

Introducere

După a doua carte publicată la Polirom (Visual FoxPro...), un cititor se declara amuzat de frământările mele și ale colegilor în a găsi un titlu adecvat fiecărei cărți, titlu care să atragă posibilul cumpărător și să semnalizeze cât mai limpede ceea ce se ascunde între copertile cărții. Amuzamentul său, de altminteri și al altor cititori care ne-au scris ulterior, ținea de faptul că *frământările* care, pe alocuri, degenerau în *frăsuiele*, au eșuat în niște titluri cu totul convenționale, atât pentru SQL, cât și pentru VFP. Mă tem că aceasta este soarta și a prezentului volum, din care vor lipsi, însă, cel puțin din introducere, *frăsuiele*.

Teritoriul proiectării bazelor de date, revendicat de analiști/proiectanți, dar „călcat”, cu sau fără nonșalanță, și de dezvoltatorii de aplicații, administratorii de baze de date sau alte categorii de informaticieni, este unul în care talentul „urechist” (în sensul de „cântăreț/interpret-care-nu-are-nevoie-de-partitură”), a jucat multe feste în istoria recentă a proiectelor informaționale de la noi și de aiurea.

Motivul principal al masivei doze de improvizație și eșecuri în proiectarea bazelor de date ține de însăși natura demersului. Fără a exagera grosolan, putem afirma că, în construirea schemei unei baze de date, rigoarea este pozitivă doar până la un punct. Intuiția, inteligența (nici aceasta în doze prea mari, deoarece strică !), dar mai ales experiența, sunt calități de neînlocuit ale unui „gânditor” într-ale bazelor de date. Prin comparație cu administrarea și cu scrierea de programe și proceduri ce alcătuiesc „logica bazei de date”, proiectarea este zona cea mai puțin rigidă, cea mai inexactă.

Al doilea motiv este, în mare măsură, o consecință a primului: lipsă cronică de cărți și monografii despre cum se gândește schema unei baze de date. Practic, chiar pe plan mondial, cărțile cu adevărat valoroase dedicate proiectării bazelor de date pot fi numărate pe degete (fără a le folosi pe toate !). Și aici aş furniza o explicație simplistă. Academistii, adică universitarii precum autorul de față, au răbdare să adune sistematic, și mai ales să compileze din alte lucrări (fără a le indica neapărat la bibliografie), chestiuni ce sunt relevante și/sau la modă, însă, de obicei, evită cu dibăcie să se implice în proiecte practice, aplicative. Senzația pe care am încercat-o citind majoritatea cărților dedicate bazelor de date este că autorii nu au lucrat la nici proiect de anvergură în domeniu. Pe de altă parte, cei care s-au desăvârșit în proiectarea bazelor de date (incomparabil mai puțini decât cei ce cred acest lucru) câștigă, de obicei, destul de binișor și-și cultivă alte plăceri decât cea a scrisului, cu atât mai mult cu cât cronofagia proiectelor informaționale afectează iremediabil condiția intelectuală, cărpită disperat cu o lectură din Patapievici și o alta din Cărtărescu¹.

¹ Părerea mea !

După cum, probabil, cei mai atenți dintre dvs. și-au dat seama, am adoptat tactica lui Ion Iliescu aplicată la Revoluție și după. Am început prin a descrie situația ca pe una disperată, aproape fără ieșire, tocmai pentru a demonstra cât de importanți suntem noi în salvarea „situațiunii”. Așa că (urmează o tuse scurtă, în doi timpi, a la Cristian Grețcu (din Divertis)), *concețățeni și concețene*, iată ce urmează.

Mai întâi, vom cheltui câteva pagini cu importanța bazelor de date pentru sistemele informaționale ale organizațiilor și locul „dânselor” în aplicațiile software. Metodologia clasică de dezvoltare a aplicațiilor este completată, în paragrafele 1.4 și 1.5, de un adevărat *lamentatio* al proiectării bazelor de date, care este destinat a lămuri ceea ce individualizează cartea de față de celelalte lucrări consacrate subiectului.

În al doilea capitol vom introduce noțiunile pe care se bazează modelul relațional și apoi vom face o schiță, în tușe grosolane, a normalizării, pornind de la nevoia de normalizare și terminând cu etapele procesului.

Din capitolul trei discuția se animă un pic, deoarece intrăm în câteva zone scandaloase ale normalizării: valori atomice, grupuri repetitive, relații universale și prima formă normală (1NF). Lucrurile sunt cu atât mai interesante cu cât chiar în tabăra „marilor preoți” ai relaționalului sunt câteva subiecte de gâlceavă, iar o parte din definiții au fost reformulate de-a lungul ultimilor ani.

Discuția animată din capitolul trei devine de-a dreptul plicticoasă în cel de-al patrulea, deoarece vor fi tratate dependențele funcționale, împreună cu proprietățile acestora, precum și modalitatea de aducere a relațiilor în a doua formă normalizată (2NF). Din păcate, ideea de a sări peste acest capitol nu-i tocmai bună, deoarece întreg procesul de normalizare de bazează pe dependențele dintre atribute.

Capitolul cinci acoperă un obiectiv intermediar în construirea schemei unei baze de date, obiectiv intermediar care, pentru unii teoreticieni și destui practicieni, este chiar final: a treia formă normalizată (3NF). Lucrurile nu se opresc aici, deoarece 3NF este completată cu discutarea formei normale Boyce-Codd (BCNF) și a câteva dintre situațiile în care normalizarea își arată limitele, sau este de-a dreptul neputincioasă.

Capitolul șase încearcă să facă mai atractivă, nu fără o anumită doză de zădărnici, probabil cea mai deprimantă zonă a normalizării, cea referitoare la dependențele multi-valoare și de joncțiune și, implicit, la formele normale 4 și 5 (4NF și 5NF). Iar dacă pentru 5NF practica nu oferă prea multă satisfacție dornicilor de exemple, problemele legate de 4NF, deși ascunse cu grijă sub preș, sunt mult mai frecvente decât ne-am dori. Pe final, sunt pomenite câteva tentative de depășire a 5FN, cu precădere DKNF, care însă nu au trecut de stadiul bunelor intenții.

În capitolul șapte intrăm deja în „sinergia faptelor”. Discuțiile coboară nepermis de la finețuri și abstracțiuni teoretice, la probleme mundane, cum ar fi cheile surogat, restricțiile complexe, atributele redundante și dependențele de incluziune. De fapt, dacă ne place să ne jucăm cu cuvintele, putem spune că în capitolul 3 recepționăm baza de dată *nenormalizată*, din capitolul 3 în capitolul 6 am tot *normalizat-o*, iar din acest capitol intrăm în *post-normalizarea* bazei de date care, în

capitolul nouă, va căpăta pe alocuri forma *denormalizării*. Cazul practic din final este gândit pentru a răzbuna cantitatea de teorie îngurgitată în precedentele capitole. Tot ca element de noutate, din acest capitol apar mostre de cod scrise aproape exclusiv în Oracle PL/SQL. Listingurile și procedurile sunt gândite ca exemple de implementare a cheștiunilor teoretice prezentate.

Capitolul opt stă sub semnul exoticii, cel puțin pentru o lucrare dedicată normalizării și post-normalizării. Deși considerate cvasi-monopol al orientării pe obiecte, generalizarea și specializarea pot fi abordate parțial, dintr-un unghi propriu, și în modelul relațional, iar scurta și incompleta discuție despre (un soi de) grafuri ne duce cu gândul la modelul rețea, astăzi pe drept uitat de cei mai tineri dintre noi. Finalul capitolului, dedicat reprezentării simultane a incluziunilor și echivalențelor într-o schemă, nu ne duce cu gândul la nimic.

Dintr-o carte în care normalizarea este la așa mare preț, nu putea lipsi un capitol dedicat *denormalizării*, denormalizare ce poate fi privită și ca o încălcare parțială (și ușor jenată) a aproape tot ceea ce am discutat în capitolele anterioare. Folosită deseori abuziv, denormalizarea poate fi asimilată mai degrabă post-normalizării, deci nu atât o încălcare a normalizării, cât o completare a sa.

Una dintre cele mai dure probe ale unei baze de date este validitatea în timp. Tocmai de aceea capitolul 10 tratează, pe scurt, câteva aspecte ale temporalității în proiectarea schemei unei baze de date. Ne interesează, deci, să găsim acele structuri de date care să furnizeze toate informațiile importante ale aplicației, indiferent de momentul producerii lor, dar și mecanismele de protecție a datelor sensibile împotriva alterării lor ulterioare.

Prin ce s-ar diferenția de concurență cartea de față ? În primul rând, prin tentativa de a umaniza un domeniu considerat imposibil – normalizarea. Arma cea mai des întrebuințată este exemplul și contraexemplul. Cu riscul de a părea dezlănat, materialul de față este înțesat de cazuri practice care ilustrează aproape fiecare noțiune discutată. Un al doilea atu al lucrării ține de mostrele de cod care aduc discuțiile legate de proiectare pe teritoriul implementării bazei de date. Prin toate mecanismele și artificiiile descrise, redactate în SQL sau Oracle PL/SQL, se dorește creșterea credibilității cărții, știindu-se că un dezavantaj curent al publicațiilor dedicate proiectării ține de apetența lor exclusiv teoretică. În al treilea rând, cartea răstoarnă canoanele clasice ale proiectării unei baze de date, întrucât renunță (de bună voie și nesilită de nimeni) la modelele consacrate ale analizei și proiectării sistemelor informaționale: diagrame entități-relații, diagrame UML etc. De fapt, nu renunță la acestea, cât amână discuția pentru un volum viitor. Pur și simplu, în acest volum am vrut să epuizăm cât mai mult din ceea ce este relevant pentru practicieni din normalizare și post-normalizare.

Să publici în 2005 o carte despre proiectarea bazelor de date fără o brumă de E-R, UML, O-R, OO pare desuet, dacă nu chiar sinucigaș (publicistic). Ei bine, citiți dvs. cartea de față, și peste un doi-trei ani de întâlnim la o bârfă de UML. N-aș vrea să se înțeleagă că se poate renunța la analiză și proiectare, că dezvoltarea de

aplicații complexe ar trebui să bazeze exclusiv pe normalizare. Am vrut să arat doar că normalizarea este mult mai generoasă decât am înțeles noi de la teoreticienii săi, că este un demers bazat pe concubinajul rigorii cu ambiguitatea, că există situații pe care normalizarea le rezolvă magistral și situații în care este de-a dreptul penibilă.

Rămâne de văzut și cât anume din intenții are acoperire în ceea ce se așterne în fața dvs. În ceea ce mă privește, ceea ce urmează reprezintă nu numai fructul compilației cărților și articolelor la care am avut acces în ultimii cincisprezece ani, ci și experiența, uneori traumatizantă, a câteva zeci de proiecte și aplicații în care am fost implicat, chiar dacă, nu rareori, rezultatele au fost dezastruoase. Dintre colegii de catedră, îi mulțumesc îndeosebi lui Florin pentru răbdarea de a fi citit și criticat prima formă a manuscrisului și, totodată, îi cer scuze pentru necazul de a-i răpi din timpul afectat treburilor casnice (scuze, Florin, n-am putut să mă abțin !).

Marin Fotache

Iași, februarie 2005

Notă:

Listingurile SQL și PL/SQL din capitolele 7-10 pot fi descărcate de pe pagina web a Facultății de Economie și Administrarea Afacerilor, Universitatea Alexandru Ioan Cuza Iași: <http://www.feaa.uaic.ro/cercetare/publicatii/>