

2. PROIECTAREA BAZELOR DE DATE.

- Modelul conceptual și rolul său
- Definiția ERD
- Ce este o entitate și cum se reprezintă ea într-un ERD
- Ce este o instanță
- Ce sunt și cum se stabilesc atributele unei entități
- Care sunt tipurile de atribute și cum se reprezintă ele în ERD
- Cum se stabilesc relațiile între entități
- Care sunt caracteristicile unei relații
- Ce tipuri de relații pot exista între entități și cum se reprezintă ele în ERD
- Cum se rezolvă relațiile many-to-many

2.1.1. DATE. INFORMAȚII. CUNOȘTINȚE.

- De multe ori cuvântul "informație" este folosit fără a-i înțelege clar sensul, diferența dintre date, informații, cunoștințe.
- În general, conținutul gândirii umane operează cu următoarele concepte:
 - Date
 - Informații
 - Cunoștințe
 - Înțelepciune



- **Date** - constau în material brut, fapte, simboluri, numere, cuvinte, poze fără un înțeles de sine stătător, neintegrate într-un context, fără relații cu alte date sau obiecte. Ele se pot obține în urma unor experimente, sondaje, etc.
- **Informații** - prin prelucrarea datelor și găsirea relațiilor dintre acestea se obțin informații care au un înțeles și sunt integrate într-un context. Datele organizate și prezentate într-un mod sistematic pentru a sublinia sensul acestora devin informații. Pe scurt informațiile sunt date prelucrate. Informațiile se prezintă sub formă de rapoarte, statistici, diagrame, etc.
- **Cunoștințele** sunt colecții de date, informații, adevăruri și principii învățate, acumulate de-a lungul timpului. Informațiile despre un subiect reținute, înțelese și care pot fi folosite în luarea de decizii, formează judecăți și opinii, devin cunoștințe. Cu alte cuvinte, cunoștințele apar în momentul utilizării informației.
- **Înțelepciunea** este un nivel superior de înțelegere a faptelor și informațiilor. Vorbim despre înțelepciune atunci când pe baza informațiilor și cunoștințelor pe care le deținem putem discerne între bine și rău, formulăm opinii, păreri personale etc. Înțelepciunea este o caracteristică a oamenilor, calculatoarele neputând opera decât cu primele trei concepte.



EXEMPLU

- Date:
 - "42" "iepuri" "4.00pm"
 - "76" "mere" "0740112233"
 - "20euro" "mare"
- Informații:
 - Sunt **42 mere** în această cutie și fiecare măr este ronțăit de către **iepuri**.
 - Costul biletului până la **mare** este de 20 euro și călătoria durează 76 minute cu trenul.
 - Numărul meu de telefon este **0740112233**. Sună-mă la ora 4.00pm!
- Aceste informații adaugă un context și un sens datelor.
- Cunoștințe:
 - În ultimii cinci ani, recolta de mere din Moldova a crescut cu 10% în fiecare an. Se prevede că și în acest an recolta va crește cu încă 10% și de aceea trebuie să găsim o piață de desfacere pentru 10% mai multe mere.
- Informațiile din ultimii câțiva ani au fost folosite pentru a estima creșterea producției de mere și necesitatea unei piețe mai mari de desfacere. Predicția făcută este *cunoștință*, cu alte cuvinte, au fost folosite informațiilor deținute.



2.1.2. COLECTAREA ȘI ANALIZAREA DATELOR. MODELUL CONCEPTUAL

- Primul pas în realizarea unei aplicații de baze de date este analiza datelor și realizarea unei scheme conceptuale - **model conceptual** - a acestora.
- În această etapă sunt analizate natura și modul de utilizare ale datelor. Sunt identificate datele care vor trebui memorate și procesate, apoi se împart aceste date în grupuri logice și se identifică relațiile care există între aceste grupuri.



- Analiza datelor este un proces care necesită mult timp, însă este o etapă obligatorie. Fără o analiză atentă a datelor și a modului de utilizare a acestora, vom realiza o bază de date care nu întrunește cerințele beneficiarului.
- Costurile modificării acestei baze de date este mult mai mare decât costurile pe care le-ar fi implicat etapa de analiză și realizare a modelului conceptual.
- Modificarea modelului conceptual este mult mai ușoară decât modificarea unor tabele deja existente, care eventual conțin și o mulțime de date.
- Ideea de bază a analizei datelor și a construirii modelului conceptual este "să măsori de două ori și să tai o singură dată".



- Informațiile necesare realizării modelului conceptual se obțin folosind metode convenționale precum **interviarea oamenilor din cadrul organizației și studierea documentelor folosite.**
- Odată obținute aceste informații ele trebuie reprezentate într-o **formă convențională** care să poată fi ușor înțeleasă de toată lumea. O astfel de reprezentare este ***diagrama entități-relații***, numită și *harta relațiilor*, sau *ERD-u* (Entity Relationship Diagram).
- Aceste scheme sunt un instrument util care ușurează comunicarea dintre specialiștii care proiectează bazele de date și programatori pe de o parte și beneficiari, pe de altă parte. Aceștia din urmă pot înțelege cu ușurință o astfel de schemă, chiar dacă nu sunt cunoscători în domeniul IT.



- Caracteristici ale ERD-urilor:
 - sunt un instrument de proiectare;
 - sunt o reprezentare grafică a unui sistem de date;
 - oferă un model conceptual de înalt nivel al bazelor de date;
 - sprijină înțelegerea de către utilizatori a datelor și a relațiilor dintre acestea sunt independente de implementare.
- Principalele elemente care intră în componența unui ERD precum și convențiile de reprezentare a acestora vor fi prezentate în continuare.



2.1.3. ENTITĂȚI. INSTANȚE. ATRIBUTE. IDENTIFICATOR UNIC

- O **entitate** este un lucru, obiect, persoană sau eveniment care are semnificație pentru afacerea modelată, despre care trebuie să colectăm și să memorăm date.
- O entitate poate fi un lucru real, tangibil precum o clădire, o persoană, poate fi o activitate precum o programare sau o operație, sau poate fi o noțiune abstractă.



- O entitate este reprezentată în ERD printr-un dreptunghi cu colțurile rotunjite.
- Numele entității este întotdeauna un *substantiv la singular* și se scrie în partea de sus a dreptunghiului cu *majuscule*.



Exemple de entități și modul de reprezentare



- O entitate este de fapt o clasă de obiecte și pentru orice entitate există mai multe **instanțe** ale sale. O instanță a unei entități este un obiect, persoană, eveniment, particular din clasa de obiecte care formează entitatea.
- De exemplu, elevul x din clasa a IX-a A de la Liceul de Informatică din localitatea Y este o instanță a entității ELEV.
- Pentru a preciza o instanță a unei entități, trebuie să specificăm unele caracteristici ale acestui obiect, să-l descriem (precizăm de exemplu numele, clasa, școala, etc).



- După ce am identificat entitățile trebuie să descriem aceste entități în termeni reali, adică să le stabilim **atributele**. Un atribut este orice detaliu care servește la identificarea, clasificarea, cuantificarea, sau exprimarea stării unei instanțe a unei entități.
- Atributele sunt informații specifice ce trebuie cunoscute și memorate.
- De exemplu, atributele entității **ELEV** sunt nume, prenume, adresa, număr de telefon, adresa de e-mail, data nașterii, etc.
- În cadrul unui ERD, atributele se scriu imediat sub numele entității, cu litere mici. Un atribut este un *substantiv la singular*



ELEV

cnp
* nume
* prenume
* data_nașterii
* adresa
o telefon
o e-mail

- Un atribut poate fi
 - **obligatoriu** - fiecare instanță a entității respective trebuie să avem o valoare pentru acel atribut, de exemplu, este obligatoriu să cunoaștem numele elevilor.
 - **opțional** - putem avea instanțe pentru care nu cunoaștem valoarea atributului respectiv. De exemplu, atributul **email** al entității **ELEV** este opțional, un elev putând să nu aibă adresă de e-mail.
- Un atribut obligatoriu este precedat în ERD de un asterisc *, iar un atribut opțional va fi precedat de un cerculeț o.



- Atributele care definesc în mod unic instanțele unei entități se numesc **Identificatori unici** (UID).
- UID-ul unei entități poate fi compus dintr-**un singur atribut**, precum codul numeric personal ce poate fi un identificator unic pentru entitatea **ELEV**.
- În alte situații, identificatorul unic este compus dintr-**o combinație de două sau mai multe atribute**.
- De exemplu combinația dintre titlu, numele autorului și data apariției poate forma unicul identificator al entității **CARTE**.
- Combinația titlu și nume autor nu era suficientă, deoarece pot exista de exemplu mai multe volume scrise de Mihai Eminescu având toate titlul „Poezii”, dar apărute la date diferite.



CARTE

titlu
autor
data_aparitiei
* format
* numar_pagini

- Atributele care fac parte din identificatorul unic al unei entități vor fi precedate de semnul diez #.
- **Atributele din UID sunt întotdeauna obligatorii.** Semnul # este suficient, nu mai trebuie pus și un semn asterisc în fața acestor atribute.



- Valorile unor attribute se pot modifica foarte des, ca de exemplu atributul „vârstă”.
- Spunem în acest caz că avem de a face cu un ***atribut volatil***.
- Dacă valoarea unui atribut însă se modifică foarte rar sau deloc (de exemplu, „data nașterii”) acesta este un atribut ***non-volatil***.
- Este de preferat să folosim attribute non-volatile atunci când acest lucru este posibil.



FIŞA 1



2.1.4. RELAȚII ÎNTRE ENTITĂȚI

- În lumea reală, obiectele nu există izolat. Percepem obiectele din lumea reală doar în conexiune cu alte obiecte, de exemplu vom spune "pământul se învârte în jurul soarelui", "el este medic", etc.
- Așadar, după identificarea entităților și atributelor lor, trebuie puse în evidență relațiile care există între aceste entități, modul în care acestea comunică între ele.
- O **relație** este o asociere, legătură, sau conexiune existentă între entități și care are o semnificație pentru afacerea modelată.
- Orice relație este bidirecțională, legând două entități sau o entitate cu ea însăși. De exemplu, elevii studiază mai multe materii, o materie e studiată de către elevi.



- Orice relație este caracterizată de următoarele elemente:
 - numele relației opționalitatea relației
 - gradul (cardinalitatea) relației.
- Exemplu: relația existentă între entitățile JUCĂTOR și ECHIPĂ.
- Vom spune:

Un **JUCĂTOR** joacă într-o ECHIPĂ.

Numele relației este: *joacă*,



OPȚIONAL - OBLIGATORIU

- Pentru a stabili **opționalitatea** relației trebuie să răspundem la următoarele întrebare: Un jucător **trebuie** să joace într-o echipă? Se **poate** ca un jucător să nu joace în nicio echipă?
- Dacă acceptăm că toți jucătorii trebuie să joace într-o echipă relația este obligatorie sau mandatorie și vom spune:

Un JUCĂTOR *trebuie* să joace într-o ECHIPĂ.

- Dacă însă acceptăm că există jucători care nu joacă în nicio echipă (de exemplu li s-a terminat contractul și în momentul de față nu mai joacă la nicio echipă), atunci relația este opțională. În acest caz vom spune:

Un JUCĂTOR *poate* juca la o ECHIPĂ.



CARDINALITATEA

- **Cardinalitatea** relației este dată de numărul de instanțe ale entității din partea dreaptă a relației care pot intra în relație cu o instanță a entității din partea stângă a relației.
- Adică va trebui să răspundem la întrebări de genul: La câte echipe poate juca un jucător? Răspunsurile posibile sunt **unul și numai unul**, sau **unul sau mai mulți**.
- Vom spune:
Un JUCĂTOR trebuie/ poate să joace la **o ECHIPĂ și numai una**. sau
Un **JUCĂTOR** trebuie/ poate să joace la **una sau mai multe** ECHIBE.
- Cea mai realistă variantă a relației dintre **JUCĂTOR și ECHIPĂ** este așadar: Un **JUCĂTOR** poate să joace la o **ECHIPĂ** și numai una.
- Am precizat însă mai înainte că orice relație este bidirecțională. Relația dintre **ECHIPĂ și JUCĂTOR** o putem enunța astfel:
La o ECHIPĂ trebuie să joace unul sau mai mulți **JUCĂTORI**.



CONVENȚII DE REPREZENTARE A RELAȚIILOR

- În cadrul diagramei entități-relații, o relație va fi reprezentată printr-o linie ce unește cele două entități.
- Deoarece o relație este bidirecțională, linia ce unește cele două entități este compusă din **două segmente distincte**, câte unul pentru fiecare entitate.
- Tipul segmentului ce pleacă de la o entitate ne va indica **opționalitatea** relației dintre această entitate și entitatea aflată în cealaltă parte a relației.
- Dacă acest segment este **continuu** este vorba de o relație **obligatorie**, o linie **întreruptă** indică o relație **opțională**.

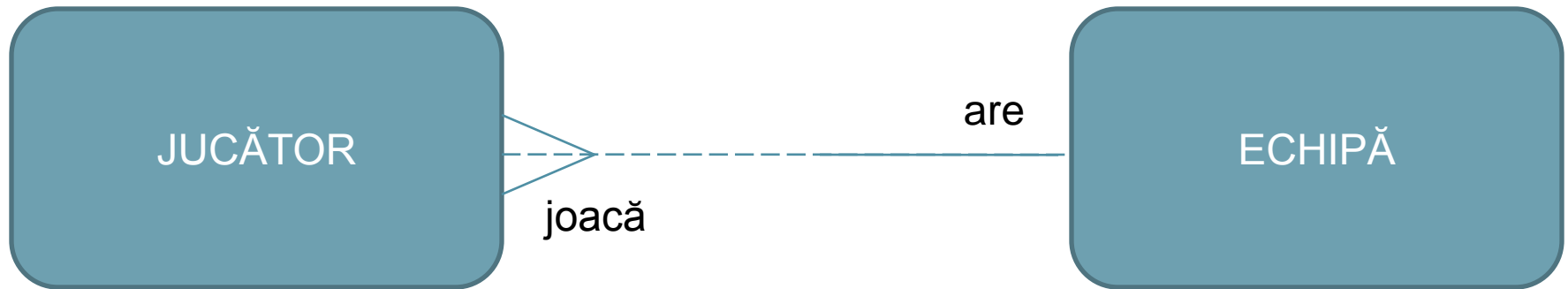


- În figura de mai jos, segmentul ce pleacă de la entitatea JUCĂTOR este **întrerupt** ceea ce înseamnă că un jucător **poate** juca la o echipă, adică relația este opțională.
- Segmentul ce pleacă dinspre entitatea ECHIPĂ este **continuu**, deci la o echipă **trebuie** să joace jucători.



Reprezentarea relațiilor









- Modul în care o linie se termină spre o entitate este important. Dacă se termină printr-o linie simplă, înseamnă că o instanță și numai una a acestei entități este în relație cu o instanță a celeilalte entități.
- Linia de la JUCĂTOR la ECHIPĂ se termină în partea dinspre ECHIPĂ cu o **linie simplă**, deci un jucător joacă la **o** echipă și **numai una**.





- Dacă linia se termină cu trei linii (picior de cioară) înseamnă că mai multe instanțe ale entității pot corespunde unei instanțe a celeilalte entități. În exemplul anterior linia de la **ECHIPĂ** la **JUCĂTOR** se termină cu ***piciorul de cioară***, înseamnă că unei instanțe a entității **ECHIPĂ** îi corespund mai multe instanțe ale entității **JUCĂTOR**, adică o echipă are ***unul sau mai mulți*** jucători.



Caracteristica relației	Valoare	Mod de reprezentare
Numele relației	un verb	se scrie deasupra relației
Opționalitatea	Relație obligatori (TREBUIE)	linie continuă 
	Relație opțională (POATE)	linie întreruptă 
Cardinalitatea	una și numai una	linie simplă 
	una sau mai multe	picioar de cioară 



TIPURI DE RELAȚII

- Variantele de relații ce pot exista între două entități sunt:
 - **relații one-to-one** - acest tip de relație este destul de rar întâlnit. Uneori astfel de relații pot fi modelate transformând una dintre entități în atribut al celeilalte entități.

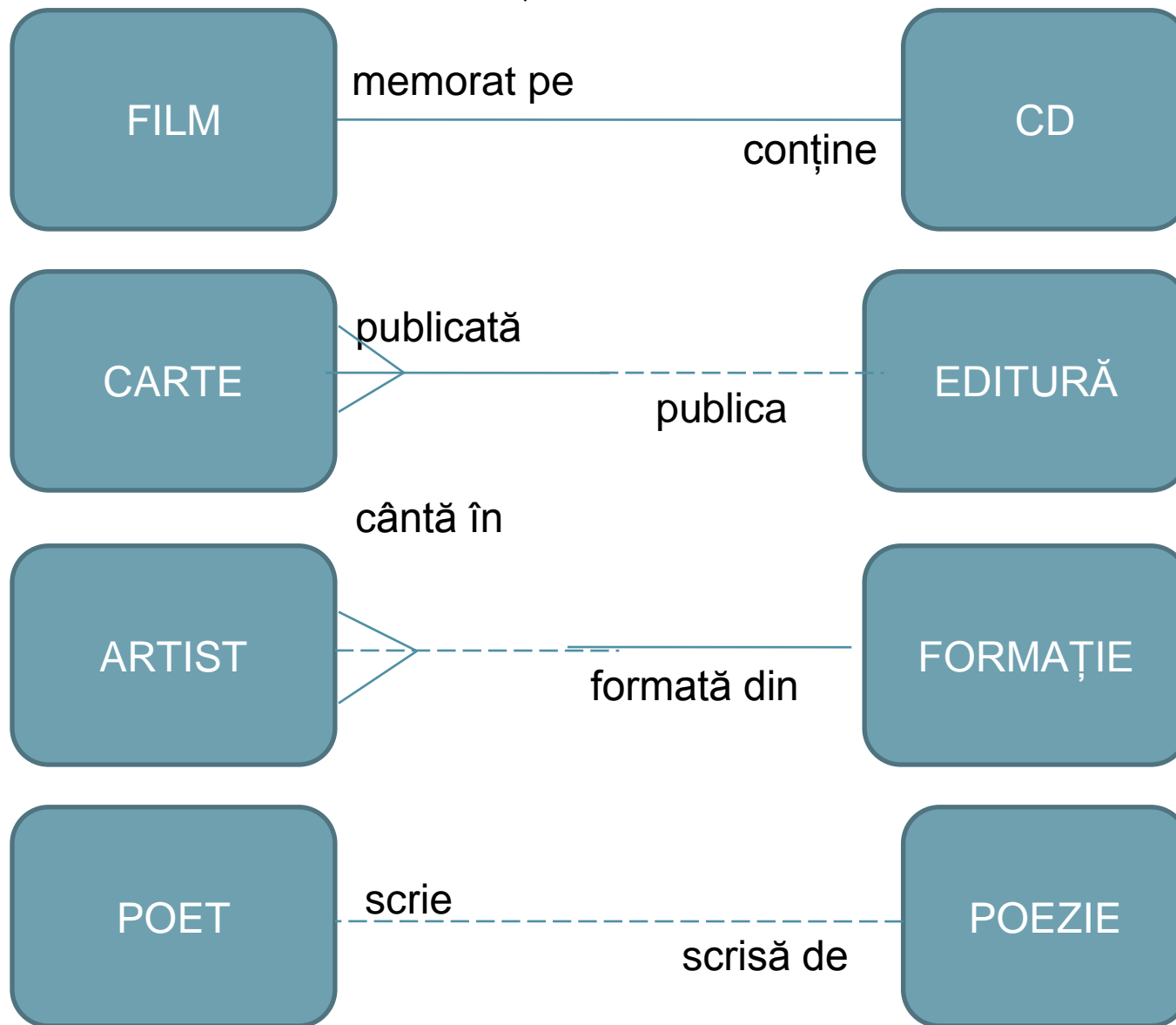


Tipuri de relații one – to – one



- **Relații one – to – many** sunt cele mai întâlnite tipuri de relații.

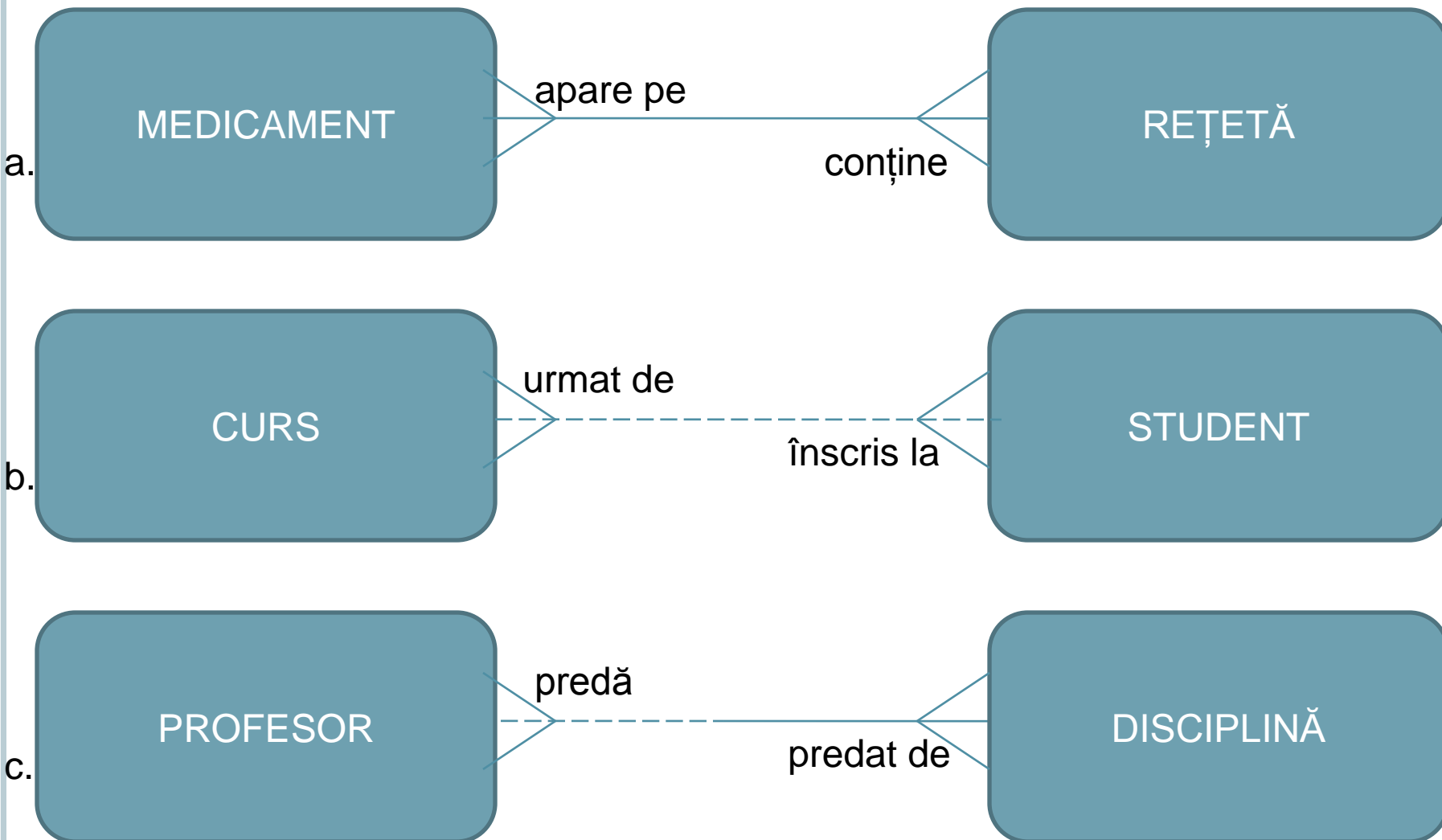
Tipuri de relații one – to – many



- **relații many-to-many** – sunt relații care apar în prima fază a proiectării bazei de date, însă ele trebuie să fie ulterior eliminate.
- Exemple de relații many-to-many.
 - La punctul **b** am considerat că un curs poate apărea pe oferta de cursuri a unei facultăți, însă poate să nu fie aleasă de niciun student de aceea un curs **poate** fi urmat de unul sau mai mulți studenți.
 - Invers, este posibil ca un student să fi terminat studiile și să se pregătească pentru susținerea examenului de licență și de aceea el nu mai frecventează nici un curs.
 - La punctul c, un profesor angajat al unei școli **trebuie** să predea cel puțin o disciplină.
 - Iar o disciplină din planul de învățământ trebuie să fie predată de cel puțin un profesor.

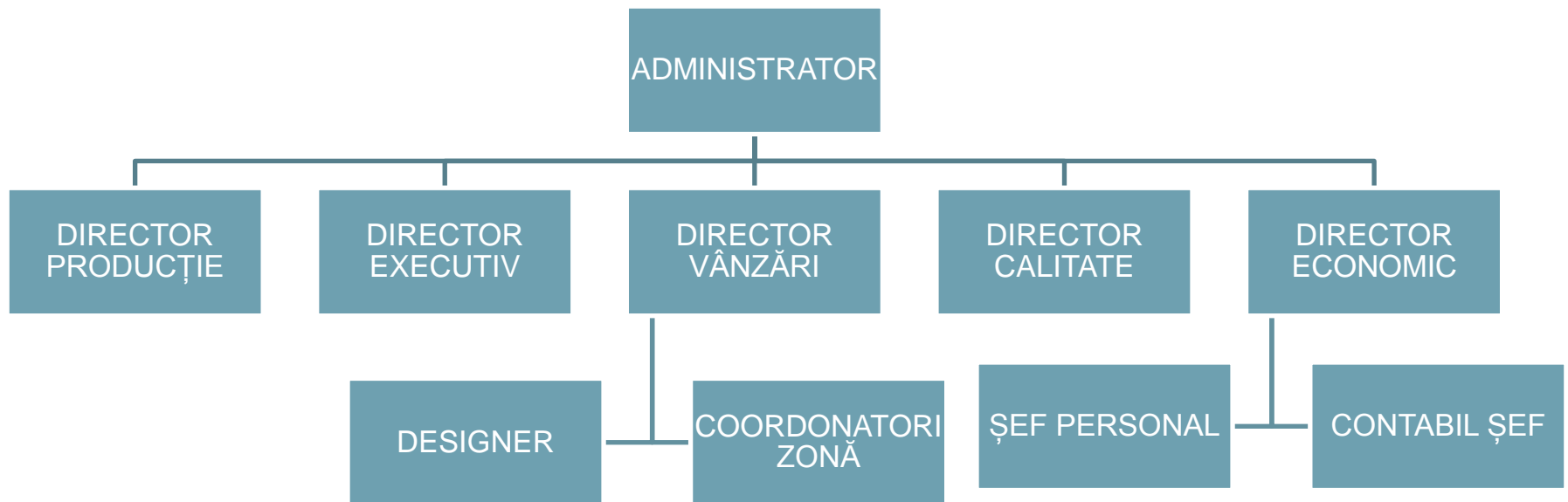


Exemple de relații many – to – many



RELAȚII IERARHICE. RELAȚII RECURSIVE.

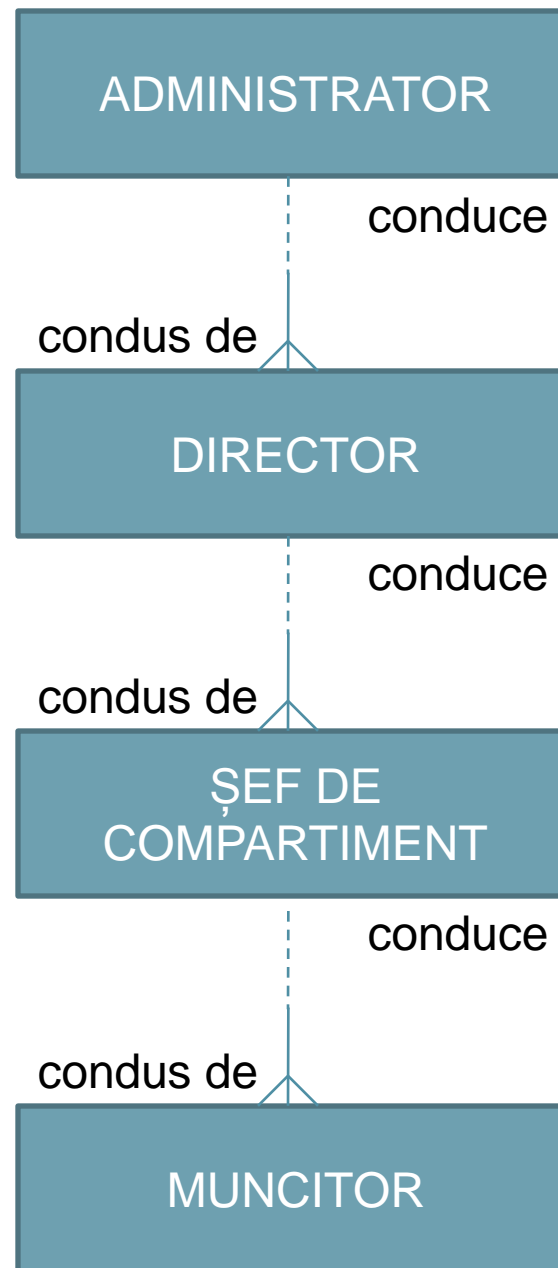
- Structura personalului într-o firmă oarecare:



Organigrama unei firme

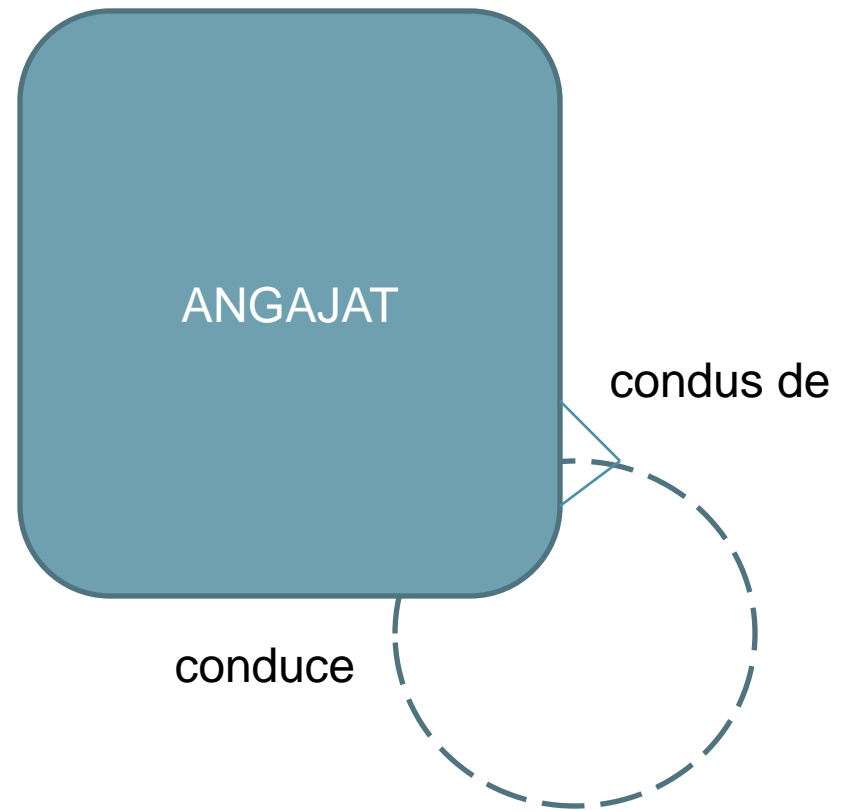


- Un model de proiectare a unei astfel de structuri într-o bază de date ar fi cea din figură:



Implementarea unei structuri ierarhice

- Problema este că fiecare tip de angajat este de fapt un angajat și există foarte multe atribute comune tuturor acestor entități ca de exemplu:
 - nume,
 - prenume,
 - adresă,
 - telefon,
 - e-mail,
 - data nașterii.
- Vom putea modela această structură cu ajutorul unei singure entități numită **ANGAJAT**.
- Fiecare angajat poate fi condus de către un alt angajat. Așadar vom avea o relație de la entitatea **ANGAJAT** la ea însăși.
- O astfel de relație se numește **relație recursivă**.

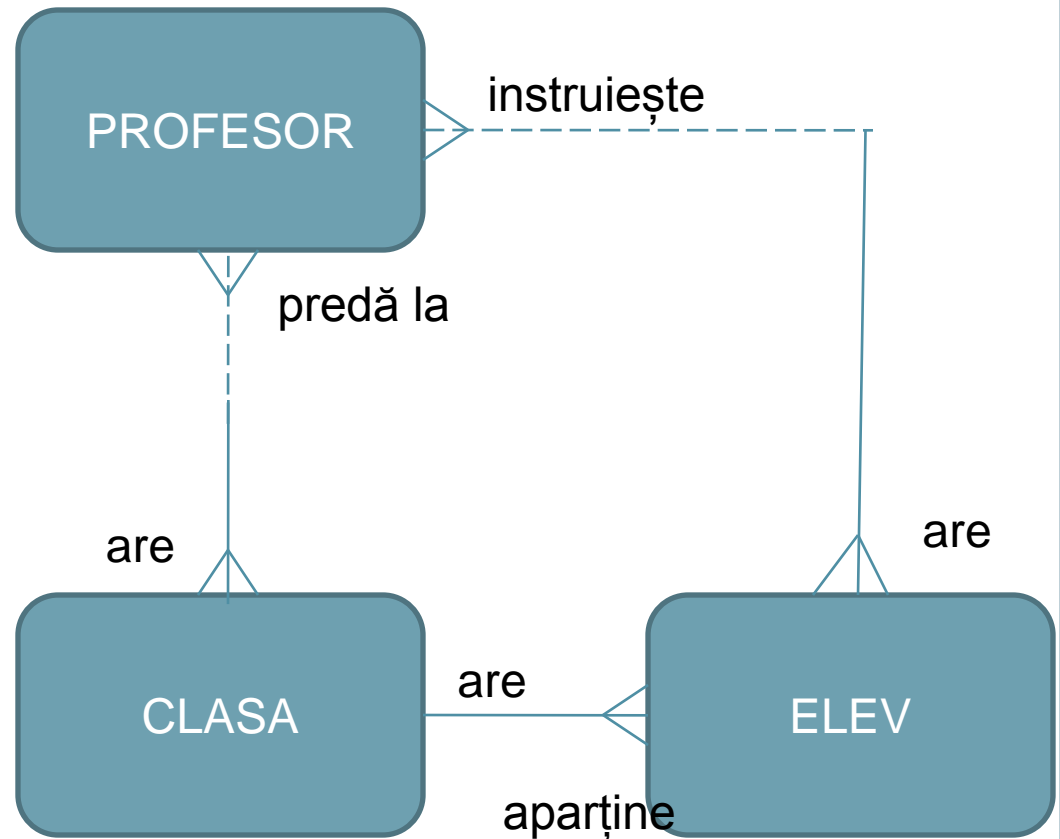


Implementarea unei structuri ierarhice folosind relații recursive



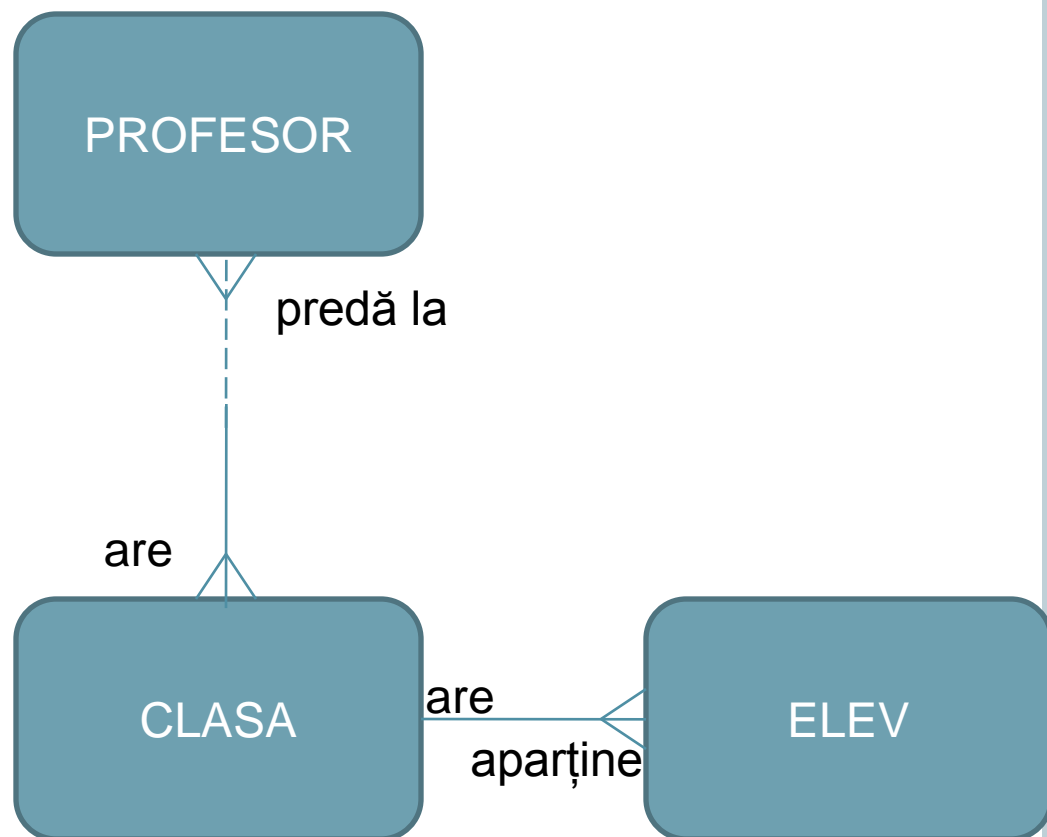
RELAȚII REDUNDANTE

- Atunci când o relație poate fi dedusă din alte relații, spunem că acea relație redundantă.



Relații redundante

- Se observă că un elev face parte dintr-o clasă, iar la acea clasă predau mai mulți profesori.
- Așadar relația dintre profesor și elev nu mai este necesară oarece putem deduce profesorii care îi predau unui elev, aflând profesorii clasei din care face parte elevul.
- Această relație poate fi deci eliminată,

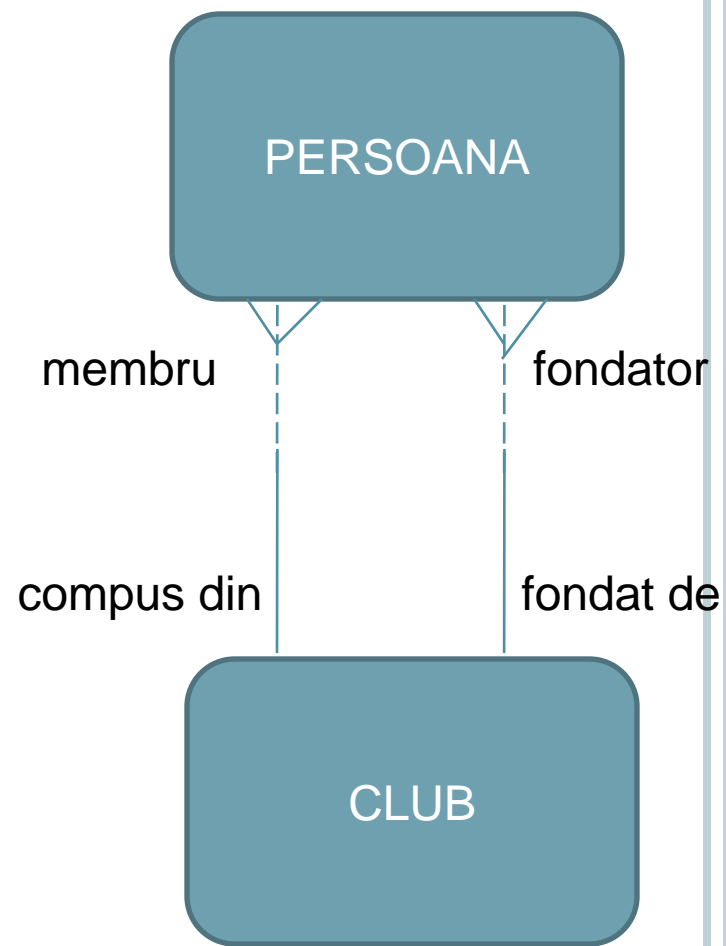


Eliminarea relației redundante

- Pentru situația anterioară, se pune întrebarea dacă un elev nu poate avea un profesor care nu predă la clasa în care învață? Desigur acest lucru depinde de situația pe care o modelăm. Dacă de exemplu ne propunem să memorăm date despre toți profesorii care îl instruiesc pe un elev, în cadrul orelor de curs, dar ne interesează de asemenea profesorii care îndrumă activitățile extracurriculare ale elevilor, atunci e posibil ca un elev să aibă și alți profesori decât cei de la clasă.
- În astfel de situații vom păstra relația dintre profesor și elev, adică vom opta pentru schema inițială, în care relația redundantă nu era eliminată.



- Există situația în care două entități pot fi legate prin mai multe relații diferite.
- Acestea nu sunt neapărat redundante. În exemplu sunt prezentate două relații diferite între entitățile **PERSOANA**, și **CLUB**.
- O persoană poate fi membră a mai multor cluburi și poate fi fondatorul unor cluburi.
- Faptul că o persoană a fondat un anumit club nu înseamnă obligatoriu că este membru al acelui club.
- De asemenea faptul că o persoană este membru al unui club nu înseamnă că este fondatorul lui. Așadar cele două relații nu se implică una pe cealaltă.

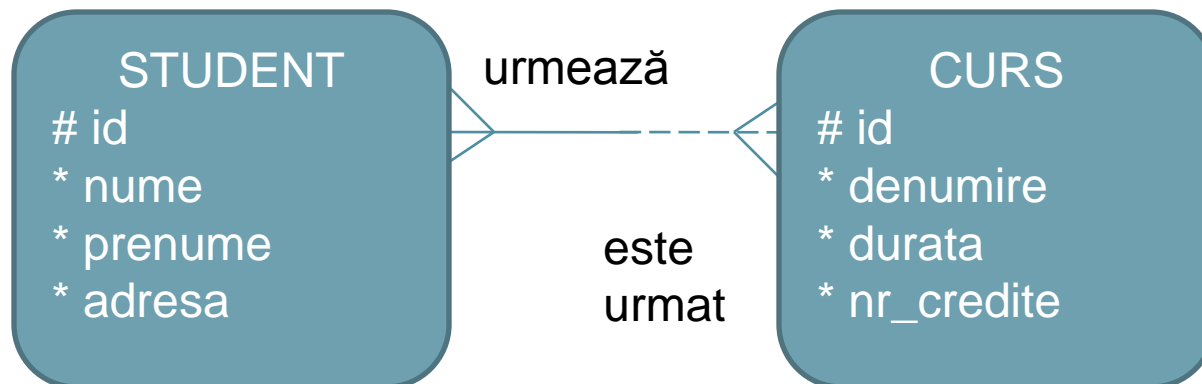


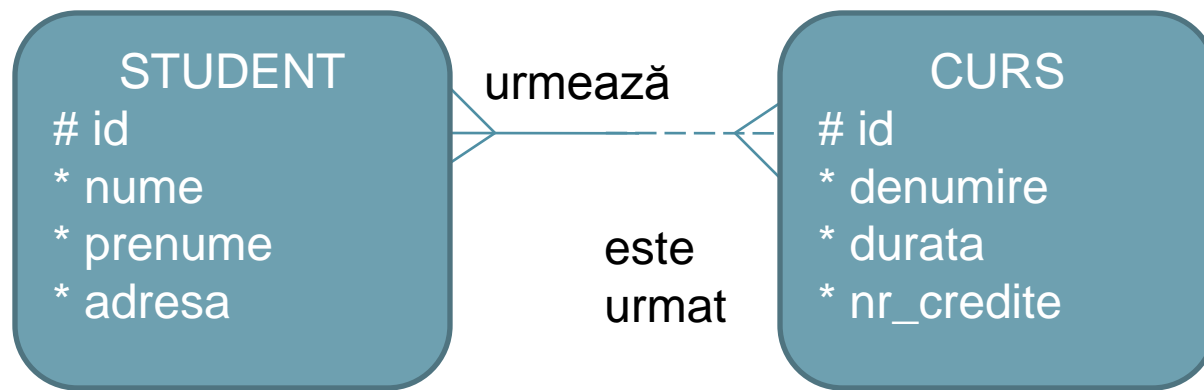
Relații multiple între entități



2.1.5. REZOLVAREA RELAȚIILOR MANY-TO-MANY

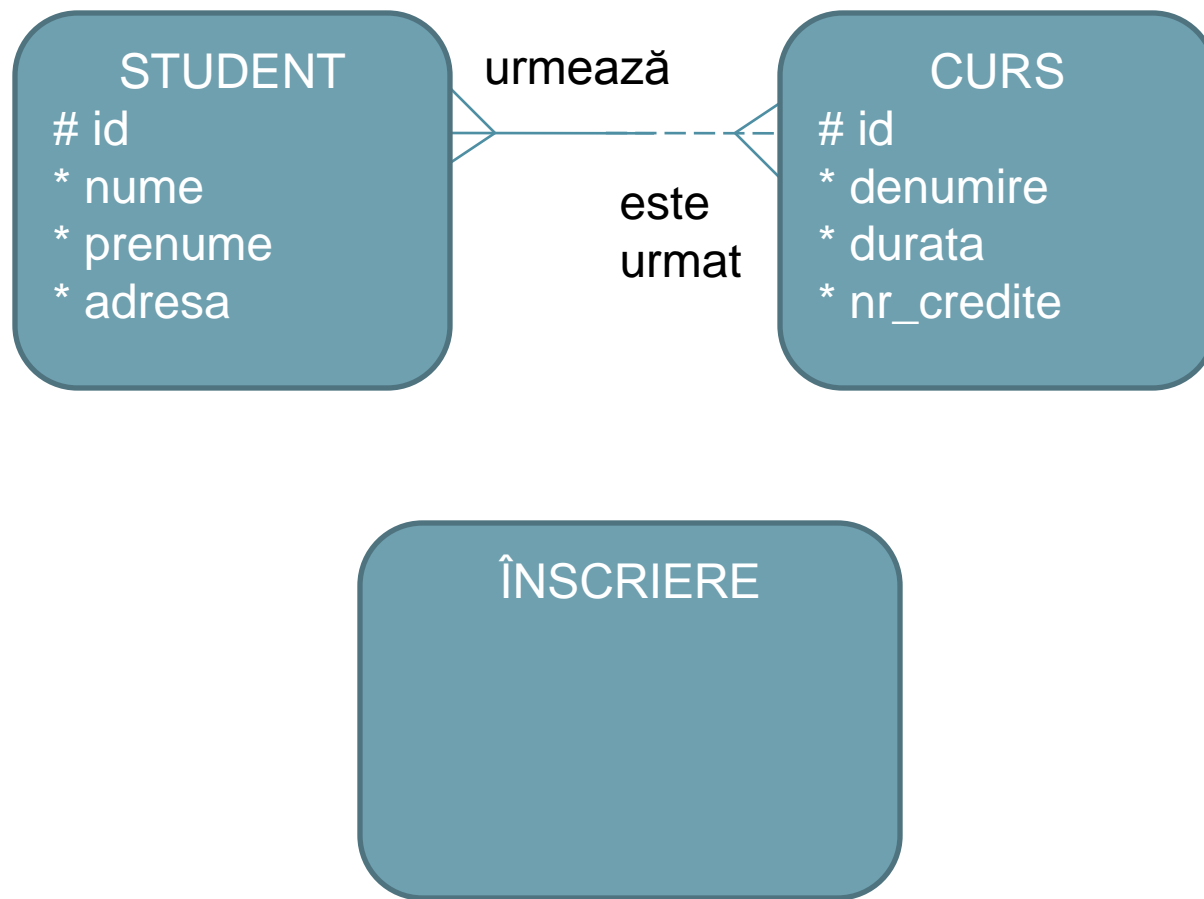
- relațiile many-to-many pot apărea într-o primă fază a proiectării bazei de date însă ele nu au voie să apară în schema finală.
- Să considerăm relația din figura de mai jos dintre entitățile **STUDENT** și **CURS**. Se știe că orice curs se termină în general cu un examen. Unde vom memora nota studentului la fiecare examen?





- Dacă încercăm să introducem atributul **NOTA** la entitatea **STUDENT**, nu vom ști cărei materii îi corespunde acea notă, întrucât unei instanțe a entității student îi corespund mai multe instanțe ale entității **CURS**.
- Invers, dacă încercăm să memorăm **NOTA** în cadrul entității **CURS**, nu vom ști cărui student îi aparține acea notă.
- Rezolvarea unei relații many-to-many constă în introducerea unei noi entități imită **entitate de intersecție**, pe care o legăm de entitățile originale prin câte o relație one-to-many.

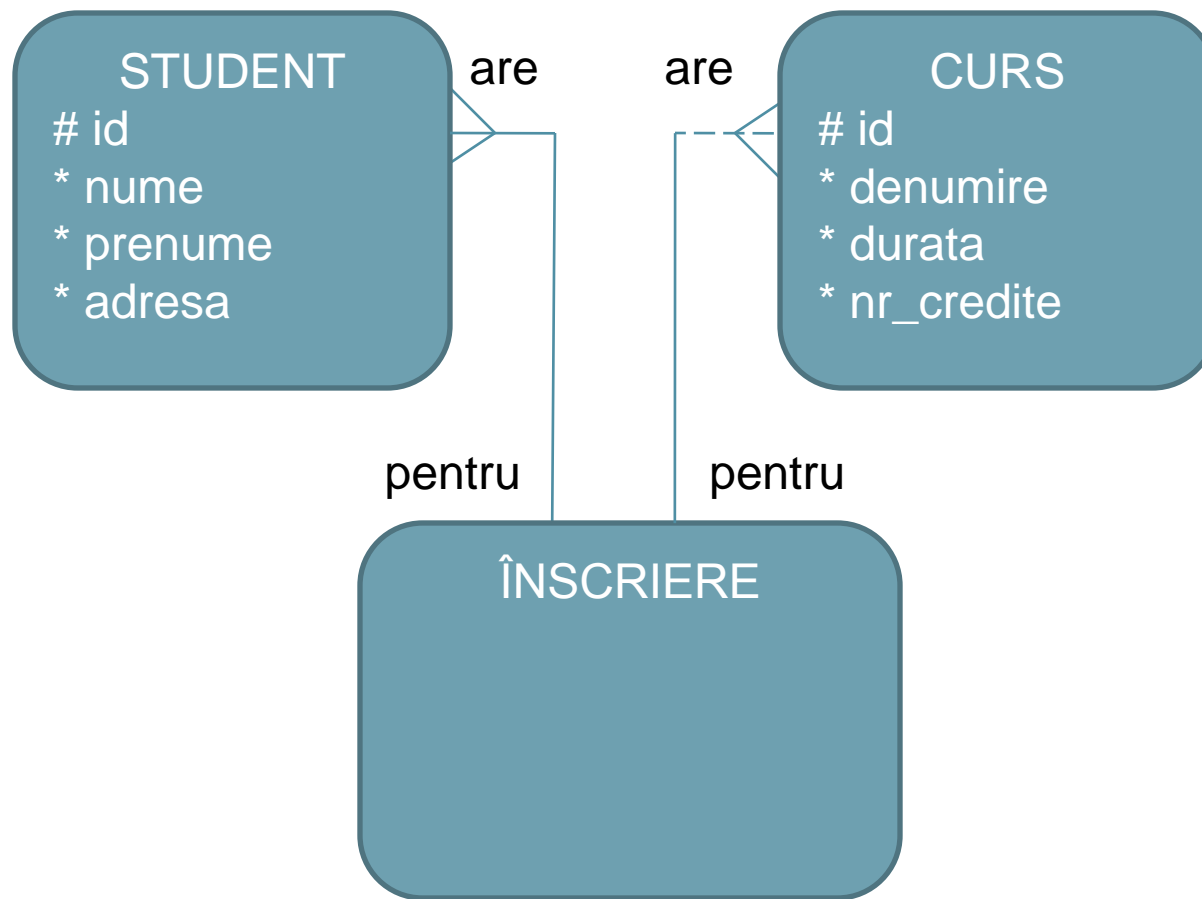




Rezolvarea relațiilor many-to-many, pasul 1

•Pașii în rezolvarea unei relații many-to-many sunt următorii:

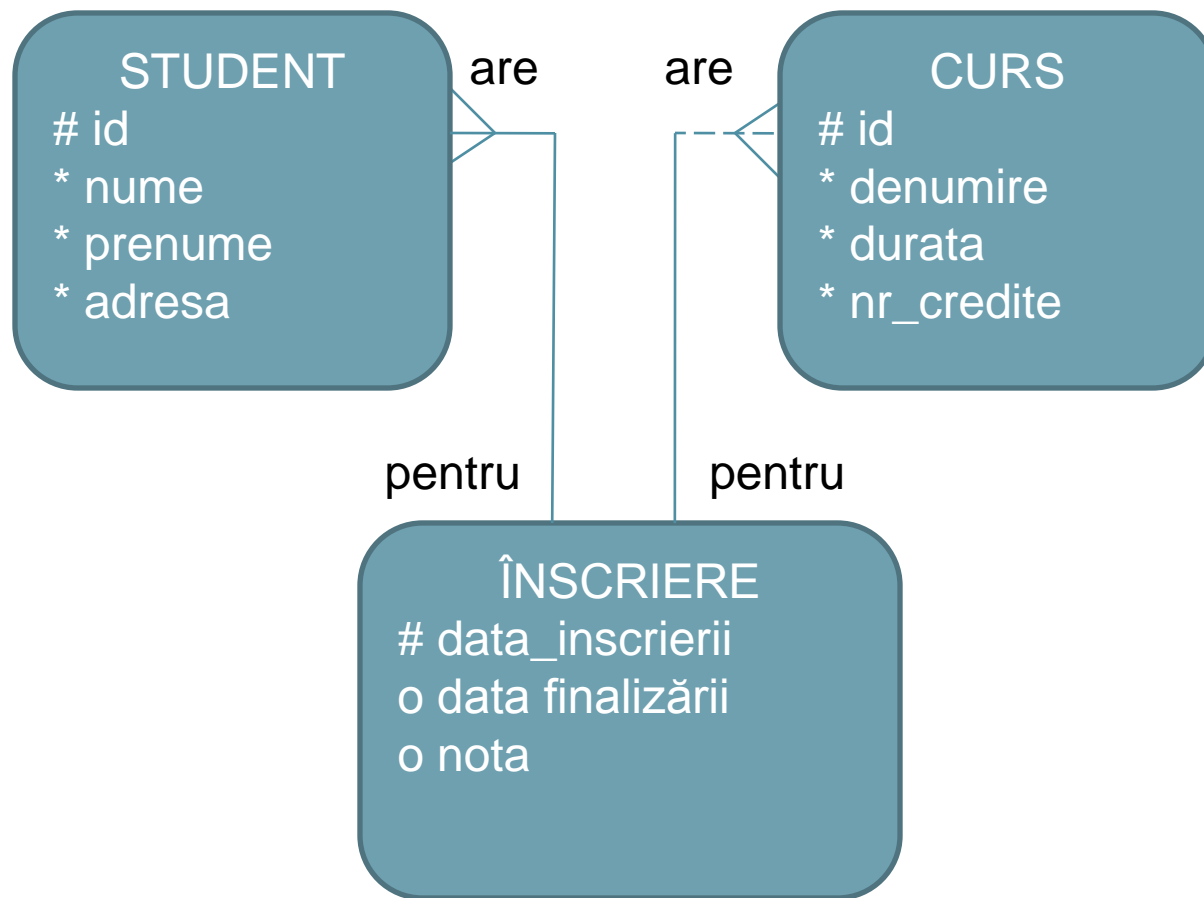
- se găsește entitatea de intersecție, pentru exemplul nostru vom introduce entitatea **ÎNSCRIERE**.



Rezolvarea relațiilor many-to-many, pasul 2

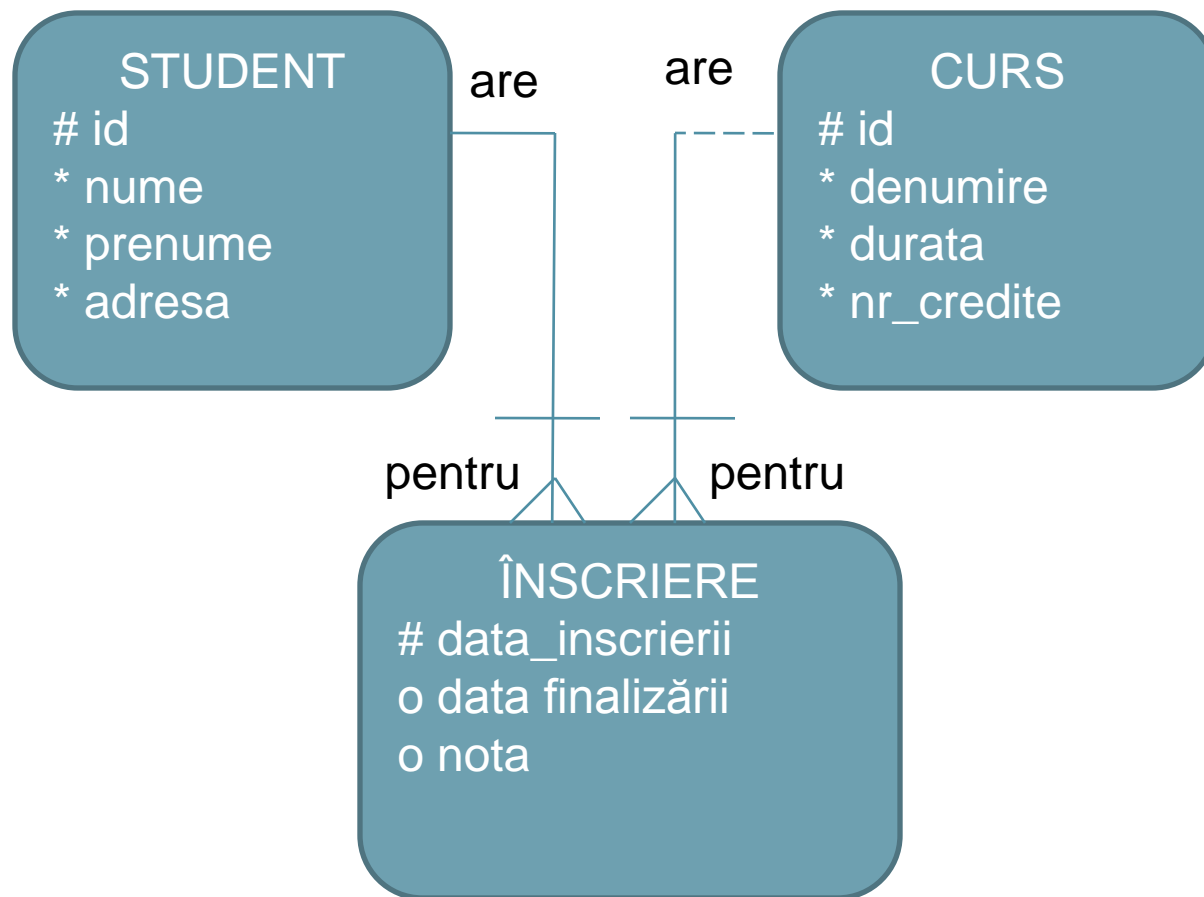
- crearea noilor relații
 - **opționalitatea:** relațiile care pleacă din entitatea de intersecție sunt întotdeauna obligatorii în această parte. În partea dinspre entitățile originale, relațiile vor păstra opționalitatea relațiilor inițiale.
 - **cardinalitatea:** ambele relații sunt de tip one-to-many, iar partea cu many va fi întotdeauna înspre entitatea de intersecție.
 - **numele** noilor relații.





Rezolvarea relațiilor many-to-many, pasul 3

- adăugarea de attribute în cadrul entității de intersecție, dacă acestea există, în exemplul nostru ne poate interesa să zicem data la care s-a înscris un student la un curs, data la care a finalizat cursul precum și nota obținută la sfârșitul cursului.



Rezolvarea relațiilor many-to-many, pasul 4

- stabilirea identificatorului unic pentru entitatea de intersecție: dacă entitatea de intersecție nu are un identificator unic propriu, atunci acesta se poate forma din identificatorii unici ai entităților inițiale la care putem adăuga attribute ale entității de intersecție.

- În exemplul nostru, identificatorul unic al entității de intersecție este format din:
 - id-ul studentului,
 - id-ul cursului
 - data înscrierii la curs.
- Faptul că identificatorul unic al unei entități preia, identificatorul unic din altă entitate cu care este legată este reprezentat grafic prin **bararea relației respective**, înspre entitatea care preia UID-ul celeilalte entități.



În această lecție am învățat despre:

- Modelul conceptual și rolul său
- Definiția ERD
- Ce este o entitate și cum se reprezintă ea într-un ERD
- Ce este o instanță
- Ce sunt și cum se stabilesc atributele unei entități
- Care sunt tipurile de atribute și cum se reprezintă ele în ERD
- Cum se stabilesc relațiile între entități
- Care sunt caracteristicile unei relații
- Ce tipuri de relații pot exista între entități și cum se reprezintă ele în ERD
- Cum se rezolvă relațiile many-to-many

SFÂRȘIT

