1. Noţiuni introductive

Cuprins

- Date și informații
- Triada entitate, atribut, valoare
- Organizarea şi prelucrarea datelor
- Baza de date
- Arhitectura bazelor de date
- Sisteme de gestiune a bazelor de date (SGBD)
- Etapele dezvoltării unei baze de date
- Modele de date
  - Modelul entitate-legătură
  - Diagrama entitate-legătură

1.1. Date și informații

- **Datele** sunt seturi de caractere acceptate ca intrări într-un sistem informaţional, intrări ce sunt memorate şi prelucrate:
  - sunt culese din lumea reală pe bază de observaţii şi măsurători
  - au o anumită semnificaţie
  - au un caracter obiectiv
  - pot fi prelucrate manual sau automat

- **Informaţia** este rezultatul prelucrării datelor, utilizată în cadrul activităţii de luare a deciziilor:
  - are un caracter subiectiv
  - are o natură variată

1.2. Triada entitate, atribut, valoare

- *Entitate*: obiectul informaţiei
  - Un obiect concret sau abstract definit prin proprietăţile sale
  - Exemple: client, furnizor, salariat, produs, etc.

- *Atribut* (sau caracteristică): element de descriere a proprietăţilor unei entităţi (o proprietate a acesteia)
  - O entitate poate avea mai multe atribute

- *Valoarea*: măsură a atributului asociat
  - Un atribut poate avea mai multe valori

- Exemplu: entitatea CLIENT poate fi specificată prin perechile (ATRIBUT, VALOARE):

  (NUME, POPESCU);
  (PRENUME, ION);
  (LOCALITATE, BUCUREŞTI);
  (TELEFON, 0213211231);

1.3. Organizarea şi prelucrarea datelor

- Sisteme tradiţionale bazate pe fişiere
  - Colecţie de aplicaţii, care efectuează servicii pentru utilizatorii finali, cum ar fi producerea de rapoarte
  - Fiecare aplicaţie defineşte şi gestionează propriile sale date
  - **Fişierul:** principalul tip de organizare a datelor
    - fiecare dată este descrisă independent în toate fişierele în care apare
    - între fişiere nu există o relaţie definită explicit

- Caracteristicile sistemelor de prelucrare bazate pe fişiere
  - Răspunsul la o nouă problemă implică scrierea unei noi aplicaţii care creează fişierele de date corespunzătoare
  - Pentru o organizaţie fişierele de date au formate diferite iar aplicaţiile pot fi scrise în limbaje diferite
  - **Cauze** ale limitării tratării anterioare:
    - definiţia datelor este încorporată în programele aplicaţie
    - nu există controlul accesului şi manipulării datelor

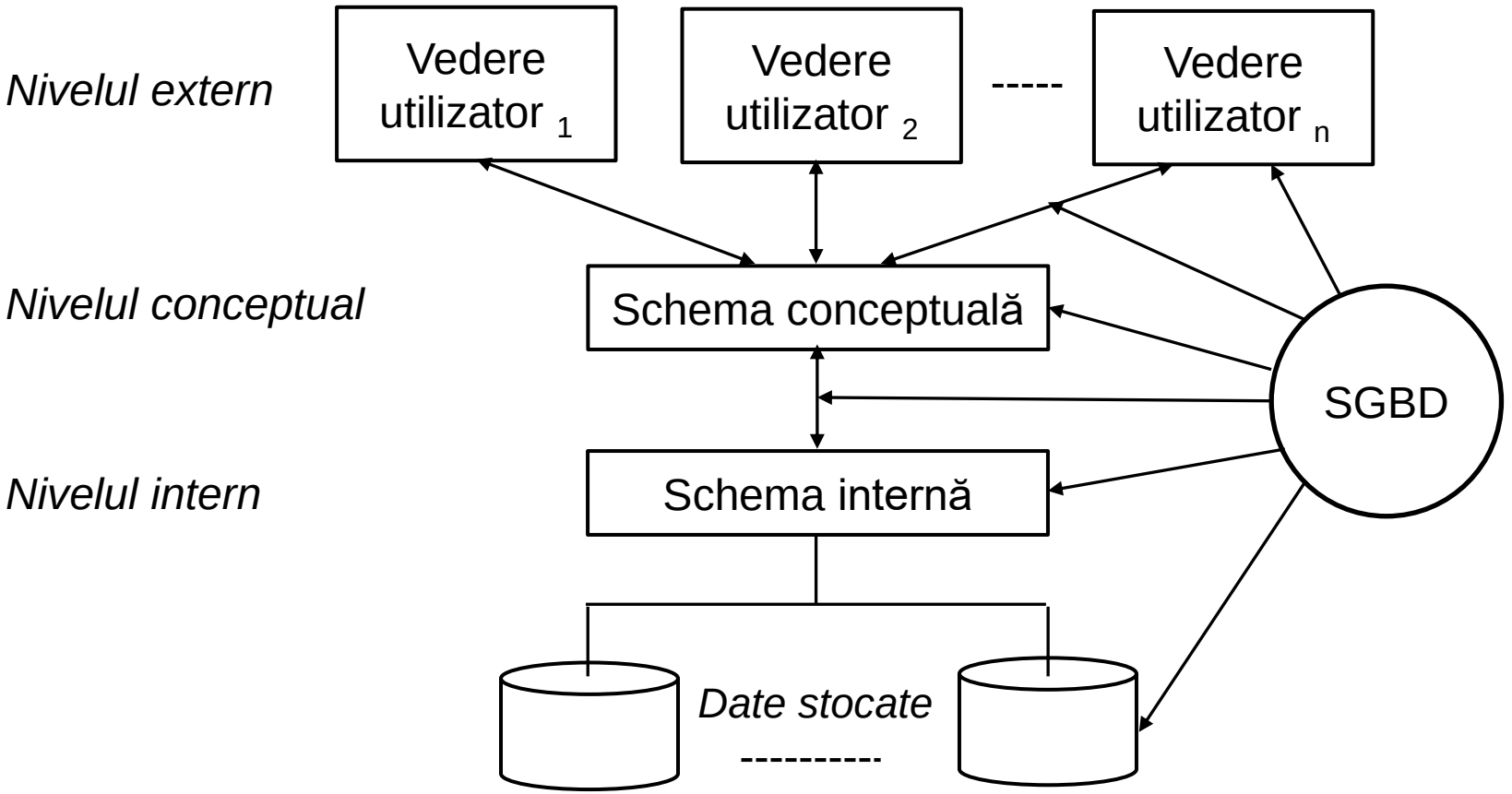
1.4. Baza de date

- Colecţie *partajată* de date, între care există relaţii logice (*interdependenţe*) şi o descriere a acestor date, proiectată pentru a satisface necesităţile informaţionale ale unei organizaţii
  - Colecţia este *autodescrisă*
  - Permite obţinerea operativă a unor informaţii utile despre un anumit subiect

- **Tratarea prin baze de date**
  - **Schema** bazei de date: descrierea generală a bazei de date
    - Este specificată în procesul de proiectare și se modifică foarte rar
  - **Instanţa** bazei de date: dată de setul de date operaţionale din baza de date la orice moment dat (se modifică frecvent)
  - **Natura autodescriptivă->**independenţa programelor faţă de date
  - Analiza necesităţilor informaţionale ale unei organizaţii

1.5. Arhitectura bazelor de date

- Arhitectura ANSI/SPARC cu trei niveluri



- **Nivelul intern**
  - Baza de date fizică:
    - colecţie de fişiere care conţin datele fizice, la care se adaugă structuri auxiliare menite să asigure accesul operativ la aceste date (directoare, indecşi, tabele de dispersie,…)
  - Probleme tratate:
    - alocarea spaţiului de stocare pentru date şi indecşi
    - descrierile înregistrărilor pentru stocare (cu dimensiunile articolelor de date)
    - plasarea înregistrărilor
    - tehnici de compresie a datelor şi de codificare a acestora
  - Schimbarea sistemului de operare sau modificări în configuraţia echipamentelor hardware pot atrage modificări ale bazei de date fizice, dar acestea nu vor afecta celelalte nivele

- **Nivelul conceptual**
  - Abstractizare a unei părţi din lumea reală
  - Descrie structura logică a datelor:
    - ce date sunt stocate într-o bază de date şi relaţiile dintre acestea, prin specificarea unor constrângeri
  - Constrângeri: proprietăţi ale datelor ce nu pot fi exprimate prin descrieri de structură
    - restricţii asupra valorilor pe care le pot lua datele
    - restricţii privind legăturile dintre diferite unităţi logice
  - Probleme tratate:
    - specificarea entităţilor, a atributelor şi a relaţiilor dintre acestea
    - constrângeri asupra datelor
    - informaţii de securitate şi integritate a datelor
  - Realizează *independenţa fizică* a datelor
  - Integrează viziunile tuturor utilizatorilor asupra bazei de date

- **Nivelul extern**
  - Vederea (*view*) utilizatorului asupra bazei de date
  - Descrie acea parte a bazei de date care este relevantă pentru fiecare utilizator
  - Cuprinde: unităţi logice din modelul conceptual şi unităţi logice care nu există în modelul conceptual şi care nu au corespondent direct în baza de date fizică (unităţi logice virtuale)
  - Fiecărui utilizator îi corespunde un **model extern propriu**, individualizat în raport cu cerinţele specifice
  - Avantaje:
    - asigurarea securităţii bazei de date prin limitarea accesului la date a anumitor categorii de utilizatori, sau prin acordarea de drepturi de acces diferite pentru un utilizator în cadrul mai multor vederi
    - viziune individualizată şi simplificată asupra bazei de date
  - Realizează *independenţa logică* a datelor

- Un sistem de baze de date suportă o schemă internă, o schemă conceptuală şi mai multe scheme externe:
  - toate aceste scheme sunt descrieri diferite ale aceleiaşi colecţii de date, care există doar în nivelul intern

- Toate aceste reprezentări ale datelor sunt gestionate de către SGBD (**S**istem de **G**estiune a **B**azelor de **D**ate) (în engleză DBMS: **D**ata**B**ase **M**anagement **S**ystem) care asigură, de asemenea, şi cele două corespondenţe (*mappings*):
  - între schemele externe şi schema conceptuală
  - între schema conceptuală şi schema internă

- **Independenţa datelor**
  - Presupune existenţa unei delimitări nete între reprezentarea fizică a datelor şi imaginea pe care o are utilizatorul despre aceste date

  - **Independenţa fizică** dă măsura imunităţii aplicaţiilor faţă de modificările în structura fizică de memorare a datelor:
    - presupune că aplicaţiile nu conţin nici o referire explicită la tipul fişierelor în care sunt memorate datele, la tipul dispozitivului de memorare sau la strategia de acces la date

- **Independenţa logică** a datelor
  - Se referă la imunitatea modelului propriu al fiecărui utilizator faţă de modificările în structura logică globală a bazei de date
    - adăugarea de noi unităţi logice (câmpuri) la structura bazei de date
    - modificarea acestora şi a relaţiilor dintre ele
  - Permite:
    - dezvoltarea bazei de date fără a afecta utilizatorii care nu au nevoie de noile date
    - reorganizarea bazei de date:
      - regruparea câmpurilor în înregistrări
      - definirea de noi câmpuri pe baza celor existente
  - Problemă delicată: eliminarea unei entităţi logice din baza de date
    - afectează utilizatorii care fac referire la entitatea eliminată
  - **D.p.d.v. al utilizatorului**, problema independenţei logice se manifestă legat de operaţiile pe care sistemul îi permite să le efectueze asupra datelor din modelul propriu astfel încât să nu afecteze modelele altor utilizatori care folosesc parţial sau total aceleaşi date

- **Avantajele utilizării bazelor de date**
  - *Compactitate ridicată*: volumul ocupat de sistemele de baze de date este mult mai redus față de volumul ocupat de documente scrise sau volumul ocupat de fişiere necorelate
  - *Viteză mare* de regăsire şi actualizare a informaţiilor
  - *Redundanţă scăzută a* datelor memorate, care se obţine prin *partajarea datelor* între mai mulţi utilizatori şi aplicaţii
    - în sistemele de baze de date, mai multe aplicaţii pot folosi date comune, memorate o singură dată
    - de exemplu, o aplicaţie de personal şi o aplicaţie de rezultate la examene dintr-o universitate care exploatează o singură bază de date, pot folosi aceleaşi informaţii referitoare la structurarea facultăţilor şi a secţiilor

- *Posibilitatea de introducere a standardelor* privind modul de stocare a datelor, ceea ce permite interschimbul informaţiilor între diferite organizaţii
- *Menţinerea integrităţii datelor* prin politica de securitate (drepturi de acces diferenţiate în funcţie de rolul utilizatorilor), prin gestionarea tranzacţiilor şi prin refacerea datelor în caz de funcţionare defectuoasă a diferitelor componente hardware sau software
- *Independenţa datelor* faţă de suportul hardware utilizat
  - Sistemele de gestiune a bazelor de date oferă o vedere (*view*) externă a datelor, care nu se modifică atunci când se schimbă suportul de memorare fizic, ceea ce asigură imunitatea structurii bazei de date şi a aplicaţiilor la modificări ale sistemului hardware utilizat

1.6. Sisteme de gestiune a bazelor de date (SGBD/DBMS)

- Reprezintă un sistem de programe care permit utilizatorului definirea, crearea, întreţinerea bazei de date şi accesul controlat la aceasta

- Un SGBD oferă:
  - **facilităţi de descriere a datelor**
    - prin intermediul limbajului de descriere a datelor (DDL-Data Definition Language)
      - specificarea tipurilor de date şi a structurilor
      - specificarea constrângerilor asupra datelor

- **facilităţi de manipulare a datelor**
  - prin limbajul de manipulare a datelor (DML-Data Manipulation Language): actualizare date, inserarea de date, ştergerea de date, extragerea şi interogarea datelor
  - există două tipuri de limbaje de manipulare a datelor:
    - *limbaje procedurale:* tratează bazele de date înregistrare cu înregistrare şi specifică ***cum*** se va obţine rezultatul dorit
    - *limbaje neprocedurale:* operează asupra unor seturi de înregistrări şi descriu numai ***ce date*** vor fi obţinute (SQL-Structured Query Language)

- **accesul controlat la baza de date,** ce presupune existenţa:
  - *unui sistem de securitate:* previne accesarea bazei de date de către utilizatori neautorizaţi
  - *unui sistem de integritate:* menţine concordanţa datelor stocate

- *unui sistem de control al concurenţei:* permite accesul partajat la baza de date
- *unui sistem de control al refacerii:* restaurează baza de date într-o stare precedentă concordantă, în cazul unei defecţiuni hard sau soft
- *unui catalog accesibil utilizatorilor:* care conţine descrierea datelor din bază

– **un mecanism de vizualizare:**
  - permite fiecărui utilizator să-şi definească propriul mod de vizualizare a bazei de date

– **o colecţie de utilitare**:
  - editoare de rapoarte, generatoare de aplicaţii, programe asistent, module de proiectare, posibilităţi de dezvoltare a unor aplicaţii de tip CASE, etc.

- **Funcţiile unui SGBD**
  - stocarea, regăsirea şi reactualizarea datelor
  - un catalog accesibil utilizatorului care să conţină descrierile articolelor de date
    - conţine meta-date (date despre date)
  - asigurarea tranzacţiilor
    - ***tranzacţia*** constă într-o serie de acţiuni realizate de un singur utilizator sau un program aplicaţie prin care se accesează sau se schimbă conţinutul bazei de date
    - SGBD-ul furnizează un mecanism care garantează că sunt efectuate toate reactualizările corespunzătoare unei anumite tranzacţii sau că nu se efectuează nici una
  - servicii de control a concurenţei:
    - mecanism care garantează că baza de date este corect reactualizată atunci când mai mulţi utilizatori efectuează simultan astfel de operaţii

- servicii de reconstituire:
  - mecanism de reconstituire a unei baze de date în cazul în care aceasta este deteriorată într-un fel oarecare
- servicii de autorizare:
  - se garantează accesul la date numai pentru utilizatorii autorizaţi -> securitatea datelor
- suport pentru comunicarea datelor
- servicii de integritate:
  - mijloace care asigură că atât datele din baza de date cât şi modificările acestora respectă anumite reguli
- servicii suplimentare:
  - servicii pentru promovarea independenţei de date
  - servicii utilitare

- **Avantajele utilizării SGBD-urilor**
  - controlul redundanţei datelor
  - asigurarea coerenţei datelor
  - mai multe informaţii obţinute din aceeaşi cantitate de date
  - posibilitatea partajării datelor
  - integritate crescută a datelor
  - securitate crescută
  - concurenţă îmbunătăţită
  - posibilitatea aplicării standardelor
  - productivitate crescută
  - servicii de salvare de siguranţă şi refacere

- **Dezavantajele utilizării SGBD-urilor**
  - complexitate sporită
  - dimensiune
  - costul sistemelor SGBD
  - costuri adiţionale pentru elemente hardware
  - costul conversiei datelor
  - performanţa
  - impactul semnificativ al unei defecţiuni

- Scurt istoric al organizării şi prelucrării datelor
  - Sisteme tradiţionale bazate pe fişiere (1950-1960)
  - SGBD bazate pe modelul de date ierarhic sau reţea (1970)
  - SGBD relaţionale
    - Apariţia modelului relaţional (1970)
    - Dezvoltarea SGBD relaţionale (1970)
    - Apariţia SGBDR comerciale (1980)
    - Maturizarea tehnolohgiei relationale pentru SGBD (1990)
  - Sisteme de baze de date obiect-relaţionale
  - Sisteme de baze de date deductive şi sisteme de baze de date orientate obiect
  - Sisteme de baze de date orientate spre aplicaţii
  - Sisteme de depozitare a datelor (*data warehousing*) şi sisteme de explorare a datelor (*data mining*)

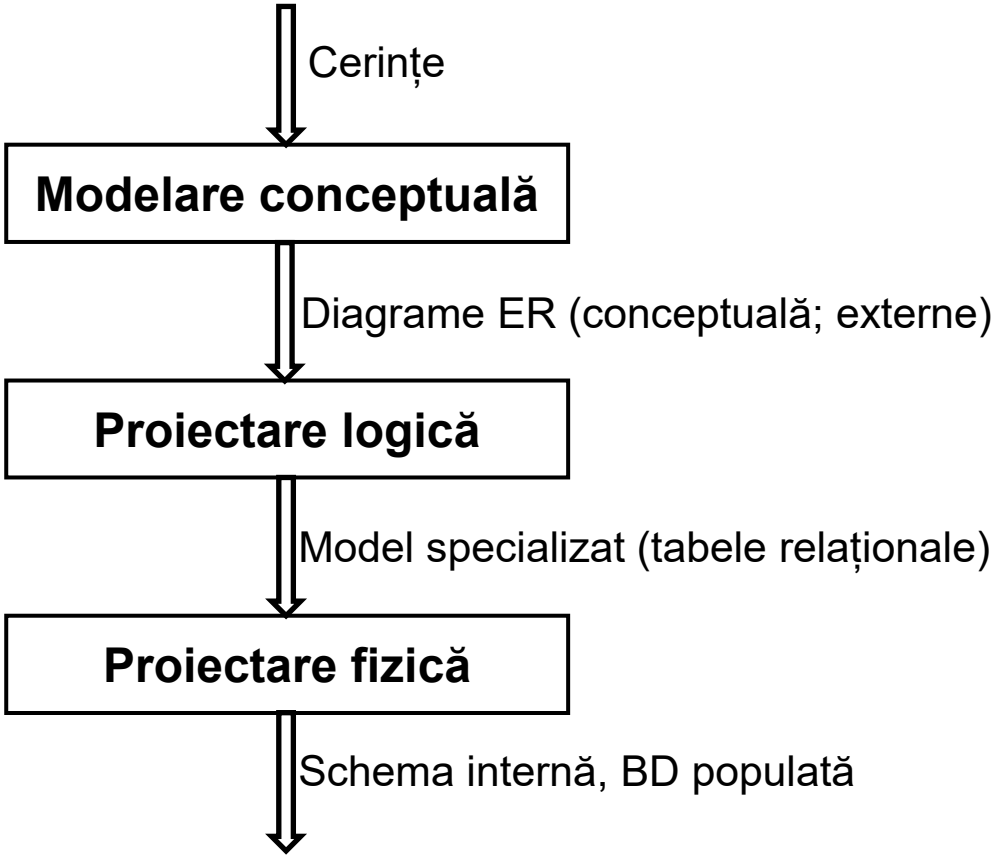
- Sisteme comerciale
  - Oracle (Oracle9i, Oracle 10g, Oracle 11g, Oracle 12c, Oracle 18c)
  - IBM (DB2, Informix)
  - Microsoft (SQL Server 2005, 2008, 2012, 2014, 2016, 2017)
  - Terradata
  - MySQL (open source)
  - Facilităţi:
    - Data management
    - BI (Business Inteligence)
    - e-business

- **Clasificare SGBD-uri**
  - *Clasificare după modelul de date*
    - Majoritatea sistemelor actuale sunt realizate în modelul de date relaţional sau în modelul de date obiectual
      - Dezvoltarea continuă a acestor modele a condus către o nouă categorie de baze de date, numite *obiect-relaţionale*, care combină caracteristicile modelului relaţional cu cele ale modelului obiectual
    - Mai sunt încă în funcţiune baze de date în modele mai vechi (*modelul ierarhic* sau *modelul reţea*)
  - *Clasificare după numărul de utilizatori*:
    - Majoritatea sistemelor actuale sunt *multiutilizator*, ce permit accesul concurent (în acelaşi timp) a mai multor utilizatori la aceeaşi bază de date
    - Un număr redus de sisteme de baze de date sunt de tip *monoutilizator*, adică suportă accesul doar al unui singur utilizator (la un moment dat)

- *Clasificare după numărul de staţii pe care este stocată baza de date:*
  - *Sistem de baze de date centralizat:*
    - este un sistem de baze de date în care datele şi sistemul de gestiune sunt stocate pe o singură staţie (calculator)
    - poate suporta unul sau mai mulţi utilizatori, dar, în orice situaţie, datele şi sistemul de gestiune rezidă în întregime pe o singură staţie
  - *Sistem de baze de date distribuit* (*Distributed Database System*):
    - poate avea atât datele, cât şi sistemul de gestiune, distribuite în mai multe staţii interconectate printr-o reţea de comunicaţie

1.7. Etapele dezvoltării unei baze de date

- Pentru a crea o bază de date operaționlă trebuie definite schemele (externe, conceptuală, internă) și apoi popularea bazei de date



- Modelarea conceptuală: folosește datele și cerințele enunțate pentru a produce diagrame entitate-legătură (ERD - Entity Relationship Diagram)
  - Cea conceptuală trebuie să reprezinte toate datele și toate formatele de prezentare a acestora
  - Cele externe reprezintă o utilizare particulară a datelor (de exemplu un raport)
- Proiectarea logică transformă modelul conceptual într-un format specific unui model specializat (de ex. cel relațional)
  - Activități:
    - Conversie: transformă diagramele în tabele
    - Normalizare
- Proiectarea fizică vizează eficiența implementării
  - Minimizarea timpului de răspuns folosind cât mai puține resurse
  - Se utilizează: indecși și politici de amplasare a datelor

1.8. Modele de date

- *Un model* este o abstractizare a unui sistem, care captează cele mai importante concepte, relevante din punct de vedere al scopului pentru care se defineşte modelul respectiv
  - Tehnica de identificare a trăsăturilor caracteristice esenţiale ale unui sistem se numeşte *abstractizare*

- *Un model de date* stabileşte regulile de organizare şi interpretare a unei colecţii de date

- In proiectarea bazelor de date se folosesc, de regulă, mai multe modele de date:
  - *modele conceptuale de nivel înalt*
  - *modele specializate*

- *Un model conceptual de nivel înalt al datelor* conţine o descriere concisă a colecţiilor de date care modelează activitatea dorită, fără să detalieze modul de reprezentare sau de prelucrare a datelor
  - Sunt analizate natura datelor și modul de utilizare a acestora
  - Sunt identificate datele ce vor fi gestionate și se împart în grupuri logice (entități)
  - Se identifică legăturile între aceste grupuri

- *Modelele specializate de date* (de ex. *modelul relaţional*) impun anumite structuri speciale de reprezentare a mulţimilor de entităţi şi a asocierilor dintre acestea, structuri pe baza cărora sunt dezvoltate sistemele de gestiune a bazelor de date:
  - într-un astfel de model de date, o bază de date este reprezentată printr-o schemă conceptuală (logică) specifică

- Trecerea de la modelul conceptual de nivel înalt la un model de date specific asigură corespondenţa dintre schema conceptuală de nivel înalt a bazei de date şi schema conceptuală specifică modelului de date respectiv

Modelul entitate-legătură (entitate-asociere)

- ***Entitatea*** desemnează un obiect care face parte dintr-o ***clasă (mulţime) de entităţi:***
  - toate aceste obiecte sunt similare ca structură, dar pot fi deosebite prin proprietăţi specifice (***atribute***)
- Exemple:
  - ***Mulţimea clienţilor unui magazin***
    - Fiecare client reprezintă o entitate a acestei clase şi are următoarele atribute: *cod personal, nume, adresa*
    - Fiecare entitate (client) este identificabilă prin atributul *cod personal*
  - ***Mulţimea produselor aflate în magazin***
    - Fiecare produs poate fi caracterizat prin: *cod produs, denumire, preţ unitar*
    - Fiecare entitate (produs) este identificabilă prin atributul *cod produs*
  - ***Mulţimea departamentelor unei firme***
    - Orice departament se caracterizează prin: *cod departament*, *denumire*, ***şef***
    - Fiecare departament este identificabil prin atributul *cod departament,*
  - ***Mulţimea angajaţilor unei firme***
    - Fiecare angajat se caracterizează prin *număr legitimaţie, nume, departament* și este identificabil prin atributul *număr legitimaţie*

- **Legătura** (asocierea sau relația) între mai multe clase de entităţi E_1, ..., E_n (nu neapărat distincte) presupune existenţa unei mulţimi de valori (e_1, ..., e_n), unde e_i este mulţimea valorilor atributelor unei entităţi din clasa E_i
- In practică întâlnim frecvent legături între două clase de entităţi (n=2) (***legături binare)***
- Exemple:
  - (111,"Personal",5001,5001,"Petrescu",111)
    - Primele trei valori sunt preluate din clasa de entităţi Departament, următoarele trei din clasa Personal
    - Se observă uşor că această mulţime de valori este un element al produsului cartezian al celor două mulţimi: Departamente şi Personal
    - După ce eliminăm valorile care se repetă (această operaţie este numită ***proiecţie***) obţinem:

      (111,"Personal",5001,"Petrescu")

- Tipuri de legături binare:
  - ***legături de tip 1:1*** (*one-to-one*)
    - legătura în cazul unei evidenţe a personalului, care indică faptul că un departament este condus de un şef (un departament nu poate fi condus de mai mulţi şefi iar o persoană nu poate conduce mai multe departamente)
  - ***legături de tip 1:N*** (*one-to-many*)
    - legătura între *Angajat* şi *Departament*
    - legătura între *Factura* şi *Client*
  - ***legături de tip M:N*** (*many-to-many*)
    - legătura între *Client* şi *Produs*

- **Diagrama entitate legătură (entitate relație – ER)**
  - Datele obținute în modelarea conceptuală se reprezintă grafic sub forma unei diagrame (*ERD- Entity Relationship Diagram*)
    - Este o reprezentare grafică a unei colecții de date
    - Este un instrument de proiectare
    - Este independentă de implementare
    - Nu există standarde !
  - Se face identificarea entităților, a atributelor asociate și apoi a legăturilor (asocierile) între entități
  - Realizează o descriere grafică a entităţilor, folosind următoarele convenţii:
    - descrierea entităţii se face într-un dreptunghi
    - numele entităţii este scris pe prima linie cu litere mari
    - atributele sunt scrise cu litere mici pe liniile următoare sau se enumeră separat sub forma *Nume_entitate(Lista atribute)*

- Reprezentarea entităților se face după următoarele reguli:
  - numele unei entităţi trebuie să fie un substantiv comun sugestiv (la singular)
  - nu pot exista două entităţi cu acelaşi nume sau o entitate cu două nume diferite
  - pentru fiecare entitate se va da o descriere completă a atributelor sale (semnificaţia, domeniul de valori)
  - fiecare entitate va avea obligatoriu un atribut sau o combinaţie de atribute care să identifice în mod unic fiecare instanţă a entităţii (identificator unic - UID)
  - fiecare entitate va fi implicată în cel puțin o legătură

- Reprezentarea atributelor se face după următoarele reguli:
  - atributele sunt scrise cu litere mici, fiecare pe câte o linie sau sub formă de listă
  - numele atributelor sunt unice în cadrul unei asocieri
  - atributele care sunt identificatori unici sunt precedate de caracterul # și se pun imediat după numele entității
  - atributele obligatorii sunt precedate de caracterul *
  - atributele opţionale sunt precedate de caracterul ˚

- Exemplu:

  STUDENT(# cnp, * nume, * prenume,
  * data_nasterii, ˚ telefon, ˚ email)

STUDENT

cnp
* nume
* prenume
* data_nasterii
˚ telefon
˚ email

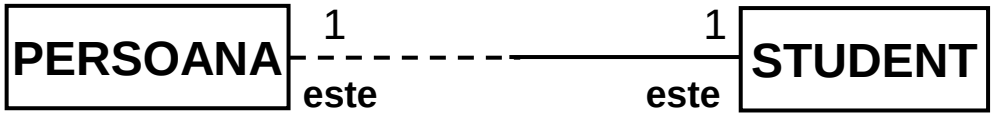
- Asocierea (relantionship) (legătura) reprezintă un raport care există între entităţi şi este exprimată prin utilizarea unor verbe care să descrie interdependenţa dintre acestea

- O asociere (binară) este o legătură bidimensională între două entități sau între o entitate și ea însăși
- O asociere se reprezintă printr-o linie care uneşte cele două entităţi, la fiecare din cele două capete scriindu-se *numele asocierii*
- Numele asocierii este format dintr-un cuvânt sau grup de cuvinte care conţin obligatoriu un verb şi exprimă modul în care entitatea din acea parte a legăturii este asociată cu entitatea din cealaltă parte a asocierii

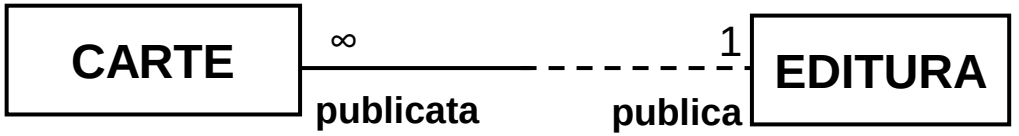
- *Opţionalitatea asocierii*
  - Proprietate a unei legături (asocieri) care exprimă câte dintre instanţele entităţii A ***pot*** sau ***trebuie*** să se asocieze instanţelor entităţii B şi reciproc
  - Există asocieri obligatorii, ce se reprezintă cu linie continuă
  - Există asocieri opționale, ce se reprezintă cu linie întreruptă

- *Cardinalitatea asocierii*
  - Proprietate a unei asocieri care exprimă câte instanţe ale entităţii A sunt asociate cu instanţele entităţii B şi reciproc
    - Trebuie specificată pentru ambele părți ale unei asocieri (legături)
  - Tipuri de cardinalități
    - Una și numai una (sau variante), reprezentată prin linie continuă sau întreruptă, la care se mai poate adăuga simbolul 1 (unu)
    - Una sau mai multe (sau variante), reprezentată printr-un simbol grafic sugestiv sau prin simbolul ∞(infinit)
    - In descrierea asocierii, cuvintele scrise italic exprimă cardinalitatea sau gradul asocierii

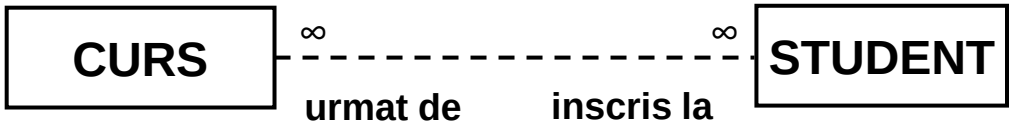
- Exemple:
  - Asociere (legătură) 1 : 1 (one to one) PERSOANA-STUDENT



  - Asociere (legătură) N : 1 (many to one) CARTE-EDITURA



  - Asociere (legătură) M : N (many to many) CURS-STUDENT



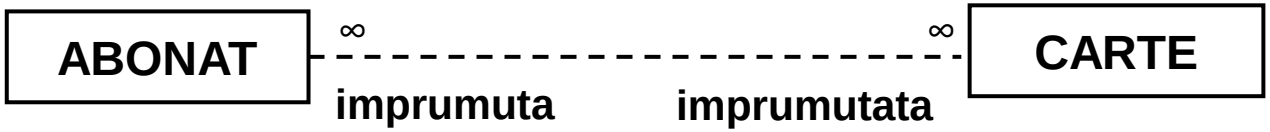
Studiu de caz: biblioteca (model simplificat)

- Entități:
  - Obiectul de activitate al unei biblioteci sunt cărțile
- CARTE (# cota, * titlu, * autor, * gen, * editura, * an, * disponibile, ˚ isbn, ˚ observatii)

- Entitatea GEN (# IDgen, * denumire, ˚ observatii)
  - Avem o legătură (asociere) GEN-CARTE de tipul 1:N

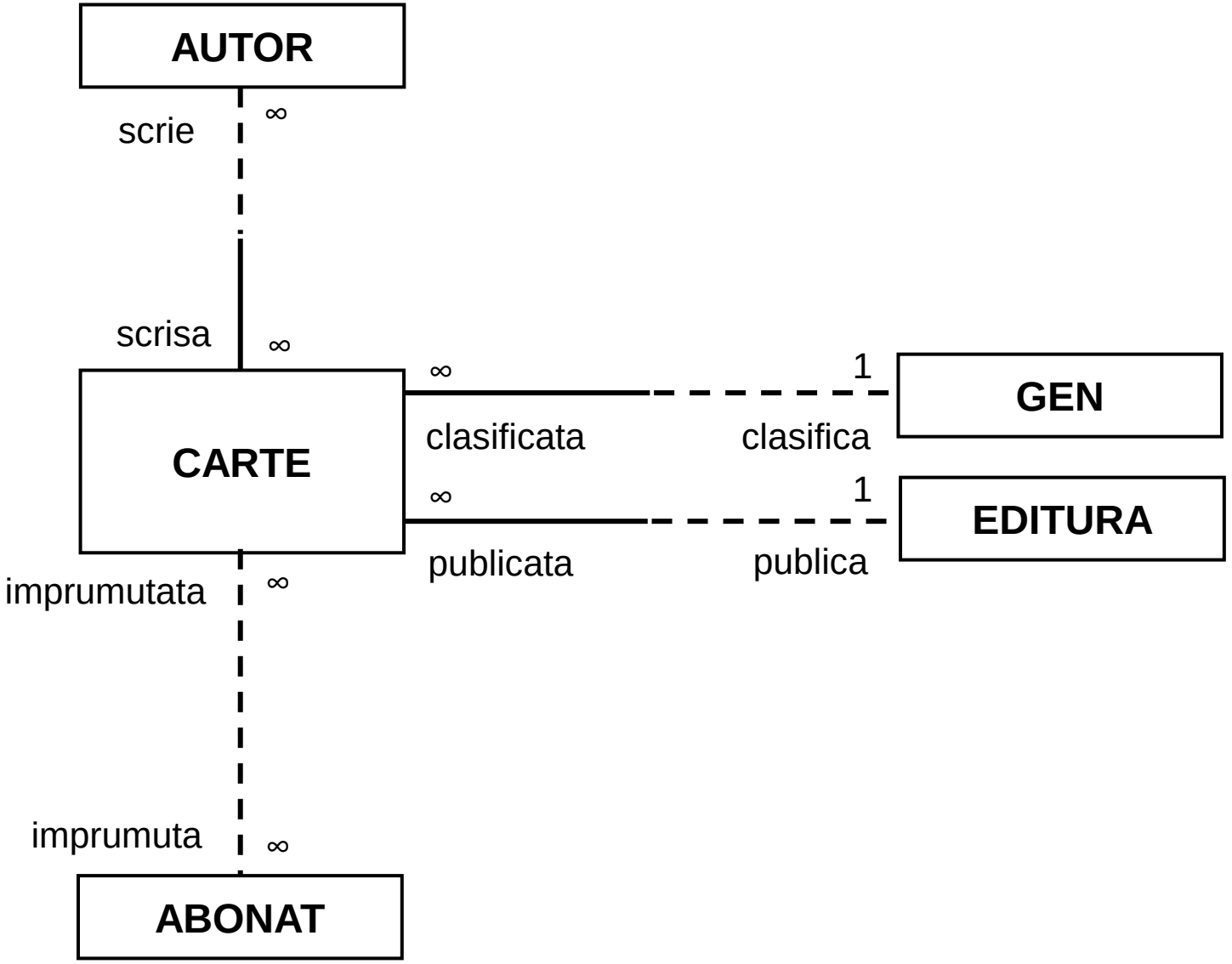
- Entitatea EDITURA(# IDed, *denumire, * adresa, ˚ cod_cncsis)
  - Avem o legătură (asociere) EDITURA-CARTE de tipul 1:N

- Cărţile, clasificate sau nu, sunt împrumutate abonaților
- Entitatea ABONAT(# cnp, * nume, * prenume, * adresa)
  - Avem o legătură (asociere) ABONAT-CARTE de tipul M:N



- Cărțile sunt căutate și după autor
- Entitatea AUTOR(# IDautor, * nume, * prenume, * naţionalitatea)
  - Avem o legătură (asociere) AUTOR-CARTE de tipul M:N

Diagrama ER pentru bibliotecă:



- Implementarea asocierilor (legăturilor) N:M
  - Este un pas opțional în această etapă
    - Poate fi făcut și în etapa de proiectare logică
  - In proiectarea modelului conceptual nu sunt acceptabile asocierile M:N (care apar deseori între entităţi) datorită gradului mare de ambiguitate generat de acestea
  - Se face rafinarea modelului iniţial prin eliminarea asocierilor N:M şi înlocuirea lor cu asocieri 1:N
  - O asociere N:M va fi înlocuită prin două asocieri 1:N, stabilite între entităţile iniţiale şi o a treia entitate, nou introdusă, numită *entitate de legătură* sau *entitate de intersecţie*

- Inlocuirea unei legături (asociere) N:M existente între entităţile A şi B se va realiza astfel:
  - Se introduce o nouă entitate C, având un identificator unic format din identificatorii unici ai celor două entităţi, plus alte atribute suplimentare ce descriu asocierea respectivă
    - Nu este permisă mutarea unor atribute din entitățile inițiale în entitatea de intersecție, deoarece modelul va deveni redundant
  - Asocierile care pleacă din entitatea de intersecție sunt întotdeauna obligatorii. La capătul celălalt își păstrează opționalitatea asocierii inițiale
  - Asocierea dintre A şi C este de tipul 1:N
  - Asocierea dintre B şi C este de tipul 1:N
  - Aceste asocieri 1:N dintre entităţile iniţiale şi entitatea de intersecţie C se reprezintă grafic prin adăugarea unei bare orizontale la capătul dinspre entitatea C
- In cazul nostru:

  EVIDENTA AUTORI (# IDautor, # Cota)

  EVIDENTA IMPRUMUT (# CNP, # Cota, * data_imprumut, * perioada)

Diagrama ER pentru bibliotecă (rafinată):