

Capitolul 6. Dependente multivaloare și de joncțiune. Formele normalizate 4 și 5

Formelele normalizate 4 și 5 constituie cea mai dificilă parte din procesul de normalizare a unei baze de date. Aceasta este vestea rea. Vestea bună este că rareori proiectanții bazei trebuie să ajungă în acest stadiu al discuțiilor, mai ales în ceea ce privește 5NF, deoarece majoritatea cazurilor practice pot fi rezolvate onorabil prin aducerea bazei în 3NF/BCNF sau, uneori, în 4NF.

Dependențele funcționale discutate până în acest paragraf sunt relativ ușor de identificat și explicat. În prezentul capitol vor fi prezentate două tipuri de dependențe mult mai greu de digerat. Este și motivul pentru care majoritatea lucrărilor dedicate normalizării bazelor de date relaționale, fie le prezintă rapid, fără prea multe explicații, păstrând un aer discret, fie nu le prezintă deloc, ceea ce le pune la adăpost de orice eroare.

6.1. Dependente multivaloare

Dependența *multivaloare* sau *multivalorică* și, implicit, a patra formă normală au fost introduse la câțiva ani după dependențele funcționale și primele trei forme normale, prin 1977 de către Ronald Fagin¹. Fie relația $R \{A_1, A_2, \dots, A_n\}$ și X, Y și Z trei subansambluri de atribute ale ansamblului $\{A_1, A_2, \dots, A_n\}$. Există o dependență multi-valoare (DMV) între X și Y dacă și numai dacă:

- la fiecare apariție (valoare) a lui X poate fi asociată una sau mai multe apariții (valori) ale lui Y ,
- această asociație nu depinde de aparițiile lui Z .

Se notează $X \twoheadrightarrow Y$ sau $X \twoheadrightarrow Y \mid Z$ și se spune că atributul Y este *multidependent față de atributul X* sau că *atributul X multidetermină pe Y* . Deși Fagin nu preciza acest lucru în lucrarea sa², Date este de părere că se poate pune problema unei DMV numai în condițiile existenței a minimum trei atribute³. De asemenea, dacă există DMV $X \twoheadrightarrow Y \mid Z$, atunci, obligatoriu și $X \twoheadrightarrow Z \mid Y$.

Dependența multi-valoare este percepută și prezentată diferit, chiar, pe alocuri, eronat. Pentru Rebecca Riordan, Florescu s.a., simpla repetare a unui atribut este un indicator cât se poate de clar că avem de-a face cu o DMV, ceea ce nu este prrea

¹ [Fagin77]

² Similară lui Fagin este și abordarea din Garcia-Molina, Ullman și Widom [Garcia-Molina s.a.02], p.119

³ [Date86], p.384

adevărat. Spre exemplu, R. Riordan ia un exemplu identic celui din figura 6.1⁴. Ceea ce nu precizează autoarea ține de faptul că există $DMV \text{ DenProdus} \longrightarrow \longrightarrow \text{UnitateMăsură}$ numai în condițiile în care *fiecare furnizor vinde toate produsele în toate unitățile de măsură ale produsului respectiv !!!*

R

DenProdus	NumeFurnizor	UnitateMasura
Produs 1	Furnizor 1	cutii 250 ml
Produs 1	Furnizor 1	cutii 500 ml
Produs 1	Furnizor 1	cutii 1 l
Produs 2	Furnizor 2	cutii 250 ml
Produs 2	Furnizor 2	cutii 500 ml
Produs 2	Furnizor 2	cutii 1 l
Produs 3	Furnizor 3	cutii 250 ml
Produs 3	Furnizor 3	cutii 500 ml
Produs 3	Furnizor 3	cutii 1 l

Figura 6.1. Exemplul R. Riordan pentru DMV

În [Florescu s.a.99] lucrurile sunt și mai departe de adevăr: “în cazul în care unui atribut îi corespund mai multe valori ale altui atribut este vorba de o dependență multivaloare”⁵, speța prezentată fiind $Client \longrightarrow \longrightarrow NrComandă$, pe motiv că unui client îi corespund mai multe comenzi. Orice comentariu este de prisos.

Nici în lucrarea lui Octavian Bâscă DMV nu stă chiar pe roze. Definiția teoretică merită a fi reținută, anume că $X \longrightarrow \longrightarrow Y$ dacă și numai dacă, oricare ar fi tuplurile u și v pentru care $u[X] = v[X]$, relația conține obligatoriu și tuplurile t și s care îndeplinesc condițiile⁶:

- $t[X] = u[X]$,
- $t[Y] = u[Y]$,
- $t[R-X-Y] = v[R-X-Y]$,
 - $s[X] = v[X]$,
 - $s[Y] = v[Y]$,
 - $s[R-X-Y] = u[R-X-Y]$

Cum aproape întodeauna se verifică postulatul “teroria ca teoria, dar practica ne omoară”, exemplul $R \{Curs, Profesor, Ora, Sala, Student, AnStudii\}$ în care $Curs \longrightarrow \longrightarrow (Ora, Sala)$ este mult mai puțin elocvent, iar eu îl consider incorect chiar în condițiile restricțiilor formulate.

În schimb, într-una dintre lucrările consacrate (la noi) în materie de baze de date, cea elaborată de un grup de profesori de la ASE București condus de dl. Ion

⁴ [Riordan99], p.40

⁵ [Florescu s.a.] p.99

⁶ [Bâscă97], p.100

Lungu, pentru DMV exemplul este corect⁷, dar nu se prezintă nici o explicație sau definiție a acesteia, adică de ce în acel exemplu $R2\{Curs, Capitol, CadruId\}$ conține o DMV. Iar câteva pagini mai încolo, în paragraful dedicat aducerii relațiilor în 4NF⁸, lucrurile se încețoșează rău, cele două DMV prezentate la pagina 186 nefiind deloc convingătoare, cel puțin pentru mine.

Și exemplul, altminteri clasic (curs - profesor titular - recomandare bibliografică), din [Oprea99] este corect, inclusiv cele trei condiții ale problemei, dintre care cea mai importantă este folosirea de către orice titular de curs a tuturor materialelor bibliografice folosite de ceilalți titulari ai cursului respectiv⁹. Definiția DMV, denumită de autor *dependență multiplă*, este însă destul de laxă: „O dependență multiplă există atunci când trei atribute se află într-o relație (notate A, B, C) și pentru fiecare valoare a lui A există o mulțime bine definită de valori ale lui B și alta bine definită de valori ale lui C. Totuși, mulțimea valorilor lui B este independentă de mulțimea valorilor lui C”¹⁰. Or, sursa problemei este că noțiunea de independență a celor două mulțimi de valori nu este una obișnuită, după cum vom vedea peste nici jumătate de pagină.

Cea mai rezonabilă definiție a DMV o furnizează C.J. Date. Fie relația $R\{A_1, A_2, \dots, A_n\}$ și X, Y și Z trei subansambluri ale ansamblului $\{A_1, A_2, \dots, A_n\}$. Există o dependență multi-valoare între X și Y în următoarea situație: *dacă (x, y, z) și (x, y', z') sunt două tuplele ale relației R , atunci (x, y', z) și (x, y, z') aparțin, de asemenea, lui R .*

Într-o formulare ușor schimbată, Elmasri și Navathe definesc DMV de maniera următoare: dacă în R există două tuple t_1 și t_2 pentru care $t_1[X] = t_2[X]$, atunci există în R alte două tuple, t_3 și t_4 , care satisfac următoarele condiții (notăm prin Z toate celelalte (decât X și Y) atribute din R , adică $(R - (X \cup Y))$):

- $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
- $t_3[Y] = t_1[Y]$ și $t_4[Y] = t_2[Y]$
- $t_3[Z] = t_2[Z]$ și $t_4[Z] = t_1[Z]$ ¹¹.

După cum afirmam mai sus, aspectul contra-intuitiv al definiției DMV ține de faptul că X este independent de Y , ceea ce s-ar traduce prin „ X oarecare, Y oarecare”. O bună explicație în acest sens o găsim în [Dollinger98]¹²: o valoare dată a lui X se găsește în R în combinație (formează tuple) cu fiecare pereche de valori (y, z) din produsul cartezian al mulțimilor $Y \times$ (valorile lui y care apar în combinație

⁷ [Lungu s.a.95], pp.167-168

⁸ [Lungu s.a.95], pp.185-186

⁹ [Oprea99], p.352

¹⁰ [Oprea99], p.353

¹¹ [Elmasri & Navathe 00], p. 514

¹² [Dollinger98], p.163

cu un x dat) și Zx . Aceasta înseamnă ca mulțimile Yx și Zx sunt independente între ele.

Cele mai tentante exemple de DMV sunt cele teoretice, așa cum este cel din figura 6.2.

XYZ		
X	Y	Z
x1	y1	z2
x1	y2	z1
x1	y2	z2
x2	y2	z1
x2	y2	z2
x3	y1	z1
x1	y1	z1

Figura 6.2. Exemplu teoretic de relație în care există DMV

În această relație există nu există nici una din dependențele funcționale:

$X \not\rightarrow Y$

$X \not\rightarrow Z$

$Y \not\rightarrow X$

$Y \not\rightarrow Z$

$Z \not\rightarrow X$

$Z \not\rightarrow Y$

La o examinare mai atentă se observă dependențele multi-valoare $X \twoheadrightarrow Y$ (sau $X \twoheadrightarrow Y \mid Z$) și $X \twoheadrightarrow Z$ (sau $X \twoheadrightarrow Z \mid Y$); aceasta deoarece în condițiile în care există tuplurile

- (x1,y1,z1)

- (x1,y2,z2),

există și tuplurile

- (x1,y1,z2) și

- (x1,y2,z1).

Cel mai dureroasă aspect al dependențelor multivaloare (și cele de joncțiune care vor urma), ține de faptul că, în realitate, acestea trebuie identificate de către cei care fac analiza și proiectarea bazei de date, pe baza legăturilor *semantice* dintre atributele bazei, și nu pe baza unui caz simplist (în cazul nostru, o tabelă cu șapte linii).

Alți copii, alte jucării

Asemănător relației DOTARE_JUCĂRII_1 (figura 3.14), în figura 6.3 este prezentată tabela FCJ în care se evidențiază jucăriile pe care le au copiii câtorva familii. Singura informație despre fiecare jucărie este numele său, astfel încât cheia primară a relației este (Familie, Copil, Jucărie).

FCJ

Familie	Copil	Jucărie
Popescu I Mihai	Ioana	Păpușa "Barbie"
Popescu I Mihai	Răzvan	Tricicletă
Popescu I Mihai	Ioana	Tricicletă
Popescu I Mihai	Răzvan	Păpușa "Barbie"
Vrânceanu V Vasile	Răzvan	Tricicletă
Vrânceanu V Vasile	Răzvan	Trenuleț electric

Figura 6.3. Relația FCJ în care există DMV

Iată restricțiile:

- o familie poate avea mai mult de un copil: Familie ---/--- Copil
- în orice familie pot exista mai multe jucării: Familie ---/--- Jucărie
- un copil poate avea mai multe jucării: Copil ---/--- Jucărie
- două sau mai multe familii pot avea copii cu același (pre)nume:
Copil ---/--- Familie
- de asemenea: Jucărie ---/--- Copil, Jucărie ---/--- Familie

Încercând a depista unele dependențe funcționale cu sursa compusă, vom avea aceleași rezultate:

(Familie, Copil) ---/--- Jucărie
 (Familie, Jucărie) ---/--- Copil
 (Copil, Jucărie) ---/--- Familie

Dacă se aplică restricția: *o aceeași jucărie trebuie cumpărată tuturor copiilor din familia respectivă (altfel iese scandal)*, avem de-a face cu o dependență multi-valoare:

Familie --->--> Copil

Deci, într-o familie oarecare, oricare ar fi copilul și o jucărie, este sigur că oricare alt copil (al aceleiași familii) va avea această jucărie. Implicit, se verifică și dependența multivaloare

Familie --->--> Jucărie

De fapt, se poate scrie:

Familie --->--> Copil | Jucărie sau Familie --->--> Jucărie | Copil

Profesori, cursuri & capitole

Cazul următor este aproape identic celui prezentat de C.J. Date în edițiile sale ale lucrării *An Introduction to Database Systems*¹³. Relația are trei atribute: CPC {Curs, Profesor, Capitol}¹⁴. La o facultate "oarecare", pot exista mai mulți profesori titulari ai unei aceleiași discipline. Dacă presupunem că toți profesorii cad

¹³ Spre exemplu, [Date86], pp. 381-382

¹⁴ La C.J. Date relația era CTX {Curs, Profesor, LucrareBibliografică}

de acord asupra unei programe unice pentru disciplina respectivă, diferențiat fiind numai modul de "tratare" și ponderea fiecărui capitol, putem imagina o situație ca în figura 6.4.

CPC

Curs	Profesor	Capitol
Bazele informaticii economice	Fătu	Arhitectura microcalculatoarelor
Bazele informaticii economice	Airinei	Arhitectura microcalculatoarelor
Bazele informaticii economice	Grama	Arhitectura microcalculatoarelor
Bazele informaticii economice	Fătu	Generații de calculatoare
Bazele informaticii economice	Airinei	Generații de calculatoare
Bazele informaticii economice	Grama	Generații de calculatoare
Baze de date	Airinei	Modelul Entitate-Asociație
Baze de date	Fotache	SQL
Baze de date	Airinei	SQL
Baze de date	Fotache	Modelul Entitate-Asociație
Birotică	Grama	Poșta electronică

Figura 6.4. Relația CPC în care există DMV

Analizăm dependențele dintre atribute:

- Un același curs poate fi "ținut" de mai mulți profesori:
Curs ---/--- Profesor
- Un curs este alcătuit din mai multe capitole: Curs ---/--- Capitol
- Fiecare profesor poate preda mai multe cursuri: Profesor ---/--- Curs
și, implicit, Profesor ---/--- Capitol
- Un același capitol poate apărea în cursuri diferite. De exemplu, un capitol dedicat *Poștei Electronice* poate apărea și în cursul *Bazele informaticii Economice* și în cursul de *Birotică*. Capitolul *Modelul Entitate-Asociație* este în planul cursului de *Baze de date*, dar și al cursului *Analiza sistemelor informatice* (pentru modelul conceptual al datelor). Prin urmare, Capitol ---/--- Curs iar Capitol ---/--- Profesor.

Numai în condițiile unei programe unice, la nivelul facultății, pentru fiecare curs există dependențele multi-valoare:

Curs --->--> Profesor

Curs --->--> Capitol

C.J. Date face o observație foarte bine venită: într-o asemenea relație, pentru a introduce trei profesori și patru capitole ar fi suficiente patru linii, și nu 12 (3×4). Aceasta naște însă câteva probleme: cum se realizează corespondența profesor-capitol, adică ce capitole a elaborat fiecare profesor? Numărul capitolelor fiind mai mare ca al autorilor, pentru acele linii, ce profesori apar? Cum este interpretată fiecare linie în acest caz? Cum se fac actualizările? Dar discuțiile acestea noi le-am purtat deja în capitolul 3.

Relația COMENZI_PRODUSE_MATERIALE

O întreprindere manufacturieră folosește pentru confecționarea unei unități de produs finit o serie de materii prime și materiale, în conformitate cu fișa tehnologică. Producția este pe comenzi. Se poate imagina o tabelă care să conțină informațiile cu privire la ce materiale sunt necesare onorării unei anumite comenzi, COMENZI_PRODUSE_MATERIALE – vezi figura 6.5.

COMENZI_PRODUSE_MATERIALE		
Comandă	Produs	Material
1122	Produs 1	Material 2
1122	Produs 1	Material 4
1122	Produs 2	Material 1
1122	Produs 2	Material 3
1122	Produs 2	Material 4
1123	Produs 1	Material 2
1123	Produs 1	Material 4
1123	Produs 2	Material 1
1123	Produs 2	Material 3
1123	Produs 2	Material 4
1123	Produs 3	Material 2
1123	Produs 3	Material 4

Figura 6.5. Relația COMENZI_PRODUSE_MATERIALE

În această tabelă, este imperios necesar ca, atunci când un produs apare într-o comandă, să se introducă câte o linie pentru fiecare material ce intră în fabricația produsului respectiv. Prin urmare, există dependențele multi-valoare:

Produs \longrightarrow Comandă | Material

Din nou despre bibliotecă

Cel mai convingător exemplu de dependență multi-valoare este cel din relația BIBLIOTECĂ_TUPLURI_NOI_SOLUȚIA_3 din figura 3.7 pe care o preluăm așa cum am lăsat-o în figurile 5.21 (graful) și 5.22 (conținutul în 3NF-BCNF). În relația TITLURI_AUTORI_CUVINTECHEIE2 fiecare linie se referă la un autor și un subiect (cuvânt cheie) pentru o carte (ISBN). Practic, pe baza acestei tabele suntem în măsură să obținem informații de genul:

- ce cărți a scris un anume autor,
- despre ce subiecte a scris un anume autor,
- care sunt autorii unei cărți,
- ce subiecte sunt tratate într-o carte,
- care sunt cărțile în care apare tratat un anume subiect,
- care sunt autorii care au scris despre un subiect dat.

Pentru ca nici una dintre aceste informații să nu se piardă, există un preț adecvat. Dacă, pentru o carte, apare o nouă sintagmă (cuvânt cheie), în relație trebuie introdus nu un tuplu, ci n , unde n este egal cu numărul autorilor acelei cărți. Pentru o carte cu 5 autori și 10 cuvinte cheie, în relația TITLURI_AUTORI_CUVINTECHEIE2 trebuie introduse nu mai puțin de 50 de linii.

Pe baza faptului că pentru o carte, un autor trebuie să apară în combinație cu toate sintagmele tratate și, vice-versa, orice sintagmă apare în combinație cu toți autorii cărții, există DMV:

$$\begin{array}{l} \text{ISBN} \longrightarrow \text{Autor} \mid \text{CuvântCheie} \\ \text{ISBN} \longrightarrow \text{CuvântCheie} \mid \text{Autor} \end{array}$$

Prin analogie cu dependența funcțională, se poate defini *dependența multi-valoare elementară* pentru cazul în care:

- $X \longrightarrow Y$
- nu există nici un subansamblu X' și Y' de attribute din X și Y astfel încât $X' \longrightarrow Y'$.

Dependențele multivaloare sunt o generalizare a celor funcționale; orice DF este o DMV, reciprocă nefiind valabilă (există DMV care nu sunt DF). Aceasta deoarece o DF este o DMV în care "setul" valorilor dependente corespunzătoare valorii unui determinant dat este întotdeauna o singură valoare¹⁵.

Poate ar mai trebui evocate câteva exemple plauzibile pentru dependențele multivaloare: cel al lui Robert Dollinger privind clienții și clientele unor agenții matrimoniale - $R \{ \text{Agenție}, \text{Nume_Femeie}, \text{Nume_Bărbat} \}$ care, însă, din păcate, nu este operațional într-o serie de țări mai „liberale” etc. În aceeași lucrare este un exemplu foarte bun legat de cursurile și limbile străine în care pot fi predate, ținând seama că orice profesor poate ține fiecare curs pe care îl predă în toate limbile pe care le cunoaște: $R \{ \text{Profesor}, \text{Disciplină}, \text{Limbă} \}$.

6.2.A patra formă normală

Începeam acest capitol nu prea politicoși cu privire la 4NF și 5NF, exprimându-ne, ce-i drept, cât am putut de discret, rezervele legate de utilitatea celor două forme normale. Deși multe lucrări, ca și prezenta, le tratează la pachet, incidența lor practică este, totuși, diferită. Aducem ca argument articolul publicat de Margaret Wu¹⁶ în care autoarea afirmă că, în timp ce majoritatea universitarilor și autorilor din zona normalizării tind să neglijeze 4NF și 5NF, un studiu întreprins într-o serie de firme ce foloseau (la începutul anilor '90) aplicații cu baze de date a scos la iveală că mai mult de 20% dintre schemele bazelor de date analizate violau 4FN. Prin comparație, Wu nu a identificat nici un caz de violare a 5NF.

În lucrarea menționată în paragraful anterior, Ronald Fagin formulează o teoremă conform căreia o relație R cu trei attribute A , B și C poate fi descompusă fără pierdere de informații în două proiecții ale sale: $R1\{A, B\}$ și $R2\{A, C\}$ dacă și numai dacă există dependență multivaloare $A \longrightarrow B \mid C$. Aceasta este teorema care ne va permite aducerea unei relații în 4NF.

¹⁵ Pentru o demonstrație în extenso - vezi și [Garcia-Molina 02] pp.121-122

¹⁶ [Wu92]

O relație este în 4NF dacă și numai dacă:

1. Este în BCNF,
2. Toate dependențele care se manifestă în cadrul său sunt dependențe funcționale, altfel spus, eventualele dependențe multi-valoare conținute sunt, de fapt, dependențe funcționale.

Vom lua pe rând exemplele prezentate în paragraful anterior și vom îndeplini formalitatea aducerii lor în 4NF. Dar, înainte de asta, trebuie să amintim faptul că 4NF a avut parte de un tratament similar DMV, iar confuziile nu sunt chiar izolate. Spre exemplu, în [Lungu s.a.95] există o definiție pe care nu o putem trata cu prea mare îngăduință: o relație este în 4NF dacă în cadrul ei nu se manifestă mai mult de o dependență multi-valoare¹⁷. Eu, unul, nu am priceput !

Începem cu cel mai îndrăgit exemplu, primul. Relația XYZ din figura 6.2 prezintă, așa cum am discutat, dependența multi-valoare $X \twoheadrightarrow Y \mid Z$, în virtutea căreia o putem descompune în două: $XY \{X, Y\}$ și $XZ \{X, Z\}$ ca în figura 6.6.

XY		XZ	
X	Y	X	Z
x1	y1	x1	z1
x1	y2	x1	z2
x2	y2	x2	z1
x2	y1	x2	z2
x3	y1	x3	z1

Figura 6.6. Aducerea relației XYZ în 4NF

Pe același calapod, relația FCJ (figura 6.3) se descompune astfel încât cele două relații din figura 6.7 nu mai conțin dependențe multi-valoare (oricum, spre deosebire de alte lucrări, noi am mers pe ideea lui Date potrivit căreia, ca să existe o DMV, e nevoie de minimum trei atribute).

FC		FJ	
Familie	Copil	Familie	Jucărie
Popescu I Mihai	Ioana	Popescu I Mihai	Păpușa "Barbie"
Popescu I Mihai	Răzvan	Popescu I Mihai	Tricicletă
Vrânceanu V Vasile	Răzvan	Vrânceanu V Vasile	Tricicletă
		Vrânceanu V Vasile	Trenuleț electric

Figura 6.7. Relația FCJ în 4NF

Relația COMENZI_PRODUSE_MATERIALE

Pe baza dependențelor multivaloare prezentate în paragraful precedent, relația COMENZI_PRODUSE_MATERIALE (figura 6.5) se descompune ca în figura 6.8.

¹⁷ [Lungu s.a. 95], p.170

COMENZI_PRODUSE		PRODUSE_MATERIALE	
Comandă	Produs	Produs	Material
1122	Produs 1	Produs 1	Material 2
1122	Produs 2	Produs 1	Material 4
1123	Produs 1	Produs 2	Material 1
1123	Produs 2	Produs 2	Material 3
1123	Produs 3	Produs 2	Material 4
		Produs 3	Material 2
		Produs 3	Material 4

Figura 6.8. Aducerea relației COMENZI_PRODUSE_MATERIALE în 4NF

Relația BIBLIOTECĂ

Relația BIBLIOTECĂ_TUPLURI_NOI_SOLUȚIA_3, după cum am văzut în paragraful precedent, conține o dependență multi-valoare prin eliminarea căreia vom aduce relația în 4NF. Deși rareori întâlnită în literatura de specialitate, reprezentarea grafică a DMV în graful dependențelor nu ridică probleme deosebite, mai ales atunci când cele trei atribute (X, Y și Z din definițiile noastre) sunt simple. Iată, în figura 6.9, care ar fi o posibilă reprezentare și modul de decupare al relațiilor în 4NF din graf.

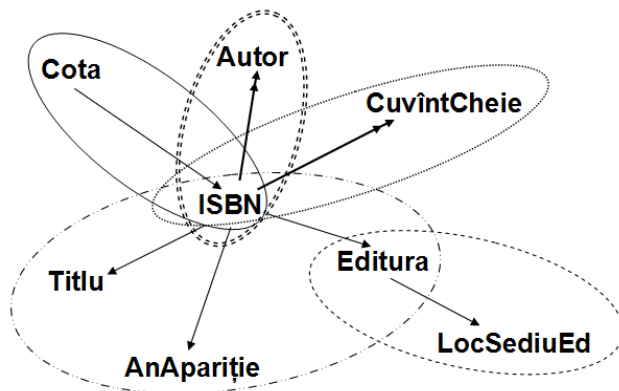


Figura 6.9. Graf ce conține deopotrivă DF și DMV

Conform obiceiului, se va constitui câte o relație pentru fiecare sursă de dependență funcțională (Cota, ISBN și Editura). La aceste relații se va adăuga câte o relație pentru fiecare destinație de dependență multi-valoare (Autor și CuvîntCheie). Schema finală a bazei de date, precum și conținutul relațiilor sunt ilustrate în figura 6.10.

EDITURI

Editura	LocSediuEd
Polirom	Iași

TITLURI

ISBN	Titlu	Editura	AnApariție
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Polirom	2002
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Polirom	2001

EXEMPLARE

Cotă	ISBN
III-13421	973-683-889-7
III-13422	973-683-889-7
III-13423	973-683-889-7
III-10678	973-683-709-2
III-10679	973-683-709-2

TITLURI_AUTORI

ISBN	Autor
973-683-889-7	Marin Fotache
973-683-889-7	Ioan Brava
973-683-889-7	Cătălin Strâmbei
973-683-889-7	Liviu Crețu
973-683-709-2	Marin Fotache

TITLURI_CUVINTECHEIE

ISBN	CuvântCheie
973-683-889-7	baze de date
973-683-889-7	SQL
973-683-889-7	proceduri stocate
973-683-889-7	FoxPro
973-683-889-7	formulare
973-683-889-7	orientare pe obiecte
973-683-889-7	client-server
973-683-889-7	web
973-683-709-2	baze de date
973-683-709-2	algebră relațională
973-683-709-2	SQL

Figura 6.10. 4NF pentru relația BIBLIOTECĂ

Iată că, după atâta amar de capitole și discuții, am reușit să rezolvăm de o manieră acceptabilă problema schemei bazei de date destinate gestionării informațiilor revelante despre cărțile aflate în biblioteca unei facultăți/universități. Este, cred, inutil să mai intrăm în delalii vis-a-vis de reducerea volumului bazei, ca și cvasi-dispariția oricărei forme de anomalie la inserarea, modificarea sau ștergerea unei linii în oricare dintre cele cinci tabele.

Baza de date FILMOGRAFIE

Discuțiile despre această bază de date au rămas suspendate încă din capitolul 3. În paragraful 3.4 semnalam necazurile pricinuite de valorile nule ale cheii primare în condițiile în care o parte dintre premii era acordată la nivel de film (*Oscar pentru*

regie, Oscar pentru cel mai bun film etc.), iar altă parte la nivel de actor pentru prestația dintr-un film (Oscar pentru cel mai bun actor/actriță în rol principal/secundar). La acel moment singura soluția era constituirea a două tabele dedicate premiilor.

Pentru ceea ce urmează considerăm atributele din tabelul 3.1. (paragraful 3.2) între care încercăm să determinăm dependențele funcționale și multivaloare. Astfel, atributul `IdFilm` identifică fără ambiguitate orice film, așa că avem următoarea serie de DF:

- (1) `IdFilm` \longrightarrow `TitluOriginal`
- (2) `IdFilm` \longrightarrow `TitluRO`
- (3) `IdFilm` \longrightarrow `AnLans`

Deoarece un film poate fi încadrat la mai multe genuri (vezi figura 2.5) și avea mai mulți producători, regizori, actori și premii:

`IdFilm` \longrightarrow `Gen`
`IdFilm` \longrightarrow `Producător`
`IdFilm` \longrightarrow `Regizor`
`IdFilm` \longrightarrow `Actor`
`IdFilm` \longrightarrow `DenPremiu`

Un premiu se decernează într-o localitate (ex. *Oscarul* la Hollywood, *Leul de argint* la Veneția, *Ursul de aur* la Berlin etc.):

- (4) `DenPremiu` \longrightarrow `LocDecernare`

Premiile au, de obicei, mai multe categorii (ex. *Oscarul* are categoriile: *Cel mai bun film*, *Cel mai bun film străin*, *Efecte speciale*, *Regie* etc.) și se acordă de mulți ani:

`DenPremiu` \longrightarrow `Categorie`
`DenPremiu` \longrightarrow `AnPremiu`

Ne punem problema reflectării prin dependențe a distribuției într-un film. După cum am văzut mai sus (de fapt, știam de mult), într-un film sunt distribuiți mai mulți actori; însă și un actor poate juca în multe filme - `Actor` \longrightarrow `IdFilm`. Am avea de ales între dependențele:

$(\text{IdFilm}, \text{Actor}) \longrightarrow \text{Rol}$ (poate era mai nimerit ca atributul să se numească `Personaj`) sau

$(\text{IdFilm}, \text{Rol}) \longrightarrow \text{Actor}$.

Dumneavoastră pe care ați alege-o ? Ei bine, prima n-ar fi chiar indicată, deoarece există filme în care un actor joacă un dublu rol (vă mai amintiți *Zmeura de aur* pentru dublul rol al lui Leonardo di Caprio din filmul *Masca de fier*, în care îl interpretează pe regele Ludovic al XIV-lea și pe fratele regelui, unul "original" iar celălalt "mascatul" ?). În acest situații valorile grupului $(\text{IdFilm}, \text{Actor})$ sunt identice, în timp ce valorile atributului `Rol` sunt diferite, ceea ce infirmă dependența. Privitor la cea de-a doua dependență, pericolul invalidării ei ar apărea în situația când într-un film același rol este jucat de mai mulți actori. Din fericire rolurile nu sunt tocmai așa. Chiar și în filmele biografice, sau marile saga de familie, diferitele

etape din viața unui personaj (copil, adolescent etc.) sunt marcate de roluri diferite. Așa că numărăm a cincea dependență a relației .

(5) (IdFilm, Rol) \longrightarrow Actor

Cel puțin la fel de interesant stau lucrurile și cu premiile. Faptul că *Globul de aur* (DenPremiu) pentru *Cea mai bună imagine* (Categorie) a fost decernat în 1988 (AnPremiu) filmului *Mai bine nu se poate* (titlul original - *As Good As It Gets*), film ce are codul (IdFilm) 11899 în baza noastră de date, ne-ar determina să scriem dependența în forma: (DenPremiu, Categorie, AnPremiu) \longrightarrow IdFilm.

Ce ne facem, însă, când un premiu la aceeași categorie este acordat simultan la două filme ? Nu e cazul Oscarurilor, dar, spre exemplu, câte o *mențiune specială din partea juriului* (Categorie) la ediția din 2007 (AnPremiu) a festivalului *Ursul de aur* (DenPremiu) ar pute fi acordată filmelor cu identificatorii 7898843, 8967883 și 9312299. Dependența este aruncată în aer. Soluția este însă pe-aproape: un film nu poate lua un același premiu, la aceeași categorie, în ani diferiți, așa că schimbăm poziția a două atribute:

(6) (DenPremiu, Categorie, IdFilm) \longrightarrow AnPremiu.

Dependența de mai sus nu rezolvă problema premiilor acordate pentru prestațiile actricești, în care laureat este un actor pentru prestația dintr-un film. Am putea alege între varianta:

(DenPremiu, Categorie, AnPremiu, IdFilm) \longrightarrow Actor

și:

(DenPremiu, Categorie, AnPremiu, Actor) \longrightarrow IdFilm.

Prima variantă este minată dacă într-un an un premiu (la aceeași categorie) este acordat simultan pentru doi actori care joacă în același film. Situația nu pare a fi prea frecventă, dar nici imposibilă, dacă ne gândim la festivalurile mai puțin poleite decât *Oscar*, *Palme D'Or* etc. A doua variantă are și ea un competitor, croit după analogia cu dependența (6): (DenPremiu, Categorie, IdFilm, Actor) \longrightarrow AnPremiu. Inițial eram tentat de această ultimă formă, deoarece era similară celei care reflecta celelalte tipuri de premii. Există, însă, o categorie de premii ce crează o serie de dureri de cap: premiile pentru întreaga carieră. Acestea nu sunt legate de un film anume, ci răsplătesc o carieră, mai lungă sau mai scurtă, de cineast. Nefiind legată de un film, sursa dependenței, pe baza căreia vor constitui o relație, ar conține în aceste situații o valoare NULL (pentru IdFilm), ceea ce e inadmisibil pentru o componentă a cheii primare. Cu toate acestea, vom păstra această dependență, întrucât premiul pentru întreaga carieră nu se acordă numai actorilor, ci și regizorilor (teoretic și altor profesioniști legate de industria/arta cinematografică), iar dacă optăm pentru a doua variantă tot nu scăpăm de valori nule, e drept, pentru un atribut necheie. Ca să tranșăm discuția, întrucât baza de date se referă la informații despre filme, nu ne propunem să preluăm premiile individuale, așa că ne declarăm mulțumiți de dependența:

(7) (DenPremiu, Categorie, IdFilm, Actor) \longrightarrow AnPremiu

Au rămas neluate în seamă trei atribute, *Producător*, *Regizor* și *Gen*. Intuiția ne spune că este imposibil de stabilit vreo dependență între ele, chiar dacă se referă la același film. Pe de altă parte, cele trei atribute par cât se poate de independente unele de altele. Verificăm, mai întâi, cerința DMV pentru primele două. Dacă filmul X cu id-ul 234556 are, printre regizori, pe Reg1 și Reg2 și cel puțin doi producători, Prod1 și Prod2, atunci trebuie să verificăm că, dacă attributele ar apărea într-o relație de tip R {IdFilm, Producător, Regizor}, oricare dintre tuplurile următoare sunt posibile:

RELx

IdFilm	Producător	Regizor
...
234556	Prod1	Reg1
234556	Prod2	Reg2
234556	Prod1	Reg2
234556	Prod2	Reg1
...

Figura 6.11. Raportul dintre atributele *Producător* și *Regizor*

Prezența ultimelor două tupluri în relația din figura 6.11 pare a fi agreată, nu însă și necesară. Schema de mai sus, însă, trebuie să furnizeze răspunsul corect la întrebarea *Care sunt filmele în care producătorul Prod1 a lucrat cu regizorul Reg1* ? Dacă ultimele două tupluri nu ar fi prezente în relație, atunci rezultatul furnizat de interogarea:

```
SELECT IdFilm
FROM RELx
WHERE Producător = 'Prod1' AND Regizor='Reg1'
```

este incorect, singura soluție acceptabilă fiind cea "ocolitoare":

```
SELECT IdFilm FROM RELx WHERE Producător = 'Prod1'
INTERSECT
SELECT IdFilm FROM RELx WHERE Regizor='Reg1'
```

sau, firește:

```
SELECT IdFilm FROM RELx WHERE Producător = 'Prod1'
AND IdFilm IN
(SELECT IdFilm FROM RELx WHERE Regizor='Reg1')
```

Faptul că, pentru a-și păstra viabilitatea informațională, relația trebuie să conțină toate cele patru tupluri constituie motiv suficient pentru a "trâmbița" dependența multivalorică:

IdFilm →→ Producător | Regizor

Pe același calapod, pornind de la răspunsul la întrebarea *Ce genuri de filme a produs/regizat un anumit cineast* ?, putem demonstra că:

IdFilm →→ Producător | Gen sau IdFilm →→ Regizor | Gen

Forma finală a grafului dependențelor funcționale și multivalorice ar putea fi cea din figura 6.12.

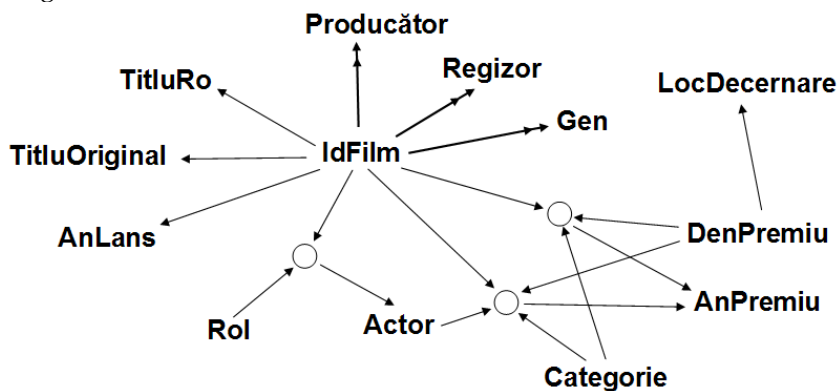


Figura 6.12. Graful dependențelor pentru BD FILMOGRAFIE

Urmează partea cea mai frumoasă - decuparea relațiilor din graf. Sunt două surse simple de DF - DenPremiu și IdFilm și trei surse compuse - (IdFilm, Rol), (IdFilm, DenPremiu, Categorie) și (IdFilm, Actor, DenPremiu, Categorie), așa că cinci relații vor fi obținute din dependențele funcționale. La acestea se mai adaugă trei corespunzătoare dependențelor multi-valoare. Iată, deci, schema finală:

PREMII_DENUMIRI {DenPremiu, LocDecernare}

FILME {IdFilm, TitluOriginal, TitluRO, AnLans}

DISTRIBUȚIE {IdFilm, Rol, Actor}

PREMII_FILME {IdFilm, DenPremiu, Categorie, AnPremiu}

PREMII_INTERPRETARE {IdFilm, Actor, DenPremiu, Categorie, AnPremiu}

PRODUCĂTORI {IdFilm, Producător}

REGIZORI {IdFilm, Regizor}

FILME_GENURI {IdFilm, Gen}

Revenind preț de o frază la discuția legată de producători/regizori/genuri, acum răspunsul la întrebarea *Care sunt filmele în care producătorul Prod1 a lucrat cu regizorul Reg1* ? poate fi obținut doar parcurgând ambele relații, PRODUCĂTORI și REGIZORI, iar corectitudinea rezultatului este garantată:

```
SELECT IdFilm FROM PRODUCĂTORI WHERE Producător = 'Prod1'
INTERSECT
SELECT IdFilm FROM REGIZORI WHERE Regizor='Reg1'.
```

Relația DOTARE_JUCĂRII

Până aici, aducerea în 4NF a unei baze de date prin identificarea și eliminarea dependențelor multivaloare nu a generat chestiuni insurmontabile, iar operațiunile se simplifică și mai mult dacă se recurge la graful dependențelor funcționale și multivaloare. Lucrurile se pot complica fără prea mare efort, dacă în relația DOTARE_JUCĂRII_1 reprezentată în figura 3.14, ce are cheie primară combinația (CodFamilie, Copil, Jucărie), presupunem că, într-o familie, orice jucărie este cumpărată în n exemplare, unde n reprezintă numărul copiilor din acea familie (prin urmare, se verifică dependența multivaloare dintre CodFamilie, și Copil/Jucărie). Graful celor două tipuri de dependențe, cel din figura 6.13, este diferit în funcție de modalitatea de cumpărare a celor n exemplare ale fiecărei jucării.

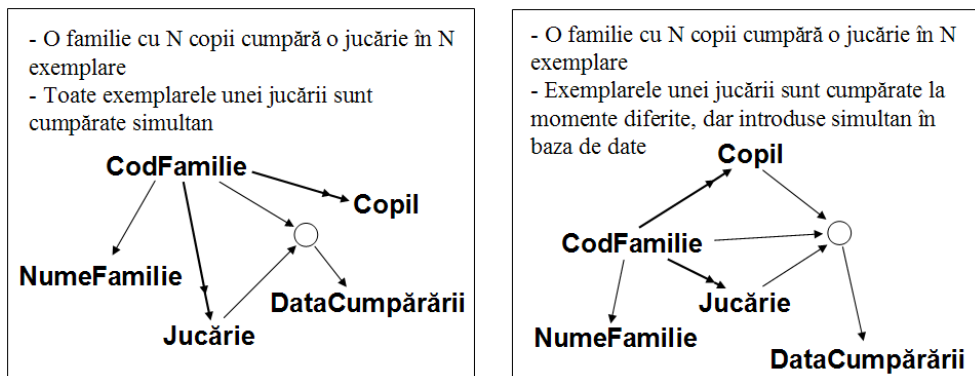


Figura 6.13. Graful dependențelor pentru DOTARE_JUCĂRII în cele două situații

Despre cele două grafuri nu se poate spune că sunt din cale-afară de simple. Să începem cu cel din partea stângă. Există DMV $\text{CodFamilie} \twoheadrightarrow \text{Jucărie} \mid \text{Copil}$, însă sursa și una dintre destinațiile DMV formează sursa unei dependențe funcționale: $(\text{CodFamilie}, \text{Jucărie}) \twoheadrightarrow \text{DataCumpărării}$. De aceea, este rezonabil ca, la descompunerea ce decurge din dependența multi-valoare, să includem DataCumpărării alături de Jucărie (relația JUCĂRII). Schema 4NF a bazei ar fi în acest caz:

FAMILII {CodFamilie, NumeFamilie}

COPII {CodFamilie, Copil}

JUCĂRII {CodFamilie, Jucărie, DataCumpărării}

A doua situație este cu mult mai interesantă. Din rațiuni financiare, o familie cu patru copii poate cumpăra trei trenulețe într-o zi, iar al patrulea trenuleț la

chenezina următoare. Abia după achiziția celui de-al patrulea, se preiau în bază toate trenulețele. Chiar dacă s-au jucat, clandestin sau nu, cu trenulețele înainte de completarea "seriei", copiii știu cu exactitate cui aparține fiecare din cele patru trenulețe. În acest caz, singura dependență funcțională cu sursa compusă este cea care decurge din calitatea de cheie primară a combinației (CodFamilie, Copil, Jucărie). Ei bine, deși conform obiceiului, atunci când într-o relație există o dependență multivaloare, relația se poate descompune (așa cum făcut în acest paragraf), dacă am opta pentru schema de mai sus - FAMILII - COPII - JUCĂRII - am pierde o informație importantă: data în care a fost cumpărată jucăria unui anume copil. Concluzia este deconcertantă: nu întotdeauna când într-o relație avem o dependență multivaloare, ruperea sa reprezintă soluția cea mai indicată, fiind necesară verificarea pierderii sau nu de informații la descompunere ! Schema prin care nu pierdem informații este:

FAMILII {CodFamilie, NumeFamilie}

COPII {CodFamilie, Copil}

JUCĂRII_FAMILII {CodFamilie, Jucărie}

JUCĂRII_COPII {CodFamilie, Jucărie, Copil, DataCumpărării}

Redundanța pare supărătoare, însă nu insuportabilă. În JUCĂRII_FAMILII fiecare tip de jucărie apare o singură dată la o familie, în timp ce în JUCĂRII_COPII îi sunt alocate n linii. Totuși, în condițiile date, JUCĂRII_FAMILII este o relație redundantă. Și relația COPII este una redundantă, singurul său avantaj fiind că elimină o importantă anomalie la inserare, și anume, se permite inserarea în baza de date a unui sau mai multor copii dintr-o aceeași familie, chiar dacă în acea familie nu a fost cumpărată nici o jucărie.

Vom vedea în capitolul viitor că mare parte din confuzia problemei ține de faptul că atât jucăriile, cât și copiii, nu sunt identificabile prin cele două atribute, Jucărie și Copil, iar folosirea cheilor surrogat mai împrăstie din ceață.

6.3.Dependențe de joncțiune

Dacă dependențele multivaloare constituie o raritate în proiectarea bazelor de date, atunci ce să mai vorbim de cele de joncțiune ? Condițiile ce trebuie întrunite pentru a "depista" o dependență de joncțiune și, implicit, de a pune problema aducerii bazei de date în a cincea formă normalizată sunt atât de restrictive, încât în mare parte din lucrările dedicate bazelor de date sau proiectării bazelor de date, 5NF nici nu se mai discută¹⁸, exemplele sunt strict teoretice¹⁹, insuficient explicate²⁰ sau sumare²¹.

¹⁸ Spre exemplu, [Garcia-Molina 02], [Florescu s.a.99], [Connolly&Begg00]

¹⁹ [Lungu s.a.95], [Băscă97]

²⁰ [Celko99]

²¹ [Riordan99], [Fleming&vonHalle89], [Teorey99]

Dependența de joncțiune definită în continuare este legată de numele lui Jorma Rissanen²² și folosește definiția generalizată a joncțiunii unei colecții de relații formulată de Aho, Beeri și Ullman²³. Fie $R \{A_1, A_2, \dots, A_n\}$ o relație și X_1, X_2, \dots, X_m , m subansambluri de atribute din R . Se spune că există o dependență de joncțiune de ordinul m între X_1, X_2, \dots, X_m , notată

$$* X_1 / X_2 / \dots / X_m \text{ sau } * \{X_1, X_2, \dots, X_m\}$$

dacă și numai dacă R reprezintă joncțiunea proiecțiilor sale pe X_1, X_2, \dots, X_m , altfel spus

$$R = R(X_1) * R(X_2) * \dots * R(X_m)$$

unde $*$ reprezintă o (altă) notație pentru operatorul joncțiune internă, iar $R(X_i)$ este proiecția lui R pe X_i .

Când $m=3$, dependența este una *mutuală*,²⁴ în timp ce pentru $m=2$ este vorba de o dependență multi-valoare.

Fie $R(X_1, X_2, X_3)$ o relație (X_i - subansamblu de atribute). X_1 și X_2 sunt disjuncte, iar X_3 complementul față de R a $\{X_1 \cup X_2\}$. Se spune că există o *dependență mutuală* între X și Y dacă și numai dacă:

atunci când (x_1, x_2, x_3) și (x_1, x_2', x_3') sunt două tupluri ale relației $R(X_1, X_2, X_3)$:

- dacă $(x_2, x_3) \in R(X_2, X_3)$, atunci $(x_1, x_2, x_3) \in R(X_1, X_2, X_3)$
- dacă $(x_2', x_3) \in R(X_2, X_3)$, atunci $(x_1, x_2', x_3) \in R(X_1, X_2, X_3)$

Se poate scrie:

$$R(X_1, X_2, X_3) = R(X_1, X_2) * R(X_1, X_3) * R(X_2, X_3).$$

Primul exemplu

Pornim de la relația XYZ, alcătuită din tuplurile din figura 6.14.

XYZ		
X	Y	Z
x1	y1	z2
x1	y2	z1
x2	y1	z1
x1	y1	z1

Figura 6.14. Relația XYZ în care există dependențe de joncțiune

Se "sparge" relația de mai sus în trei relații ce reprezintă proiecțiile, pe câte două atribute, ale acesteia - figura 6.15.

²² [Rissanen77]

²³ [Aho s.a.79]

²⁴ Sintagma *dependență mutuală* a fost introdusă în [Nicolas 78]

XY		YZ		XZ	
X	Y	Y	Z	X	Z
x1	y1	y1	z2	x1	z2
x1	y2	y2	z1	x1	z1
x2	y1	y1	z1	x2	z1

Figura 6.15. Proiecțiile, pe câte două atribute, ale relației XYZ

Operațiunea $XY_YZ \leftarrow \text{JONCȚIUNE } (XY, YZ; XY.Y=YZ.Y)$ conduce la rezultatul din figura 6.16, iar prin joncționarea XY_YZ cu XZ se obține o relație cu aceleași linii ca și XYZ .

XY_YZ		
X	Y	Z
x1	y1	z2
x1	y1	z1
x1	y2	z1
x2	y1	z2
x2	y1	z1

Figura 6.16. Tabela XY_YZ

Prin urmare, se poate scrie:

$$XYZ = XY * YZ * XZ$$

După Date²⁵, afirmația *XYZ este egală cu joncțiunea celor trei proiecții ale sale, XY, XZ și YZ* este sinonimă cu a demonstra că, dacă perechea (x1,y1) apare în XY, și (y1, z1) apare în YZ și (x1, z1) apare în XZ, atunci (x1, y1, z1) apare în XYZ. Tot el formulează condiția pentru dependența de joncțiune și astfel: dacă în XYZ există trei tupluri, (x1, y1, z2), (x2, y1, z1) și (x1, y2, z1), atunci obligatoriu apare și un al patrulea, (x1, y1, z1).

Agenți, produse, companii

Pornind de la un exemplu preluat din lucrarea lui Nicolas²⁶, Ronald Fagin ilustrează într-unul din articolele sale²⁷ dependența de joncțiune cu o relație de genul R {Agent, Produs, Companie}. Un tuplu al acestei relații, spre exemplu (Vasilescu, Detergent Spuma Mării, Hurstman) semnifică faptul că agentul Vasilescu reprezintă detergentul Spuma Mării produs de compania Hurstman. În R există o dependență de joncțiune dacă și numai dacă,

presupunând că:

- agentul *a* reprezintă produsul *p* fabricat de compania *c'*
- agentul *a* reprezintă produsul *p'* fabricat de compania *c* și că

²⁵ [Date86], p.387

²⁶ [Nicolas78]

²⁷ [Fagin81]

- agentul a' reprezintă produsul p fabricat de compania c
atunci agentul a reprezintă produsul p fabricat de compania c !

Relația PCS

Pornim de la relația PCS prezentată în figura 6.17 în care este reflectată, parțial, situația cursurilor de informatică la specializări ale *Facultății de Economie și Administrarea Afacerilor* (situație dinainte de 1998, deoarece acum primii doi ani sunt comuni la profilul *Economic*).

PCS

Profesor	Curs	Specializare
Fătu	Bazele informaticii economice	Finanțe
Fătu	Bazele informaticii economice	Contabilitate
Filip	Bazele informaticii economice	Finanțe
Filip	Baze de date	Finanțe
Filip	Bazele informaticii economice	Management
Filip	Baze de date	Management
Airinei	Baze de date	Marketing
Airinei	Bazele informaticii economice	Marketing
Airinei	Bazele informaticii economice	Administrație Publică
Airinei	Baze de date	Management
Airinei	Bazele informaticii economice	Management

Figura 6.17. Relația PCS aflată în 4NF, nu însă și în 5NF

Comentarii:

- Un profesor predă mai multe cursuri: Profesor \rightarrow Curs
- Un profesor predă (unul sau mai multe cursuri) la mai multe specializări:
Profesor \rightarrow Specializare
- Un același curs poate fi ținut de mai mulți profesori: Curs \rightarrow Profesor
- Un același curs poate fi în planul de învățământ al mai multor specializări:
Curs \rightarrow Specializare
- La o specializare, în planul de învățământ sunt incluse mai multe cursuri:
Specializare \rightarrow Curs
- La o specializare țin cursuri mai mulți profesori:
Specializare \rightarrow Profesor

• La o specializare, un același curs poate fi predat de mai mulți profesori.
Explicație: la unele specializări, în cadrul aceleiași an de studii există doi semianii de predare; prin urmare, (Specializare, Curs) \rightarrow Profesor

- Un profesor poate ține un același curs la mai multe specializări:
(Profesor, Curs) \rightarrow Specializări
- La o specializare, un același profesor poate ține mai multe cursuri:
(Specializare, Profesor) \rightarrow Curs

Până în acest moment, am demonstrat inexistența oricărei dependențe funcționale, simple sau compuse. În continuare verificăm existența vreunei dependențe multivaloare. Există tuplurile:

(Airinei, Bazele informaticii economice, Administrație Publică)

și

(Airinei, Baze de date, Management).

Dacă ar opera dependența multi-valoare Profesor \rightarrow Curs | Specializare, ar trebui ca în relația PCS să existe și tuplurile

(Airinei, Bazele informaticii economice, Management)

și

(Airinei, Baze de date, Administrație Publică).

Or, ultimul tuplu nu există, deoarece cursul *Baze de date* nu este în planul de învățământ al specializării *Administrație Publică*. Rezultă că în PCS nu există nici o dependență multivaloare. Nu ne mai rămâne decât să dovedim, cu ultimele puteri, că PCS, adică $R(\text{Profesor, Curs, Specializare}) = R(\text{Profesor, Curs}) * R(\text{Profesor, Specializare}) * R(\text{Curs, Specializare})$.

Mai întâi, din PCS obținem tabelele PC, PS și CS, prin proiecțiile, pe câte două atribute ale acestei relații - vezi figura 6.18.

PC

Profesor	Curs
Fătu	Bazele informaticii economice
Filip	Bazele informaticii economice
Filip	Baze de date
Airinei	Baze de date
Airinei	Bazele informaticii economice

PS

Profesor	Specializare
Fătu	Finanțe
Fătu	Contabilitate
Filip	Finanțe
Filip	Management
Airinei	Marketing
Airinei	Administrație Publică
Airinei	Management

CS

Curs	Specializare
Bazele informaticii economice	Finanțe
Bazele informaticii economice	Contabilitate
Baze de date	Finanțe
Bazele informaticii economice	Management
Baze de date	Management
Baze de date	Marketing
Bazele informaticii economice	Marketing
Bazele informaticii economice	Administrație Publică

Figura 6.18. Proiecțiile pe câte două atribute ale relației PCS

Comentarii:

- Prof. Fătu ține cursul de *Bazele informaticii economice*, prof. Fătu face ore la specializarea *Finanțe* și disciplina *Bazele informaticii economice* apare în planul de învățământ la specializarea *Finanțe*, și
prof. Fătu ține cursul de Bazele Informaticii Economice la specializarea Finanțe.

• Prof. Fătu ține cursul de *Bazele informaticii economice*, prof. Fătu are ore la specializarea *Contabilitate* și *Bazele informaticii economice* este în planul de învățământ al specializării *Contabilitate*, și

prof. Fătu ține cursul de Bazele Informaticii Economice la specializarea Contabilitate.

• Prof. Filip ține cursul de *Bazele informaticii economice*, prof. Filip face ore la specializarea *Finanțe* și *Bazele informaticii economice* este în planul de învățământ la specializarea *Finanțe*, și

prof. Filip ține cursul de Bazele Informaticii Economice la specializarea Finanțe.

Generalizând:

Dacă profesorul P ține cursul C,

profesorul P ține ore la specializarea S și

cursul C este în planul de învățământ al specializării S,

atunci

profesorul P ține cursul C la specializarea S.

Numai în aceste condiții există dependența de joncțiune (mutuală, în acest caz)

* {(Profesor, Curs), (Profesor, Specializare), (Curs, Specializare)}.

Vânzări și credite

De câțiva ani a început și la noi vânzarea bunurilor de folosință îndelungată (mașini de spălat, frigidere, televizoare etc.) cu plata în rate. Magazinele încheie în prealabil acorduri cu una sau mai multe bănci pentru acordarea creditelor pentru bunurile vândute. La vânzare, clienților li se prezintă băncile la care pot apela pentru credite. Clienții decid asupra băncii cele mai convenabile și, astfel, vânzarea-i gata (cu rambursarea e altă discuție, mult mai interesantă). Relația cumpărător-magazin-bancă este reflectată în tabela CREDITE_RATE din figura 6.19.

CREDITE_RATE

Cumpărător	Magazin	Bancă
Alexa V. Gheorghe	Onitrade Ltd.	BRD
Alexa V. Gheorghe	Domo	BCR
Georgescu P. Mircea	Onitrade Ltd.	Banca Ion Țiriac
Brava H. Ion	Moldova	Banca Ion Țiriac
Brava H. Ion	Onitrade Ltd.	Banca Ion Țiriac
Brava H. Ion	Onitrade Ltd.	BRD
Finaru W. Luminița	Onitrade Ltd.	Banca Ion Țiriac
Finaru W. Luminița	Domo	BCR
Finaru W. Luminița	Moldova	Banca Albina

Figura 6.19. Relația protagoniștilor pentru vânzările în rate

Să presupunem că sunt valabile următoarele restricții:

- dacă un magazin (firmă) M a stabilit un acord cu o bancă B privind vânzări în rate,
 - dacă un client C cumpără bunuri de la magazinul M ,
 - dacă clientul C a ales să ia credit de la banca B ,
- atunci pentru bunurile achiziționate de la magazinul M clientul C va apela la serviciile băncii B .

În aceste condiții, există o dependență de joncțiune notată:

* { CumpărătorMagazin, MagazinBancă, CumpărătorBancă }

sau

* { { Cumpărător, Magazin }, { Magazin, Bancă }, { Cumpărător, Bancă } }

Ca și în cazul dependențelor multi-valoare și 4NF, lucrarea românească (din câte cunosc eu) în care se explică cel mai bine dependențele de joncțiune și 5NF este cea a lui Robert Dollinger. Exemplul său este relația PDL {Profesor, Disciplină, Limbă}, iar condiția în care se manifestă dependențele de joncțiune (autorul le numește *dependențe de cuplare*²⁸) este: *Dacă o anumită disciplină este predată într-o anumită limbă, atunci acest lucru este făcut de fiecare profesor care predă disciplina respectivă și cunoaște limba în cauză*²⁹.

Și exemplul Rebecăi Riordan este sugestiv. Dacă, într-o relație R {Furnizor, Produs, Client} se verifică restricția: *Atunci când un furnizor F vinde un produs P , un client C cumpără acel produs P , iar furnizorul F are o relație (comercială) cu clientul C (în sensul că F a vândut măcar un produs clientului C), obligatoriu furnizorul F va vinde produsul P clientului C .* Ulterior, aveam să descopăr că exemplul este preluat din [Fleming&vonHalle 89].

6.4.A cincea formă normală

Cea de-a cincea formă normală - denumită și *forma normală proiecție-joncțiune* - (Project-Join Normal Form) are la bază dependența de joncțiune și a fost definită tot de Ronald Fagin³⁰. O relație R este în 5NF dacă și numai dacă toate dependențele de joncțiune sunt consecințe ale cheilor candidate ale lui R ³¹.

De remarcat că 5NF, 4NF și BCNF corespund definiției unice: *Orice determinant al unei dependențe este o cheie*; ceea ce diferă ține de faptul că, pentru BCNF, este vorba de dependență *funcțională*, pentru 4NF -dependență *multi-valoare*, iar pentru

²⁸ Probabil că dacă era moldovean (ca mine) le-ar fi zis *dependențe de alăturare*

²⁹ [Dollinger98], p.174

³⁰ [Fagin 79]

³¹ [Date86], p.389

5NF - *dependență de joncțiune*. 5NF este o generalizare a lui 4NF care e o generalizare a BCNF.

Până la 5NF, o relație era descompusă prin proiecție în două relații ce puteau, prin joncțiune naturală, să "refacă", în orice moment, relația inițială. Cu dependența de joncțiune, această descompunere fără pierderi poate fi generalizată nu la două, ci la mai multe relații. În fapt, dependența de joncțiune corespunde existenței mai multor legături care sunt independente în cadrul relației și care vor fi izolate prin descompunerea în 5NF.

Cele trei exemple

- Pe baza dependenței de joncțiune demonstrate, tabela XYZ din figura 6.14 este transpusă în 5NF prin descompunerea sa în relațiile XY, XZ și YZ, prezentate în figura 6.15.

- Relația PCS din figura 6.17, care se află în 4NF, este adusă în 5NF sub forma tabelelor PC, PS și CS din figura 6.18.

- Datorită DJ notată * {CumpărătorMagazin, MagazinBancă, CumpărătorBancă} (figura 6.19), în 5NF relația CREDITE_RATE are compoziția din figura 6.20.

CumpărătorMagazin		MagazinBancă		CumpărătorBancă	
Cumpărător	Magazin	Magazin	Bancă	Cumpărător	Bancă
Alexa V. Gheorghe	Onitrade Ltd.	Onitrade Ltd.	BRD	Alexa V. Gheorghe	BRD
Alexa V. Gheorghe	Domo	Domo	BCR	Alexa V. Gheorghe	BCR
Georgescu P. Mircea	Onitrade Ltd.	Onitrade Ltd.	Banca Ion Țiriac	Georgescu P. Mircea	Banca Ion Țiriac
Brava H. Ion	Moldova	Moldova	Banca Ion Țiriac	Brava H. Ion	Banca Ion Țiriac
Brava H. Ion	Onitrade Ltd.	Moldova	Banca Albina	Brava H. Ion	BRD
Finaru W. Luminița	Onitrade Ltd.			Finaru W. Luminița	Banca Ion Țiriac
Finaru W. Luminița	Domo			Finaru W. Luminița	BCR
Finaru W. Luminița	Moldova			Finaru W. Luminița	Banca Albina

Figura 6.20. Relația CREDITE_RATE în 5NF

6.5. Alte forme normale

Cele cinci forme normale au fost completate și cu alte tentative de a ameliorare a schemei bazei prin identificarea unor restricții și tipuri de dependență. Ronald Fagin, inițial, a demonstrat că orice relație aflată în 5NF nu mai poate fi descompusă fără pierderi de informații³², însă tot el este autorul a, probabil, cea mai cunoscută formă normală post-5FN - *forma normală domeniu-cheie* (DK/NF -

³² [Fagin 79], citat în [Miranda & Busta 90] vol.II, p. 152.

Domain-Key Normal Form)³³. DK/FN nu utilizează dependențe funcționale, multivaloare sau de joncțiune. O relație este în DK/NF dacă *orice restricție poate fi determinată doar cunoscând ansamblul numelor atributelor și domeniilor pe care sunt definite acestea, precum și cheile relației*³⁴, altfel spus, toate restricțiile și dependențele care se manifestă într-o relație sunt o consecință a restricțiilor de domeniu și de cheie ale relației³⁵. După Fagin, o schemă este în DK/NF dacă și numai dacă nu prezintă anomalii de inserare și ștergere³⁶. Trebuie precizat însă că Fagin privește diferit noțiunea de anomalie decât teoria clasică a normalizării (accepțiunea lui Codd): pentru el, spre exemplu, o anomalie de inserare apare atunci când adăugarea aparent corectă a unei linii într-o tabelă violează una din restricțiile relației.

Deși pomenită în destule lucrări, DK/NF a rămas la stadiu de deziderat, neexistând lucrări care să se constituie îndreptar în aducerea efectivă a schemei unei baze de date în DK/NF. Probabil, problema principală ține de dificultatea definirii restricțiilor laborioase ce se manifestă în aplicațiile complexe actuale.

În afara acestei forme normale, Chris Date mai pomenește despre alte două³⁷:

- Forma Normală „Selecție-Reuniune” („Restriction-union” normal form) care se referă la fragmentarea unei relații (tuplurilor) în relații diferite prin *selecție*, și nu *proiecție*, cum face normalizarea clasică; de exemplu, în relația CLIENTI_SUA s-ar regăsi clienții americani, în CLIENTI_UK cei din Anglia etc., ca să ne oprim doar la celebra axă băsească. Relația inițială s-ar reface prin reuniune, nu prin joncțiune.
- A șasea formă normală – 6NF, pe care Date o definește ca acea situație în care într-o schemă nu există nici o dependență de joncțiune ne-trivială, o dependență de joncțiune fiind trivială dacă și numai dacă cel puțin una din proiecții este definită pe toate atributele relației³⁸. După cum observați, lucrurile nu stau deloc simplu, așa că implicațiile practice asupra proiectării bazelor de date nu se întrezăresc.

³³ [Fagin81]

³⁴ [Fagin81]

³⁵ [Elmasri & Navathe 00], p.523

³⁶ [Fagin81]

³⁷ [Date04], pp.399-400

³⁸ [Date04], pp.762-763