

Utilizatorii pot sa efectueze mai multe categorii de operatii: insertare, stergere, actualizare si interogare dupa un criteriu dorit.

Ce este o baza de date?

Este o colectie logica coerenta de date, creata si mentinuta computerizat in scopul prerucrarii datelor in anumite aplicatii. Prelucrarea datelor se refera la operatiile de introducere, stergere, actualizare si interogare a datelor. Ea are un grup de utilizatori si se adreseaza unui anumit grup de aplicatii. Avantaje: controlul centralizat, viteza mare de regasire si actualizare a informatiilor, compacte: volumul ocupat mult mai mic decat documentele scrise, flexibilitate, deci putem modificare structura bazei de date fara sa modificam programul aplicatiilor, posibilitatea introducerii standardelor, ceea ce permite interschimbarea datelor intre organizatii.

Clasificarea sistemelor de baze de date:

1. modelul de date relational:  
se bazeaza pe notiune de relatie din matematica, are o reprezentare usor de inteles si de maipulat, alcatuit dintr-un tabel bidimensional compus din linii si coloane. datele sunt percepute de utilizatori ca tabele
2. modelul de date orientat obiect:  
se bazeaza pe limbaje de programare orientate obiect  
sunt independente de timpul de viata al programerol care le creeaza sau acceseaza  
caracterisiticile importante: abstractizarea, mostenirea, incapsularea, modularizarea, persistenta: asigura memorarea transparenta pe disc obiectelor care alcatuiesc o baza de date orientata obiect.
3. obiect-relationale: combina caracteristicile modelului relational cu car. modelului de date orientat obiect. Reprezinta extinderea modelului relational cu caracterisci ale modelului obiect, care este necesara pentru definirea si prelucrarea tipurilor de date complexe. Datele sunt reprezentate ca tabele, dar mai adauga posibilitatea definirii unor noi tipuri de date.
4. Modelul de date Ierarhic: datele se reprezinta printr-o structra ierarhica de inregistrari (records) conectate cu legaturi (links). Modelul ierarhice se reprezinta printr-un numar oarecare de scheme ierarhice, aceasta fiind un

arbore directionat, avand mai multe niveluri in care nodurile sunt tipurile de inregistrari iar arcele sunt tipurile de legaturi.

5. Modelul de date retea: foloseste o structura de graf; nodurile sunt tipuri de entitati iar muchiile reprezinta asocierile dintre tipurile de entitati.

Doua categorii de sisteme de baze de date:

1. Centralizate: este un sistem de bdd in care datele si sistemul de gestiune sunt stocate pe un singur calculator.
2. Distribuite: poate avea atat datele cat si sistemul de gestiune distribuite pe mai multe calculatoare

Prin protectia si securitatea datelor se intelege totalitatea mijloacelor, metodelor si mecanismelor pentru a preveni distrugerea, modificare sau folosirii neautorizate a informatiei protejate. Se definesc urmatoare concepte de baza:

1. Securitatea datelor: totalitatea masurilor de protectie impotriva distrugerii, modificarii neautorizate sau a divulgarii acestora
2. Caracterul secret: consta in dreptul acestora de a decide ce informatii se pot folosi in comun si in ce conditii
3. Confidentialitatea: se refera la statutul acordat, acesta reprezentand nivelul sau gradul de protectie ce trebuie acordat informatiei
4. Integritatea: restrictia ca sensul datelor sa nu difere fata de cel inscris pe documentul sursa

**La nivelul** sistemelor informatice se pot diferentia aspecte de securitate la nivel fizic, la nivelul sistemului de operare si la nivelul SGBD.

Responsabilul pentru securitatea bazelor de date este administratorul, cine are capabilitati foarte puternice, pe care altii nu au. Adminul poate sa creaza conturi, sa acorde sau sa retrage privilegii, sa stearga conturi, etc. Cine doreste sa conecteze la o baza de date trebuie sa aiba un cont si o parola. Sistemul de gestiune le verifica daca sunt corecte. Alta tehnica de protectie este criptarea datelor, prin care datele sunt criptate cu ajutorul unei algoritmi de criptare. O alta tehnica securizarea a bazei de date pentru aplicatiile web ar fi instalarea unui firewall, care permite accesarea calculatorului pe care este stocata baza de date.

## Curs 2:

Sistem baza de date: asigura crearea, utilizarea, intretinerea bazelor de date.

Componente: hardware, software, utilizatrori, datele persistente

**Hardware:** de obicei bazele de date sunt pe PC -uri standard, dar si pe servere multiprocesor. Performanta sistemului bazelor de date este influentata de performanta calculatoarelor. Pc -ul trebuie sa aiba o capacitate mare de stocare, utilizata pentru memorarea datelor, deoarece intr-un sistem de baze de dae este foarte importanta sa accesam datele rapid. Pentru memorarea acestora se folosesc SSD -uri si HDD -uri. Discurile se folosesc pentru duplicare.

**Software:** Intre baza de date si utilizatorii sistemului exista un sistem de gestiune a bazei de date SGBD. (DataBase Management System). O baza de date poate fi generata cu mai multe programe specifice, sau cu SGBD. **SGBD** -ul este un interpretor de cereri, care primeste cereri de acces la baza de date, le interpreteaza, executa si returneaza rezultatul catre utilizatori. PACHetul software ce asigura manipularea datelor impreuna cu datele insasi formeaza SGBD -ul, asta ica ofera o viziune a datelor si asigura o protectie a datelor si asigura protectia fata de accese neautorizate sau defecte de functionare.

### Utilizatori:

1. **Programatorii:** ei dezvoltă aplicațiile de baze de date. Aplicațiile pot fi desktop sau client-server. Aplicațiile desktop se instalează și rulează pe un anumit calculator. Aplicațiile client-server sunt aplicații instalate pe un calculator numit server și rulează de pe orice calculator aflat în acea rețea. Aceste sunt aplicații web, implementate în PHP sau ASP, iar interfața scrisă în HTML
2. **Utilizatorii:** accesează bazele de date prin aplicații, au drepturi limitate, nu cunosc structura și datele din acea bază de date.
3. **Administratorul** o persoană autorizată care are rolul de administrarea resurselor, autorizarea accesului a cordonării și monitorizării utilizatorilor bazei de date. El mai face și operații periodice de crearea backup -urilor și de refacere a lor când este nevoie.

Datele persistente:

Datele memorate în baze de date sunt datele persistente, adică acele care rămân stocate pe ssd sau hdd-uri. Astea se introduc, se șterg sau se actualizează. Datele devin persistente după ce au fost validate de către SGDB. Datele care ies, ele sunt tot nepersistente, deoarece devin din interogări puse sub formă de view, raport sau afișare.

Arhitectura internă a sistemelor de date:

3 niveluri de abstractizare a datelor: nivel extern (vedere grup utilizator), nivelul conceptual (schema conceptuală), nivelul intern (schema internă). Sub structura bazei de date se înțelege tipul datelor, relațiile dintre ele și restricțiile ce trebuie îndeplinite.

- a) **Nivelul intern:** descrie structura de stocare fizică a datelor. La acest nivel se descriu detaliile ale stocării precum și modul de acces la date.
- b) **Nivelul conceptual:** descrie structura pentru utilizatori. Aici se face o descriere completă a bazei de date, ascunzând detaliile legate de stocarea fizică, concentrându-se descrierii entităților, tipurilor de date, relațiilor dintre ele precum și a restricțiilor asociate.
- c) **Nivel extern:** este nivelul vizual care include o colecție de scheme externe ce descriu baza de date. La acest nivel utilizatorii pot ascunde detaliile de care nu sunt interesate.

**În multe** sgbd-uri nu se pot face o distincție între cele trei nivele. Procesul transferului cererilor și rezultatelor între nivele este numit **mapping**.

**Independența datelor:** poate fi definită în două moduri:

1. Independența logică:

Se referă la capacitatea de a schimba schema conceptuală, fără a atrage după sine modificarea schemei externe. Se poate schimba schema conceptuală prin adăugarea sau ștergerea datelor.

2. Independența fizică:

Se referă la schimbarea schemei interne, fără a modifica schema conceptuală sau externe. Asta poate fi făcut prin reorganizarea fișierelor sau prin crearea unei noi structuri de acces ca să se facă accesul la date cât mai rapid.

## Limbaje SGBD:

**Data definition language** este utilizat de adminul/proiectantul a bazei de date in definirea schemei interne si conceptuale. Compilatorul DDL proceseaza instructiunile pentru a afla detaliile despre constructie si astea le memoreaza in catalogul SGBD.

Limbajul **storage definition language** este utilizat pentru specificarea schemei interne. Legatura intre cele doua nivele de implementare este asigurata de unul din cele doua. Fara specificarea explicita, cand se defineste baza de date se foloseste DDL -ul.

**Limbajul Data manipulation language (DML)** se foloseste pentru inserarea, cautarea, stergerea si modificarea datelor.

### **Structured Query Language SQL:**

DML – comenzile pentru extragere si actualizare

SELECT – extrage date din BD

UPDATE – actualizeaza date

DELETE – sterge inregistrari

INSERT INTO – insereaza date noi

DDL – comenzi pentru creare/modificare baza de date, tabele, indecsi, stabilire legaturi intre tabele, constrangeri:

CREATE DATABASE, ALTER DATABASE, DROP DATABASE

CREATE TABLE, ALTER TABLE, DROP TABLE

CREATE INDEX, DROP INDEX

**DCL data control language:** comenzi pentru controlul accesului utilizatorilor la anumite date: GRANT, REVOKE

TCL Transaction control language: comenzi ptr controlul si procesarea tranzactiilor cu o baza de date: commit, rollback, savepoint

## INTERFETE SGBD:

Sgbd -ul trebuie sa ofere o interfata corespunzatoare tuturor categoriilor de utilizatori. Principalele tipuri de interfere oferite de sgbd sunt:

1. Interfata bazate pe meniuri: Oferă o listă de opțiuni care îl ajută la formularea cererilor.
2. Interfete grafice: aceste interfete afișează utilizatorului o diagramă. Utilizatorul poate formula cererea prin manipularea acestei diagrame. Aceste fiind des combinate cu meniuri.
3. Interfete bazate pe forme: aici utilizatorul poate completa formele cu noile date pe care le dorește să le insereze, sau folosește aceste forme pentru a cere SGBD să delete datele dorite.
4. Interfete în limbaj natural: acceptă cereri scrise în limba engleză sau scrise în alte limbi de circulație internațională. Interpretarea cererilor se face baza unui set de cuvinte cheie. Dacă interpretarea se realizează cu succes, atunci programul de interfață generează cererea de nivel înalt și va fi transmisă către sgbd.
5. Interfete specializate aferente cererilor repetate: sunt destinate pentru a scurta timpul necesar introducerii comenzii sau utilizarea de chei functionale.
6. Interfete pentru administratorii bazelor de date: sunt utilizate în implementarea comenzilor privilegiate ce sunt folosite de adminii bazelor de date. Astfel de comenzi includ crearea de conturi, autorizarea intrării într-un anumit cont, etc.

### Curs 3

Proiectarea unei baze de date consta din proiectarea logica si fizica a acesteia, pentru a corespunde cerintelor utilizatorilor.

1. Analiza cererilor si strangerea de informatii: proiectantii DB analizeaza rezultatele de care potentialii utilizatori ar avea nevoie. De ex: aplicatie de salarizat
2. Proiectarea conceptuala: reprezinta o descriere a datelor utilizatorului si o descriere a tipurilor de date, a relatiilor si restrictiilor acestora.
3. Alegerea unui SGBD: se face in functie de complexitatea aplicatiei, capacitatea de stocare, refacerii datelor, numarul de utilizatori, etc.
4. Proiectarea logica: se poate realiza in 2 sub-faze. Aceste se pot realiza impreuna cu ajutorul SGBD -ului.
5. Proiectarea fizica: reprezinta procesul de alegere a structurilor de memorare si de acces la fisiere pentru a obtine o performanta cat mai buna. Ca parametrii generali se pot enumera: timpul de raspuns, utilizarea spatiului de memorare si capacitatea tranzactionala. Deciziile de proiectare fizica se pot lua numai dupa o analiza a aplicatiilor a interogarilor si tranzactiilor.
6. Implementarea bazei de date si a aplicatiei: aici se creaza obiectele bazei de date, se introduce datele, se verifica constrangerile, se creaza interfetele cu utilizatorul si rapoartele necesare cu datele extrase.

Modelul entitate-relatie: este cel mai utilizat model conceptual care reprezinta schema conceptuala cu ajutorul entitatilor si a relatiilor dintre acestea.

O entitate este un obiect al lumii reale, sau ceva care poate reprezenta un obiect fizic, o activitate, etc. O entitate este un obiect cu existenta fizica. Aceasta are proprietati numite atribute, ce descriu entitatea respectiva. Pentru denumirea de entitate se mai foloseste si “tabel”, iar in loc de atribut se mai zice “campurile tabelului”. Atributelor li se asociaza valori pentru a identifica entitatile. Aceste formeaza o inregistrare a tabelului. Atributele complexe sunt facute din atributele atomice, deci din atributele care nu sunt compuse. Atributele derivate se pot determina din alte atribute, de ex varsta unei persoane: an\_acum- an\_nastere. Daca o entitate nu are valori la toate atributele sale, atunci atributele goale se numim atributele null.

Constructia schemelor relatie:

O relatie este o corespondenta intre entitati. Gradul unei relatii este dat de numarul de multimi asociate. Relatiile pot fi binare, multiple sau unare.

Consideram E1 si E2 doua multimi.

a) Relatiile binare: impartit in 3 categorii:

1. Relatia “unu la unu”:un element din E1 corespunde un singur element din E2 si reciproc, rar folosit in realitate, se poate folosi pentru a reduce numarul de attribute dintr-o entitate.
2. Relatie “unu-la-multe: un elemnt din E1 ii corespunde unul sau mai multe elemente din multimea E2, si unui elemnt ii corespunde mai multe elemnte din multimea E1.
3. Relatia “multe-la-multe” este relatia in care unui elemnt din E1 corespunde unul sau mai multe element din E2 si viceversa. Este foarte des folosita, dar nu poate fi implementata in bazele de date relationale

b) Relatia unara: foloseste o relatie pe o singura entitate, aceasta fiind asociata cu ea insasi.

Diagrama entitate-relatie este modelul entitate-relatie reprezentat prin multimile de entitati si relatii dintre acestea. Exista mai multe variante de notatii pentru redarea acestei diagrame:entitate tip puternica, entitate tip slaba, atribut, tipul relatiilor.

Constrangeri de integritate: sunt reguli care se definesc la proiectarea unei baze de date si care trebuie sa fie respectate de-a lungul existentei acesteia. Se pot clasifica astfel:

1. constrangeri in cadrul tabeli:

asigura integritatea datelor acestora si se pun numai in cadrul unei singure tabele. 3 categorii:

- a) constrangeri de domeniu: sunt conditii care se impun valorilor atributelor pentru a asigura integritatea domeniilor atributelor.
- b) Constrangeri de nuplu: sunt conditii care se impun nuplurilor unei entitati si asigura identificarea a nuplurilor prin intermediul cheilor primare.
- c) Constangeri impuse de dependente de date sunt constrangeri prin care valorile unor attribute determina valorile altor attribute ale aceleiasi entitati.



2. constrangeri între tabele: sunt reguli care se impun între două sau mai multe relații. Cele mai importante sunt constrangerile de integritate referențială, care se realizează prin intermediul cheilor străine.

Constrangeri de domeniu: sunt criteriile impuse valorilor atributelor pentru ca acestea să corespundă semnificației pe care o au în realitatea modelată. Având în vedere că atributele reprezentăm vizual ca pe o coloană, se mai numesc și constrangeri de coloană.

Constrangere NOT NULL. Valoare NULL reprezintă lipsa de informație. Această valoare poate apărea când nu știm informațiile respectivi.

Constrangere DEFAULT este folosită pentru stabilirea unei valori implicite pentru atribut. În cazul în care nu se introduc valori la un atribut, atunci el primește valoare implicită, dacă s-a definit, dacă nu, atunci valoarea NULL. Dacă nu s-a definit o valoare implicită și nici nu se admit valori NULL, da eroare.

Constrangerea CHECK: este o constrangere de verificare. Pentru fiecare atribut se pot adăuga constrangeri de verificare la definirea tabelului.

Constrangerea referitoare la nupluri:

N-uplurile entității trebuie să fie distincte deci într-o entitate nu pot exista mai multe n-upluri care să conțină aceeași combinație de valori pentru fiecare atribut.

Cheia primară: reprezintă unul sau mai multe câmpuri ale tabelului care identifică unic fiecare înregistrare. Proprietăți cheie primară:  
-este unică, este stabilă și nu se admit valori NULL

O cheie străină este o submulțime de atribute ale unei entități  $e_1$  care referă entitatea  $e_2$  și îndeplinește următoarele condiții: atributele cheii străine din  $e_1$  sunt compatibile cu cele ale atributelor cheii din entitatea  $e_2$  și cheia din entitatea  $e_2$  este cheie primară.

## Curs 4

Dependenta functionala defineste relatia dintre un atribut dintr-un tabel si un alt atribut dintr-un alt tabel. Atributele sunt campurile tabelului, deci trebuie sa edem ce campuri depind de alte campuri.

2 categorii de dep functionale:

1. Dependente functionale determinate de cheile tabelului -> nu produc redundanta datelor, nici anomalii de actualizare a relatiei.
2. Dependente functionale in care atributul determinat nu este o cheie a tabelului -> produc redundanta datelor si anomalii de actualizare a tabelului.

Constrangerile de cheie sunt constrangeri implicite si este impus de catre sistemul de gestiune.

Dependente functionale in care atributul nu este o cheie a relatiei, nu este verificat nici impus automat de catre sistemul de gestiune. Astea se pot impune numai procedural.

Normalizare: este un proces in timpul caruia schemele nesatisfacatoare sunt decompuse prin impartire atributelor in tabele cu attribute mai putine.

Tipuri de anomalii evitate prin normalizare:

1. Redundanta datelor: stocarea repetata a datelor
2. Anomalii la update: la actualizarea datelor se poate intampla ca unele copii ale acestora sa ramana cu valorile vechi.
3. anomalii la stergere: dupa stergerea unei inregistrari pot ramane copii sau parti din aceasta
4. Anomalii la inserare: se poate ajunge sa inseram intr-o inregistrare inexistentă

Forma normala ofera proiectantului BD:

1. Un schelet pentru analiza relatiilor bazat pe chei si pe dependenta functionala intre attribute.
2. O serie de teste ce pot elimina tabelele, astfel baza de date poate fi normalizata in orice grad.

Forma normala de ordin 1:

Este parte a definitiei formale a unei tabele. Nu permite attribute cu mai multe valori, attribute compuse sau combinatii ale lor, deci domeniul trebuie sa include numai valori unice. Valorile permise de FN1 sunt atomice sau indivizibile.

Forma normala de ordin 2: impune ca fiecare atribut non-primar sa ie dependent de fiecare parte a cheii principale. O tabela indeplineste fn2, daca:

1. Indeplineste fn1
2. Contine numai attribute care dau informatii despre cheia tablei

Daca tabela are o cheie primara formata din numai un atribut, ea este automat in FN2.

Forma normala de ordin 3: tabelul trebuie sa fie deja in fn2 + toate campurile non-primare sa depinda numai de cheile candidat.

Forma normala Boyce-Codd: versiune mai restrictiva de forma normala 3, astfel toate campurile primare sau non-primare trebuie sa deinde functional de o cheie candidat in FNBC.

Forma normala de ordin 4: Dependenta multivaloare are loc cand prezenta unei inregistrari intr-un tabel implica prezenta a cel putin inca unei inregistrati in acelasi tabel. Ca sa fie in FN4 tabela trebuie sa fie in FNBC si sa nu contina mai mult de o dependenta multivaloare.

Structuri de indecsi:

Un index reprezinta o cale rapida de localizare a inregistrarilor. E utilizat in doua scopuri:

1. Accelerarea cautarilor
2. Asigurarea unicitatii inregistrarilor

Indexul unei tabele este o structura de date aditionala care permite accesul rapid la inregistrarile prin ordonarea acestora.

Clasificarea indecsilor:

Indexul primar: e un index asociat unei tabele ordonate dupa campul cheie al tablei si in structura de index se foloseste campul cheie.

Index secundar: constuit pe baza unu camp cheie, dar tabela nu e ordonata dupa campul cheie.

Indexul de grup e un index facut dupa campuri ce nu sunt campuri cheie, iar tabela poate fi ordonata sau nu relativ la criteriul de acces.

Indexul multinivel: principiul e de a construi niveluri de indexare pana cand structura aditionala corespunde indexului de cel mai mare nivel poate fi memorata intr-un singur bloc.

Indexul primar: e un fisier ordonat cu intregirari de lungime fixa avand doua campuri. Primul camp este de acelasi tip cu un camp cheie ordonat al tebelei, iar al doile camp e un pointer catre un bloc.

Index secundar: se aplica la tabele neordonate, indiferent daca valorile campului sunt distincte sau nu. E un fisier ordonat cu doua campuri, in care primul e identic cu cel al tabelei de date, al doilea camp e un pointer.

Indecsii de grup: se foloseste cand inregistrarile unui tabel sunt ordonate fizic dupa un camp noncheie. E un fisier cu 2 campuri: primul contine aceeasi informatie cu campul noncheie de ordonare, al doilea e un pointer la un block de date.

Indexul multinivel:

Pentru un index multinivel, fisierul index este vazut ca un nou fisier la care se construieste un nou index si asa mai departe. Primul fisier index contine cate o valoare distincta pentru fiecare cheie de indexare.