

Capitolul 2. La început a fost normalizarea

Motto:

Se poate elabora un întreg capitol sau chiar carte despre normalizare, însă este greu de scris ceva ușor și plăcut de citit (Rober J. Muller¹)

O carte dedicată proiectării bazelor de date ce începe abrupt cu normalizarea și se termină aproape la fel poate naște discuții îndreptățite. După cum am văzut și în primul capitol, multe lucrări conturează la început una sau mai multe metodologii de proiectare a bazelor de date, iar normalizarea este inclusă undeva spre miez, ca etapă, distinctă sau nu, pentru obținerea schemei finale a bazei de date². Pe de altă parte, normalizarea este cât se poate de tehnică, ba chiar științifică pe alocuri, ceea ce poate îndepărta cititorii mai puțin răbdători de la lectura următoarelor capitole.

De ce, totuși, ne încapățânăm să încalcăm o rețetă de (relativ) succes și consacrăm primele capitole dependențelor dintre atribute și formelor normalizate ale unei baze de date relaționale ? Mai întâi, din rațiuni istorico-sentimentale. Edgar F. Codd, Dumnezeu să-l odihnească - s-a prăpădit în primăvara lui 2003 -, a fundamentat modelul relațional de organizare a datelor împreună cu primele trei forme normale³. Continuată prin munca multor specialiști, normalizarea a fost prima teorie încheată de elaborare a schemei unei baze de date.

Este drept că astăzi sunt mai la îndemână modele precum E-R (entități-relații sau entități-asociații), mai lizibile, expresive și ușor de „mănuit”, însă acestea apar mai târziu. Cel la care facem trimitere, modelul *entități-relații*, a fost elaborat de Peter P. Chen pe la mijlocul anilor '70, iar altele pe parcursul anilor '80 și '90.

În al doilea rând, dacă modelele ce-și au sorgintea în E-R, OO, ORM etc. au creat un veritabil turn babel al formalismelor și notațiilor, fiecare variantă având deopotrivă suporteri și detractori la fel de înverșunați, normalizarea are o imagine de standard, metodologie unică, riguroasă (este drept, cam „scorțoasă” pe alocuri). Vom vedea, însă, nu peste mult timp, că și în privința normalizării lucrurile nu sunt definitive, ba, mai mult, excesul de normalizare strică de multe ori schema bazei de date.

Nu în ultimul rând, pentru aplicațiile relativ simple (chiar medii) și stabile în timp, normalizarea rămâne un mijloc la îndemână de a „croi” schema unei baze de date. Dacă lucrurile se complică, însă, este necesar să se recurgă la alte metodologii de proiectare.

¹ [Muller99], p.321

² [Harrington02], [Fleming & von Halle 89], [Riordan99], [Teorey99], [Hernandez97], [O'Neil & O'Neil 01], [Elmasri & Navathe00]

³ După cum preciza și Chris Date [Date86], p.364, normalizarea nu este considerată parte a modelului relațional, ci o teorie separată construită "deasupra" modelului

Cum normalizarea este legată ombilical de modelul relațional, primul paragraf în vom cheltui cu parcurgerea sumară a noțiunilor fundamentale ale relaționalului.

2.1. Concepte ale modelului relațional de organizare a bazelor de date

Descrierea unui model de date se bazează, în principal, pe trei elemente: *structura* - modul în care sunt organizate efectiv datele în bază, *integritatea* - mecanismul de asigurare a corectitudinii datelor, și *manipularea* - în sensul modificării și extragerii informațiilor din bază. După cum spuneam și în capitolul anterior, pentru această lucrare ne interesează primele două aspecte, deși, vrând-nevrând, atunci când ne vom pune problema implementării tuturor restricțiilor bazei în oricare dintre SGBD-uri, vom recurge copios la opțiunile manipulative ale modelului.

Modelul relațional s-a conturat în două articole publicate în 1969 și 1970 de către E.F. Codd, matematician la centrul de cercetări din San Jose (California) al firmei IBM. În acel moment, tehnologia bazelor de date era centrată pe modelele ierarhic și rețea, modele ce depind într-o mai mare măsură de organizarea internă a datelor pe suportul de stocare. Codd a propus o structură de date tabelară, independentă de tipul de echipamente și software de sistem pe care este implementată, structură "înzestrată" cu o serie de operatori pentru extragerea datelor. Deși puternic matematizat, modelul relațional este relativ ușor de înțeles, fiind definit printr-o serie de structuri de date (relații alcătuite din tupluri), operații aplicate asupra structurilor de date (selecție, proiecție, joncțiune etc.) și reguli de integritate care să asigure consistența datelor (chei primare, restricții referențiale s.a.).

Față de modelele ierarhice și rețea, modelul relațional prezintă câteva avantaje:

- propune structuri de date ușor de utilizat;
- ameliorează independența logică și fizică;
- pune la dispoziția utilizatorilor limbaje ne-procedurale;
- optimizează accesul la date;
- îmbunătățește integritatea și confidențialitatea datelor;
- ia în calcul o largă varietate de aplicații;
- abordează metodologic definirea structurii bazei de date.

Dacă, teoretic, modelul s-a consacrat în anii '70, produsele software care să gestioneze baze de date relaționale au devenit populare abia în deceniul 8. Deși IBM a fost prima care a inițial un proiect destinat elaborării unui SGBD relațional (System/R, începând cu 1974), prima firmă care a lansat primul SGBDR comercial a fost Relational Software Inc., astăzi Oracle. Inființarea firmei a avut loc în 1977, iar lansarea produsului în 1979. Impunerea pe piață a SGBD-urilor grefate pe modelul relațional a fost mult mai dificilă decât s-ar putea înțelege astăzi. Abia în deceniul 9, datorită vitezei ameliorate, securității sporite și mai ales productivității dezvoltatorilor de aplicații, modelul relaționat a câștigat supremația.

Există o largă tipologie a SGBDR-urilor, așa încât orice categorisire își are riscul său. Cele mai accesibile și, implicit, cele mai utilizate sunt SGBD-urile dedicate inițial uzului individual: Access, Paradox, Visual FoxPro. Astăzi, multe dintre acestea au caracteristici profesionale și pot fi folosite la dezvoltarea aplicațiilor pentru simulant a zeci de utilizatori. Pentru aplicațiile complexe din bănci, corporații, organizații și instituții de mari dimensiuni s-au impus SGBDR-urile de "categoria grea": Oracle, DB2 (IBM), Informix (IBM), Sybase (Sybase), SQL Server (Microsoft). Acestea sunt mult mai robuste, mult mai fiabile, dar și costisitoare.

În ultimul timp și-au făcut apariția, ca alternative la sus-amintiții coloși, așa zisele SGBD-uri (aproape) gratuite, precum PostgreSQL, MySQL, mSQL, Firebird etc. Acestea rulează, de obicei, pe sisteme de operare de tip Linux (foarte ieftine) și se întrevăd ca adversari serioși ai marilor producători.

2.1.1. Relații/tabele, domenii, attribute, valori nule

La modul simplist, o bază de date relațională (BDR) poate fi definită ca un ansamblu de relații (tabele); fiecare tabelă (sau tabel), alcătuită din linii (tupluri), are un nume unic și este stocată pe suport extern (de obicei disc). La intersecția unei linii cu o coloană se găsește o valoare atomică. O relație conține informații omogene legate de anumite entități, procese, fenomene: CĂRȚI, ELEVI, LOCALITĂȚI, PERSONAL, FACTURI etc. Spre exemplu, în figura 2.1 este reprezentată tabela STUDENȚI ce stochează informații privitoare la studenții înscriși la cursurile unei facultăți.

Diagram illustrating the structure of the STUDENȚI table:

- relație/tabelă**: Points to the entire table structure.
- atribute/coloane**: Points to the column headers.
- tuplu**: Points to a single row of data.
- domenii**: Points to the domain values (e.g., 1, 2, 3, 4) in the Anstudiu, Modul_spec, and Seriec columns.

Matricol	Numepren	Cnp	Adresa	Anstudiu	Modul_spec	Seriec	Grupa	Tip_bursa
M1	Topiceanu R Sorina	2222222222222	Str. Restantei nr. 13 bis, Mangalia, jud. Constant	1	2	4	2	NULL
M2	Cojocaru W Simona	2222222222223	Str. Promovarii nr. 10, Iasi, jud. Iasi	1	2	4	2	M
M3	Zet T Marian	1222222222222	Str. Durerii nr. 145 bis, Focsani, jud. Vrancea	1	2	4	2	S1
M4	Vasilescu H Ioana	2222222222222	Str. Independentei nr. 4, Mangalia, jud. Constanta	1	2	4	2	S1
M5	Strat I Costel	1222222222223	NULL	1	2	4	2	NULL
M6	Ionescu Y Vasile	1222222222224	Bd. Pacii, Iasi, jud. Iasi	3	1	1	1	NULL
M7	Calancea J Neculai	1222222222225	Str. Unirii, Vaslui, jud. Vaslui	3	1	1	1	S2
M8	Manolache R Iolanda	2222222222226	NULL	3	1	1	2	S1
M9	Vascauteanu G Ion	1222222222228	Bd. Ion Voda cel Cumplit, Suceava, jud. Suceava	1	2	4	2	S2
M10	Vlase G Georgel	1222222222229	Str. Macroeconomiei nr. 123, Iasi, jud. Iasi	1	1	1	1	S2
M11	Topiceanu T Ionela	2222222222210	Bd. Garii nr. 17, Roman, jud. Neamt	1	1	1	1	NULL

Figura 2.1. Relația (tabela) STUDENȚI

În teoria relațională se folosește termenul *relație*. Practica, însă, a consacrat termenul *tabelă* (engl. table). Un *tuplu* sau o *linie* este o succesiune de valori de diferite tipuri. În general, o linie regroupează informații referitoare la un obiect, eveniment etc., altfel spus, informații referitoare la o entitate: o carte (un titlu sau un exemplar din depozitul unei biblioteci, depinde de circumstanțe), un/o student(ă), o localitate (oraș sau comună), un angajat al firmei, o factură emisă etc. În figură este pus în evidență șaselea tuplu din tabela STUDENȚI, tuplu referitor la studentul Ionescu Y Vasile. Linia respectivă este alcătuită din nouă valori ce desemnează: matricolul studentului; numele, inițiala tatălui și prenumele; codul numeric personal (CNP-ul), adresa (stabilă); anul de studiu; modulul (dacă e în anul 1 sau 2) sau specializarea (dacă e în anii 3, 4 sau 5) la care este înscris studentul; seria de curs; grupa; tipul de bursă de care beneficiază.

Teoretic, orice tuplu reprezintă o relație între clase de valori (în cazul nostru, între nouă clase de valori); de aici provine sintagma *baze de date relaționale*, în sensul matematic al relației, de asociere a două sau mai multe elemente. Firește, toate tuplurile relației au același format (structură). Ordinea tuplurilor nu prezintă importanță din punctul de vedere al conținutului informațional al tabeli.

Fiecare atribut este caracterizat printr-un nume și un domeniu de valori pe care le poate lua. *Domeniul poate fi definit ca ansamblul valorilor acceptate (autorizate) pentru un element component al relației:*

- într-o tabelă destinată datelor generale ale angajaților, pentru atributul *Sex*, domeniul este alcătuit din două valori: *Femeiesc* și *Bărbătesc*;
- domeniul atributului *Județ* este alcătuit din numele fiecărui județ (plus București).
- domeniul unui atribut precum *PrețUnitar*, care se referă la prețul la care a fost vândut un produs/serviciu, este cu mult mai larg, fiind alcătuit din orice valoare cuprinsă între 1 și 9999999999 lei (având în vedere rata inflației).

Reținem corespondența noțiunilor *relație-tabelă*, *tuplu-linie* și *atribut-coloană*. Numărul de tabele pe care le conține o bază de date, atributele “adunate” în fiecare tabelă, domeniul fiecăruia dintre atribute prezintă diferențe majore de la o bază la alta, chiar dacă uneori reflectă același tip de procese. Intrăm astfel în sfera proiectării bazelor de date, a dependențelor și normalizării.

În figura 2.1 se observă că atributul *Adresa* conține, pe linia (tuplul) 8, o valoare curioasă, notată NULL. Valoarea NULL este considerată o *metavaloare* și indică faptul că, în acel loc, informația este necunoscută sau inaplicabilă. În cazul nostru, studentei *Manolache R. Iolanda* nu i se cunoaște adresa. Valoarea NULL este diferită de valorile 0 sau spațiu.

În încheiere, principalele caracteristici ale unei relații sunt sistematizate după cum urmează:

- în cadrul unei baze de date, o relație prezintă un nume distinct de al celorlalte relații.
- valoarea unui atribut într-un tuplu oarecare conține a singură valoare (o valoare atomică).

- fiecare atribut are un nume distinct.
- orice valoare a unui atribut face parte din domeniul pe care a fost definit acesta.
- ordinea dispunerii atributelor în relație nu prezintă importanță.
- fiecare tuplu este distinct, adică nu pot exista două tupluri identice.
- ordinea tuplurilor nu influențează conținutul informațional al relației.

2.1.2. Restricții ale bazei de date

Principale restricții definibile în modelul relațional sunt: de domeniu, de atomicitate, de non-nulitate, de unicitate, referențială și cele utilizator.

Restricția de domeniu

După cum am văzut în paragraful anterior, un atribut este definit printr-un nume și un domeniu. Orice valoare a atributului trebuie să se încadreze în domeniul definit. Există mai multe moduri de percepție a acestei restricții. O parte din informaticieni substituie domeniul tipului atributului: numeric, șir de caractere, dată calendaristică, logic (boolean) etc. și, eventual, lungimii (numărul maxim de poziții/caractere pe care se poate "întinde" un atribut). După cum se observă, este luat în calcul numai aspectul sintactic al domeniului. Faptul că anul de studiu al unei clase poate fi una din valorile: 9, 10, 11, 12 reprezintă o restricție de comportament sau, mai simplu, o restricție definită de utilizator.

Cea de-a doua categorie privește domeniul deopotrivă sintactic și semantic. Astfel, domeniul sintactic al atributului `LitClasa` (litera) dintr-o relație precum `ELEVI_LICEU` este un caracter, obligatoriu literă, și, chiar mai restrictiv, litera e obligatoriu majusculă. Din punct de vedere semantic, `LitClasa` poate lua una din valorile: A, B, C, ... în funcție de numărul de clase dintr-un an de studiu (cinci clase de a IX-a, patru de a X-a etc.).

Majoritatea SGBD-urilor permit definirea tuturor elementelor ce caracterizează domeniul (sintactic și semantic) atributuelor prin declararea tipului și lungimii atributului și prin așa-numitele reguli de validare la nivel de câmp (field validation rule). Sunt însă și produse la care domeniul poate fi definit explicit, sintactic și semantic, dându-i-se un nume la care vor fi legate atributele în momentul creării tabelelor.

Nenulitatea

Deși inventată de însuși E.F. Codd, valoarea nulă este vehement contestată de cea mai mare parte a liniei purist-relaționale (Date, Darwen, Pascal). Nu discutăm aici legitimitatea unei asemenea meta-valori, ci doar amintim că într-o baza de date unora dintre atribute li se pot interzice valoarea NULL.

Atomicitate

Conform teoriei bazelor de date relaționale, orice atribut al unei tabele oarecare trebuie să fie atomic, în sensul imposibilității descompunerii sale în alte atribute. În

aceste condiții, se spune că baza de date se află în prima formă normală sau prima formă normalizată (1NF). În tabela STUDENȚI un atribut cu valori neatomice poate fi considerat, pentru majoritatea situațiilor, Adresa, întrucât valorile sale îngrămădesc elemente ce pot fi și individualizate: strada, număr, bloc, scara, etaj, apartament, localitate și județ.

Astăzi, atomicitatea valorii atributelor a devenit o țință predilectă a "atacurilor dușmănoase" la adresa modelului relațional, datorită imposibilității înglobării unor structuri de date mai complexe, specifice unor domenii ca: proiectare asistată de calculator, baze de date multimedia etc. Pentru detalii privind atomicitatea, vezi capitolul următor.

Unicitate

Într-o relație nu pot exista două linii identice (două linii care prezintă aceleași valori pentru toate attributele). Mai mult, majoritatea relațiilor prezintă un atribut, sau o combinație de attribute, care diferențiază cu siguranță un tuplu de toate celelalte tupluri ale relației. Cheia primară a unei relații (tabele) este un atribut sau un grup de attribute care identifică fără ambiguitate fiecare tuplu (linie) al relației (tabelei). Există trei restricții pe care trebuie să le verifice cheia primară:

- unicitate: o cheie identifică un singur tuplu (linie) al relației.
- compoziție minimală: atunci când cheia primară este compusă, nici un atribut din cheie nu poate fi eliminat fără distrugerea unicității tuplului în cadrul relației; în cazuri limită, o cheie poate fi alcătuită din toate attributele relației.
- valori non-nule: valorile atributului (sau ale ansamblului de attribute) ce desemnează cheia primară sunt întotdeauna specificate, deci ne-nule; mai mult, nici un atribut din compoziția cheii primare nu poate avea valori nule.

Cea de-a treia condiție se mai numește și *restricție a entității*.

Domeniul unui atribut care este cheie primară într-o relație este denumit domeniu primar. Dacă într-o relație există mai multe combinații de attribute care conferă unicitate tuplului, acestea sunt denumite chei candidate. O cheie candidată care nu este identificator primar este referită ca și cheie alternativă. În tabela STUDENȚI, există două chei candidat, Matricol și CNP. Pe baza unor criterii tainice (deocamdată), am ales Matricol drept cheie primară, CNP-ul devenind cheie alternativă.

Dacă cheia primară a tablei STUDENȚI este una simplă, există însă suficiente cazuri în care cheia primară este compusă din două, trei s.a.m.d. sau, la extrem, toate attributele relației. Să luăm spre analiză o relație PERSONAL care conține date generale despre angajații firmei. Fiecare tuplu al relației se referă la un angajat, attributele fiind: Nume, Prenume, DataNașterii, Vechime, SalariuTarifar.

- Atributul Nume nu poate fi cheie, deoarece chiar și într-o întreprindere de talie mijlocie, este posibil să existe doi angajați cu același nume.
- Dacă apariția a două persoane cu nume identice este posibilă, atunci apariția a două persoane cu același Prenume este probabilă.
- Nici unul din aceste attribute DataNașterii, Vechime, SalariuTarifar nu poate fi "înzestrat" cu funcțiunea de identificator.

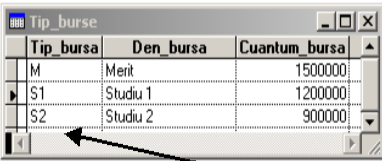
- În acest caz, se încearcă gruparea a două, trei, patru s.a.m.d. attribute, până când se obține combinația care va permite diferențierea clară a oricărei linii de toate celelalte.
- Combinația Nume+Adresă pare, la primele două vederi, a îndeplini "cerințele" de identificator. Ar fi totuși o problemă: dacă în aceeași firmă (organizație) lucrează împreună soțul și soția ? Ambii au, de obicei, același nume de familie și, tot de obicei, același domiciliu. Este adevărat, cazul ales nu este prea fericit. Dar este suficient pentru a "compromiterea" combinației.
- Următoarea tentativă este grupul Nume+Prenume+Adresă, combinație neoperantă dacă în organizație lucrează tatăl și un fiu (sau mama și o fiică) care au aceleași nume și prenume și domiciliul comun.
- Ar rămâne de ales una dintre soluțiile (Nume+Prenume+Adresă+Vechime) sau (Nume+Prenume+Adresa+DataNașterii).

Oricare din cele două combinații prezintă riscul violării restricției de entitate, deoarece este posibil ca, la preluarea unui angajat în bază, să nu i se cunoască adresa sau data nașterii, caz în care atributul respectiv ar avea valoarea NULL. Dificultățile de identificare fără ambiguitate a angajaților au determinat firmele ca, la angajare, să aloce fiecărei persoane un număr unic, număr denumit *Marcă*. Prin adăugarea acestui atribut la cele existente, pentru relația PERSONAL problema cheii primare este rezolvată mult mai simplu. Actualmente, sarcina este simplificată și prin utilizarea codului numeric personal (CNP), combinație de 13 cifre care prezintă avantajul că rămâne neschimbată pe tot parcursul vieții persoanei.

Restricția referențială

O bază de date relațională este alcătuită din relații (tabele) aflate în legătură. Stabilirea legăturii se bazează pe mecanismul cheii străine și, implicit, a restricției referențiale. Figura 2.2 prezintă o relație în care sunt implicate tabelele TIP_BURSE și STUDENȚI. Atributul *Tip_Bursa* joacă un rol de "agent de legătură" între cele două relații. Pentru tabela TIP_BURSE, atributul *Tip_Bursa* este cheie primară, în timp ce în tabela STUDENȚI, *Tip_Bursa* reprezintă o *coloană de referință* sau *cheie străină* (externă), deoarece numai pe baza valorilor sale se poate face legătura cu relația părinte TIP_BURSE. Cheile străine sau coloanele de referință sunt deci attribute sau combinații de attribute care pun în legătură linii (tupluri) din relații diferite. Tabela în care atributul de legătură este primară se numește tabelă-părinte (în cazul nostru, TIP_BURSE), iar cealaltă tabelă-copil.

Legat de noțiunea de cheie străină apare conceptul de *restricție referențială*. O restricție de integritate referențială apare atunci când o relație face referință la o altă relație. Când două tabele (relații), T1 și T2, prezintă atributul sau grupul de attribute notat CH, care, pentru T1, este cheie primară, iar pentru T2 cheie străină, dacă în T2 se interzice apariția de valori nenule ale CH care nu există în nici un tuplu din T1, se spune că între T2 și T1 s-a instituit o restricție referențială.



The image shows two database tables. The top table, 'Tip_bursa', has columns 'Tip_bursa', 'Den_bursa', and 'Cuantum_bursa'. The bottom table, 'Studenti', has columns 'Maticol', 'Numepren', 'Cnp', 'Adresa', 'Anstudiu', 'Modul_spec', 'Seriec', 'Grupa', and 'Tip_bursa'. An arrow points from the 'Tip_bursa' column in the 'Studenti' table to the 'Tip_bursa' table above it, indicating a foreign key relationship.

Tip_bursa	Den_bursa	Cuantum_bursa
M	Merit	1500000
S1	Studiu 1	1200000
S2	Studiu 2	900000

Maticol	Numepren	Cnp	Adresa	Anstudiu	Modul_spec	Seriec	Grupa	Tip_bursa
M1	Topiceanu R Sorina	222222222222	Str. Restantei nr. 13 bis, Mangalia, jud. Constanta	1	2	4	2	NULL
M2	Cojocaru W Simona	222222222223	Str. Promovarii nr. 10, Iasi, jud. Iasi	1	2	4	2	M
M3	Zet T Marian	122222222222	Str. Durerii nr. 145 bis, Focsani, jud. Vrancea	1	2	4	2	S1
M4	Vasilescu H Ioana	222222222222	Str. Independentei nr. 4, Mangalia, jud. Constanta	1	2	4	2	S1
M5	Strat I Costel	122222222223	NULL	1	2	4	2	NULL
M6	Ionescu Y Vasile	122222222224	Bd. Pacii, Iasi, jud. Iasi	3	I	1	1	NULL
M7	Calancea J Neculai	122222222225	Str. Unirii, Vaslui, jud. Vaslui	3	I	1	1	S2
M8	Manolache R Iolanda	222222222226	NULL	3	C	1	2	S1
M9	Vascauteanu G Ion	122222222228	Bd. Ion Voda cel Cumpilt, Suceava, jud. Suceava	1	2	4	2	S2
M10	Vlase G Georgel	122222222229	Str. Macroeconomiei nr. 123, Iasi, jud. Iasi	1	1	1	1	S2
M11	Topiceanu T Ionela	222222222210	Bd. Garii nr. 17, Roman, jud. Neamt	1	1	1	1	NULL

Figura 2.2. Mecanismul de legare a tabelelor

Instituirea restricției referențiale între tabela TIP_BURSE (părinte) și STUDENȚI (copil) permite cunoașterea, pentru fiecare student, a denumirii bursei și a cunctumului lunar. Dacă în STUDENȚI ar exista vreo linie în care valoarea atributului Tip_Burse ar fi, spre exemplu S3, este clar că acea linie ar fi orfană (nu ar avea linie corespondentă în tabela părinte).

Observații

- Pentru mulți utilizatori și profesioniști ai bazelor de date, denumirea de "relațional" desemnează faptul că o bază de date este alcătuită din tabele puse în legătură prin intermediul cheilor străine. Aceasta este, de fapt, a doua accepțiune a termenului de BDR, prima, cea "clasică", având în vedere percepția fiecărei linii dintr-o tabelă ca o relație între clase de valori.
- Majoritatea SGBD-urilor prezintă mecanisme de declararea și gestionare automată a restricțiilor referențiale, prin actualizări în cascadă și interzicerea valorilor care ar încălca aceste restricții.
- Respectarea restricțiilor referențiale este una din cele mai complicate sarcini pentru dezvoltatorii de aplicații ce utilizează baze de date. Din acest punct de vedere, tentația este a "sparge" baza de date în cât mai puține tabele cu puțință, altfel spus, de a avea relații cât mai "corpolente". Gradul de fragmentare al bazei ține de normalizarea bazei de date, care, ca parte a procesului de proiectare a BD, se bazează pe dependențele funcționale, multivaloare și de joncțiune între atribute.

Restricții-utilizator

Restricțiile utilizator mai sunt denumite și restricții de comportament sau restricții ale organizației. De obicei, aceste restricții iau forma unor reguli de validare la nivel de atribut, la nivel de linie/tabelă sau a unor reguli implementate prin declanșatoare (triggere). Spre exemplu, se poate institui o regulă care interzice emiterea unei noi facturi (o nouă vânzare) dacă datoriile firmei client sunt mai mari de 2 000 000 000 lei, iar directorul acesteia nu este membru în partidul/partidele de guvernământ.

2.2.Nevoia de normalizare

Unul dintre cele mai bune argumente în favoarea normalizării ține de punerea în evidență a ceea ce s-ar întâmpla în absența sa. Să luăm un prim exemplu, cel din figura 2.3 care este dedicat stocării datelor privitoare la studenții *Facultății de Economie și Administrarea Afacerilor* (sau oricare alta), mai ales în ceea ce privește situația școlară a acestora. Fiecare student are un identificator unic - *Matricol*, este înscris la o specializare într-un an de studii și susține examenele la disciplinele din planul de învățământ. Orice disciplină are alocat un număr de credite prin planul de învățământ al specializării. Fiecare student poate susține un examen de mai multe ori până îl promovează (sau este exmatriculat). Astfel, Zăineanu Ion a picat examenul la *Baze de date I* în prima sesiune (pe 29 ianuarie 2004), dar l-a luat în restanțe (pe 12 februarie).

STUDENȚI_EXAMENE

Matricol	NumePrenume	An	Specializare	CodDisc
EL13455	Popovici I Vasile	3	Informatică economică	AI3501
EL13456	Zăineanu W Ion	3	Informatică economică	AI3501
EL13457	Abălășei R Zicu	3	Informatică economică	AI3501
EL13455	Popovici I Vasile	3	Informatică economică	AI3502
EL13456	Zăineanu W Ion	3	Informatică economică	AI3502
EL13457	Abălășei R Zicu	3	Informatică economică	AI3502
EL13456	Zăineanu W Ion	3	Informatică economică	AI3501
EL13457	Abălășei R Zicu	3	Informatică economică	AI3502
EL13458	Șpagă M Michael	3	Informatică economică	AI3503

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

Figura 2.3. Relație supusă normalizării

Deoarece fiecare linie se referă la un examen susținut la o anumită dată de un student, cheia primară a relației este combinația (*Matricol*, *CodDisc*, *DataExamen*).

2.2.1. Redundanțe

Este lesne de observat că relația de mai sus conține redundanțe. Astfel, dacă un student susține, pe parcursul primelor n semestre de studii, 25 de examene, atunci matricolul, numele, anul și specializarea sunt prezente în toate cele 25 de linii. Dacă pentru o disciplină s-au consemnat în baza de date 1500 de examinări, atunci nu numai codul, ci și denumirea și numărul de credite alocat disciplinei apar de același număr de ori.

Cu cât baza de date este mai mare, cu atât risipita de spațiu este mai importantă. Chiar dacă spațiul de stocare nu mai este o problemă din punctul de vedere al costului, obezitatea unei tabele precum cea de mai sus poate atrage serioase probleme de viteză în exploatare (timpi de așteptare la preluarea noilor note etc.).

2.2.2. Anomalii la inserare

Să luăm în discuție studenții din anul I. După examenul de admitere, ce are loc în lunile iulie și septembrie aceștia sunt înmatriculați. Dacă, însă, baza de date are schema relației de mai sus, preluarea este imposibilă până în momentul primului examen al studentului respectiv. Aceasta deoarece în cheia primară sunt incluse și atributele *CodDisc* și *DataEx*, iar modelul relațional interzice valori nule pentru atributele-cheie (restricția de entitate). Ori, la data înmatriculării, studenții din anul I nu au nici un examen susținut (asta ar mai trebui !), iar prima sesiune e abia în ianuarie viitor, așa că și *CodDisc* și *DataEx* sunt în acel moment NULL.

2.2.3. Anomalii la modificare

Presupunem că într-o furtunoasă ședință de catedră se decide ca disciplina *Programare vizuală și RAD* să fie redenumită *Programare I*, titlatură sub care vă apărea și în foile matricole ale studenților din actualul an III care vor absolvi, mai devreme sau mai târziu, specializarea *Informatică economică*. În baza de date există câteva sute de înregistrări referitoare la studenți examinați la această disciplină, și toate trebuie modificate. Dacă, dintr-un motiv sau altul, modificarea se face pe numai o parte dintre liniile cu pricina, putem afirma că datele își pierd consistența. După O'Neil & O'Neil, avem de-a face cu o anomalie de modificare într-o relație atunci când modificarea valorii unui atribut atrage obligativitatea actualizării aceleași valori pe mai multe linii⁴.

⁴ [O'Neil & O'Neil 01] p.357

2.2.4. Anomalii la ștergere

Anomaliile la ștergere se manifestă atunci când prin eliminarea unei linii dintr-o tabelă se pierde involuntar nu numai informațiile despre entitatea reflectată pe linia respectivă, ci și alte informații. Astfel, dacă ștergem ultima linie din relația de mai sus, cea care conține nota examenului la *Analiza sistemelor informaționale* susținut de studentul Șpagă Michael pe 4 februarie 2004, se pierde nu numai datele despre examenul respectiv, ci și toate informațiile despre studentul Șpagă M. Aceasta deoarece aceasta era singura linie în care apărea studentul cu pricina.

2.3. Normalizarea - câteva tușe

Văzută ca rigoare (matematică sau nu), artă (în sensul de curat... interpretabilă, dar și de bazată pe intuiție, creatoare), "meserie" (deprinsă prin imitație și desăvârșită prin experiență), ceea ce se poate spune cu siguranță despre normalizare este, vorba lui C. J. Date, că nu reprezintă un panaceu.

Proiectarea bazelor de date relaționale (BDR) presupune definirea atributelor, gruparea lor în tabele, stabilirea legăturilor dintre ele, a mecanismelor de integritate și securitate, fiind un proces dificil în care cunoștințele teoretice, experiența, inteligența, intuiția și creativitatea proiectanților se îmbină într-un mod mai mult sau mai puțin armonios.

Structura finală a BD trebuie să asigure un echilibru între, pe de o parte, obținerea unui volum maxim de informații într-un interval de timp cât mai scurt și, pe de altă parte, eliminarea anomaliilor de stocare și actualizare, toate în condițiile unui grad apreciabil de securitate și integritate.

Elaborarea schemei unei BD poate începe (și) cu gruparea tuturor atributelor ale bazei de date într-o singură tabelă (relație universală). Dincolo de avantajul compactării, lucrul cu o singură (și gospodărească) relație atotcuprinzătoare ridică serioase probleme privind redundanța datelor și generează anomalii importante la adăugarea, modificarea sau ștergerea unor linii (tupluri), după cum văzut în frugalul exemplu din paragraful 2.1.

Ca și consecință directă, este necesară spargerea bazei în mai multe tabele care sunt legate prin restricții de integritate referențială. Este tocmai obiectivul central al normalizării. Spargerea relației nu trebuie făcută oricum, deoarece apare riscul pierderilor de informații. De asemenea, un număr prea mare de tabele necesită un efort sporit pentru controlul bazei și respectarea restricțiilor. În plus, trebuie să ținem seama de faptul că obținerea multor informații dintr-o bază de date fragmentată este posibilă prin operațiunea de joncțiune, mare consumatoare de resurse.

Într-o enumerare mai mult decât seacă (deci... științifică), principalele obiective ale normalizării sunt:

- minimizarea spațiului necesar stocării datelor;
- minimizarea riscului apariției de date inconsistente în cadrul BD;

- minimizarea anomaliilor ce pot apărea la actualizare (inserarea datelor, dar mai ales modificare și ștergere).;
- ameliorarea structurii bazei, reprezentarea diverselor conexiuni dintre atributele bazei.
- diminuarea nevoii de reorganizare periodică a modelului.

Obiect al numeroase studii, nu se poate afirma că există o unanimitate de idei și instrumente privind normalizarea. Importanța normalizării nu trebuie absolutizată, deoarece adesea aceasta vine uneori în contradicție cu parametri extrem de importanți ai aplicațiilor de lucru cu bazele de date: viteză de acces, securitate mărită, resurse hard/soft disponibile, reducerea traficului pe rețea etc. Se poate spune că o BD riguros normalizată nu este neapărat una optimă.

Teoria clasică a normalizării este construită în jurul a cinci forme care în literatura noastră sunt referite ca *normale* [Lungu s.a.95] sau *normalizate* [Oprea99]. Deși termenul original este *normal form* (și nu *normalized form*), mai românește sună *formă normalizată*. Nepunând prea mult preț pe academisme, voi folosi în cele ce urmează ambele sintagme, fără resentimente. Pentru a fi într-o anumită formă normală/normalizată o bază de date trebuie să respecte un anumit set de restricții.

Codd, părintele modelului relațional, a definit inițial trei forme normale, notate prin 1NF, 2NF și 3NF. Întrucât, într-o primă formulare, definiția 3FN ridica ceva probleme, Codd și Boyce au elaborat o nouă variantă, cunoscută sub numele *Boyce-Codd Normal Form* (BCNF). Deși este vorba, în principiu, de o formulare mai riguroasă a aceleași 3FN, BCNF este prezentată separat în majoritatea lucrărilor.

Formele 4 și 5, legate de numele lui Ronald Fagin, sunt tratate mai cu reținere, ca să nu spun pudoare, în literatura consacrată analizei și proiectării bazelor de date. Ba chiar unele lucrări cu tentă mai pragmatică se opresc declarat la 3FN, pe care o consideră suficientă în elaborarea BDR.

2.4.Etapele normalizării

Fundamentul normalizării BDR îl constituie dependențele dintre atribute. Primele trei forme normale pot fi determinate pe baza dependențelor funcționale elementare (totale) și tranzitive. Forma a patra (4FN) se bazează pe dependențele multivaloare, în timp ce a cincea formă normală (5FN) pe dependențele de joncțiune. Problema este că dependențele multivaloare, și (mai ales) cele de joncțiune sunt dificil de identificat și rar întâlnite în practică.

Normalizarea BDR poate fi imaginat ca un proces prin care, pornindu-se de la relația inițială R (universală), se realizează descompunerea succesivă a acesteia în sub-relații după succesiunea din figura 2.4⁵. Relația R poate fi ulterior re-construită din cele n relații obținute în urma normalizării, prin operații de joncțiune aplicate asupra acestora.

⁵ Vezi și [Oprea99], p.343

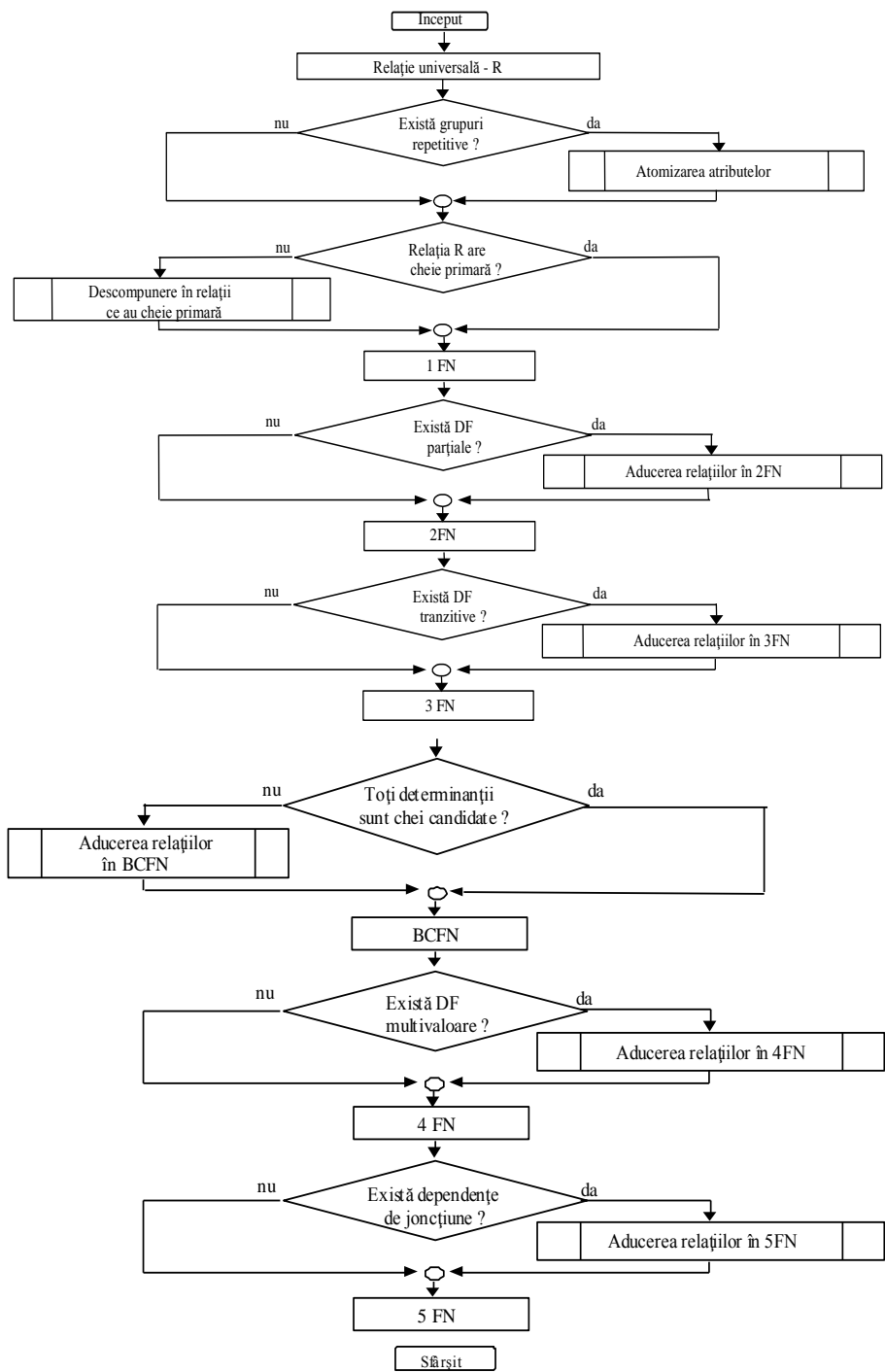


Figura 2.4. Etapele clasice ale normalizării bazelor de date relaționale

Nu putem încheia acest crochiu al normalizării fără a insista asupra faptului că, esențialmente, ingredientul acesteia este de natură semantică. Normalizarea reflectă relații care se manifestă între atribute ale bazei. Este imposibil de normalizat o relație fără a cunoaște fenomenul, procesul pe care îl reflectă.

2.5.Descompunere fără pierdere de informații

Pe baza elementelor discutate în paragraful 2.2, rezultă că prin normalizare se încearcă a se elimina redundanțe și anomalii manifestate la actualizare. Eliminarea se realizează prin "sfărâmarea" relației în două sau mai multe relații "legate" prin restricții de integritate referențială. Ulterior, prin operațiuni de joncțiune, tabela inițială poate fi recompusă, ori de câte ori va fi cazul.

Cu cât "desfacem" relația inițială (universală) în mai multe tabele, cu atât redundanța datelor este mai redusă (și, implicit, anomalii evocate). Există (și aici), însă, o limită, deoarece la descompunere trebuie avută în vedere păstrarea unor structuri care să permită, ulterior, obținerea informațiilor necesare prin operațiuni de joncțiune (apare noțiunea de *joncțiune conservativă*).

Să presupunem că, în urma depistării redundanței și anomaliilor de inserare-modificare-stergere, proiectantul bazei a ales soluția descompunerii relației STUDENȚI_EXAMENE (figura 2.3) în patru: STUDENȚI, DISCIPLINE, EXAMENE și NOTE (figura 2.5).

STUDENȚI

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
EL13458	Șpagă M Michael	3	Informatică economică

DISCIPLINE

CodDisc	DenumireDisc	NrCredite
AI3501	Baze de date I	6
AI3502	Programare vizuală și RAD	7
AI3503	Analiza sistemelor informaționale	6

EXAMENE

CodDisc	DataExamen
AI3501	29/01/2004
AI3502	01/02/2004
AI3501	12/02/2004
AI3502	15/02/2004
AI3503	04/02/2004

NOTE

Matricol	CodDisc	Nota
EL13455	AI3501	8
EL13456	AI3501	4
EL13457	AI3501	9
EL13455	AI3502	10
EL13456	AI3502	8
EL13457	AI3502	4
EL13456	AI3501	8
EL13457	AI3502	9
EL13458	AI3503	7

Figura 2.5. O descompunere a relației universale STUDENȚI_EXAMENE

Redundanțele au fost reduse simțitor. Numele și prenumele fiecărui student apar pe o singură linie; la fel și denumirea disciplinei și numărul de credite. Cu toate acestea, dacă încercăm să refacem relația inițială prin interogarea SQL:

```
SELECT s.matricol, s.numeprenume, s.an,
       s.specializare, d.coddisc, d.denumiredisc,
       d.nrcredite, e.dataexamen, n.nota
FROM studenti s INNER JOIN note n ON s.matricol=n.matricol
      INNER JOIN discipline d ON d.coddisc=n.coddisc
      INNER JOIN examene e ON d.coddisc=e.coddisc
```

s-ar obține tabela STUDENȚI_EXAMENE2 din figura 2.6.

STUDENȚI_EXAMENE2

Matricol	NumePrenume	An	Specializare	CodDisc
EL13455	Popovici I Vasile	3	Informatica economica	AI3501
EL13455	Popovici I Vasile	3	Informatica economica	AI3501
EL13456	Zaineanu W Ion	3	Informatica economica	AI3501
EL13456	Zaineanu W Ion	3	Informatica economica	AI3501
EL13457	Abalasei R Zicu	3	Informatica economica	AI3501
EL13457	Abalasei R Zicu	3	Informatica economica	AI3501
EL13455	Popovici I Vasile	3	Informatica economica	AI3502
EL13455	Popovici I Vasile	3	Informatica economica	AI3502
EL13456	Zaineanu W Ion	3	Informatica economica	AI3502
EL13456	Zaineanu W Ion	3	Informatica economica	AI3502
EL13457	Abalasei R Zicu	3	Informatica economica	AI3502
EL13457	Abalasei R Zicu	3	Informatica economica	AI3502
EL13456	Zaineanu W Ion	3	Informatica economica	AI3501
EL13456	Zaineanu W Ion	3	Informatica economica	AI3501
EL13457	Abalasei R Zicu	3	Informatica economica	AI3502
EL13457	Abalasei R Zicu	3	Informatica economica	AI3502
EL13458	Spaga M Michael	3	Informatica economica	AI3503
EL13455	Popovici I Vasile	3	Informatica economica	AI3501

	DenumireDisc	NrCredite	DataExamen	Nota
	Baze de date I	6	29.01.2004	8
	Baze de date I	6	12.02.2004	8
	Baze de date I	6	29.01.2004	4
	Baze de date I	6	12.02.2004	4
	Baze de date I	6	29.01.2004	9
	Baze de date I	6	12.02.2004	9
	Programare vizuala si RAD	7	15.02.2004	10
	Programare vizuala si RAD	7	01.02.2004	10
	Programare vizuala si RAD	7	15.02.2004	8
	Programare vizuala si RAD	7	01.02.2004	8
	Programare vizuala si RAD	7	15.02.2004	4
	Programare vizuala si RAD	7	01.02.2004	4
	Baze de date I	6	12.02.2004	8
	Baze de date I	6	29.01.2004	8
	Programare vizuala si RAD	7	01.02.2004	9
	Programare vizuala si RAD	7	15.02.2004	9
	Analiza sistemelor informationale	6	04.02.2004	7
	Baze de date I	6	29.01.2004	8

Figura 2.6. Prin recompunere se obține o altă relație universală decât cea inițială

Se observă că rezultatul diferă sensibil de tabela inițială. Exemplul este un pic forțat, dar în practică deseori apar cazuri când erorile de proiectare a bazei sunt greu de identificat. Ceea ce s-a dorit a se demonstra este faptul că normalizarea nu înseamnă descompunerea arbitrară a tabelelor în altele mai mici.

2.6. Relația universală

Prima etapă a normalizării constă în identificarea informațiilor ce trebuie stocate și gestionate cu ajutorul bazei de date. Este momentul în care la aplicațiile complexe intră în scenă analiștii de sistem, cei care observă cum se derulează procesele, operațiunile ce constituie obiectul aplicației/bazei de date, discută cu toate categoriile de utilizatori, încearcă să identifice soluții anterioare de la aceeași organizație sau de la organizații similare pentru a putea anticipa problemele cele mai dificile etc. Pentru aplicațiile mari munca analiștilor și proiectanților este extrem de laborioasă și consumatoare de timp și nervi. Este unul dintre motivele pentru care această categorie profesională are salarii mari și este destul de "curtată". Noi vom încerca să ne plasăm într-o zonă mai "aerisită", fără a cădea în derizoriu.

Noțiunea de relație universală este unul dintre destulele ingrediente ale modelului relațional care încep prin a părea jenant de banal, și din care, în final, fiecare înțelege ce vrea, fără a ajunge la un punct comun, ci doar la un „nor de puncte” de vedere comun. Noi o să considerăm relația universală drept relația alcătuită din toate atributele identificate ca fiind relevante pentru aplicație și toate tuplurile posibile cu valorile acestor atribute⁶. Prin urmare, schema relației universale conține toate atributele bazei de date.

Baza de date VÎNZĂRI

Pe parcursul acestui capitol și următoarelor ne propunem să elaborăm (și) schema unei prime baze de date prin care să poată fi gestionate informațiile relevante despre vânzările unei firme comerciale, de producție etc. Pentru o mai bună vizualizare, se poate recurge la o formă tabelară de prezentare a datelor, în care coloanele ar fi: numele atributului, așa cum este folosit în baza de date (ceva mai scurt), numele întreg al atributului, tipul atributului, lungimea, explicații suplimentare, restricții etc.

Tabelul 2.1 este ceva mai modest, având doar două coloane, dedicate numelui atributului și unor explicații sumare. Atributele care interesează în mod deosebit sunt cele legate de:

- clienți: codul, denumirea, adresa, codul poștal, localitatea, județul (inclusiv regiunea), codul fiscal, numărul de telefon;

⁶ Din acest punct de vedere, de plasăm mai aproape de accepțiunea lui Chris Date ([Date04], p.195)

- produse: codul, denumirea, unitatea de măsură, grupa (textile, încălțăminte, băuturi răcoritoare, băuturi spirtoase etc.), procentul de TVA aplicat;
- facturi: numărul, data întocmirii, clientul, observații (note despre clauze din factură), apoi produsele vândute în factura respectivă: linia pe care apare fiecare produs, cantitatea și prețul facturat.
- încasări: codul unic asociat fiecărei încasări, date despre documentul pe baza căruia se face încasarea (ordin de plată, cec etc.), numărul facturii încasate, tranșa încasată din factura respectivă.

Tabel 2.1. VÎNZĂRI - dicționarul datelor (simplificat)

Atribut	Descriere
Jud	Indicativul auto al județului (2 caractere)
Județ	Denumirea județului
Regiune	Regiunea țării din care face parte județul
CodPost	Codul poștal al unei adrese
Localitate	Denumirea orașului sau satului
Comuna	Denumirea comunei din care face parte satul curent
CodCl	Codul firmei client
DenCl	Denumirea firmei-client
CodFiscal	Codul fiscal al firmei client
StradaCl	Strada pe care se află sediul clientului
NrStradaCl	Numărul la care se afla adresa clientului
BlocScApCl	Eventualele informații suplimentare despre adresa clientului: bloc, scară, etaj, apartament
TelefonCl	Telefonul firmei client
CodPr	Codul produsului / sortimentului comercializat
DenPr	Denumirea produsului
UM	Unitatea de măsură a produsului/sortimentului
Grupa	Grupa sortimentală (Bere, Ciocolată, Cafea etc.) în care se încadrează produsul curent
ProcTVA	Procentul TVA aplicat în mod obișnuit produsului
NrFact	Numărul facturii emise
DataFact	Data întocmirii facturii
Obs	Observații privitoare la factură
Linie	Linia curentă (corespunzătoare unui produs) dintr-o factură
Cantitate	Cantitatea vândută din produs în factură (pe linia curentă)
PrețUnit	Prețul unitar de vânzare (fără TVA) al produsului în factură (pe linia curentă)
CodInc	Codul încasării
DataInc	Data încasării (în care banii intră în casierie sau în contul bancar)
CodDoc	Codul documentului de încasare
NrDoc	Numărul documentului de încasare
DataDoc	Data la care a fost întocmit documentul de încasare
Tranșă	Tranșa primită din factură prin încasarea (documentul) curentă

Una dintre cele mai importante observații ce se cuvine a fi făcute în acest moment ține de faptul că o serie de informații foarte importante, precum:

- valoarea fără TVA a unui produs facturat (de pe o linie a fiecărei facturi);
- TVA calculată pe linia unei facturi (pentru un produs vândut);
- valoarea cu TVA a unei linii (produs) dintr-o factură;
- valoarea fără TVA a unei facturi;
- TVA colectată aferentă unei facturi;
- valoarea totală (cu TVA) a unei facturi;
- valoarea totală a încasărilor pentru o factură;
- valoarea totală a vânzărilor către fiecare client;
- valoarea totală a încasărilor de la un client;
- restul de plată pentru fiecare client (diferența vânzări - încasări);
- valoarea vânzărilor pentru fiecare localitate;
- valoarea vânzărilor pentru fiecare județ;
- valoarea vânzărilor pentru fiecare regiune (Moldova, Banat etc.);

și multe altele au fost eliminate din dicționarul de date, deoarece pot fi calculate pornind ce la celelalte prezente în dicționar. Vom vedea ceva mai târziu că atunci când este vorba de date accesate în mod frecvent, pentru calcularea cărora este necesar un consum de resurse important, se poate recurge la introducerea controlată a unor atribute redundante.

O factură conține mai multe linii și poate fi încasată în una, două sau mai multe tranșe. Deoarece există un interval între momentul întocmirii de către client a documentului de plată (DataDoc) și momentul intrării efective a banilor în contul (sau casieria) firmei (DataÎnc), se folosesc ambele atribute.

Baza de date FILMOGRAFIE

Pentru un centru de închirieri video se pune problema constituirii unei baze de date cu informații despre filmele aflate pe casetele sau DVD-urile oferite (legal) spre închiriere. Fiecare film are un cod unic (identificator), iar atributele principale sunt cele din tabelul 2.2.

Tabel 2.2. FILMOGRAFIE - dicționarul datelor (simplificat)

Atribut	Descriere
IdFilm	Codul unic al filmului
TitluOriginal	Titlul în engleză, franceză etc., așa cum apare la lansarea filmului
TitluRO	Traducerea românească a titlului original
AnLans	Anul lansării
Producători	Producătorul sau producătorii filmului
Regizori	Regizorul sau regizorii filmului
Distribuție	Actorii și rolurile interpretate în film
Genuri	Genul/genurile la care se încadrează filmul (horror, comedie etc.)
Premii	Numele premiului - tipul (Oscar, Leul de argint, Ursul de aur etc.), categoria (cel mai bun film, cel mai bun actor în rol principal) anul premiului și, eventual, numele actorului.

Un film poate fi produs și regizat de una, două sau mai multe persoane. Pentru fiecare actor distribuit în film interesează și personajul sau personajele (dacă actorul e în dublu rol) interpretate. O peliculă se poate încadra la mai multe categorii (dramă, comedie) simultan. Dacă un premiu se referă la regie, scenariu etc., interesează numele, tipul și anul decernării, iar dacă se referă la o performanță actoricească, la informațiile anterioare se adaugă și numele actorului laureat.

Practic, două tupluri din această relație s-ar prezenta că în figura 2.5.

IdFilm	TitluOriginal	TitluRO	AnLans	Producători	Regizori
11899	As Good As It Gets	Mai bine nu se poate	1997	James Brooks Bridget Johnson Kristi Zea	James Brooks
12345	Bicentennial Man	Omul bicentenar	1999	Michael Barnathan Chris Columbus Gail Katz	Chris Columbus

	Distribuție	Genuri	Premii
	Jack Nicholson (Levin Udall) Helen Hunt (Carol Connelly) Greg Kinnear (Simon Bishop) Cuba Gooding Jr. (Frank Sachs)	comedie dramă romantic	Oscar - cel mai bun actor în rol principal (Jack Nicholson) - 1998 Oscar - cea mai bună actriță în rol principal (Helen Hunt) - 1998 Globul de aur - cea mai bună imagine - 1998 Globul de aur - cel mai bun actor într-o comedie/musical (Jack Nicholson) - 1998 Globul de aur - cea mai bună actriță într-o comedie/musical (Helen Hunt) - 1998
	Robin Williams (Andrew Martin) Embeth Davidtz (Little Miss Amanda Martia / Portia Charney) Sam Neil (Richard Martin) Oliver Platt (Rupert Burns) Kiersten Warren (Galatea)	SF dramă romantic	

Figura 2.5. O descompunere a relației universale FILMOGRAFIE

Aceste două relații universale, ca și altele, vor fi supuse, în capitolele următoare, normalizării și tuturor discuțiilor și problemelor legate de acest demers.

Încheiem acest capitol prin a mărturisi că, în ciuda relativei accesibilități a noțiunilor expuse vis-a-vis de relația universală, literatura de specialitate a consemnat câteva dispute pe acest subiect. Spre exemplu, în 1981 William Kent afirma că includerea tuturor coloanelor într-o singură relație universală forțează folosirea valorilor nule⁷. Așa cum vom discuta spre finalul capitolului următor, chiar dacă ne-am număra printre liber-nulliști (cei care sunt de acord cu folosirea valorilor NULL fără prea multă strângere de inimă, printre care se număra însuși E.F. Codd), de cele mai multe ori relația universală prezintă attribute componente

⁷ [Kent81]

ale cheii sale primare ce riscă să ia valori nule, ceea ce modelul relațional nu acceptă, indiferent de tabere.

În articolul său din 1981, Kent este chiar tranșant spunând că relația universală este un model nesatisfăcător pentru teoria și practica relațională⁸. Afirmatia sa a stârnit reacția a o serie de autori, printre care Jeffrey Ullman⁹, pentru care care, dincolo de inadvertențe, conceptul de relație universală este unul polimorfic, fiind suficient de amintit două dintre abordări: pe de o parte, ca un soi de creuzet, rezervor al tuturor atributelor și dependențelor dintre atribute și, pe de altă parte, ca o perspectivă generală a utilizatorilor asupra bazei de date.

Fără a-l supăra prea mult pe Ullman, vom considera relația universală drept o construcție ipotetică de pornire în demersul normalizării, fără a fi anxioși privitor la unicitatea sau neunicitatea sa, și alte proprietăți care nu au prea multă tangența cu proiectarea efectivă a unei baze de date.

⁸ [Kent81]

⁹ Vezi, spre exemplu, [Ullman82], [Ullman83]