

Toate aceste tipuri de structuri interne de date își găsesc aplicabilitatea în realizarea programelor complexe pentru calcule economice, științifice, la realizarea jocurilor etc.

2.2. Organizarea datelor în fișiere

2.2.1. Caracteristicile fișierelor

Principala formă de organizare externă a datelor este fișierul. **Fișierul** reprezintă o colecție organizată de date, omogenă din punct de vedere al conținutului și al prelucrării, stocată pe un suport de memorie externă.

Fișierul ocupă un anumit spațiu pe suportul de memorie externă, spațiul alocat putând fi sau nu continuu, fiind sarcina sistemului de operare să gestioneze alocarea fizică a acestuia pe suportul de memorie.

Orice fișier este definit prin următoarele elemente:

- de natură externă:
 - nume și extensie fișier;
 - caracteristicile suportului de memorie;
 - attribute (Read Only, Hidden, System, Archive);
 - data creării, a ultimei accesări și modificări;
 - lungimea;
- de natură internă:
 - conținutul informațional (fișier de date, fișier program, fișier de comenzi, fișier text, fișier grafic etc.);
 - modul de organizare a conținutului (secvențial, relativ, secvențial-indexat);
 - modul de acces la înregistrări (secvențial, direct).

Datele într-un fișier sunt structurate în înregistrări logice. Organizarea înregistrărilor într-un fișier poate fi privită atât ca organizare logică cât și ca organizare fizică.

Din punct de vedere al *organizării externe* datele conținute într-un fișier sunt structurate în *înregistrări logice*. În principiu, un fișier conține același tip de înregistrări logice, dispuse într-o anumită ordine. Mulțimea înregistrărilor logice conținute într-un fișier reprezintă o mărime finită și caracterizează, în mod direct, mărimea oricărui fișier.

Din punct de vedere al utilizatorului, înregistrarea logică reprezintă și unitatea de acces la datele conținute într-un fișier. Organizarea logică a înregistrărilor, într-un fișier, ca mod de organizare externă a datelor, este impusă de natura lucrărilor și cerințele de prelucrare, în timp ce organizarea fizică este o organizare internă, fiind supusă rigorilor sistemului de calcul avut la dispoziție.

Totalitatea datelor conținute într-o înregistrare logică caracterizează o entitate informațională, se referă la același obiect, fenomen, persoană etc., în timp ce totalitatea datelor grupate într-o înregistrare fizică caracterizează o unitate de stocare fizică pe suportul de memorie a calculatorului.

2.2.2 Relația fișier – memorie – program

Cele două moduri de organizare a datelor, indiferent că se numesc organizare logică și fizică sau organizare externă și internă, nu se pot disocia. Întotdeauna utilizatorul operează cu concepte, privind organizarea datelor, la nivel logic, în vreme ce sistemul de operare stochează și manipulează datele, pe suportul de memorie, numai la nivel de înregistrare fizică.

În funcție de natura și complexitatea lucrărilor, lungimea unei înregistrări logice poate să corespundă sau nu cu lungimea unei înregistrări fizice. În multe cazuri o înregistrare fizică poate să conțină mai multe înregistrări logice (figura 2.9.)

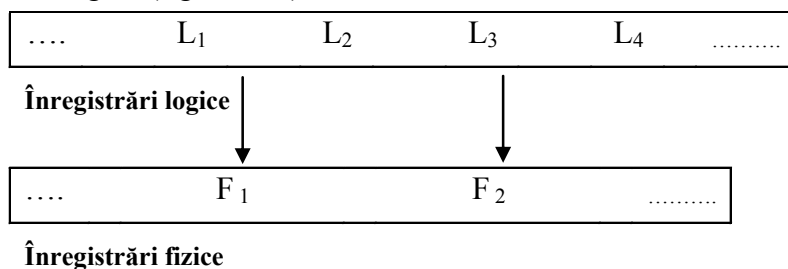


Figura 2.9. Gruparea înregistrărilor logice în înregistrări fizice

În timpul prelucrării fișierelor există un flux continuu de informații între fișierul de date, memoria internă și programul de prelucrare. Ordinea logică de încărcare și prelucrare în memorie este următoarea:

- se încarcă programul în memoria internă ;

- se aduc pe rând blocurile de înregistrări fizice în memoria internă, într-un spațiu de așteptare (buffer de intrare/ieșire), la dispoziția programului;
- fiecare înregistrare fizică este descompusă în înregistrări logice;
- înregistrările logice sunt prelucrate pe rând de program, iar rezultatele sunt furnizate la ieșire.

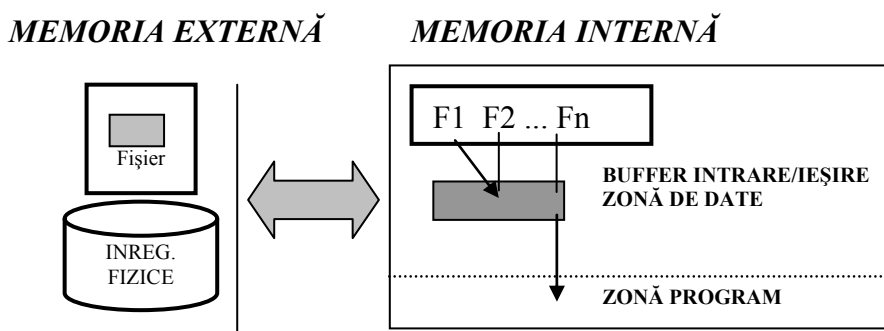


Figura 2.10. Fluxul informațiilor fișier de date, memorie, program de prelucrare

Fluxul informațiilor fișier de date – memorie – program de prelucrare este prezentat în figura 2.10.

2.2.3. Metode de organizare a fișierelor

Stocarea datelor în fișiere și accesarea acestora se realizează după anumite metode, care au în vedere, pe de o parte, specificul lucrărilor și cerințele prelucrării, iar pe de altă parte, suportul de memorie externă utilizat.

Dacă este avută în vedere complexitatea colecțiilor de date gestionate, metodele de organizare și de accesare a datelor în și din colecții există două metode de organizare a datelor: în *fișiere clasice* și în *baze de date*.

Metodele cele mai uzuale în cazul fișierelor clasice sunt:

- organizarea secvențială,
- organizarea indexată,
- organizarea relativă,
- organizarea directă.

Organizarea secvențială este o metodă simplă fiind admisă de orice suport de memorie. Ea constă în stocarea înregistrărilor pe suportul de memorie în ordinea cronologică a apariției lor. Pentru accesul la datele din fișierele organizate secvențial se utilizează numai modul de acces secvențial, care constă în parcurgerea una câte una, la rând, a tuturor înregistrărilor dinaintea celei căutate. Prelucrarea unui fișier secvențial se oprește la întâlnirea etichetei **EOF**. Organizarea secvențială se aplică fișierelor cu conținut informațional stabil în timp sau cu mișcări puține; se mai spune fișiere cu un grad mic de volatilitate.

Organizarea secvențial indexată utilizează ca suport de memorie discul magnetic și constă în stocarea înregistrărilor, ordonate ascendent după o *cheie de acces*, asociată fiecărei înregistrări, în momentul creării fișierului. Cheia de acces este unică, nu admite duplicate, și este a unui câmp al înregistrării. Un index reprezintă o pereche formată din cheia de indexare și adresa fizică de pe suport: $index = f(cheie, adresă)$. La crearea fișierului, înregistrările vor fi sortate ascendent după valoarea cheii de acces.

Organizarea relativă constă în asocierea, la fiecare înregistrare, a câte unui număr de ordine de la 0 la n (numărul articolului în fișier), care va indica poziția fiecărei înregistrări în fișier, în raport cu prima înregistrare și va servi pentru *accesul direct* al înregistrărilor.

Organizarea directă a fișierelor presupune stabilirea dinainte a adresei fizice a fiecărei înregistrări, care să servească atât la scriere cât și la citire. Operația de stabilire a acestei adrese fizice, deci a locului ocupat de fiecare înregistrare, se realizează pe baza unui algoritm care stabilește adresa fiecărei înregistrări ținând seama de caracteristicile și parametrii suportului de memorie, de modul cum este împărțit suportul la momentul formării discului.

Accesul la înregistrările dintr-un fișier se realizează atât în funcție de modul de organizarea a înregistrărilor în fișiere cât și de suportul de memorie externă pe care este memorat fișierul. Astfel, există acces: secvențial, direct și dinamic.

Accesul secvențial presupune regăsirea înregistrărilor în ordinea în care acestea au fost dispuse pe suportul tehnic. Pentru a determina o înregistrare oarecare, este necesar să se parcurgă toate înregistrările precedente. Accesul secvențial este adecvat lucrărilor seriale care prelucrează seturi voluminoase de date. Nu este însă eficient pentru căutarea rapidă a unei înregistrări sau a unui grup de înregistrări. Este

specific suporturilor neadresabile (banda magnetică).

Accesul direct se aplică suporturilor adresabile și permite identificarea directă a unei înregistrări pentru care se cunoaște adresa acesteia de pe suportul magnetic.

Accesul dinamic îmbina modelele anterioare realizând căutarea în două etape:

- poziționarea directă pe o anumită înregistrare a fișierului;
- consultarea secvențială a înregistrărilor următoare.

Organizarea secvențială a fișierelor permite doar accesul secvențial, organizările secvențial-indexată și relativă permit accesul direct după valoarea cheii (cheia trebuie cunoscută dinainte și precizată), accesul secvențial, care se realizează prin parcurgerea tuturor înregistrărilor până la cea căutată și accesul dinamic.

2.3. Organizarea datelor în baze de date

2.3.1. De la fișiere la baze de date

Organizarea datelor în fișiere clasice este o metodă rigidă care creează destule neajunsuri utilizatorilor sistemelor de programe bazate pe acest tip de fișiere. Cel mai mare dezavantaj este acela că modificările în structura unui fișier obligă la modificarea tuturor programelor care utilizează fișierul a cărui structură a fost modificată. Acest lucru se datorează faptului că programele scrise în limbajele algoritmice care lucrează cu fișiere trebuie să conțină clauze pentru descrierea detaliată a acestora.

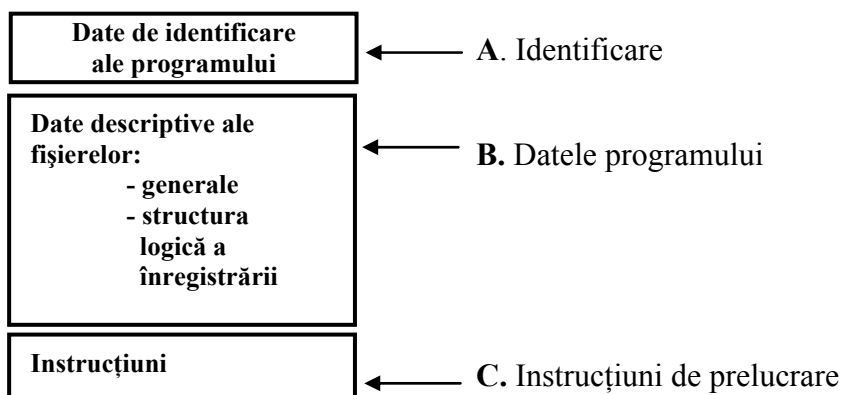


Figura 2.11. Structura de principiu a unui program clasic cu fișiere

Este vorba despre următoarele elemente care sunt descrise în cadrul programului:

- metoda de organizare și eventual modul de acces;
- structura logică a înregistrării;
- cheia de acces (*record key*), în cazul în care fișierul este indexat;

Aceste elemente sunt obligatorii pentru orice program. Instrucțiunile de prelucrare din program trebuie să fie în concordanță cu descrierea datelor, altfel apar erori în funcționarea programelor, deoarece nici un fișier nu poate fi prelucrat corect dacă descrierea sa din program nu este identică cu cea a fișierului real de pe disc.

În cazul în care structura logică a unei înregistrări este modificată prin adăugarea unui câmp, sau redimensionarea altuia, atunci toate programele care operează cu fișierul a cărui structură a fost modificată trebuie actualizate.

Pentru eliminarea acestor neajunsuri descrierea fișierelor nu a mai fost inclusă în fiecare program ci într-o tabelă sau într-un dicționar de date memorat separat, sau memorată în fișierul respectiv. Orice program care lucrează cu fișierul își va identifica datele prin intermediul acestei tabele sau dicționar. Despre fiecare câmp se va ști precis unde este localizat în înregistrare și ce descriere are.

Orice modificare în descrierea structurii logice va afecta numai tabela de descriere. Programele deja scrise nu mai trebuie actualizate, decât dacă operează chiar cu noile câmpuri incluse în fișier.

Această metodă de organizare a datelor în *bănci de date*, oferă atât avantajul separării descrierii datelor de programele de aplicație, dar și alte facilități.

În figura 2.12 este prezentat un exemplu de separare a descrierii structurii logice de datele propriu-zise prin intermediul unui dicționar de date.

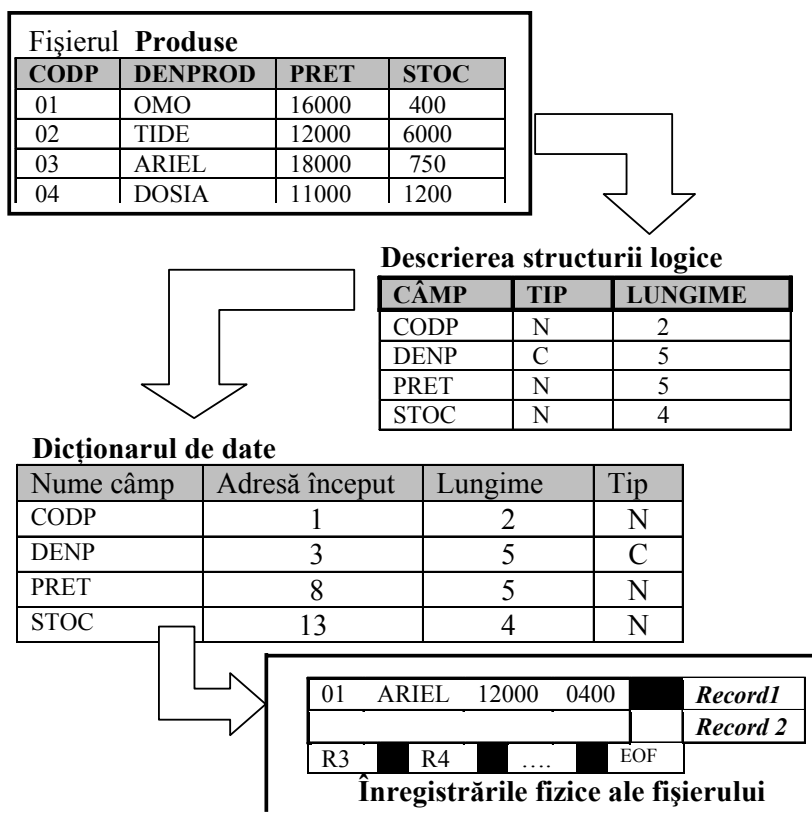


Figura 2.12. Separarea datelor de structura acestora

În această viziune de organizare apar trei elemente intercorelate:

- colecțiile de date propriu-zise;
- tabelele cu descrierea structurii logice a colecțiilor de date;
- relațiile, adică tabelele cu legăturile dintre colecții.

Toate aceste trei elemente reunite formează o bază de date. **Baza de date** se definește ca fiind ansamblul colecțiilor de date împreună cu structura acestora și relațiile dintre ele.

2.3.2. Obiectivele organizării datelor în baze de date

În continuare sunt prezentate principalele obiective ale organizării datelor în baze de date.

Asigurarea independenței relative a programelor față de structura datelor este primul obiectiv al organizării datelor în baze de date, și desigur cel mai important. El este îndeplinit tocmai prin memorarea distinctă a structurii logice a datelor în tabele – dicționare separat, așa cum am prezentat mai sus.

Asigurarea unei redundanțe minime și controlate a datelor este al doilea obiectiv major al organizării datelor în baze de date. Aceasta înseamnă că se urmărește pe cât posibil ca fiecare câmp să apară numai într-un singur fișier, deci numai o singură dată în bază. Excepție fac câmpurile comune ce apar în fișiere distincte și câmpurile care fac necesară întregirea semnificației semantice a datelor, a înțelesului acestora. Duplicarea datelor, redundanța, trebuie redusă la minim și menținută sub control.

O anumită redundanță trebuie acceptată pentru a asigura performanțe sporite programelor de accesare și prelucrare a bazei datelor.

Redundanța minimă se asigură prin tehnicile de proiectare a bazelor de date, de exemplu normalizarea bazelor de date relaționale.

Celor două obiective menționate până acum li se mai pot adăuga și altele la fel de importante, cum ar fi:

- asigurarea unor facilități sporite de utilizare a datelor;
- sporirea gradului de securitate și integritate a datelor;
- asigurarea partajabilității datelor.

Asigurarea unor facilități sporite de utilizare se realizează prin punerea la dispoziția utilizatorilor a unei interfețe prietenoase, ce permite un dialog concis pentru actualizarea și exploatarea bazei de date. Utilizatorii obișnuiți nu trebuie să cunoască structura întregii baze de date, ei pot să utilizeze un limbaj mai apropiat de limbajul natural al procedurilor de actualizare, interogare și afișarea datelor din baza de date.

Securitatea datelor vizează prevenirea accesului neautorizat la baza de date, lucru posibil prin filtrarea drepturilor de acces ale fiecărui utilizator și printr-un sistem de parole multinivel.

Păstrarea integrității datelor înseamnă prevenirea distrugerii accidentale a fișierelor, fapt care obligă la instituirea unui set de proceduri de autorizare, dar și de confirmare a operațiilor de ștergere, adăugare, precum și proceduri de realizare a unor copii de siguranță, a unor jurnale de urmărire a actualizărilor și proceduri de refacere a bazei de date, de restaurare a acesteia, în caz de incidente.

Partajabilitatea datelor trebuie înțeleasă nu numai sub aspectul utilizării în comun de către mai multe aplicații ale aceleiași baze de date, ci ca posibilitate a dezvoltării de noi aplicații, de scriere de noi programe ce necesită poate chiar extinderea/modificarea aplicațiilor aflate deja în funcțiune.

2.3.3. Sisteme de gestiune a bazelor de date (SGDB)

Pentru crearea unei baze de date, pentru actualizarea permanentă a conținutului acesteia și pentru consultarea și prelucrarea datelor stocate aici este utilizat un sistem specializat de programe cunoscut sub denumirea de **sistem de gestiune a bazei de date (SGDB)**. El este un instrument software la care utilizatorul apelează pentru comunicarea cu baza de date la fel cum apelează la sistemul de operare Windows pentru a comunica cu computerul. El este plasat la un nivel intermediar între utilizator și baza de date și asigură:

- o interfață prietenoasă pentru comunicarea cu utilizatorul;
- un mediu de programare performant, format dintr-un editor propriu de texte, un compilator, un utilitar pentru depanarea programelor, alte utilitare și instrumente de lucru cu baza de date.

Sistemul de gestiune a bazei de date dispune de un *limbaj de descriere a datelor* (LDD) și de un *limbaj de manipulare a datelor* (LMD) cu ajutorul cărora se asigură îndeplinirea funcțiilor sale și anume:

- definirea structurii bazei de date;
- manipularea datelor, adică: încărcarea datelor în baza de date, adăugarea, ștergerea, modificarea acestora, sortarea, indexarea, concatenarea ș.a.;
- utilizarea bazei de date, utilizarea în modul conversațional sau în modul program, ceea ce înseamnă: accesare rapidă, listare, calcule, editarea rapoartelor etc.;

- administrarea bazei de date, care presupune: optimizarea structurii bazei de date, validarea accesului la date, reorganizarea și compactarea bazei, jurnalizarea actualizărilor și refacerea bazei în caz de incidente pentru asigurarea securității, integrității sau partajabilității bazei de date.

2.3.4. Clasificarea bazelor de date

În funcție de condițiile concrete în care se realizează, o bază de date se încadrează într-o anumită categorie, având anumite caracteristici. Astfel criteriile ce se au în vedere la proiectarea și realizarea bazelor de date și particularitățile SGBD-ului pot constitui, în același timp, și criterii de clasificare a bazelor. Dintre acestea, următoarele sunt considerate ca fiind cele mai importante:

- domeniul de aplicații;
- modul de organizare, structurare și accesare a datelor;
- gradul de centralizare a datelor;
- modul de prelucrare.

După *domeniul de aplicații* bazele de date pot fi:

Baze de date universale, care satisfac toate cerințele în cadrul unei anumite structuri organizatorice și funcționale (la nivelul unei unități economice);

Baze de date specializate, care satisfac cerințele unor anumite sectoare de activitate (baze de date privind sistemul de sănătate, baze de date privind activitatea de cercetare, de învățământ, baze de date multimedia, baze de date GIS, baze de date spațiale etc.).

Din punct de vedere al *modului de organizare, structurare și accesare a datelor*, cele mai cunoscute tipuri de baze de date sunt:

- baze de date de tip rețea;
- baze de date ierarhice;
- baze de date relaționale;
- baze de date orientate pe obiecte.

Toate acestea au caracteristicile modelelor de structurare a datelor corespunzătoare (ierarhice, rețea, relaționale sau pe obiecte), astfel:

Bazele de date de tip rețea au la bază organizarea și structurarea datelor după modelul de tip rețea.

Bazele de date ierarhice se caracterizează prin faptul că elementele componente au relații de subordonare de tip unu la mulți, astfel încât fiecare entitate are în subordine una sau mai multe entități și este subordonată, la rândul ei, unei singure entități superioare, conform modelului ierarhic de organizare a datelor.

Bazele de date relaționale au la bază modele de date relaționale și vor fi tratate în detaliu în capitolele următoare.

Bazele de date orientate pe obiecte, numite și baze de date obiectuale, sunt construite pe modele de date complexe, structurate pe obiect. Conceptul de obiect definește o entitate informațională atât ca structură cât și ca funcții (comportament).

Bazele de date orientate pe obiect reprezintă o abordare mai eficientă față de cele menționate, dar nu le substituie total. Superioritatea acestora poate fi rezumată în următoarele:

- se construiesc o singură dată și pot fi utilizate de nenumărate ori, ceea ce asigură crearea rapidă de aplicații complexe;
- modificările intervenite pe parcurs nu afectează toată logica aplicației, ci doar obiectele respective, asigurând astfel o flexibilitate sporită a întregului sistem.

Din punct de vedere al *gradului de centralizare a datelor*, bazele de date pot fi:

- baze de date centralizate;
- baze de date distribuite.

Bazele de date centralizate sunt colecții de date grupate, atât din punct de vedere fizic cât și logic, într-un punct central unde se asigură prelucrarea integrată a acestora. În cadrul bazelor de date centralizate se poate asigura un înalt grad de protecție și securitate a datelor.

Bazele de date distribuite reprezintă colecții de date care, din punct de vedere fizic, sunt dispersate în diferite puncte (stații) ale unei rețele, iar din punct de vedere logic sunt integrate în cadrul unui sistem informatic, fie prin relații funcționale, fie prin aceea că reprezintă copii ale aceluiași fișier, astfel încât acestea constituie o colecție unică de date. În cadrul autonomiei pe care o au, colecțiile de pe fiecare stație a rețelei pot fi prelucrate, pentru a se realiza aplicații locale, sau pot participa la prelucrări globale care antrenează mai multe stații. Datorită dispersării și autonomiei colecțiilor de date, în cadrul bazelor de date distribuite, nu se poate asigura o bună securitate și protecție a datelor.

După *modul de prelucrare* bazele de date se împart în:

- baze de date operaționale
- baze de date analitice.

Bazele de date operaționale sunt utilizate în principal în scenarii de tip OLTP (On-Line Transaction Processing), când datele sunt introduse și actualizate zilnic. Datele stocate sunt dinamice, acest tip de baze de date, conținutul acestora fiind disponibil la puțin timp după actualizare.

Bazele de date analitice sunt utilizate în scenarii de tip OLAP (On-Line Analytical Processing) în care datele stocate sunt istorice și dependente de timp. Acest tip de date sunt statice, actualizarea lor realizându-se rar.

Dintre toate bazele de date amintite, cele relaționale și cele orientate pe obiect s-au dovedit mult superioare, motiv pentru care acestea au cea mai mare aplicabilitate în practica informatică actuală, SGBD-urile cele mai folosite fiind Oracle, Microsoft SQL Server, MySQL, Visual FoxPro și Microsoft Access.

2.4. Baze de date relaționale (BDR)

2.4.1. Caracteristicile BDR

Modelul relațional asociază unei entități o tabelă bidimensională numită **relație**, în care:

- coloanele tabelului reprezintă *atributele* entității;
- liniile (rândurile) tabelului reprezintă *membrii* entității.

Fiecare coloană are un nume distinct, prima linie fiind destinată amplasării acestor nume de atribute. O linie dintr-o relație se numește *tuplu*. Tuplurile sunt distincte, dublurile nu sunt admise. Numărul de tupluri dintr-o relație reprezintă *cardinalitatea* relației.

La nivelul organizării fizice există următoarele corespondențe:

- o *relație* poate fi asociată unui *fișier*,
- un *tuplu* se asociază cu o *înregistrare*,
- o *coloană* corespunde unui *câmp* din înregistrare.

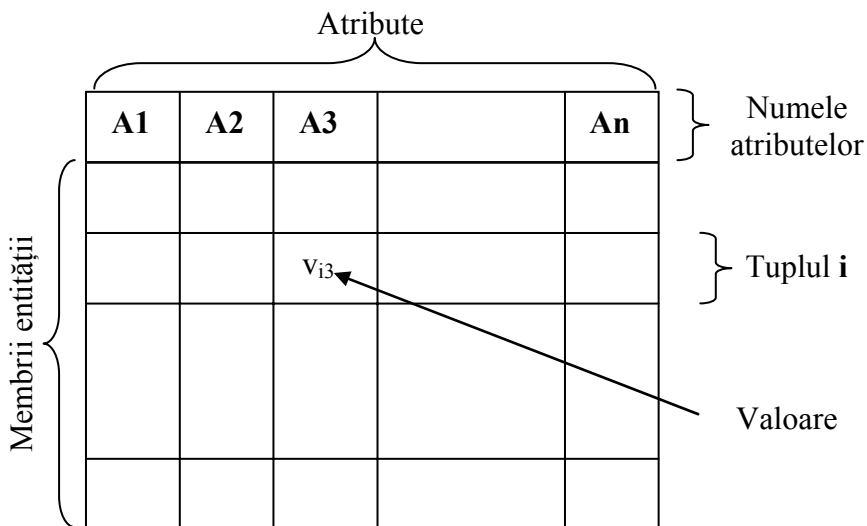


Fig. 2.13. Elementele unei relații (tabel)

La intersecția unui rând cu o coloană se află valoarea câmpului. În cadrul unei relații se poate alege un atribut sau un grup de atribute, care formează **cheia primară** pentru identificarea unică a fiecărui tuplu. Cheia primară obținută prin concatenarea mai multor atribute este necesară atunci când nu există un singur câmp care să identifice unic un tuplu. În cazul în care există mai multe atribute care pot fi definite drept chei de identificare, acestea formează mulțimea *cheilor candidate*, din care se alege cheia primară.

Schema relației reprezintă mulțimea atributelor prin care se descrie o relație, împreună cu domeniile asociate acestora și numele atributelor

Restricțiile de integritate sunt reguli care asigură corectitudinea și coerența datelor:

- unicitatea cheii (UNIQUE)
- referențială - cheie externă (REFERENTIAL)
- integritatea entității (NOT NULL)
- cheie primară (PRIMARY KEY)
- comportament (CHECK)
- dependența datelor (la proiectare)

În legătură cu cheia primară, este necesar a fi respectate două cerințe de integritate:

- a) *integritatea entității* – conform căreia nici un atribut care participă la formarea cheii nu poate avea valori nule;
- b) *integritatea referențială* – conform căreia, dacă într-o relație apare un atribut prin care se face referință la un alt tuplu din relația curentă sau o altă relație, atribut care se numește în acest caz cheie externă, atunci el trebuie să aibă valori valide, care să existe în relația către care face referință; cu alte cuvinte, dacă tuplul t_1 referă un tuplu t_2 atunci acest tuplu trebuie să existe.

Cheia secundară (externă) reprezintă atribut sau grup de atribute dintr-o relație R1 ale cărui valori sunt definite pe același domeniu ca și cheie primară a unei relații R2, care are rolul de a modela asocierea între entităților reprezentate prin R1 și R2

Pentru a ajunge la o schemă de relații și chei primare convenabile pentru reprezentarea corectă a entităților și a legăturilor dintre ele este nevoie de un proces de proiectare care include și optimizarea structurii relațiilor.

Această modalitate de descriere a structurii datelor face mai flexibilă și descrierea legăturilor dintre colecțiile de date, relațiile dintre aceste colecții. Aceste relații se concretizează în:

- câmpuri comune existente în cadrul entităților aflat într-o anumită relație;
- tabele sau fișiere de legături, fișiere index, constituite pe baza valorilor acestor câmpuri comune sau a unor adrese de legătură determinate cu algoritmi specifici.

Indiferent de nivelul la care se stabilesc relațiile între date, atunci când colecțiile sunt supuse prelucrării, relațiile devin operaționale la nivel de atribut și se realizează în funcție de valoarea acestora. Cu ajutorul acestor relații, atât câmpurile cât și înregistrările fișierelor aflate într-o anumită relație pot fi identificate, accesate și prelucrate cu ușurință. După numărul de entități implicate, relațiile pot fi:

- relații binare, în care sunt implicate elementele a două entități;
- relații complexe, în care sunt implicate mai multe entități;
- relații recursive, în care sunt implicate elementele aceleiași entități.

Relațiile binare, la rândul lor, sunt de două tipuri:

- unidirecționale, când presupun existența a două colecții (entități): una principală și alta secundară;

- bidirecționale, care presupun existența a două colecții de aceeași complexitate.

În funcție de numărul de elemente, între care se stabilesc relații, aparținând celor două colecții, aceste relații pot fi de tip unu la unu, unu la mulți și mulți la mulți.

Relațiile de tipul $1 \rightarrow 1$ (unu la unu), care presupune că unui membru din colecția A îi corespunde un singur membru din colecția B, (figura 2.14).

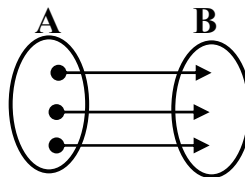


Figura 2.14 Relație de tip 1 la 1

Relațiile de tipul $1 \rightarrow m$ sau $m \rightarrow 1$ (unu la mulți sau mulți la unu), care presupune că unui membru din prima entitate A îi corespund mai mulți membri din a doua entitate B; astfel de relații se mai numesc și relații ierarhice (figura 2.15)..

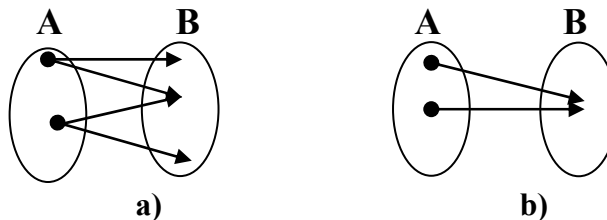


Figura 2.15 Relație de tip 1 la m (a) și m la 1 (b)

Relațiile de tipul $m \rightarrow m$ (mulți la mulți), în care unui membru din entitatea A îi corespund mai multe date din colecția B și mai multor date din colecția A îi corespunde o singură dată din colecția B, (figura 2.16).

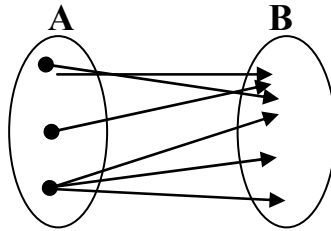


Figura 2.16 Relație de tip m la m

Relații de tip mulți la mulți se mai numesc și relații de tip rețea. O relație mulți la mulți se va descompune întotdeauna în două relații, o relație tip unu la mulți și respectiv o a doua relație de tip mulți la unu prin intermediul unei entități de legătură.

2.4.2 Etapele proiectării BDR

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

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

Odată ce acest proces a fost finalizat se continuă cu:

- proiectarea procedurilor tehnologice, pentru prelucrarea bazei de date;
- elaborarea programelor;
- testarea programelor;
- definitivarea documentației.

Toate aceste activități necesită, pentru proiectele reale complexe, o muncă în echipă pe baza unei metodologii riguroase, cunoscute ca metodologia de analiză și proiectare a sistemelor informatice. În cadrul unui sistem informatic baza de date reprezintă elementul central în jurul căruia se concentrează celelalte componente ale sistemului.

Formularea problemei presupune stabilirea obiectivelor aplicației informatice care asigura actualizarea și exploatarea bazei de