

Forma normală 4 (FN4)

prof. dr. ing. Mircea Petrescu

Pentru a înțelege FN4, fie o relație cu 2 DMV. Relația este DPC, de la Departament=Dept, Proiecte, Componente=Comp. Iată o reprezentare a relației DPC în formă nenormalizată (prima dată când folosim o astfel de „notație” pentru o relație nenormalizată):

DPC

Dept	Proiecte	Comp
D1	Pr1	C1
	Pr2	C2
D2	Pr3	C2
	Pr4	C4
	Pr5	
D3	Pr2	C5
		C6

Iar prin
normalizare:

DPC

Dept	Proiecte	Comp
D1	Pr1	C1
D1	Pr1	C2
D1	Pr2	C1
D1	Pr2	C2
D2	Pr3	C2
D2	Pr3	C4
D2	Pr4	C2
D2	Pr4	C4
D2	Pr5	C2
D2	Pr5	C4
D3	Pr2	C5
D3	Pr2	C6

Din forma nenormalizată se vede ușor că avem: Dept→Proiecte, Dept→Componente.

Se vede că așa cum este concepută „aplicația”, proiectele nu determină componentele (cum ar părea natural, chiar dacă nu obligatoriu), deci nu avem DF Proiect→Comp. Departamentele mențin un stoc de componente și ele dezvoltă proiectele care pot folosi unele din componentele disponibile – sau toate.

Relația DPC are cheia formată din toate atributele sale, deci este în FNBC. Relația DPC prezintă următoarele anomalii:

1. *Anomalii de inserare.* Dacă un departament, de exemplu D2, adaugă o componentă nouă C7 în stocul său, aceasta va face necesară adăugarea mai multor tupluri – în cazul nostru trei, câte un tuplu pentru fiecare proiect al departamentului, pentru aceeași componentă C7.
2. *Anomalii de eliminare* = reprezintă contrariul anomaliilor de inserare – prin faptul că atunci când o componentă este eliminată, trebuie eliminate, de asemenea, toate tuplurile care conțin acea componentă pentru fiecare proiect al departamentelor.
3. *Anomalii de actualizare.* Schimbările (modificările) care intervin într-un proiect sau pentru o componentă (de pildă, schimbarea numărului), vor face necesare mai multe actualizări (modificări), într-un număr de tupluri egal cu numărul de valori ale celuilalt atribut independent. De exemplu, dacă numele proiectului Pr2 devine Pr9, vor trebui actualizate toate tuplurile care conțin valori pentru atributul Comp, pentru departamentul corespunzător. În exemplul de mai sus, vor trebui schimbate tuplurile al treilea, al patrulea și ultimele două tupluri ale relației.

Așadar, anomaliile de mai sus determină o „explozie” a numărului de tupluri. Putem evita această comportare prin descompunerea relației DPC în: DP(Dept, Proiecte) și DC (Dept, Comp).

DP

Dept	Proiecte
D1	Pr1
D1	Pr2
D2	Pr3

DC

Dept	Comp
D1	C1
D1	C2
D2	C2

D2	Pr4
D2	Pr5
D3	Pr2

D2	C4
D3	C5
D3	C6

În acest mod, am convertit schema de relație din FNBC în FN4. Pe baza componentelor de mai sus, putem defini FN4 a unei scheme de relație ca fiind o schemă de relație în FNBC, în care nu pot exista două sau mai multe DMV care nu sunt totodată și DF. Sau, altfel spus, partea stângă („determinantul”) a fiecărei DMV netriviiale trebuie să conducă la o schemă de relație separată. Cu alte cuvinte, în cazul exemplului nostru, în care aveam: Dept→Proiecte, Dept→Componente, am luat fiecare pereche A→B și am construit o schemă de relație AB, cu condiția ca A→B să fie netriviială.

Forma Normală 5 (FN5)

Vom (modifica) schimba conținutul relației DPC normalizate folosite în prezentarea FN4, astfel încât să nu mai prezinte DMV. Numele noii relații obținute o să fie DPRC.

DPRC

Dept	Proiecte	Comp
D1	Pr1	C1
D1	Pr1	C2
D1	Pr2	C2
D2	Pr3	C2
D2	Pr4	C2
D2	Pr4	C4
D2	Pr5	C4
D3	Pr2	C5

Schema de relație DPRC este în FNBC, deoarece cheia sa conține toate atributele. De asemenea, DPRC este în FN4, deoarece nu conține DMV.

Pe relația de mai sus, vom efectua acum operația de proiecție și obținem relațiile DPR=(Dept, Proiecte), DC=(Dept, Comp) și PRC=(Proiecte,Comp):

DPR

Dept	Proiecte
D1	Pr1
D1	Pr2
D2	Pr3
D2	Pr4
D2	Pr5
D3	Pr2

DC

Dept	Comp
D1	C1
D1	C2
D2	C2
D2	C4
D3	C5

PRC

Proiecte	Comp
Pr1	C1
Pr1	C2
Pr2	C2
Pr3	C2
Pr4	C2
Pr4	C4
Pr5	C4
Pr2	C5

Examinând relațiile obținute, observăm că nu există posibilitatea de a reconstrui DPRC prin operațiuni de joncțiune, pe niciuna din perechile de relații de mai sus.

Prin operațiuni de joncțiune, vor rezulta joncțiuni cu pierderi. Reprezentăm mai jos numai tuplurile „nelegate” ale acestor joncțiuni:

$DPR \bowtie DC$

D1	Pr2	C1
D2	Pr3	C4

$DC \bowtie PRC$

D1	Pr3	C2
D1	Pr4	C2
D2	Pr5	C2
D2	Pr2	C2

$DPR \bowtie PRC$

D3	Pr2	C2
D2	Pr1	C2

Niciunul din tuplurile de mai sus nu există în relația DPRC. Pentru a reconstrui DPRC fără pierderi, trebuie să efectuăm joncțiunea tuturor celor trei scheme: $DPR \bowtie DC \bowtie PRC$.

Faptul că o relație poate fi reconstruită fără pierderi din unele proiecții ale sale este cunoscut ca „dependență joncțională”. Dacă joncțiunile includ și relația, avem de a face cu o dependență joncțională trivială.

Preferăm o schemă cu DPR, DC și PRC în loc de DPRC, deoarece relațiile descompuse au cardinalul inferior celui al relației DPRC (sau egal cu acesta). Aceasta înseamnă că DPRC va conduce la anomalii de actualizare datorită redundanței sale, deoarece DPRC, deși este în FNBC și în FN4, nu este în FN5 sau în Forma Normală Proiecție Joncțiune (FNPJ).

O relație este în FN5 dacă:

- dependența joncțională care reconstruiește schema originală este o dependență joncțională trivială, sau
- fiecare relație în dependența joncțională constituie (cu toate atributele sale) o supercheie a relației originale.

Deoarece avem dependența joncțională $DPR \triangleright \triangleleft DC \triangleright \triangleleft PRC$, DPRC nu este în FN5 (niciuna din aceste relații nu formează o supercheie pentru DPRC). Prin urmare, schema descompusă cu DPR, DC și PRC este în FN5, întrucât aceste relații satisfac oricare din condiții (de fapt, ambele condiții) și întrucât fiecare relație reprezintă o dependență joncțională pentru ea însăși, iar atributele sale sunt superchei (ne amintim că în acest exemplu toate cele trei relații au cheile formate din toate atributele lor).

Forma Normală Domeniu-Cheie (FNDC)

FN5 privește dependențele relativ obscure. Încă de studiat! Acum:

Fiecare din formele normale prezentate până acum au fost introduse treptat de cercetători pentru a elimina anomalii pe care le găseau în formele normale inferioare (în FN5 – obscure!). De aceea, a apărut ideea de a găsi o formă normală în care să nu apară anomalii de nici un fel. Așa „s-a născut” FNDC în 1981 (Faglia).

Definiții:

- a) *constrângere* – în sens foarte general: orice legătură asupra valorilor statice ale atributelor, suficient de precisă, pentru a putea afirma că este adevărată sau falsă. Exemple: DF, DMV, restricții interne relației (de exemplu, privind forma unor atribute – vom vedea un exemplu).
- b) *cheie* – identificator unic al unui tuplu;
- c) *domeniu* – o descriere a valorilor admise pentru un atribut; are două părți: o descriere fizică și o descriere semantică, sau logică. Descrierea fizică = mulțimea valorilor pe care le poate avea atributul. Descrierea semantică – se referă la înțelesul atributului.

Definiția neformală a FNDC: o schemă de relație este în FNDC dacă prin introducerea (satisfacerea) restricțiilor de cheie și de domeniu, este determinată satisfacerea tuturor celorlalte constrângeri.

În practica conceperii, a implementării și exploatării bazei de date, prin „fortarea” restricțiilor de cheie și de domeniu, nu apar anomalii și relațiile sunt în FNDC.

FNDC – este o problemă practică. Nu sunt cunoscuți algoritmi pentru construirea unei scheme în FNDC – este mai mult o „artă” decât o știință.

Exemplu:

Relația – Student(Ids, Anul, Cămin, Chirie)

Cheia – Ids (Identificator student).

Constrângeri: Cămin→Chirie, Ids nu trebuie să înceapă cu digitul 1.

Dacă putem exprima contrângerile de mai sus ca fiind o consecință logică a definițiilor privind cheia și domeniile, atunci conform unei teoreme a lui Fagin putem fi siguri că nu vor fi (nu vor apărea) anomalii de modificare (de actualizare).

Pentru a forța constrângerea că Ids nu începe cu 1, vom defini – pur și simplu – domeniul pentru Ids astfel încât să țină seama de această condiție:

Aici – definiția domeniului pentru Ids: CDDD, unde C este un digit zecimal $\neq 1$, iar D este digit zecimal.

Apoi, trebuie să facem astfel încât constrângerea Cămin \rightarrow Chirie să fie o consecință logică a cheilor.

Așadar: dacă atributul Cămin ar fi el însuși o cheie, atunci Cămin \rightarrow Chirie ar fi o consecință logică a unei chei. Dar, în relația student, atributul Cămin nu poate fi cheie, deoarece în aceeași clădire a căminului locuiesc mai mulți studenți. Ca urmare, o soluție ar putea fi ca atributul Cămin să fie o cheie a propriei sale relații. Atunci, definim relația ca schema Cămin_Chirie, în care attributele sunt Cămin și Chirie, iar atributul Cămin este cheie.

Prin introducerea noii relații, putem elimina atributul Chirie din relația Student. Mai jos avem definițiile finale pentru domenii și relații.

Definiții de domeniu:

Ids = CDDD, unde C = digit zecimal $\neq 1$, D = digit zecimal

Anul ('I', 'II', 'III', 'IV')

Cămin: CHAR(4)

Chirie: DEC(5)

Definiții de scheme de relație și chei:

Student (Ids, Anul, Cămin), cheie: Ids

Cămin_Chirie (Cămin, Chirie), cheie: Cămin

Schemele de relație sunt în FNDC, potrivit definiției formulate anterior (Fagin).