BAZE DE DATE

CURS 9

Necesitatea normalizării

- In momentul proiectarii si modelarii unei baze de date relationale pot sa apara anomalii. Procesul de optimizare a relatiilor conceptuale poarta numele de **normalizare**. Pentru a depista eventualele redundante se utilizeaza **dependentele functionale**. (Vezi Curs 8)
- **X -> Y** - X determina functional pe Y sau Y depinde functional de X
- Anomaliile care apar în lucrul cu baza de date se produc din cauza dependenţelor care există între datele din cadrul relaţiilor bazei de date.
  - Dependenţele sunt plasate greşit în tabele!!!
- O dependență poate provoca:
  - **anomalii** la inserare, modificare sau ştergere (Vezi Curs 8 – pag 18)
  - **redundanţă** în date

Necesitatea normalizării

- **Normalizarea** are drept **scop**:
  - suprimarea **redundanţei** logice,
  - evitarea **anomaliilor** la reactualizare,
  - rezolvarea **problemei reconexiunii**.
- Există o teorie matematică a normalizării al cărei autor este E.F. Codd.
  - Soluţia: construirea unor tabele standard (forme normale).
- Normalizarea este **procesul reversibil de transformare a unei relaţii**, în relaţii de structură mai simplă.
  - Procesul este reversibil în sensul că **nicio informaţie nu este pierdută în timpul transformării**.
  - O relaţie este într-o formă normală particulară dacă ea satisface o **mulţime specificată de constrângeri**.

Necesitatea normalizării

- Relaţie universală + mulţime de anomalii
  - Orice formă normală se obţine aplicând o schemă de descompunere. Există două tipuri de descompuneri.
- **Descompuneri ce conservă dependenţele**.
  - descompunerea relaţiei R în proiecţiile R_1, R_2, ..., R_k, a.î. dependenţele lui R sunt echivalente (au închideri pseudo-tranzitive identice) cu reuniunea dependenţelor lui R_1, R_2, ..., R_k.
- **Descompuneri fără pierderi de informaţie** (*L-join*).
  - descompunerea relaţiei R într-o mulţime de proiecţii R_1, R_2, ..., R_j, a.î. pentru orice instanță a lui R este adevărată relaţia:

R = JOIN(Π_{B1} (R), Π_{B2} (R), ...,Π_{Bj} (R))

  - Relaţia iniţială = compunerea naturală a relaţiilor obţinute prin descompunere.

Necesitatea normalizării

- Formele normale sunt obţinute prin descompuneri fără pierderi de informaţie.
- O descompunere fără pierdere de informaţie, utilizată în procesul normalizării, este dată de **regula Casey-Delobel**:
- Fie $R(A)$ o schemă relaţională şi fie α, β, γ o partiţie a lui A. Presupunem că α determină funcţional pe β. Atunci:

$$R(A) = \text{JOIN}(\Pi_{\alpha\cup\beta}(R), \Pi_{\alpha\cup\gamma}(R)).$$

  - $\alpha\cup\beta$ → mulţimea atributelor care intervin în dependenţele funcţionale;
  - $\alpha\cup\gamma$ → reprezintă reuniunea determinantului cu restul atributelor lui A.

Exemplu regula Casey-Delobel

Avion

A#	nume	capacitate	localitate
1	AIRBUS	250	PARIS
2	AIRBUS	250	PARIS
3	AIRBUS	250	LONDRA
4	CAR	100	PARIS
5	B707	150	LONDRA
6	B707	150	LONDRA

- Se considera relatia: **AVION**(A#, nume, capacitate, localitate)

α = {nume}

β = {capacitate}

γ = {A#, localitate}

Regula Casey-Delobel: Fie *R*(*A*) o schemă relaţională şi fie α, β, γ o partiţie a lui *A*. Presupunem că α determină funcţional pe β. Atunci:

$$R(A) = \text{JOIN}(\Pi_{\alpha\cup\beta}(R), \Pi_{\alpha\cup\gamma}(R))$$

α -> β

nume -> capacitate **=>** AVION1 = (nume#, capacitate)

- Avem $\Pi_{\alpha\cup\beta}(R)$, unde α ∪ β – este multimea atributelor care intervin in dependentele functionale
- $\Pi_{\alpha\cup\gamma}(R)$, unde α ∪ γ – reprezinta reuniunea determinatului cu restul atributelor relatiei AVION

In final relatia AVION se va descompune in urmatoarele scheme relationale:

AVION1 = (nume#, capacitate)
AVION2 = (A#, nume, localitate)

Exemplu regula Casey-Delobel

Avion

A#	nume	capacitate	localitate
1	AIRBUS	250	PARIS
2	AIRBUS	250	PARIS
3	AIRBUS	250	LONDRA
4	CAR	100	PARIS
5	B707	150	LONDRA
6	B707	150	LONDRA

In final relatia AVION se va descompune in urmatoarele scheme relationale:

AVION1 = (nume#, capacitate)
AVION2 = (A#, nume, localitate)

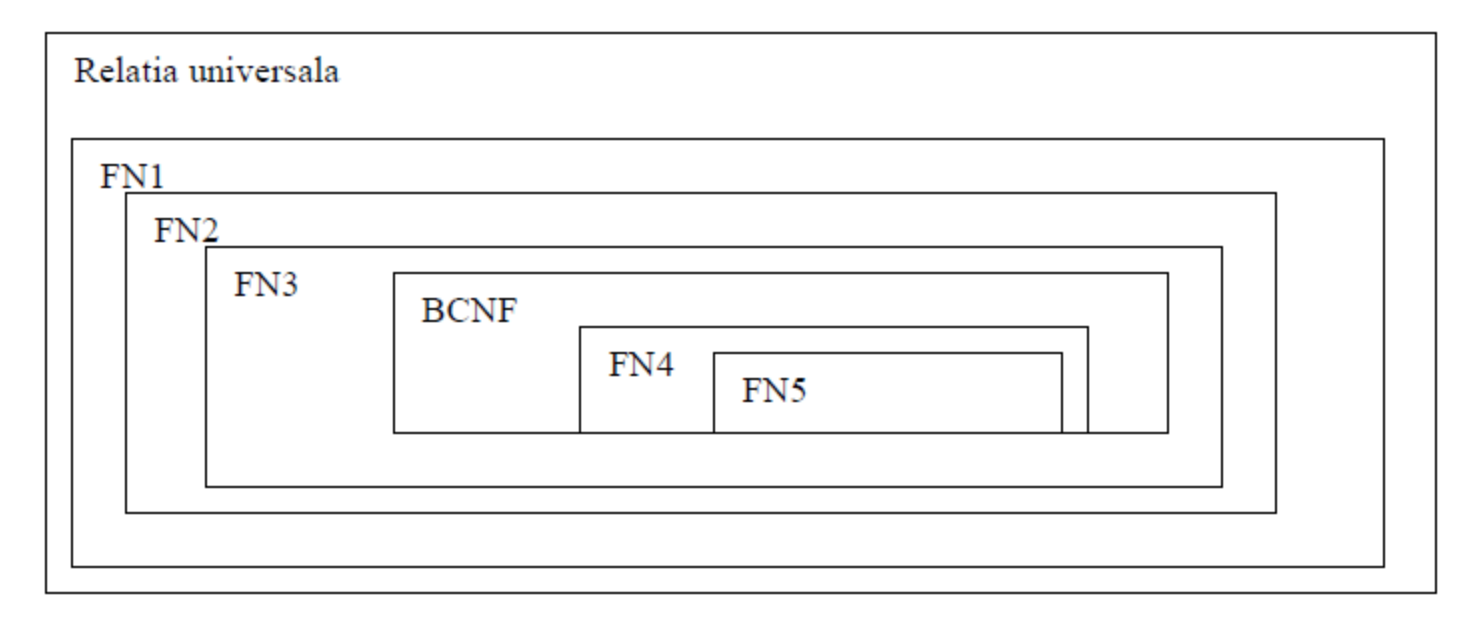
AVION1

Nume	Capacitate
AIRBUS	250
CAR	100
B707	150

AVION2

A#	Nume	Localitate
1	AIRBUS	PARIS
2	AIRBUS	PARIS
3	AIRBUS	LONDRA
4	CAR	PARIS
5	B707	LONDRA
6	B707	LONDRA

Necesitatea normalizării



Forma normală 1 (FN1)

- O relaţie este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă (atomică).
- Exemplu:

VEHICUL (Non FN1)

cod_persoana#	vehicule
P1	DL, RC, FF
P2	RM, VW
P3	DL

Forma normală 1 (FN1)

Varianta 1:

VEHICUL (FN1)

cod_persoana#	vehicul#
P1	DL
P1	RC
P1	FF
P2	RM
P2	VW
P3	DL

Varianta 2:

VEHICUL (FN1)

cod_persoana#	vehicul1	vehicul2	vehicul3
P1	DL	RC	FF
P2	RM	VW	null
P3	DL	null	null

Forma normală 2 (FN2)

- O relaţie *R* este în a doua formă normală dacă şi numai dacă:
  - relaţia *R* este în FN1;
  - fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară.
- FN2 interzice manifestarea unor dependenţe funcţionale parţiale în cadrul relaţiei *R*.

Forma normală 2 (FN2)

atasat_la

Cod_salariat#	Job_cod	Nr_proiect#	Functia	Suma
S1	Programator	P1	Supervizor	60
S1	Programator	P2	Cercetator	25
S1	Programator	P3	Auxiliar	10
S3	Vanzator	P3	Supervizor	60
S5	Inginer	P3	Supervizor	60

- Un salariat are mai multe functii si poate lucra la mai multe proiecte
- O relaţie *R* este în a doua formă normală dacă şi numai dacă:
  - relaţia *R* este în FN1; -> **relatia atasat_la se afla in FN1 deoarece avem identificator unic pentru toate intrarile din tabel**
  - fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară – in cazul nostru atributele **job_cod, functia, suma** nu sunt chei si trebuie sa depinda direct de intreaga cheie primara **cod_salariat#** si **nr_proiect#** -> aceste atribute nu depind direct de intreaga cheie primara deoarece se observa dependenta directa dintre **job_cod** si **cod_salariat**, insemnand ca job_cod depinde direct doar de o parte a cheii primare, si anume doar de cod_salariat -> **relatia atasat_la nu se afla in FN2**

Forma normală 2 (FN2)

atasat_la

Cod_salariat#	Job_cod	Nr_proiect#	Functia	Suma
S1	Programator	P1	Supervizor	60
S1	Programator	P2	Cercetator	25
S1	Programator	P3	Auxiliar	10
S3	Vanzator	P3	Supervizor	60
S5	Inginer	P3	Supervizor	60

- Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară – in cazul nostru atributele **job_cod, functia, suma** nu sunt chei si trebuie sa depinda direct de intreaga cheie primara **cod_salariat#** si **nr_proiect#** -> aceste atribute nu depind direct de intreaga cheie primara deoarece se observa dependenta directa dintre **job_cod** si **cod_salariat**, insemnand ca job_cod depinde direct doar de o parte a cheii primare, si anume doar de cod_salariat -> **relatia atasat_la nu se afla in FN2**
- Astfel avem:
  - {cod_salariat#} -> {job_cod} – cod_salariat determina functional job_cod
  - {cod_salariat#, nr_proiect#} -> {functia, suma}

Forma normală 2 (FN2)

atasat_la

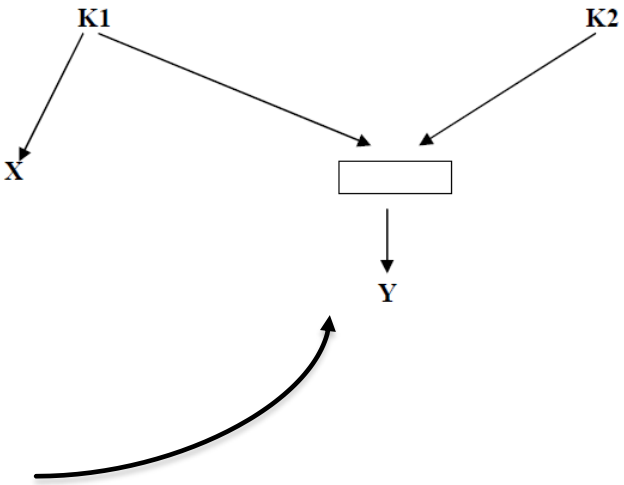
Cod_salariat#	Job_cod	Nr_proiect#	Functia	Suma
S1	Programator	P1	Supervizor	60
S1	Programator	P2	Cercetator	25
S1	Programator	P3	Auxiliar	10
S3	Vanzator	P3	Supervizor	60
S5	Inginer	P3	Supervizor	60

- Astfel avem:
  - {cod_salariat#} -> {job_cod} – cod_salariat determina functional job_cod
  - {cod_salariat#, nr_proiect#} -> {functia, suma}

Aplicarea regulii Casey-Delobel pentru FN2:

Fie relaţia *R*(*K*1, *K*2, *X*, *Y*), unde *K*1 şi *K*2 definesc cheia primară, iar *X* şi *Y* sunt mulţimi de atribute, astfel încât $K1 \rightarrow X$.

Din cauza dependenţei funcţionale $K1 \rightarrow X$ care arată că *R* nu este în FN2, se înlocuieşte *R* (fără pierdere de informaţie) prin două proiecţii (Vezi pag 6):
*R*1(*K*1, *K*2, *Y*) şi *R*2(*K*1, *X*).



Forma normală 2 (FN2)

atasat_la

Cod_salariat#	Job_cod	Nr_proiect#	Functia	Suma
S1	Programator	P1	Supervizor	60
S1	Programator	P2	Cercetator	25
S1	Programator	P3	Auxiliar	10
S3	Vanzator	P3	Supervizor	60
S5	Inginer	P3	Supervizor	60

Astfel avem:

{cod_salariat#} -> {job_cod} – cod_salariat determina functional job_cod

{cod_salariat#, nr_proiect#} -> {functia, suma}

Transformarea in FN2:

atasat_2a

Cod_salariat#	Nr_proiect#	Functia	Suma
S1	P1	Supervizor	60
S1	P2	Cercetator	25
S1	P3	Auxiliar	10
S3	P3	Supervizor	60
S5	P3	Supervizor	60

atasat_2b

Cod_salariat#	Job_cod
S1	Programator
S3	Vanzator
S5	Inginer

Forma normală 2 (FN2)

Exemplu:

- Presupunem că un şantier poate executa mai multe lucrări de bază şi că o lucrare poate fi executată de mai multe şantiere, iar o lucrare este definita de intervalul de timp in care este executata.

LUCRARE(cod_obiectiv#, cod_lucrare#, nume);

SANTIER(nr_santier#, specialitate, sef);

EXECUTA(cod_obiectiv#, cod_lucrare#, nr_santier#, descriere, functie, conducator, data_inceput, data_sfarsit).

- Pentru relaţia EXECUTA sunt evidente dependenţele:

{cod_obiectiv#, cod_lucrare#} → {data_inceput, data_sfarsit},

{cod_obiectiv#, cod_lucrare#, nr_santier#} → {descriere, functie, conducator}.

- Relaţia EXECUTA este în FN1, dar nu este în FN2. Se aplică regula Casey Delobel:

EXECUTA_1(cod_obiectiv#, cod_lucrare#, nr_santier#, descriere, functie, conducator)

EXECUTA_2(cod_obiectiv#, cod_lucrare#, data_inceput, data_sfarsit).

Forma normală 3 (FN3)

- **Intuitiv**, o relaţie *R* este în a treia formă normală dacă şi numai dacă:
  - relaţia *R* este în FN2;
  - fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară.
- Fie *R* o relaţie, *X* o submulţime de atribute ale lui *R* şi *A* un atribut al relaţiei *R*. *A* este **dependent tranzitiv** de *X* dacă există *Y* astfel încât $X \rightarrow Y$ şi $Y \rightarrow A$ (*A* nu aparţine lui *Y* şi *Y* nu determină pe *X*). *X* nu este dependent funcţional de *Y* sau *A*!
  - De exemplu, dacă $K_1, K_2, K_3 \rightarrow A_1$ şi dacă $K_1, K_2, A_1 \rightarrow A_2$.

  Rezulta ca A_2 este dependent tranzitiv de K_1, K_2, K_3. (A_2 este dependent de K_1, K_2, K_3 prin A1)

Forma normală 3 (FN3)

atasat_2a

Cod_salariat#	Nr_proiect#	Functia	Suma
S1	P1	Supervizor	60
S1	P2	Cercetator	25
S1	P3	Auxiliar	10
S3	P3	Supervizor	60
S5	P3	Supervizor	60

- Formal, o relaţie R este în a treia formă normală dacă şi numai dacă:
  - relaţia R este în FN2; -> **este in FN2** deoarece este in FN1 si indeplineste conditia ca fiecare atribut care nu este cheie (nu participă la cheia primară) sa fie dependent de întreaga cheie primară
  - fiecare atribut care nu este cheie (nu participă la o cheie) nu este dependent tranzitiv de nicio cheie a relatiei R. Cu alte cuvinte **o relaţie este în FN3 dacă şi numai dacă fiecare atribut (coloană) care nu este cheie, depinde de cheie, de întreaga cheie şi numai de cheie.**

In exemplul **atasat_2a** se observa ca atributul ***suma*** depinde tranzitiv de cheia primara compusa ***cod_salariat*** si ***nr_proiect*** prin intermediul atributului ***functia.***

Forma normală 3 (FN3)

atasat_2a

Cod_salariat#	Nr_proiect#	Functia	Suma
S1	P1	Supervizor	60
S1	P2	Cercetator	25
S1	P3	Auxiliar	10
S3	P3	Supervizor	60
S5	P3	Supervizor	60

- In exemplul **atasat_2a** se observa ca atributul ***suma*** depinde tranzitiv de cheia primara compusa ***cod_salariat*** si ***nr_proiect*** prin intermediul atributului ***functia.***
- Astfel aveam:
  - {cod_salariat#, nr_proiect#} -> {functia} – cod_salariat si nr_proiect determina functional atributul functia
  - {cod_salariat#, nr_proiect#} -> {functia} -> {suma}

Pentru a aduce relaţia **atasat_2a** în FN3 se aplică regula Casey-Delobel. Relaţia se descompune, prin eliminarea dependenţelor funcţionale tranzitive, în proiecţiile:

ATASAT_3A(cod_salariat#, nr_proiect#, functia)

ATASAT_3B(functia, suma)

Forma normală 3 (FN3)

atasat_2a

Cod_salariat#	Nr_proiect#	Functia	Suma
S1	P1	Supervizor	60
S1	P2	Cercetator	25
S1	P3	Auxiliar	10
S3	P3	Supervizor	60
S5	P3	Supervizor	60

- Pentru a aduce relaţia **atasat_2a** în FN3 se aplică regula Casey-Delobel. Relaţia se descompune, prin eliminarea dependenţelor funcţionale tranzitive, în proiecţiile:

  ATASAT_3A(cod_salariat#, nr_proiect#, functia)

  ATASAT_3B(functia, suma)

- **Transformarea in FN3**

atasat_3a

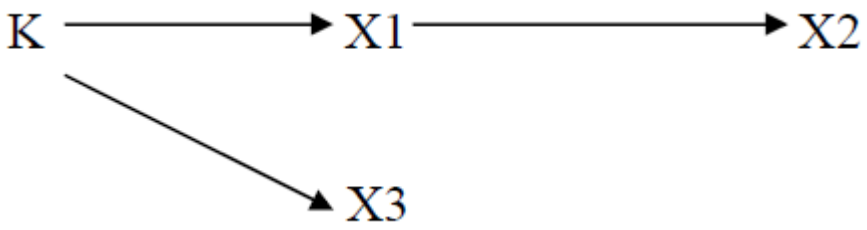
Cod_salariat#	Nr_proiect#	Functia
S1	P1	Supervizor
S1	P2	Cercetator
S1	P3	Auxiliar
S3	P3	Supervizor
S5	P3	Supervizor

atasat_3b

Functia	Suma
Supervizor	60
Cercetator	25
Auxiliar	10

Forma normală 3 (FN3)

- Aplicarea regulii Casey-Delobel pentru FN3
  - Fie relaţia $R(K, X_1, X_2, X_3)$, unde atributul X_2 depinde tranzitiv de K, iar K este cheia primară a lui R. Presupunem că $K \rightarrow X_1 \rightarrow X_2$.
  - Din cauza dependenţei funcţionale $X_1 \rightarrow X_2$ care arată că R nu este în FN3, se înlocuieşte R (fără pierdere de informaţie) prin două proiecţii $R1(K, X_1, X_3)$ şi $R2(X_1, X_2)$.



Forma normală 3 (FN3)

- Exemplu:

În tabelul **EXECUTA1**(cod_obiectiv#, cod_lucrare#, nr_santier#, descriere, functie, conducator) continuă să existe redundanţă în date.

- Atributul ***conducator*** depinde indirect de cheia primară prin intermediul atributului ***functie***. Între atributele relaţiei există dependenţele:

{cod_obiectiv#, cod_lucrare#, nr_santier#} → {descriere},

{cod_obiectiv#, cod_lucrare#, nr_santier#} → {functie} → {conducator}.

- Se aplică regula Casey-Delobel. Relaţia se descompune, prin eliminarea dependenţelor funcţionale tranzitive, în proiecţiile:

EXECUTA1_1(cod_obiectiv#, cod_lucrare#, nr_santier#, descriere, functie)

EXECUTA1_2(functie, conducator).

Schema de sinteză pentru obţinerea lui FN3

- **Algoritmul de sinteză** construieşte o acoperire minimală *F* a dependenţelor funcţionale totale.
  - Se elimină atributele şi dependenţele funcţionale redundante.
  - Mulţimea *F* este partiţionată în grupuri F_i, astfel încât în fiecare grup F_i sunt dependenţe funcţionale care au acelaşi membru stâng şi nu există două grupuri având acelaşi membru stâng.
  - Fiecare grup F_i produce o schemă FN3.
- Algoritmul realizează o descompunere ce conservă dependenţele.

Schema de sinteză pentru obţinerea lui FN3

- **Algoritm SNF3** (aducerea unei relaţii în FN3 prin utilizarea unei scheme de sinteză):

1. Se determină *F* o acoperire minimală a lui *E* (mulţimea dependenţelor funcţionale).
2. Se descompune mulţimea *F* în grupuri notate F_i, astfel încât în cadrul fiecărui grup să existe dependenţe funcţionale având aceeaşi parte stângă.
3. Se determină perechile de chei echivalente (*X*, *Y*) în raport cu *F* (două mulţimi de atribute X, Y sunt chei echivalente dacă în mulţimea de dependenţe E există atât dependenţa X →Y, cât şi dependența Y → X).

Schema de sinteză pentru obţinerea lui FN3

4. Pentru fiecare pereche de chei echivalente: se identifică grupurile F_i şi F_j care conţin dependenţele funcţionale cu partea stângă X şi respectiv Y; se formează un nou grup de dependenţe F_{ij}, care va conţine dependenţele funcţionale având membrul stâng (X, Y); se elimină grupurile F_i şi F_j, iar locul lor va fi luat de grupul F_{ij}.
5. Se determină o acoperire minimală a lui F, care va include toate dependenţele $X \rightarrow Y$, unde X şi Y sunt chei echivalente (celelalte dependenţe sunt redundante).
6. Se construiesc relaţii FN3 (câte o relaţie pentru fiecare grup de dependenţe funcţionale).

Schema de sinteză pentru obţinerea lui FN3

- Se observă că **algoritmul solicită**:
  - determinarea unei **acoperiri minimale** (algoritmii EAR şi EDF);
  - determinarea **închiderii (A^+) unei mulţimi de atribute A** în raport cu mulţimea de dependenţe funcţionale E (algoritm AIDF).
- Determinarea acoperirii minimale presupune eliminarea atributelor şi dependenţelor redundante.
- Acoperirea minimală **nu este unică** şi depinde de ordinea în care sunt eliminate aceste atribute şi dependenţe redundante.

Schema de sinteză pentru obţinerea lui FN3

- **Algoritm EAR** (elimină atributele redundante din determinantul dependenţelor funcţionale)
- Pentru fiecare dependenţă funcţională din E şi pentru fiecare atribut din partea stângă a unei dependenţe funcţionale:
  - **Pas1**. Se elimină atributul considerat.
  - **Pas2**. Se calculează închiderea părţii stângi reduse.
  - **Pas3**. Dacă închiderea conţine toate atributele din determinantul dependenţei, atunci atributul eliminat la pasul 1 este redundant şi rămâne eliminat. În caz contrar, atributul nu este redundant şi se reintroduce în partea stângă a dependenţei funcţionale.

Schema de sinteză pentru obţinerea lui FN3

- **Algoritm EDF** (elimină dependenţele funcţionale redundante din E)
- Pentru fiecare dependenţă funcţională X →Y din E:
  - **Pas1**. Se elimină dependenţa din E.
  - **Pas2**. Se calculează închiderea X^+ , în raport cu mulţimea redusă de dependenţe.
  - **Pas3**. Dacă Y este inclus în X^+, atunci dependenţa X →Y este redundantă şi rămâne eliminată. În caz contrar, se reintroduce în E.

Schema de sinteză pentru obţinerea lui FN3

- **Algoritm AIDF** (determină închiderea lui A)
  - **Pas1.** Se caută dacă există în E dependenţe X →Y pentru care determinantul X este o submulţime a lui A, iar determinatul Y nu este inclus în A.
  - **Pas2**. Pentru fiecare astfel de dependenţă funcţională se adaugă mulţimii A atributele care constituie determinatul dependenţei.
  - **Pas3**. Dacă nu mai există dependenţe funcţionale de tipul de la pasul 1, atunci $A^+ = A$.

Forma normală Boyce-Codd (BCNF)

- Determinantul este un atribut sau o mulţime de atribute neredundante, care constituie un identificator unic pentru alt atribut sau altă mulţime de atribute ale unei relaţii date.
- **Intuitiv**, o relaţie *R* este în forma normală Boyce-Codd dacă şi numai dacă **fiecare determinant este o cheie candidat**.
- **Formal**, o relaţie *R* este în forma normală Boyce-Codd dacă şi numai dacă pentru orice dependenţă funcţională totală $X \rightarrow A$, *X* este o cheie (candidat) a lui *R*.

Forma normală Boyce-Codd (BCNF)

- O relaţie R este în forma normală **Boyce-Codd** dacă şi numai dacă **fiecare determinant este o cheie candidat**.
- **Regula Casey Delobel** pentru **R(K1#, K2#, X**) presupunând că există dependenţa: X → K2 (X determina functional pe K2)

  ➔ R1(K1#, X) şi R2(X#, K2)

BCNF este o versiune putin mai restrictiva a lui FN3. In FN3 atributele depind de cheia primara, doar de cheia primara si de nimic altceva in afara de cheie. Deci:

{K1#, K2#} -> {X} (**in FN3, X depinde de cheia K1#, K2#**)

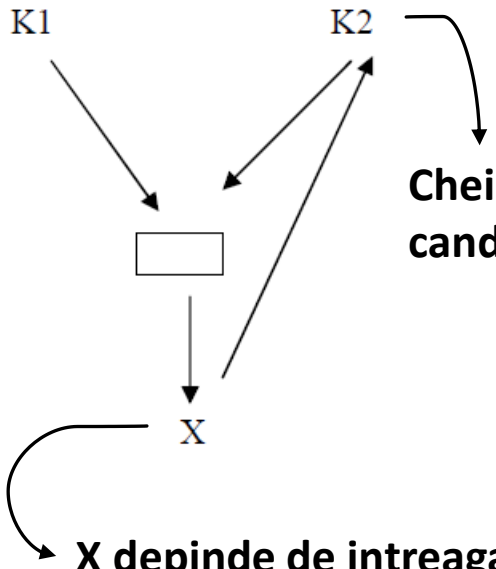
In BCNF fiecare determinant (in cazul exemplului curent - X) este o cheie candidat (adica poate fi cheie primara).

Asadar, X -> K2 (cheia K2 depinde de atributul X, deci X este o cheie candidat pentru relatia R)

Forma normală Boyce-Codd (BCNF)

- O relaţie R este în forma normală **Boyce-Codd** dacă şi numai dacă **fiecare determinant este o cheie candidat**.
- **Regula Casey Delobel** pentru **R(K1#, K2#, X**) presupunând că există dependenţa: X → K2 (X determina functional pe K2)

  ➔ R1(K1#, **X**) şi R2(**X#**, K2) -> X este cheia candidat



Cheia K2 depinde de atributul X => X este o cheie candidat

X depinde de intreaga cheie K1#, K2#

Forma normală Boyce-Codd (BCNF)

- **Exemplul 1**:

Relaţia **INVESTESTE_IN** leagă entităţile **INVESTITOR** şi **OBIECTIV_INVESTITIE**. Ea are schema relaţională:

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

- Între atributele relaţiei există dependenţele:

{cod_contractant#, **cod_obiectiv#**} → {nr_contract, cota_parte},

{nr_contract} → {**cod_obiectiv**} (se observa ca nr_contract este o cheie candidat, cheia cod_obiectiv depinzand de atributul nr_contract)

- Se aplică regula Casey-Delobel şi **se aduce relaţia în BCNF**.

INVESTESTE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESTE_IN_2 (cod_contractant#, nr_contract, cota_parte). (nr_contract este cheie externa)

Forma normală Boyce-Codd (BCNF)

- **Exemplul 1**:

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

Se descompune in:

INVESTESTE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESTE_IN_2 (cod_contractant#, nr_contract, cota_parte). (nr_contract este cheie externa)

cod_contractant#	cod_obiectiv#	nr_contract	cota_parte
C1	O1	NR1	20%
C1	O2	NR2	20%
C2	O2	NR2	30%
C2	O1	NR3	30%
C3	O1	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)

- **Exemplul 1**:

INVESTESTE_IN_1 (cod_obiectiv, nr_contract#); (nr_contract fiind cheie candidat devine cheie primara)

INVESTESTE_IN_2 (cod_contractant#, nr_contract, cota_parte). (nr_contract este cheie externa)

cod_contractant#	cod_obiectiv#	nr_contract	cota_parte
C1	O1	NR1	20%
C1	O2	NR2	20%
C2	O2	NR2	30%
C2	O1	NR3	30%
C3	O1	NR1	35%

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

cod_obiectiv	nr_contract#
O1	NR1
O2	NR2
O1	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 urmatoarea relatie: R (Student#, Materie#, Profesor) – un student este inscris la mai multe materii, urmand cursurile mai multor profesori, iar un profesor preda o singura materie.

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

{Profesor} -> {Materie} (atributul profesor este o cheie candidat deoarece un profesor preda o singura materie, deci apare o singura data alaturi de materia lui)

- Un student este inscris la mai multe materii (avem nevoie de cheia primara compusa astfel incat studentul sa poata participa in cadrul deferitelor materii). Un profesor poate preda o singura materie.

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

Se observa ca studentul S1 este inscris la materiile M1 si M2, predate de profesorii P1 si P2. De asemenea, materia M1 are mai multi student inscrisi.

Forma normală Boyce-Codd (BCNF)

- **Exemplul 2**:

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

{Profesor} -> {Materie} (atributul profesor este o cheie candidat deoarece un profesor preda o singura materie, deci apare o singura data alaturi de materia lui)

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

Aplicam regula Casey-Delobel:

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

R2 (Student#, Profesor#) – atributul Profesor nu trebuie sa fie neaparat cheie externa in R2. Poate fi si cheie primara compusa deoarece (student#, profesor#) este cheie primara compusa, dar atributul profesor individual este cheie externa. In acest caz este utila cheia primara compusa. De ce?

Forma normală Boyce-Codd (BCNF)

- Pentru ca o relaţie să fie adusă în BCNF nu trebuie în mod obligatoriu să fie în FN3.
- Se pot aduce în BCNF şi relaţii aflate în FN1 sau FN2.
  - Acest lucru este posibil întrucât dependenţele funcţionale parţiale şi cele tranzitive sunt tot dependenţe noncheie, adică dependenţe ai căror determinanţi nu sunt chei candidat.

Forma normală Boyce-Codd (BCNF)

- **Algoritm TFBCNF** (aducerea unei relaţii *R* din FN1 în BCNF)

1. Dacă relaţia conţine cel mult două atribute, atunci *R* este în BCNF şi algoritmul s-a terminat.
2. Dacă relaţia conţine mai mult de două atribute, se consideră toate perechile (*X*, *Y*) de atribute distincte din *A*.
3. Se determină A_1^+, închiderea mulţimii $A_1 = A - \{X, Y\}$.
4. Dacă pentru orice pereche (*X*, *Y*), $X \notin A_1^+$ atunci relaţia *R* este în BCNF şi algoritmul s-a terminat.
5. În caz contrar (pentru cel puţin o pereche (*X*, *Y*), *X* aparţine lui A_1^+), relaţia *R* nu este în BCNF.
6. Se reduce progresiv schema relaţiei şi se reia algoritmul, exploatând relaţia redusă. Orice relaţie obţinută prin reducerea lui *R* şi care este în BCNF se consideră ca făcând parte din descompunerea lui *R* în procesul aducerii sale în BCNF.

Forma normală 4 (FN4)

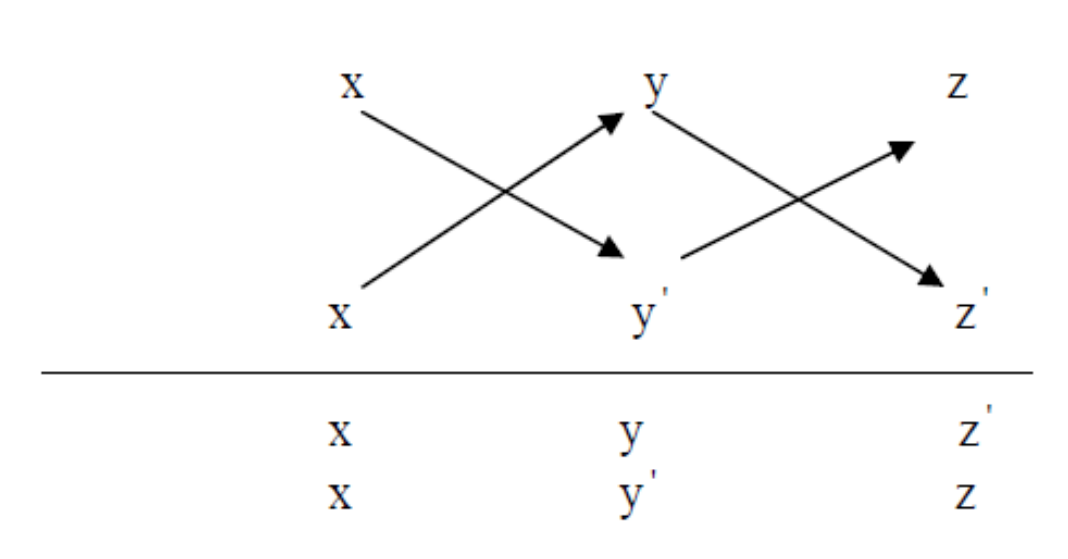
- FN4 elimină redundanţele datorate relaţiilor *m*:*n*, adică datorate **dependenţei multiple**.
- **Intuitiv**, o relaţie *R* este în a patra formă normală dacă şi numai dacă relaţia este în BCNF (adica fiecare determinant trebuie sa fie o cheie candidat) şi nu conţine relaţii *m*:*n* independente.
- Multidependentele se noteaza astfel:
  - **X $\rightarrow\!\!\!\rightarrow$ Z** (X multidetermina pe Z sau Z este multidependent de X)

Forma normală 4 (FN4)

- Fie R o relaţie definită pe o mulţime de atribute $A = \{A_1, A_2, ..., A_n\}$ şi fie $X, Y, Z \subset A$. Se spune că X **multideterminină** 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 dependenţă, numită şi multivaloare sau multidependenţă (MVD) se notează prin $X \twoheadrightarrow Z$.
- **Intuitiv, multidependenţa reprezintă situaţia în care valoarea unui atribut (sau a unei mulţimi de atribute) determină o mulţime de valori a altui atribut (sau mulţimi de atribute)!!!**

Forma normală 4 (FN4)

- Multidependenţa $X \rightarrow\rightarrow Y$ poate fi gândită ca o regulă de deducţie:
  - dacă tuplurile <*x*, *y*, *z*> şi <*x*, *y*′, *z*′> sunt în relaţie la un moment *r*, atunci la momentul *r* sunt în relaţie şi tuplurile <*x*, *y*, *z*′> şi <*x*, *y*′, *z*>.



Forma normală 4 (FN4)

- Orice dependenţă funcţională este o multidependenţă.
- Afirmaţia inversă nu este adevărată.
- Dacă $X \rightarrow Y$ (FD), atunci pentru oricare două tupluri $<x, y, z>$ şi $<x, y', z'>$, se obţine $y = y'$. Prin urmare în relaţie apar tuplurile $<x, y', z>$ şi $<x, y, z'>$ şi deci $X \rightarrow\rightarrow Y$ (MVD).

Forma normală 4 (FN4)

- Fie *W*, *V*, *X*, *Y* şi *Z* submulţimi de atribute ale unei scheme relaţionale *R*. Fiind dată o mulţime *T* de multidependenţe există o mulţime completă de axiome (Ax1–Ax8) care permit obţinerea tuturor multidependenţelor ce se pot deduce din mulţimea *T.*
- O **multidependenţă elementară** este o multidependenţă care are părţi stângi şi drepte minimale (nu există $X' \subset X$ şi $Y' \subset Y$ a.i. $X' \rightarrow\rightarrow Y'$).
- **Formal**, relaţia *R* este în a patra formă normală dacă şi numai dacă:
  - *R* este în BCNF;
  - orice dependenţă multivaloare este o dependenţă funcţională.

Forma normală 4 (FN4)

- O relaţie BCNF este în FN4 dacă pentru orice multidependenţă elementară de forma $X \rightarrow\rightarrow Y$, X este o supercheie a lui R.
- **Regula de descompunere în relaţii FN4**.
  - Fie $R(X, Y, Z)$ o schemă relaţională care nu este în FN4 şi fie $X \rightarrow\rightarrow Y$ o multidependenţă elementară care nu este de forma „CHEIE $\rightarrow\rightarrow$ atribut“.
  - Această relaţie este descompusă prin proiecţie în două relaţii:

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

Forma normală 4 (FN4)

FN4 elimină redundanţele datorate relaţiilor *m*:*n*, adică datorate **dependenţei multiple**.

Intuitiv, o relaţie *R* este în a patra formă normală dacă şi numai dacă relaţia este în BCNF (adica fiecare determinant trebuie sa fie o cheie candidat) şi nu conţine relaţii *m*:*n* independente.

Formal, relaţia *R* este în a patra formă normală dacă şi numai dacă:

- *R* este în BCNF;
- orice dependenţă multivaloare este o dependenţă funcţională;

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.

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(RESTAURANT#, TIP#, ARIE#)**

In acest exemplu avem urmatoarele dependente multiple:

restaurant -> -> tip de pizza
restaurant -> -> arii de distributie

Regula de descompunere în relaţii FN4.

- Fie $R(X, Y, Z)$ o schemă relaţională care nu este în FN4 şi fie $X \rightarrow\rightarrow Y$ o multidependenţă

elementară care nu este de forma „CHEIE $\rightarrow\rightarrow$ atribut“. Această relaţie este descompusă prin proiecţie în două relaţii:

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

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

R1 (RESTAURANT#, TIP#)
R2 (RESTAURANT#, ARIE#)

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(RESTAURANT#, TIP#, ARIE#)**

In acest exemplu avem urmatoarele dependente multiple:

restaurant -> -> tip de pizza

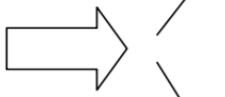
restaurant -> -> arii de distributie

Relatiile rezultate dupa aplicarea lui FN4:

R1 (RESTAURANT#, TIP#)

R2 (RESTAURANT#, ARIE#)

Restaurant	Tip de pizza	Aria de distributie
Pizza Hut	Clasic	Dr. Taberei
Pizza Hut	Clasic	Militari
Pizza Hut	Pufos	Dr. Taberei
Pizza Hut	Pufos	Militari
Springtime	Pufos	Domenii
Springtime	Clasic	Domenii
Jerry's	Clasic	Dr. Taberei
Jerry's	Clasic	Militari
Jerry's	Clasic	Crangasi
Jerry's	Pufos	Dr. Taberei
Jerry's	Pufos	Militari
Jerry's	Pufos	Crangasi



Restaurant	Tip de pizza
Pizza Hut	Clasic
Pizza Hut	Pufos
Springtime	Pufos
Springtime	Clasic
Jerry's	Clasic
Jerry's	Pufos

Restaurant	Aria de distributie
Pizza Hut	Dr. Taberei
Pizza Hut	Militari
Springtime	Domenii
Jerry's	Dr. Taberei
Jerry's	Militari
Jerry's	Crangasi

Forma normală 4 (FN4)

Exemplul 2:

Avem urmatoarea relatie R(Curs#, Prof#, Carte#) si regula – un curs este predat de mai multi profesori si se afla in mai multe carti.

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

Relatia nu se afla in FN4 din cauza multidependentelor:

Curs -> -> Prof

Curs -> -> Carte

Aducem in FN4:

Curs1(Curs#, Carte#)

1	Carte1
1	Carte2

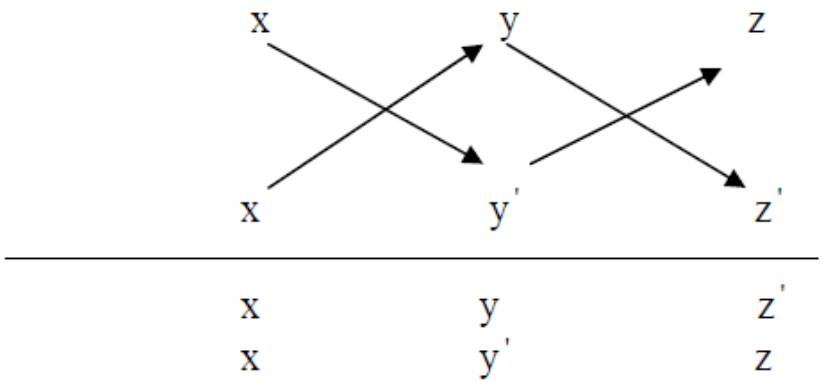
Curs2(Curs#, Prof#)

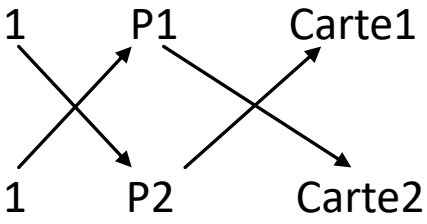
1	P1
1	P2

Forma normală 4 (FN4)

Cum se poate aplica regula de deductie in acest caz:

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





Pornim de la cursul 1 situat in partea de sus, urmand sagetile =>

$\Rightarrow$ **1 P2 Carte1**

Pornim de la cursul 1 din partea de jos =>

=> **1 P1 Carte2**

Forma normală 4 (FN4)

Exemplul 3:

Fie relaţia:
INVESTITIE(cod_contractant#, denumire#, telefon#) şi presupunem că un investitor poate avea mai multe numere de telefon şi că poate investi în mai multe obiective.

Între atributele relaţiei există multidependenţele:
cod_contractant# $\rightarrow\!\!\rightarrow$ denumire;
cod_contractant# $\rightarrow\!\!\rightarrow$ telefon;

Relaţia **INVESTITIE** este în **BCNF**. Pentru a aduce relaţia în **FN4** o vom descompune prin proiecţie în două relaţii:

INVESTITIE_1(cod_contractant#, denumire#),
INVESTITIE_2(cod_contractant#, telefon#).

INVESTITIE = JOIN(INVESTITIE_1, INVESTITIE_2).

Forma normală 5 (FN5)

- **FN5** îşi propune eliminarea redundanţelor care apar în relaţii *m:n* dependente.
  - În general, aceste relaţii nu pot fi descompuse.
  - S-a arătat că o relaţie de tip 3 este diferită de trei relaţii de tip 2. Există totuşi o excepţie, şi anume, **dacă relaţia este ciclică**
- **Intuitiv**, o relaţie *R* este în forma normală 5 dacă şi numai dacă:
  - relaţia este în FN4;
  - **nu conţine dependenţe ciclice**.

Forma normală 5 (FN5)

- Dependenţa funcţională şi multidependenţa permit descompunerea prin proiecţie, fără pierdere de informaţie, a unei relaţii în **două relaţii**.
- Regulile de descompunere (FN1 – FN4) nu dau toate descompunerile posibile prin proiecţie ale unei relaţii.
- Există relaţii care nu pot fi descompuse în două relaţii dar pot fi descompuse în **trei, patru sau mai multe relaţii fără a pierde informaţii**.
- Pentru a obţine descompuneri L-*join* în trei sau mai multe relaţii, s-a introdus conceptul de ***join*-dependenţă sau dependenţă la compunere** (JD).

Forma normală 5 (FN5)

- Fie $\{R_1, R_2, ..., R_p\}$ o mulţime de scheme relaţionale care nu sunt disjuncte şi a căror reuniune este R.
- R satisface ***join*-dependenţa** $*\{R_1, R_2, ..., R_p\}$ dacă la fiecare moment are loc egalitatea:

  $$R = \text{JOIN}(\Pi_{\alpha 1}(R), \Pi_{\alpha 2}(R), ..., \Pi_{\alpha p}(R))$$

  unde α_k reprezintă mulţimea atributelor corespunzătoare lui $R_k (1 \leq k \leq p)$.
- *Join*-dependenţa $*\{R_1, R_2, ..., R_p\}$ are loc în R, dacă $R_1, R_2, ..., R_p$ este o descompunere L-*join* a lui R.
- Pentru $p = 2$ se regăseşte multidependenţa.
- O *join*-dependenţă $*\{R_1, R_2, ..., R_p\}$ în care una dintre R_i este chiar R, defineşte o *join*-dependenţă trivială.

Forma normală 5 (FN5)

- *Join*-dependenţa **generalizează multidependenţa**.
- Într-adevăr, multidependenţa $X \rightarrow\rightarrow Y$ în relaţia $R(X, Y, Z)$ (deci şi $X \rightarrow\rightarrow Z$), corespunde *join*-dependenţei $*\{X \cup Y, X \cup Z\}$.
- Invers, *join*-dependenţa $*\{R1, R2\}$ corespunde multidependenţei $R1 \cap R2 \rightarrow\rightarrow R1 - (R1 \cap R2)$.
- Formal, o relaţie R este în FN5 dacă şi numai dacă orice *join* dependenţă $*\{R_1, R_2, ..., R_p\}$ care are loc în R fie este trivială, fie conţine o supercheie a lui R (adică, o anumită componentă R_i este o supercheie a lui R).
- Cu alte cuvinte, o relaţie R este în FN5 dacă orice *join*-dependenţă definită pe R este implicată de cheile candidat ale lui R.

Forma normală 5 (FN5)

- Între mulţimile de atribute *X*, *Y* şi *Z* din cadrul relaţiei *R* există o *join* dependenţă dacă există multidependenţe între fiecare dintre perechile de mulţimi (*X*, *Y*), (*Y*, *Z*) şi (*X*, *Z*).
- Aducerea în FN5 prin eliminarea join dependenţelor!

Forma normală 5 (FN5)

FN5 îşi propune eliminarea redundanţelor care apar în relaţii *m:n* dependente.
S-a arătat că o relaţie de tip 3 este diferită de trei relaţii de tip 2. Există totuşi o excepţie, şi anume, **dacă relaţia este ciclică**

Intuitiv, o relaţie *R* este în forma normală 5 dacă şi numai dacă:

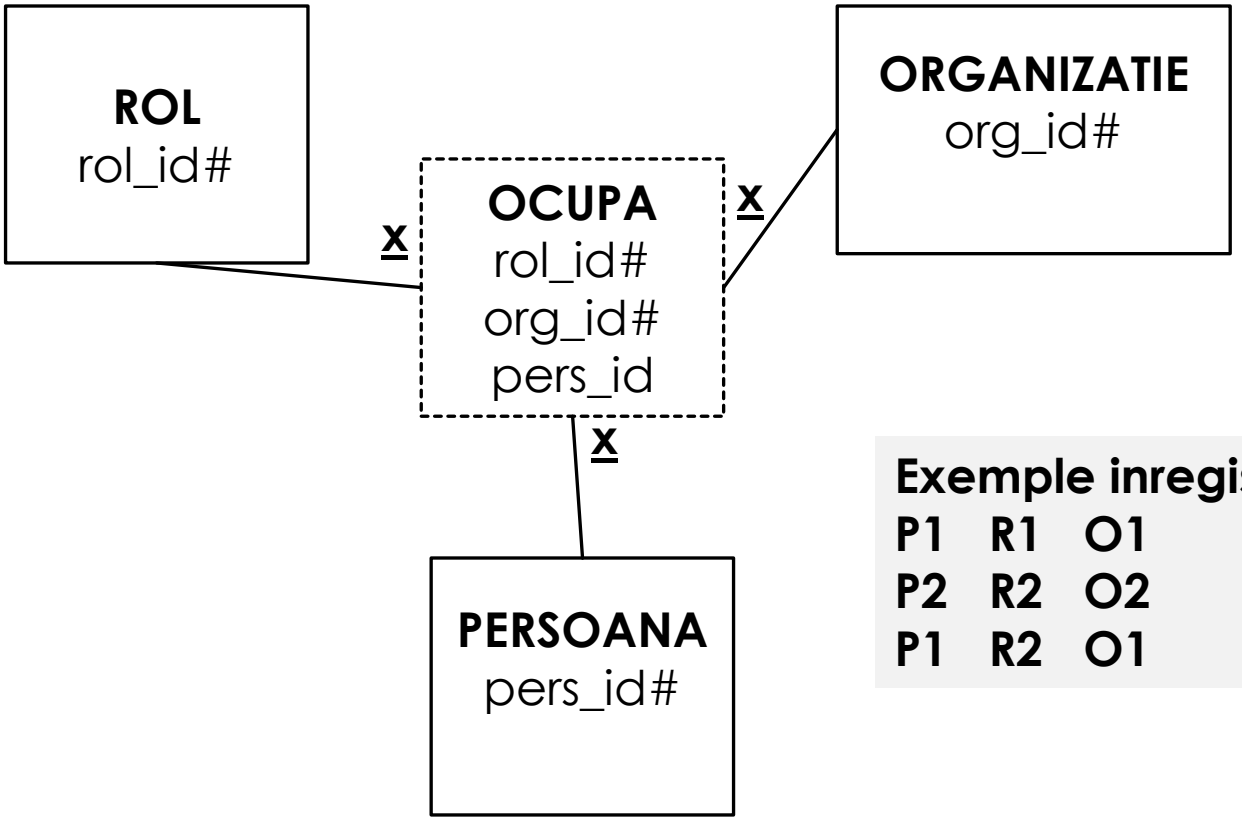
- relaţia este în FN4;
- **nu conţine dependenţe ciclice**;

Exemplu:
O persoana poate ocupa mai multe roluri si poate face parte din mai multe organizatii.

Forma normală 5 (FN5)

Exemplu:

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

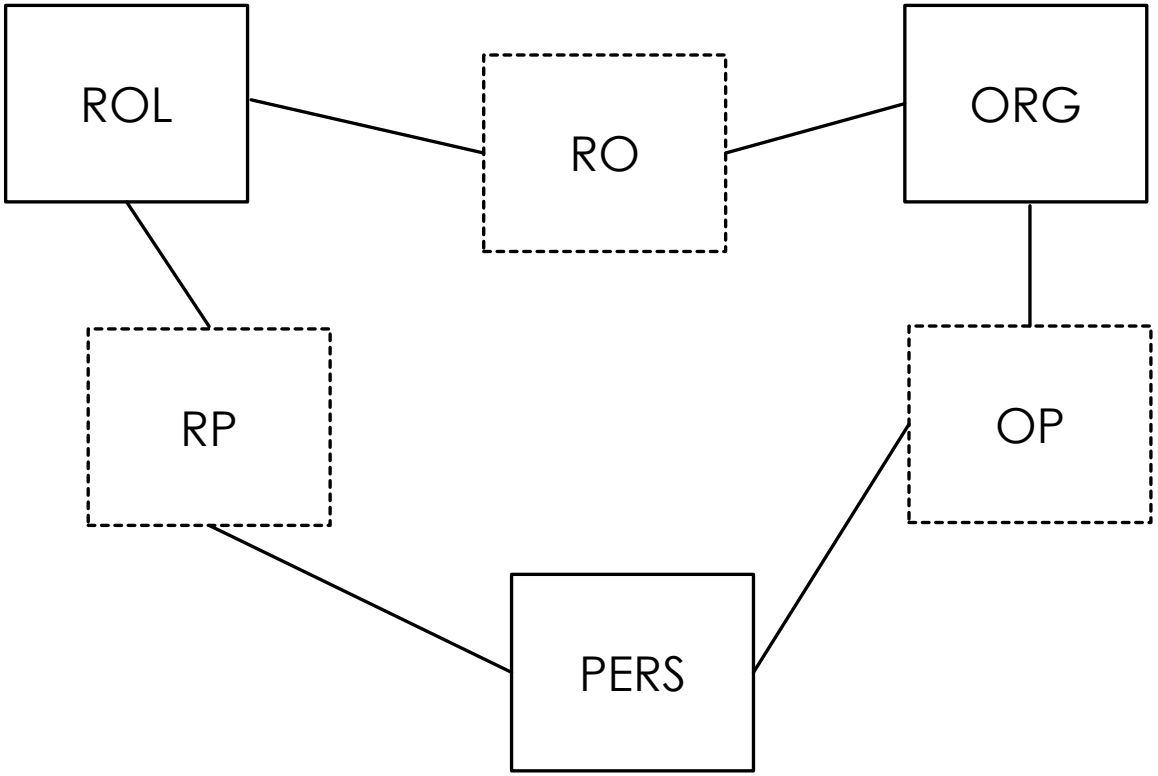


Exemple inregistrari:

P1	R1	O1
P2	R2	O2
P1	R2	O1

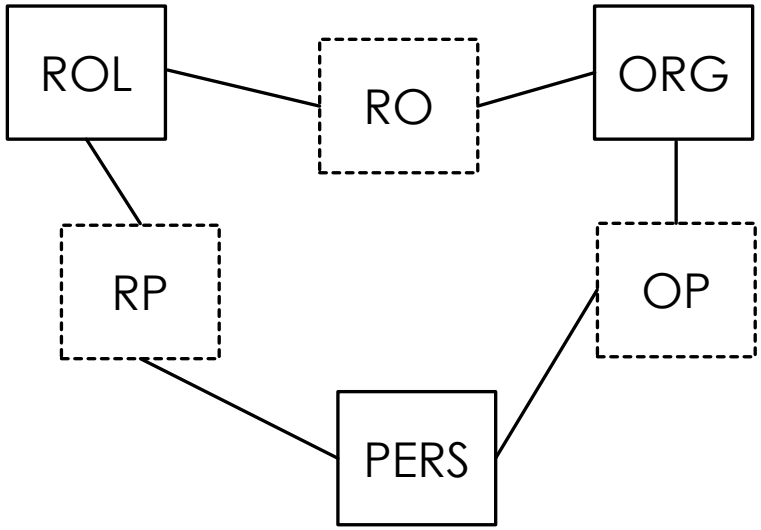
Forma normală 5 (FN5)

- Relatia de tip 3 precedenta poate fi echivalenta cu 3 relatii de tip 2 doar daca aceste relatii de tip 2 sunt ciclice.



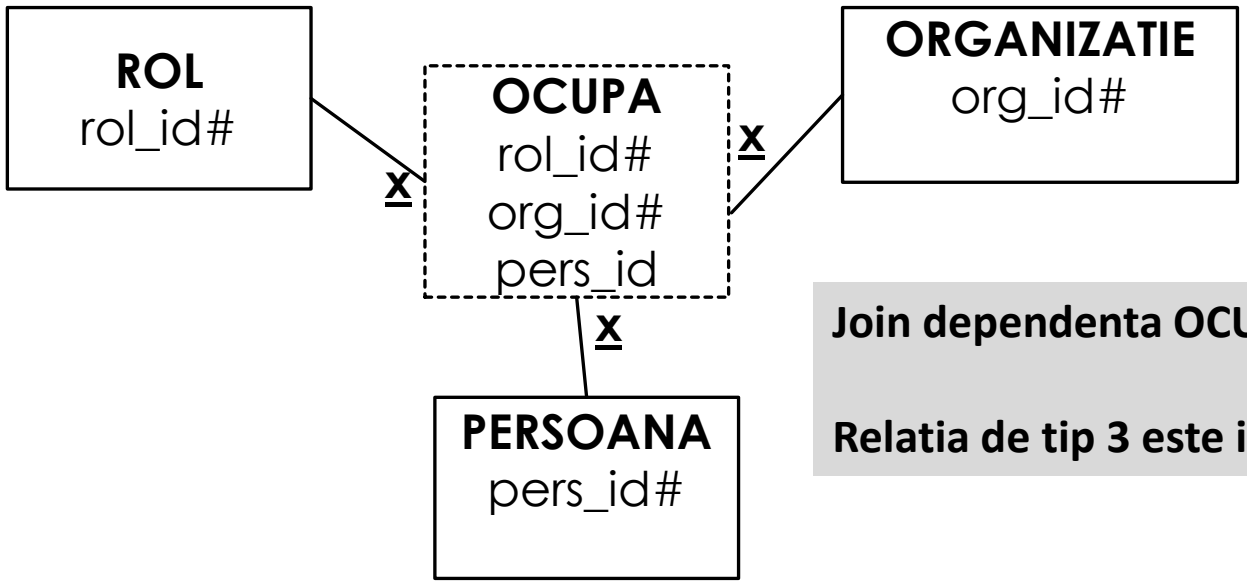
Forma normală 5 (FN5)

- Relatia de tip 3 precedenta poate fi echivalenta cu 3 relatii de tip 2 doar daca aceste relatii de tip 2 sunt ciclice.



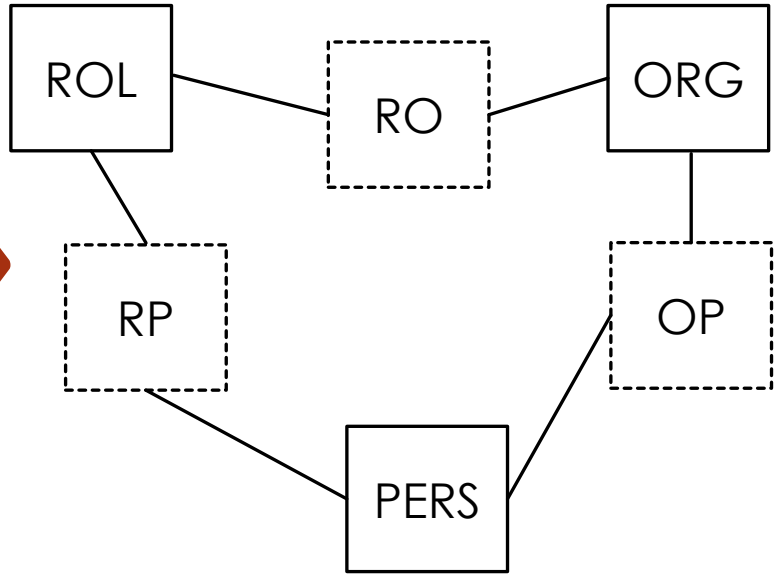
- Relatia fiind ciclica, atunci cand facem toate join-urile vom obtine un rezultat echivalent cu cel obtinut din relatia de tip 3. O relatie pentru a fi in FN5 trebuie sa fie in FN4 si **sa nu contina dependente ciclice**. Se observa ca cele 3 relatii de tip 2 compun o diagrama care contine dependente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

Forma normală 5 (FN5)



Join dependenta OCUPA = JOIN (RO, RP, OP)

Relatia de tip 3 este in FN5



Concluzii normalizare

1. FN1 → FN2 elimină redundanţele datorate dependenţei netotale a atributelor care nu participă la o cheie, faţă de cheile lui *R*. Se suprimă dependenţele funcţionale care nu sunt totale.
2. FN2 → FN3 elimină redundanţele datorate dependenţei tranzitive. Se suprimă dependenţele funcţionale tranzitive.
3. FN3 → BCNF elimină redundanţele datorate dependenţei funcţionale. Se suprimă dependenţele în care partea stângă nu este o supercheie.
4. BCNF → FN4 elimină redundanţele datorate multidependenţei. Se suprimă toate multidependenţele care nu sunt şi dependenţe funcţionale.
5. FN4 → FN5 elimină redundanţele datorate dependentei ciclice. Se suprimă toate *join*-dependenţele care nu sunt implicate de o cheie.
6. BCNF, FN4 şi FN5 corespund la regula că orice determinant este o cheie, dar de fiecare dată dependenţa cu care se defineşte determinantul este alta şi anume dependenţa funcţională, multidependenţa sau *join*-dependenţa).
7. Descompunerea unei relaţii FN2 în FN3 conservă datele şi dependenţele, pe când descompunerea unei relaţii FN3 în BCNF şi, respectiv, a unei relaţii 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 care trebuie 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).
- In 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

A) Obiectivul denormalizării constă în **reducerea numărului de join-uri** care trebuie efectuate pentru rezolvarea unei interogări, prin realizarea unora dintre acestea în avans, ca făcând parte din proiectarea bazei de date.

DENORMALIZARE

B) 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 in care sunt stocate produse, numit **PRODUSE**. Produsele au si un atribut ***greutate.*** Aceasta coloana contine valori repetitive, aceeasi greutate definind mai multe produse. In acest caz, daca in baza de date exista un tabel separat in care se afla greutatea impreuna cu id-ul produsului caruia ii corespunde, este necesar procesul de **denormalizare** in urma caruia atributul ***greutate*** se va plasa in tabelul **PRODUSE** deoarece nu este eficient ca acest atribut sa se afle intr-un tabel separat.

DENORMALIZARE

Exemple:

- In cazul in care avem **User** si **Profile** intre care exista relatia one-to-many (un user are un singur profil), iar tabelul profil contine atributele ***nume*** si ***prenume***, atunci se poate realiza o denormalizare in urma careia cele doua atribute se vor plasa in tabelul User deoarece operatiile de join sunt costisitoare, fiind mult mai eficienta in acest caz mutarea coloanelor in tabelul User. Pe de alta parte, daca tabelul Profile contine mai multe atribute, atunci se vor pastra ambele tabele in baza de date.

DENORMALIZARE

- Se presupune că informațiile despre **Creatori**, **Vestimentatii** și **Accesorii** se reprezintă printr-un tabel. In această structură fizică interogarea « **Obținerea informațiilor despre creatorii care oferă vestimentații cu accesorii din margele** » se rezolvă cu ușurință.
- Interogarea « **Obținerea informațiilor despre creatorii din Sibiu** » va prezenta performanțe mai reduse în această structură fizică, decât dacă am fi menținut trei tabele de bază pentru cele 3 entități.

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 rata a interogarilor foarte ridicata, denormalizarea poate constitui o opțiune viabilă.
- Nu există reguli fixe pentru stabilirea situatiilor in care este indicată denormalizarea relațiilor.