

# MODELAREA DATELOR

Proiectarea corectă a structurii unei baze de date relaționale este o premisă fundamentală în scrierea programelor de aplicație și în funcționarea lor fără anomalii care pot apărea în cazul unei structuri defectuoase. Acest capitol prezintă un model de date folosit în proiectarea conceptuală de nivel înalt numit **modelul entitate asociere (EA)** în varianta clasică (cu unele extensii). Într-un alt capitol vor fi prezentate regulile de transformare din modelul entitate-asociere în modelul relațional.

## 2.1. Etapele dezvoltării unei aplicații

Proiectarea structurii bazelor de date relaționale este un domeniu în care cercetările au evoluat spre elaborarea unor metodologii teoretice și a unor instrumente software care asigură un grad ridicat de normalizare a schemelor de relație, păstrarea integrității, evitarea anomaliilor datorate unei proiectări nesistematice și eliminarea subiectivismului proiectantului prin înlocuirea lui cu exactitatea metodei.

Până la apariția unor astfel de metodologii se pornea de la o colecție de tabele și de la dependentele funcționale și multivalențelor asociate și se ajungea, prin aplicarea unor algoritmi sau metode de normalizare la schema dorită a bazei de date. Această abordare de tip bottom-up crează însă dificultăți în cazul bazelor de date complexe în care numărul de tabele și de dependente este mare. Probabilitatea ca unele interdependente între date să nu fie sesizate în procesul de proiectare este în aceste cazuri ridicată rezultând în exploatarea anomaliilor care duc la reluări ale ciclului analiză cerințelor → schema bazei de date → programe de aplicație.

Cercetările în domeniul proiectării schemei conceptuale s-au concentrat pe găsirea unor metodologii top-down pentru obținerea unei structuri optime a BD. Ele au fost influențate de rezultatele obținute în domenii care folosesc bazele de date: inteligența artificială, proiectarea asistată de calculator, abordarea orientată pe obiecte.

Impactul conceptelor de abstractizare și generalizare precum și elaborarea unui model de descriere informală a datelor, și anume modelul entitate-asociere (EA), au dus la găsirea unor cai algoritmicizabile de proiectare optimă a bazelor de date. Modelul entitate-asociere este în acest moment cel mai popular model de comunicare a structurii bazelor de date datorită intuitivității și simplității elementelor sale. Îmbunătățiri sale ulterioare prin

folosirea abstractizarilor și generalizarilor au dus la crearea de variante ale modelului, doua dintre acestea fiind descrise in acest capitol.

Extensiile modelului EA au aparut și pentru alte necesitati:

- modelarea cerintelor de secretizare a datelor,
- documentarea programelor de aplicatie și usurarea comunicării între proiectantul de sistem și utilizatorul obisnuit,
- proiectarea bazelor de date complexe pe portiuni și integrarea ulterioara a acestora (asa numita integrare a vederilor).

Avantajul principal al modelului EA este acela al simplitatii sale și al caracterului sau intuitiv. Rezultatul proiectarii consta într-o diagrama entitate-asociere care poate fi apoi translata in modelul de date folosit de sistemul de gestiune a bazelor de date ales pentru dezvoltarea aplicatiei.

Figura 2.1. prezinta schematic etapele proiectării unei noi aplicatii care gestioneaza o baza de date, cu accentul pe partea de proiectare a structurii acesteia. Aceste etape sunt detaliate in paragrafele urmatoare.

### 2.1.1. Analiza de sistem

In aceasta etapa se realizeaza analiza segmentului din lumea reala care va fi gestionat de aplicatia respectiva. Rezulta o specificatie neformalizata a cerintelor constand din doua componente:

1. **Cerinte privind continutul bazei de date:** categoriile de date care vor fi stocate și interdependentele dintre acestea.
2. **Cerinte privind prelucrarile efectuate de aplicatie:** prelucrarile efectuate asupra datelor, arborele de meniuri al aplicatiei, machetele formatelor de introducere și prezentare a datelor și ale rapoartelor tiparite de aceasta.

In general aceasta etapa nu poate fi asistata prin programe de tip CASE dar exista reguli care ajuta proiectantul in realizarea sa. Activitatile desfasurate includ:

- Analiza activitatii desfasurate la momentul respectiv de beneficiarul aplicatiei sau de o multime reprezentativa de beneficiari in cazul aplicatiilor de uz general.
- Analiza continutului de date și a functionalitatii aplicatiilor software - daca exista - care vor fi inlocuite de noua aplicatie incluzand meniuri, machete ecran și machete de rapoarte.
- Analiza formularelor tipizate și a altor documente utilizate de beneficiar pentru realizarea activitatii respective.
- Identificarea tuturor interdependentelor dintre datele care vor fi stocate in baza de date și a restrictiilor privind valorile pe care le pot lua anumite categorii de date.
- Identificarea - daca este cazul - a prelucrarilor care se declanseaza automat atat in cazul modificarii bazei de date cat și la momente prestabilite de timp (de exemplu sfarsit de luna, de an, etc.)
- Identificarea operatiilor care sunt necesare beneficiarului in activitatea curenta dar care in acest moment nu sunt realizate prin intermediul aplicatiilor software folosite precum si a operatiilor care pot fi incluse in mod natural in noua aplicatie.

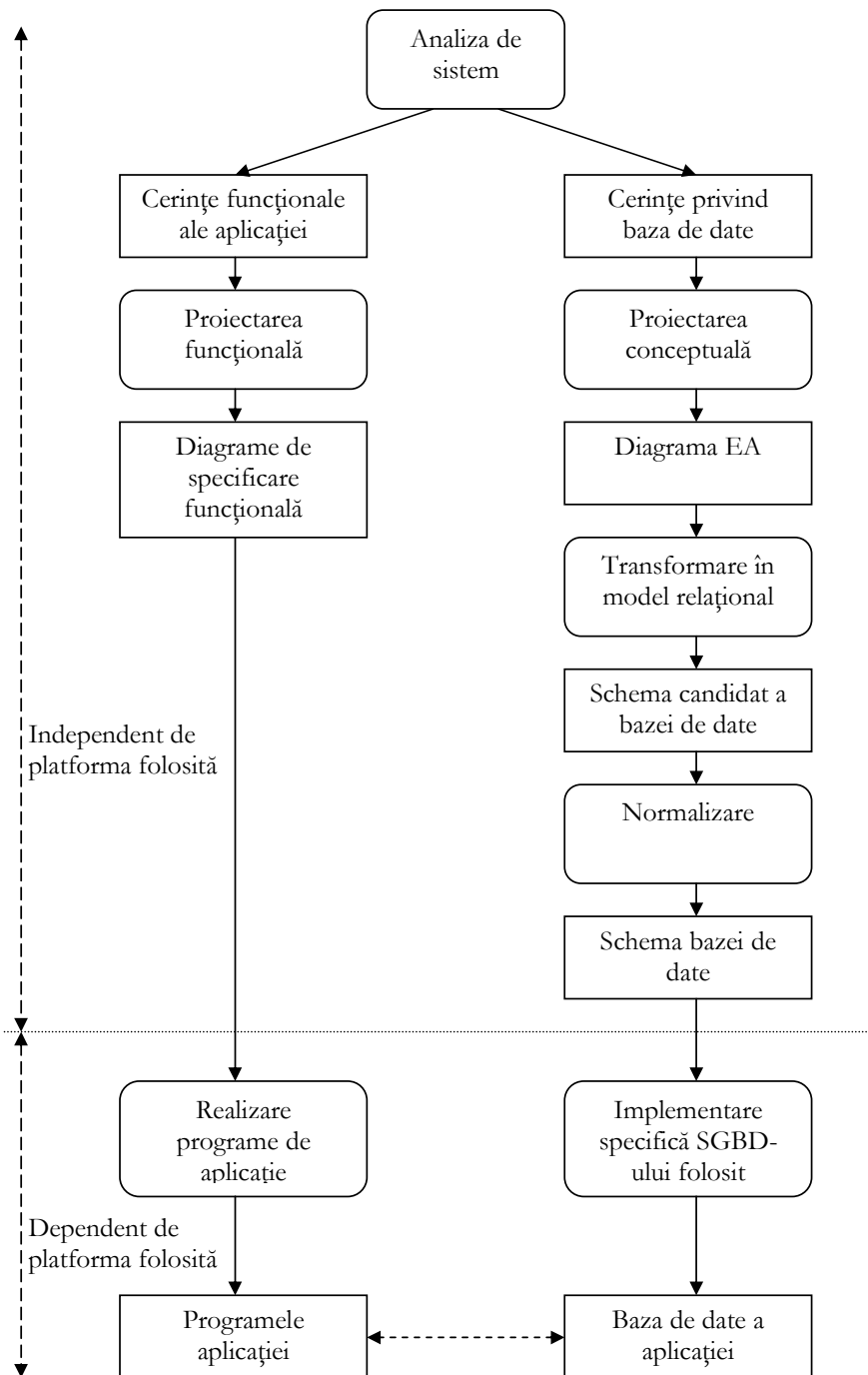


Fig. 2.1. Etapele proiectării unei aplicații

- Identificarea bazelor de date existente care pot fi folosite de noua aplicatie - direct sau printr-un import initial de date - evitandu-se in acest fel reintroducerea manuala a acestora.
- Identificarea modalitatilor de transfer de date intre noua aplicatie și alte aplicatii care ruleaza deja la beneficiar și care vor fi folosite și in viitor de catre acesta.
- Identificarea necesitatilor privind datele și prelucrarile care pot fi in viitor necesare beneficiarului, deci a posibilelor dezvoltari in timp ale aplicatiei.

Aceasta etapa este efectuata de personal calificat avand in vedere ca rezultatele sale sunt baza de la care se pleaca in etapele urmatoare, eventualele erori putand fi corectate ulterior cu costuri semnificative.

### 2.1.2. Proiectarea conceptuală a bazei de date

In aceasta etapa, pornind de la rezultatele analizei de sistem, se realizeaza modelarea cerintelor privind datele folosind un model de nivel inalt. Cel mai popular model folosit pentru aceasta este modelul entitate-asociere (EA). Actualmente exista pe piata numeroase instrumente CASE care folosesc diverse variante ale modelului. Motivele pentru care a fost ales sunt urmatoarele:

- Nu este legat direct de nici unul dintre modelele folosite de sistemele de gestiune a bazelor de date (relational sau orientat obiect) dar exista algoritmi bine pusi la punct de transformare din model EA in celelalte modele de date.
- Este intuitiv, rezultatul modelarii fiind o diagrama care defineste atat datele stocate in baza de date cat și interdependentele dintre acestea.
- Poate fi usor de inteles de nespécialisti. Aceasta caracteristica este foarte importanta in momentul in care se face punerea de acord cu beneficiarul asupra structurii bazei de date a aplicatiei, evitandu-se in acest fel o proiectare neconforma cu realitatea sau cu cerintele exprimate de acesta.
- Proiectarea se poate face pe portiuni, diagramele parțiale rezultate putand fi apoi integrate pe baza unor algoritmi și metode bine puse la punct.

### 2.1.3. Transformare în model relațional

In aceasta etapa entitatile și asocierile care formeaza diagrama EA se transforma pe baza unor reguli clare in structura relationala a bazei de date. Rezulta schema preliminara a acesteia formata din tabele (relatii in terminologia relationala), coloanele acestora (atribute ale relatiilor) și constrangerile de integritate care pot fi deduse automat din diagrama incluzand unele interdependente intre date numite și **dependente functionale**. In capitolul 3 este descrisa o metoda de transformare din modelul EA clasic in modelul relational. In cazul variantei specifice uneltelor CASE transformarea se face automat de catre acestea.

### 2.1.4. Normalizare

Exista o serie de reguli care descriu ce inseamna o structura corecta a unei tabele și care definesc asa numitele **forme normale**. Pe baza structurii bazei de date și a dependentelor rezultate atat din transformarea in model relational cat și a altor dependente identificate de proiectant in analiza de sistem se poate face o operatie numita **normalizare** modificand structura bazei de date astfel incat toate tabelele din aceasta sa fie in forma normala dorita. Capitolul 3 contine definitia formelor normale uzuale și descrierea unor

algoritmi de normalizare care sunt folositi de uneltele CASE pentru efectuarea automata a operatiiei.

### 2.1.5. Implementarea specifică sistemului de gestiune folosit

In aceasta etapa se realizeaza crearea structurii bazei de date obtinuta in etapa precedenta pe baza facilitatilor oferite de sistemul de gestiune a bazelor de date ales. Rezultatul ei este programul de creare scris in limbajul de definitie a datelor acceptat de SGBD-ul utilizat. Iata un exemplu:

Schema conceptuala:

**Student(CodStudent:Numeric, Nume:Şir, DataNasterii:Dată)**

Program de creare (SQL-Oracle):

```
Create table Student(  
    CodStudent Number(5) Primary Key,  
    Nume Varchar2(40),  
    DataNasterii Date);
```

Programul contine atat definirea tabelor cat și definirea celorlalte obiecte ale bazei de date și a interdependentelor dintre acestea (de exemplu constrangerile de integritate intra-tabela și inter-tabele).

De asemenea aici se fac și toate operatiile privind proiectarea la nivel fizic a bazei de date. In cazul folosirii de unor uneltele CASE programul de creare poate fi generat și executat automat.

Prezentam in continuare elementele care definesc modelul entitate asociere in cele doua variante: clasic și specific instrumentelor CASE.

## 2.2. Modelul entitate-asociere clasic

Acest model a fost introdus de P. P. Chen in 1976 ([Ch 76]). El se constituie într-o abordare de tip grafic a proiectării bazelor de date și a fost adoptat de comunitatea științifică precum și de producătorii de software în domeniu datorită simplității și expresivității sale.

### 2.2.1. Elementele modelului

Modelul entitate-asociere permite reprezentarea informațiilor despre structura bazelor de date folosind trei elemente de construcție: entități, atribute ale entităților și asocieri între entități. Definiția lor informală este următoarea:

**Entitățile** modelează clase de obiecte concrete sau abstracte despre care se colectează informații, au existență independentă și pot fi identificate în mod unic. Exemple de entități: Studenți, Orașe, Angajați, etc. Ele definesc de obicei persoane, amplasamente, obiecte sau evenimente cu importanță informațională. Membrii unei clase care formează o astfel de entitate poartă numele de **instante** ale acelei entități. De remarcat că întreaga literatură de specialitate definește întâi conceptul de mulțime de entități (sau tip de entități) pentru ca apoi să adopte pentru ușurința exprimării prescurtarea de entitate pentru acest concept. Deci entitatea este un obiect generic care reprezintă mulțimea tuturor instanțelor sale.

Entitățile sunt de două categorii:

1. Entitati independente (sau tari) sunt cele care au existenta independenta de alte entitati,
2. Entitati dependente (sau slabe) sunt formate din instante care isi justifica incadrarea in clasa respectiva doar atita timp cit intr-o alta entitate (tata) exista o anumita instanta de care sunt dependente. De exemplu in cazul unei baze de date de personal, fiecare instanta a entitatii COPII ramine in clasa respectiva (copiii angajatilor) atit timp cit in entitatea ANGAJATI exista instanta reprezentand pe tatal/mama acelu copil.

Element al modelului	Tip	Reprezentare	Exemplu
Entitate	Tare		
	Slaba		
Atribut	De identificare		
	De Descriere		
Asociere	Asociaza 1-2 entitati		
	Asociaza mai mult de 2 entitati (exemplu: 3 entitati)	3:  ..... 6:  .....	

Fig. 2.2. Conventia de reprezentare a elementelor modelului EA

**Atributele** modeleaza proprietati atomice distincte ale entitatilor. De exemplu entitatea STUDENTII poate avea ca atribute Matricola, Nume, Prenume, Varsta, Anul, Grupa, etc. In procesul de modelare vor fi luate in considerare doar acele proprietati ale entitatilor care sunt semnificative pentru aplicatia respectiva. Din acest motiv, la entitatea STUDENTII nu vom lua in considerare caracteristici ca Talia sau Culoarea\_parului acestea nefiind necesare pentru baza de date a universitatii (astfel de atribute ar putea exista de exemplu intr-o baza de date privind personalul militar).

Atributele unei entitati sunt de doua feluri:

- **atributele de identificare** (formand impreuna **identificatorul entitatii**) reprezinta acea multime de atribute care permit distinctia intre instantele aceleiasi entitati

- **atributele de descriere** (sau descriptori) sunt folositi pentru memorarea caracteristicilor suplimentare ale instantelor.

In exemplul de mai sus Matricula este atribut de identificare (deoarece nu pot exista doi studenti cu aceeași matricula într-o facultate) pe când celelalte atribute sunt descriptori.

**Asocierile** modeleaza interdependentele dintre clasele de obiecte reprezentate prin entitati. De exemplu între entitatile STUDENTII și FACULTATI se poate figura o asociere INSCRIS\_LA care descrie impartirea studentilor pe facultati.

In crearea diagramei nu vor fi luate in considerare decit interdependentele care sunt necesare aplicatiei respective, in lumea reala putand exista între entitatile diagramei și alte asocieri care nu sunt semnificative in contextul dat.

Figura 2.2. prezinta conventia de reprezentare grafica a celor trei tipuri de constructii care participa la formarea unei diagrame EA. Se observa ca:

- Entitatile se reprezinta prin dreptunghiuri in care este inscris numele entitatii. In cazul entitatilor dependente (slabe), conturul va fi cu linie dubla.
- Atributele se reprezinta prin cercuri (sau ovale) in interiorul carora apare numele atributului. Ele sunt conectate cu un segment de dreapta la entitatea de care apartin. Pentru a distinge atributele de identificare de cele de descriere, numele primelor va fi subliniat.
- Asocierile se reprezinta prin romburi (daca conecteaza una sau doua entitati) sau poligoane regulate (daca conecteaza mai mult de doua entitati) conectate prin segmente de dreapta la entitatile asociate, avand inscris in interior (sau alaturi) numele asocierii. Alte elemente grafice specificand caracteristici suplimentare ale asocierilor vor fi introduse in paragrafele urmatoare.