

## Noțiuni introductive:

- **Bază de date**

= o colecție de informații de preferat relaționate și organizate.

Informația care se dorește a fi stocată constituie *datele*. Organizarea/structura bazei de date este descrisă de *meta-date*. Atât datele cât și meta-datele sunt stocate în baza de date.

- **Schema bazei de date**

= specifică structura bazei de date

- **Sistem de gestiune a bazelor de date (SGBD, echivalent DBMS – Database Management System)**

= un complex format din date, hardware, software și utilizatori care gestionează/actualizează datele. Sistemele de gestiune a bazelor de date sunt în general instalate pe calculatoare de uz general.

**Componenta hardware** trebuie în primul rând să asigure stocarea persistentă a datelor (chiar și în cazul căderii sistemului) și acces rapid la acestea.

**Componenta software** asigură interacțiunea utilizatorilor cu sistemul de baze de date prin intermediul **limbajelor de interogare** și a unor unelte de dezvoltare de aplicații.

Limbajul de interogare îndeplinește două sarcini:

- definirea structurii datelor (meta-datele), această componentă a limbajului fiind denumită *componenta de definire a datelor (DDL – Data Definition Language)*
- regăsirea rapidă și modificarea datelor, *componentă de manipulare (DML – Data Manipulation Language)*

O sarcină esențială a componentei software este managementul tranzacțiilor. O tranzacție reprezintă o secvență de operații executate asupra bazei de date, cu următoarele proprietăți, denumite global cu acronimul ACID :

- **Atomicitate** – operațiile vor fi executate în totalitate, altfel niciuna nu va fi executată
- **Consistență** – o tranzacție aduce o bază de date dintr-o stare consistentă într-o alta stare tot consistentă
- **Izolare** – execuția unei tranzacții nu este influențată de execuția alteia
- **Durabilitate** – efectul execuției tranzacției trebuie să fie de durată (modificarea este una persistentă)

**Utilizatorii unui SGBD sunt:**

- Proiectantul bazei de date;
- Administratorul bazei de date (poate fi și cel care proiectează) care ia decizii asupra naturii datelor care sunt stocate, politica de acces, monitorizarea și ajustarea performanței bazei de date;
- Utilizatorii finali care au drepturi de acces limitate stabilite de administrator și care nu necesită decât cunoștințe tehnice minimale asupra bazei de date.

Din punct de vedere **funcțional arhitectura unui SGBD** constă din:

- Procesorul de interogări - convertește o interogare în instrucțiuni pe care SGBD-ul le procesează eficient făcând apel la metadate.
- Managerul de memorie – obține datele cerute în urma compilării interogărilor de către procesorul de interogări și administrează structurile ce conțin date în conformitate cu directivele DDL
- Managerul de tranzacții – asigură că tranzacțiile satisfac proprietățile ACID descrise mai sus și facilitează recuperarea datelor în cazul căderii sistemului.

**La nivel de aplicație, arhitectura unui SGBD este cea de client-server**

**Arhitectura logică cunoscută drept arhitectura ANSI/SPARC** a fost elaborată la începutul anilor '70 și distinge trei nivele de abstractizare a datelor:

- Nivelul fizic conține informații care descriu cum sunt stocate datele: adrese ale componentelor datelor, lungimi în bytes. SGBD-urile asigură independența datelor, organizarea bazei de date la nivel fizic fiind indiferentă la programele aplicație.
- Nivelul logic descrie datele într-o manieră similară structurilor din C. Detaliile de stocare de nivel fizic sunt eliminate, nivelul logic având un caracter conceptual. Nivelul logic este esențial în înțelegerea datelor stocate și formularea ulterioară de interogări.
- Nivelul utilizator conține perspectiva fiecărui utilizator asupra conținutului bazei de date.

Din perspectivă istorică, se evidențiază următoarele etape în evoluția SGBD-urilor:

- **Modelul ierarhic** de tip arbore dezvoltat de IBM la sfârșitul anilor '60 concretizat în produsul IMS – Information Management System
- **Modelul rețea** apărut în 1971 și propus de CODASYL Database Group
- **Modelul relațional** care constituie fundamentul bazelor de date contemporane. Originile sunt în munca lui Codd de la începutul și mijlocul anilor 70. Dezvoltarea bazelor de date relaționale a început la sfârșitul anilor 70 cu un system experimental dezvoltat la IBM și denumit System R, precursor a versiunilor comerciale SQL/DS și DB2. În anii 80 au apărut multe sisteme relaționale: Oracle, Ingres,... În anii 90 tehnologia relațională a evoluat prin adăugarea ideilor și tehnicilor din programarea orientată obiect.

## • **Modelul relațional**

- Un model simplu, cu implementări foarte eficiente și un limbaj de interogare (SQL) descriptiv simplu și expresiv
- este utilizat de toate sistemele comerciale majore la momentul actual

Elementele modelului relațional:

- Baza de date constă dintr-un set de **tabele/relații**

- Fiecare tabel/relație are un nume unic în baza de date și conține un set de **coloane/atribute/câmpuri**
- Fiecare atribut are un nume unic în tabelul la care aparține și are asociat **tip/domeniu** din care poate lua valori
- Fiecare **linie/înregistrare/uplu** din tabel conține valori pentru toate atributele
- Ordinea liniilor/uplelor și a coloanelor/atributelor nu are semnificație

**Schema** bazei de date relaționale reprezintă descrierea structurală a tabelelor în baza de date.

**Instanța** bazei de date reprezintă conținutul bazei de date la un anumit moment

**NULL** – este o valoare specială utilizată când nu se cunoaște (sau e nedefinită) valoarea unui atribut pentru o înregistrare.

**Cheie candidat** – un atribut care are valori unice pentru fiecare înregistrare

- sau un set de atribute ale căror valori combinate sunt unice, minimală în raport cu această proprietate (nici o submulțime nu e cheie); dacă nu este îndeplinită proprietatea de mulțime minimală avem de a face cu o **supercheie**

**Cheie primară** – una dintre cheile candidat aleasă pentru a identifica în mod unic înregistrările din tabel; nici un atribut implicat nu poate avea valoarea NULL

**Cheie alternativă** – cheie candidat care nu a fost aleasă drept cheie primară

**Cheie străină** – un atribut sau un set de atribute dintr-o relație care are corespondent o cheie candidat a altei relații (sau chiar a relației curente, caz în care o numim cheie străină recursivă)

Obținerea de informații din baza de date se realizează cu ajutorul **interogărilor SQL**. Rezultatul unei interogări este un nou tabel.

- **Algebra relațională**

- Un limbaj formal
- Operatorii de bază din algebra relațională: selecție, proiecție, produs cartezian, intersecție, diferență, redenumire
- Bazată pe mulțimi (duplicata se elimină)

- **SQL (Structured Query Language)**

- limbajul utilizat în manipularea bazelor de date relaționale
- un limbaj esențial declarativ; exprimă *ce* se dorește de la baza de date, nu și *cum* se obține rezultatul

- un limbaj implementat cu fundamente în algebra relațională (semantica este definită cu ajutorul algebrei relaționale), bazat pe multi-seturi (nu se elimină duplicatele)
- deși considerat un limbaj de programare non-procedural, majoritatea SGBD-urilor relaționale oferă și extensii procedurale