 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.

Quiz

- ▶ 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 numele.
- ▶ Olimpici nu este in 3NF. Ce probleme de inconsistență a datelor pot să apară?

Quiz

- ▶ Aplicație(CNP,uNume,data,specializare)
- ▶ Un student poate aplica la o universitate o singură dată și la o singură specializare
- ▶ Universitățile au date de aplicație care nu se suprapun
- ▶ Este Aplicație în 3NF relativ la regulile specificate mai sus?

BCNF

- ▶ O schemă de relație R împreună cu o mulțime de dependențe funcționale D este în BCNF dacă pentru orice dependență funcțională netrivială $X \rightarrow A \in D^+$ X este (super)cheie pentru R
- ▶ Orice schemă de relație în BCNF este în 3NF
- ▶ 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

Quiz

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

Descompunerea schemelor de relatie

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

Exemplu

Descompunere

- ▶ Student(CNP,sNume,adresa,ICod,INume,IOras,medie,prioritate)
- ▶ $\rho_1 = \{S_1(\text{CNP}, s\text{Nume}, \text{adresa}, \underline{I\text{Cod}}, \text{medie}, \text{prioritate}), S_2(\underline{I\text{Cod}}, I\text{Nume}, I\text{Oras})\}$
- ▶ $\rho_2 = \{S_1(\text{CNP}, \underline{s\text{Nume}}, \text{adresa}, I\text{Cod}, \underline{I\text{Nume}}, I\text{Oras}), S_2(\underline{s\text{Nume}}, I\text{Nume}, \text{medie}, \text{prioritate})\}$
- ▶ ρ_1 - de tip join fără pierdere
- ▶ ρ_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 ρ 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 $R_1 \cap R_2 \rightarrow R_2 - R_1 \in F^+$.

► Exemplu

- $R(A, B, C)$
- $F = \{A \rightarrow B\}$.
- $\rho_1 = (R_1(A, B), R_2(A, C))$
- $AB \cap AC = A, AB - AC = B, A \rightarrow B \in F^+$
- ρ_1 este de tip join fără pierdere

- $\rho_2 = (R_1(A, B), R_2(B, C))$.
- $AB \cap BC = B, AB - BC = A, B \rightarrow A \notin F^+,$
- $AB \cap BC = B, BC - AB = C, B \rightarrow C \notin F^+,$
- ρ_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 lui $\rho = (R_1, \dots, R_k)$, astfel incat ρ 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 ρ , 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 ρ 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.**

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

Exemplu

Descompunere în BCNF

Student(CNP, sNume, adresa,
lCod, lNume, lOras, medie, prioritate)

CNP \rightarrow sNume, adresa, medie

medie \rightarrow prioritate

lCod \rightarrow lNume, lOras

{R1(lCod, lNume, lOras),
R2(medie, prioritate),
R3(CNP, sNume, adresa, medie),
R4(CNP, lCod)}

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?
 - ▶ `Aplicatie(CNP, uNume, hobby)`
 - ▶ d.f.? NU
 - ▶ Chei? Toate attributele
 - ▶ BCNF? DA
 - ▶ Schemă bună? ...

Dependențe multivaluate

► Reguli generatoare de uple

$X \twoheadrightarrow Y$ dacă

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

(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 atribute U

X, Y – submulțimi ale lui U

$Z = U - XY$

► Orice d.f. este d.mv.

Exemplu

Dependențe multivaluate

- ▶ `Aplicatie(CNP, uNume, hobby)`

- ▶ Cerințe:

 - ▶ Aceleași hobbyuri la toate univ

- ▶ Regula corespunzătoare:

 - ▶ $CNP \twoheadrightarrow uNume$

▶ Exemplu extins

- ▶ `Aplicatie(CNP, uNume, data, specializare, hobby)`

- ▶ Cerințe:

 - ▶ Hobbyurile sunt introduse selectiv în funcție de universitate

 - ▶ Un student aplică într-o singură zi la o anumită universitate

 - ▶ Un student poate aplica la mai multe specializări

- ▶ Regulile corespunzătoare:

 - ▶ $CNP, uNume \rightarrow data$

 - ▶ $CNP, uNume, data \twoheadrightarrow specializare$

Quiz

- ▶ 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 upe 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 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 ρ 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 ρ , 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 ρ 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

- ▶ $\text{Aplicatie}(\text{CNP}, \text{uNume}, \text{hobby})$
- ▶ $\text{CNP} \rightarrow \text{uNume}$
- ▶ $\rho = \{A_1(\text{CNP}, \text{uNume}), A_2(\text{CNP}, \text{hobby})\}$ este descompunere în 4NF de tip join fără pierdere
- ▶ $u * h, u + h$
- ▶ **Exemplu extins**
 - ▶ $\text{Aplicatie}(\text{CNP}, \text{uNume}, \text{data}, \text{specializare}, \text{hobby})$
 - ▶ $\text{CNP}, \text{uNume} \rightarrow \text{data}$
 - ▶ $\text{CNP}, \text{uNume}, \text{data} \rightarrow \text{specializare}$
 - ▶ $\rho = \{A_1(\text{CNP}, \text{uNume}, \text{data}), A_2(\text{CNP}, \text{uNume}, \text{specializare}), A_3(\text{CNP}, \text{uNume}, \text{hobby})\}$
este în 4NF de tip join fără pierdere

Neajunsuri ale normalizării

Exemplu 1

- ▶ Aplicație(CNP,uNume,data,specializare)
- ▶ CNP,uNume→data,specializare
- ▶ data→uNume

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

Neajunsuri ale normalizării

Exemplu 2

- ▶ Student(CNP,INume,medie,prioritate)
 - ▶ CNP→medie
 - ▶ medie →prioritate
 - ▶ CNP →prioritate
-
- ▶ {S1(CNP,prioritate),S2(CNP,medie),S3(CNP,INume)} în 4NF este o schemă bună?

Neajunsuri ale normalizării

- ▶ Supra-descompunere
- ▶ Interogări supra-încărcate

- ▶ Ca soluție se poate aplica denormalizarea

Bibliografie

- ▶ V.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)