

# Capitolul 1. Obiectul proiectării bazelor de date

De la portofoliul de mega-aplicații circumscrise avalanșei acronimice în care este cu neputință să nu facem confuzii – ERP, SCM, CRM, BI, B2B, B2C etc. – până la „anticele” (în termeni strict informatici) și pernicioasele aplicații de contabilitate generală scrise în FoxPro 2.0, vorbim de sisteme (sau fragmente) informaționale, de porțiuni ale memoriei organizaționale. Putem caracteriza bazele de date ca un ingredient esențial, ce-i drept, mai puțin vizibil, al sistemului informațional al oricărei firme, organizații, instituții.

Tocmai caracterul oarecum discret, dacă nu cumva misterios, a făcut din bazele de date un domeniu interesant și atractiv pentru profesioniștii IT, dar a și generat numeroase confuzii, convulsii, bonjurisme, partizanate și idiosincrasii. Cele câteva tușe din acest capitol, dedicate bazelor de date și abordărilor privitoare la proiectarea bazelor de date nu vor tranșa nicidecum disputele, dar nici nu le vor acutiza.

## 1.1. Locul bazelor de date în sistemul informațional

Mai toate lucrările de profil includ datele, alături de echipamente (hardware), programe (software), proceduri, metodologii și personal, drept componente esențiale ale sistemelor informaționale în organizații.

Putem privi bazele de date pieziș sau cu simpatie, din diferite unghiuri, în funcție de interes, pregătire de specialitate și competență în domeniu. Înainte de a îneca subiectul în câteva definiții curat academice menite a semăna panica printre studenți, trebuie să spunem că mulți dintre cei dispuși sau siliți să fie preocupați de domeniul IT privesc bazele de date ca pe niște rezervoare imense în care sunt stocate informații dintre cele mai diverse, informații ce ulterior pot fi extrase, prelucrate, sistematizate după dorințele și nevoile diverselor tipuri de utilizatori.

Analogia cu rezervoarele funcționează până la un punct. În primul rând, rezervoarele trebuie croite corespunzător, pentru a nu pierde din caracteristicile substanțelor stocate, pentru a putea face sintetizările (distilările) ulterioare și pentru a nu fi în situația de a extinde ulterior, din mers, încăperi ale rezervoarelor deja în uz (vorbim de o schemă judicioasă de stocare). În al doilea rând, n-ar strica să existe câteva filtre care să blocheze introducerea de substanțe „cu defecte”, întrucât mai toată lumea cunoaște principiul informatic cu numele GIGO – *Garbage In, Garbage Out*, adică „Gunoi introduci, gunoi extragi!” (aici am putea vorbi de un mecanism de integritate al bazei de date). Pe urmă, introducerea substanțelor în rezervor, ca și extragerea lor, brută sau prin rafinare, necesită utilaje speciale (acesta este Sistemul de Gestiune a Bazei de Date) și angajați recomandabil

pricepuți (aici intră, de-a valma, proiectanți, administratori de baze de date, programatori, utilizatori mai răsați în ale bazelor de date etc.).

Aducând discuția pe făgașul academic, o bază de date (BD) reprezintă un ansamblu structurat de fișiere care grupează datele prelucrate în aplicațiile informatice ale unei persoane, grup de persoane, întreprinderi, instituții etc. Formal, BD poate fi definită ca *o colecție de date aflate în interdependență, împreună cu descrierea datelor și a relațiilor dintre ele sau, similar, o colecție de date utilizată într-o organizație, colecție care este automatizată, partajată, definită riguros (formalizată) și controlată la nivel central*<sup>1</sup>. Într-o abordare mai analitică, o BD este un ansamblu de date: *structurate, coerente, neredundante, independente de orice program de aplicație, direct accesibile după mai multe criterii*.

O bază de date evoluează în timp, în funcție de volumul și complexitatea proceselor, fenomenelor și operațiunilor pe care le reflectă. Ansamblul informațiilor stocate în bază la un moment dat constituie *conținutul* sau *instanțierea* sau *realizarea* acesteia. Organizarea bazei de date se relectă în *schema* sau *structura* sa, ce reprezintă un ansamblu de instrumente pentru descrierea datelor, a relațiilor dintre acestea, a semanticii lor și a restricțiilor la care sunt supuse. În timp ce volumul prezintă o evoluție spectaculoasă în timp, schema unei baze rămâne relativ constantă pe tot parcursul utilizării acesteia.

Cu bazele de date lucrează o generoasă tipologie de profesioniști IT și utilizatori, lucru vizibil și din clasică schemă din figura 1.1 ce ilustrează cele trei niveluri la care se poate discuta despre schema și conținutul unei baze de date: *fizic* sau *intern*, *conceptual* sau *global* și *extern*.

La nivel *fizic* (intern) structura datelor este descrisă foarte detaliat, fiind accesibilă numai specialiștilor (ingineri de sistem, programatori în limbaje de asamblare sau alte limbaje apropiate de "mașină"). Esențiale pentru acest nivel sunt fișierele stocate în memoria externă a calculatorului, fișiere ce conțin datele propriu-zise, alcătuite din articole sau înregistrări cu format comun.

Nivelul *conceptual* (sau *global*) este nivelul imediat superior celui fizic, datele fiind privite prin prisma semanticii lor; interesează conținutul lor efectiv, ca și relațiile care le leagă de alte date. Obiectivul acestui nivel îl constituie modelarea realității considerate, asigurându-se independența bazei față de orice restricție tehnologică sau echipament anume. Toți utilizatorii își exprimă nevoile de date la nivel conceptual, prezentându-le administratorului bazei de date, acesta fiind cel care are o viziune globală necesară satisfacerii tuturor cerințelor informaționale.

Nivelul *extern* este ultimul nivel de abstractizare la care poate fi descrisă o bază de date. Structurile de la nivelul conceptual sunt relativ simple, însă volumul lor poate fi deconcertant. Iar dacă la nivel conceptual baza de date este abordată în ansamblul ei, în practică, un utilizator sau un grup de utilizatori lucrează numai cu o porțiune specifică a bazei, în funcție de departamentul în care își desfășoară activitatea și de atribuțiile sale (lor). Simplificarea interacțiunii utilizatori-bază, precum și creșterea securității bazei sunt deziderate ale unui nivel superior de

---

<sup>1</sup> Pentru amatorii de definiții, vezi [Lungu s.a.95], p.14; [Băscă97], p.11; [Popescu97], pp.12-14; [Velicanu s.a.99], pp.7-8; [Fotache s.a.02-2], p.227; [Fotache01], p.12; [Fotache97], p.29

abstractizare, care este nivelul extern. Astfel, structura BD se prezintă sub diferite machete, referite, uneori și *ca sub-scheme*, scheme externe sau imagini, în funcție de nevoile fiecărui utilizator sau grup de utilizatori.

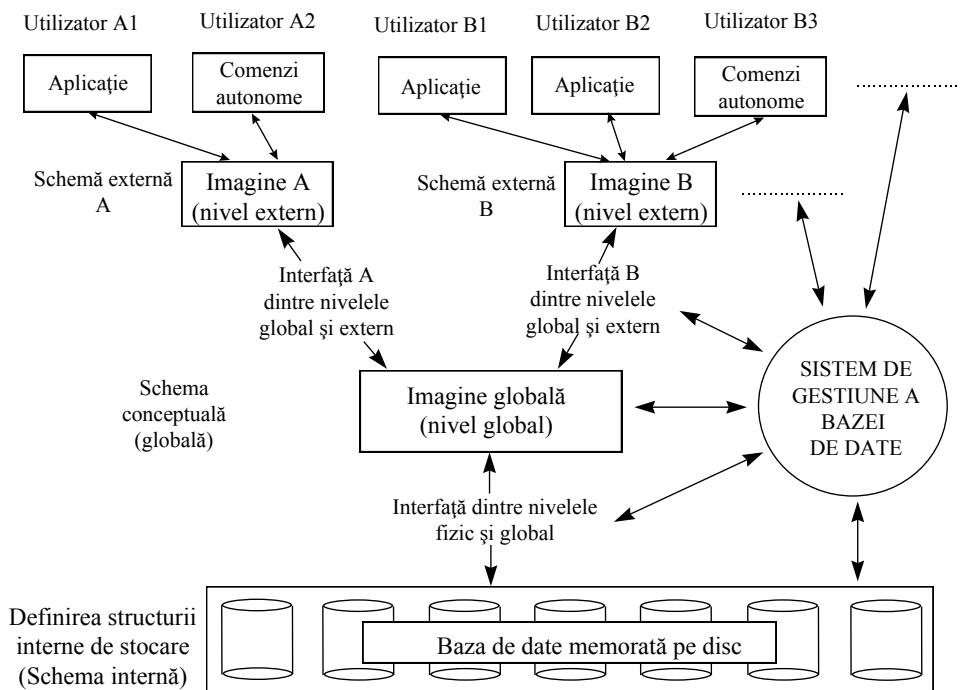


Figura 1.1. Schematizare a unui sistem de lucru cu o bază de date<sup>2</sup>

Accesul utilizatorilor la informațiile din bază este posibil numai prin intermediul componentei software reprezentate de sistemul de gestionare a bazei de date (SGBD). În general, interfața utilizator-SGBD se poate realiza în două moduri: printr-un mecanism de apel inserat în programele scrise într-un limbaj "tradițional", acesta fiind cazul SGBD-urilor cu limbaj gazdă, sau prin comenzi speciale utilizate autonom (în afara aplicațiilor-program), în cazul SGBD-urilor autonome.

## 1.2. Principalele arhitecturi ale aplicațiilor în care sunt implicate bazele de date

Una dintre caracteristicile esențiale ale bazelor de date ține de caracterul lor public, de accesibilitatea lor, e drept, diferențiată, unei largi tipologii de utilizatori. Rareori se întâmplă ca baza de date să fie pe același calculator de pe care

<sup>2</sup> Prelucrare a schemei din [Date86], p.33

utilizatorul o accesează. Lucrul acesta este valabil doar pentru aplicațiile simple, *mono-utilizator*, cum sunt referite în folclorul informatic. De obicei, însă, baza de date se află pe un calculator special sau chiar pe mai multe calculatoare, dacă vorbim de baze de date replicate sau distribuite.

Dacă sistemul pe care se află baza de date este folosit doar la stocarea acesteia, iar SGBD-ul care o exploatează este instalat pe fiecare calculator client, se poate vorbi de o arhitectură *file-server*. Inițial, rețelele de PC-uri se bazau pe descărcarea pe stația de lucru a fișierelor de pe server și efectuarea tuturor prelucrărilor local. Problema de bază a unei asemenea arhitecturi era incapacitatea deservirii simultane a unui număr mare de utilizatori.

Ca răspuns la limitările conceptului *file-server* a apărut arhitectura *client/server*, în care serverul de fișiere a fost înlocuit cu un server de baze de date pe care este instalat un SGBD care "ascultă" cererile clienților și le răspunde în mod direct. Arhitectura *client/server* se constituie ca o infrastructură versatilă, bazată pe mesaje, modulară, născută cu intenția îmbunătățirii utilizabilității, flexibilității, interoperabilității și scalabilității aplicațiilor software. Clientul este definit ca un solicitant de servicii, iar serverul este definit ca un furnizor de servicii, ținând seama de faptul că una și aceeași stație poate fi și client și server în funcție de configurația software. Clientul și serverul comunică prin schimb de mesaje<sup>3</sup>.

În general, clienții sunt calculatoare personale (PC-uri) utilizate pentru activități de gestionare a datelor. După Sinha<sup>4</sup>, un post *client* se caracterizează prin faptul că: (a) prezintă o interfață utilizator, care e, de obicei, grafică (GUI); (b) "formulează" interogări (cereri, consultări) sau comenzi pe care le "înaintează" serverului; (c) transmite interogările/comenzile respective serverului prin intermediul unei tehnologii de comunicație; (d) analizează datele din rezultatele interogărilor/comenzilor primite de la server, iar un *server* prin faptul că: (a) furnizează un serviciu clientului; (b) răspunde la interogările/comenzile clientului; (c) ascunde detaliile sistemului, făcând transparent dialogul dintre client și server.

Serverele pot fi calculatoare personale sau sisteme de calcul specializate (minicalculatoare, mainframe-uri) în vederea asigurării legăturii dintre clienți și bazele de date din care se extrag informațiile dorite. Dezideratul este autonomia informatică a fiecărui angajat, în limita unor atribuții delegate, angajat care poate astfel consulta și prelucra datele din orice loc aflat în interiorul sau în exteriorul întreprinderii. Prin accesul la toată gama serviciilor disponibile în rețea se poate reacționa mai rapid și mai eficace la evenimentele care apar și, deci, exercita mai eficient responsabilitățile încredințate.

Orice sistem *client/server* este alcătuit din minimum trei componente principale: interfața cu utilizatorul (*sistem de operare/mediu grafic*), aplicația (*prelucrările sau procesele*) și sistemul de gestiune a bazelor de date. În practică, există

---

<sup>3</sup> Vezi [Fotache s.a.02-2]

<sup>4</sup> Sinha, R. - *Client/server computing*, Communications of the ACM, iulie 1992, pp.77-98

mai multe modalități de repartizare a funcțiunilor între client și server, așa cum rezultă și din figura 1.2<sup>5</sup>.

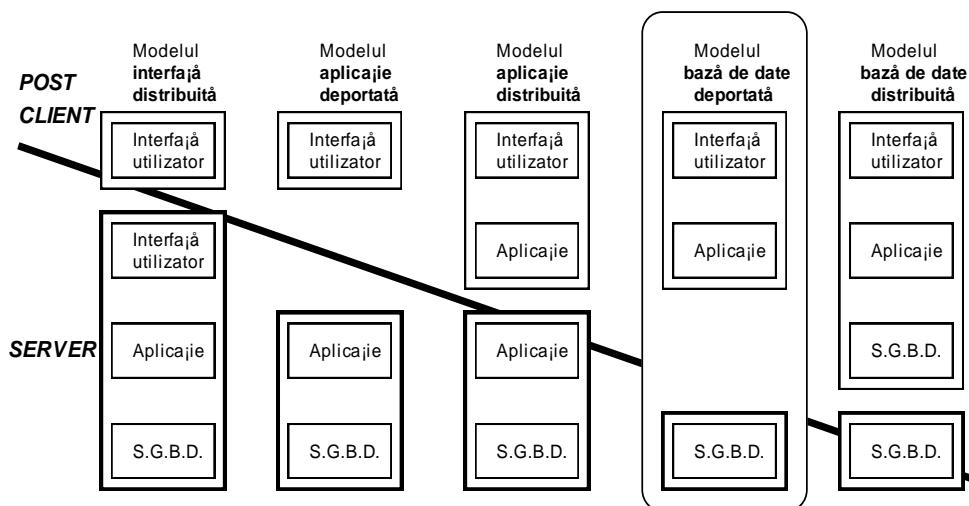


Figura 1.2. Cinci moduri de repartizare a funcțiunilor între client și server (sursa: Gartner Group)

- *Modelul interfață distribuită.* Codul aferent interfeței utilizator este împărțit între platformele client și server, iar aplicația și SGBD-ul sunt, ambele, rezidente pe server. Este modelul unei arhitecturi clasice cu un calculator central gazdă la care sunt cuplate terminale pasive (sistemele de minicalculatoare românești INDEPENDENT, CORAL, mini-sistemele PDP ale firmei DEC etc.).
- *Modelul aplicație deportată.* Codul aferent interfeței este plasat pe platforma client, în timp ce aplicația și SGBD sunt situate pe server (XWindows rulate sub Unix).
- *Modelul aplicație distribuită.* Presupune localizarea interfeței pe calculatorul-client, a SGBD-ului pe calculatorul-server, în timp ce o parte a aplicației este rezidentă pe stație și cealaltă parte pe server.
- *Modelul bază de date deportată.* Interfața și aplicația sunt rezidente pe platforma client, în timp ce SGBD-ul este plasat pe server. Este modelul arhitecturii client-server "clasice", pe logica căruia sunt realizate majoritatea sistemelor comercializate sub titulatura "client-server".
- *Modelul bază de date distribuită.* Interfața, aplicația și o primă parte a SGBD-ului se găsesc pe sistemul-client, în timp ce o a doua parte a SGBD este rezidentă pe server.

<sup>5</sup> Preluare din Harmon, P. - *Les systèmes client-serveur*, Génie logiciel et systèmes experts no. 34, mart. 1994

În prezent, asistăm la schimbarea paradigmei în dezvoltarea aplicațiilor, modelul client/server fiind considerat depășit de mulți autori, iar marile companii din industria software-ului ne urează „bun venit” în lumea aplicațiilor Web<sup>6</sup>. În esență, arhitecturile pe trei straturi, față de arhitecturile client/server pe două straturi, separă *logica afacerii* pe un strat distinct, localizat de regulă pe un server de aplicații care comunică strâns cu serverul de baze de date. Paradigma dezvoltării software-ului bazat pe componente ce comunică între ele prin interfețe standardizate a făcut posibilă creșterea numărului de straturi armonizate în arhitecturi generice numite *n-tier* (vezi figura 1.3).

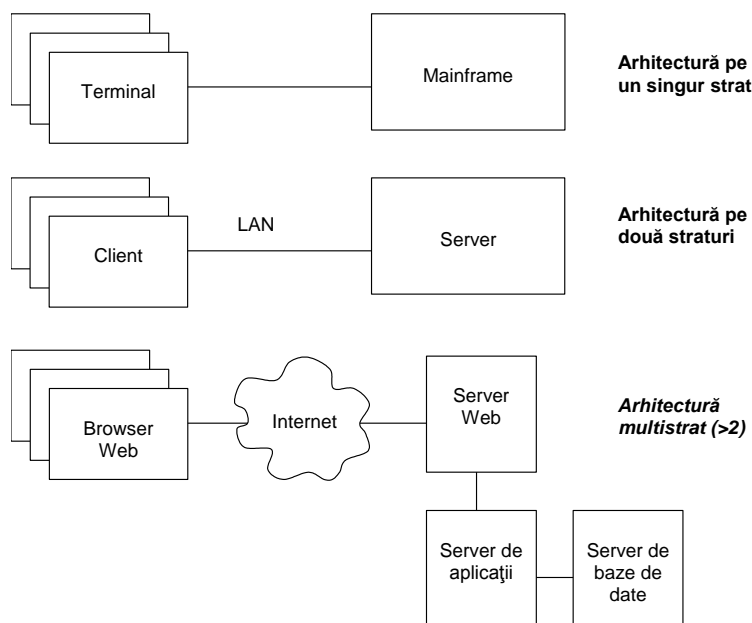


Figura 1.3. Trei tipuri arhitecturi

Tehnologiile Web permit accesarea aplicațiilor folosind browser-e Web, esențiale fiind protocoale ca HTML (pentru prezentare) sau HTTP pentru comunicare între clienții Web și server-ele Web.

### 1.3. O metodologie clasică de dezvoltare a aplicațiilor

Până acum am punctat importanța datelor în sistemul informațional al organizațiilor, nivelurile la care se poate aborda structura și conținutul unei baze de date, precum și locul „stratului” *date* în aplicațiile informatice actuale. În acest paragraf vom încerca să încadrăm activitatea (activitățile) de proiectare a bazei de date în demersul mai larg al dezvoltării de aplicații.

<sup>6</sup> Vezi [Fotache s.a.02-2], pp.252-275

De la bun început trebuie precizat că subiectul paragrafului de față constituie obiect de studiu pentru cel puțin trei domenii ale informaticii aplicate în organizații, sau, mai bine spus, sistemelor informaționale în organizații. Mai întâi, este vorba de ceea ce e în literatura anglo-saxonă se numește *software engineering*, în franceză *génie logiciel*, iar la noi *dezvoltare de aplicații software*. Este un domeniu conturat încă din anii '60 și, aflat de atunci, cel puțin după spusele marilor specialiști, într-o permanentă criză, criză ce nu dă semne de epuizare.

Al doilea domeniu este mai larg și țintește mult mai mult decât realizarea de aplicații, deși aceasta reprezintă, totuși, un obiectiv central. Este vorba de *analiza și proiectarea sistemelor informaționale*<sup>7</sup>. Analiza și proiectarea se ocupă, printre altele, și cu investigarea tuturor proceselor, operațiunilor și tranzacțiilor dintr-o firmă sau instituție, cerințelor utilizatorilor și perspectivelor organizaționale, astfel încât toate aceste aspecte să fie luate în calcul în realizarea unui sistem informațional coerent, aliniat misiunii, obiectivelor și politicilor firmei. Cu alte cuvinte, analiza și proiectarea pornesc mai din amonte, de la utilizatori, procese, tranzacții economice, încercând să formalizeze/modelizeze realitatea sub forma unei largi game de diagrame, scheme, specificații pe le vor înainta realizatorilor modulelor de interfață, prelucrări și date.

Una dintre cele mai cunoscute scheme de realizare (dezvoltare) a aplicațiilor de lucru cu baze de date, sau, altfel spus, schemă de principiu a ciclului de viață al aplicațiilor cu BD este cea din edițiile lucrării lui Thomas Connoly și Carolyn Begg<sup>8</sup> prezentată în figura 1.4<sup>9</sup>.

Amplarea activităților din ciclul de viață al BD depinde de anvergura aplicației. Când sistemul dezvoltat vizează un număr redus de utilizatori și se referă la un ansamblu de funcțiuni nu din cale-afară de complex, multe etape sunt sărite sau parcurse sumar.

### Planificarea bazei de date

Planificarea BD presupune eșalonarea pașilor ciclului de viață pentru atingerea unui maximum de eficiență și eficacitate. Ca și în cazul planificării software-ului, planificarea BD presupune identificarea și evaluarea *activităților* ce trebuie derulate (întreprinse), *resurselor* necesare derulării activităților, *fondului de timp*, *specialiștilor* și *banilor disponibili*. Planificarea BD trebuie integrată în strategia de ansamblu a firmei, unul dintre obiectivele esențiale fiind catalizarea activităților, politicilor și strategiei unității.

### Definirea sistemului

*Definirea sistemului* presupune specificarea domeniului și granițelor aplicației ce lucrează cu baza de date, utilizatorii, aplicabilitatea sa, precum și celelalte componente ale sistemului informațional la care se va conecta noul subsistem.

---

<sup>7</sup> Vezi [Oprea99], [Oprea s.a.05]

<sup>8</sup> Până la ediția a treia, apărea și un al treilea autor, A.D. Strachan - vezi [Connoly s.a.96].

<sup>9</sup> [Connoly&Begg 02], pp.271-294

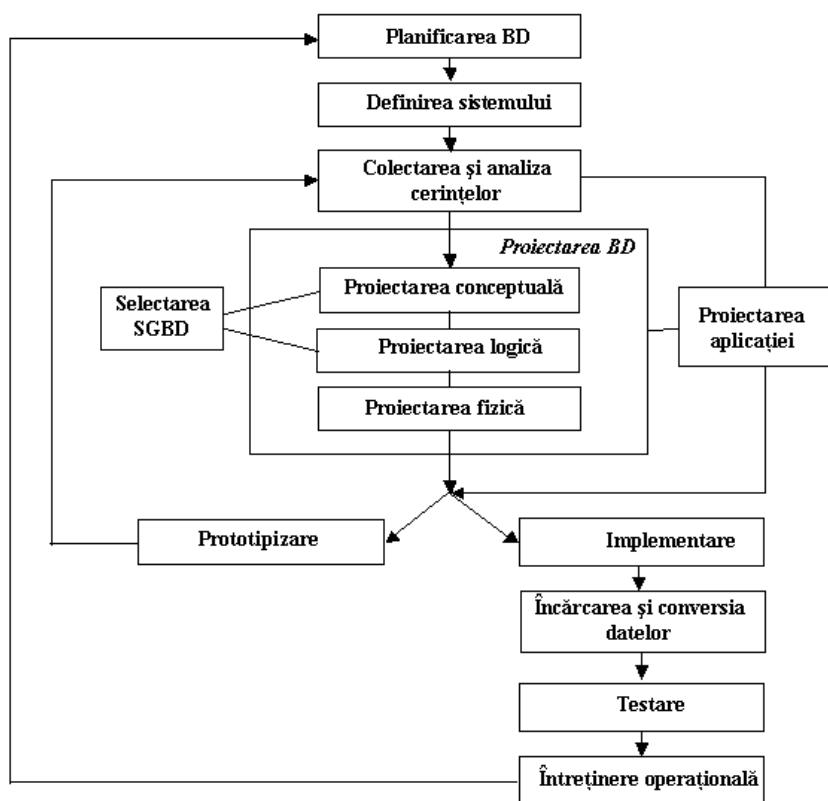


Figura 1.4. Ciclul de viață al aplicațiilor ce utilizează baze de date

### Colectarea și analiza cerințelor

Colectarea și analiza cerințelor implică adunarea și analiza cerințelor aplicațiilor din partea utilizatorilor. Cerința reprezintă o opțiune, un element ce trebuie inclus, tratat în noul sistem. Proiectarea BD se bazează pe informații despre organizații, informații ce trebuie preluate și gestionate de bază. Informațiile pot fi culese în diferite moduri:

- interviuarea personalului din întreprindere, cu precădere a celor mai apropiați, prin specificul activității lor, de profilul aplicației, avându-se în vedere acordarea unei atenții mărite experților din compartimentele funcționale considerate;
- observarea modului de derulare a operațiilor în cadrul firmei;
- examinarea documentelor, în principala a celor purtătoare de informații (primare, rapoarte);
- utilizarea chestionarelor pentru preluarea informațiilor de la un mare număr de utilizatori;
- utilizarea experienței acumulate la proiectarea unor sisteme similare.



Informația culeasă trebuie să includă principalele domenii (zone) și grupuri de utilizatori; documentația utilizată sau generată de și despre aceste domenii/grupuri de utilizatori; detaliile tranzacțiilor reclamate de fiecare domeniu/grup ; o listă de cerințe, pe priorități, a fiecărui domeniu/grup.

Rezultatul acestei activități îl reprezintă *specificațiile cerințelor organizaționale*, prezentate, de obicei, sub forma unui set de documente în care sunt descrise, din diferite unghiuri, operațiile întreprinderii. De obicei, cerințele specificate se prezintă într-o formă puțin structurată și necesită transformarea utilizând tehnici consacrate, cum ar fi cele de tip *Structured Analysis and Design - SAD* (analiză și proiectare structurată), *Data Flow Diagrams - DFD* (diagrame ale fluxurilor de date), diagrame *Hierarchical Input Process Output - HIPO* (ierarhice - intrări-prelucrări-ieșiri) etc.

### Proiectarea bazei de date

Proiectarea BD include proiectarea logică și fizică a BD. În urma acestei etape va fi elaborat modelul BD care va constitui suportul obiectivelor și operațiunilor organizației. Principalele obiective ale proiectării BD sunt:

- reprezentarea datelor și relațiilor dintre date, formulare de diferitele zone ale aplicației și grupurilor de utilizatori;
- furnizarea unui model al datelor care să permită orice tranzacție autorizată asupra datelor;
- construirea unui proiect care va atinge cerințele de performanță ale sistemului, cum ar fi: securitatea, timpul de răspuns etc.

Deseori, însă, realizarea unui obiectiv se face în detrimentul altuia. Spre exemplu, un nivel ridicat de securitate atrage după sine scăderea vitezei de răspuns.

După Connolly și Begg<sup>10</sup>, există două abordări majore ale proiectării BD. Abordarea *bottom-up* pornește de la nivelul elementar, cel al atributelor, attribute care sunt grupate în clase de entități și asociații. Normalizarea este un exemplu tipic de demers bottom-up, care dă bune rezultate atunci când numărul atributelor nu este prea mare.

Când numărul de attribute este ridicat, mai nimerită este abordarea *top-down* ce dezvoltă mai întâi un model sintetic, simplificat, al datelor. Cele câteva entități sunt decompuse ulterior în mai multe etape, până la identificarea atributelor și entităților elementare. Modelarea folosind diagrame E-R are la bază abordarea top-down.

Există și alte abordări ale proiectării BD; spre exemplu, *inside-out* (de la interior către exterior) seamnă cu bottom-up, dar diferă prin faptul că, mai întâi, se stabilește un set de concepte majore și apoi analiza se lărgțește prin includerea în model și a altor concepte care se află în relație cu cele majore. *Strategia mixtă* utilizează deopotrivă, pentru diferitele părți ale modelului, și bottom-up și top-down, combinând rezultatele.

---

<sup>10</sup> [Connolly&Begg 02], pp.279-280

După cea mai mare parte a autorilor, proiectarea bazelor de date este de două tipuri, logică și fizică<sup>11</sup>, în timp ce după alții există trei tipuri: conceptuală, logică, fizică<sup>12</sup>.

### Selectarea sistemului de gestiune a bazei de date

Considerată un pas opțional, selectarea SGBD presupune alegerea celui mai adecvat SGBD pentru realizarea și exploatarea aplicației. Nu toate SGBD-urile satisfac cerințele aplicației, iar, pe de altă parte, cele mai performante sunt și cele mai scumpe. Cerințele de funcționalitate trebuie coroborate cu resursele disponibile, în vederea identificării celui mai adecvat SGBD comercial pentru realizarea și exploatarea aplicației.

Criteriile de selecție au fost generos sistematizate de Connolly și Begg ca în tabelul 1.1<sup>13</sup>.

Tabel 1.1. Criterii de selecție a SGBD

#### Definirea datelor

Gestiunea cheilor primare  
 Specificarea cheilor străine  
 Tipuri de date disponibile  
 Extensibilitatea tipurilor de date  
 Specificarea domeniilor  
 Ușurința restructurării  
 Mecanism de tabele derivate  
 Controlul integrității  
 Dicționar de date  
 Independența datelor  
 Tipul de model de date utilizat

#### Accesibilitate

Suport pentru standardele SQL  
 Interfață cu 3GL  
 Multi-utilizator  
 Securitate  
 - controlul accesului  
 - mecanism de autorizare

#### Utilitare

Măsurarea performanței  
 Acordare (tuning)/optimizare  
 Facilități de încărcare/descărcare  
 Suport pentru administrarea BD

#### Dezvoltare

Instrumente 4GL  
 Instrumente CASE  
 Facilități vizuale

#### Definirea fizică

Structuri fizice disponibile  
 Întreținerea structurii de fișiere  
 Ușurința reorganizării  
 Indexare  
 Atribute/înregistrări de mărime variabilă  
 Compresia datelor  
 Rutine de criptare  
 Cerințe de memorie

#### Alte facilități/opțiuni

Evolutivitate  
 Stabilitatea furnizorului  
 Baza de utilizatori  
 Pregătire și suport pentru utilizatori  
 Documentare  
 Sistem de operare cerut  
 Cost  
 Help online  
 Standarde utilizate  
 Managementul versiunilor  
 Optimizare a interogărilor  
 Scalabilitate  
 Suport pentru instrumente OLAP  
 Interoperabilitate cu alte SGBD  
 Integrare Web  
 Utilitare de replicare  
 Facilități de distribuire  
 Portabilitate  
 Cerințe hardware  
 Suport pentru rețea

<sup>11</sup> Vezi, spre exemplu, lucrările: [Date04], [Teorey99], [Muller99], [Simsion01]

<sup>12</sup> Vezi, spre exemplu, [Connolly&Begg 02]

<sup>13</sup> Preluare din [Connolly&Begg 02], pp.284-287

Proceduri stocare, declanșatoare  
Instrumente de dezvoltare Web

**Controlul tranzacțiilor**

Rutine de salvare și restaurare  
Puncte de salvare intermediare  
Suport pentru jurnalizare  
Granularitatea accesului simultan  
Strategia de rezolvare a interblocajelor  
Procesare paralelă a interogărilor

Facilități orientare pe obiecte  
Arhitecturi pe două, trei... straturi  
Performanță  
Rata de tranzacții pe secundă (minut)  
Număr maxim de utilizatori simultani  
Suport XML

**Proiectarea aplicației**

Proiectarea aplicației se referă la conceperea secvențelor de cod (instrucțiuni în limbaje de programare) ce vor lucra cu baza. Din figura 1.4 se observă că proiectarea BD și proiectarea aplicației sunt activități distincte, paralele ale ciclului de viață al aplicațiilor cu BD. În cea mai mare parte a cazurilor, proiectarea aplicației nu poate fi finalizată înainte de proiectarea bazei. Pe de altă parte, informațiile proiectării vor fi transmise proiectării BD.

În această etapă trebuie verificat dacă toate funcțiunile specificate în cerințele-utilizator se regăsesc în proiectul aplicației. Proiectarea aplicației include și activitatea de *proiectare a tranzacțiilor*. De asemenea, tot acum se elaborează și modelul interfeței utilizator a aplicației.

**Prototipizarea**

Deși opțională, prototipizarea se poate dovedi deosebit de utilă prin construirea unui model de lucru inițial simplificat, model ce permite dezvoltatorilor și utilizatorilor să testeze și amelioreze incremental noua aplicație. Un mare avantaj al prototipizării este gradul mult mai mare de acceptare a noului sistem la final, acesta fiind dezvoltat împreună cu utilizatorii ce pot să-și exprime pe parcurs doleanțele. Etapele prototipizării sunt descrise în figura 1.5<sup>14</sup>.

**Implementarea**

Implementarea presupune crearea definițiilor BD la nivelele conceptual, extern și intern, precum și punerea în lucru a programelor (aplicației), fiind realizată utilizând limnajul de definire a datelor (DDL) pus la dispoziție de SGBD-ul selectat. Implementarea aplicației se realizează într-un mediu de programare utilizând un limbaj de tip 3GL sau 4GL.

**Conversia și încărcarea datelor**

Acest pas este necesar atunci când sistemul (aplicația) dezvoltat înlocuiește un altul. Datele trebuie transferate din vechea în noua aplicație, operație ce implică de multe ori schimbarea structurii fișierelor, a formatului (logic și fizic) al datelor, în concordanță cu cerințele noii aplicații, dar și ale noului SGBD. Uneori sunt necesare (și posibile) și conversii ale unor programe din vechiul în noul sistem. Majoritatea

---

<sup>14</sup> Preluare din [Connoly s.a.96]

SGBD-urilor actuale au module de import-export din/în alte SGBD-uri. Conversia aplicațiilor este, în general, mult mai complexă atunci când se schimbă limba-jul/mediul de programare.

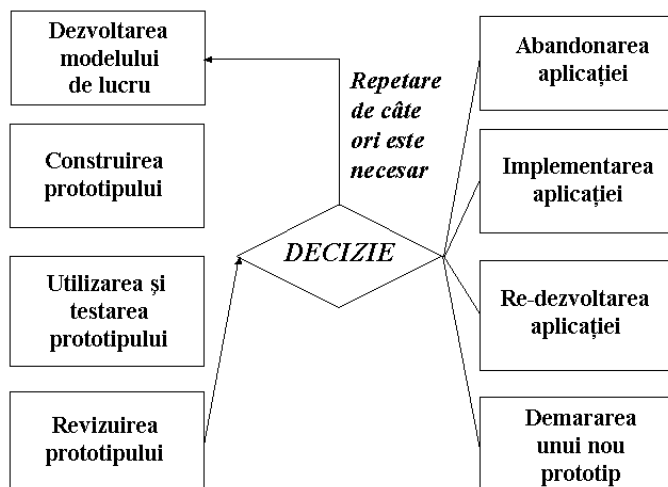


Figura 1.5. Schema de principiu a prototipizării

## Testarea

Testarea este operațiunea sistematică de verificare a funcționalității sistemului, a gradului de adecvare la cerințele utilizatorilor. Prin testare aplicația este lansată în execuție, urmărindu-se depistarea eventualelor erori și ameliorarea unor parametri, precum viteză, securitate etc. Urmând o metodologie riguroasă, măsurătorile din cadru testării dau posibilitatea evaluării calității și fiabilității software-ului.

Este de preferat ca utilizatorii aplicației să fie pe deplin implicați în procesul testării, iar aplicația testată să fie instalată pe un alt sistem; în orice caz, trebuie avut în vedere și un mecanism de recuperare a datelor corupte în caz de avarie.

Există mai multe strategii de testare a corectitudinii și funcționalității unei aplicații de lucru cu BD ce pot fi aplicate individual sau combinate în cadrul aceluiași sistem:

- top-down;
- bottom-up;
- fire de execuție;
- test de stres.

Testarea *top-down* începe de la nivelul superior, al funcțiunilor majore ale aplicației. Este utilă în verificarea arhitecturii sistemului, re-proiectările majore fiind semnalate din primele momente ale testării.

Testarea *bottom-up* începe cu modulele elementare și continuă pe verticală cu modulele compozite până la nivelul general al aplicației.

Testarea *firilor de execuție* este foarte importantă în sistemele de prelucrare în timp real, în care simultan sunt rulate o serie de procese ce cooperează. Acest tip de testare este mult mai complex, fiind legat de detaliile tehnice ale sistemului de operare. Fiecărui proces i se alocă un fir de execuție (thread) care este urmărit de la declanșarea sa, acordându-se o mare importanță punctelor de întâlnire cu celelalte fire executate pe sistem.

Testul de *stres* presupune suprasolicitarea bazei și aplicațiilor. Astfel, BD se încarcă automat cu un număr extrem de mare de înregistrări, iar la aplicație se conectează cât mai mulți utilizatori. Astfel se depistează până la ce limite rezistă aplicația.

### Întreținerea operațională

Această activitate se derulează pe toată durata de utilizare a aplicației. În afară de monitorizarea BD, a programelor, de "curățare" periodică, reparare a eventualelor erori hard/soft, tot în această activitate sunt incorporate operațiunile de actualizare a datelor și programelor (ca urmare a unor noi oportunități tehnologice, modificarea unor parametri (procente TVA, impozite etc.). Monitorizarea performanței sistemului se realizează prin raportarea la un nivel prestabilit de acceptanță. Dacă este cazul, o situație a performanței generale sub acest punct critic poate antrena reorganizarea BD. Altminteri, chiar și în cazul funcționării în parametri normali, BD trebuie optimizată, având în vedere și facilitățile oferite de SGBD.

Întreținerea și actualizarea aplicației sunt necesare în mai mare sau mai mică măsură. Spre exemplu, într-un mediu economic și legislativ dinamic, cum este cel din țara noastră, deseori este necesară modificarea unor părți importante ale aplicației. Uneori, modificarea presupune reluarea unora sau tuturor pașilor din ciclul de viață al aplicației de lucru cu BD.

## 1.4. Privire de ansamblu asupra proiectării bazelor de date

Deși au fost publicate destule cărți dedicate proiectării bazelor de date, subiectul continuă să fie departe de epuizare. Chiar dacă unii autori s-au străduit să aducă rigurozitate, uneori cu prețul unei anume rigidități, proiectarea bazelor de date rămâne, după cum spunea și Chris Date<sup>15</sup>, preponderent o artă și mai puțin o știință.

Dacă luăm în considerare cele trei niveluri de lucru cu o bază de date din celebra figură 1.1, problema proiectării ține de elaborarea unei structuri logice și fizice ale uneia sau mai multor baze de date pentru întâmpinarea nevoilor informaționale ale utilizatorilor într-o organizație pentru un set definit de aplicații<sup>16</sup>.

---

<sup>15</sup> [Date04], p.330

<sup>16</sup> [Elmasri & Navathe 00], p.532

Una dintre cele mai influente cărți de proiectare a bazelor de date este cea publicată de Candace Flemming și Barbara von Halle în 1989<sup>17</sup>. Din păcate, cele două autoare sunt și întemeietoarele unui întreg șir de inadvertențe propagate în multe lucrări până în ziua de azi. Spre exemplu, autoarele mărturisesc că materialul lor prezintă două metodologii: *modelarea logică a datelor* și *proiectarea bazelor de date relaționale*, definind modelarea logică a datelor ca procedură pentru reprezentarea cerințelor informaționale într-un format corect, consistent și stabil, iar proiectarea BD relaționale drept procedură de transformare a modelului logic al datelor într-o schemă relațională stabilă<sup>18</sup>.

Cartea de față este una din dovezile clare că modelul logic al datelor, mai precis, structura conceptuală a bazei de date poate fi concepută și strict în termenii modelului relațional. Greșeala autoarelor ține de faptul că, pentru ele, o schemă relațională alcătuită din tabele și restricțiile la care sunt supuse datele din tabele nu ar corespunde nivelului logic (conceptual) al bazei de date, ci mai degrabă nivelului fizic. După Fleming și von Halle, metodologia de *modelare logică a datelor* (MLD) se derulează în 12 pași<sup>19</sup>:

1. identificarea entităților majore;
2. determinarea relațiilor dintre entități;
3. determinarea cheilor primare și alternative;
4. determinarea cheilor străine;
5. determinarea celor mai importante reguli organizaționale (key business rules);
6. adăugarea celorlalte atribute (atributele non-cheie);
7. validarea perspectivelor utilizator folosind normalizarea;
8. determinarea domeniilor (restricțiilor asupra valorilor autorizate ale atributelor);
9. determinarea operațiunilor declanșatoare (regulile care guvernează efectele modificărilor asupra atributelor și entităților);
10. combinarea perspectivelor-utilizator;
11. integrarea cu modelele de date existente;
12. analiza stabilității și dezvoltărilor ulterioare.

Este evident că, deși autoarele separă artificial proiectarea logică a bazei de date de proiectarea bazei de date relaționale, o parte din pașii modelări logice sunt raportați explicit la modelul relațional. Perspectiva poate fi explicată prin faptul că, la nivelul anului 1989, modelul suveran în materie de gestiune a bazelor de date era cel relațional (suveranitate manifestată din plin și astăzi), iar SGBD-urile ce puteau exploata baze de date obiectuale erau ca și inexistente.

Pentru a vă face o idee completă asupra părții a doua din proiectarea bazelor de date în viziunea celor două autoare, iată o scurtă enumerare a principalelor etape

---

<sup>17</sup> [Flemming & vonHalle 89]

<sup>18</sup> [Flemming & vonHalle 89]. p.6

<sup>19</sup> [Flemming & vonHalle 89]. pp.19-25

din așa-numita *proiectare a bazei de date relaționale* (PBDR) care continuă operațiunile derulate în MLD<sup>20</sup>:

1. identificarea tabelelor;
2. identificarea atributelor;
3. adaptarea structurii datelor la specificul SGBD-ului folosit;
4. inventarierea regulilor organizaționale privind entitățile;
5. inventarierea regulilor organizaționale privind relațiile (asociațiile) dintre entități;
6. inventarierea regulilor adiționale despre atribute;
7. optimizarea structurii în vederea unui mecanism de acces cât mai eficient;
8. definirea secvențelor de "înmănunchiere" (clustering);
9. definirea cheilor pentru calcularea adreselor logice ale înregistrărilor în tabele (hash keys);
10. adăugarea indecșilor;
11. adăugarea de date redundante;
12. redefinirea coloanelor;
13. redefinirea tabelelor.

Literatura și practica se raportează însă preponderent la doar două componente majore ale proiectării bazelor de date: proiectarea *logică* și proiectarea *fizică*. Din acest punct de vedere, cele două 12 etape din MLD și primele 6 din așa numită PBDR ar corespunde, în linii mari, proiectării logice a bazei, în timp ce etapele următoare (7-13) din PBDR proiectării fizice.

O altă abordare, prezentată în [Teorey99], structurează ciclul de viață al bazelor de date pe următoarele etape<sup>21</sup>:

- I. Analiza cerințelor.
- II. Proiectarea logică, care se materializează în schema globală ce conține datele și relațiile dintre ele, fiind derulată astfel:
  - a. modelarea E-R;
  - b. integrarea perspectivelor utilizator;
  - c. conversia diagramelor E-R în tabele<sup>22</sup>;
  - d. normalizarea tabelor.
- III. Proiectarea fizică: selectarea indecșilor, metodelor de acces, clusterelor de date; tot aici, Teorey include și denormalizarea;
- IV. Distribuirea datelor, materializată în schema de fragmentare și schema de alocare a datelor;
- V. Implementarea, monitorizarea și modificarea ulterioară a bazei de date

Și aici apar dubii legate de modul de derulare a normalizării (II.d), odată ce prin conversia diagramelor E-R se obțin deja tabelele relaționale (II.c).

Deși clasice, ambele lucrări din care am extras activitățile de mai sus sunt legate îndeosebi de paradigma relațională, folosind, ce-i drept, în primele etape modelul

---

<sup>20</sup> [Flemming & vonHalle 89], pp.72-79

<sup>21</sup> [Teorey99], pp.4-10

<sup>22</sup> Teorey le numește *tabele SQL*

E-R, mai apropiat de lumea reală. Robert Muller abordează în cartea sa atât aspectele generale ale proiectării, cât și cele specifice fiecăruia dintre cele trei modele în vogă astăzi, relațional, obiectual și relațional-obiectual. Pentru autor, procesul proiectării bazelor de date este prin excelență unul iterativ, incremental, într-o măsură cu atât mai mare cu cât ne apropiem de obiectual. Fiecare iterație se materializează într-o formă a schemei bazei, pornindu-se de la modelare, apoi trecându-se la proiectare la construcție (implementare)<sup>23</sup>. Cu toate acestea, activitățile principale ale ciclului de viață al bazei de date nu diferă prea mult, cel puțin la nivel general<sup>24</sup>:

- analiza cerințelor informaționale;
- modelarea datelor;
- proiectarea și optimizarea bazei de date (proiectare logică și fizică);
- testarea și revizuirea bazei;
- certificarea bazei de date;
- întreținerea și ameliorarea bazei.

Specifică lucrării lui Muller este activitatea de certificare a bazei care se constituie ca un fel de atestat de bună funcționare acordat bazei, atestat merit a întări încrederea beneficiarilor bazei.

Connolly și Begg<sup>25</sup> definesc însă trei tipuri: *proiectarea conceptuală*, ce ține de construirea unui model informațional independent de orice considerent privind aspectul fizic al datelor; *proiectarea logică*, care constă în construirea unui model informațional pe calapodul unui model de date consacrat (E-R, relațional, OO, OR), dar independent de tipul SGBD-ului ales și de celelalte aspecte fizice ale modelului; *proiectarea fizică* care se materializează în implementarea efectivă bazei pe suportul de stocare, inclusiv aspecte ce țin de asigurarea unui acces eficient și securității.

Iar ca lucrurile să se complice și mai rău, se cuvine de remarcat că Elmasri și Navathe delimitează, precum Connolly și Begg, trei categorii ale proiectării, însă pentru ei proiectarea logică constă în conversia schemei conceptuale într-un model de date *dependent* de SGBD-ul care va fi folosit<sup>26</sup>.

Fideli majorității, vom alege varianta mai simplă, cu cele două tipuri de proiectare, logică și fizică, deși în capitolele următoare ne vom raporta, aproape exclusiv, la primul.

## 1.5. Proiectarea bazelor de date în prezentul volum

Materialul prezentat până în acest moment este destinat a contura areal acoperin de restul acestei lucrări. Mai precis, acest prim volum dedicat proiectării

---

<sup>23</sup> [Muller99], p.1

<sup>24</sup> [Muller99], pp.1-10

<sup>25</sup> [Connolly&Begg 02], pp.281-282

<sup>26</sup> [Elmasri & Navathe 00], pp.532-535



bazelor de date se „remarcă” prin deraierea de la metodologia ortodoxă de proiectare a bazelor de date. Practic, nu vom discuta nici de modelul Entități-Relații, nici de diagrame UML, nici de perspective-utilizator. Este posibilă proiectarea bazelor de date fără a avea deprinderi elementare în diagrame E-R, UML etc. ? Contrar majorității covârșitoare a autorilor, eu spun că da !

Aceasta nu înseamnă că tot ce ține de analiza și proiectarea sistemelor informațional reprezintă un balast inutil, ci doar că, în foarte multe situații, o bază de date poate fi adusă la o formă rezonabilă urmând metodologia descrisă în următoarele capitole. Dacă vreți, este o repunere în drepturi, ce-i drept, destul de tardivă a metodologiei curat... relaționale într-o perioadă în care cuvintele cheie în analiză și proiectare sunt E-R, OO, Rational Rose, UML etc. Abia după ce vom fi epuizat discuția despre sușurile și coborâșurile metodologiei bazate pe normalizare și post-normalizare, vom reveni în „lumea bună” a proiectării, în speratul volum al doilea dedicat proiectării bazelor de date.

De fapt, și celelalte câteva cărți publicate la noi în care se abordează problematica proiectării bazelor de sunt destul de eterogene ca metodologie. Spre exemplu, Octavian Băscă tratează problema în capitolul 5 al cărții sale din 1997<sup>27</sup>, însă, pe de o parte, modalitatea de tratare este copleșitor teoretică, exemplele fiind cu totul fragmentare, iar, pe de altă parte, în capitol sunt incluse și chestiuni mai puțin legate de construirea unei baze de date, cum ar fi optimizări ale interogărilor.

Metodologia prezentată în Lungu s.a.<sup>28</sup> este ceva mai laborioasă, fără însă a rivaliza cu cea din lucrările clasice străine, bazându-se pe patru etape:

- analiza sistemului economic și cerințelor informaționale asociate;
- proiectare structurilor conceptuale, externe și interne ale bazei de date;
- încărcarea datelor în bază;
- exploatarea și întreținerea bazei de date.

Metodologia folosește modelul E-R, însă, curios lucru, diagramele E-R sunt incluse în etapa de analiză a sistemului, nu în cea de proiectare a structurii. De asemenea, este discutabilă desemnarea activității de alegere a sistemului de gestiune a bazei de date drept primă etapă în proiectarea structurii, precum și modul în care autorii delimitează faza de încărcare a datelor de cea de exploatare și întreținere a bazei.

Probabil singura carte românească dedicată, cel puțin explicit, exclusiv proiectării bazelor de date este cea a dnei Ileana Popescu<sup>29</sup>. Trecând peste amănuntul includerii arhitecturii interne a sistemului Oracle într-o lucrare de modelare (pp.26-37), lucrării i se pot aduce și laude și comentarii (răutăcioase) cum ar fi:

- în capitolul 2, dedicat proiectării bazelor de date relaționale, mai întâi se transformă diagramele E-R în tabele relaționale (paragraful 2.3), iar abia în

---

<sup>27</sup> [Băscă97], pp.90-155

<sup>28</sup> [Lungu s.a.95], pp.26-59

<sup>29</sup> [Popescu01]

paragraful se prezintă normalizarea, fără a se înțelege ce anume se normalizează, atâta vreme cât schema bazei a fost deja obținută;

- este discutabilă includerea în capitolul 2 a limbajelor de interogare, de fapt concentrarea exclusivă pe aspectul manipulativ al lor (DML), și nu pe cel de definire a datelor (DDL);
- regulile lui Codd (paragraful 2.5) sunt reguli ale unui SGBD relațional, nu ale proiectării unei baze de date relaționale;
- prezentarea utilitarului SQL\*Plus (paragraful 2.8.2) și a limbajului de programare (extensia procedurală a SQL-ului) Oracle PL/SQL (paragraful 2.8.3) în subcapitolul dedicat normalizării este într-un totu discutabilă.

În concluzie, cărților românești dedicate proiectării bazelor de date li se poate imputa un cert „defect metodologic”, dar mai ales inapetența cronică pentru aplicativ, lipsa oricăror soluții destinate practicienilor. Nu este de mirare faptul că, atât la noi, cât și în majoritatea lucrărilor internaționale, bazele de date se află în postura unui ghiveci din care fiecare înțelege ce vrea și ce poate.

Pornind de la faptul că, în esență, un model de date este legat de trei aspecte, *structură* (modelul de organizare a datelor), *integritate* (restricțiile ce pot fi definite asupra datelor pentru a asigura corectitudinea informațiilor din bază) și *manipulare* (modalități de editare și extragere a informațiilor din bază), capitolele ce urmează sunt concentrate pe aspectul structural și mecanismul de integritate ale unei baze de date. Proiectarea fizică, mecanismele de control, concurență și parametrii fizice ai bazei de date sunt, în cea mai mare parte, eludate în acest volum.

Practic, lucrarea de față își propune să vă supună atenției cum anume s-ar putea „gândi” o schemă de bază de date pentru a o aplica anume, ce elemente ar trebui luate în considerare, chestiunile mai riguroase și cele mai laxe. Întregul demers este bazat pe normalizare și operațiuni post-normalizare. În plus, ca parte integrantă a bazei de date, un spațiu generos este acordat mecanismului de declararea a restricțiilor pentru asigurarea într-o cât mai mare măsură a integrității, coerenței și consistenței datelor din bază. Din acest punct de vedere, vom urma observația lui Chris Date, după care proiectarea BD nu este doar o chestiune de găsire a celor mai bune structuri de stocare; integritatea datelor nu reprezintă doar un aspect, ci aspectul cheie<sup>30</sup>.

---

<sup>30</sup> [Date04], p.330