



BAZE DE DATE

CURS 9

Partea II

Forma normală Boyce-Codd (BCNF)

- Determinantul este un atribut sau o mulțime de attribute neredundante, care constituie un identificator unic pentru alt atribut sau altă mulțime de attribute ale unei relații date.
- Intuitiv, o relație R este în **forma normală Boyce-Codd** dacă și numai dacă **fiecare determinant este o cheie candidat**.
- Formal, o relație R este în forma normală Boyce-Codd dacă și numai dacă pentru orice dependență funcțională totală $X \rightarrow A$, X este o cheie (candidat) a lui R .

Forma normală Boyce-Codd (BCNF)

- O relație R este în **forma normală Boyce-Codd** dacă și numai dacă **fiecare determinant este o cheie candidat**.
- **Regula Casey Delobel** pentru $R(K1\#, K2\#, X)$ presupunând că există dependența: $X \rightarrow K2$ (X determina functional pe K2)

→ $R1(K1\#, X)$ și $R2(X\#, K2)$

BCNF este o versiune puțin mai restrictivă a lui FN3. În FN3 attributele depind de cheia primară, doar de cheia primară și de nimic altceva în afara de cheie. Deci:

$\{K1\#, K2\# \} \rightarrow \{X\}$ (în FN3, X depinde de cheia K1#, K2#)

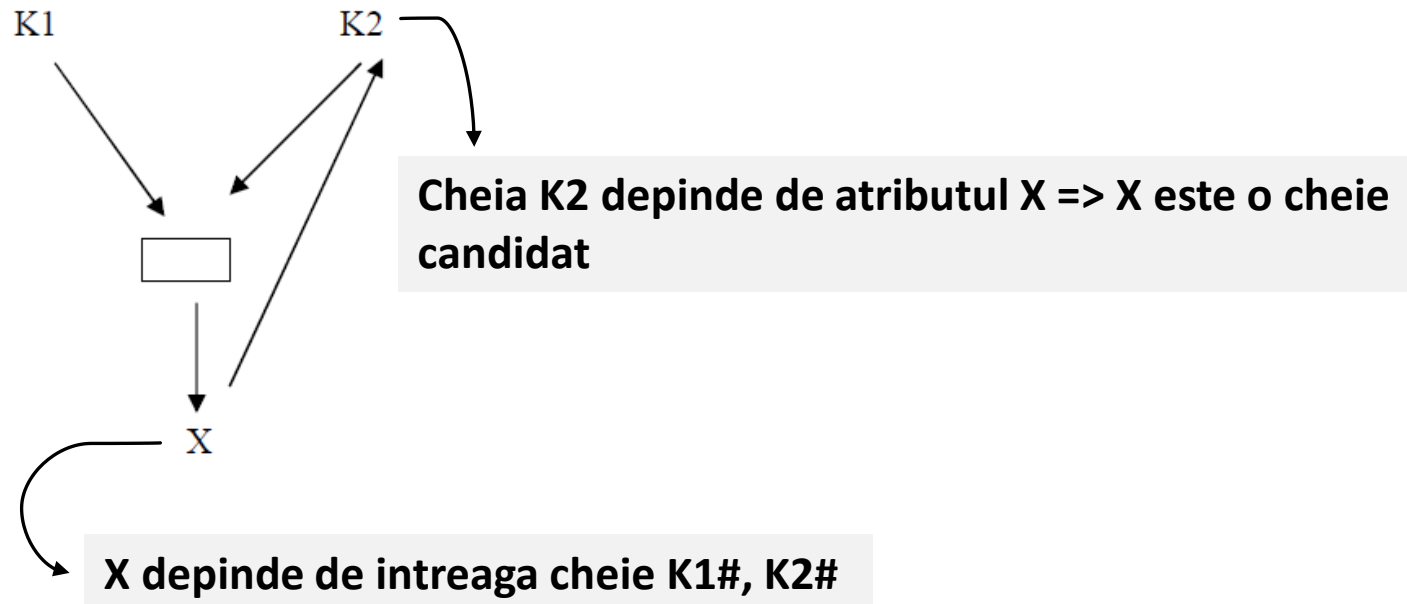
În BCNF fiecare determinant (în cazul exemplului curent - X) este o cheie candidat (adică poate fi cheia primară).

Asadar, $X \rightarrow K2$ (cheia K2 depinde de atributul X, deci X este o cheie candidat pentru relația R).

Forma normală Boyce-Codd (BCNF)

- O relație R este în **forma normală Boyce-Codd** dacă și numai dacă **fiecare determinant este o cheie candidat**.
- **Regula Casey Delobel** pentru $R(K1\#, K2\#, X)$ presupunând că există dependența: $X \rightarrow K2$ (X determina functional pe K2)

➔ $R1(K1\#, X)$ și $R2(X\#, K2)$ -> X este cheia candidat



Forma normală Boyce-Codd (BCNF)

Exemplul 1:

Relația **INVESTESE_IN** leagă entitățile **INVESTITOR** și **OBIECTIV_INVESTITIE**. Ea are schema relațională:

INVESTESE_IN (cod_contractant#, cod_obiectiv#, nr_contract, cota_parte)

- Între atributele relației există dependențele:

$\{\text{cod_contractant\#}, \text{cod_obiectiv\#}\} \rightarrow \{\text{nr_contract}, \text{cota_parte}\},$

$\{\text{nr_contract}\} \rightarrow \{\text{cod_obiectiv}\}$ (se observa ca nr_contract este o cheie candidat, cheia cod_obiectiv depinzand de atributul nr_contract)

- Se aplică regula Casey-Delobel și **se aduce relația în BCNF.**

INVESTESE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESE_IN_2 (cod_contractant#, nr_contract, cota_parte); (nr_contract este cheie externa)

Forma normală Boyce-Codd (BCNF)

INVESTESE_IN (cod_contractant#, cod_obiectiv#, nr_contract, cota_parte).

Se descompune in:

INVESTESE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESE_IN_2 (cod_contractant#, nr_contract, cota_parte).
(nr_contract este cheie externa)

cod_contractant#	cod_obiectiv#	nr_contract	cota_parte
C1	01	NR1	20%
C1	02	NR2	20%
C2	02	NR2	30%
C2	01	NR3	30%
C3	01	NR1	35%

Se poate observa cheia candidat **nr_contract**

Atributul nr_contract devine cheie primara, iar relatia se va descompune astfel:

Forma normală Boyce-Codd (BCNF)

INVESTESE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESE_IN_2 (cod_contractant#, nr_contract#, cota_parte). (nr_contract este cheie externa)

cod_contractant#	cod_obiectiv#	nr_contract	cota_parte
C1	01	NR1	20%
C1	02	NR2	20%
C2	02	NR2	30%
C2	01	NR3	30%
C3	01	NR1	35%

Atributul nr_contract devine cheie primara, iar relatia se va descompune astfel:

cod_obiectiv	nr_contract#
01	NR1
02	NR2
01	NR3

cod_contractant#	nr_contract#	cota_parte
C1	Nr1	20%
C1	Nr2	20%
C2	Nr2	30%
C2	Nr3	30%
C3	Nr1	35%

Forma normală Boyce-Codd (BCNF)

Exemplul 2:

Avem următoarea relație: **R (Student#, Materie#, Profesor)** – un student este înscris la mai multe materii, urmând cursurile mai multor profesori, iar un profesor predă o singură materie.

{Student#, Materie#} -> {Profesor}

{Profesor} -> {Materie} (atributul profesor este o cheie candidat deoarece un profesor predă o singură materie, deci apare o singură dată alături de materia lui)

- Un student este înscris la mai multe materii (avem nevoie de cheia primară compusă astfel încât studentul să poată participa în cadrul diferitelor materii). Un profesor poate predă o singură materie.

S1	M1	P1
S2	M1	P1
S3	M1	P1
S1	M2	P2

Se observă că studentul S1 este înscris la materiile M1 și M2, predate de profesorii P1 și P2. De asemenea, materia M1 are mai mulți studenți înscrși.

Forma normală Boyce-Codd (BCNF)

$\{\text{Student\#}, \text{Materie\#}\} \rightarrow \{\text{Profesor}\}$

$\{\text{Profesor}\} \rightarrow \{\text{Materie}\}$ (atributul profesor este o cheie candidat deoarece un profesor predă o singură materie, deci apare o singură dată alături de materia lui)

S1	M1	P1
S2	M1	P1
S3	M1	P1
S1	M2	P2

Aplicăm regula Casey-Delobel:

R1 (Profesor#, Materie#) – cheia candidat **Profesor** a devenit cheie primară

R2 (Student#, Profesor#) – atributul Profesor nu trebuie să fie neapărat cheie externă în R2. Poate fi și cheie primară compusă (student#, profesor# - este cheie primară compusă, dar atributul profesor individual este cheie externă). **În acest caz este utilă cheia primară compusă. De ce?**

Forma normală Boyce-Codd (BCNF)

- Pentru ca o relație să fie adusă în BCNF **nu trebuie în mod obligatoriu să fie în FN3.**
- Se pot aduce în BCNF și relații aflate în FN1 sau FN2.
 - Acest lucru este posibil întrucât dependențele funcționale parțiale și cele tranzitive sunt tot dependențe noncheie, adică dependențe ai căror determinanți nu sunt chei candidat.

Forma normală 4 (FN4)

- FN4 elimină redundanțele datorate relațiilor $m:n$, adică datorate **dependenței multiple**
- Intuitiv, o relație R este în **a patra formă normală** dacă și numai dacă relația este în BCNF (adică fiecare determinant trebuie să fie o cheie candidat) și nu conține relații $m:n$ independente
- Multidependentele se notează astfel:
 - $X \twoheadrightarrow Z$ (X multidetermină pe Z sau Z este multidependent de X)

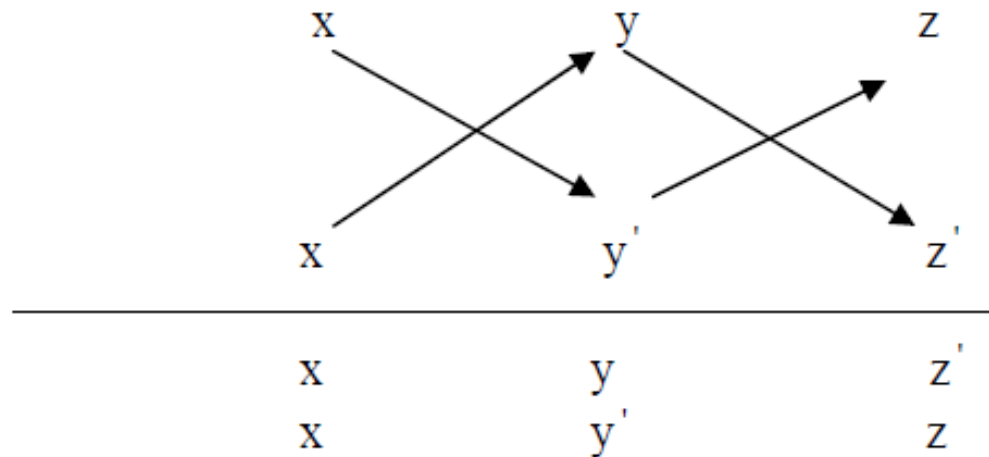
Forma normală 4 (FN4)

- Fie R o relație definită pe o mulțime de attribute $A = \{A_1, A_2, \dots, A_n\}$ și fie $X, Y, Z \subset A$. Se spune că X **multidetermină** pe Z sau că Z este multidependent de X :
 - dacă pentru fiecare valoare a lui Z în R există numai o valoare pentru perechea (X, Y) ;
 - dacă valoarea lui Z depinde numai de valoarea lui X .
- Acest tip de dependență, numită și multivaloare sau multidependență (MVD) se notează prin $X \twoheadrightarrow Z$
- Intuitiv, multidependența reprezintă situația în care valoarea unui atribut (sau a unei mulțimi de attribute) determină o mulțime de valori a altui atribut (sau mulțimi de attribute)

Forma normală 4 (FN4)

Multidependența $X \twoheadrightarrow Y$ poate fi gândită ca o **regulă de deducție**:

- dacă tuplurile $\langle x, y, z \rangle$ și $\langle x, y', z' \rangle$ sunt în relație la un moment r , atunci la momentul r sunt în relație și tuplurile $\langle x, y, z' \rangle$ și $\langle x, y', z \rangle$.



Forma normală 4 (FN4)

FN4 elimină redundanțele datorate relațiilor $m:n$, adică datorate **dependenței multiple**.

Intuitiv, o relație R este **în a patra formă normală** dacă și numai dacă relația este în BCNF (adică fiecare determinant trebuie să fie o cheie candidat) și conține relații $m:n$ dependente (dependenta – materie și profesor / independenta – user și song)

Formal, relația R este **în a patra formă normală** dacă și numai dacă:

- R este în BCNF;
- orice dependență multivaloare este o dependență funcțională;

Cu alte cuvinte, o relație BCNF este în FN4 dacă pentru orice multidependență elementară de forma $X \twoheadrightarrow Y$, X este o **supercheie** a lui R .

Forma normală 4 (FN4)

Exemplul 1:

Un restaurant are mai multe tipuri de pizza si mai multe arii de distributie. Astfel, daca este necesara inserarea unui nou restaurant se poate realiza aceasta operatie doar daca se insereaza si un tip de pizza, dar si o arie de distributie.

Exista urmatoarea relatie: **R (FIRMA#, TIP_LIVRARE#, ZONA#)**

In acest exemplu avem urmatoarele dependente multiple:

firma $\rightarrow \rightarrow$ tip_livrare

firma $\rightarrow \rightarrow$ zona

Regula de descompunere în relații FN4:

- Fie $R(X, Y, Z)$ o schemă relațională care nu este în FN4 și fie $X \rightarrow \rightarrow Y$ o multidependență elementară care nu este de forma „CHEIE $\rightarrow \rightarrow$ atribut”. Această relație este descompusă prin proiecție în două relații:

$$R = \text{JOIN}(\pi_{X \cup Y}(R), \pi_{X \cup Z}(R)).$$

In cazul exemplului anterior, relatia R se va descompune astfel:

R1 (FIRMA#, TIP_LIVRARE#)

R2 (FIRMA#, ZONA#)

Forma normală 4 (FN4)

Exemplul 1:

O companie de curierat are mai multe tipuri de livrari si mai multe zone in care livreaza.

Exista urmatoarea relatie:

R (FIRMA#, TIP_LIVRARE#, ZONA#)

In acest exemplu avem urmatoarele dependente multiple:

FIRMA -> -> TIP_LIVRARE

FIRMA -> -> ZONA

Relatiile rezultate dupa aplicarea lui FN4:

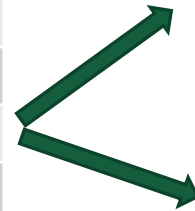
R1 (FIRMA#, TIP_LIVRARE#)

R2 (FIRMA#, ZONA#)

FIRMA	TIP_LIVRARE	ZONA
GLS	RAPID	ZONA1
GLS	RAPID	ZONA2
GLS	CLASIC	ZONA1
GLS	CLASIC	ZONA2
FAN	RAPID	ZONA3
FAN	CLASIC	ZONA3
CARGUS	CLASIC	ZONA1
CARGUS	CLASIC	ZONA2
CARGUS	CLASIC	ZONA4
CARGUS	RAPID	ZONA1
CARGUS	RAPID	ZONA2
CARGUS	RAPID	ZONA4

Forma normală 4 (FN4)

FIRMA	TIP_LIVRARE	ZONA
GLS	RAPID	ZONA1
GLS	RAPID	ZONA2
GLS	CLASIC	ZONA1
GLS	CLASIC	ZONA2
FAN	RAPID	ZONA3
FAN	CLASIC	ZONA3
CARGUS	CLASIC	ZONA1
CARGUS	CLASIC	ZONA2
CARGUS	CLASIC	ZONA4
CARGUS	RAPID	ZONA1
CARGUS	RAPID	ZONA2
CARGUS	RAPID	ZONA4



FIRMA	TIP_LIVRARE
GLS	RAPID
GLS	CLASIC
FAN	RAPID
FAN	CLASIC
CARGUS	CLASIC
CARGUS	RAPID

FIRMA	ZONA
GLS	ZONA1
GLS	ZONA2
FAN	ZONA3
CARGUS	ZONA1
CARGUS	ZONA2
CARGUS	ZONA4

Forma normală 4 (FN4)

Exemplul 2:

Avem următoarea relație **R(Curs#, Prof#, Carte#)** și regula – un curs este predat de mai mulți profesori și se află în mai multe cărți.

Curs#	Prof#	Carte#
1	P1	Carte1
1	P1	Carte2
1	P2	Carte1
1	P2	Carte2

Relația nu se află în FN4 din cauza multidependențelor:

Curs \twoheadrightarrow Prof

Curs \twoheadrightarrow Carte

Aducem în FN4:

Curs1(Curs#, Carte#)

1	Carte1
1	Carte2

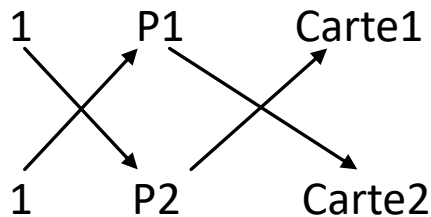
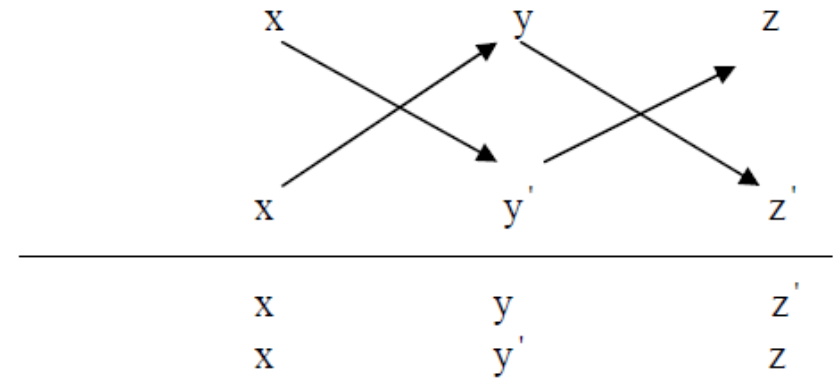
Curs2(Curs#, Prof#)

1	P1
1	P2

Forma normală 4 (FN4)

Cum se poate aplica regula de deducție în acest caz:

Curs#	Prof#	Carte#
1	P1	Carte1
1	P1	Carte2
1	P2	Carte1
1	P2	Carte2



Pornim de la cursul 1 situat în partea de sus, urmând săgețile =>

=> **1 P2 Carte1**

Pornim de la cursul 1 din partea de jos =>

=> **1 P1 Carte2**

Forma normală 4 (FN4)

Exemplul 3:

Fie relația:

INVESTITIE(cod_contractant#, cod_obiectiv#, telefon#) și ca regulă de proiectare -> un investitor poate avea mai multe numere de telefon și poate investi în mai multe obiective.

Între attributele relației există multidependențele:

cod_contractant# $\rightarrow\rightarrow$ cod_obiectiv;

cod_contractant# $\rightarrow\rightarrow$ telefon;

Relația **INVESTITIE** este în **BCNF**. Pentru a aduce relația în **FN4** o vom descompune prin proiecție în două relații:

INVESTITIE_1(cod_contractant#, cod_obiectiv#),

INVESTITIE_2(cod_contractant#, telefon#).

INVESTITIE = JOIN(INVESTITIE_1, INVESTITIE_2).

Forma normală 5 (FN5)

- **FN5** își propune eliminarea redundanțelor care apar în relații $m:n$ dependente.
 - În general, aceste relații nu pot fi descompuse.
 - S-a arătat că o relație de tip 3 este diferită de trei relații de tip 2. Există totuși o excepție, și anume, **dacă relația este ciclică**
- **Intuitiv**, o relație R este **în forma normală 5** dacă și numai dacă:
 - relația este în FN4;
 - **nu conține dependențe ciclice.**

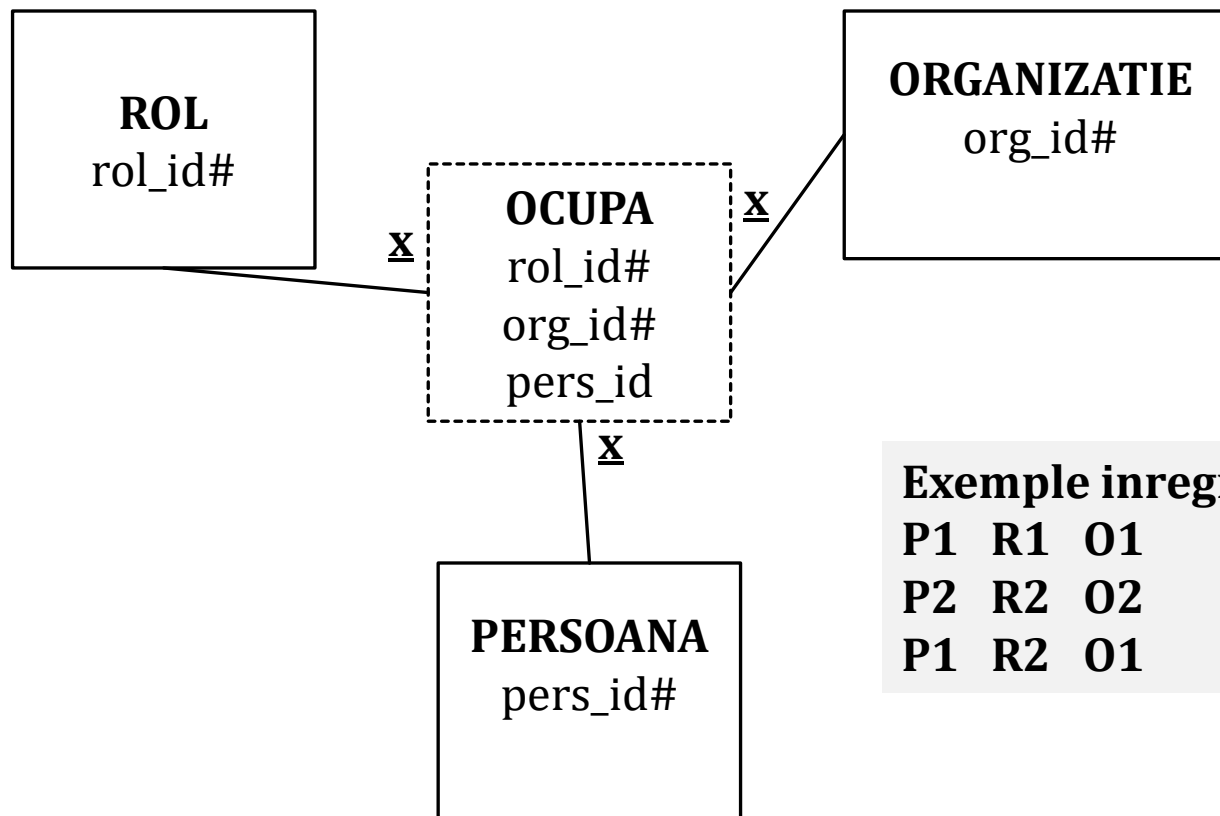
Forma normală 5 (FN5)

- Dependența funcțională și multidependența permit descompunerea prin proiecție, fără pierdere de informație, a unei relații în **două relații**;
- Regulile de descompunere (FN1 – FN4) nu dau toate descompunerile posibile prin proiecție ale unei relații;
- Există relații care nu pot fi descompuse în două relații, dar pot fi descompuse în trei, patru sau mai multe relații fără a pierde informații.
- Pentru a obține descompuneri *L-join* în trei sau mai multe relații, s-a introdus conceptul de **join-dependență sau dependență la compunere** (JD).

Forma normală 5 (FN5)

Exemplu:

O persoana poate ocupa mai multe roluri si poate face parte din mai multe organizatii. O organizatie contine mai multe roluri.

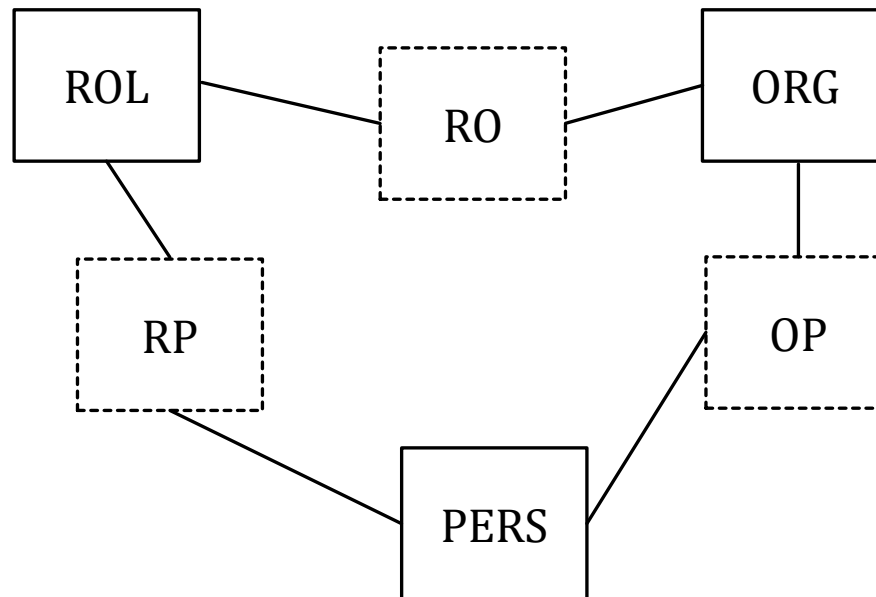


Exemple inregistrari:

P1	R1	01
P2	R2	02
P1	R2	01

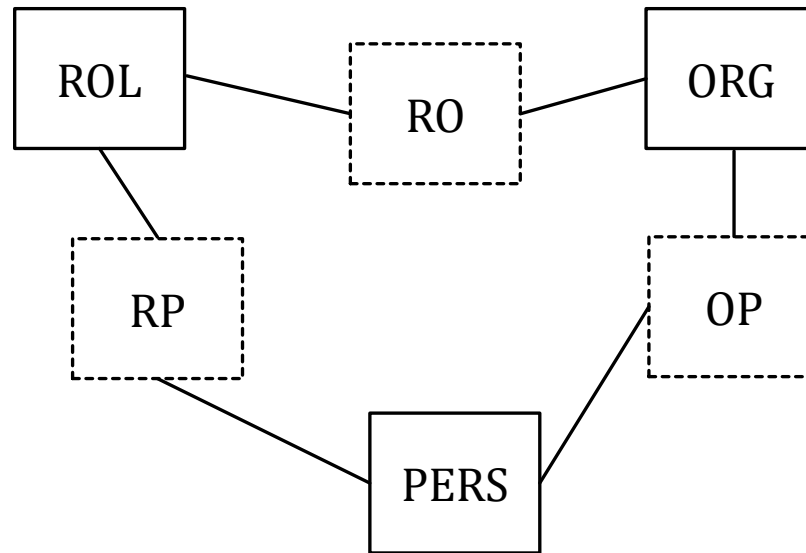
Forma normală 5 (FN5)

Relatia de tip 3 precedenta poate fi echivalenta cu 3 relatii de tip 2 doar daca aceste relatii de tip 2 sunt ciclice. In cazul de mai jos, relatiile de tip 2 existente sunt ciclice, conform regulilor modelului.



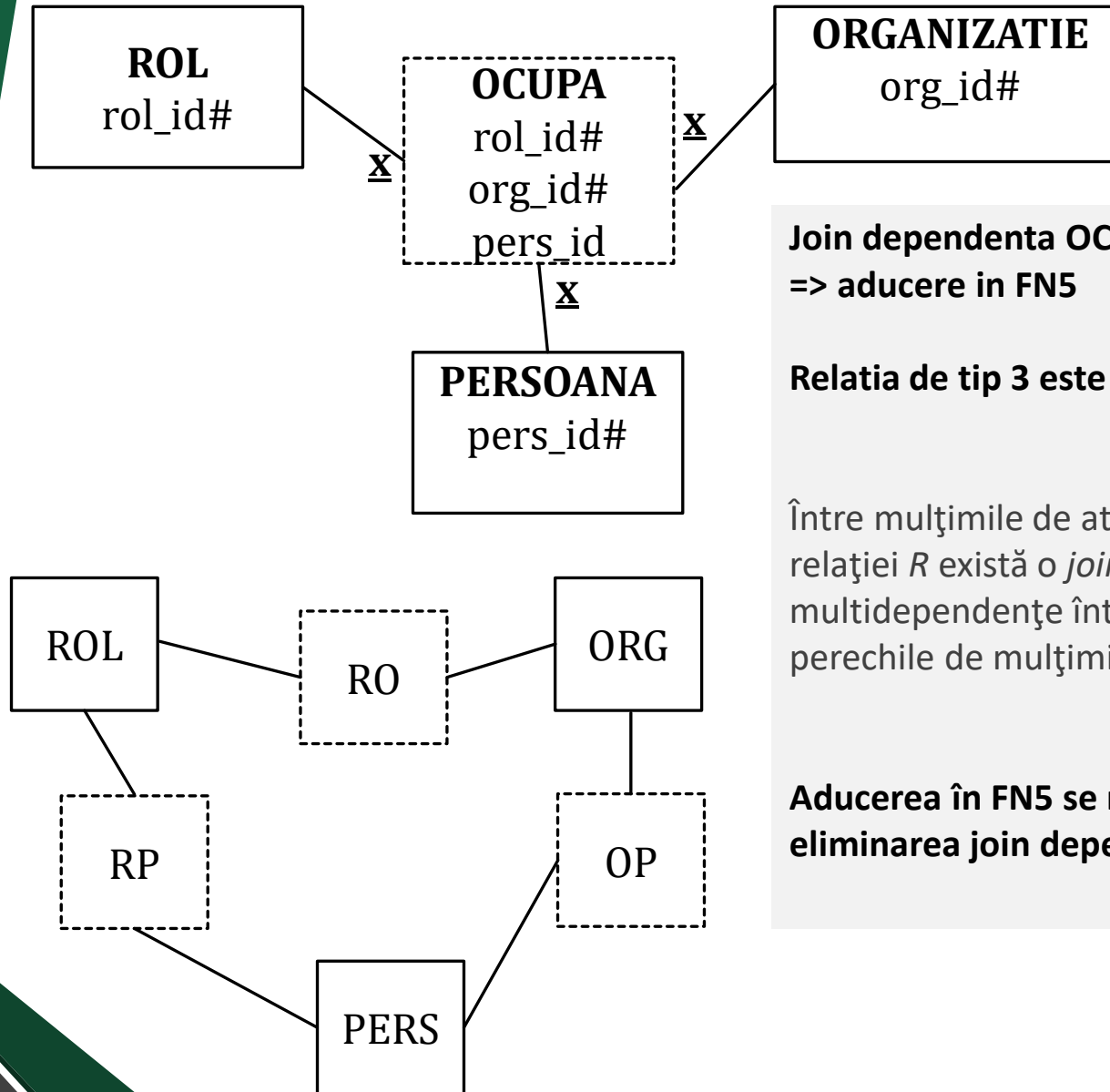
Forma normală 5 (FN5)

Relatia de tip 3 precedenta poate fi echivalenta cu 3 relatii de tip 2 doar daca aceste relatii de tip 2 sunt ciclice. In cazul de mai jos, relatiile de tip 2 existente sunt ciclice, conform regulilor modelului.



Relatia fiind ciclica, atunci cand facem toate join-urile vom obtine un rezultat echivalent cu cel obtinut din relatia de tip 3. O relatie pentru a fi in FN5 trebuie sa fie in FN4 si **sa nu contina dependente ciclice**. Se observa ca cele 3 relatii de tip 2 compun o diagrama care contine dependente ciclice, deci relatia de mai sus **nu se afla in FN5**.

Forma normală 5 (FN5)



Join dependenta **OCUPA** = JOIN (RO, RP, OP)
=> aducere în FN5

Relatia de tip 3 este în FN5.

Între mulțimile de attribute X , Y și Z din cadrul relației R există o *join* dependență dacă există multidependențe între fiecare dintre perechile de mulțimi (X, Y) , (Y, Z) și (X, Z) .

Aducerea în FN5 se realizează prin eliminarea join dependențelor.

Concluzii normalizare

1. **FN1 \rightarrow FN2** elimină redundanțele datorate dependenței netotale a atributelor care nu participă la o cheie, față de cheile lui R . Se suprimă dependențele funcționale care nu sunt totale.
2. **FN2 \rightarrow FN3** elimină redundanțele datorate dependenței tranzitive. Se suprimă dependențele funcționale tranzitive.
3. **FN3 \rightarrow BCNF** elimină redundanțele datorate dependenței funcționale. Se suprimă dependențele în care partea stângă nu este o supercheie.
4. **BCNF \rightarrow FN4** elimină redundanțele datorate multidependenței. Se suprimă toate multidependențele care nu sunt și dependențe funcționale.
5. **FN4 \rightarrow FN5** elimină redundanțele datorate dependenței ciclice. Se suprimă toate *join*-dependențele care nu sunt implicate de o cheie.
6. **BCNF, FN4 și FN5** corespund regulii că orice determinant este o cheie, dar de fiecare dată dependența cu care se definește determinantul este alta și anume dependența funcțională, multidependența sau *join*-dependența.
7. Descompunerea unei relații FN2 în FN3 conservă datele și dependențele, pe când descompunerea unei relații FN3 în BCNF și, respectiv, a unei relații BCNF în FN4 conservă doar datele.

Denormalizare

- Procedura de normalizare elimină redundanțele prin efectuarea unor proiecții, DAR NU toate redundanțele pot fi eliminate în acest mod.

=> Uneori este necesară **denormalizarea** care presupune:

- [➤ Mărirea redundanței;
- [➤ Reducerea numărului de join-uri efectuate => micșorare timp de execuție;

Denormalizare

- Ideile normalizării sunt utile în proiectarea BD, dar **nu sunt obligatorii**
- Dependența și normalizarea sunt de natură semantică (cu alte cuvinte, se referă la ceea ce înseamnă datele)
- În schimb, algebra relațională și calculul relațional (limbajele SQL) se referă doar la valorile efective ale datelor și în multe cazuri nu necesită mai mult decât FN1

Denormalizare

- Obiectivul denormalizării constă în **reducerea numărului de join-uri** efectuate pentru rezolvarea unei interogări, prin realizarea unora dintre acestea în avans, ca făcând parte din proiectarea bazei de date;
- Conceptul de denormalizare suferă de un număr de probleme binecunoscute:
 - Odată începută denormalizarea, nu este clar unde trebuie să se oprească;
 - Nu se mai lucrează cu relații normalizate și astfel pot apărea anomaliile pe care normalizarea le corectează;

Denormalizare

Exemple:

- Avem un tabel în care sunt stocate produse, numit **PRODUSE**. Produsele au și un atribut **greutate**. Această coloană conține valori repetitive, aceeași greutate definind mai multe produse. În acest caz, dacă în baza de date există un tabel separat în care se află greutatea împreună cu id-ul produsului căruia îi corespunde, este necesar procesul de **denormalizare** în urma căruia atributul **greutate** se va plasa în tabelul **PRODUSE** deoarece nu este eficient ca acest atribut să se afle într-un tabel separat.
- Pe de o parte, este un avantaj ca atributul greutate să fie într-un tabel separat pentru cazurile în care **se modifică** greutatea produselor destul de des. Fiind într-un tabel separat, modificarea va avea loc o singură dată în acest tabel, fiind mai rapid și mai optim.
- Pe de alta parte, dacă se consideră că **accesarea** greutății produselor are loc frecvent, odată cu accesarea informațiilor despre produse, atunci denormalizarea este cea mai bună alegere.

Denormalizare

Exemple:

- În cazul în care avem **User** si **Profile** între care există relația one-to-many (un user are un singur profil), iar tabelul profil conține attributele **nume** și **prenume**, atunci se poate realiza o denormalizare în urma căreia cele doua attribute se vor plasa în tabelul User deoarece operațiile de join sunt costisitoare, fiind mult mai eficientă în acest caz mutarea coloanelor în tabelul User. Pe de altă parte, dacă tabelul Profile conține mai multe attribute, atunci se vor păstra ambele tabele în baza de date.

Denormalizare

Când este utilă denormalizarea?

- Ca o regulă empirică, se poate afirma că, dacă performanțele nu sunt satisfăcătoare și relația are o rată de reactualizare scăzută, dar o rată a interogarilor foarte ridicată, denormalizarea poate constitui o opțiune viabilă;
- Nu există reguli fixe pentru stabilirea situațiilor în care este indicată denormalizarea relațiilor;