Baze de date

Curs 1

Introducere - concepte fundamentale în bazele de date

Baze de date

- C2 + S1 + L2
- nota finală
  - 50% examen scris (E)
  - 25% examen practic (P)
  - 25% medie laborator (L)
  - pentru promovare: E, L, P >= 5
- pentru intrarea în examen sunt necesare cel puțin 5 prezențe la seminar și cel puțin 12 prezențe la laborator, atât în sesiunea normală, cât și în cea de restanțe, conform Hotărârii Consiliului Departamentului de Informatică: http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf
- http://www.cs.ubbcluj.ro/~sabina
- sabina@cs.ubbcluj.ro

Context

- omniprezență baze de date
  - învățământ
  - cercetare
  - finanțe
  - presă
  - comerț electronic
  - rețele sociale
  - turism
  - telecomunicații
  - …
- discuție curs: gestiune cheltuieli personale studenți, breșa de securitate

Context

- explozie informațională fără precedent, seturi de date complexe, de dimensiuni foarte mari
- o organizație trebuie să își gestioneze datele eficient, să poată obține informații corecte și în timp util despre operațiunile sale, informații pe care le folosește pentru a-și susține activitățile
- nevoia de sisteme de gestiune a datelor puternice și flexibile – simplificarea gestiunii datelor și a extragerii de informații utile în timp optim, abilitatea de a găsi rapid informația relevantă pentru o anumită întrebare

1. Componentele unei aplicații

- date (memorate în fișiere sau baze de date)
- algoritm de gestiune
- interfața cu utilizatorul

2. Metode de memorare a datelor

- fișiere
- baze de date
- baze de date distribuite

3. Caracteristicile fișierelor

- e.g., colecție foarte mare de date a unei bănci: angajați, clienți, conturi, tranzacții etc; datele sunt accesate concurent de mai mulți angajați
- întrebările (i.e., interogările) referitoare la date trebuie să primească un răspuns rapid
- modificările asupra datelor făcute de diverși utilizatori trebuie să fie aplicate corect
- accesul la anumite părți din date trebuie să fie restricționat (e.g., salariile)
- care sunt dificultățile pe care le-am întâmpina dacă am încerca să stocăm aceste date într-o colecție de fișiere ale sistemului de operare?

3. Caracteristicile fișierelor

- prezintă mai multe formate de memorare a datelor; codul scris într-o manieră ad-hoc pentru o aplicație nu poate fi exploatat pentru aplicații pe alte fișiere
- există redundanță în memorarea datelor, unele dintre acestea se regăsesc în mai multe fișiere; acest fapt poate duce la inconsistența datelor
- operațiile de citire / scriere sunt descrise în program; se ia în considerare o anumită structură a înregistrărilor, fapt ce conduce la greutăți în dezvoltarea unui program (prin schimbarea structurii fișierelor trebuie modificat programul)
- este dificilă obținerea datelor care îndeplinesc anumite condiții
- actualizarea datelor este complexă (e.g., modificarea unor valori din înregistrări, ștergerea unor înregistrări)
- verificarea anumitor condiții de integritate (corectitudine) se face din program
- trebuie gestionată memoria internă (e.g., cum se încarcă o colecție de date de zeci sau sute de GB în memorie pentru procesare?)

3. Caracteristicile fișierelor

- nu există proceduri de securitate adecvate, i.e., e nevoie de politici de securitate în care diferiților utilizatori li se acordă sau nu permisiunea de a accesa anumite porțiuni din date
- nu se poate controla ușor accesul concurent la date
- datele trebuie readuse la o formă corectă dacă sistemul întâmpină probleme în timp ce se operează modificări asupra lor, e.g., o operațiune bancară care transferă bani din contul A în contul B este întreruptă de o pană de curent după ce a scos bani din contul A, dar înainte de a-i fi depus în B; datele trebuie readuse la forma corectă, banii trebuie puși înapoi în contul A
- fișierele sunt utile pentru programe care necesită puține date și sunt folosite de un singur utilizator, însă astăzi crește atât cantitatea de date, cât și numărul utilizatorilor care folosesc o anumită aplicație

4. Baza de date

- **proiectarea** bazei de date
  - cum descriem o organizație (e.g., o companie) în termenii datelor dintr-o bază de date?
- **analiza** datelor
  - cum răspundem la întrebări referitoare la organizația respectivă formulând interogări pe datele din baza de date?

4. Baza de date

- e.g., o bază de date a unui liceu ar putea conține informații despre:
  - **entități**, cum ar fi elevi, clase, profesori, discipline
  - **relații între entități**, cum ar fi apartenența elevilor la clase de elevi, predarea unor discipline la anumite clase de către anumiți profesori, calitatea de diriginte a unui profesor pentru o clasă
- cum reprezentăm aceste entități și relațiile dintre ele?

5. Modele de descriere a datelor

- pentru ca datele să poată fi gestionate automat, este necesar să fie descrise conform unui model
- un **model de descriere a datelor** este o mulțime de **concepte și reguli** folosite pentru modelarea datelor; aceste concepte permit descrierea structurii datelor, precizarea restricțiilor de corectitudine și descrierea relațiilor cu alte date
- în cadrul unui model, pentru a descrie o anumită colecție de date, memorată într-o bază de date, se folosesc anumite structuri de date, care constituie **schema** bazei de date (șablonul sau structura datelor); datele din colecție respectă schema și pot fi considerate **instanțe** sau realizări ale schemei (analogie cu clasele și obiectele din programarea orientată obiect)
- construcțiile de descriere a datelor din cadrul unui model de date sunt *high-level,* ascunzând multe detalii *low-level* legate de stocare, e.g., de la entitatea Elev până la biții stocați de calculator e cale lungă ☺

5. Modele de descriere a datelor

- entitate-relație
- relațional
- rețea
- ierarhic
- orientat obiect
- noSQL
- semistructurat (XML)
- fluxuri de date

5. Modele de descriere a datelor: modelul relațional

- construcția principală pentru descrierea datelor este **relația**, i.e., o mulțime de înregistrări
- în modelul relațional, schema unei relații precizează numele acesteia, numele și tipul fiecărui câmp (atribut, coloană)
  - e.g., reprezentarea informației despre filme:

    Film(*codf*: string, *titlu*: string, *regizor*: string, *an*: integer)

5. Modele de descriere a datelor: modelul relațional

- e.g., instanță a relației Film, în care fiecare înregistrare are 4 câmpuri:

fid	*titlu*	*regizor*	*an*
84386	Hibernatus	Édouard Molinaro	1969
7583	Moscova nu crede in lacrimi	Vladimir Menshov	1980
47288	Close Encounters of the Third Kind	Steven Spielberg	1977
32	Contact	Robert Zemeckis	1997
46747	E.T. the Extra-Terrestrial	Steven Spielberg	1982

5. Modele de descriere a datelor: modelul entitate-relație

- este un model **semantic**, mai abstract, de nivel înalt, care ușurează sarcina utilizatorului de a realiza o bună descriere inițială a datelor
- modelele semantice sunt utile întrucât, deși modelul sistemului care gestionează baza de date ascunde multe detalii, totuși e mai apropiat de modul în care se stochează datele decât de perspectiva utilizatorului, de modul în care utilizatorul se gândește la datele sale
- un design într-un astfel de model e transformat ulterior în termenii modelului de date al sistemului care gestionează baza de date, e.g., transformarea ER – relațional
- modelul entitate-relație e extrem de utilizat și ne permite să ilustrăm grafic entitățile și relațiile dintre acestea

5. Modele de descriere a datelor: modelul entitate-relație

- principalele **concepte** utilizate în acest model sunt: entitatea, atributul și relația
- **entitatea** – este o dată, reprezintă un obiect din lumea reală; pentru precizarea ei se folosesc anumite atribute (proprietăți)
- mulțimea entităților cu aceeași structură (de exemplu, mulțimea studenților) sunt instanțe ale unui **tip de entitate** (clasă / schemă a entității)
- un tip de entitate are un nume și o listă de atribute
- în precizarea tipului de entitate, fiecare **atribut** are: nume, domeniu pentru valorile posibile și eventuale condiții pentru a verifica dacă valoarea este corectă
- la un tip de entitate se poate defini o **cheie** (care este o restricție): o mulțime de atribute care iau valori distincte în instanțele tipului de entitate

5. Modele de descriere a datelor: modelul entitate-relație

- **relația** – este o dată și precizează o legătură (asociere) între două sau mai multe entități; la această asociere se pot folosi și atribute suplimentare
- toate relațiile cu aceeași structură (legături între entități de aceleași tipuri) trebuie descrise printr-un **tip de relație** sau schemă a relației
- un tip de relație are un nume, tipurile de entitate folosite în asociere și posibile atribute suplimentare

- **schema modelului** este formată dintr-o mulțime de tipuri de entitate și tipuri de relație

5. Modele de descriere a datelor: modelul entitate-relație

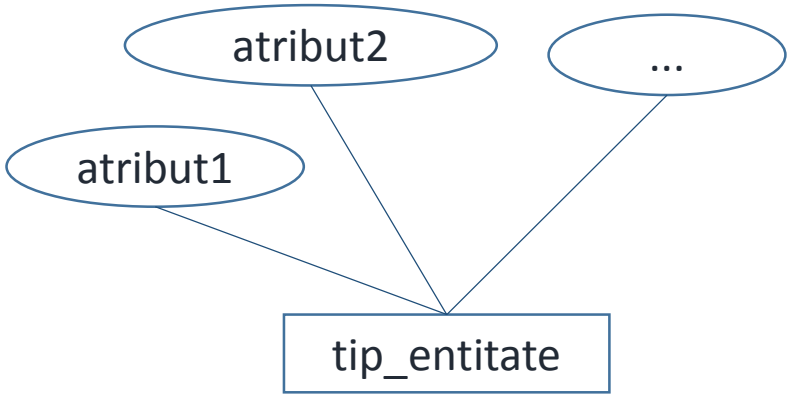
- pentru **relațiile binare** (între tipurile de entitate T1 și T2) se pot defini următoarele **tipuri particulare de relație:**
  - **1:1**: dacă o entitate de tipul T1 se asociază cu cel mult o entitate de tipul T2, iar o entitate de tipul T2 se asociază cu cel mult o entitate de tipul T1
    - e.g., asocierea dintre grupă și cadru didactic – pentru a preciza tutorii grupelor
  - **1:n**: dacă o entitate de tipul T1 se asociază cu oricâte entități de tipul T2, iar o entitate de tipul T2 se asociază cu cel mult o entitate de tipul T1
    - e.g., asocierea dintre grupă și studenți – pentru a preciza componența grupelor
  - **m:n:** dacă o entitate de tipul T1 se asociază cu oricâte entități de tipul T2, iar o entitate de tipul T2 se asociază cu oricâte entități de tipul T1
    - e.g., asocierea dintre discipline și studenți – pentru a preciza contractele de studiu

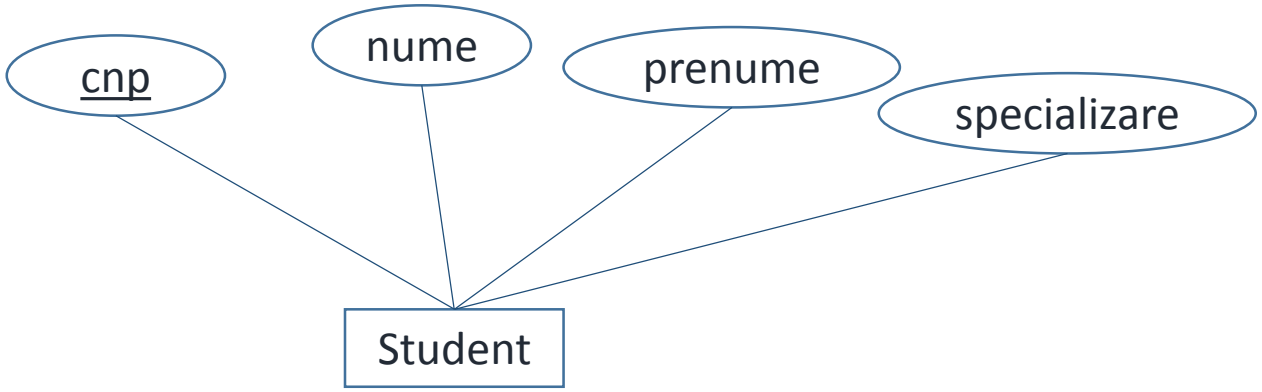
5. Modele de descriere a datelor: modelul entitate-relație

- acestea pot fi considerate **restricții** pentru baza de date (sistemul trebuie să verifice, la fiecare modificare a bazei de date, dacă relația este de tipul precizat)

5. Modele de descriere a datelor: modelul entitate-relație

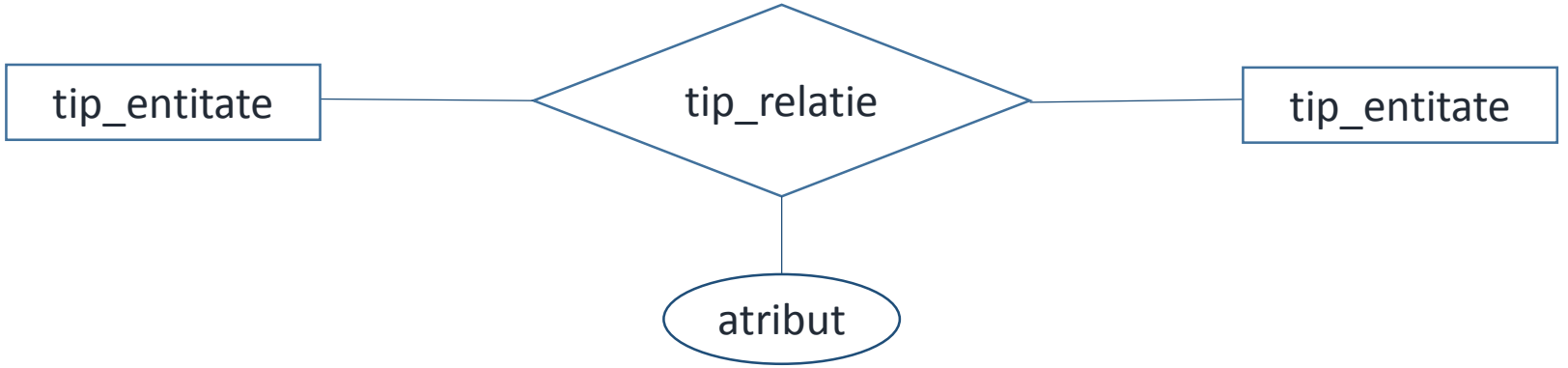
- reprezentarea grafică a modelului
  - tipul de entitate și atributele asociate

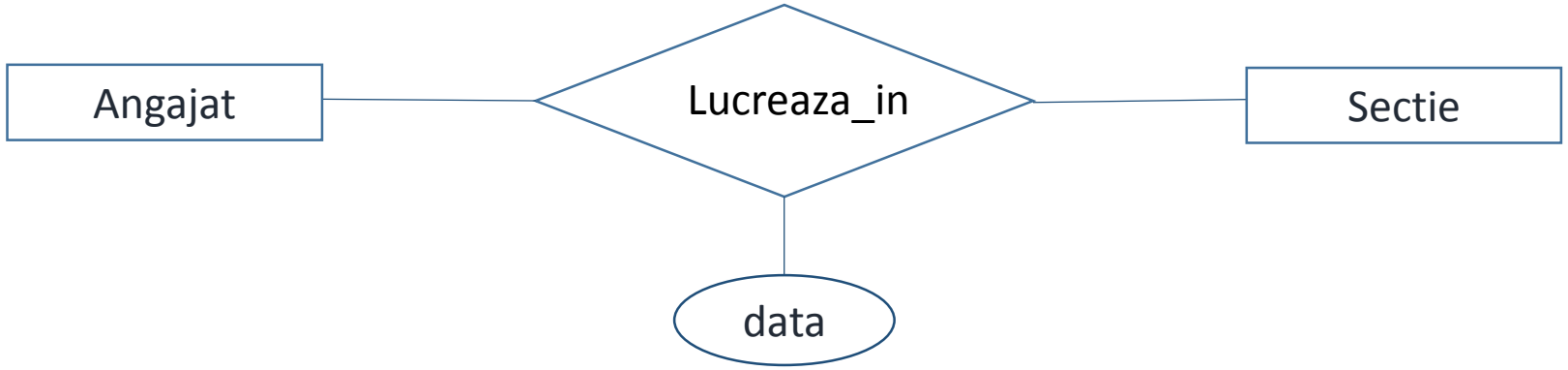




5. Modele de descriere a datelor: modelul entitate-relație

- reprezentarea grafică a modelului
  - tipul de relație și atributele asociate



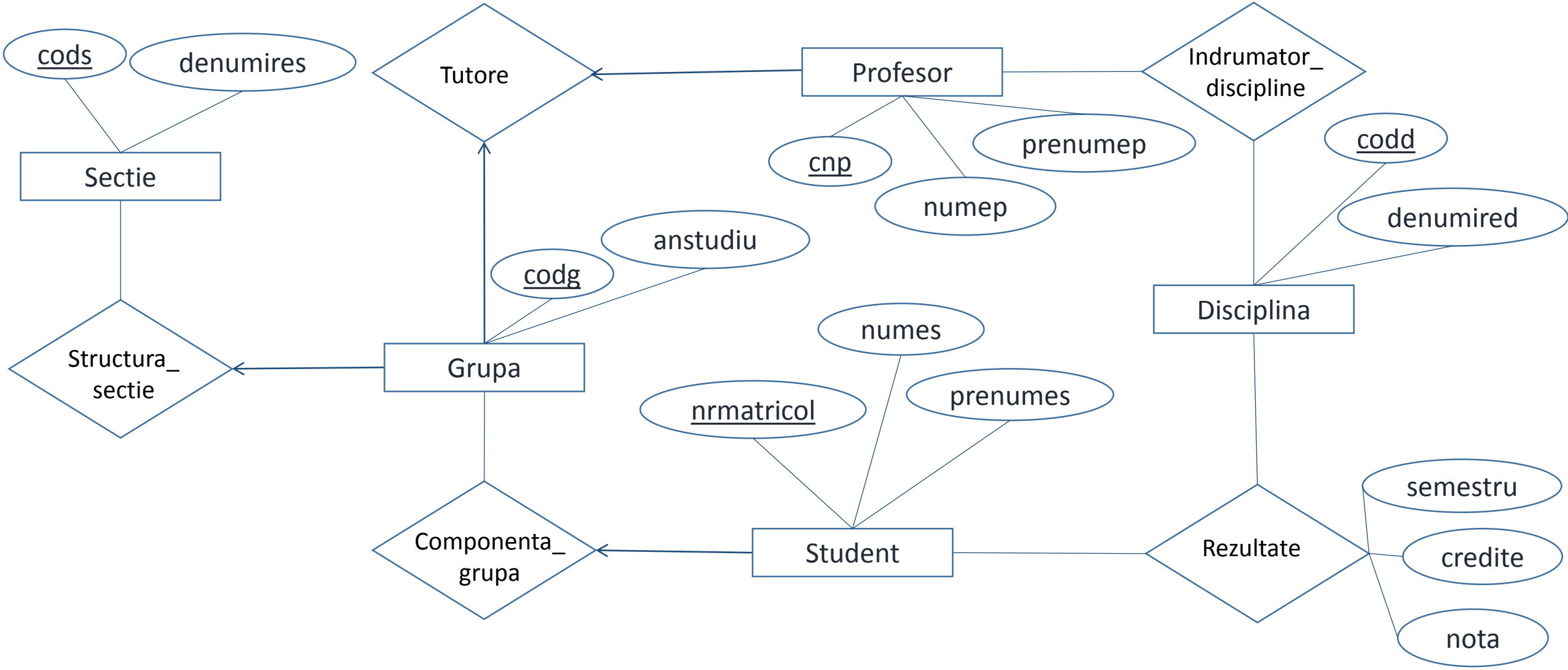


5. Modele de descriere a datelor: modelul entitate-relație

- reprezentarea grafică a modelului
  - convenție reprezentare tip de relație 1:n:

5. Modele de descriere a datelor: modelul entitate-relație

- **exemplu:**



6. Baza de date și SGBD

- **baza de date** conține:
  - **descrierea structurilor de date** folosite pentru modelarea datelor (schema bazei de date) într-un dicționar al bazei de date*
  - o **mulțime de date** – realizări sau instanțe ale schemei (colecția de date care respectă un model de organizare)
  - **componente** diverse: view-uri, proceduri și funcții, roluri, utilizatori etc
- apare separarea între:
  - **definirea datelor** (păstrată într-un dicționar al bazei de date)
  - **gestiunea datelor** (adăugare, ștergere, modificare) și interogare

* dicționar (catalog) – conține metadate (informații cu caracter descriptiv despre date); o schemă pentru un astfel de dicționar ar putea fi:

(nume_atribut, nume_relatie, tip_atribut, pozitie_atribut_in_relatie)

6. Baza de date și SGBD

- **sistem de gestiune a bazei de date** (SGBD)
  - colecție de programe necesare pentru gestiunea bazei de date
- exemple de SGBD
  - Oracle, DB2 (IBM), Microsoft SQL Server, Sybase (SAP), Informix (IBM), Teradata, MySQL, PostgreSQL, Access (Microsoft), Paradox, Foxpro, ...
- **sistem de baze de date**
  - baza de date + SGBD

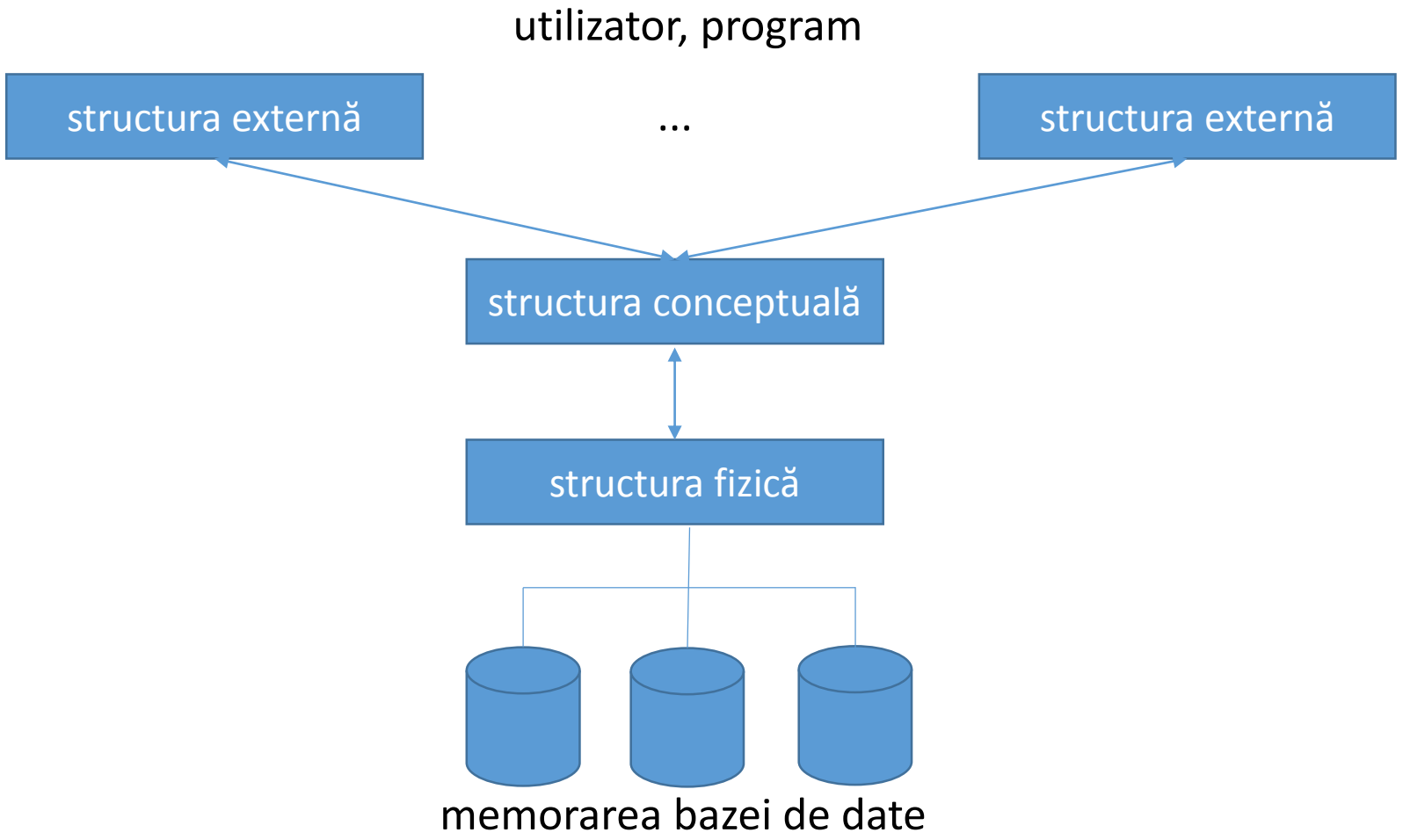
7. Structurile unei baze de date

- când se gândește cum ar putea fi organizate și stocate informațiile despre o organizație de exemplu într-o bază de date, utilizatorul operează cu concepte de nivel înalt corespunzătoare entităților din organizație și relațiilor dintre acestea
- pe de altă parte, SGBD stochează datele sub forma unui număr foarte mare de biți
- diferența dintre modul în care utilizatorii se gândesc la datele lor și maniera în care acestea sunt stocate este soluționată prin nivelurile de abstractizare pe care le permit SGBD-urile

7. Structurile unei baze de date

- în 1975 s-a propus **arhitectura ANSI-SPARC**, o arhitectură pentru un sistem de baze de date organizat pe trei niveluri; în general, acest model este respectat de principalele sisteme de gestiune și cuprinde:
  - **structura conceptuală (schema bazei de date)**: descrie toate structurile de date si restricțiile folosite în baza de date
  - **structuri externe**: descrierea structurilor de date folosite de un utilizator/program particular; această descriere se face într-un anumit model de organizare, iar SGBD poate să regăsească datele în structura conceptuală
  - **structura fizică (structura internă)**: descrierea structurilor de memorare a bazei de date (fișiere de date, indecși etc)

7. Structurile unei baze de date



7. Structurile unei baze de date

- e.g., structura conceptuală – informații despre entități, e.g., elevi și discipline, respectiv despre relații între entități, e.g., rezultatele obținute de elevi:

  Elev(*cnp*: string, *nume*: string, *prenume*: string, *medie*: real)

  Profesor(*codp*: string, *numep*: string, *prenumep*: string, *salariu*: real)

  Disciplina(*codd*: string, *denumire*: string)

  Rezultat(*cnp*: string, *codd*: string, *nota*: real)

  Predare(*codp*: string, *codd*: string)

- e.g., structura fizică – informații despre stocarea pe disc a relațiilor, despre crearea unor indecși (structuri de date menite să accelereze regăsirea datelor):

  se creează indecși pe prima coloană din relațiile Elev și Profesor

7. Structurile unei baze de date

- e.g., structură externă – informații despre numele profesorilor care predau discipline și numărul de evaluări pentru fiecare disciplină

InfoDisciplina(*codd*: string, *numep*: string, *prenumep*: string, *evaluari:* integer)

- InfoDisciplina e un *view*, o relație din punct de vedere conceptual, însă înregistrările sale nu sunt stocate în sistem, ci se obțin utilizând o definiție pentru view care ia în calcul relațiile stocate în baza de date
- introducerea InfoDisciplina în schema conceptuală => redundanță în stocarea anumitor date + baza de date ar fi predispusă la erori, e.g., dacă introducerea unei înregistrări pentru un rezultat nou în Rezultat nu ar fi urmată și de modificarea câmpului evaluari din InfoDisciplina
- o bază de date poate conține mai multe structuri externe, fiecare dintre acestea fiind adaptată pentru un anumit grup de utilizatori

8. Independența logică și independența fizică

- cele trei niveluri de abstractizare fac posibilă **independența datelor**: aplicațiile sunt izolate de modificările care pot apărea în structura sau stocarea datelor
- **independența logică**: posibilitatea schimbării structurii conceptuale fără schimbarea structurii externe, deci a programelor care folosesc date din baza de date
  - important: permite dezvoltarea în etape a aplicațiilor
    - e.g., relația Profesor se înlocuiește cu:

      ProfesorPublic(*codp*: string, *numep*: string, *prenumep*: string)
      ProfesorPrivat(*codp*: string, *salariu*: real)

definiția InfoDisciplina poate fi modificată pentru a lua în calcul ProfesorPublic în acest caz; un utilizator care interoghează InfoDisciplina va obține același răspuns ca înainte de modificarea schemei conceptuale

8. Independența logică și independența fizică

- **independența fizică**: posibilitatea schimbării structurii fizice fără schimbarea structurii conceptuale și a structurilor externe (deci a programelor care folosesc date din baza de date)
  - important: se pot adăuga / elimina fișiere (e.g., index) pentru optimizarea căutării, programele utilizatorilor nu consultă direct fișierele (structura fizică)

9. Funcțiile sistemelor de gestiune a bazelor de date

- **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 utilizării bazei de date, monitorizarea performanței bazei de date, optimizarea performanței 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)

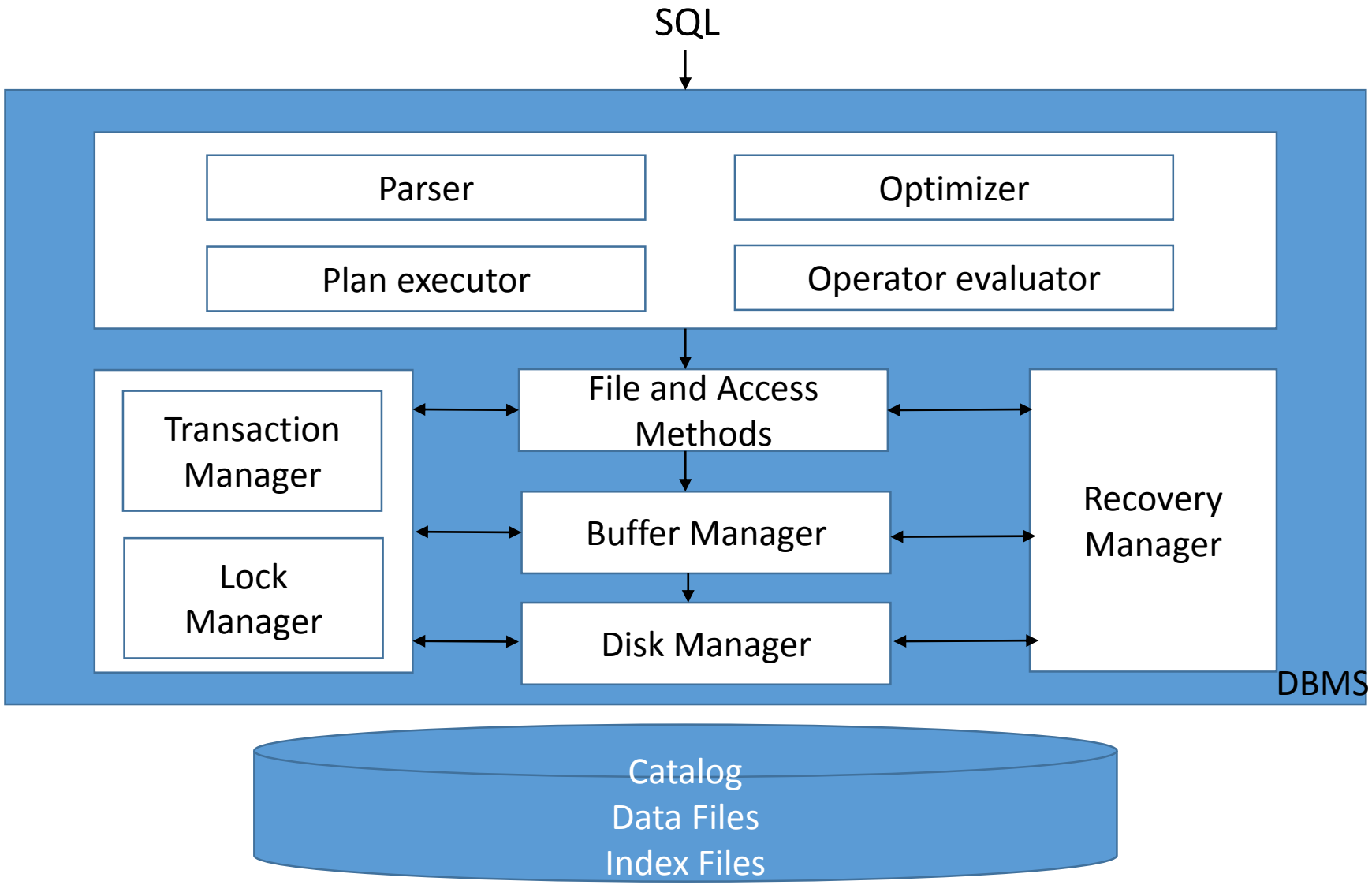
10. Tipuri de utilizatori ai bazei de date

- **administratori** ai bazei de date
- **proiectanți** ai bazei de date (designers): responsabili cu schema bazei de date, restricțiile definite, funcțiile si procedurile puse la dispoziția utilizatorilor, optimizările necesare
- **utilizatorii unor aplicații** care gestionează date
- **programatori aplicații**

- aplicațiile sunt scrise în diverse limbaje sau medii de programare (aplicații Web, Java, .NET etc)
- pentru efectuarea unei operații asupra bazei de date se trimite din aplicație la sistemul de baze de date o comandă, aici se execută comanda și se trimite răspunsul la aplicație (tehnologia client/server)
- cererea este trimisă în **limbajul SQL** (Structured Query Language), limbaj standard pentru acces la bazele de date

11. Arhitectura unui SGBD

- în [Ra07] se propune următoarea structură a unui SGBD:



11. Arhitectura unui SGBD

- comenzile SQL pot proveni de la o multitudine de interfețe utilizator (Web Forms, interfață SQL etc) și pot fi incluse în aplicații scrise în diferite limbaje de programare, e.g., Java, C# etc
- Optimizer – produce un plan de execuție eficient pentru evaluarea interogărilor, luând în calcul informații despre maniera în care sunt stocate datele
- File & Access Methods, Buffer Manager, Disk Manager – abstractizarea fișierelor și a indecșilor, aducerea paginilor de pe disc în memorie, gestiunea spațiului pe disc
- Transaction Manager, Lock Manager – controlul concurenței, monitorizarea cererilor pentru blocări și acordarea blocărilor pentru obiectele din bază când acestea devin disponibile
- Recovery Manager – menține un log și restabilește starea consistentă a sistemului după un incident (e.g., o pană de curent)

12. Avantajele utilizării unui sistem de baze de date

- permite gestiunea unor colecții mari de date cu diverse legături între date
- aplicațiile nu gestionează detaliile de implementare a bazei de date: ele trimit o comandă SQL și primesc un răspuns (evaluarea comenzii se face de SGBD, folosind diverse programe de acces la date); oferă o vizualizare abstractă a datelor pentru a izola codul aplicațiilor de detalii referitoare la reprezentarea și stocarea datelor (independența datelor)
- permite dezvoltarea în etape a sistemelor informatice (modificarea schemei bazei de date, modificarea aplicațiilor, dezvoltarea de noi aplicații)
- utilizează tehnici sofisticate pentru stocarea și regăsirea eficientă a datelor (optimizarea accesului la date – util pentru colecții de date de dimensiuni mari)
- permite accesul la date din aplicații dezvoltate în diverse limbaje sau medii de programare

12. Avantajele utilizării unui sistem de baze de date

- dacă datele sunt accesate întotdeauna prin sistem, acestea sunt actuale și corecte (se verifică automat unele restricții de integritate)
- controlul accesului la baza de date (pentru utilizatori cu diverse roluri), i.e., ce date sunt vizibile pentru ce clase de utilizatori
- gestiunea accesului concurent
- face posibilă recuperarea în cazul erorilor (log)
- oferă posibilitatea de import / export a datelor în diverse formate
- oferă instrumente de analiză a datelor (depozite de date, data mining)
- reduce timpul de dezvoltare a aplicațiilor

13. Referințe

- [Ta13] ȚÂMBULEA, L., Curs Baze de date, Facultatea de Matematică și Informatică, UBB, 2013-2014
- [Ra00] RAMAKRISHNAN, R., GEHRKE, J., Database Management Systems (2nd Edition), McGraw-Hill, 2000
- [Da03] DATE, C.J., An Introduction to Database Systems (8th Edition), Addison-Wesley, 2003
- [Ga08] GARCIA-MOLINA, H., ULLMAN, J., WIDOM, J., Database Systems: The Complete Book, Prentice Hall Press, 2008
- [Ha96] HANSEN, G., HANSEN, J., Database Management And Design (2nd Edition), Prentice Hall, 1996
- [Ra07] RAMAKRISHNAN, R., GEHRKE, J., Database Management Systems, McGraw-Hill, 2007, http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html
- [Si10] SILBERSCHATZ, A., KORTH, 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, http://infolab.stanford.edu/~ullman/fcdb.html