
Normalizarea Bazelor de Date Relationale (partea 1)

Ioana Ciuciu

oana@cs.ubbcluj.ro

<http://www.cs.ubbcluj.ro/~oana/>

Planificare

Saptamana	Curs	Seminar	Laborator
S1	1. Concepte fundamentale ale bazelor de date	1. Modelul Entitate-Relatie. Modelul relational	1. Modelarea unei BD in modelul ER si implementarea ei in SQL Server
S2	2. Modelare conceptuala		
S3	3. Modelul relational de organizare a bazelor de date	2. Limbajul SQL – definirea si actualizarea datelor	2. Interogari SQL
S4	4. Gestiunea bazelor de date relationale cu limbajul SQL		
S5-6	5-6. Dependente functionale, forme normale	3. Limbajul SQL – regasirea datelor	3. Interogari SQL avansate
S7	7. Interogarea bazelor de date relationale cu operatori din algebra relationala	4. Proceduri stocate	
S8	8. Structura fizica a bazelor de date relationale		
S9-10-11	9-10-11. Indeksi. Arbori B. Fisiere cu acces direct	5. View-uri. Functii definite de utilizator. Executie dinamica	4. Proceduri stocate. View. Trigger
S12	12. Evaluarea interogarilor in bazele de date relationale	6. Formele normale ale unei relatii. Indeksi	
S13	13. Extensii ale modelului relational si baze de date NoSQL	7. Probleme	Examen practic
S14	14. Aplicatii		

Planul cursului

▶ Curs 5:

- ▶ Normalizarea bazelor de date relationale
- ▶ Prima forma normala (1NF)
- ▶ A 2-a forma normala (2NF)
- ▶ A 3-a forma normala (3NF)

▶ Curs 6:

- ▶ Forma normala Boyce-Codd (BCNF)
- ▶ A 4-a forma normala (4NF)
- ▶ A 5-a forma normala (5NF)



Normalizarea bazelor de date

- ▶ **Normalizarea bazelor de date** este procesul organizarii relatiilor (si al atributelor) unei baze de date relationale cu scopul de a **reduce redundanta** si de a **imbunatati integritatea datelor**
- ▶ Normalizarea garanteaza faptul ca *dependentele* intre attribute sunt corect impuse de catre constrangerile de integritate ale bazei de date
- ▶ Normalizarea se realizeaza prin procesul de *descompunere*
- ▶ **Scopul:** acela de a face posibile actualizarea si interogarea bazei de date intr-o maniera cat mai simpla



Normalizarea bazelor de date

► De ce este necesara normalizarea?

► Fie urmatoarea relatie:

PRODUCT	CLIENT	ADDRESS	QUANTITY	AMOUNT
Lotion	Martin	Paris	10	250
Polish	Martin	Paris	250	2500
Cream	Martin	Paris	20	1000
Lotion	John	London	30	750
Cream	Dupont	Lyon	20	1000
Lotion	Dubois	Paris	20	500

► Singura cheie posibila pentru relatie este: PRODUCT-CLIENT



Normalizarea bazelor de date

► De ce este necesara normalizarea?

- O astfel de relatie, chiar daca corecta din punctul de vedere al unui SGBD, va pune probleme de mentenanta si ar putea determina anomalii
 - Adresa clientului se repeta pentru fiecare comanda
 - Actualizarea unei adrese impune accesarea BD de mai multe ori

- ☐ Anomalii posibile

Exemplu: La inserarea urmatoarei linii

<Cream, John, New-York, 10, 500>

Sistemul nu va detecta faptul ca John are doua adrese (ceea ce nu este realist)



Normalizarea bazelor de date

► De ce este necesara normalizarea?

- O astfel de relatie, chiar daca corecta din punctul de vedere al unui SGBD, va pune probleme de mentenanta si ar putea determina anomalii
 - Adresa clientului se repeta pentru fiecare comanda
 - Actualizarea unei adrese impune accesarea BD de mai multe ori
 - Anomalii posibile

Exemplu: La inserarea urmatoarei linii

<Cream, John, New-York, 10, 500>

Sistemul nu va detecta faptul ca John are doua adrese (ceea ce nu este realist)

PRODUCT	CLIENT	ADDRESS	QUANTITY	AMOUNT
Lotion	Martin	Paris	10	250
Polish	Martin	Paris	250	2500
Cream	Martin	Paris	20	1000
Lotion	John	London	30	750
Cream	Dupont	Lyon	20	1000
Lotion	Dubois	Paris	20	500
Cream	John	New-York	10	500

Normalizarea bazelor de date

► De ce este necesara normalizarea?

- O astfel de relatie, chiar daca corecta din punctul de vedere al unui SGBD, va pune probleme de mentenanta si ar putea determina anomalii
 - Adresa clientului se repeta pentru fiecare comanda
 - Actualizarea unei adrese impune accesarea BD de mai multe ori

- ☐ Anomalii posibile

Exemplu: La inserarea urmatoarei linii

<Cream, John, New-York, 10, 500>

Sistemul nu va detecta faptul ca John are doua adrese (ceea ce nu este realist)

- ☐ Daca, de ex. ,Dupont isi anuleaza comanda, numele si adresa sa sunt pierdute

PRODUCT	CLIENT	ADDRESS	QUANTITY	AMOUNT
Lotion	Martin	Paris	10	250
Polish	Martin	Paris	250	2500
Cream	Martin	Paris	20	1000
Lotion	John	London	30	750
Cream	Dupont	Lyon	20	1000
Lotion	Dubois	Paris	20	500

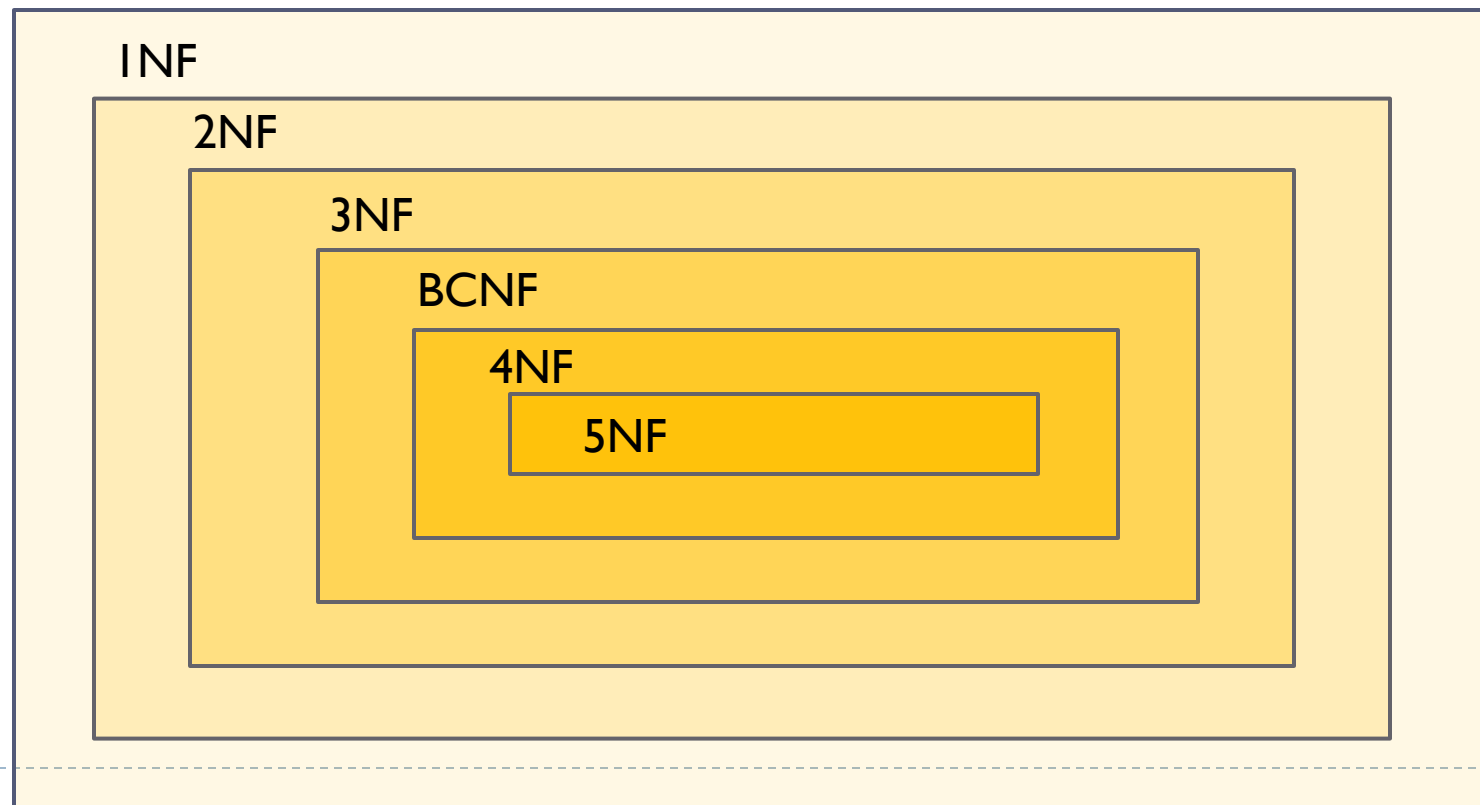
Normalizarea bazelor de date

- ▶ Se cunosc mai multe forme normale pentru relatii
- ▶ Cele mai cunoscute sunt: **1NF, 2NF, 3NF, BCNF, 4NF, 5NF**

Codd

Boyce-Codd

Fagin



Incluziuni pentru relatiile aflate in diferite forme normale

Prima forma normala (1NF)

Notiuni Preliminare

- Pentru **descompunerea** unei relatii se foloseste operatorul de **proiectie**

Fie $R[A_1, A_2, \dots, A_n]$ o relatie si $\alpha = \{A_{i_1}, A_{i_2}, \dots, A_{i_p}\}$ un set de attribute, unde $\alpha \subset \{A_1, A_2, \dots, A_n\}$. Prin proiectia relatiei R pe α se intelege relatia:

Unde:

$$R' \left[A_{i_1}, A_{i_2}, \dots, A_{i_p} \right] = \Pi_{\alpha}(R) = \Pi_{\{A_{i_1}, A_{i_2}, \dots, A_{i_p}\}}(R) = \{r[\alpha] \mid r \in R\}$$

$$\forall r = (a_1, a_2, \dots, a_n) \in R \Rightarrow \Pi_{\alpha}(r) = r[\alpha] = (a_{i_1}, a_{i_2}, \dots, a_{i_p}) \in R'$$

- Pentru **compunerea** relatiilor se foloseste operatorul de **join natural**

Fie $R[\alpha, \beta], S[\beta, \gamma]$ doua relatii peste multimile de attribute precizate, $\alpha \cap \gamma = \emptyset$

Prin joinul natural al relatiilor R si S se intelege relatia:

$$R * S[\alpha, \beta, \gamma] = \left\{ \left(\Pi_{\alpha}(r), \Pi_{\beta}(r), \Pi_{\gamma}(s) \right) \mid r \in R, s \in S \text{ and } \Pi_{\beta}(r) = \Pi_{\beta}(s) \right\}$$



Prima forma normala (1NF)

Notiuni Preliminare

- ▶ O relatie R se poate descompune in mai multe relatii noi R_1, R_2, \dots, R_n . Aceasta descompunere este **buna** daca $R = R_1 * R_2 * \dots * R_n$
- ▶ Deci datele din R se pot regasi din datele memorate in relatiile R_1, R_2, \dots, R_n ; **nu apar si nu se pierde date** cu aceste operatii de compunere (join natural)



Prima forma normala (1NF)

Notiuni Preliminare

- ▶ Exemplu de descompunere care **nu este buna**
- ▶ Fie relatia ContracteStudiu(Student, Teacher, Discipline), si doua noi relatii obtinute prin proiectia acestei relatii:
CS(Student, Teacher) and T(Teacher, Discipline)

Presupunem ca in relatia initiala avem urmatoarele valori:

Student	Teacher	Discipline
S1	T1	D1
S2	T2	D2
S1	T2	D3



Prima forma normala (1NF)

Notiuni Preliminare

Folosind definitia proiectiei, se obtin urmatoarele valori pentru cele doua relatii obtinute din R:

Student	Teacher
S1	T1
S2	T2
S1	T2

Teacher	Discipline
T1	D1
T2	D2
T2	D3

Si pentru joinul natural al acestor relatii (SC^*T)

Student	Teacher	Discipline
S1	T1	D1
S2	T2	D2
S2	T2	D3
S1	T2	D2
S1	T2	D3

Se observa ca in relatia SC^*T se obtin inregistrari suplimentare fata de relatia initiala, deci descompunerea sugerata **nu este buna**

Prima forma normala (1NF)

Notiuni Preliminare

- ▶ **Atribut simplu:** un atribut oarecare din relatie
- ▶ **Atribut compus:** o multime de attribute (cu cel putin doua attribute) din relatie
- ▶ **Atribut repetitiv:** attribute (simple sau compuse) care iau mai multe valori (vectori) pentru o inregistrare din relatie
- ▶ **Atribut atomic:** atribut care nu se descompune in alte elemente
- ▶ **Atribut scalar:** atribut care nu se poate memora in tablouri



Prima forma normala (1NF)

O relatie este in prima forma normala (1NF) daca ea contine doar attribute atomice si nu contine attribute repetitive

► **Exemplu:**

Enroll(Sid, Sname, Cname)

Obiectiv: gasiti toti studentii care urmeaza cursul de C (SQL+Application program)

Sid	Sname	Cname
1	A	C,C++
2	B	C,Java,C#
...

Prima forma normala (1NF)

O relatie este in prima forma normala (1NF) daca ea contine doar attribute atomice si nu contine attribute repetitive

► Exemplu:

Enroll(Sid, Sname, Cname)

Obiectiv: gasiti toti studentii care urmeaza cursul de C (SQL+Application program)

Solutia??

Sid	Sname	Cname
1	A	C,C++
2	B	C, Java, C#
...

Sid	Sname	Cname
1	A	C
1	A	C++
2	B	C
2	B	Java
2	B	C#

Prima forma normala (1NF)

O relatie este in prima forma normala (1NF) daca ea contine doar attribute atomice si nu contine attribute repetitive

► Exemplu:

Enroll(Sid, Sname, Cname)

Obiectiv: gasiti toti studentii care urmeaza cursul de C (SQL+Application program)

Solutia??

REDUNDANTA

Sid	Sname	Cname
1	A	C,C++
2	B	C, Java, C#
...

Sid	Sname	Cname
1	A	C
1	A	C++
2	B	C
2	B	Java
2	B	C#

Prima forma normala (1NF)

O relatie este in prima forma normala (1NF) daca ea contine doar attribute atomice si nu contine attribute repetitive

► Exemplu:

Enroll(Sid, Sname, Cname)

Obiectiv: gasiti toti studentii care urmeaza cursul de C (SQL+Application program)

Solutia2: descompunerea tablei in doua tabele

Sid	Sname	Cname
1	A	C,C++
2	B	C,Java,C#
...

Sid	Sname
1	A
2	B

Sid	Cname
1	C
1	C++
2	C
2	Java
2	C#

A doua forma normala (2NF)

Notiuni Preliminare

- ▶ **Dependenta functionala (DF):** Fie R o relatie si fie X si Y doua submultimi arbitrare de attribute din R. Atributul (simplu sau compus) Y este **dependent functional** de atributul X (simplu sau compus), notatie:

$$X \rightarrow Y$$

daca si numai daca fiecare valoare a lui X din R are asociata o **valoare precisa si unica** pentru Y.

- valoarea X se numeste **determinant**
- valoarea Y se numeste **determinat**

▶ Observatii

- ▶ Dependenta functionala se poate folosi ca proprietate (restrictie) pe care BD trebuie sa o indeplineasca: se pot adauga sau modifica valori in relatie numai daca DF este verificata
- ▶ Stabilirea DF este o sarcina a administratorului BD
- ▶ DF depind de semnificatia (*semantica*) datelor care se memoreaza in relatie
- ▶ Umatoarele trei forme normale (i.e., 2NF, 3NF, BCNF) utilizeaza notiunea de dependenta functionala
- ▶ Cand o relatie contine cel putin o DF, aceasta poate fi doar in 1NF, 2NF, 3NF, sau BCNF



A doua forma normala (2NF)

Notiuni Preliminare

- Proprietatile si axiomatizarea DF (axiomele lui Armstrong)

Fie X, Y, Z si W multimi de attribute din relatia R

Reflexivitatea

Daca Y este submultime a lui X , atunci $X \rightarrow Y$

Marirea:

Daca $X \rightarrow Y$, atunci $XZ \rightarrow YZ$

Tranzitivitatea:

Daca $X \rightarrow Y$ si $Y \rightarrow Z$,
atunci $X \rightarrow Z$

Auto-determinarea

$X \rightarrow X$

Reuniunea:

Daca $X \rightarrow Y$ si $X \rightarrow Z$, atunci $X \rightarrow YZ$

Decompunerea:

Daca $X \rightarrow YZ$, atunci $X \rightarrow Y$ si $X \rightarrow Z$

Pseudotranzitivitatea:

Daca $X \rightarrow Y$ si $WY \rightarrow Z$, atunci $WX \rightarrow Z$

Compunerea:

Daca $X \rightarrow Y$ si $Z \rightarrow W$, atunci $XZ \rightarrow YW$

A doua forma normala (2NF)

Notiuni Preliminare

- ▶ **Inchiderea unei multimi de dependente functionale:**
multimea tuturor DF obtinute logic din multimea S de DF se numeste inchiderea multimii S , notata S^+
(S^+ se poate obtine folosind axiomele lui Armstrong)
- ▶ **Inchiderea unei multimi de attribute (sub o multime de DF):**
 - ▶ In practica, nu e necesar sa se calculeze inchiderea unei multimi de DF;
 - ▶ Ceea ce se doreste este calculul unei anume submultimi a inchiderii: i.e., acea submultime continand toate DF avand o multime precizata Z de attribute in partea *stanga* (*determinant*);
 - ▶ Mai precis, inchiderea unei multimi de attribute dandu-se o relatie R , o multime Z de attribute din R si o multime S de DF valide pentru R , determina *multimea tuturor atributelor din R care sunt functional dependente de Z – inchiderea Z^+ lui Z sub R .*



A doua forma normala (2NF)

Notiuni Preliminare

► **Inchiderea unei multimi de attribute:**

- Algoritm de determinare a inchiderii Z^+ a unei multimi Z sub R (fie S o multime de DF sub R).

```
inchidere := Z;  
WHILE (inchidere se schimba) DO  
    FOR fiecare FD  $X \rightarrow Y$  din  $S^+$  DO  
        IF  $X$  submultime pentru inchidere THEN  
            inchidere := inchidere UNION  $Y$ 
```



A doua forma normala (2NF)

Notiuni Preliminare

- ▶ **Acoperire minimala de dependente functionale:**
 - ▶ O multime F de DF este acoperire minimala daca si numai daca indeplineste urmatoarele trei proprietati:
 1. Partea dreapta a fiecarei DF din F este formata dintr-un singur atribut
 2. Partea stanga a fiecarei DF din F este la randul ei minimala
 3. Reducand oricare DF se va schimba continutul lui F



A doua forma normala (2NF)

Notiuni Preliminare

- ▶ **Atribut prim:** atribut inclus in cheie
 - ▶ **Atribut neprim:** atribut care nu este inclus in nici o cheie
 - ▶ **Dependentia partiala**
 - ▶ $X \rightarrow a$ este dependenta partiala
- Daca - X submultime proprie a unei chei candidate
- a este atribut neprim



A doua forma normala (2NF)

► Anomalii cauzate de dependentele functionale

► Fie urmatorul exemplu

Exam(Sname, Discipline, Mark, Professor)

Unei discipline ii corespunde un singur profesor, dar unui profesor ii corespund mai multe discipline, deci $\{Discipline\} \rightarrow \{Professor\}$

Sname	Discipline	Mark	Professor
Bob	Databases	10	Boyce
Alice	Operating Systems	9	Codd
Bob	Operating Systems	8	Codd
John	Databases	9	Boyce
Mike	Databases	10	Boyce

Daca pastram o astfel de DF, pot aparea urmatoarele probleme, care necesita efort de programare aditional pentru a o rezolva (SQL nu este suficient):



A doua forma normala (2NF)

- ▶ Anomalii cauzate de dependentele functionale
 - ▶ **Risipa de spatiu, redundanta:** aceeași informație este stocată de mai multe ori (asocierea dintre *Databases* și *Boyce* e memorată de trei ori, și cea dintre *Operating Systems* și *Codd* de două ori)
 - ▶ **Anomalii la actualizare:** actualizarea datelor dintr-o asociere implică aceeași actualizare în toate asocierile corespunzătoare pentru a evita inconsistența BD. De ex., dacă schimbam valoarea din coloana *Professor* în prima înregistrare dar nu și din înregistrările 4 și 5, actualizarea va introduce o eroare în relație
 - ▶ **Anomalii la inserare:** la inserarea unei noi înregistrări, valorile atributelor trebuie cunoscute (nu se accepta valori nedefinite pentru attributele implicate în DF)
 - ▶ **Anomalii la ștergere:** la ștergerea înregistrărilor, se pot elimina în mod ireversibil asocieri. De ex., dacă înregistrările 2 și 3 sunt șterse, asocierea dintre *Discipline (Operating Systems)* și *Professor (Codd)* e pierdută



A doua forma normala (2NF)

► Solutia

- Anomaliile determinate de DF se pot elimina pastrand dependentele in tabele separate
- Aceasta necesita ca tabela initiala sa fie descompusa printr-o descompunere buna
- O astfel de descompunere se face in stadiul de design al BD, cand se stabilesc DF



A doua forma normala (2NF)

○ tabela este in 2NF daca si numai daca

- ▶ Este in 1NF si
- ▶ Orice atribut neprim (simplu sau compus) (deci care nu este inclus intr-o cheie) este complet dependent functional de oricare cheie a relatiei (nu exista dependente partiale)

Exemplu:

Exam(Sname, Discipline, Mark, Professor)

{Discipline} → {Professor}



A doua forma normala (2NF)

○ tabela este in 2NF daca si numai daca

- ▶ Este in 1NF si
- ▶ Orice atribut neprim (simplu sau compus) (deci care nu este inclus intr-o cheie) este complet dependent functional de oricare cheie a relatiei (nu exista dependente partiale)

Exemplu:



Exam(Sname, Discipline, Mark, Professor)

{Discipline} → {Professor} – dependenta partiala

} Nu e 2NF

A doua forma normala (2NF)

○ tabela este in 2NF daca si numai daca

- ▶ Este in 1NF si
- ▶ Nu exista dependente partiale

Example:

Exam(Sname, Discipline, Mark, Professor)

{Discipline} → {Professor} – dependenta partiala } Nu e 2NF

Solutia: Se elimina DF prin urmatoarea descompunere a relatiei initiale:

Results(Sname, Discipline, Mark)

DiscProf(Discipline, Professor)



A treia forma normala (3NF)

Notiuni preliminare

- ▶ Doua sau mai multe attribute sunt **mutual independente** daca nici una dintre ele nu este functional dependenta de nici o combinatie a celorlalte
- ▶ O astfel de independenta asigura posibilitatea ca fiecare atribut sa poata fi actualizat separat de restul atributelor
- ▶ **Dependenta tranzitiva:** Un atribut Z este tranzitiv dependent de atributul X daca exista Y astfel incat
 - $X \rightarrow Y$,
 - $Y \rightarrow Z$,
 - $Y \rightarrow X$ nu se verifica
 - Z nu e inclus in reuniunea multimilor X si Y



A treia forma normala (3NF)

○ tabela este in 3NF daca si numai daca

- ▶ Este in 2NF si
- ▶ Orice atribut neprim nu este tranzitiv dependent de oricare cheie a relatiei

Observatii:

- O relatie R care este in 2NF dar nu in 3NF are o dependenta $X \rightarrow Y$, X e neprim; aceasta dependenta se poate elimina descompunand R in mod similar ca in cazul 2NF
- Intr-o relatie 3NF
 - Nu exista nici o DF determinata de o submultime a cheii
 - Nu exista nici o DF intre doua attribute neprime

Exemplu: fie relatia

Graduation_Thesis(Sname, Mark, Supervisor, Department), unde
 $\{Supervisor\} \rightarrow \{Department\}$



A treia forma normala (3NF)

○ tabela este in 3NF daca si numai daca

- ▶ Este in 2NF si
- ▶ Orice atribut neprim nu este tranzitiv dependent de oricare cheie a relatiei

Exemplu: fie relatia

Graduation_Thesis(Sname, Mark, Supervisor, Department), unde
 $\{\text{Supervisor}\} \rightarrow \{\text{Department}\}$



} Nu e 3NF

A treia forma normala (3NF)

○ tabela este in 3NF daca si numai daca

- ▶ Este in 2NF si
- ▶ Orice atribut neprim nu este tranzitiv dependent de oricare cheie a relatiei

Exemplu: fie relatia

Graduation_Thesis(Sname, Mark, Supervisor, Department), unde
 $\{\text{Supervisor}\} \rightarrow \{\text{Department}\}$

} Nu e 3NF

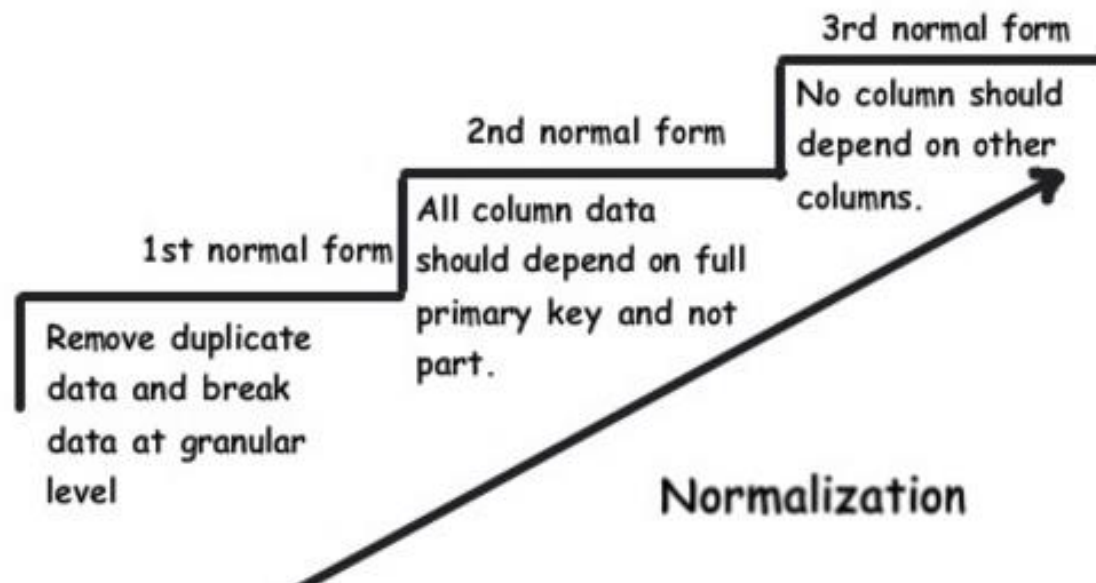
Solutia: Eliminarea DF prin descompunerea urmatoare a relatiei initiale

Results(Sname, Mark, Supervisor)

Professors(Supervisor, Department)



Concluzie



Sursa: <http://www.xpode.com/ShowArticle.aspx?ArticleId=578>



Cursul 6

- ▶ Normalizarea bazelor de date relationale (partea 2)
 - ▶ Boyce-Codd Normal Form (BCNF)
 - ▶ A 4-a Forma Normala (4NF)
 - ▶ A 5-a Forma Normala (5NF)



References

- ▶ Date, C.J., An Introduction to Database Systems (8th Edition), Addison-Wesley, 2004, *chapters 11, 12*
- ▶ Garcia-Molina, H., Ullman, J., Widom, J., *Database Systems: The Complete Book*, Pearson Prentice Hall, 2008, *chapter 3*
- ▶ Surdu Sabina, *Database course, course 5*

