

## Capitolul 3. Prima formă normalizată

Prima formă normalizată (1NF) este, în general, tratată cu superficialitate de majoritatea practicienilor în materie de proiectare a bazelor de date. Aceasta deoarece principala cerință – atomicitatea valorilor – este un deziderat ușor de îndeplinit, cel puțin la prima vedere. Cele mai sensibile chestiuni în aducerea relațiilor în 1NF sunt legate de atomicitate, eliminarea grupurilor repetitive, precum și de stabilirea cheii primare, cheie primară care devine o problemă delicată atunci când se pornește de la o relație cu zeci sau sute de atribute.

Pe de altă parte, noțiunea de atomicitate este una destul de alunecoasă. Deși în primele lucrări Codd nu formulează o definiție riguroasă a atomicității, timp de aproape trei decenii majoritatea autorilor din "zona" relaționalului au considerat ca pe o primă "poruncă" a unei relații (tabele) necesitatea ca valoarea oricărui atribut pe orice linie (tuplu) să fie una scalară (atomică, nedecompozabilă). De câțiva timp, însă, Chris Date și alți relaționiști încearcă să depășească limitele atomicității, înlocuind domeniile cu *tipuri* care pot fi atât scalare, cât și compozite (definite de utilizator).

### 3.1. Definiții ale primei forme normalizate

Teoria normalizării nu face parte, propriu-zis, din modelul relațional. Dintre toate formele normalizate, doar prima are caracter de obligativitate. O bază de date este normalizată dacă toate relațiile (tabelele) care o alcătuiesc se află *măcar* în 1NF. Celelalte forme normale sunt dezirabile pentru diminuarea redundanțelor, spațiului ocupat pe disc sau anomaliilor manifestate la actualizare, însă nu sunt obligatorii.

După Codd, o relație este în prima formă normală dacă... nici unul dintre domeniile sale nu conține elemente care sunt, la rândul lor, seturi (ansambluri)<sup>1</sup>. Se cuvine de adăugat remarcă lui Date care crede că ar fi fost mai nimerit ca, în locul termenului *domenii*, Codd să-l fi folosit pe cel de *atribute*. Actualmente, mulți "relaționiști" preferă termenul *tip* celui de *domeniu*<sup>2</sup>.

Unele lucrări<sup>3</sup> definesc o relație aflată în 1NF ca acea relație în care fiecare atribut prezintă numai valori atomice, adică toate atributele sunt ne-decompozabile; după R.Riordan, o relație este în 1NF dacă domeniile pe care sunt definite atributele relației sunt scalare<sup>4</sup>. În ediția a treia a cărții dedicate sistemelor de baze de date, Connolly și Begg sunt mai tranșanți: o relație în 1NF este o relație în care intersecția oricărei linii cu

---

<sup>1</sup> [Codd72]

<sup>2</sup> [Date04]

<sup>3</sup> [Date86], [Miranda&Busta90], [Connolly s.a.96], [Dollinger98]

<sup>4</sup> [Riordan99]

oricare coloană conține o valoare și numai una<sup>5</sup>. Astfel, 1NF respinge ideea de set de valori, tuplu de valori sau combinația acestora ca valoare a unui singur atribut pe un oricare tuplu (linie) a relației.

Alți autori formează explicit pentru 1NF și restricția: *toate atributele ce compun relația să fie în dependență funcțională față de cheia relației*, ceea este, oarecum, o tautologie, întrucât una dintre "poruncile" valabile pentru orice tabelă stipulează negru pe alb că într-o relație nu pot exista două tupluri identice sau, altfel spus, într-o relație trebuie să existe măcar un atribut sau combinație de atribute ale căror valori să deosebească orice tuplu de toate celelalte sau orice relație posedă o cheie primară.

Formularea care a generat cele mai multe confuzii este legată de noțiunea de *grupuri repetitive*<sup>6</sup>: O relație în 1NF nu trebuie să conțină grupuri repetitive.

De obicei, noțiunea de grup repetitiv este raportată direct la cea de atomicitate. Diferențierea atomicitate – grup repetitiv apare în [Lungu s.a.95], pentru care un atribut atomic este unul simplu (ne-compus): „O relație este în 1NF dacă domeniile pe care sunt definite atributele relației sunt constituite numai din valori atomice (elementare). În plus, un tuplu nu trebuie să conțină atribute sau grupuri de atribute repetitive”<sup>7</sup>. Din păcate, autorii se opresc aici cu definiția, păstrând o remarcabilă discreție vis-a-vis de ceea ce consideră grup repetitiv. La fel stau lucrurile și la dna. Ileana Popescu<sup>8</sup>.

Un plus de claritate aduce definiția lui Toby Teorey, pentru care o relație este în 1NF dacă și numai dacă toate coloanele conțin numai valori atomice; aceasta înseamnă că *nu există grupuri (coloane) repetitive în cadrul vreunei linii a tabelului*. Imediat autoarea precizează că *un grup repetitiv apare într-o tabelă atunci când unui atribut multi-valoare îi sunt permise două sau mai multe valori reprezentate în cadrul unei aceleiași linii*<sup>9</sup>. Sau, după cum spunea Chris Date până în ediția a șasea a celei mai cunoscute cărți a sa, un grup repetitiv reprezintă o coloană care conține câteva valori în fiecare linie, sau, altfel spus, un număr diferit de valori pe linii diferite<sup>10</sup>.

În concluzie, aducerea unei relații în 1NF presupune discutarea următoarelor elemente:

- raportul atribut simplu/atribut compus;
- grupuri repetitive de atribute (pe orizontală);
- grupuri repetitive de valori în cadrul fiecărei celule a tabelului.

---

<sup>5</sup> [Connoly & Begg 02], p.388

<sup>6</sup> [Date86], [Pratt&Adamski91], [Oprea99]), [Lungu s.a.95]

<sup>7</sup> [Lungu s.a. 95], p.169

<sup>8</sup> [Popescu 01], p.78

<sup>9</sup> [Teorey99], p.99

<sup>10</sup> Preluare din ediția a opta a cărții ([Date 04], p.153), unde Date se autocitează pentru a demonstra că înțelesese greșit noțiunea de domeniu (tip) și, în consecință greșise atunci când a formulat definiția 1NF.

## 3.2. Atribute simple și atribute compuse

Primele vizate de "stigmatul" neatomicității sunt atributele compuse. Cu ani buni în urmă, în cursurile de profil, exemplul clasic al unui atribut compus era *DataCalendaristică*, alcătuit, cum altfel, din câmpurile: *Ziuă*, *Lună*, *An*. Era vremea înfloritoare a COBOLului și FORTRANului, când nu existau mecanisme de declarare și gestionare a variabilelor și atributelor de tip *DATE*; verificarea corectitudinii datei calendaristice cădea în sarcina programatorului ce trebuia să scrie o rutină specială.

Astăzi, nu există SGBD sau mediu de programare care să nu prezinte tipul de dată *DATE* prin care se gestionează simultan toate cele trei elemente; prin funcții de conversie se pot obține: ziua (*DAY*), luna (*MONTH*), anul (*YEAR*), ba chiar calcula numărul de zile, luni, ani dintre două date calendaristice și alte minunății. Ca să nu mai vorbim de tipul *DATETIME* și acesta omniprezent în produsele software ale zilelor noastre.

Un alt exemplu celebru privind atribute compuse (non-atomice) este *Adresa*. Fiind alcătuită din *Stradă*, *Număr*, *Bloc*, *Scară*, *Etaj*, *Apartament*, plus *CodPoștal* (după noul sistem de codificare) discuția despre atomicitatea adresei pare de prisos, iar descompunerea sa imperativă. Din nou însă trebuie să ne raportăm la obiectivele bazei. Fără îndoială că, pentru BD a unei filiale CONEL sau ROMTELECOM, sau pentru poliție, preluarea separată a fiecărui element constituent al adresei este vitală. Interesează, spre exemplu, ce abonați sunt pe strada *Tranziției* sau câte cereri sunt depuse de cetățeni de pe străzile *Lascăr Catargi*, *Titu Maiorescu* și *Marina Scupra*. Sau, un cioranian ce lucrează în cadrul Poliției, ar putea fi interesat să afle dacă gradul de sinucidere al persoanelor ce locuiesc la etajul trei este mai mare decât cel al persoanelor de la parter.

Cu totul altfel stau lucrurile pentru un importator direct, pentru un mare engrossist sau pentru o firmă de producție. Partenerii de afaceri ai acestora sunt persoane juridice, iar adresa interesează numai la nivel general. Este drept că imediat după Revoluție deveniseră mari furnizori naționali de cupru și aluminiu și persoane fizice (vezi cazul bulibașei din Ciurea, dar asta-i altă poveste...).

Alte exemple de atribute ce pot fi considerate, în funcție de circumstanțe, simple sau compuse: *DataOperațiuniiBancare* (*Data+Ora*), *BuletinIdentitate* (*Seria+Număr*), *NrÎnmatriculareAuto* (privit global, sau pe cele trei componente: județ, număr și combinația trei de litere).

În relația universală *FILMOGRAFIE*, al cărei dicționar de date simplificat este cel din tabelul 2.2, cel puțin două atribute sunt compuse și trebuie "sparte", *Distribuție* și *Premii*. Astfel, în *Distribuție* pot fi delimitate măcar două informații, *Actor* și *Rol*. În cazul premiilor, ar trebui detaliată discuția prin introducerea atributelor:

- denumire premiu (*Oscar*, *Globul de aur* etc.);
- locul atribuirii (*Hollywood*, *Cannes*, *Berlin*, *Veneția*, *București* etc.);
- categoria (regie, scenariu, actor în rol principal etc.);
- anul decernării;

- numele actorului, dacă categoria este pentru interpretare.
- Astfel, noua formă a dicționarului de date este cea din tabelul 3.1.

Tabel 3.1. FILMOGRAFIE - dicționarul datelor (simplificat) - versiunea 2

Atribut	Descriere
IdFilm	Codul unic al filmului
TitluOriginal	Titlul în engleză, franceză etc, așa cum apare la lansare filmului
TitluRO	Traducerea românească a titlului original
AnLans	Anul lansării
Producători	Producătorul sau producătorii filmului
Regizori	Regizorul sau regizorii filmului
Rol	Rolul din film
Actor	Actorul care interpretează rolul/filmul curente
Genuri	Genul/genurile la care se încadrează filmul (horror, comedie etc.)
DenPremiu	Numele premiului - tipul (Oscar, Leul de argint, Ursul de aur etc.)
LocDecernare	Locul în care se organizează festivitatea sau festivalul
Categorie	Categoria premiului (pentru ce anume se acordă premiul)
AnPremiu	Anul decernării

Concluzionând în termeni rabinici, ambele variante, descompunerea sau nedescompunerea atributelor compuse, sunt valabile de la caz la caz, alegerea soluției fiind în concordanță cu specificul bazei de date (și inspirația proiectantului). În plus, nu trebuie uitat că și pentru atributele compuse pot fi extrase destule informații utilizând în SQL operatorul `LIKE`.

Atomicitatea atributelor într-o bază de date relațională a fost privită ca o limită serioasă a modelului, datorită imposibilității adaptării BDR la specificul unor domenii precum multimedia, CAD etc. Mulți autori, dintre care merită amintiți cu deosebire Chris J. Date și Hugh Darwen, se opun ideii de atomicitate formulată de Codd (vezi și paragraful 3.5).

### 3.3.Despre grupuri repetitive și urmările lor

O relație (tabelă) în 1NF nu trebuie să conțină atribute care se repetă ca grupuri. Într-o altă formulare, toate liniile unei tabelă trebuie să conțină același număr de atribute. Fiecare celulă a tabelă (intersecția unei coloane cu o linie), altfel spus, valoarea unui atribut pe o linie (înregistrare), trebuie să fie atomică.

După [Connolly s.a.96], un grup repetitiv este un atribut sau grup de atribute dintr-o tabelă care apare cu valori multiple pentru o singură apariție a cheii primare a tabelă ne-normalizate.

Să luăm exemplul tabelă BIBLIOTECĂ din figura 3.1. Tabela gestionează informații despre cărțile existente în depozitul bibliotecă *Facultății de Economie și Administrarea Afacerilor* (FEAA). De remarcat că în bibliotecă există două exemplare ale cărții cu ISBN-ul 973-683-709-2 și trei exemplare de cea dedicată Visual FoxPro. Prima carte a fost scrisă de patru autori și îi sunt asociate opt

cuvinte-cheie, iar a doua are un singur autor. Relația nenormalizată conține trei grupuri repetitive: Cote, Autori și CuvinteCheie.

**BIBLIOTECĂ**

ISBN	Titlu	Cote	Autori
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421, III-13422, III-13423	Marin Fotache, Ioan Brava, Cătălin Strâmbei, Liviu Crețu
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10678, III-10679	Marin Fotache

Editura	LocSediuEd	AnApariție	CuvinteCheie
Polirom	Iași	2002	baze de date, SQL, proceduri stocate, FoxPro, formulare, orientare pe obiecte, client-server, web
Polirom	Iași	2001	baze de date, algebră relațională, SQL

Figura 3.1. Relația universală nenormalizată BIBLIOTECĂ

Paragrafele următoarele sunt dedicate prezentării unor soluții pentru eliminarea grupurilor repetitive. Trebuie precizat, însă, că nu întotdeauna grupurile repetitive reprezintă o problemă majoră. Spre exemplu, într-o relație gen PERSOANE\_CONTACT atributul Telefoane poate avea valori ne-atomice, deoarece pot exista mai multe numere atât la birou, cât și mobile (mulți manageri au câte un aparat pe fiecare rețea - Orange, Conex, Zapp)...

**3.3.1. Grupuri repetitive pe orizontală**

Una dintre cele mai nerecomandate soluții pentru eliminarea grupurilor repetitive ține de extinderea pe orizontală a tabelelor, prin duplicarea forțată a unor atribute. Revenim la figura anterioară. Pentru a înlătura grupurile repetitive de pe primele două linii, putem fi tentați să modificăm structura tabeli, ca în figura 3.2.

**BIBLIOTECĂ\_GRPURI\_REPETITIVE\_PE\_ORIZONTALĂ**

ISBN	Titlu	Cota1	Cota2	Cota3
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	III-13422	III-13423
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10678	III-10679	NULL

Autor1	Autor2	Autor3	Autor4	Editura	LocSediuEd
Marin Fotache	Ioan Brava	Cătălin Strâmbei	Liviu Crețu	Polirom	Iași
Marin Fotache	NULL	NULL	NULL	Polirom	Iași

AnApariție	CuvântCheie1	CuvântCheie2	CuvântCheie3	CuvântCheie4
2002	baze de date	SQL	proceduri stocate	FoxPro
2001	baze de date	algebră relațională	SQL	NULL

CuvântCheie5	CuvântCheie6	CuvântCheie7	CuvântCheie8
--------------	--------------	--------------	--------------

formulare	orientare pe obiecte	client-server	web
NULL	NULL	NULL	NULL

Figura 3.2. O soluție (destul de penibilă) de eliminare a grupurilor repetitive

Deși tabela se află în prima formă normală, structura acesteia este hilară. Ar trebui să "rezervăm" câteva zeci de atribute pentru cotele cărților, altă serie de atribute pentru autori (apropo, care este numărul maxim de autori ce au participat la scrierea unei cărți ?), ca să nu mai vorbim de cuvintele cheie.

Atenție, însă ! Deși hilară, schema de mai sus nu violează conceptul de atomicitate, oricare ar fi formularea acestuia. Unii autori respectabili, precum Toby Teorey<sup>11</sup>, greșesc atunci când dau exemple de relații nenormalizate de tipul celor din figura 3.2 (este drept, la Teorey atributele care se repetă în cadrul relației au același nume, ceea ce modelul relațional respinge din start).

Această variantă de normalizare funcționează totuși în cazuri punctuale. Spre exemplu, nu este exagerat ca într-o tabelă să avem atribute de genul celei din figura 3.3.

#### PERSOANE\_1

CNP	Nume	Prenume	DataNașterii	CNPMamă	CNPTată
2641121390802	Bucur	Cerasela	21-11-1964	2440611390167	1401205390102
...					

Figura 3.3. Un exemplu rezonabil de atribut pseudo-repetitiv

Cum orice persoană are cel mult doi părinți (re)cunoscuți, putem introduce grupul de atribute CNPMamă, CNPTată, deși, din punct de vedere relațional, este corectă și varianta din figura 3.4.

#### PERSOANE\_2

CNP	Nume	Prenume	DataNașterii	CNPPărinte	Părinte
2641121390802	Bucur	Cerasela	21-11-1964	2440611390167	Mamă
2641121390802	Bucur	Cerasela	21-11-1964	1401205390102	Tată
...					

Figura 3.4. O altă primă formă normală a relației PERSOANE\_1

### 3.3.2. Eliminarea grupurilor repetitive prin adăugarea de tupluri

A doua variantă păstrează structura relației (universale sau nu), în sensul că nu se modifică numărul de atribute, adăugându-se tupluri (linii) suplimentare astfel încât orice valoare a celor trei atribute de tipul "grupuri repetitive" devine atomică. Apare însă problema: cum anume completăm tuplurile pentru ca să fie preluate toate informațiile, iar, pe de altă parte, eventualele actualizări să nu antreneze

<sup>11</sup> [Teorey99], pp.99-100

pierderi sau alterări de informații ? În acest sens, am putea lua în discuție trei soluții, după cum urmează.

### **Număr minim de linii - o singură valoare atomică pentru fiecare atribut compozit**

Avem trei atribute non atomice (multivaloare). În figura 3.1, pentru fiecare carte, numărul de linii din tabelă va fi maximul dintre numărul cotelor, numărul autorilor și numărul cuvintelor cheie. Astfel, pentru *Visual FoxPro* sunt trei cote, patru autori și opt cuvinte cheie; așadar, vom avea opt linii (primele opt din figura 3.5). Valorile cotelor în liniile 4-8 vor fi NULL; la fel valorile atribului *Autor* pentru liniile 5-8.

### **BIBLIOTECĂ TUPLURI NOI SOLUȚIA\_1**

<b>ISBN</b>	<b>Titlu</b>	<b>Cotă</b>	<b>Autor</b>
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Marin Fotache
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13422	Ioan Brava
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13423	Cătălin Strîmbei
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	Liviu Crețu
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10678	Marin Fotache
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10679	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	NULL	NULL

<b>Editura</b>	<b>LocSediuEd</b>	<b>AnApariție</b>	<b>CuvântCheie</b>
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	SQL
Polirom	Iași	2002	proceduri stocate
Polirom	Iași	2002	FoxPro
Polirom	Iași	2002	formulare
Polirom	Iași	2002	orientare pe obiecte
Polirom	Iași	2002	client-server
Polirom	Iași	2002	web
Polirom	Iași	2001	baze de date

Polirom	Iași	2001	algebră relațională
Polirom	Iași	2001	SQL

Figura 3.5. Completarea cu tupluri - soluția 1

Deoarece o cotă nu poate identifica decât o singură carte, cheia primară a noii relații este alcătuită din trei atribute, (Cotă, Autor, CuvântCheie). Redundanța datelor este evidentă. Dacă pentru cartea de Visual FoxPro am dori să mai introducem un nou cuvânt cheie, ar trebui inserată o linie nouă în tabelă, în care valorile atributelor Autor și Cotă să fie nule. Aceasta nu este însă singura problemă. Să presupunem că sintagma *proceduri stocate* este greșit asociată cărții de Visual FoxPro. În mod normal, corectarea acestei greșeli presupune ștergerea celui de-al treilea tuplu:

```
DELETE FROM BIBLIOTECĂ_TUPLURI_NOI_SOLUTIA_1
WHERE ISBN = '973-683-889-7' AND
      cuvântcheie = 'proceduri stocate')
```

Necazul e că ștergerea liniei se face cu două pierderi de informații. Prima pierdere ține de calitatea de coautor al acestei cărți a lui Cătălin Strîmbei, iar a doua este legată de exemplarul cărții ce are cota *III-13423*. Aceasta deoarece tuplul șters era singurul care se referea la autorul și la cota cu pricina.

Am putea încerca un artificiu, și anume ca ștergerea unui cuvânt cheie sau autor sau cotă să nu se facă prin ștergerea liniei respective, ci setarea pe NULL a valorii de pe linia incriminată:

```
UPDATE BIBLIOTECĂ_TUPLURI_NOI_SOLUTIA_1
SET cuvântcheie = NULL
WHERE ISBN = '973-683-889-7'
      AND cuvântcheie = 'proceduri stocate')
```

Nici această idee nu e din cale-afară de inteligentă. Dacă am dori eliminarea sintagmei *web* pentru aceeași carte:

```
UPDATE BIBLIOTECĂ_TUPLURI_NOI_SOLUTIA_1
SET cuvântcheie = NULL
WHERE ISBN = '973-683-889-7' AND cuvântcheie = 'web')
```

atunci linia modificată devine cu totul de prisos în tabelă, toate cele trei atribute, Cota, Autor și CuvântCheie având valori NULL. Cu alte cuvinte, după modificare, tuplul nu spune nimic, nu are valoare informațională. Practic, după fiecare NULL-izare pe o linie a unuia dintre cele trei atribute trebuie verificat dacă pe linia respectivă celelalte două atribute au valori NULL, caz în care se poate șterge.

Mai există o problemă și din punct de vedere al conținutului informațional. Să presupunem că ne interesează să aflăm care sunt autorii ce au scris despre SQL. Ținând seama de structura relației de mai sus, am fi tentați să folosim o selecție după cuvântul cheie SQL urmată de o proiecție a valorilor atributului Autor:

```
SELECT autor
FROM BIBLIOTECĂ_TUPLURI_NOI_SOLUTIA_1
WHERE cuvântcheie = 'SQL')
```



Rezultatul furnizat este alcătuit din două linii, una ce conține valoarea *Ioan Brava*, iar celalaltă valoarea NULL. Lipsesc, deci, ceilalți trei autori ai cărții despre Visual FoxPro, din care unul (săracul) a scris chiar o carte despre SQL. Interogarea ce furnizează răspunsul corect trebuie să folosească o joncțiune sau corelare după atributul ISBN. Prin urmare, este necesar de știut că structura informațională pe care o reprezintă tabela de mai sus are o serie de neajunsuri.

Cel mai net motiv de renunțare la această variantă este prezența valorilor nule în componența atributelor cheie primară, după cum vom discuta în paragraful 3.4.

### **Număr de linii egal cu suma valorilor elementare ale atributelor de tip grup repetitiv**

A doua soluție de completare are în vedere eliminarea problemei de la ștergerea unei linii în cazul soluției precedente, și anume: pe oricare linie, o valoare nenulă a unuia dintre cele trei atribute atrage după sine nulitatea celorlalte două - vezi figura 3.6. Spre exemplu, prima carte are 3 cote, 4 autori și 8 cuvinte cheie; prin urmare vor fi necesare  $3 + 4 + 8 = 15$  linii.

### **BIBLIOTECĂ\_TUPLURI\_NOI\_SOLUȚIA\_2**

ISBN	Titlu	Cotă	Autor
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13422	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13423	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	Marin Fotache
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	Ioan Brava
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	Cătălin Stâmbel
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	Liviu Crețu
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	NULL	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10678	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	III-10679	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	NULL	Marin Fotache

973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	NULL	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	NULL	NULL
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	NULL	NULL

<b>Editura</b>	<b>LocSediuEd</b>	<b>AnApariție</b>	<b>CuvântCheie</b>
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	NULL
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	SQL
Polirom	Iași	2002	proceduri stocate
Polirom	Iași	2002	FoxPro
Polirom	Iași	2002	formulare
Polirom	Iași	2002	orientare pe obiecte
Polirom	Iași	2002	client-server
Polirom	Iași	2002	web
Polirom	Iași	2001	NULL
Polirom	Iași	2001	NULL
Polirom	Iași	2001	NULL
Polirom	Iași	2001	baze de date
Polirom	Iași	2001	algebră relațională
Polirom	Iași	2001	SQL

Figura 3.6. Completarea cu tupluri - soluția 2

Această secundă soluție, deși presupune un număr de linii mai mare, elimină necazul pierderii involuntare de informații. Astfel, ștergerea unui cuvânt cheie nu va afecta, cu siguranță, nici un autor sau exemplar al cărții respective.

Ca și în cazul soluției anterioare, indezirabilitatea variantei este legată de prezența valorilor nule în componența atributelor cheie primară.

### **Număr maxim de linii, egal cu produsul valorilor elementare ale atributelor de tip grup repetitiv**

De data această, tăiem valorile nule de la rădăcină, introducând în relație toate combinațiile Autor-Cotă-CuvântCheie. Cum prima carte are 3 cote, 4 autori și 8

cuvinte cheie, în relația BIBLIOTECĂ\_TUPLURI\_NOI\_SOLUȚIA\_3 vor exista  $3 * 4 * 8 = 96$  de tupluri referitoare la aceasta - vezi figura 3.7.

### BIBLIOTECĂ\_TUPLURI\_NOI\_SOLUȚIA\_3 (fragment)

ISBN	Titlu	Cotă	Autor
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Marin Fotache
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Ioan Brava
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Cătălin Stâmbei
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Liviu Crețu
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Marin Fotache
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Ioan Brava
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Cătălin Stâmbei
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	III-13421	Liviu Crețu
	.... (încă 88 de linii pentru cartea de Visual FoxPro, plus 6 pentru cartea de SQL )		

Editura	LocSediuEd	AnApariție	CuvântCheie
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	baze de date
Polirom	Iași	2002	SQL
Polirom	Iași	2002	SQL
Polirom	Iași	2002	SQL
Polirom	Iași	2002	SQL
	...		

Figura 3.7. Completarea cu tupluri - soluția 3

Cele opt linii din figură se referă doar la primele două cuvinte cheie (*baze și date* și *SQL*) pentru primul exemplar (cota III-13421) din prima carte. Aducerea relației în 1NF eliminând grupurile repetitive prin adăugarea de noi tupluri de această manieră se dovedește mai mult decât costisitoare. Aceasta nu numai din punctul de vedere al spațiului excesiv consumat datorită redundanței masive, dar și dacă ne gândim la anomalii de actualizare. Spre exemplu, pentru a nu pierde informații, atunci când adăugăm un cuvânt-cheie nou pentru cartea luată spre

analiză, e nevoie să inserăm în tabelă nu mai puțin de 3 (cote) \* 4 (autori) = 12 linii. Dezvoltând această idee optimistă, imaginați-vă cum stau lucrurile pentru o lucrare monumentală cu 10 autori și 78 de cuvinte-cheie din care în bibliotecă există 40 de exemplare ?

### 3.3.3. Eliminarea grupurilor repetitive prin construirea de noi relații

După năduful provocat de redundanța masivă a ultimei soluții de normalizare, cea care urmează este ca o ușurare: "ruperea" din relația universală a grupurilor și constituirea de relații noi<sup>12</sup>. Spre exemplu în [Lungu s.a.95] este formulată "rețeta" de aducere a unei relații în 1NF prin parcurgerea a patru pași:

- în relația universală, în locul atributelor compuse se trec componentele acestora, ca atribute simple;
- se constituie câte o relație pentru fiecare grup repetitiv;
- în schema fiecărei noi relații obținute la pasul doi se introduce și cheia primară a relației R nenormalizată;
- cheia primară a fiecărei noi relații va fi compusă din atributele/atributele cheie ale R plus unul sau mai multe atribute proprii.

Aplicând acest algoritm la relația BIBLIOTECĂ obținem patru tabele, COTE, AUTORI\_CĂRȚI, CĂRȚI\_CUVCHEIE și CĂRȚI cu structura din figura 3.8.

**COTE**

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

**CĂRȚI\_CUVCHEIE**

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

**AUTORI\_CĂRȚI**

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

**CĂRȚI**

ISBN	Titlu	Editura	LocSediuEd	AnApariție
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Polirom	Iași	2002

<sup>12</sup> Soluția este sugerată în [Lungu s.a.95], [Oprea99], [Connolly s.a.96], [Elmasri & Navathe 00] etc.

973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Polirom	Iași	2001
---------------	--	---------	------	------

Figura 3.8. O patra tentativă de aducere în 1NF a relației BIBLIOTECĂ

Structura obținută se detașează prin cel mai mic volum de redundanță de până acum, ceea ce antrenează și economii de stocare și reducerea sensibilă a anomaliilor de actualizare. Personal, însă, reproșez acestui stil de normalizare bazat pe eliminarea grupurilor repetitive caracterul prea “ochiometric”. Observarea grupurilor repetitive depinde decisiv de ordinea în care sunt dispuse atributele în relație (teoria relațională prevede clar că nici ordinea coloanelor, nici cea a liniilor nu influențează conținutul informațional al unei tabeli).

Să discutăm relația BIBLIOTECĂ\_2, ce are aceeași structură ca BIBLIOTECĂ, dar ordinea atributelor este schimbată – vezi figura 3.9.

### BIBLIOTECĂ\_2

Cotă	ISBN	Titlu	Autori
III-13421	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Marin Fotache, Ioan Brava, Cătălin Strâmbei, Liviu Crețu
III-13422	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Marin Fotache, Ioan Brava, Cătălin Strâmbei, Liviu Crețu
III-13423	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Marin Fotache, Ioan Brava, Cătălin Strâmbei, Liviu Crețu
III-10678	973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Marin Fotache
III-10679	973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Marin Fotache

Editura	LocSediuEd	AnAparitie	CuvinteCheie
Polirom	Iași	2002	baze de date, SQL, proceduri stocate, FoxPro, formulare, orientare pe obiecte, client-server, web
Polirom	Iași	2002	baze de date, SQL, proceduri stocate, FoxPro, formulare, orientare pe obiecte, client-server, web
Polirom	Iași	2002	baze de date, SQL, proceduri stocate, FoxPro, formulare, orientare pe obiecte, client-server, web
Polirom	Iași	2001	baze de date, algebră relațională, SQL
Polirom	Iași	2001	baze de date, algebră relațională, SQL

Figura 3.9. Relația BIBLIOTECĂ cu ordinea atributelor schimbată

Tabela BIBLIOTECĂ\_2 este mai aproape de realitate decât BIBLIOTECĂ. În depozit sunt trei exemplare ale cărții dedicate Visual FoxPro, acestora corespunzându-le trei tupluri în noua relație. Cheia primară este Cotă (observați că ISBN-ul se repetă pe primele trei linii și pe ultimele două).

După calapodul aplicat ceva mai sus, există două grupuri repetitive, Autori și CuvinteCheie. Drept care ne grăbim a descompune relația universală în trei relații aflate în 1NF ca în figura 3.10.

### COTE\_AUTORI

### COTE\_CUVCHEIE

Cotă	Autor	Cotă	CuvântCheie
III-13421	Marin Fotache	III-13421	baze de date
III-13421	Ioan Brava	III-13421	SQL
III-13421	Cătălin Strâmbei	III-13421	proceduri stocate
III-13421	Liviu Crețu	III-13421	FoxPro
III-13422	Marin Fotache	III-13421	formulare
III-13422	Ioan Brava	III-13421	orientare pe obiecte
III-13422	Cătălin Strâmbei	III-13421	client-server
III-13422	Liviu Crețu	III-13421	web
III-13423	Marin Fotache	III-13422	baze de date
III-13423	Ioan Brava	III-13422	SQL
III-13423	Cătălin Strâmbei	III-13422	proceduri stocate
III-13423	Liviu Crețu	III-13422	FoxPro
III-10678	Marin Fotache	III-13422	formulare
III-10679	Marin Fotache	III-13422	orientare pe obiecte
		III-13422	client-server
		III-13422	web
		III-13423	baze de date
		III-13423	SQL
		III-13423	proceduri stocate
		III-13423	FoxPro
		III-13423	formulare
		III-13423	orientare pe obiecte
		III-13423	client-server
		III-13423	web
		III-10678	baze de date
		III-10678	algebră relațională
		III-10678	SQL
		III-10679	baze de date
		III-10679	algebră relațională
		III-10679	SQL

## CĂRȚI2

Cota	ISBN	Titlu	Editura	LocSe- diuEd	AnApa- riție
III-13421	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Polirom	Iași	2002
III-13422	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Polirom	Iași	2002
III-13423	973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Polirom	Iași	2002
III-10678	973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Polirom	Iași	2001
III-10679	973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Polirom	Iași	2001

Figura 3.10. Prima formă normalizată a relației BIBLIOTECĂ\_2

Teoretic, baza de date din figura 3.10 este în 1NF. Schema sa diferă sensibil însă de aceeași BD care în 1NF arată ca în figura 3.8. Intuim că autorii și cuvintele se referă la o carte (ISBN), deci la toate exemplarele sale, deci suntem înclinați să alegem varianta din figura 3.8. Atunci când numărul atributelor este mare și

situația reflectată este complexă, identificarea grupurilor repetitive poate deveni, pe alocuri, o loterie.

Pentru remedierea problemelor datorate caracterului arbitrar al dispunerii atributelor în relație, am putea institui o regulă: se începe cu atributele cele mai generale ("non-repetitive") și se continuă apoi către cele mai "repetitive". În cazul relației BIBLIOTECĂ, s-ar putea începe cu ISBN-ul, continuând cu Titlu, Editură, LocSediuEd, AnApariție, apoi Autori și CuvinteCheie.

Similar, putem să descompunem relația universală FILMOGRAFIE pe baza următoarelor grupuri repetitive:

- Producători (atributul Producător);
- Regizori (atributul Regizor);
- Distribuție (atributele Rol și Actor);
- Genuri (atributul Gen);
- Premii (atributele DenPremiu, LocDecernare, Categorie, AnPremiu, Actor).

Obținem șase tabele corespunzătoare celor cinci grupuri repetitive, plus ce rămâne din relația inițială (FILME) - vezi figura 3.11.

### FILME

<u>IdFilm</u>	<u>TitluOriginal</u>	<u>TitluRO</u>	<u>AnLans</u>
11899	As Good As It Gets	Mai bine nu se poate	1997
12345	Bicentennial Man	Omul bicentenar	1999

### PRODUCĂTORI

<u>IdFilm</u>	<u>Producător</u>
11899	James Brooks
11899	Bridget Johnson
11899	Kristi Zea
12345	Michael Barnathan
12345	Chris Columbus
12345	Gail Katz

### REGIZORI

<u>IdFilm</u>	<u>Regizor</u>
11899	James Brooks
12345	Chris Columbus

### DISTRIBUȚIE

<u>IdFilm</u>	<u>Rol</u>	<u>Actor</u>
11899	Levin Udall	Jack Nicholson
11899	Carol Connelly	Helen Hunt
11899	Simon Bishop	Greg Kinnear
11899	Frank Sachs	Cuba Gooding Jr.
12345	Andrew Martin	Robin Williams
12345	Little Miss Amanda Martia	Embeth Davidtz
12345	Portia Charney	Embeth Davidtz
12345	Richard Martin	Sam Neil
12345	Rupert Burns	Oliver Platt
12345	Galatea	Kiersten Warren

### GENURI

<u>IdFilm</u>	<u>Gen</u>
11899	comedie
11899	dramă
11899	romantic
12345	SF
12345	dramă
12345	romantic

### PREMII

<u>IdFilm</u>	<u>DenPremiu</u>	<u>LocDe-</u>	<u>Categorie</u>	<u>AnPre-</u>	<u>Actor</u>
---------------	------------------	---------------	------------------	---------------	--------------

		<b>cernare</b>		<b>miu</b>	
11899	Oscar	Hollywood	cel mai bun actor în rol principal	1998	Jack Nicholson
11899	Oscar	Hollywood	cea mai bună actriță în rol principal	1998	Helen Hunt
11899	Globul de aur	New York	cea mai bună imagine	1998	NULL
11899	Globul de aur	New York	cel mai bun actor într-o comedie/musical	1998	Jack Nicholson
11899	Globul de aur	New York	cea mai bună actriță într-o comedie/musical	1998	Helen Hunt

Figura 3.11. Constituirea de relații separate pentru fiecare grup repetitiv pentru BD FILMOGRAFIE

Găselnița funcționează în cazul relației referitoare la cărțile din biblioteca FEAA și la filmografie. Schimbăm însă datele problemei. Luăm în discuție tabela DOTARE\_JUCĂRII din figura 3.12 care gestionează “dotarea” cu jucării a fiecărui copil al unei familii. În spiritul dreptului inviolabil la proprietate și al luptei contra colectivismului, fiecare jucărie aparține, cel puțin oficial, unui singur copil al familiei, eventualele împrumuturi fiind rodul exclusiv al negocierilor (și violenței juvenile).

### DOTARE\_JUCĂRII

<b>CodFamilie</b>	<b>NumeFamilie</b>	<b>Copil</b>	<b>Jucării</b>	<b>DataCumpărării</b>
1111	Popescu Ioan	Marc Loredana Elvis	Trenuleț electric Păpușă Barbie Puzzle 200 piese Puzzle 250 piese Puzzle 200 piese Căsuța Lego	14-05-1998 15-09-1998 23-12-1998 14-05-1999 24-12-1999 08-03-2000
...				

Figura 3.12. Relație ne-normalizată

Din examinarea relației de mai sus, reiese că există două grupuri repetitive: Copii și (Jucării, DataCumpărării). Prin aplicarea regulii de separare a grupurilor repetitive, obținem trei relații, COPII, JUCĂRII, FAMILII - vezi figura 3.13.

### COPII

<b>CodFamilie</b>	<b>Copil</b>
1111	Marc
1111	Loredana
1111	Elvis

### JUCĂRII

<b>CodFamilie</b>	<b>Jucării</b>	<b>DataCumpărării</b>
1111	Trenuleț electric	14-05-1998
1111	Păpușă Barbie	15-09-1998
1111	Puzzle 200 piese	23-12-1998
1111	Puzzle 250 piese	14-05-1999
1111	Puzzle 200 piese	24-12-1999
1111	Căsuța Lego	08-03-2000

### FAMILII

<b>CodFamilie</b>	<b>NumeFamilie</b>
1111	Popescu Ioan



Figura 3.13. Baza de date DOTARE\_JUCĂRII adusă în 1NF prin eliminarea grupurilor repetitive

Deși nu am făcut decât să urmărim un algoritm prescris ca sigur, avem de a face cu un caz tipic de descompunere cu pierdere de informații. În cazul nostru, cele trei relații nu furnizează o informație esențială: cărui copil îi aparține o jucărie anume.

Oricum am proceda, eliminarea grupurilor repetitive nu rezolvă corect întotdeauna problema normalizării (aduceri relației în 1NF și continuării apoi cu 2NF s.a.m.d.), deoarece identificarea lor se face prin observație și nu pe baza unei analize semantice riguroase a relațiilor dintre atribute.

Deși atrage un grad de redundanță considerabil, este recomandabil ca în această situație 1NF a relației să fie obținută prin introducerea în tabelă a unei linii pentru fiecare jucărie și duplicarea valorilor celorlate atribute, ca în figura 3.14.

#### DOTARE\_JUCĂRII\_1

<u>CodFamilie</u>	<u>NumeFamilie</u>	<u>Copil</u>	<u>Jucărie</u>	<u>DataCumpărării</u>
1111	Popescu Ioan	Marc	Trenuleț electric	14-05-1998
1111	Popescu Ioan	Loredana	Păpușă Barbie	15-09-1998
1111	Popescu Ioan	Elvis	Puzzle 200 piese	23-12-1998
1111	Popescu Ioan	Loredana	Puzzle 250 piese	14-05-1999
1111	Popescu Ioan	Marc	Puzzle 200 piese	24-12-1999
1111	Popescu Ioan	Loredana	Căsuța Lego	08-03-2000

Figura 3.14. Relația adusă în 1NF

Esențială în rezolvarea problemei este conștientizarea entității căreia i se alocă o linie în tabelă. În cadrul relației DOTARE\_JUCĂRII\_1, fiecare linie se referă la o jucărie afectată unui copil al unei familii.

Nici relația universală VÎNZĂRI, ale cărei atribute sunt cele din tabelul 2.1, nu este scutită de probleme la identificarea atributelor repetitive, dar amânăm discuția pentru paragraful următor.

### 3.4. Prima formă normală și problema cheii primare a relației universale

De cele mai multe ori, constituirea unei relații universale atotcuprinzătoare se lovește de o problemă majoră – cea a cheii primare. Asupra acestui necaz, câțiva autori au atras atenția cu ceva timp în urmă<sup>13</sup>. Modelul relațional prezintă două restricții esențiale vis-a-vis de cheia primară: unicitate și valori nenule pentru atributele din alcătuirea cheii (restricția de entitate). Or, dacă dorim să identificăm fiecare linie a relației, trebuie să definim, în cele mai multe situații, o cheie primară cu multe atribute, care, însă, în anumite tupluri ar avea valori nule. Pe de altă

<sup>13</sup> Vezi, spre exemplu, [Kent81]

parte, dacă nu vrem să fie violată restricția de entitate, trebuie să renunțăm la a prelua în relația universală linii vitale din punctul de vedere al conținutului informațional al bazei de date.

Să luăm în discuție câteva cazuri.

### Vechea codificare poștală, valabilă pâna în mai 2003

Până în luna mai 2003 codurile poștale erau atribuite la nivel de oraș și comună (cu unele excepții), ca în figura 3.15. Dacă într-o comună erau mai multe sate, era posibil ca unele dintre acestea să partajeze același cod poștal. Singurul oraș pentru care existau mai multe coduri era Bucureștiul.

#### LOCALITĂȚI\_CODURI\_VECHI

CodPoștal	OrașComună	Sat	Județ
5319	Vînători	Vînători	Vrancea
5319	Vînători	Jorăști	Vrancea
5319	Vînători	Mirceștii-Vechi	Vrancea
5613	Roznov	Roznov	Neamț
5613	Roznov	Slobozia	Neamț
5300	Focșani	NULL	Vrancea

Figura 3.15. Vechile coduri poștale - tabela nu este în 1NF

O localitate desemnează un oraș sau un sat. Ei bine, cheia primară ar trebui să fie combinația (CodPoștal, OrașComună, Sat). Dar valoarea atributului Sat este NULLă atunci când linia respectivă se referă la un oraș. Prin urmare, relația de mai sus nu se află în prima formă normală, deoarece încalcă restricția de entitate (un atribut din cheie prezintă valori nule). Practic, pentru a fi în 1NF, relația nu trebuie să conțină orașele, ceea ce este de neacceptat. Aducerea sa în 1NF se realizează prin intermediul depedendențelor funcționale. Este drept, că putem păcăli teoria și SGBD-urile actuale, definind pentru atributul Sat o valoare implicită nenulă – spre exemplu un spațiu. Aceasta miroase însă a cârpeală mioritică și vom vedea în paragraful următor că avem soluții mai onorabile.

### Noua codificare poștală, valabilă după mai 2003

Sistemul de codificare poștală introdus în mai 2003 este mult mai fin, în sensul că un cod are acum șase cifre, în loc de patru, iar alocarea se mai face la nivel de localitate doar în cazul satelor și orașelor mai mici; în rest, la nivel de stradă sau chiar imobil - vezi figura 3.16.

Astfel, în tabela CODURI\_NOI\_V1 prima linie indică faptul că localitatea Vînători din județul Vrancea are codul poștal 627395, localitatea fiind și "capitală" de comună, cel puțin după nume. Următoarele două linii se referă la alte două sate din aceeași comună. Valorile NULL ale atributelor Strada și Numere pe aceste prime trei linii semnalizează faptul că pentru aceste sate nu se operează o delimitare a codurilor pe străzi (de fapt, ulițe).

Lucrurile devin cu adevărat interesante odată cu liniile 4-7 care conțin coduri poștale ale unor adrese (locații, în limbaj contemporan) din municipiul Iași. Pentru

simplificare, toate cele patru coduri sunt atribuite unor adrese aflate pe bulevardul *Independenței*. Astfel, codul 700106 este asociat tuturor imobilelor din *Bd. Independenței*, iar valoarea 1-5, 9-13 pentru atributul *Numere* desemnează următoarele numere: 1, 3, 5 (1-5) și 9, 11, 13 (9-13). Valoarea aceluiași atribut pe linia următoare (codul 700099) - 18-T indică un set de valori pare ce începe cu 18 și continuă cu 20, 22, ... până la ultima adresă pară de pe bulevardul *Independenței*. T-ul înseamnă deci "Terminare", epuizare, și asigură atribuirea automată a acestui cod poștal eventualelor viitoare construcții ridicate pe acest bulevard (dacă o mai fi loc). Pe linia de mai jos valoarea 25-T se referă la toate numele impare începând cu 25 până la ultimul imobil situat pe acea parte ("impară") a bulevardului. Penultima linie a tabelului semnalează un lucru oarecum deranjant: un cod poștal poate fi alocat chiar și la două sau mai multe străzi, în cazul nostru, numerelor pare de la 2 la 26 și numărului 28 de pe *bulevardul Carol I*, dar și tuturor numelor de pe *aleea Veronica Micle* (în tabelă este preluată titulatura de pe [www.posta-romana.ro](http://www.posta-romana.ro)).

### CODURI\_NOI\_V1

CodPoștal	Localitate	Strada	Numere	Comuna	Județ
627395	Vînători	NULL	NULL	Vînători	Vrancea
627397	Jorăști	NULL	NULL	Vînători	Vrancea
627399	Mirceștii-Vechi	NULL	NULL	Vînători	Vrancea
700106	lași	Bd. Independenței	1-5 9-13	NULL	lași
700099	lași	Bd. Independenței	18-T	NULL	lași
700102	lași	Bd. Independenței	25-T	NULL	lași
700100	lași	Bd. Independenței	7-13	NULL	lași
700505	lași	Bd. Carol I Aleea Micle Veronica	2-26 28 T	NULL	lași
700482	lași	Bd. Carol I	26A 30-32 36	NULL	lași
700504	lași	Bd. Carol I	28A 34 38 42	NULL	lași
707295	Mircești	NULL	NULL	Mircești	lași

Figura 3.16. Noile coduri poștale - tabelă nenormalizată

Firește, ne punem întrebarea: de ce codul 700505 nu a fost atribuit succesiunii 2-28 de pe *Bd. Carol I*, ci avem două intervale, 2-26 și 28-28 ? Răspunsul ni-l oferă rândul următor, din care aflăm că 26A de pe *Bd. Carol I* are alt cod - 700482 ! Prin urmare, același număr poate fi alocat codurilor diferite, diferențierea făcându-se prin litere sau binecunoscutul *bis*.

Cheia primară este atributul *CodPoștal*, însă relația este departe de 1NF. Avem de a face cu două atribute multivaloare: *Strada* și *Numere*. O primă idee ar fi să rupem relația, constuind câte o tabelă pentru fiecare grup repetitiv, ca în figura 3.17.

### CODURI\_LOCALITĂȚI

CodPoștal	Localitate	Comuna	Județ
627395	Vînători	Vînători	Vrancea
627397	Jorăști	Vînători	Vrancea
627399	Mirceștii-Vechi	Vînători	Vrancea
700106	Iași	NULL	Iași
700099	Iași	NULL	Iași
700102	Iași	NULL	Iași
700100	Iași	NULL	Iași
700505	Iași	NULL	Iași
700482	Iași	NULL	Iași
700504	Iași	NULL	Iași
707295	Mircești	Mircești	Iași

### CODURI\_STRĂZI

CodPoștal	Strada
700106	Bd. Independenței
700099	Bd. Independenței
700102	Bd. Independenței
700100	Bd. Independenței
700505	Bd. Carol I
700505	Aleea Micle Veronica
700482	Bd. Carol I
700504	Bd. Carol I

### CODURI\_NUMERE

CodPoștal	Numere
700106	1-5
700106	9-13
700099	18-T
700102	25-T
700100	7-13
700505	2-26
700505	28
700505	T
700482	26A
700482	30-32
700482	36
700504	28A
700504	34
700504	38
700504	42

Figura 3.17. Tentativă eșuată de normalizare a relației CODURI\_NOI\_V1

După cum am mai pățit-o în paragraful anterior, ruperea relației inițiale s-a făcut cu pierderi de informații. Deoarece un cod poștal poate fi alocat la două străzi diferite (vezi 700505), numerele din CODURI\_NUMERE nu pot fi alocate corect întotdeauna.

O altă cale ar fi să înlocuim atributul *Numere* cu nu mai puțin de patru câmpuri, pentru a câștiga în limpezime informațională, după cum urmează:

- *TipNr* (tip număr) care să conțină doar valorile:
  - NULL - pentru localități fără o delimitare oficială pe străzi;
  - "pare" - adică 1, 3, 5, ...;
  - "impare" - adică 2, 4, 6;
  - "ambele" - adică numere consecutive;
- *NrInițial* (numărul inițial) indică numărul limită inferioară a intervalului;
- *LitInițială* (litera inițială, dacă este cazul) se folosește pentru a indica dacă intervalul pornește de la 11B sau 24A etc.;

- **NrFinal** (numărul final din interval) indică numărul limita superioară a intervalului;
- **LitFinală** (litera finală) semnalizează că intervalul se oprește la 11B sau 24A etc.

Astfel, tabela de mai sus ar prelua forma din figura 3.18.

### CODURI\_NOI\_V2

CodPoștal	Localitate	Strada	TipNr	NrInițial	LitInițială
627395	Vînători	NULL	NULL	NULL	NULL
627397	Jorăști	NULL	NULL	NULL	NULL
627399	Mirceștii-Vechi	NULL	NULL	NULL	NULL
700106	Iași	Bd. Independenței	impare	1	NULL
700106	Iași	Bd. Independenței	impare	9	NULL
700099	Iași	Bd. Independenței	pare	18	NULL
700102	Iași	Bd. Independenței	impare	25	NULL
700100	Iași	Bd. Independenței	impar	7	NULL
700505	Iași	Bd. Carol I	pare	2	NULL
700505	Iași	Bd. Carol I	pare	28	NULL
700505	Iași	Aleea Micle Veronica	ambele	1	NULL
700482	Iași	Bd. Carol I	pare	26	A
700482	Iași	Bd. Carol I	pare	30	NULL
700482	Iași	Bd. Carol I	pare	36	NULL
700504	Iași	Bd. Carol I	pare	28	A
700504	Iași	Bd. Carol I	pare	34	NULL
700504	Iași	Bd. Carol I	pare	38	NULL
700504	Iași	Bd. Carol I	pare	42	NULL
707295	Mircești	NULL	NULL	NULL	NULL

NrFinal	LitFinală	Comuna	Județ
NULL	NULL	Vînători	Vrancea
NULL	NULL	Vînători	Vrancea
NULL	NULL	Vînători	Vrancea
5	NULL	NULL	Iași
13	NULL	NULL	Iași
NULL	NULL	NULL	Iași
NULL	NULL	NULL	Iași
13	NULL	NULL	Iași
26	NULL	NULL	Iași
28	NULL	NULL	Iași
NULL	NULL	NULL	Iași
26	A	NULL	Iași
32	NULL	NULL	Iași
36	NULL	NULL	Iași
28	A	NULL	Iași
34	NULL	NULL	Iași
38	NULL	NULL	Iași
42	NULL	NULL	Iași
NULL	NULL	Mircești	Iași

Figura 3.18. Noile coduri poștale - tabelă normalizată dar fără cheie primară

Spre deosebire de forma nenormalizată, tabela de mai sus ar permite găsirea imediată a codului poștal al unei adrese de genul *Bd. Carol I nr 29*. Deși am putea folosi un limbaj mai abstract de genul celui din ediția a opta a lucrării lui Chris Date, vom recurge la SQL:

```
SELECT codpostal
FROM CODURI_NOI_V2
WHERE județ = 'Iași' AND localitate='Iași' AND Strada='Bd.
Carol I' AND
(
  (nrinițial = 29 and litinițială IS NULL) OR
  (nrfinal = 29 and litfinală IS NULL)
OR
  (tipnr <> 'pare' AND nrinițial < 29 AND nrfinal > 29 )
)
```

Ca exercițiu pentru acasă, încercați să aflați aceeași informație din schema nenormalizată a tablei (CODURI\_NOI\_V2).

În modelul relațional este obligatoriu ca orice relație să posede cheie primară, alcătuită, la limită, din toate atributele relației. Altfel spus, dacă în SQL o tabelă poate avea linii identice, în teoria relațională nu. Este una dintre "îmbunătățirile" SQL care au atras mânia multor autori majori în domeniul bazelor de date relaționale, precum E.F. Codd, C.J. Date, H. Darwen, F. Pascal.

Ei bine, pentru a identifica un tuplu unic în relația de mai sus am avea nevoie de combinația (CodPoștal, Strada, NrInițial, LitInițială). Or, în majoritatea tuplurilor tablei de mai sus, cel puțin unul dintre acestea are valori nule. După cum observăm, problema vine din regimul diferit al localităților fără codificare la nivel străzi (satele și micile orașe). Așa încât vom sparge relația CODURI\_NOI\_V2 în două, CODURI\_SATE și CODURI\_ORAȘE. Denumirea este discutabilă, deoarece este posibil să existe sate mai mari, în curs de a deveni orașe, cărora să li se repartizeze mai multe coduri. Pentru simplitate, păstrăm cele două denumiri, iar conținutul ar fi cel din figura 3.19.

#### CODURI\_SATE

CodPoștal	Localitate	Comuna	Județ
627395	Vînători	Vînători	Vrancea
627397	Jorăști	Vînători	Vrancea
627399	Mircești-Vechi	Vînători	Vrancea
707295	Mircești	Mircești	Iași

#### CODURI\_ORAȘE

Cod Poștal	Localitate	Strada	TipNr	NrInițial	LitInițială	NrFinal	LitFinală	Județ
700106	Iași	Bd. Independenței	impare	1		5	NULL	Iași
700106	Iași	Bd. Independenței	impare	9		13	NULL	Iași
700099	Iași	Bd. Independenței	pare	18		NULL	NULL	Iași
700102	Iași	Bd. Independenței	impare	25		NULL	NULL	Iași
700100	Iași	Bd. Independenței	impar	7		13	NULL	Iași

700505	Iași	Bd. Carol I	pare	2		26	NULL	Iași
700505	Iași	Bd. Carol I	pare	28		28	NULL	Iași
700505	Iași	Aleea Micle Veronica	ambele	1		NULL	NULL	Iași
700482	Iași	Bd. Carol I	pare	26	A	26	A	Iași
700482	Iași	Bd. Carol I	pare	30		32	NULL	Iași
700482	Iași	Bd. Carol I	pare	36		36	NULL	Iași
700504	Iași	Bd. Carol I	pare	28	A	28	A	Iași
700504	Iași	Bd. Carol I	pare	34		34	NULL	Iași
700504	Iași	Bd. Carol I	pare	38		38	NULL	Iași
700504	Iași	Bd. Carol I	pare	42		42	NULL	Iași

Figura 3.19. Tabele separate pentru localitățile cu un singur cod și pentru localitățile cu mai multe coduri

CODURI\_SATE are o mândrețe de cheie primară: CodPoștal. La CODURI\_ORAȘE n-am scăpat de improvizații. Astfel, pentru a evita nulitatea literei inițiale, am folosit spațiul, așa încât cheia primară este (CodPoștal, Strada, NrInițial, LitInițială).

Am rezolvat o problemă și am dat de alta: la majoritatea căutărilor trebuie să efectuăm în prealabil reuniunea celor două tabele, ceea ce e destul de apăsător. Spre exemplu, primul și ultimul cod poștal alocat județului Iași este necesar o interogare de genul:

```
SELECT MAX(codpoștal) AS cod_minim,
       MAX(codpoștal) AS cod_maxim
FROM
  (SELECT codpoștal, localitate, NULL AS strada,
         NULL AS tipnr, NULL AS nrinițial,
         NULL AS litinițială, NULL AS nrfinal,
         NULL AS litfinală, comuna, județ
   FROM coduri_sate
  UNION
   SELECT codpoștal, localitate, stradă, tipnr, nrinițial,
         litinițială, nrfinal, litfinală, NULL, județu
   FROM coduri_orașe)
WHERE judet='Iași'
```

Marea majoritate a practicienilor preferă soluția *gordiană*, adică folosirea unei chei surogat care este unică pentru fiecare linie (atributele de tip cheie surogat pot fi în majoritatea SGBD-urilor gestionate automat) - vezi tabela CODURI\_NOI\_V3 din figura 3.20. Risipa de spațiu generată de introducerea acestui atribut reprezintă un preț rezonabil la durerile de cap pricinuite de valorile nule ale "fostelor" attribute cheie. Atributul Id nu are nici o relevanță informațională, ci doar ne ajută să asigurăm unicitatea cheii primare fără pericolul încălcării restricției de entitate.

### CODURI\_NOI\_V3

Id	CodPoștal	Localitate	Strada	TipNr	NrInițial
3456789	627395	Vînători	NULL	NULL	NULL
3456790	627397	Jorăști	NULL	NULL	NULL
3456791	627399	Mirceștii-Vechi	NULL	NULL	NULL

3456792	700106	lași	Bd. Independenței	impare	1
3456793	700106	lași	Bd. Independenței	impare	9
3456794	700099	lași	Bd. Independenței	pare	18
3456795	700102	lași	Bd. Independenței	impare	25
3456796	700100	lași	Bd. Independenței	impar	7
3456797	700505	lași	Bd. Carol I	pare	2
3456798	700505	lași	Bd. Carol I	pare	28
3456799	700505	lași	Aleea Micle Veronica	ambele	1
3456800	700482	lași	Bd. Carol I	pare	26
3456801	700482	lași	Bd. Carol I	pare	30
3456802	700482	lași	Bd. Carol I	pare	36
3456803	700504	lași	Bd. Carol I	pare	28
3456804	700504	lași	Bd. Carol I	pare	34
3456805	700504	lași	Bd. Carol I	pare	38
3456806	700504	lași	Bd. Carol I	pare	42
3456807	707295	Mircești	NULL	NULL	NULL

LitInițială	NrFinal	LitFinală	Comuna	Județ
NULL	NULL	NULL	Vânători	Vrancea
NULL	NULL	NULL	Vânători	Vrancea
NULL	NULL	NULL	Vânători	Vrancea
NULL	5	NULL	NULL	lași
NULL	13	NULL	NULL	lași
NULL	NULL	NULL	NULL	lași
NULL	NULL	NULL	NULL	lași
NULL	13	NULL	NULL	lași
NULL	26	NULL	NULL	lași
NULL	28	NULL	NULL	lași
NULL	NULL	NULL	NULL	lași
A	26	A	NULL	lași
NULL	32	NULL	NULL	lași
NULL	36	NULL	NULL	lași
A	28	A	NULL	lași
NULL	34	NULL	NULL	lași
NULL	38	NULL	NULL	lași
NULL	42	NULL	NULL	lași
NULL	NULL	NULL	Mircești	lași

Figura 3.20. Noile coduri poștale - tabelă normalizată dar fără cheie primară

Atenție, însă ! Cheile surogat constituie o tentație căreia (ca oricărei tentații) trebuie să i se dea curs cu moderație.

### Baza de date VÎNZĂRI

Și relația universală a bazei de date VÎNZĂRI "suferă" de aceeași problemă a cheii primare, și, implicit, prima formă normală este mai mult decât discutabilă. Conform "preceptelor" relaționale, VÎNZĂRI {Jud, Județ, Regiune, CodPost, Localitate, Comună, CodCl, DenCl, CodFiscal, StradaCl, NrStradaCl, BlocScApCl, TelefonCl, CodPr, DenPr, UM, Grupa, ProcTVA, NrFact, DataFact, Obs, Linie, Cantitate, PrețUnit, CodÎnc, DataÎnc, CodDoc, NrDoc, DataDoc, Tranșă} trebuie să prezinte o combinație de atribute a cărei valori conferă unicitate fiecărui tuplu.



Ținând seama că:

- un client are un singur sediu central (poate avea filiale, însă ne limităm la a considera că facturarea se face pe adresa sediului central);
- o factură emisă are un număr unic și se întocmește unui singur client;
- o factură conține una sau mai multe linii;
- pe fiecare linie este înregistrat un produs, fiecare vândut într-o anumită cantitate și la un anumit preț unitar;
- deoarece este posibilă re-diferențierea procentului de taxă pe valoarea adăugată în funcția de categoria produselor (alimentare, medicamente, cărți, bunuri considerate de lux etc.), *ProcTVA* este bine de asociat fiecărui produs;
- o încasare are la bază un document justificativ;
- o factură poate de plătită de client în tranșe;
- la o încasare se pot achita, integral sau parțial, una sau mai multe facturi,

cheia primară ar fi combinația (*NrFact*, *Linie*, *CodÎnc*). Necazul este că la momentul întocmirii nu se cunosc datele despre încasări. Mai mult, aceasta poate surveni la distanțe mari în timp, mai ales în condițiile blocajelor financiare care ne-au făcut celebri în mediul economic european.

Potrivit primei soluții care nu încalcă restricția de entitate, inserarea de linii în tabela *VÎNZĂRI* se face abia în momentul încasării facturii, ceea ce este inadmisibil, întrucât baza de date va fi incapabilă să furnizeze informații vitale precum:

- vânzările pe ziua, săptămâna, luna, anul curente;
- vânzările pe produse;
- vânzările pe clienți;
- creanțele (valoarea rămasă de încasat) față de clienți.

A doua soluție este spargerea relației universale. Cum ? Prin eliminarea grupurilor repetitive. Dar care sunt grupurile repetitive, în condițiile în care o factură poate fi achitată în tranșe (mai multe încasări), iar o încasare poate "regla" mai multe facturi ? Vom vedea în capitolele viitoare cum, pe baza dependențelor funcționale, putem obține o structură mult mai acceptabilă.

## Baza de date FILMOGRAFIE

În paragraful anterior am obținut o schemă ce se apropie destul de mult de rezonabil, cel puțin în materie de eliminare a grupurilor repetitive. Dintre cele șase tabele ale bazei de date FILMOGRAFIE din figura 3.11 să examinăm preț de câteva minute pe ultima - *PREMIU*. Cheia primară este combinația (*IdFilm*, *DenPremiu*, *Categorie*, *AnPremiu*) în condițiile în care orice premiu de interpretare se acordă unui singur actor/actriță. Ce se fi întâmplă însă dacă *Oscarul pentru cel mai bun actor în rol principal* ar fi fost acordat simultan lui Robert de Niro și Al Pacino pentru prestația lor dintr-un film celebru, în care, practic, nu putem spune că unul din roluri este principal, iar celălalt secundar. Toate cele patru atribute ale cheii ar

prezenta valori identice pe două tupluri (un tuplu pentru de Niro, celălalt pentru Pacino), iar restricția de cheie primară ar fi încălcată.

Când proiectăm baze de date trebuie să ne gândim și la astfel de spețe marginale ce pot compromite o structură chiar după ani buni de funcționare a aplicației. Fiind tentați să adăugăm celor patru atribute din cheie pe al cincilea, *Actor*, nu putem să nu observăm că, pentru premiile acordate la categoriile regie, scenariu etc., acest atribut are valori NULLe, ceea ce violează restricția de entitate.

Soluția cea mai la îndemână ține de ruperea acestei relații în două, una dedicată premiilor de interpretare, iar a doua celorlalte premii - vezi figura 3.21.

### PREMII INTERPRETARE

<u>IdFilm</u>	<u>DenPremiu</u>	<u>LocDe-cernare</u>	<u>Categorie</u>	<u>AnPre-miu</u>	<u>Actor</u>
11899	Oscar	Hollywood	cel mai bun actor în rol principal	1998	Jack Nicholson
11899	Oscar	Hollywood	cea mai bună actriță în rol principal	1998	Helen Hunt
11899	Globul de aur	New York	cel mai bun actor într-o comedie/musical	1998	Jack Nicholson
11899	Globul de aur	New York	cea mai bună actriță într-o comedie/musical	1998	Helen Hunt

### CELELALTE PREMII

<u>IdFilm</u>	<u>DenPremiu</u>	<u>LocDe-cernare</u>	<u>Categorie</u>	<u>AnPre-miu</u>
11899	Globul de aur	New York	cea mai bună imagine	1998

Figura 3.21. Ruperea relației PREMII din cauza problemelor de cheie primară

## 3.5. Dezbateri pe marginea și în centrul primei forme normale

Prima formă normală este, fără îndoială, un ciudat amestec de facil și confuz. Putem spune chiar că mare parte din doza de facil, de superficialitate din cursurile universitare, cărțile dedicate bazelor de date sau chiar proiectării bazelor de date (inclusiv normalizării) își au sorgintea tocmai în confuzia care a învăluit, încă de la începuturile relaționalului, definirea primei forme normale și atomicității<sup>14</sup>. Unii autori, precum Elmasri și Navathe, consideră restricția de atomicitate proprie modelului relațional, restricție care este eliminată din modelele relațional imbricat (nested relational model) și obiectual-relațional care, ambele, permit relații nenormalizate<sup>15</sup>.

<sup>14</sup> Există lucrări respectabile, precum [Atzeni s.a. 99], care nici nu fac aluzie la prima formă normală sau atomicitatea valorilor într-o relație

<sup>15</sup> [Elmasri & Navathe 00], p. 485

Ceea ce se înțelegem îndeobște prin atomicitatea atributelor ține de caracterul scalar al valorilor: întregi, șiruri de caractere, date calendaristice. Prin comparație, un atribut nonatomic este unul definit pe un domeniu de valori compozite, complexe. Fiecare componentă dintr-o valoare compozită poate fi, la rândul ei, compozită, ajungându-se la o structură ierarhică și flexibilă, în funcție de natura obiectului sau procesului modelat. Elmasri și Navathe sunt cât se poate de limpezi: pentru atributele unei relații în 1NF, singurele valori permise sunt cele atomice sau indivizibile<sup>16</sup>.

În ceea ce mă privește, materialul favorit legat de 1NF este cel al lui Chris Date publicat în iunie 2003 pe site-ul <http://www.dbdebunk.com><sup>17</sup>, site conceput și gestionat de un apropiat al lui Date și un necruțător observator al ignoranței comunității IT (și academice) în materie de baze de date - Fabian Pascal. Materialul evocat constituie deopotrivă încununarea și convergența lucrărilor scrise singur (cea mai celebră este cartea a cărei a opta ediție a fost tradusă și în românește) sau împreună cu Hugh Darwen (vezi *The Third Manifesto*). Meritoriu este și faptul că, din capul locului, Date se autodenunță ca fiind unul dintre cei care, de-a lungul timpului, au contribuit la confuzia în care se scaldă atomicitatea.

Prima definiție a 1NF aparține, firește, lui E.F. Codd pentru care o relație este în 1NF dacă nici unul dintre domeniile sale nu are elemente (valori) de tip set<sup>18</sup>. Câțiva ani mai târziu, Codd afirmă că un domeniu este simplu dacă toate valorile sale sunt atomice, adică *nedeconpozabile de către SGBD*<sup>19</sup>. Date identifică în cartea lui Codd publicată în 1990<sup>20</sup> două afirmații similare:

- valorile din domeniile pe care fiecare relație este definită sunt necesarmente atomice vis-a-vis de SGBD;
- datele atomice sunt cele care nu pot fi descompuse de către SGBD în subcomponente (cu excepția unor funcții speciale).

Din păcate, Codd elimină o neclaritate înlocuind-o cu alta. Și Date se întreabă: ce înseamnă *cu excepția unor funcții speciale* ? Faptul că dintr-un șir de caractere (exemplul clasic de valoare atomică) pot fi extrase diferite componente prin funcții precum SUBSTR, funcții prezente în mai toate SGBD-urile actuale, face din valoarea de tip șir de caractere una nonatomică ? Sună cam hilar. Mai ales că, dacă raportăm atomicitatea la SGBD-uri, deseori vom fi în situația în care o valoare atomică într-o bază de date gestionată cu un SGBD să fie nonatomică în cazul unei aceleași baze de date (structuri) implementată pe un alt SGBD. Ceea ce, să recunoaștem, nu e prea onorant pentru un model atât de riguros fundamentat așa cum este relaționalul.

Date, împreună cu Darwen și Pascal propun renunțarea la atomicitate ca o cerință a 1NF. După Date, atomicitatea nici nu are o semnificație absolută, întrucât

---

<sup>16</sup> [Elmasri & Navathe 00], p. 485

<sup>17</sup> [Date03]

<sup>18</sup> [Codd72]. Preluare din [Date03]

<sup>19</sup> [Codd79]

<sup>20</sup> [Codd90]

depinde de ceea ce dorim să facem cu datele asupra atomicității cărora ne pronunțăm<sup>21</sup>. Practic, toate tabelele care respectă principiile modelului relațional sunt în 1NF, iar această primă normală păstrează din relevanță doar în două privințe:

- indică faptul că relația la care se referă poate să nu fie într-o formă normalizată superioară (2NF, 3NF...);
- o structură de date ce respectă 1NF este una relațională, cu alte cuvinte tabelele care nu sunt în 1NF sunt non-relaționale.

Acestui din urmă aspect merită o discuție detaliată, deoarece ține de aspectul structural al modelului relațional.

Renunțarea la atomicitate nu este singura schimbare fundamentală în modelul relațional. Domeniile pot conține orice tip de valori: scalare sau compozite, șiruri de caractere, numere, dar și vectori, liste, imagini, înregistrări audio/video, documente XML sau orice tip de definit de utilizatori. Dealminteri, dacă inițial Codd s-a străduit să impună sintagma *domeniu* în detrimentul *tipului*, tocmai pentru a opera o distincție netă dintre baze de date și limbaje de programare, Date, Darwen și Pascal sunt, în ultimul timp, mult mai apropiați de noțiunea de *tip*, pe care o consideră mai sugestivă și, în plus, identic aplicabilă și altor modele de organizare a datelor date. Atâta vreme cât există suport din partea SGBD-ului pentru definirea, stocarea și accesarea sa, se poate folosi orice tip de date într-o BD.

Revenim la relația BIBLIOTECĂ din figura 3.1. Pe baza tipului *șir de caractere*, putem defini un domeniu de tip *vectori de șiruri de caractere*, iar cele trei atribute, *Cote*, *Autori* și *CuvinteCheie* se vor declara pe acest nou domeniu de tip set. Chiar dacă noua relație, denumită BIBLIOTECĂ\_SET (figura 3.22), seamănă izbitor cu forma denormalizată, este de o schemă radical diferită.

### BIBLIOTECĂ\_SET

ISBN	Titlu	Cote_SET	Autori_SET
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	{III-13421, III-13422, III-13423}	{Marin Fotache, Ioan Brava, Cătălin Strâmbei, Liviu Crețu}
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	{III-10678, III-10679}	{Marin Fotache}

  

Editura	LocSediuEd	AnApariție	CuvinteCheie_SET
Polirom	Iași	2002	{baze de date, SQL, proceduri stocate, FoxPro, formulare, orientare pe obiecte, client-server, web}
Polirom	Iași	2001	{baze de date, algebră relațională, SQL}

Figura 3.22. Domenii de tip set

A doua inovație o constituie atributele ale căror valori pot fi chiar relații (*relation-valued attributes*) - ATR. În exemplul nostru, în locul celor trei seturi putem folosi câte o relație. Câștigul este important, deoarece pentru ingredientul folosit este pur relațional (relația !), iar mecanismul de declarare a restricțiilor și cel de

<sup>21</sup> [Date03]

interogare (algebra relațională) este, în esență, același. Atributele *Cote\_REL*, *Autori\_REL* și *CuvinteCheie\_REL* din relația *BIBLIOTECĂ\_ATR* (figura 3.23) sunt de acest tip, iar valorile lor pot fi supuse operatorilor clasici: selecție, proiecție, joncțiune etc. Date și Harwen propun câțiva noi operatori algebrici relaționali pentru a asigura comparabilitatea tabelelor cu și fără *ATR*, *GROUP* și *UNGROUP*<sup>22</sup>.

### BIBLIOTECĂ\_ATR

ISBN	Titlu	Cote_REL
973-683-889-7	Visual FoxPro. Ghidul dezvoltării aplicațiilor profesionale	Cote
		III-13421
		III-13422
		III-13423
973-683-709-2	SQL. Dialecte DB2, Oracle și Visual FoxPro	Cote
		III-10678
		III-10679

Autori_REL	Editura	LocSediuEd	AnApariție	CuvinteCheie_REL
<b>Autori</b> Marin Fotache Ioan Brava Cătălin Strâmbei Liviu Crețu	Polirom	Iași	2002	<b>CuvinteCheie</b>
				baze de date
				SQL
				proceduri stocate
				FoxPro
				formulare
<b>Autori</b> Marin Fotache	Polirom	Iași	2001	orientare pe obiecte
				client-server
				web
				<b>CuvinteCheie</b>
				baze de date
				algebră relațională
				Stocate

Figura 3.23. Atribute de tip relații (ATR)

Din contră, dacă am folosi prima variantă - cea a seturilor - ar fi necesari operatori speciali: reuniune de seturi, intersecție de seturi etc.

Această a doua inovație a modelului relațional este o consecință directă a primeia. Practic, dacă un atribut *X* este de tip relație, iar *D* este domeniul lui *X*, atunci toate valorile lui *D* sunt relații. În plus, orice atribut de tip relație poate conține atribute care sunt, la rândul lor, de tip relații, astfel încât numărul nivelurilor de imbricare este nelimitat.

<sup>22</sup> Vezi [Date & Darwen00]

Interesantă este în acest sens și lucrarea publicată în 1984 de către Serge Abiteboul și Nicole Bidoit. Deși autorii afirmă că propun un model de date nou (l-au botezat Verso), în mare idee era ca valoarea unui atribut să fie nu numai atomică, ci și o instanță a unui *format* (autorii au ezitat să folosească termenul *relație*). Ceea ce este notabil la modelul propus este că prin acest mecanism era obținută o ierarhie ce putea fi interogată folosind aceeași operatori algebrici relaționali. Cu atât mai mult cu cât valorile nule aveau o formă de reprezentare care ar mulțumi astăzi pe Date și compania (relații vide)<sup>23</sup>.

Howard Dreizen și Shi-Huo Chang propun în numărul din decembrie 1989 al *ACM Transactions on Database Systems* acceptarea, cu titlu restrictiv, a unor condiții excepționale în schema bazei de date, pentru a rezolva o serie de situații practice relativ rare care însă crează anomalii în schema bazei. Cu acest prilej, autorii se pronunță pentru folosirea *relațiilor incluse*, altfel spus, includerea relațiilor în alte relații, de o manieră nerestrictivă; astfel, o relație  $R(D_1, D_2, \dots, D_n)$  de  $n$  domenii este recursivă omogenă dacă fiecare domeniu este fie (1) un set de valori atomice fie (2) un set de relații recursive omogene cu scheme identice<sup>24</sup>.

Odată trecută euforia contactului cu atributele de tip relații, ne putem întreba retoric, precum Date: în proiectarea schemei bazei chiar avem nevoie de ATR-uri? Întrebarea corectă ar fi, mai degrabă: când ATR-urile sunt mai avantajoase, prin comparație cu cele atomice (și mecanismul de normalizare expus în acest capitol)? Principalul avantaj ține, după cum am discutat, de enorma lor flexibilitate. Dintre limite sau, după caz, dezavantaje, ar merita început cu structura ierarhică a ATR, ca și în lumea orientării pe obiecte, în general. Nu întotdeauna realitatea este ierarhică. Relațiile dintre obiecte pot fi cu mult mai complexe, iar de cele mai multe ori sunt necesare regrupări, resistematisări ale datelor, situații în care stângăciile structurilor ierarhice se manifestă în toată splendoarea lor.

Să luăm discuție relația STUDENȚI\_EXAMENE din figura 2.3 pentru care încercăm să valorificăm ATR-urile. Prima variantă - vezi figura 3.24 - folosește un atribut de tip relație numite *Examen\_ATR* ce conține informații despre fiecare toate examenele susținute de un student. Relația STUDENȚI\_EXAMENE\_ATR1 va conține doar câte un tuplu pentru fiecare student.

#### STUDENȚI\_EXAMENE\_ATR1

Matricol	NumePrenume	An	Specializare
EL13455	Popovici I Vasile	3	Informatică economică
EL13456	Zăineanu W Ion	3	Informatică economică
EL13457	Abălașei R Zicu	3	Informatică economică

<sup>23</sup> Vezi [Abiteboul & Bidoit 84]

<sup>24</sup> [Dreizen & Chang 89]

EL13458	Șpagă M Michael	3	Informatică economică

Examene_ATR				
CodDisc	DenumireDisc	NrCredite	DataExamen	Nota
AI3501	Baze de date I	6	29/01/2004	8
AI3502	Programare vizuală și RAD	7	01/02/2004	10
CodDisc	DenumireDisc	NrCredite	DataExamen	Nota
AI3501	Baze de date I	6	29/01/2004	4
AI3502	Programare vizuală și RAD	7	01/02/2004	8
AI3501	Baze de date I	6	12/02/2004	8
CodDisc	DenumireDisc	NrCredite	DataExamen	Nota
AI3501	Baze de date I	6	29/01/2004	9
AI3502	Programare vizuală și RAD	7	01/02/2004	4
AI3502	Programare vizuală și RAD	7	15/02/2004	9
CodDisc	DenumireDisc	NrCredite	DataExamen	Nota
AI3503	Analiza sistemelor informaționale	6	04/02/2004	7

Figura 3.24. Prima variantă de folosire a ATR

În egală măsură se poate însă folosi și un ATR care să conțină toți examinații pentru o disciplină dată într-o sesiune - *StudentiNote\_ATR*. Relația s-ar prezenta după cum este sugerat în figura 3.25.

**STUDENTI\_EXAMENE\_ATR2**

CodDisc	DenumireDisc	NrCredite	DataExamen
AI3501	Baze de date I	6	29/01/2004
AI3501	Baze de date I	6	12/02/2004
AI3502	Programare vizuală și RAD	7	01/02/2004
AI3502	Programare vizuală și RAD	7	15/02/2004
AI3503	Analiza sistemelor informaționale	6	04/02/2004

StudentiNote_ATR				
Matricol	NumePrenume	An	Specializare	Nota
EL13455	Popovici I Vasile	3	Informatică economică	8
EL13456	Zăineanu W Ion	3	Informatică economică	4
EL13457	Abălașei R Zicu	3	Informatică economică	9
Matricol	NumePrenume	An	Specializare	Nota

EL13456	Zăineanu W Ion	3	Informatică economică	8
<b>Matricol</b>	<b>NumePrenume</b>	<b>An</b>	<b>Specializare</b>	<b>Nota</b>
EL13455	Popovici I Vasile	3	Informatică economică	10
EL13456	Zăineanu W Ion	3	Informatică economică	8
EL13457	Abălașei R Zicu	3	Informatică economică	4
<b>Matricol</b>	<b>NumePrenume</b>	<b>An</b>	<b>Specializare</b>	<b>Nota</b>
EL13457	Abălașei R Zicu	3	Informatică economică	9
<b>Matricol</b>	<b>NumePrenume</b>	<b>An</b>	<b>Specializare</b>	<b>Nota</b>
EL13458	Șpagă M Michael	3	Informatică economică	7

Figura 3.25. A doua variantă de folosire a ATR

Care dintre cele două variante reflectă mai bine realitatea ? Prima variantă prezintă avantajul grupării tuturor examenelor unui student sub "umbrela" tuplului care se referă la studentul respectiv. Ar fi relativ simplu de calculat media studentului, de aflat la ce examene a picat o singură dată, sau de două (trei...) ori etc. Dar dacă ne interesează studenții care au luat la *Programare orientată pe obiecte* aceeași notă ca și Șpagă Michael, atunci interogarea s-ar complica destul. A doua variantă este, ca structură, mai aproape de realitate, deoarece notele sunt preluate de pe cataloage care se întocmesc la fiecare examen. Avantajul obținerii lejere a catalogului de la examen este "compensat" de dificultatea calculării mediilor pentru un student sau formație de studiu, pentru comparații între situațiile școlare ale studenților etc.

Din acest punct de vedere, renunțarea la ATR și lucrul cu attribute atomice se materializează într-o structură, să-i spunem, mai neutră, care, deși nu atât de intuitivă precum cea ierarhică, se pretează mult mai bine la prelucrări dintre cele mai diverse.

După discuția din acest paragraf, în care domeniile pot fi definite cât se poate de flexibil, în funcție de nevoile aplicației, ne putem pune întrebarea: ce rost mai are să discutăm despre aducerea relații în 1NF, operațiune care, uneori, presupune creșterea numărului de tupluri de câteva ori, astfel încât în loc să micșorăm redundanța, mai degrabă o mărim, cel puțin aparent ?

Două ar fi argumentele în sprijinul celor prezentate în acest capitol până în paragraful în care ne aflăm (oricum este ultimul !). Mai întâi, în procesul normalizării, acesta e doar primul pas. Redundanța pe care o introducem rezolvă o problemă extrem de importantă, cea a pierderilor de informații. După cum o să vedem în capitolele următoare, este aproape sigur că în celelalte forme normalizate vom scăpa de aproape tot ce este redundant într-o relație.

Al doilea argument ține de instrumentele software de care dispunem. La acest moment suportul SGBD-urilor pentru domenii de tip set, ca să nu mai vorbim de attribute de tip relație, este mai degrabă unul modest. Chiar dacă produsele importante au facilități importante în definirea de obiecte, când se pune problema mecanismului de declarare a integrității obiectelor, și mai ales a celui de interogare, lucrurile se acutizează. Chiar dacă teoretic opțiunile sunt generoase, atunci când se



---

pune problema punerii în operă a unei aplicații de lucru cu baze de date nu putem eluda "meandrele concretului", cu atât mai puțin "sinergia faptelor".