

1 CONCEPTE FUNDAMENTALE

1.1 Introducere

Acum, bazele de date fac parte din viața noastră de zi cu zi în așa măsură, încât adeseori nu suntem conștienți că le utilizăm.

Exemple:

- a) Dacă aveți o **agendă cu adrese**, aceasta este *aproape* o bază de date. Ca să fie recunoscută ca o bază de date propriu-zisă trebuie să copiați informația într-un computer și să o salvați - într-o manieră ordonată – într-un fișier
- b) Atunci când cumpărați ceva de la **supermarket**, vânzătorul trece un cititor de coduri de bare peste fiecare din mărfurile cumpărate. Acesta este legat la un program aplicație pentru baze de date, care utilizează codul de bare pentru a afla prețul articolului respectiv dintr-o bază de date de produse. Apoi programul reduce numărul articolelor respective din stoc și afișează prețul la casă. Dacă nivelul stocului scade sub o anumită limită, este posibil ca sistemul să emită automat o comandă pentru a obține un stoc suplimentar din acel articol.
- c) Acolo unde se pot face cumpărături utilizând **cardul de credit**, de regulă vânzătorul verifică dacă aveți suficient credit. Această verificare poate fi efectuată prin telefon, sau automat printr-un cititor de card legat la un sistem de calculatoare. În ambele cazuri, undeva se află o bază de date, care conține informații despre cumpărăturile efectuate utilizând cardul de credit. Pentru a verifica creditul dvs, există un program aplicație care utilizează numărul cărții de credit pentru a controla dacă prețul bunurilor pe care doriți să le cumpărați, împreună cu totalul cumpărăturilor deja efectuate în luna respectivă, se află în limita creditului dvs. Atunci când cumpărarea este confirmată, detaliile acesteia sunt adăugate în baza de date respectivă. Înainte de a autoriza cumpărarea, programul aplicație va accesa baza de date și pentru a verifica dacă nu cumva cardul de credit se află pe lista celor furate sau pierdute.

1.2 Conceptul de bază de date

1.2.1 Noțiunile de dată și informație

În activitatea noastră suntem permanent confrunțați cu necesitatea de a lua **decizii**. Luarea unei decizii impune existența unor informații referitoare la situația concretă asupra căreia trebuie să decidem. Atenție: noțiunile de *dată* și *informație* au semnificații complet diferite.

Datele sunt fapte culese din viața reală pe bază de observații și măsurători.

Informația este rezultatul interpretării datelor de către un subiect și conferă acestuia capacitatea de a lua decizii.

Datele devin informații numai în momentul interacțiunii lor cu un sistem capabil să le interpreteze. Aceeași dată poate fi interpretată în mod diferit de subiecți diferiți, deci generează informații diferite. Nu există o relație biunivocă între date și informații.

Sistemele de calcul prelucrează date, nu informații.

Datele au un caracter **obiectiv**, în timp ce informația este **subiectivă**. Pentru luarea unei **decizii corecte**, informația trebuie să fie:

1. disponibilă în timp util
2. corectă
3. necontradictorie
4. neredundantă
5. să aibă o formă adecvată necesităților factorului de decizie

Pentru a fi utile, datele trebuie **culese, memorate, organizate, regăsite** și **prelucrate** în mod adecvat pentru utilizarea lor ca informație. O astfel de activitate este legată în informatică de noțiunea de **bază de date**.

Organizarea datelor în vederea prelucrării pe calculator este o activitate tot așa de importantă ca și **realizarea programelor**. Aceste două componente ale unui sistem informatic, datele și programele, se găsesc într-o strânsă interdependență. Un program, oricât de elaborat și rafinat ar fi, nu va produce rezultatele scontate dacă va lucra cu o bază de date depășită sau prost organizată. Similar, o structură de date bine pusă la punct nu va putea suplini carențele programului care o prelucrează.

1.2.2 Sistemul tradițional, bazat pe fișiere

Sistemele bazate pe fișiere au constituit o încercare de început de a computeriza sistemul clasic de îndosariere manuală.

Sistemul bazat pe fișiere este o colecție de **programe aplicație**, care efectuează servicii pentru utilizatorii finali (servicii de tipul *producerea de rapoarte*). **Fiecare program își definește și gestionează propriile date**.

Altfel spus, într-o organizație, în loc să existe un depozit centralizat cu datele operaționale, fiecare departament își stochează și controlează propriile date. De exemplu, departamentul *Contabilitate* stochează detalii referitoare la salariul fiecărui angajat, și anume:

SalariuPersonal (NumărMarcă, Nume, Prenume, Adresa, DataNașterii, Salariu, NumărAsigurare, NumărSecție)

Departamentul *Personal* stochează deasemenea detalii despre angajați:

Personal (NumărMarcă, Nume, Prenume, Adresa, NumărTelefon, Funcție, DataNașterii, NumărAsigurare, NumărSecție)

Se observă că există o frecvență **dublare a datelor** în aceste departamente, fapt adevărat pentru toate sistemele bazate pe fișiere.

Legat de terminologia utilizată în aceste sisteme, trebuie menționat că un **fișier** este un set de **înregistrări** care conțin **date** între care există anumite relații logice. Fiecare înregistrare conține un set de unul sau mai multe **câmpuri** între care există legături logice, **fiecare câmp reprezentând o anumită caracteristică a obiectului modelat**.

Limitările sistemelor bazate pe fișiere:

- a. separarea și izolarea datelor
- b. dublarea datelor
- c. dependența de date
- d. incompatibilitatea fișierelor
- e. interogarea fixă a programelor aplicație, ⇒ proliferarea programelor aplicație

Separarea și izolarea datelor are ca efect faptul că, pentru anumite aplicații, este necesară **prelucrarea și sincronizarea a două sau mai multe fișiere**.

Dublarea datelor implică două dezavantaje:

1. *risipa de spațiu de stocare*
2. *posibila alterare a integrității datelor*: de exemplu, dacă un angajat își schimbă adresa sau este promovat într-o funcție superioară, cu o creștere corespunzătoare de salariu și această schimbare este anunțată numai la Personal, nu și la Contabilitate. Cum pentru cei de la Personal nu există o modalitate automată de a reactualiza datele din fișierele de la Contabilitate, este foarte probabilă apariția unor neconcordanțe.

Dependența de date implică *modificarea tuturor programelor de aplicație* în cazul modificării tipului sau dimensiunii unui câmp de date, activitate ce necesită mult timp și este supusă apariției erorilor.

Legat de **incompatibilitatea fișierelor**, trebuie subliniat că, deoarece *structura fișierelor este încorporată în programele aplicație*, ea este dependentă de limbajul în care sunt scrise acestea. De exemplu, structura unui fișier generat de un program scris în COBOL poate fi diferită de cea a unui generat de un program scris în limbaj C. *Incompatibilitatea fișierelor face dificilă prelucrarea lor simultană, pentru extragerea de rapoarte de date aflate în fișiere diferite.*

Interogarea fixă a programelor aplicație a fost generată de faptul că sistemele bazate pe fișiere sunt în mare măsură *dependente de programatorul de aplicații*. Interogările sau rapoartele necesare trebuie să fie scrise de către acesta. Din acest motiv au apărut două situații:

- ori tipurile de interogări sau rapoarte care puteau fi realizate erau *fixe* și nu exista nici un fel de facilitate de a realiza interogări neplanificate
- ori apărea o *proliferare* a fișierelor și programelor de aplicație la nivelul fiecărei organizații

Evident, în ambele cazuri consecințele erau inacceptabile și era necesară altă soluție.

1.2.3 Tratarea prin baze de date

Toate limitările tratării bazate pe fișiere se datorează următorilor doi factori:

- 1) **Definiția datelor este încorporată în programele aplicație**, în loc de a fi stocată separat și independent
- 2) **Nu există un control al accesului și manipulării datelor**, în afară de cel impus de către programele aplicație

Pentru a face mai eficientă depozitarea datelor, era necesară o nouă tratare. Așa au apărut **bazele de date și sistemele de gestionare a bazelor de date (SGBD)**.

Baza de date	o colecție partajată de date elementare sau structurate , între care există relații logice , proiectată pentru a satisface nevoile informaționale ale unei organizații.
---------------------	---

Pentru a înțelege conceptul de bază de date, să analizăm detaliat definiția dată mai sus. Baza de date este un **depozit de date unic**, care este **definit o singură dată** și este **utilizabil simultan** de mai mulți utilizatori. În loc de a exista fișiere separate cu date redundante, toate datele sunt integrate, cu o dublare minimă. Baza de date nu mai este deținută de un singur departament, ci constituie acum o resursă comună, **partajată**. Baza de date conține nu numai datele operaționale, ci și o descriere a acestora. Din acest motiv ea este definită și ca o **colecție autodescrisă** de înregistrări integrate. Natura autodescriptivă a bazelor de date este cea care produce **independența program – date**.

1.3 Utilitatea și avantajul bazelor de date

Caracteristica principală a aplicațiilor de baze de date constă în faptul că accentul este pus pe operațiile de **memorare** și **regăsire** efectuate asupra unui volum mare de date și mai puțin asupra operațiilor de prelucrare a acestora. Principala operație care apare în aplicațiile de baze de date este **regăsirea** datelor în scopul obținerii de informații din baza de date. Acesta este sensul existenței oricărei baze de date.

O bază de date este creată pentru a fi interogată.

Alături de operația de regăsire, apar mai mult sau mai puțin frecvenat, operații de **memorare** – pentru introducerea de noi date în baza de date, **ștergere** – pentru datele devenite inutile și de **actualizare** a unor date deja existente în baza de date.

Organizarea datelor într-o bază de date constituie o **CENTRALIZARE** a acestora. Aceasta implică existența unui **administrator** al bazei de date (o persoană sau grup de persoane cu atribuții bine definite în **actualizarea și întreținerea** bazei de date).

Centralizarea datelor într-o bază de date prezintă avantajele:

1. **Reducerea redundanței datelor** memorate. În cazul sistemelor bazate pe fișiere, este posibil ca aceleași date să apară de mai multe ori în fișiere diferite, aparținând unor aplicații diferite ⇒ risipă a spațiului de memorie.
2. **Evitarea inconsistenței datelor** memorate: atunci când există mai multe copii ale aceleiași date este posibil ca prin actualizarea numai a unora dintre ele, să avem valori diferite pentru una și aceeași dată. Aceasta atrage după sine inconsistența bazei de date.
3. **Posibilitatea partajării datelor**, se referă nu numai la posibilitatea utilizării în comun a datelor de către mai mulți utilizatori, ci și la posibilitatea de a dezvolta aplicații noi folosind datele deja existente în baza de date.
4. **Posibilitatea aplicării restricțiilor de securitate**: existând controlul centralizat al datelor, se pot introduce verificări de autorizare a accesului la date. Se pot impune restricții diferite pentru fiecare *tip de acces* (regăsire, actualizare, ștergere), pentru fiecare *tip de dată* și la nivelul *fiecărui utilizator*.
5. **Posibilitatea introducerii unor proceduri de validare** a datelor în cazul operațiilor de **actualizare**, pentru a asigura cerința ca baza de date să conțină date corecte.

1.4 Independența datelor

Independența datelor înseamnă că există o **delimitare strictă** între **reprezentarea fizică** a datelor și **imaginea** pe care o are utilizatorul asupra acestor date. Modul concret în care este realizată memorarea și organizarea datelor este **invizibil** pentru utilizator. Acesta trebuie să fie preocupat numai de problema concretă pe care o are de rezolvat și pe care o modelează într-un anumit fel. El lucrează la nivelul bazei de date cu acest model propriu. Detaliile de implementare rămân în sarcina sistemului. Problema independenței datelor prezintă 2 aspecte:

1. **independența fizică** a datelor
 2. **independența logică** a datelor
1. **Independența fizică a datelor** este o măsură a imunității datelor **față de modificările care pot să apară în structura fizică de memorare a datelor**. O modificare a acestei structuri nu va afecta aplicația și reciproc, modificări ale aplicației vor lăsa nealterată structura fizică a datelor. Structura fizică a datelor este **determinantă pentru strategiile de acces** folosite pentru regăsirea datelor. O aplicație care este independentă de structura fizică a datelor nu conține nici o referire la **dispozitivele de memorare** folosite sau la **strategiile de acces** la date. Din punct de vedere al aplicației, datele sunt entități cu nume. Orice referire la date în cadrul aplicației se face prin aceste nume. Detaliile legate de dispozitivele de memorare sau strategiile de acces nici nu trebuie să fie cunoscute de utilizator.
 2. **Independența logică a datelor** se referă la imunitatea sistemului propriu al fiecărui utilizator față de modificările în structura logică globală a bazei de date. Independența logică permite **adăugarea de noi entități** logice (câmpuri sau înregistrări) la structura bazei de date și/sau **modificarea relațiilor** existente între ele, ceea ce asigură **dezvoltarea bazei de date fără a afecta utilizatorii** deja existenți. Este nerezolvabilă

problema eliminării de entități logice din baza de date, întrucât orice operație de acest gen se repercutează asupra utilizatorilor care fac referire la entitatea eliminată.

Observație: Independența logică este mult mai greu de asigurat decât cea fizică.

1.5 Obiectivele fundamentale ale unei baze de date

1. **Centralizarea datelor** permite:
 - Suprimarea redundanței
 - Asigurarea unicității înregistrării
 - Controlul centralizat asupra datelor
2. **Independența între date și program.** Baza de date, ca imagine a unei anumite realități, trebuie actualizată permanent. Acest lucru nu trebuie să afecteze programele aplicație. Pentru aceasta trebuie ca fiecare program să aibă o *viziune proprie* asupra bazei de date.
3. **Realizarea de legături între entitățile de date,** necesare pentru exploatarea eficientă a sistemului informatic. Ex.
4. **Integritatea datelor** asigură fiabilitatea și coerența bazei de date. Pentru aceasta trebuie definite *restricții de integritate* cum ar fi:
 - Apartenența la o listă de valori sau la un interval
 - Apartenența la un anumit format
 - Reguli de coerență cu alte dateAceste reguli trebuie să respecte atât aspectele statice, cât și cele dinamice. De exemplu, după o actualizare de creștere a salariului, nivelul acestuia trebuie să fie mai mare decât cel precedent.
5. **Securitatea datelor.** Baza de date trebuie să fie protejată împotriva unei **distrugerii logice** (anomalii de actualizare) **sau fizice**. Pentru aceasta, există instrumente care permit:
 - Crearea unor *puncte de repriză* = salvarea din timp în timp a unor copii coerente ale bazei de date
 - Gestiunea unui *jurnal de tranzacții* = lista operațiilor realizate asupra bazei de date după ultimul punct de reprizăDacă apare o anomalie care ar putea distruge baza de date, pe baza copieii de la ultima repriză și a jurnalului de tranzacții se poate reface baza de date în forma inițială.
6. **Confidențialitatea datelor** este asigurată prin proceduri de:
 - Identificare a utilizatorilor prin nume sau cod
 - Autentificarea prin parole
 - Autorizarea accesului diferențiat prin drepturi de creare, consultare, modificare sau ștergere pentru anumite sectoare de date
7. **Partajarea datelor** permite înlănțuirea tranzacțiilor solicitate simultan pe aceeași înregistrare din baza de date, prin *blocarea cererilor în așteptare* și *deservirea ulterioară* a acestora.

1.6 Sistemul de Gestiune a Bazelor de Date (SGBD)

SGBD	un sistem de programe care permite utilizatorului definirea, crearea și întreținerea bazei de date precum și accesul controlat la aceasta.
-------------	--

Deci SGBD constă în elemente de software care interacționează cu programele aplicație ale utilizatorului și cu baza de date.

1.6.1 Facilități oferite de SGBD

De obicei, un SGBD oferă următoarele facilități:

- Permite utilizatorilor să **definească baza de date**, de obicei printr-un **limbaj de definire a datelor** (Data Definition Language **DDL**). DDL permite utilizatorului specificarea tipurilor de date și a structurilor
- Permite **inserarea, reactualizarea, ștergerea și extragerea datelor** din baza de date, de obicei printr-un **limbaj de manipulare a datelor** (Data Manipulation Language **DML**).
- Oferă **accesul controlat** la baza de date. De exemplu, poate furniza:
 - Un sistem de **securitate**, care previne accesarea bazei de date de către utilizatori neautorizați
 - Un sistem de **integritate**, care menține concordanța datelor stocate
 - Un sistem de **control al concurenței**, care permite accesul partajat la baza de date
 - Un sistem de **control al refacerii**, care restaurează baza de date după o defecțiune hardware sau software
 - Un **catalog** accesibil utilizatorilor, care conține descrieri ale datelor din baza de date
- Oferă un anumit **nivel de securitate**. Modurile de vizualizare pot fi realizate astfel încât să nu includă datele care nu trebuie cunoscute de anumiți utilizatori
- Oferă un mecanism de **personalizare a aspectului** bazei de date
- Pot prezenta o **imagine coerentă**, neschimbată a structurii bazei de date, chiar dacă baza a fost modificată

Această analiză este generală. Nivelul real de funcționalitate diferă de la produs la produs. De ex, s-ar putea ca un SGBD pentru un PC să nu accepte accesul partajat concurrent, ci să aibă numai un control limitat al securității, integrității și refacerii.

1.6.2 Componentele mediului SGBD

Se pot identifica 5 componente principale ale mediului SGBD: hardware, software, date, proceduri și persoane.

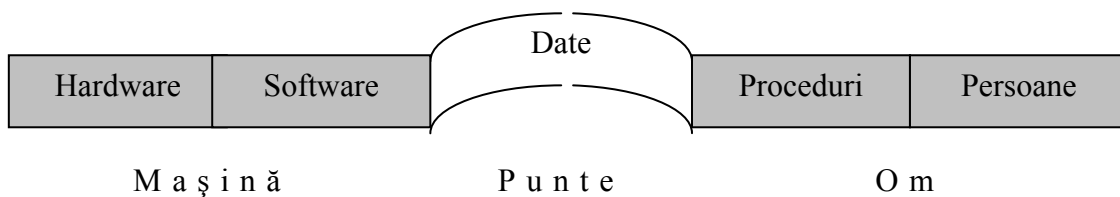


Fig. 1.1 Mediul SGBD

Hardware

Pentru a funcționa, SGBD și aplicațiile au nevoie de hardware. **Elementele specifice de hardware depind de cerințele organizației și de sistemul SGBD utilizat**, și poate fi de la un PC sau un singur mainframe, până la o întreagă rețea de calculatoare.

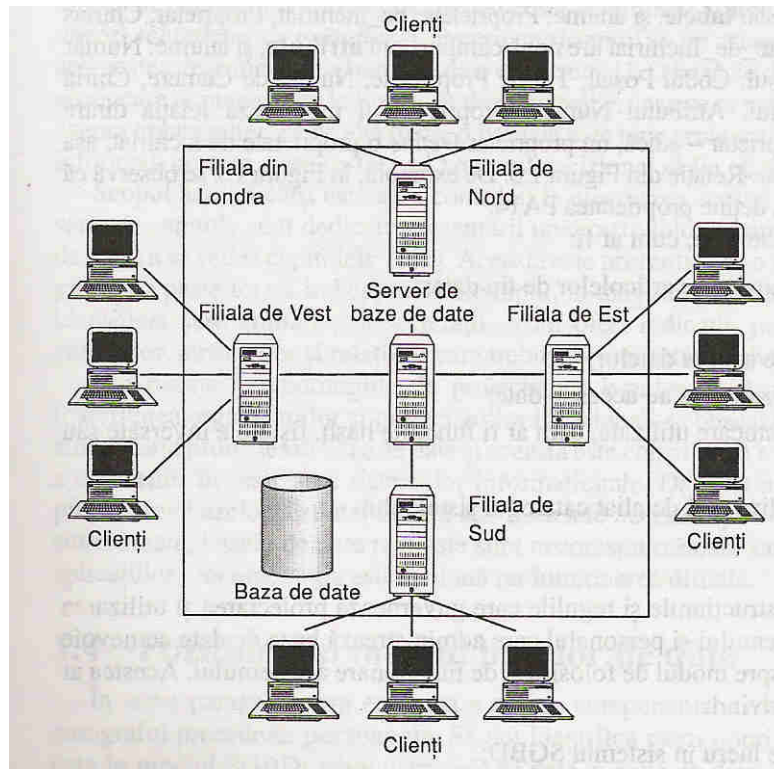


Fig. 1.2 Exemplu de configurație hardware

Software

Componenta software cuprinde **programele SGBD** și **programele de aplicație**, împreună cu **sistemul de operare** și **soft-ul de rețea** dacă SGBD este utilizat în rețea.

Datele

Sunt probabil cea mai importantă componentă a unui mediu SGBD. Datele acționează ca o punte între componentele mașină și cele umane. Baza de date conține atât **datele operaționale** cât și **meta-datele** („datele despre date”).

Procedurile

Se referă la **instrucțiunile** și **regulile** care guvernează proiectarea și utilizarea bazei de date.

Persoanele

Se pot identifica patru tipuri distincte de persoane implicate în SGBD:

- **Administratorii** de date și de baze de date
- **Proiectanții** de baze de date
- **Programatorii** de aplicații
- **Utilizatorii finali**