

Tehnica normalizării relațiilor

La proiectarea structurii unei baze de date relaționale trebuie stabilite (după cum s-a văzut în cursurile anterioare) în primul rând tabelele în care vor fi memorate datele și asocierile dintre tabele. Acestea sunt stabilite într-o formă inițială, după care, prin rafinare succesivă se ajunge la forma definitivă. Acestei structuri inițiale îi sunt aplicate un set de reguli care reprezintă pașii de obținere a unei *baze de date normalizate*. Dacă o bază de date nu este normalizată ea nu poate fi utilizată cu un maxim de eficiență. Algoritmul de normalizare a bazelor de date relaționale precum și pașii acestuia au fost descriși de către E. F. Codd în 1972.

Normalizarea este procesul reversibil de transformare a unei relații în relații de structură mai simplă. (Procesul este reversibil în sensul că nici o informație nu este pierdută în timpul transformării). Scopul normalizării este de a suprima *redundanțele* logice, de a evita *anomaliile* la reactualizare și rezolvarea problemei reconexiunii.

Exemplu: Pentru a evidenția câteva exemple de redundanțe și anomalii, se va considera cazul relației inițiale OFERTANTI. Pentru a nu încărca relația, se vor considera valori ale atributelor prescurtate.

OFERTANTI:

```
cnp#numeleadresa_clientnr_  
telefonofertaadresa_imobilCnp1N1Str. Victoriei, nr.22/12, Baia  
Mare, MaramuresNr1casaA_imobil1Cnp1N1Str. Victoriei, nr. 4/5,  
Cluj-Napoca, ClujNr1halaA_imobil2Cnp2N2Str. Viilor, nr.55/4,  
Oradea, BihorNr2casaA_imobil3Cnp3N3Str. Gării, nr. 14, Bucuresti  
Nr3terenA_imobil4
```

Fig.7.1. Relația OFERTANTI

- Redundanța logică: Tripletul („N1”, “Str. Victoriei, nr.22/12, Baia Mare, Maramures”, ‘Nr1’) apare de două ori.
- Anomalii la inserare: Dacă o persoană oferă spre vânzare mai multe imobile, pentru înregistrarea ofertei trebuie rescris codul numeric personal încă o dată, deci cheia devine duplicat.
- Anomalii de ștergere: Ștergerea unei persoane din baza de date atrage după sine pierderea informațiilor despre oferta respectivă.
- Anomalii la modificare: Dacă se modifică numele străzii Victoriei din localitatea Baia Mare în strada Independenței, modificarea trebuie efectuată pentru fiecare

ofertă din Baia Mare amplasată pe strada Victoriei. Dacă ar exista 25 de oferte în această localitate pe strada Victoriei, costul modificării ar fi mare pentru a modifica toate înregistrările. Această redundanță este eliminată dacă atributul „adresa” este împărțit în alte trei atribute: „simbol_judet”, „cod_loc”, „id_strada”. Valorile acestea vor fi codul județului, localității, respectiv a străzii preluate din relațiile deja existente JUDETE, LOCALITATI, respectiv STRAZI. În acest caz, modificarea se face doar o singură dată, în tabela STRAZI.

Normalizarea

Codd a definit inițial 3 forme normale, notate prin FN_1 , FN_2 și FN_3 . Întrucât într-o primă formulare, definiția FN_3 ridică ceva probleme, Codd și Boyce au elaborat o nouă variantă, cunoscută sub numele de Boyce-Codd Normal Form (BCNF). Astfel BCNF este reprezentată separat în majoritatea lucrărilor. R. Fagin a tratat cazul FN_4 și FN_5 .

O relație este într-o *formă normală* dacă satisface o mulțime de constrângeri specificată în figura 7.2. De exemplu, se spune că o relație se află în a doua formă normală FN_2 dacă și numai dacă se află în FN_1 .

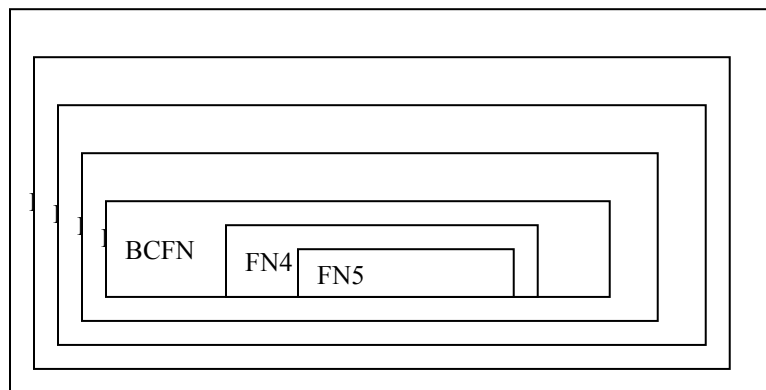


Fig.7.2. Formele normale ale relațiilor dintr-o BDR

Normalizarea bazei de date relaționale poate fi imaginată ca un proces prin care pornindu-se de la relația inițială/universală R se realizează descompunerea succesivă a acesteia în subrelații, aplicând operatorul de proiecție. Relația R poate fi ulterior reconstruită din cele n relații obținute în urma normalizării, prin operații de joncțiune.

7.1 Prima formă normală (FN_1)

FN_1 este strâns legată de noțiunea de *atomicitate* a atributelor unei relații. Astfel, aducerea unei relații în FN_1 presupune introducerea noțiunilor de:

- atribut simplu;
- atribut compus;
- grupuri repetitive de atribute.

➤ *Atributul simplu- Atribut compus*

Prin *atribut simplu (atribut atomic)* se înțelege un atribut care nu mai poate fi descompus în alte atribute, în caz contrar, atributul este *compus (atribut neatomic)*.

Exemplu: Următoarele exemple de atribute pot fi considerate simple sau compuse în funcție de circumstanțe și de obiectivele bazei de date.

- Data calendaristică – este un atribut în care apar câmpurile: zi, lună, an;
- Adresa – este un atribut în care apar câmpurile: strada, nr, bloc, scara, etaj, apartament, localitate, județ;
- Data operațiunii bancare – este un atribut în care apar câmpurile data, ora;
- Buletin/carte identitate – este un atribut în care apar câmpurile: seria, nr.

Aceste atribute pot fi atomice sau neatomice. Astfel adresa clienților agenției imobiliare interesează la nivel global, pe când pentru adresa ofertei sau a cererii de imobile este vitală prelucrarea separată a fiecărui câmp considerat.

Analog, atributul „nume” reprezintă un atribut simplu al acestei baze de date, deoarece numele clientului interesează la nivel global.

➤ *Grupuri repetitive de atribute*

Un *grup repetitiv* este un atribut (grup de atribute) dintr-o relație care apare cu valori multiple pentru o singură apariție a cheii primare a relației nenormalizate.

Exemplu: Fie relația nenormalizată (primară) FACTURI. Dorim să stabilim o structură de tabele care să permită stocarea informațiilor conținute în document (factură) și obținerea unor situații sintetice privind evidența sumelor facturate pe produse, pe clienți, pe anumite perioade de timp.

FACTURI
nr_factura#
data_factura
nume_client
adresa_client
banca_client
nr_cont_client
delegat
cod_produș
denumire_produș
unitate_de_masura
cantitate
pret_unitar
valoare
valoare_tva
total_valoare_factura
total_valoare_tva

Fig. 7.3. Relația FACTURI nenormalizată

În cazul în care o factură conține mai multe produse, relația de mai sus va avea grupurile repetitive: „cod_produș”, „denumire_produș”, „cantitate”, „pret_unitar”, „valoare”, „valoare_tva”.

Aducerea unei relații universale la FN_1

FN_1 este tratată în general cu superficialitate, deoarece principala cerință – atomicitatea valorilor – este ușor de îndeplinit (cel puțin la prima vedere).

Dintre toate formele normale, doar FN_1 are caracter de obligativitate. Se spune că o bază de date este normalizată dacă toate relațiile se află măcar în FN_1 .

O relație este în FN_1 dacă domeniile pe care sunt definite attributele relației sunt constituite numai din valori atomice. Un tuplu nu trebuie să conțină attribute sau grupuri de attribute repetitive.

Aducerea relațiilor în FN_1 presupune eliminarea atributelor compuse și a celor repetitive.

Se cunosc trei soluții pentru determinarea grupurilor repetitive:

- eliminarea grupurilor repetitive pe orizontală (în relația R inițială, în locul atributelor compuse se trec componentele acestora, ca attribute simple);
- eliminarea grupurilor repetitive prin adăugarea de tupluri;
- eliminarea grupurilor repetitive prin construirea de noi relații

Primele două metode generează relații stufoase prin duplicarea forțată a unor atribute, respectiv tupluri, creându-se astfel redundanțe masive cu multiple anomalii de actualizare.

Metoda a treia presupune eliminarea grupurilor repetitive prin construirea de noi relații, ceea ce generează o structură ce oferă cel mai mic volum de redundanță.

Etapele de aducere a unei relații în FN_1 sunt:

- I. se construiește câte o relație pentru fiecare grup repetitiv;
- II. în schema fiecărei noi relații obținute la pasul 1 se introduce și cheia primară a relației R nenormalizate;
- III. cheia primară a fiecărei noi relații va fi compusă din atributele chei ale relației R, plus unul sau mai multe atribute proprii.

Exemplu: Deoarece o factură poate avea unul sau mai multe produse înscrise pe aceasta, informațiile legate de produse vor fi separate într-o altă tabelă. Aplicând etapele de aducere la FN_1 , se obțin două relații:

FACTURI	LINII_FACTURI
nr_factura#	nr_factura#
data_factura	cod_produș#
nume_client	denumire_produș
adresa_client	unitate_de_masura
banca_client	cantiate
nr_cont_client	pret_unitar
delegat	valoare
total_valoare_factura	valoare tva
total_valoare_tva	

Fig. 7.4. Relația FACTURI adusă în forma normală FN_1

Observația1: Câmpul „adresa_client” cuprinde informații despre județul, localitatea, strada și numărul domiciliului clientului. Dacă se consideră că este de interes o evidență a sumelor factorizate pe județe sau localități, se vor pune în locul câmpului „adresa_client” trei câmpuri distincte: „judet_client”, „localitate_client”, „adresa_client”, ușurând în acest fel interogările.

Observația2: Între tabela FACTURI și tabela LINII_FACTURI există o relație de „unu la mulți”, adică unui număr unic de factură îi pot corespunde unul sau mai multe produse care sunt memorate ca înregistrări în tabele LINII_FACTURI. Cheia primară în această tabelă este o cheie compusă, formată din două câmpuri: „nr_factura” și „cod_produș”.

Însă eliminarea grupurilor repetitive, adică aducerea unei relații la FN_1 , nu rezolvă complet problema normalizării.

7.2. A doua formă normală (FN_2)

FN₂ este strâns legată de noțiunea de *dependență funcțională*. Noțiunea de dependență funcțională a fost prezentată în cursul 5: „Restricții de integritate ale modelului relațional”.

O relație se află în a doua formă normală FN2 dacă:

1. se află în forma normală FN1 și
2. fiecare atribut care nu este cheie este dependent de întreaga cheie primară.

Etapele de aducere a unei relații de la FN₁ la FN₂ sunt:

- I. Se identifică posibila cheie primară a relației aflate în FN1;
- II. Se identifică toate dependențele dintre atributele relației, cu excepția acelor în care sursa este un atribut component al cheii primare;
- III. Se identifică toate dependențele care au ca sursă un atribut sau subansamblu de atribute din cheia primară;
- IV. Pentru fiecare atribut (sau subansamblu) al cheii de la pasul III se creează o relație care va avea cheia primară atributul (subansamblul) respectiv, iar celelalte atribute vor fi cele care apar ca destinație în dependențele de la etapa III.

Exemplu: Relația care conține date redundante (de exemplu, modificarea denumirii unui produs atrage după sine modificarea în fiecare tuplu în care apare acest produs) este relația LINII_FACTURI. Se observă ca nu există nici o dependență funcțională între atributele necomponente ale cheii. În schimb, toate atributele care nu intră în alcătuirea cheii compuse sunt dependente de aceasta, iar DF dintre atributul component al cheii primare sunt: cod_produc --> denumire_produc, cod_produc --> unitate_de_masura. Ca urmare se formează încă două relații.

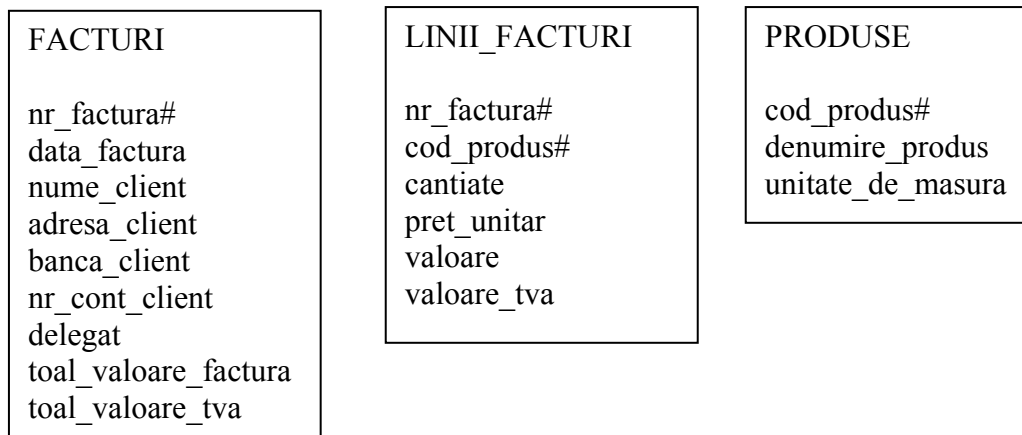


Fig. 7.5. Relația FACTURI în a doua formă normală FN2

Chiar dacă au fost eliminate o parte din redundanțe, mai rămân și alte redundanțe ce se vor elimina aplicând alte forme normale.

A treia formă normală

7.3. A treia formă normală (FN₃)

O relație este în *forma normală trei FN3* dacă:

1. se găsește în FN₂ și
2. fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară.

A treia regulă de normalizare cere ca toate câmpurile din tabele să fie independente între ele.

Etapele de aducere a unei relații de la FN₂ la FN₃ sunt:

- I. Se identifică toate attributele ce nu fac parte din cheia primară și sunt surse ale unor dependențe funcționale;
- II. Pentru aceste attribute, se construiește câte o relație în care cheia primară va fi atributul respectiv, iar celelalte attribute, destinațiile din DF considerate;
- III. Din relația de la care s-a pornit se elimină attributele destinație din DF identificată la pasul I, păstrându-se attributele surse.

Exemplu: În relația FACTURI se observă că atributul „nume_client” determină în mod unic attributele „adresa_client”, „banca_client” și „nr_cont_client”. Deci pentru atributul „nume_client” se construiește o relație CLIENTI în care cheia primară va fi acest atribut, iar celelalte attribute vor fi „adresa_client”, „banca_client” și „nr_cont_client”. Câmpurile „valoare” și „valoare_tva” depind de câmpurile „cantitate”, „pret_unitar”, și de un procent fix de TVA. Fiind câmpuri ce se pot calcula în orice moment ele vor fi eliminate din tabelă LINII FACTURI deoarece constituie informație memorată redundant.

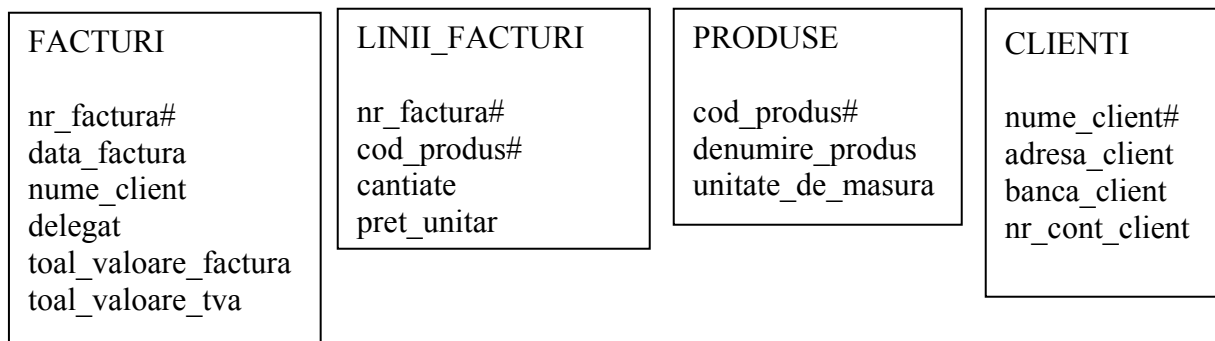


Fig. 8.1. Relația FACTURI în a treia formă normală FN₃

Observația 1: Această a treia formă normală mai poate suferi o serie de rafinări pentru a putea obține o structură performantă de tabele ale bazei de date. De exemplu se observă că „nume_client” este un câmp în care este înscris un text destul de lung format dintr-o succesiune de litere, semne speciale (punct, virgulă, cratimă), spații, numere. Ordonarea și regăsirea informațiilor după astfel de

câmpuri este lentă și mai greoaie decât după câmpuri numerice. Din acest motiv se poate introduce un nou atribut „cod_client” care să fie numeric și care să fie cheia primară de identificare a pentru fiecare client.

Observația 2: O altă observație care poate fi făcută în legătură cu tabelele aflate în cea de a treia formă normală este aceea că „total_valoare_factura” este un câmp care ar trebui să conțină informații sintetice obținute prin însumarea valorii tuturor ofertelor aflate pe o factură. Este de preferat ca astfel de câmpuri să fie calculate în rapoarte sau interogări și să nu fie memorate în tabelele bazei de date.

FACTURI nr_factura# data_factura delegat	LINII_FACTURI nr_factura# cod_produș# cantitate pret_unitar	PRODUSE cod_produș# denumire_produș unitate_de_masura	CLIENTI cod_client# nume_client adresa_client banca_client nr_cont_client
--	---	---	---

Verificarea aplicării corecte a procesului de normalizare se realizează astfel încât uniunea acestora să producă relația inițială, cu alte cuvinte, descompunerea este fără pierderi.

Celelalte forme normale se întâlnesc mai rar în practică. Aceste forme nu sunt respectate, în general, pentru că beneficiile de eficiență pe care le aduc nu compensează costul și munca de care este nevoie pentru a le respecta