
Baze de Date

Ioana Ciuciu

ioana.ciuciu@ubbcluj.ro

<http://www.cs.ubbcluj.ro/~oana/>

Obiectivele cursului

- ▶ Cunoasterea **conceptelor fundamentale** referitoare la bazele de date
- ▶ Aprofundarea **modelului relational** de descriere a datelor
- ▶ Exprimarea nevoilor unei aplicatii cu ajutorul unui **formalism conceptual**
- ▶ Traducerea acestor nevoi in vederea conceperii unei **BD relationale**
- ▶ **Gestionarea** (creare, modificare) bazelor de date relaționale în SQL Server
- ▶ Elaborarea de **interogări SQL** complexe pentru analiza datelor
- ▶ **Optimizarea** interogărilor SQL



Obiectivele cursului

▶ Laborator

- ▶ Proiectarea si implementarea in SQL Server a unei baze de date relationale care stocheaza date dintr-un domeniu de aplicare concret
- ▶ Analiza datelor folosind interogari SQL



Planificare

*Sunt posibile mici
modificari ale planificarii in
timpul semestrului*

Saptamana	Curs	Seminar	Laborator
S1	1. Concepte fundamentale ale bazelor de date. Modelare conceptuala	1. Modelul Entitate-Relatie. Modelul relational	1. Modelarea unei BD in modelul ER si implementarea ei in SQL Server
S2	2. Modelul relational de organizare a bazelor de date. Modelare conceptuala		
S3	3. Gestiunea bazelor de date relationale cu limbajul SQL (DDL)	2. Limbajul SQL – definirea si actualizarea datelor	2. Interogari SQL
S4	4. Gestiunea bazelor de date relationale cu limbajul SQL (DML)		
S5-6	5-6. Dependente functionale, forme normale	3. Limbajul SQL – regasirea datelor	3. Interogari SQL avansate
S7	7. JDBC (Java Database Connectivity)	4. Proceduri stocate	4. Proceduri stocate. View. Trigger
S8	8. Interogarea bazelor de date relationale cu operatori din algebra relationala		
S9	9. Structura fizica a bazelor de date relationale	5. View-uri. Functii definite de utilizator. Trigger	
S10-11	10-11. Indeksi. Arbori B. Fisiere cu acces direct	6. Formele normale ale unei relatii. Indeksi	
S12	12. Evaluarea interogarilor in bazele de date relationale		
S13	13. Extensii ale modelului relational si baze de date NoSQL	7. Probleme	Examen practic
S14	14. Aplicatii		

Alte informatii

▶ Evaluare

- ▶ Examen scris – 50% din nota finala
- ▶ Laborator – 25% din nota finala
- ▶ Examen practic (ultimul laborator, SI3-SI4) – 25% din nota finala

- ▶ Criteriu minim de promovare: minim nota 5 la examenul scris, examenul practic si temele de laborator

▶ Prezenta

- ▶ Seminar: obligatorie in proportie de minim 75% (1 absenta nemotivata)
- ▶ Laborator: obligatorie in proportie de minim 90% (1 absenta nemotivata)

▶ Suport de curs

- ▶ Team BD Mate-Info
- ▶ Cod de acces la team: **3qiooec**

▶ Suport seminar / laborator

- ▶ Fiecare semigrupa va avea cate un Team alocat / seminar / laborator (detalii la responsabilii de seminar & laborator)



Persoane de Contact

- ▶ Ioana Ciuciu (ioana.ciuciu@ubbcluj.ro)

Responsabila de curs, seminar, laborator

- ▶ Paul Rus (ruspauladrian.1994@gmail.com)

Responsabil de seminar & laborator



Baze de Date

Curs 1. Concepte fundamentale ale bazelor
de date. Modelare conceptuala



Plan

- ▶ Istoric
- ▶ Baza de date (BD)
- ▶ Sistem de gestiune a bazelor de date (SGBD)
- ▶ Functiile unui SGBD
- ▶ Ciclul de viata al unei BD
- ▶ Arhitectura unui SGBD
- ▶ Modele de descriere a datelor
- ▶ Modelare conceptuala. Modelul Entitate-Relatie



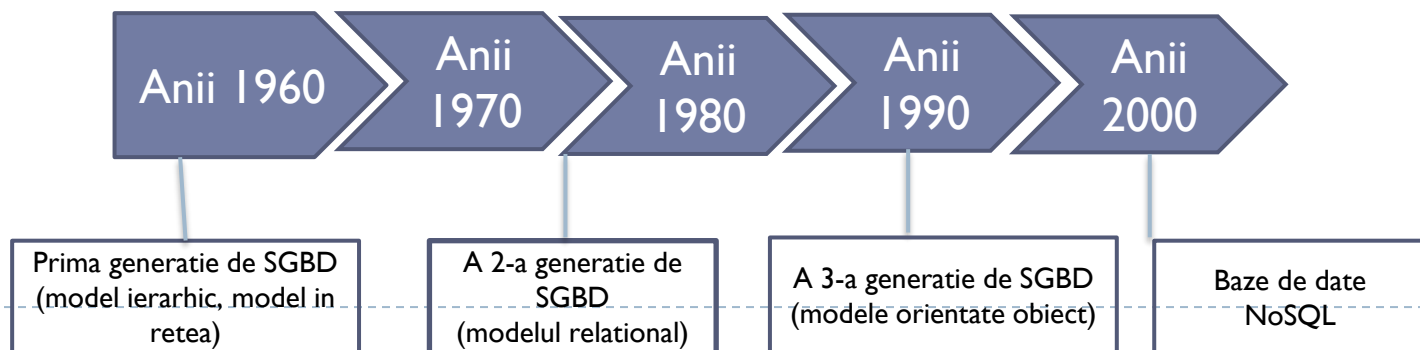
Istoric

- ▶ Inca de la inceput, organizatiile au cautat sa isi **structureze informatiile**
- ▶ La inceput, **informatiile** erau sub forma de **fise**, clasate in ordine alfabetica, cronologica, etc.
 - ▶ Acestea exista inca
- ▶ Apoi, cu ajutorul suportului informatic, **informatiile** au inceput sa fie memorate pe **suport magnetic**
- ▶ Acestea trebuie **organizate** pentru a se putea lucra cu ele (regasirea unei anume informatii, vizualizarea valorii ei, stergerea, actualizarea, etc.), de unde apare notiunea de ***fisier***



Istoric (cont.)

- ▶ **Pana in anii 60:** organizare clasica in fisiere, gestionata de catre sistemele de gestiune a fisierelor
- ▶ **Anii 60:** prima generatie de SGBD (Sistem de Gestiune a Bazelor de Date)
 - ▶ La inceputul aparitiei bazelor de date, nivelul conceptual este strans legat de reprezentarea datelor pe suport fizic
 - ▶ Model ierarhic, model in retea
- ▶ **Anii 70-80:** a 2a generatie de SGBD, mai independente de suportul fizic
 - ▶ Modelul relational
- ▶ **Inceputul anilor 90:** a 3a generatie de SGBD
 - ▶ Modele orientate obiect, ...
- ▶ **Anii 2000:** baze de date NoSQL
 - ▶ Date semi-structurate/nestructurate, de volum din ce in ce mai mare



Componentele unei aplicatii. Metode de memorare a datelor

- ▶ **Componentele unei aplicatii**
 - ▶ **Datele** – memorate in fisiere sau in baze de date
 - ▶ **Algoritm** de gestiune
 - ▶ **Interfata** cu utilizatorul
- ▶ **Metode de memorare a datelor**
 - ▶ Fisiere
 - ▶ Baze de date
 - ▶ Baze de date distribuite



Metode de memorare a datelor – caracteristici fisiere

- ▶ Inconveniente ale utilizarii fisierelor:



Metode de memorare a datelor – caracteristici fisiere

- ▶ Inconveniente ale utilizarii fisierelor:
 - ▶ diverse **formate**
 - ▶ **redundanta** (unele date se memoreaza în mai multe fisiere) ⇒ inconsistenta
 - ▶ descrierea în **program** a **operatiilor de citire/scriere** ⇒ greutate la dezvoltarea unui program (prin schimbarea structurii fisierelor trebuie modificat programul)
 - ▶ dificil de obtinut datele care îndeplinesc anumite **conditii** (pentru simplificare au aparut diverse organizari ale fisierelor)
 - ▶ complexitatea **actualizarilor** datelor (stergeri de înregistrari, modificarea unor valori din înregistrari)
 - ▶ verificarea prin program a unor restrictii de integritate (corectitudine)
 - ▶ lipsa procedurilor de **securitate** si **integritate**
 - ▶ nu se poate controla **accesul concurent** la date



Metode de memorare a datelor – caracteristici fisiere

- ▶ Inconveniente ale utilizarii fisierelor:
 - ▶ diverse **formate**
 - ▶ **redundanta** (unele date se memoreaza în mai multe fisiere) ⇒ inconsistenta
 - ▶ descrierea în **program** a **operatiilor de citire/scriere** ⇒ greutate la dezvoltarea unui program (prin schimbarea structurii fisierelor trebuie modificat programul)
 - ▶ dificil de obtinut datele care îndeplinesc anumite **conditii** (pentru simplificare au aparut diverse organizari ale fisierelor)
 - ▶ complexitatea **actualizarilor** datelor (stergeri de înregistrari, modificarea unor valori din înregistrari)
 - ▶ verificarea prin program a unor restrictii de integritate (corectitudine)
 - ▶ lipsa procedurilor de **securitate** si **integritate**
 - ▶ nu se poate controla **accesul concurent** la date
- ▶ Fisierelor: utile pentru programe care necesita putine date si sunt folosite de un singur utilizator



Baza de date (BD)

- ▶ Este o colectie formata din urmatoarele componente:
 - ▶ **descrierea structurilor de date** folosite (schema bazei de date) pentru modelarea datelor (memorate în dictionarul bazei de date)
 - ▶ **colectia de date** (instante ale schemei)
 - ▶ **componente diverse**: view-uri, proceduri si functii, utilizatori, roluri, etc.

Apare **separarea** între:

- ▶ **definirea** datelor (pastrata într-un dictionar al bazei de date),
- ▶ **gestiunea** datelor (adaugare, stergere, modificare date) si interogare



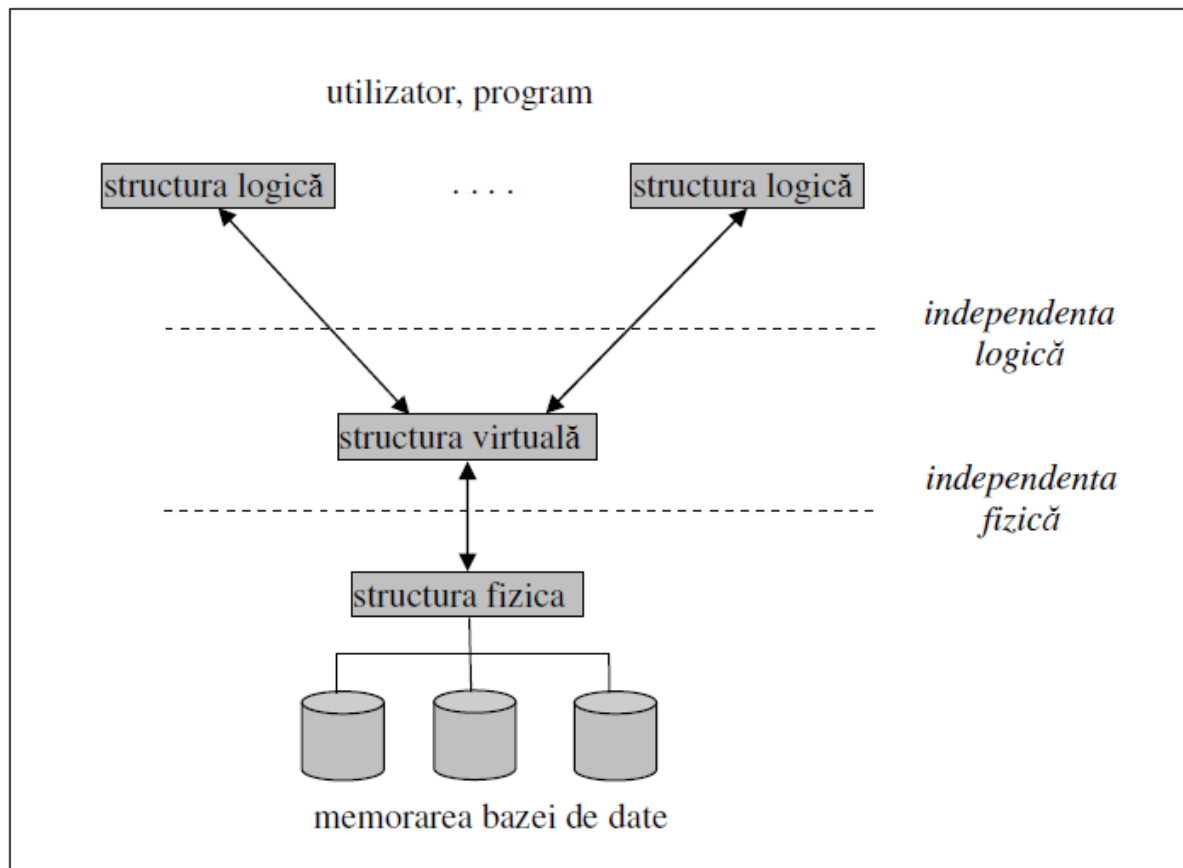
Structurile unei baze de date

- ▶ Arhitectura ANSI-SPARC (1975) – trei niveluri de descriere a datelor:
 - ▶ **structuri logice (structuri externe):** descrierea structurilor de date folosite de un utilizator/program particular. Aceasta descriere se face într-un anumit model de organizare, iar SGBD poate să regasească datele în structura virtuală
 - ▶ **structura virtuală (structura conceptuală, schema bazei de date):** descrie toate structurile de date și restricțiile folosite în baza de date, independent de aplicațiile individuale și de modul în care datele sunt stocate
 - ▶ **structura fizică (structura internă):** descrierea structurilor de memorare a bazei de date (fișiere de date, indexuri, etc.)



Structurile unei baze de date

- Pentru o bază de date se pot defini mai multe structuri (niveluri) logice (externe), o structura virtuala (conceptuala) și o structura fizica (interna)



Independenta logica si independenta fizica

- ▶ Fiecare nivel de abstractizare este protejat la modificarile in structura din nivelul inferior
- ▶ Independența datelor poate fi:
 - ▶ **Independența logica:** posibilitatea schimbarii structurii virtuale fara schimbarea structurii logice, deci a **programelor** care folosesc date din baza de date
Important: dezvoltarea in etape a aplicatiilor
 - ▶ **Independența fizica:** posibilitatea schimbarii structurii fizice fara schimbarea structurilor virtuale si a structurilor logice (deci a programelor care folosesc date din baza de date)
Important: se pot adauga/elimina fisiere (de date, index) pentru optimizarea cautarii, programele utilizatorilor nu consulta direct fisierele (structura fizica)
- ▶ Nivelurile de abstractizare asigura independenta datelor, autorizeaza manipularea datelor, garanteaza integritatea datelor si optimizeaza accesul la date



Tipuri de utilizatori ai bazei de date

- ▶ **administratorii** bazei de date
- ▶ **proiectantii bazei de date** (designers): responsabili cu schema bazei de date, restrictiile definite, functiile si procedurile puse la dispozitia utilizatorilor, optimizarile necesare
- ▶ **utilizatorii unor aplicatii** care gestioneaza date
- ▶ **programatori de aplicatii**

Aplicatiile sunt scrise în diverse limbaje sau medii de programare (aplicatii web, java, .net, C, etc).

Pentru efectuarea unei operatii asupra bazei de date se trimite din aplicatie la sistemul de baze de date o comanda, aici se executa comanda si se trimite raspunsul la aplicatie (tehnologia client/server).

Cererea este trimisa în **limbajul SQL** (Structured Query Language), limbaj standard pentru acces la bazele de date.



Sistem de gestiune a bazelor de date (SGBD)

- ▶ **SGBD** (Sistem de Gestiune a Bazei de Date): colectie de programe necesare pentru gestiunea bazei de date
- ▶ **Sistem de baze de date**: baza de date + SGBD
- ▶ Exemple de SGBD:
 - ▶ Oracle, DB2 (IBM), Microsoft SQL Server, Sybase (SAP), Informix (IBM), Teradata, ...
 - ▶ MySQL, PostgreSQL, ...
 - ▶ Access (Microsoft), Paradox, Foxpro, ...

https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems



Funcțiile unui SGBD

- ▶ **definirea** bazei de date: limbaj de definire (sau aplicații dedicate care generează comenzi în limbajul de definire)
- ▶ **gestiunea** datelor: adăugare, modificare, ștergere, interogare
- ▶ **administrarea** bazei de date: autorizarea controlului la baza de date, monitorizarea folosirii bazei de date, urmărirea performanțelor bazei de date, optimizarea performanțelor bazei de date, etc.
- ▶ **protecția** informațiilor din baza de date: *confidențialitatea* (protecție împotriva accesului neautorizat la date), *integritatea* (protecție împotriva alterării conținutului bazei de date)

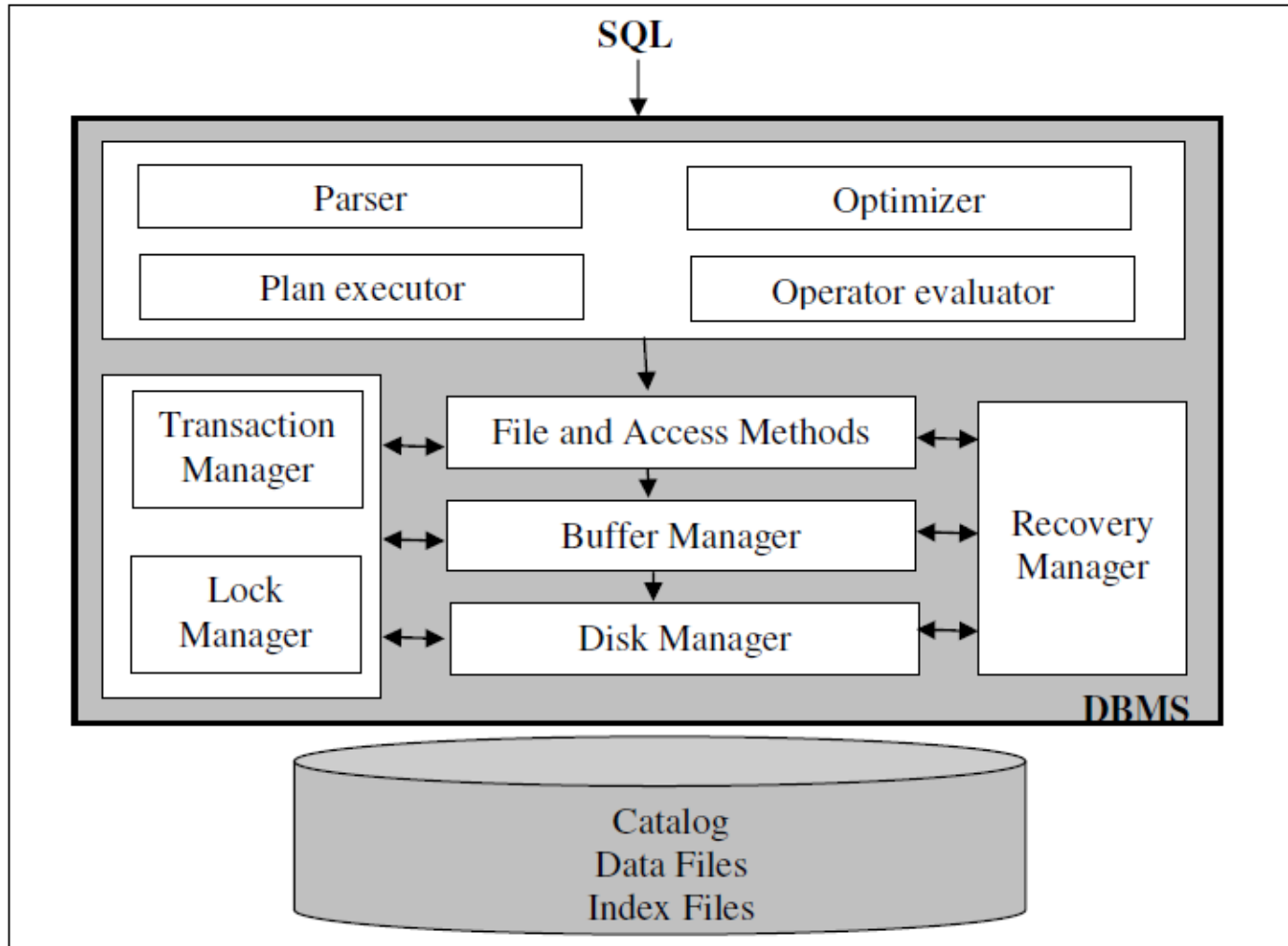


Ciclul de viata al unei baze de date

1. Domeniu din lumea reala
2. Design (analiza--> model): proiectantul, administratorul BD
3. Crearea structurii BD: proiectantul, administratorul BD
4. Crearea BD (DDL si inserari initiale): administratorul BD
5. Optimizarea BD (indecsi, etc.): administratorul BD
6. Manipularea (DML): utilizator
7. Mentenanta BD (interogari specifice): administratorul BD



Arhitectura unui SGBD



Avantajele utilizarii unui SGBD

- ▶ permite **gestiunea** unor **colectii mari de date** cu diverse **legaturi** între date
- ▶ aplicatiile nu gestioneaza detaliile de implementare a bazei de date: ele trimit o comanda SQL si primesc un raspuns (evaluarea comenzii se face de SGBD, folosind diverse programe de acces la date). Legatura dintre aplicatie si baza de date (conexiunea) este temporara
- ▶ **acces** rapid la **date**, care sunt **actuale** si **corecte** (se verifica automat unele restrictii de integritate)
- ▶ **controlul accesului** la baza de date (pentru utilizatori cu diverse roluri)
- ▶ gestiunea **accesului concurent**
- ▶ optimizarea accesului la date (util pentru colectii de date de dimensiune mare)
- ▶ permite **accesul la date din aplicatii** dezvoltate în diverse limbaje sau medii de programare
- ▶ ofera posibilitatea de **import/export** a datelor în diverse formate
- ▶ face posibila **recuperarea la erori** (fisier arhiva + fisier jurnal)
- ▶ permite **dezvoltarea în etape** a sistemelor informatice (modificarea schemei bazei de date, modificarea aplicatiilor, dezvoltarea de noi aplicatii)
- ▶ ofera **instrumente de analiza a datelor** (depozite de date, data mining)



Modele de descriere a datelor

- ▶ Pentru ca datele sa poata fi gestionate automat este necesar sa fie descrise conform unui model
- ▶ Un **model de descriere a datelor** este o multime de **concepte si reguli** folosite pentru modelarea datelor.
- ▶ Aceste concepte permit
 - ▶ descrierea *structurii* datelor,
 - ▶ precizarea *restrictiilor de consistenta* (corectitudine) si
 - ▶ a *relatiilor* cu alte date
- ▶ In cadrul unui model, pentru a descrie o anumita colectie de date (memorata într-o baza de date), se folosesc anumite structuri de date, care constituie **schema** bazei de date (sablonul, sau structura datelor). Datele din colectie respecta schema si pot sa fie considerate **instante** (sau **realizari**) ale schemei



Modele de descriere a datelor

- ▶ **Modele la nivel conceptual:**

- ▶ entitate-relatie (ER)
- ▶ relational
- ▶ retea
- ▶ ierarhic
- ▶ orientat obiect
- ▶ NoSQL
- ▶ semistructurat (XML), RDF (semantic web)

- ▶ **Modele fizice** (se precizeaza detalii cu privire la memorarea bazelor de date în fișiere de diverse tipuri)

- ▶ **Modele pentru implementare:** intermediar între precedentele două (în majoritatea sistemelor se folosește modelul relational)



Modelul Entitate-Relatie (ER)

- ▶ **Entitatea** - este o **data** si reprezinta un obiect din lumea reala (din domeniul de interes)
- ▶ Daca entitatea (data) este complexa (de exemplu un student, sau o disciplina din facultate), atunci pentru precizarea ei se vor folosi anumite **attribute** (sau proprietati)
- ▶ Multimea entitatilor cu aceeasi structura (de ex. multimea studenilor) sunt **instante** ale unui **tip de entitate**, sau **clasa**, sau **schema entitatii**



Modelul Entitate-Relatie (ER) – cont.

- ▶ **Entitatea** - este o **data** si reprezinta un obiect din lumea reala (din domeniul de interes)
 - ▶ In precizarea tipului de entitate (a clasei) fiecare **atribut** va avea:
 - ▶ nume,
 - ▶ domeniu pentru valorile posibile si
 - ▶ eventuale conditii pentru a se verifica daca valoarea este corecta
 - ▶ Un tip de entitate va avea (cel putin)
 - ▶ un nume si
 - ▶ o lista de attribute
 - ▶ La un tip de entitate se poate defini o **cheie** (care este o restrictie): o multime de attribute care iau valori distincte în instanele acestui tip de entitate (i.e. nu pot exista doua instante cu aceeasi valoare a cheii)



Modelul Entitate-Relatie (ER) – cont.

- ▶ **Relatia** - este o **data** si precizeaza o **legatura** (asociere) între doua sau mai multe entitati
 - ▶ La aceasta asociere se pot folosi si attribute suplimentare
 - ▶ Toate relatiile cu aceeasi structura (legaturi între entitati de aceleasi tipuri) trebuie descrise printr-un **tip de relatie** (nume, tipurile de entitati folosite în asociere), sau **schema relatiei**
- ▶ **Schema modelului** este formata dintr-o multime de **tipuri de entitate** si **tipuri de relatie**

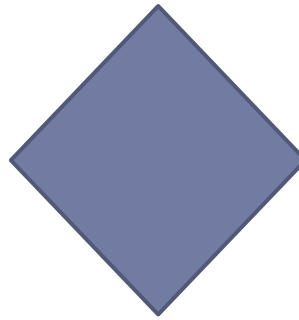


Modelul Entitate-Relatie (ER) – cont.

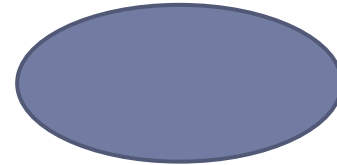
► Diagrama Entitate-Relatie – elemente componente



Tip de entitate



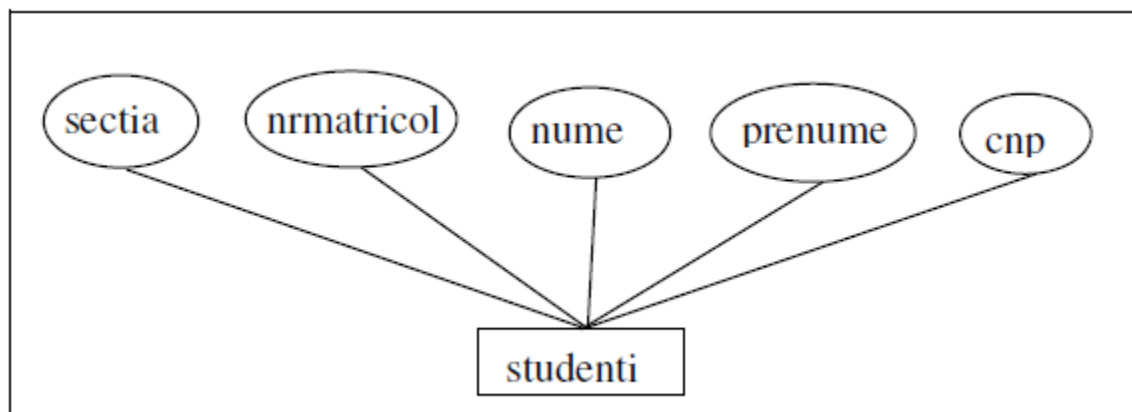
Tip de relație



Atribut

Modelul Entitate-Relatie (ER) – cont.

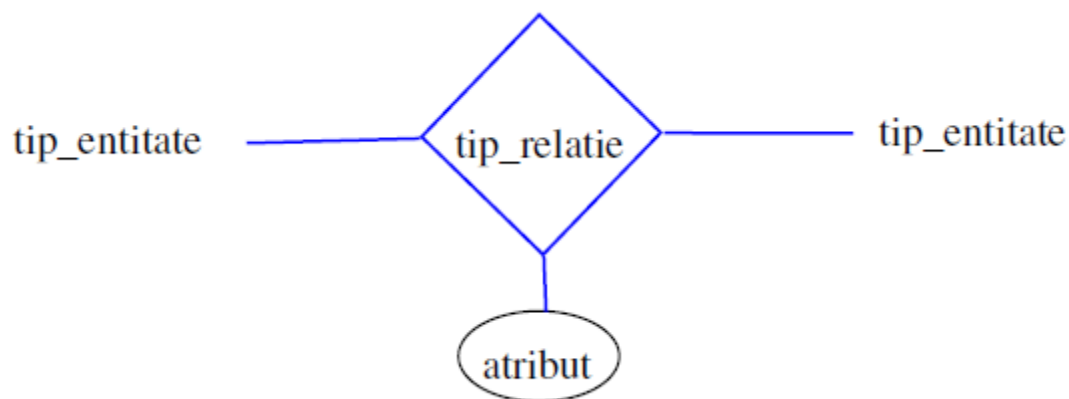
► Tipul de entitate si attributele asociate



sau:

studenti
sectia
nrmatricol
nume
prenume
cnp

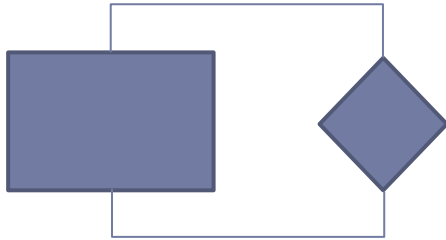
► Tipuri de relatii



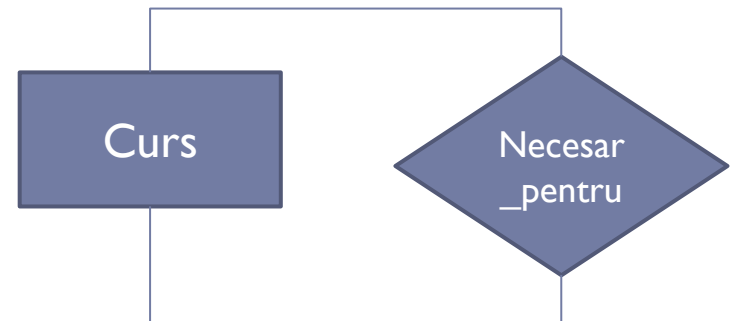
Modelul Entitate-Relatie (ER) – cont.

► Gradul relatiilor

► Relatie unara:



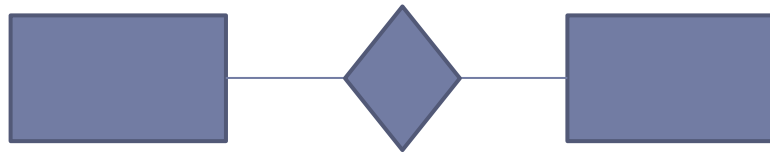
Exemplu:



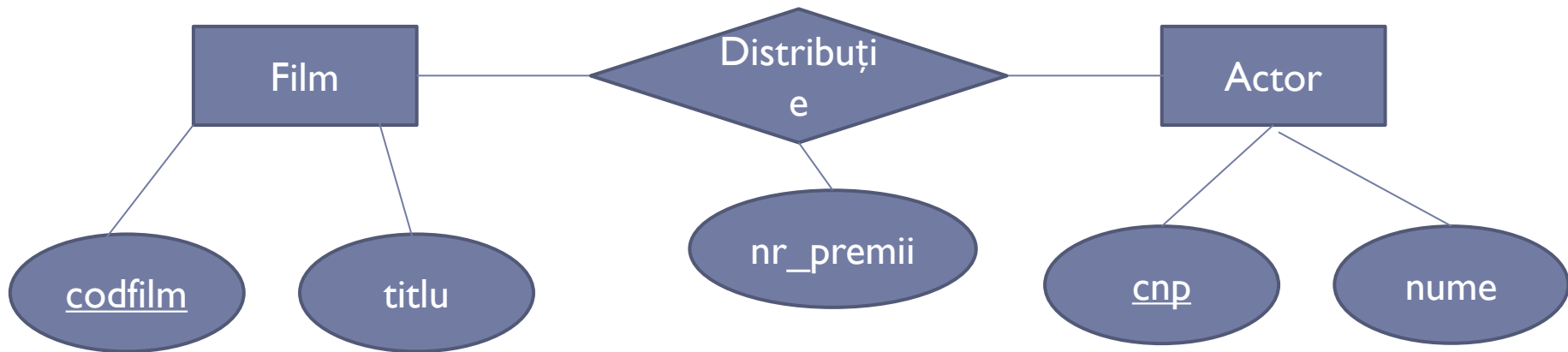
Modelul Entitate-Relatie (ER) – cont.

► Gradul relatiilor

► Relatie binara:



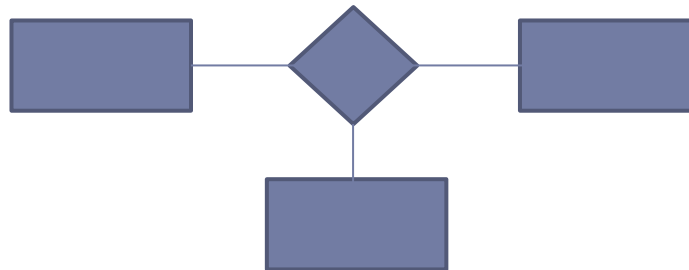
► Exemplu:



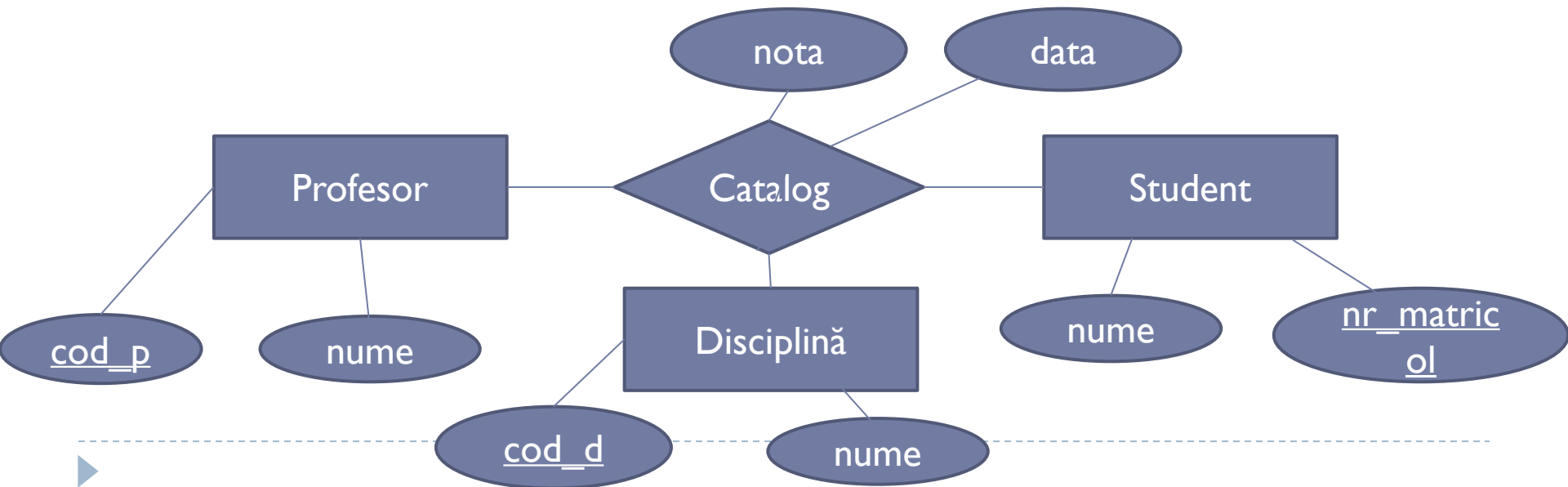
Modelul Entitate-Relatie (ER) – cont.

► Gradul relatiilor

► Relatie ternara:



► Exemplu:



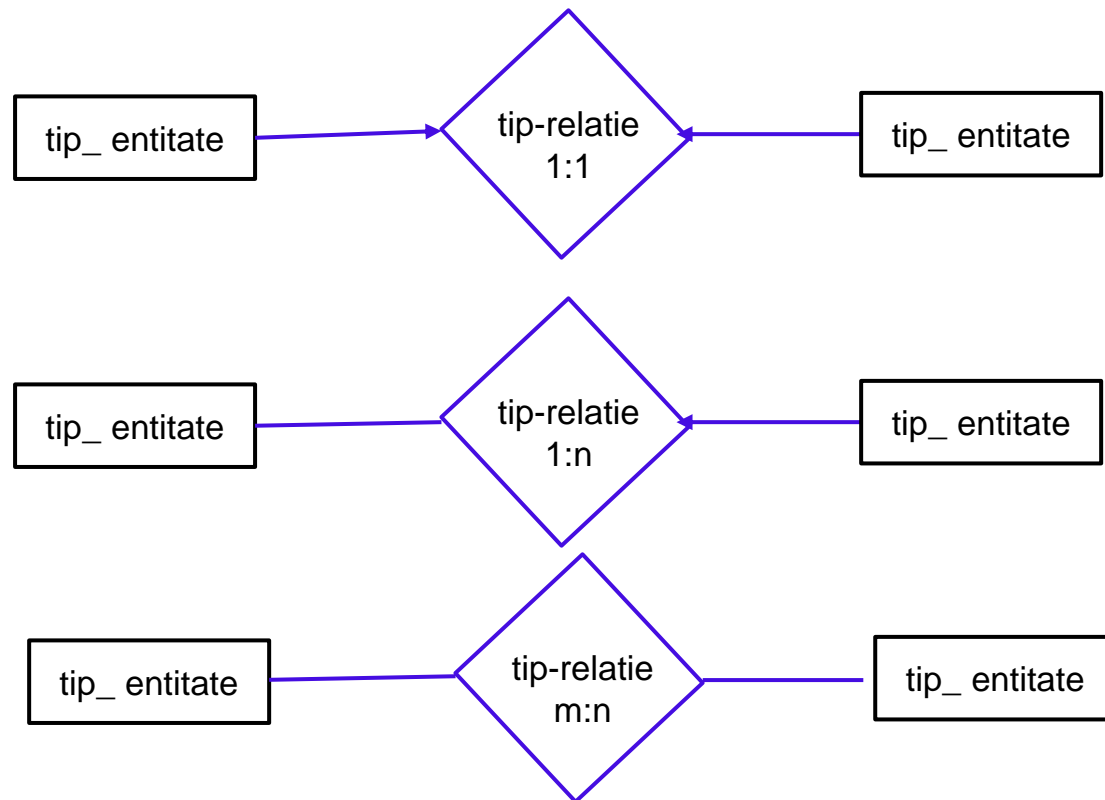
Modelul Entitate-Relatie (ER) – cont.

- ▶ Pentru **relatiile binare** (între doua tipuri de entitate T1 si T2) se pot defini urmatoarele **tipuri particulare de relatie**, care pot fi considerate **restrictii** pentru baza de date (sistemul trebuie sa verifice, la fiecare modificare a bazei de date, daca relatia este de tipul precizat):
 - ▶ **relatie 1:1**: daca o entitate de tipul T1 se asociaza cu cel mult o entitate de tipul T2, iar o entitate de tipul T2 se asociaza cu cel mult o entitate de tipul T1
De exemplu: asocierea dintre **grupa** si **cadru didactic** (pentru a preciza tutorii grupelor)
 - ▶ **relatie 1:n**: daca o entitate de tipul T1 se asociaza cu oricâte entitati de tipul T2, iar o entitate de tipul T2 se asociaza cu cel mult o entitate de tipul T1
De exemplu: asocierea dintre **grupa** si **studenti** (pentru a preciza componenta grupelor)
 - ▶ **relatie m:n** daca o entitate de tipul T1 se asociaza cu oricate entitati de tipul T2, iar o entitate de tipul T2 se asociaza cu oricate entitati de tipul de tipul T1
De exemplu: asocierea dintre **discipline** si **studenti** (pentru a preciza contractele de studiu)



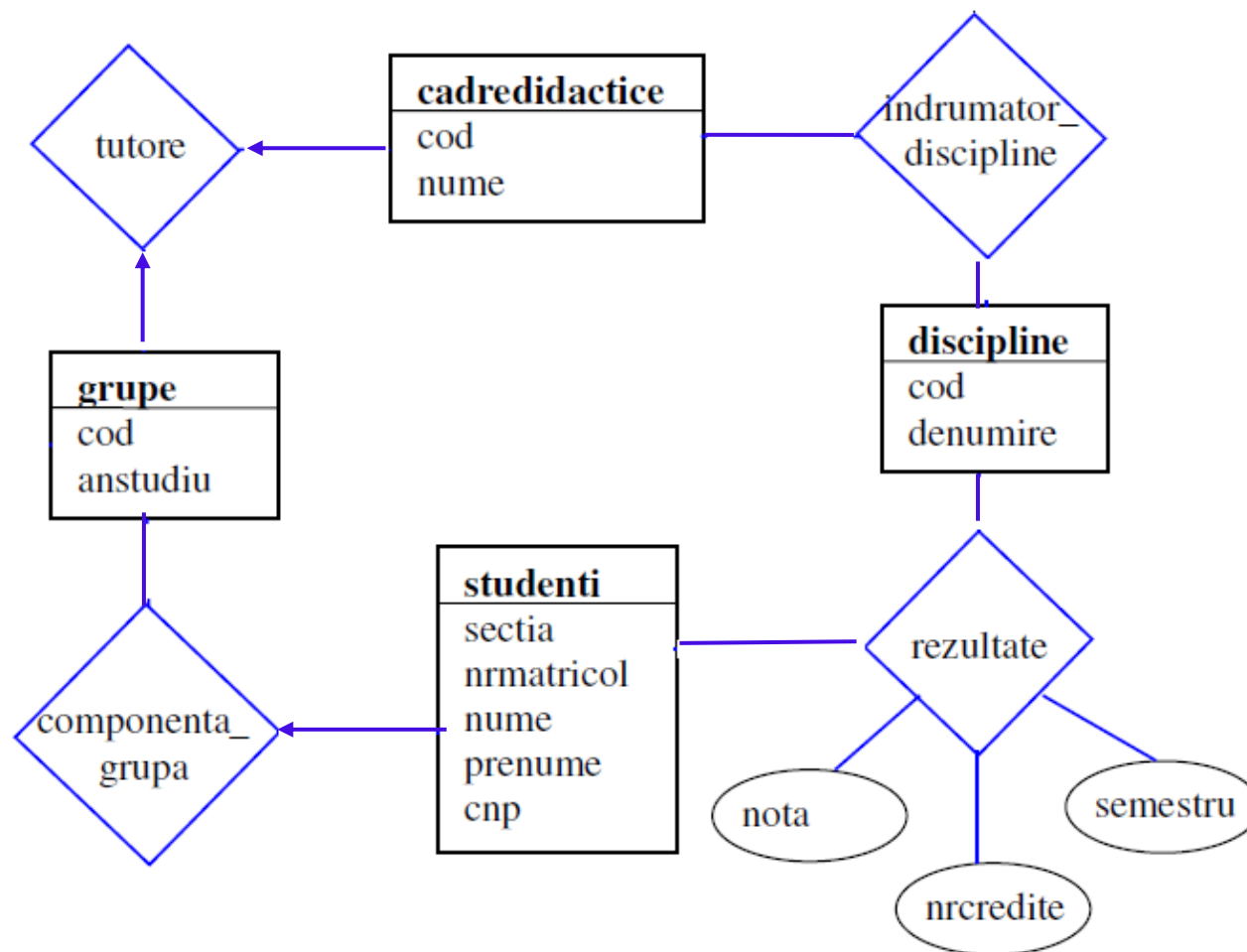
Modelul Entitate-Relatie (ER) – cont.

- Pentru relatiile binare, segmentul ce uneste tipul de relatie cu tipul de entitate este marcat într-un anumit mod, de exemplu:



Modelul Entitate-Relatie (ER) – cont.

- ▶ **Exemplu** (atributele dintr-un tip de entitate s-au precizat într-un mod simplificat):



Entitati si relatii pentru exemplul precedent

Referinte

- ▶ [Da04] DATE, C.J., An Introduction to Database Systems (8th Edition), Addison-Wesley, 2004.
- ▶ [El10] ELMASRI R., NAVATHE S.B., *Fundamentals of Database Systems*, 6th Ed., Pearson, 2010.
- ▶ [Ga08] GARCIA-MOLINA, H., ULLMAN, J., WIDOM, J., *Database Systems: The Complete Book*, Pearson Prentice Hall, 2008
- ▶ [Mi09] MIU, L., OZSU, M.T., *Encyclopedia of Database Systems*, Springer 2009 (3818 pages).
- ▶ [Ra07] RAMAKRISHNAN, R., *Database Management Systems*. McGraw-Hill, 2007, <http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html>
- ▶ [Si10] SILBERSCHATZ A., KORTZ H., SUDARSHAN S., *Database System Concepts*, McGraw-Hill, 2010, <http://codex.cs.yale.edu/avi/db-book/>
- ▶ [Ul11] ULLMAN, J., WIDOM, J., *A First Course in Database Systems* (3rd Edition), Addison-Wesley + Prentice-Hall, 2011
- ▶ Leon Tambulea, curs de Baze de Date, UBB Cluj-Napoca
- ▶ Surdu Sabina, seminar de Baze de Date, Matematica-Informatica, UBB Cluj-Napoca, 2016
- ▶ Sara BouchenaK, Curs Bases de données et systemes d'information, licenta MIAGE, Université Joseph Fourier Grenoble, 2012

