7 PROIECTAREA SI CREAREA BAZELOR DE DATE RELAȚIONALE

1. Tabele şi relații dintre tabele

Într-o bază de date relațională entitățile se reprezintă logic prin conceptul de TABEL. Tabelul a fost numit „relație" – relation, de teoreticienii care au dezvoltat acest tip de SGBD, de unde şi numele de BDR – baze de date relaționale.

Un tabel se compune din :

- *rânduri,* care corespund membrilor entității;
- *coloane,* care corespund atributelor entității.

Tabelele se memorează în fişiere. Fiecare rând înseamnă o înregistrare iar fiecare coloană înseamnă un câmp. Fiecare tabel se descrie prin intermediul structurii sale logice care este asemenea structuri logice a oricărei entități.

Sistemele de gestiune a bazelor de date relaționale permit reprezentarea cu uşurință a relațiilor dintre tabele. O relație reflectă asocierea logică ce se poate face între înregistrările ce aparțin de tabele diferite, uneori chiar şi între înregistrările aceluiaşi tabel. De exemplu înregistrările din tabelul ***furnizori*** se asociază firesc cu înregistrările din tabelul ***comenzi.*** Un furnizor poate încheia mai multe contracte. Pentru a le analiza, datele din cele două tabele trebuie alipite, concatenate.

Tabelul: **FURNIZORI**

Câmp	Tip	Lungime
CODF*	T	2
DENF	T	20
ADRESA	T	20

câmp comun — **CODF**

Tabelul: **COMENZI**

Câmp	Tip	Lungime
NRCDA*	**N**	**4**
DATA_CONT	D	8
CODF	***T***	***2***
VALOARE	N	12

CODF	DENF	ADRESA
F1	AGER	BUC
F2	DD	CLUJ
F3	TIM	BUC
F4	DACIA	PITESTI

NRCDA	DATA_C	CODF	VALOARE
400	11/01/04	F2	14000
401	13/01/08	F1	23000
402	04/02/08	F4	18000
403	04/02/08	F1	40000
404	12/02/08	F4	38000
405	18/02/08	F3	60000
406	18/02/08	F2	80000

ASOCIERE - JOIN — **CODF**

NRCDA	DATA-C	CODF	DENF	ADRESA	VALOARE
400	16/01/08	F2	DD	CLUJ	14000
406	18/02/08	F2	DD	CLUJ	80000

Relațiile dintre tabele pot fi de tipul:

- unu la unu, **1 : 1**
- unu la mulți , **1 : M**
- mulți la unu, **M : 1**
- mulți la mulți, **M : M**

Pentru a se realiza fizic asocierea dintre două tabele este necesar să avem câmpuri comune în cele două tabele.

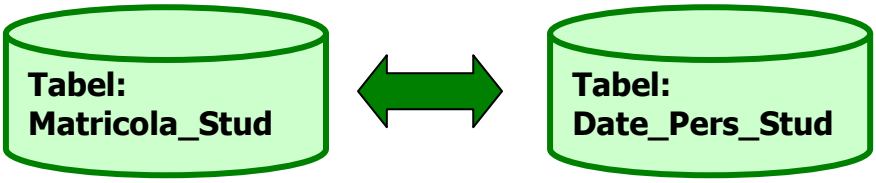


Fig.1 Asociere unu-la-unu

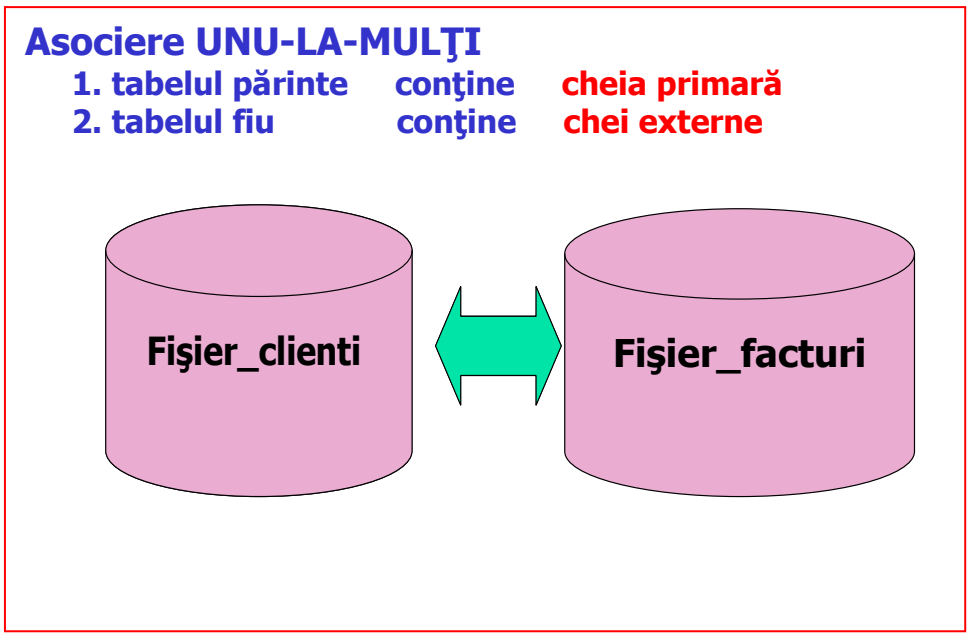
Pentru a ilustra mai bine asocierea de tip arborescent unu-la-mulți, de altfel cel mai uzual tip de asociere dintre două tabele, ele se denumesc tabela părinte şi tabela fiu..
Câmpurile comune pe baza cărora se face asocierea dintre cele două tabele se numesc:

- **cheie primară**, pentru tabela părinte – primary key ;
- **cheie externă**, pentru tabela fiu – foreign key.

Ele pot avea acelaşi nume sau nume diferite, dar în mod obligatoriu trebuie să fi de acelaşi tip, să aibă aceeaşi lungime şi să ia valori în cadrul aceluiaşi domeniu de date.
În exemplul prezentat mai sus, câmpul comun prezent în cele două tabele este CODF, codul firmei care are următorul rol:

- CODF, este cheie primară în tabelul CLIENȚI
- CODF, este cheie externă în tabelul COMENZI

Se remarcă faptul că nu sunt doi furnizori cu acelaşi cod, pentru că o cheie primară nu admite valori duplicate, este unică. Dar în tabelul COMENZI, o firmă poate să apară de mai multe ori pentru că poate să încheie mai multe coMENZI cu un partener. Deci cheia externă admite valori duplicate.



Sistemele de gestiune a bazelor de date asigură posibilitatea descrierii şi gestiuni printr-o interfață grafică explicită a structurilor logice a tabelelor şi a relațiilor de asociere dintre tabele, pe baza cheilor. În plus la momentul definirii structurii logice SGBD-ul prin intermediul limbajului său LDD – limbajul de definire a datelor, permite introducerea unor restricții de integritate a datelor de intrare. Aceste restricții, denumite CONSTRAINTS, sunt condiții de validare logică pe care trebuie să le îndeplinească valorile introduse de operatori în tabele bazei de date, pentru ca baza de date să fie coerentă şi corectă. Iată şi câteva exemple.

- cheia primară trebuie să fie unică, fără duplicate, (UNIQUE);
- cheia primară trebuie completată obligatoriu, (NOT NULL);
- unei chei externe din tabelul fiu trebuie să-i corespundă întotdeauna o cheie primară validă în tabelul părinte, (REFERENTIAL INTEGRITY CONSTRAINT)
- valorile reale introduse de la tastatură să se încadreze în domeniul de valori al atributului, ca tip, lungime, semn, valoare minimă, valoare maximă sau alte corelații, formulate ca expresii logice de control.

2 Etapele proiectării BDR

Proiectarea unei baze de date este o activitate laborioasă şi necesită parcurgerea următoarelor etape:

- formularea problemei;
- analiza cerințelor informaționale şi definirea datelor de ieşire şi de intrare;
- definirea tabelelor, a structurii logice a acestora şi a relațiilor dintre tabele;
- optimizarea structurii bazei de date.

Odată ce procesul de proiectare a fost finalizat, folosind SGBD-ul ales, se trece efectiv la:

- realizarea programelor pentru crearea şi actualizarea bazei de date;
- elaborarea programelor de interogare a datelor şi afişare a rapoartelor;
- testarea programelor;
- definitivarea documentației.

Formularea problemei presupune stabilirea temei şi a obiectivelor aplicației informatice care va utiliza baza de date. Obiectivele sunt legate de asigurarea informațională a procesului de conducere. Deci, noi trebuie să ne gândim ca, prin existența unei baze de date, asigurăm fondul de informații corespunzătoare cerințelor managementului firmei. Baza de date trebuie să permită atât obținerea informațiilor de detaliu, elementare, cât şi calculul şi prezentarea unor indicatori sintetici, agregați.

Analiza cerințelor informaționale, pornind de la obiectivele formulate anterior, se concentrează asupra a două probleme:

- indicatorii, rapoartele, listele şi datele de ieşire care trebuie obținute;
- datele de intrare necesare pentru obținerea datelor de ieşire.

Datele de intrare se regăsesc de regulă, în documentele primare. Datele finale sunt rapoarte, liste, situații cu rezultate pe care le solicită compartimentele de conducere. Pentru indicatorii incluşi în rapoartele finale, trebuie să fie foarte clar precizat modul în care se calculează fiecare indicator. În consecință, se precizează algoritmii de calcul, regulile de totalizare, sau alte reguli de obținere a fiecărei coloane sau linie de total din rapoartele finale.

Definirea tabelelor şi a relațiilor dintre tabele este etapa următoare în proiectarea bazei de date. Analiza cerințelor informaționale şi a proceselor de prelucrare va conduce la identificarea datelor ce vor alcătui tabelele bazei de date. O entitate poate fi stocată în unu sau chiar mai multe tabele. Gruparea câmpurilor în tabele se realizează prin diverse metode. Dintre aceste metode, două sunt cele mai utilizate:

- analiza concordanței IEŞIRI – INTRĂRI;
- analiza semnificației semantice a datelor.

Analiza concordanței IEŞIRI – INTRĂRI este o tehnică specifică proiectării sistemelor informatice care identifică documentele primare din care se preiau câmpurile folosite în calculul datelor de ieşire. Aceste documente vor fi folosite la crearea şi actualizarea tabelelor bazei de date.

Analiza semnificației semantice a datelor are ca punct de plecare inventarierea câmpurilor prezente în situațiile finale şi apoi gruparea lor în tabele şi în documente primare.

Acest proces de analiză trebuie să stabilească şi cheile primare, cheile externe şi astfel să se definească relațiile dintre tabele.

Optimizarea structurii bazei de date este un proces prin care se urmăreşte:

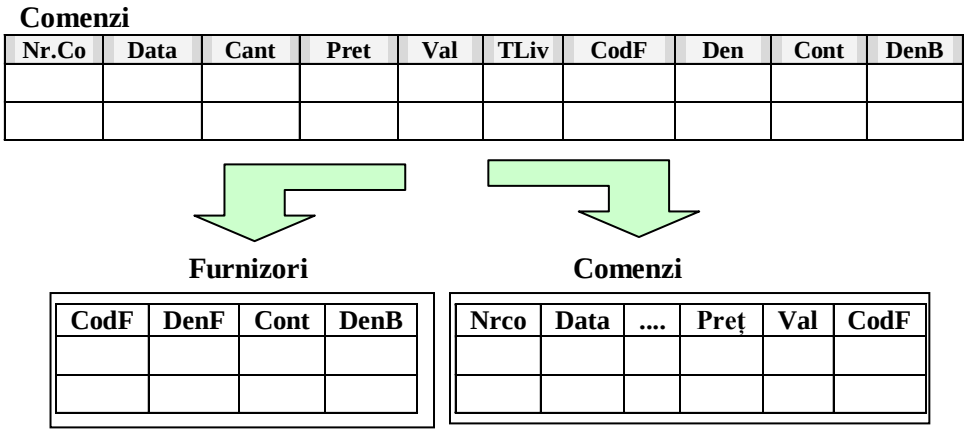
- reducerea redundanței datelor;
- eliminarea anomaliilor de actualizare.

Reducerea redundanței datelor până la un nivel minim şi controlat urmăreşte eliminarea duplicării inutile a unor câmpuri în mai multe tabele sau eliminarea câmpurilor obținute prin calcul pe baza câmpurilor atomice. Un anumit nivel de redundanță, însă, trebuie admis pentru a nu denatura realitatea reflectată de date. De exemplu, câmpul **VALOAREA_CONTRACTULUI**, se calculează după relația : ***VALOAREA_CONTRACTULUI=CANTITATE*PRET***

Nu este recomandată eliminarea acestui câmp pe considerentul că el se obține automat prin calcul. De exemplu, în cazul în care după un anumit interval de timp, prețurile suportă o majorare globală, cum se practică foarte des, atunci automat se vor modifica şi valorile contractelor încheiate anterior datei de majorare a prețurilor, ori acest lucru nu este corect, contractul odată perfectat nu-şi poate modifica prețul convenit prin negociere.

Anomaliile de actualizare se referă la anomaliile de ştergere, respectiv de modificare. sa consideram de exemplu o firmă care derulează lunar sute de comenzi de aprovizionare, pentru un nomenclator foarte mare de produse, dar care operează numai cu câțiva furnizori. Dacă în tabela **COMENZI** includem nu numai codul furnizorului ci şi denumirea furnizorului, contul său bancar sau denumirea băncii cu care lucrează, atunci va apărea următoarea anomalie de actualizare în cazul schimbării băncii şi a contului bancar al furnizorului. În loc

să modificăm aceste datele o singură dată doar în tabelul FURNIZORI va trebui să operăm modificarea în zeci, poate sute de rânduri în tabelul COMENZI-.



Fig, 2 Eliminarea anomaliilor de actualizare

Soluția este de a crea o tabelă separat, care are în structură: codul furnizorului, denumirea, contul şi banca acestuia, în tabela **COMENZI** păstrându-se doar codul furnizorului ca element de legătură, ceea ce previne pierderea de informații prin spargerea unui tabel în două.

Pornind de la această idee simplă a apărut una dintre cele mai utilizate proceduri de optimizare a structurii bazelor de date care se numeşte **normalizare.** Teoria normalizării aparține celui ce a fundamentat modelul relațional al bazelor de date în 1970, americanul E. F. Codd.[1]

3. Modelul conceptual, modelul logic şi modelul relațional

Simplificând puțin teoria, este suficient să reținem că activitatea de proiectare a unei baze de date se materializează în:

- modelul conceptual al bazei de date;
- modelul logic al bazei de date, şi am adăuga noi
- modelul relațional al bazei de date.

Mai intră în discuție şi modelul fizic al bazei de date care este dependent de particularitățile SGBD-ului utilizat, de particularitățile rețelei şi ale sistemului de operare. Prea tehnic pentru ceea ce ne interesează pe noi la acest nivel de informare asupra bazelor de date.

a) modelul conceptual este o reprezentare schematică a entităților şi a relațiilor dintre aceste la nivel global. este vorba de a reprezenta entitățile părinte, entitățile fiu, eventualele entități de legătură şi liniile de legătură dintre ele.

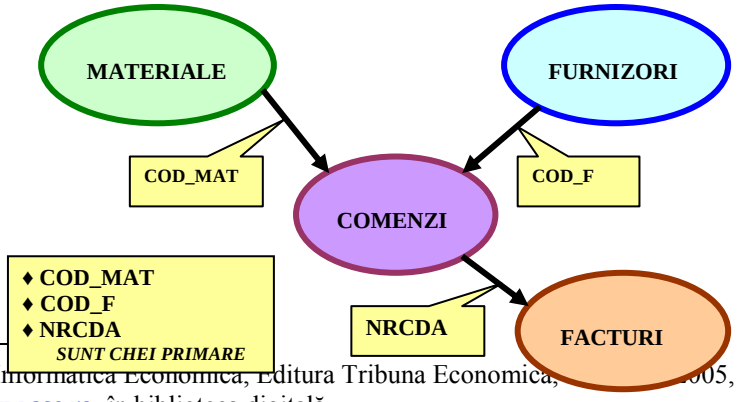


Fig. 3 Modelul conceptual al bazei de date

[1] Traian Surcel, ş.a Informatica Economică, Editura Tribuna Economica, [illegible]005, pg. 407, vezi şi versiunea online www.ase.ro, în bibliotaca digitală

b) modelul logic al bazei de date

Acest model urmăreşte definitivarea structurii logice a fiecărui tabel, pornindu-se de la lista câmpurilor tabelului, proprietăţile informaţionale ale fiecărui câmp şi precizarea restricţiilor de integritate, neapărat a cheilor primare şi a cheilor externe din fiecare tabel. Fiecare SGBD pune la dispoziţia userului o interfaţă specifică pentru lucru componenta LDD – limbajul de descriere a datelor. Am considerat că et mai didactic să reprezentăm separat „modelul relaţional" care trasează relaţiile (relationships) dintre tabele.

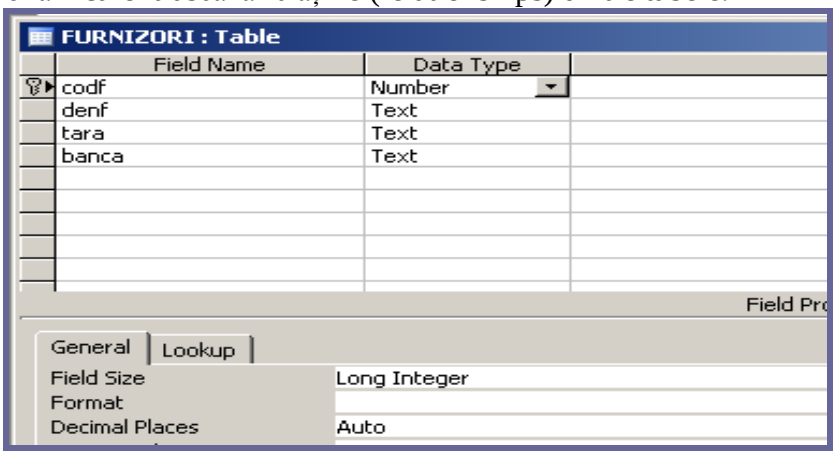


Fig. 4 Structură logică

c) modelul relaţional al bazei de date

Modelul relaţional reprezintă schema relaţiilor dintre tabelele bazei de date construită prin punerea în corespondenţă a cheilor primare cu cheile externe corespunzătoare.

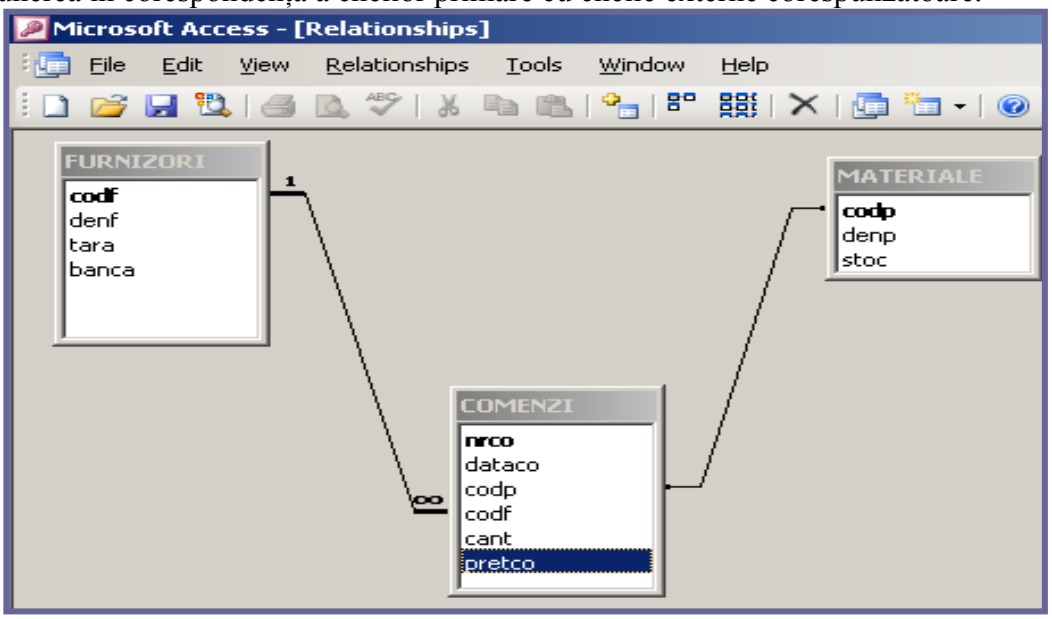


Fig. 5 Model relaţional

În spatele modelului logic şi al modelului relaţional se află instrucţiunile limbajului SQL – Structured Query Language, limbajul universal adoptat de SGBD-urile relaţionale.

4. Crearea bazei de date

Crearea bazei de date se realizează în trei paşi:

- construirea structurii logice a tabelelor bazei de date;
- construirea relațiilor dintre tabele;
- popularea tabelelor cu date reale.

Construirea structurii logice se realizează pentru fiecare tabel. Structurile create se memorează în tabele speciale denumite după caz: proiect, catalog sau master_data_base_file..

Construirea relațiilor folosind interfața grafică sau explicit cu instrucțiuni SQL va finaliza modelul relațional al bazei de date.

Popularea tabelelor înseamnă introducerea efectivă a datelor şi se realizează de regulă cu ajutorul formularelor electronice. Problema principală o reprezintă rezolvarea erorilor.