# Cursul 1-2

**Baza de date** este o colectie de date persistente, care sunt folosite de catre sistemele de aplicatii ale unei anumire „intreprinderi”. ( Ansamblu structurat de date corente, fara redundanta inutila care pot fi prelucrate si accesate concurent ).

**Date intregrate si partajate**:

* Notiunea de **intregrat** se refera la faptul ca baza de date poate fi considerata ca o unificare a mai multor fisiere.
* Notiunea de **partajare** se intelege ca baza de date poate fi partajata concurent intre diferiti utilizatori.

**SGBD** (Data Base Management System) este un produs software care asigura interactiunea cu o baza de date, permitand definirea, consultarea si actualizarea datelor din baza de date. Toate cererile de acces la baza de date sunt tratate si controlate de catre SGBD.

**Dictionarul datelor** (catalog de sistem), structurat si administrat ca o baza de date (metabaza de date), contine „date despre date”, furnizeaza descrierea tututor obiectelor unei baze de date, starea acestor obiecte, diversele constrangeri de securitare si de intregritate etc. Dictionarul poate fi interogat, la fel, ca orice alta baza de date.

*Exista 4 categorii de oameni care sunt implicati in mediul bazelor de date:*

* ***Administratorul de date*** *(DA) care este un manger si nu un tehnician (deci nu scrie cod) si are posibilitatea de a decide ce date pot fi introduse in baza de date si poate stabili diferite reguli de intretinere si de tratare a acestor date dupa ce sunt stocate.*
* ***Proiectantii de baze de date*** *isi pun amprenta pe logica si aspectul fizic al BD.*
* ***Utilizatorii finali*** *sunt cei care acceseaza BD din interfata grafica.*
* ***Programatorii de aplicatii*** *scriu codul necesar pentru implementarea functionalitatilor cerute de utilizatorii finali.*

**Niveluri de abstractizare si de perceptie**: intern, conceptual, logic si extern.

* **Nivelul fizic** (intern) este descris de schema fizica a datelor(bit, octet, adresa);
* **Nivelul conceptual** este descris de schema conceptuala a datelor (viziunea programatorilor de sistem asupra datelor->proiectantii de bd)
* **Nivelul logic** este descris de una din schemele logice (viziunea programtorilor de aplitie asupra datelor)

**Limbaje pentru baze de date:**

* **LDD** (limbaje pentru definirea datelor) este folosita pentru a construi si a modifica structura tabelelor din baza de date. (CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME)
* **LMD** (limbaj de prelucrare a datelor) este folosita la lucrul cu date in baza de date. Cererea SELECT se considera parte din limbajul LMD de aseamenea ( INSERT, UPDATE, DELETE)
* **LCD** (limbaj pentru controlul datelor) se refera la privilegierea utilizatorilor cu drepturi de acces.

**Nivelul conceptual.** Acest nivel corespunde structurii semantice a detelor fara implementarea pe calculator. Schema conceptuala permite definirea tipurilor de date ce caracterizeaza proprietatile elementare ale entitatilor, definirea tipurilor de date compuse care permit regruparea atributelor pentru a descrie entitatile modelului si legaturile intre aceste entitati, definirea refulilor pe care trebuie sa le respecte datele etc.

**Nivelul intern** corespunde struncturii interne de stocare a datelelor. Schema interna permite descrierea datelor unei baze sub forma in care sunt stocate in memoria calculatorului. Sunt definite fisierele care contin aceste date, articolele din fisiere, drumurile de acces la aceste articole etc.

**Nivelul extern** schemele descriu doar o parte din date care reprezinta interes pentru un utilizator sau un grup de utilizatori. Schema externa reprezinta o descriere a unei parti a bazei de date ce corespunde viziunii unui program sau unui utilizator. Un grup de utilizatori va accesa doar datele descrise in schema sa externa.

Un **SMB** (sistem multibaze de date) este alcatuit din mai multe sistem de baze de date privite integrat, in care se construiesc una sau mai multe scheme globale pe baza schemelor fiecarei baze de date componente, astfel incat sa se poata realiza accesul uniform si integrat la fiecare din bazele de date componente.

O baza de date **Data Warehouse**  este o baza de date de dimensiuni foarte mari in care sunt retinute date istorice. Din aceasta baza de date se pot obtine rapoarte cu privire la istoricul anumitor intrari din tabel. Data Warehouse este o baza de date a carei structura este proiectata pentru a facilita analiza datelor.

Data Warehouse si sistemele OLTP sunt supuse unor cerinte diferite, dintre care cele mai semnificative se refera la operatii, actualizarea datelor, proiectare, operatii tipice si date istorice.

* **Operatii.** Depozitele sunt create pentru a permite interogari ad hoc. Ele trebuie sa fie suficient de flexibile pentru a putea raspunde interogarilor spontane ale utilizatorilor. Sistemele OLTP suporta numai operatii predefinite. Aplicatiile pot fi optimizate sau create special numai pentru acele operatii.
* **Actualizarea datelor.** Utilizatorii finali ai unui depozit de date nu fac in mod direct actualizari ale depozitului. In sistemele OLTP, utilizatorii realizeaza, de obicei, in mod individual procedurile de modificare a bazei de date. In acest fel, baza de date OLTP este intotdeauna la zi si reflecta starea curenta a fiecarei tranzitii.
* **Proiectare**. Data Warehouse folosesc in mod uzual, scheme denormalizate, in timp ce sistemele OLTP folosesc scheme normalizate pentru a optimiza performantele operatiilor.
* **Operatii tipice**. O simpla interogare a depozitului de date poate scana mii sau chiar miliane de inregistrari , in timp ce o operatie tipica OLTP acceseaza doar o parte mai mica din inregistrari.
* **Date istorice**.Data Warehouse stocheaza datele pe o perioada lunga de timp, luni sau ani. Acest lucru ofera suport pentru analiza istorica a informatiei. Sistemele OLTP retin date istorice atat timp cat este necesar pentru a indeplini cu succes cerintele tranzactiilor curente.

Data Warehoouse, care cuprinde de obicei informatii despre o intreaga companie poate fi subimpartita in baze de date departamentale, numite rafturi de date **(Data Marts).**

**Arhitecturi multi-user pentru SGBD**

**Teleprocesare** este arhitectura traditionala, ce cuprinde un calculator cu o singura unitate CPU si un numar de terminale care sunt incapabile sa functioneze singure. Terminalele trimit mesaje si programele aplicatie ale utilizatorilor, care la randul lor, utilizeaza serviciile SGBD.

**Arhitectura fisier-server** presupune deja ca procesarea este distribuita in retea (deobicei LAN). Arhitectura cuprinde fisierele cerute de aplicatii si SGBD-ul. Aplicatiile si functiile SGBD sunt executate pe fiecare statie de lucru, solicitand cand este nevoie fisiere de pe server-ul de fisiere.

**Arhitectura client-server** se refera la modul in care interactioneaza componentele software pentru a forma un sistem. Exista un proces client, care cere resurse si un proces server, care ofera resurse.

**Middleware** este un strat intre aplicatia postului client si server-ul de baze de date, constituit dintr-o interfata de programare a aplicatiilor si un protocol de retea.

**Modelul three tier** care este format din:

* Un strat de interfata cu utilizatorul (client)
* Un strat de logica a afacerii si prelucrare a datelor (server de aplicatii)
* Un sistem SGBD distribuit pe calculatoare diferite

**Arhitectura multitier a sistemului Oracle 9i**

Arhitectura multitier contine urmatoarele elemente:

* Unul sau mai multi clienti care initiaza operatii
* Unul sau mai multe servere de aplicatii care executa parti ale operatiilor
* Un server de baze de date care stocheaza datele folosite de operatii

Clientul, care poate fi un browser Web sau un proces user, initiaza o cerere pentru a executa o operatie referioate la informatiile stocate in baza de date.

Serverul de aplicatii contituie interfata dintre client si serverul bazei de date, asigurand accesul la informatii. De asemenea, el include un nivel aditional pentru securitate. Serverul de aplicatii isi asuma identitatea clientului, atunci cand executa, pe serverul de baze de date, operatiile solicitate de acesta.

Serverul de baze de date pune la dispozitia serverului de aplicatii informatiile necesare pentru solutionarea operatiilor lansate de catre client. De asemenea, acesta face distinctia intre operatiile pe care server-ul de aplicatii le cere in favoarea client-ului si cele pe care le solicita in nume propriu.

# Curs 3-6 -> completari

Un **model** este o reprezentare a obiectelor si evenimentelor lumii reale si a asocierilor dintre ele.

Un **model de date** reprezinta o colectie integrata de concepte necesare descrierii:

* Datelor
* Relatiilor dintre ele
* Constrangerilor existente asupra datelor sistemului real analizat

Modelul trebuie sa asigure conceptele de baza care permit proiectantului bazei de date si utilizatorilor sa comunica, fara ambiguitati, cunostintele lor privind functionarea si organizarea modelului real analizat. Prin urmare, un model de date trebuie sa reprezinte datele si sa le faca intelese.

Modelul entitate-relatie(E/R) este cel mai cunoscut abordarii ale modelarii semantice.

**Diagramele E/R** constituie o tehnica de reprezentare a structurii logice a bazei de date, intr-o maniera grafica. Aceste diagrame ofera un mijloc simplu si inteligibil de comunicare a caracteristicilor importante ale designului unei anumite baze de date.

Diagrama E/R este un model de date conceptual de nivel inalt, independent de platforma hardware utilizata si de tipul SGBD-ului.

Modelul E/R imparte elementele unui sistem real in doua categorii: entitati si relatii. Entitatile si legaturile au anumite caracteristici, numite atribute.

# Curs 3-6 design

**Cheia primara** este un identificator unic in cadrul entitatii, facand distinctie intre valori diferite ale acesteia.

* Trebuie sa fie unica si cunoscuta la orice moment
* Trebuie sa fie controlata de administratorul bazei
* Trebuie sa nu contina informatii descriptive, sa fie simpla, fara ambiguitati
* Sa fie stabila
* Sa fie familiara utilizatorului

**Relatie** reprezinta o comunicare intre doua sau mai multe entitati. Existenta unei relatii este subordonata existentei entitatilor pe care le leaga.

**Atribut** – proprietate descriptiva a unei entitati sau a unei relatii.

**Entitate dependenta** – nu poate exista in mod independent. Cheia primara a unei entitati dependente include cheia primara a sursei si cel putin o descriere a entitatii. Entitatea dependenta se deseneaza prin dreptunghiuri cu linii mai subtiri.

**Subentitate** (subclasa) – submultime a unei alte entitati, numita superentitate (superclasa). Subentitatea se deseneaza prin dreptunghiuri incluse in superentitate. Exista o relatie intre o subentitate si o superentitate numita ISA, care este cardinalitatea maxima 1:1 si minima 1:0.

Tabelul **vizualizare** (view) constituie un filtru relativ la unul sau mai multe tabele, care contine numai informatia necesara unei anumite abordari sau aplicatii.

**Reguli de integritate** -> asertiuni pe care datele continute in baza de date trebuie sa le satisfaca.

* Una dintre cheile candidat va fi aleasa pentru a identifica efectiv tupluri si ea va primi numele de **cheie primara**. Cheia primara nu poate fi reactualizata. Atributele care reprezinta cheia primara sunt fie subliniate, fie urmate de semnul #.
* O cheie indentifica linii si este diferita de un index care localizeaza liniile. O **cheie secundara** este folosita ca index pentru a accesa tupluri. Un grup de atribute din cadul unei relatii care contine o cheie a relatiei poarta numele de supercheie.

**Reguli de integritate structurala:**

* **Regula 1** – unicitatea cheii. Cheia primara trebuie sa fie unica si minimala.
* **Regula 2** – integritatea entitatii. Atributele cheii primare trebuie sa fie diferite de valoarea null.
* **Regula 3** – intregritatea referirii. O cheie externa trebuie sa fie ori null in intregim, ori sa corespunda unei valori a cheii primare asociate.

# Curs 7-11

**Algebra relationala** este o multime de operatii formale actionand asupra unor relatii si avand ca rezultat alte relatii.

**Calculul relational**  reprezinta o adaptare a calculului predicatelor la domeniul bazelor de date relationale.

**Operatorul PROJECT**

Proiectia este o operatie unara care elimina anumite atribute ale unei relatii producand o submultime „pe verticala” a acesteia. Suprimarea unor atribute poate avea ca efect aparitia unor tupluri duplicate, care trebuie eliminate.

**Operatorul SELECT**

Selectia (restrictia) este o operatie unara care produce o submultime pe „orizontala” a unei relatii R. Aceasta submultime se obtine prin extragerea tuplurilor din R care satisfac o conditie specificata.

**Operatorul UNION**

Reuniunea a doua relatii R si S ese multimea tuplurilor apartinand fie lui R, fie lui S, fie ambelor relatii.

**Operatorul DIFFERENCE**

Diferenta a doua relatii R si S este multimea tupluilor care apartin lui R, dar nu apartin lui S. Diferenta este o operatie binara necomutativa care permite obtinerea tuplurilor ce apar numai intr-o relatie.

**Operatorul INTERSECT**

Intersectia a doua relatii R si S este multimea tuplurilor care apartin si lui R si lui S. Operatoul INTERSECT este un operator binar, comutatic, derivat.

**Operatorul PRODUCT**

Fie R si S realtii de aritate m, respectiv n. Produsul cartezian al lui R cu S este multimea tuplurilor de aritate m+n unde primele m componente formeaza un tuplu in R, iar ultimile n componente formeaza un tuplu in S.

**Operatorul DIVISION**

Diviziunea este o operatie binara care defineste o relatie ce contine valorile atributelor dintr-o relatie care apar **in toate** valorile atributelor din cealalta relatie.

**Operatorul JOIN**

Operatorul de compunere (uniune) permite regasirea informatiei din mai multe relatii corelate. Operatorul combina produsul cartezian, selectia si proiectia.

**Operatorul NATURAL JOIN**

Operatorul de compunere naturala (NATURAL JOIN) combina tupluri din doua relatii R si S, cu conditia ca atributele comune sa aiba valori identice.

**Operatorul TETHA-JOIN**

Operatorul TETHA-JOIN combina tupluri din doua relatii (nu neaparat corelate) cu conditia ca valorile atributelor specificate sa satisfaca o anumita conditie specificata explicit in cadrul operatiei.

**Operatorul SEMI-JOIN**

Operatorul SEMI-JOIN conserva atributele unei singure relatii participante la compunere si este utilizat cand nu sunt necesare toate atributele compunerii.

**Operatorul OUTER JOIN**

Operatia de compunere externa combina tupluri din doua relatii pentru care sunt satisfacute conditiile de corelare.

**Evaluarea si optimizarea interogarilor**

**Procesarea interogarilor**

O **expresie** a algebrei relationale este constituita din relatii legate prin operatii din algebra relationala. O expresie se poatea reprezenta grafic cu ajutorul unui arbore, numit **arbore algebric**, in care nodurile corespund operatorilor din cadrul expresiei respective.

In majoritatea sistemelor de gestiune, in special in cele relationale, interfata cu utilizatorul este de tip **neprocedural**. Utilizatorul defineste datele pe care doreste sa le vizualizeze fara a da algoritmii de acces.

Un SGBD este **minimal relational**  daca:

* Toate datele din cadrul baze sunt reprezentate prin valori in tabele
* Nu exista pointeri observabili de catre utilizator
* Sistemul suporta operatorii relationali de proiectie, selectie si compunere naturala, fara limitari impuse din considente interne.

Un SGBD este **complet relational** daca este minimal relational si satisface in plus conditiile:

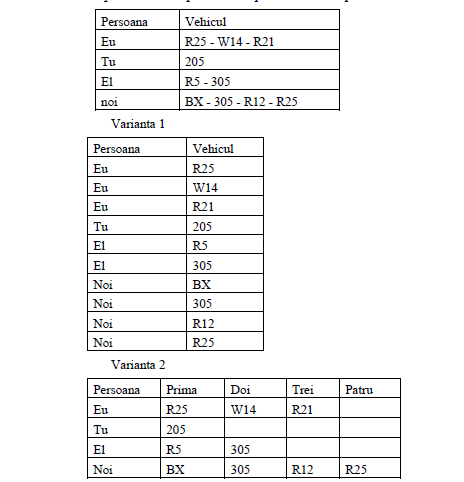
* Sistemul suporta restrictiile de integritate de baza
* Sistemul suporta toate operatiile de baza ale alegebrei relationale.

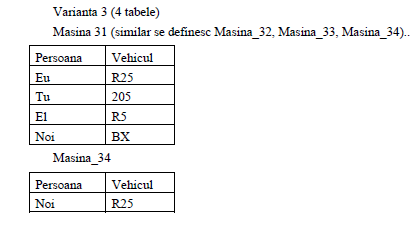
**Normalizarea Relatiilor**

Tehnica normalizarii permite obtinerea unei scheme conceptuale rafinate printr-un proces de ameliorare progresiva a unei scheme conceptuale initiale a bazei de date relationale. Dupa fiecare etapa de ameliorare, relatiile bazei de date ating un anumit grad de perfectiune, deci se afla intr-o anumita forma normala. Trecerea unei relatii dintr-o forma normala in alta, presupune eliminarea unui anumit tip de dependente nedorite, care sunt transformate in dependente admisibile, adica dependente care nu provoaca anomalii.

**Forma normala 1**

O relatie este in prima forma normala daca fiecarui atribut care o compune ii corespunde o valoare indivizibila (atomica).





Cu alte cuvinte o baza de te se afla in prima forma normala daca toate atributele trebuie sa fie atomice, adica sa contina o singura informatie.

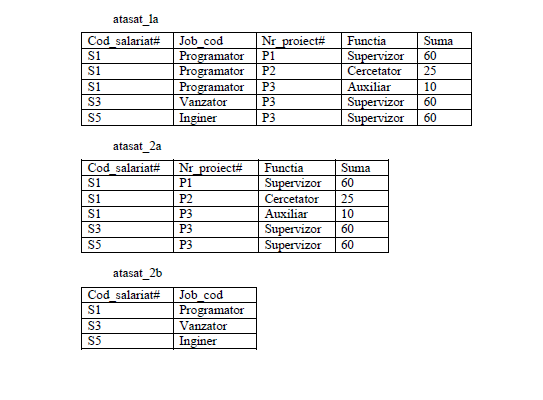
**Forma normala 2**

O relatie R este in a doua forma normala daca si numai daca:

* Relatia R este in FN1
* Fiecare atribut care nu este cheie (nu participa la cheia primara) este dependent de intreaga cheie primara.

O entitate se gaseste in a doua forma normala daca si numai daca se gaseste in prima forma normala si in plus orice atribut care nu face parte din UID va depinde de intregul UID nu doar de o parte a acestuia.

Atunci cand un atribut nu depinde de cheia primara. In cazul asta Job\_cod nu depinde de Cod\_salariat si Nr\_proiect deci poate fi scos din tabel si pun in altul.

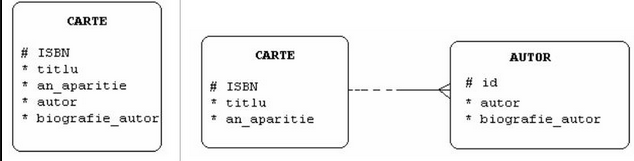


**Forma normala 3**

Intuitiv, o relatie R este in a treia forma normala daca si numai daca:

* Relatia R este in FN2
* Fiecare atribut care nu este cheie (nu participa la o cheie) depinde direct de cheia primara.

O entitate se gaseste in a treia forma normala daca si numai daca se gaseste in a doua forma normala si in plus nici un atribut care nu este parte a ID-ului nu depinde de un alt atribut non-ID.



In cazul asta biografie\_autor nu are ce cauta in entitatea CARTE pentru ea face parte din entitatea AUTOR.

**Forma normala Boyce-Codd (BCNF)**

Relatia R este in FNCB daca si numai daca ifecare determinant e o cheie candidat.

**Forma normala 4**

m:n

Dependenta: situatia in care valoarea unui atribut determina o multime de valori altui atribut.

# Curs 12 – SQL –LDD

**SQL** (Structured Query Language) este un limbaj universal care poate fi utlizat pentru a defini, interoga, reactualiza si gestiona baze de date relationale.

O comanda CREATE TABLE poate contine optional clauza TABLESPACE, care specifica spatiul tabel in care va fi stocat tabelul. De asemenea, poate contine optional clauza STORAGE care este folosita pentru setarea parametrilor de stocare prin intermediul carora e specifica marimea si modul de alocare a extinderilor segmentului tabel.

Optiunea ON DELETE CASCADE specifica ca suprimarea oricarui domeniu de carte din tabelul domeniu este autorizata si implica suprimarea automara a tuturor cartilor din domeniul respectiv care se gasesc in tabelul carte.

**Constrangerea** este un mecanism care asigura ca valorile unei coloane sau a unei multimi de coloane satisfac o conditie declarata.

Constrangerea **CHECK** la nivel de tabel poate compara coloane intre ele, poate face referinta la una sau mai multe coloane, dar nu poate contine subcereri.

Cheile externe se folosesc de urmatoarele cuvinte cheie:

* FOREIGN KEY este utilizat intr-o constrangere la nivel de tabel pentru a defini coloana din tabelul „copil”.
* REFERENCES identifica tabelul „parinte” si coloana corespunzatoare din acest tabel.
* ON DELETE CASCADE determina ca, odata cu stergerea unei linii din tabelul „parinte”, sa fie sterse si liniile dependente din tabelul „copil”.
* ON DELETE SET NULL determina modificarea automata a valorilor cheii externe la valoarea null, atunci cand se sterge valoare „parinte”.

**Modificare structurii unui tabel**

Comanda care realizeaza modificarea structurii tabelului, dar nu modificarea continutului acestuia, este ALTER TABLE.

Optiunea **SET UNUSED** permite marcarea uneia sau mai multor coloane ca fiind nefolosite, cu scopul de a fi sterse atunci cand necesitatile sistemului impun acest lucru.

Clauza CASCADE CONSTRAINTS permite suprimarea tuturor constrangerilor de integritate referentiala corespunzatoare cheilor primare si unice din tabelul supus stergerii.

**Tabele temporare**

Optiunea GLOBAL TEMPORARY permite crearea unui tabel temporar, al carui scop este de a stoca date specifice unei sesiuni. Aceste date sunt stocate in tabel numai pe durata unei tranzatii sau a intregii sesiuni.

**Indecsi**

Un index este un obiect al schemei unei baze de date care:

* Creste viteza de executie a cererilor
* Garanteaza ca o coloana contine valori unice.

**Secvente**

O secventa este un obiect in baza de date care serveste pentru a genera intregi unici in sistemele multi-utilizator, evitand aparitia conflictelor si a blocarii.

Clauza USING specifica numele serviciului corespunator bazei de date distante. Daca se precizeaza doar numele bazei, sistemul adauga implicit numele domeniului. Prin urmare, daca numele domeniului baze distante este diferit de numele domeniului bazei curente, trebuie specificat intreg numele serviciului.

Clauza CONNECT TO permite activarea conexiunii la baza distanta. Cu ajutorul optiunii CURRENT\_USER se poate crea o legatura a utilizatorului curent.

Clauza IDENTIFIED BY specifica numele utlizatorului si parola pentru conexiunea la baza de date distanta, creand o legatura a unui utlizator fixat.

**Vizualizari**

Vizulizarea (view) este un tabel logic (virtual) relativ la date din una sau mai mult tabele sau vizualizari. Vizualizarea este definita plecand de la o cerere a limbajului de interogare a datelor, mostenind caracteristicile obiecteleor la care se refera. Vizualizarea, fiind virtuala, nu solicita o alocare de memorie pentru date. Ea este difinita in DD cu acealeasi caracteristici ca si un tabel.

**Grupari**

Clusterul este o regrupare fizica a doua sau mai multe tabele, relativ la una sau mai multe coloane, cu scopul maririi performantelor. Coloanele comune definesc cheia cluster-ului

Un cluster este un obiect al bazei care necesita:

* Un nume unic la nivelul schemei
* Specificare a coloanelor care compun cheia cluster-ului
* Specificare a spatiului de stocare
* Un index