

# BAZE DE DATE

## CURS 2

### Modelarea entitate-relație

# PROIECTAREA BAZELOR DE DATE RELAȚIONALE

Modelarea entitate-  
relație (E/R)

Diagrame entitate-  
relație

Modelul relațional

# Modelarea entitate-relație (E/R)



# Modelarea Entitate-Relație

---

**Ce înțelegem prin model?**

**Model vs. Implementare?**

Ce trebuie să cunoască  
utilizatorii?

# Modelarea Entitate-Relație

---

**Model** = reprezentare a **obiectelor** și **evenimentelor** lumii reale și a **asocierilor** dintre ele.

- abstractizare asupra aspectelor semnificative ale unei „întreprinderi“, ale unui sistem real

**Model vs. Implementare?**

- Caz particular al deosebirii uzuale dintre logic și fizic.

# Modelarea Entitate-Relație

---

- 3 tipuri fundamentale de modele, care descriu aspecte:
    - Statice
    - Dinamice
    - Funcționale
- } ale procesului de modelat

# Modelarea Entitate-Relație

---

- **Ce este un model de date?**

# Modelarea Entitate-Relație

---

- **Model de date** = o colecție integrată de **concepte** necesare descrierii:
  - **datelor**,
  - **relațiilor** dintre ele,
  - **constrângerilor** existente asupra datelor sistemului real analizat.



# Modelarea Entitate-Relație

---

- Ce reprezintă **modelarea unei baze de date**?

# Modelarea Entitate-Relație

---

- **Modelarea unei baze de date** -> trecerea de la **percepția** unor fapte din lumea reală la reprezentarea lor prin **date**.
- **Modelul de date** trebuie:
  - să **reflecte** fidel fenomene ale lumii reale,
  - să **urmărească evoluția** acestei lumi și
  - să permită **comunicarea** dintre fenomenele lumii reale.

# Modelarea Entitate-Relație

---

- **Modelul de date** -> 3 componente :
  - o mulțime de **reguli** conform cărora sunt construite bazele de date (**partea structurală**);
  - o mulțime de **operații** permise asupra datelor, care sunt utilizate pentru **reactualizarea** sau **regăsirea** datelor (**partea de prelucrare**);
  - o mulțime de **reguli de integritate**, care asigură **coerența** datelor.

# Modelarea Entitate-Relație

---

- **Modelarea semantică a datelor** -> 4 etape:

1. Se identifică o **mulțime de concepte semantice** care sunt utile în descrierea lumii reale.
  - Se presupune că lumea reală (modelul real analizat) este formată din **entități** care au anumite **proprietăți**, că fiecare entitate are o **identitate**, că există **legături**, corelații între entități. Conceptul de corelație, ca și cel de entitate, este util, în mod intuitiv, la descrierea modelului.
2. Se caută o **mulțime de obiecte formale**, simbolice care sunt utilizate pentru reprezentarea conceptelor semantice anterioare.

# Modelarea Entitate-Relație

---

3. Se dau **reguli de integritate formale și generale** (constrângeri) care să reflecte restricțiile la care este supus modelul.
4. Se definește o **mulțime de operatori formali** prin care pot fi prelucrate și analizate obiectele formale.

# Exemple de scenarii

---

- Specificația unui model pornește de la un “*story*” (descrierea modelului)

## Exemple:

- Studenții unei facultăți participă la stagii oferite de către companii. Stagiile pot fi de tip *dezvoltare* sau *suport\_tehnic*. Fiecare stagiou are o dată de început și o dată de final. Un stagiou de tip *dezvoltare* necesită cunoașterea unor limbaje de programare. Pe perioada efectuării unui stagiou studenții pot lucra pe sarcini asociate unor proiecte ale companiei și sunt coordonați de către specialiști din compania respectivă. În timpul unui stagiou, un student lucrează cu anumite tehnologii (limbaje, sisteme).
- O firmă de distribuție are angajați de tip *funcționar* și *curier*. Firma deține autovehicule având diferite capacități de transport. Funcționarii firmei preiau comenzile de livrare ale clienților. Clienții comandă transportul de la o adresă de ridicare până la o adresă de recepție. Funcționarul care a preluat comanda unui client delegă curierilor efectuarea transportului corespunzător. În funcție de volumul livrat, transportul acestuia este delegat unuia sau mai multor curieri, ce vor folosi unul sau mai multe autovehicule. În cazul transportului cu autovehicule, sunt desemnați curierii care le vor conduce.

# Modelul entitate-relație



# Modelul Entitate-Relație

---

- **P. Chen, 1976**
- **Abordare a modelării semantice**
- Model de date conceptual, pentru a ușura proiectarea bazelor de date
- De **nivel înalt**, independent de platforma *hardware* utilizată și de tipul SGBD-ului
- Reprezentat grafic prin **diagrame E/R**



# Modelul Entitate-Relație

---

**Baza de date** -> mulțime de date ce **modelează** un sistem real format din:

- **Obiecte**
- **Legături între ele**

=> **Modelul E/R** împarte elementele unui sistem real în două categorii:

- **entități**
  - **relații** (legături, asocieri) între aceste entități.
- 
- Entitățile și legăturile -> **caracteristici (attribute)**.

# Modelul Entitate-Relație

---

- Conceptul de *relație, în sensul de asociere*, care intervine în definirea diagramei E/R

!=

- conceptul de *relație* care este *specific modelului relațional*.

# Diagrame entitate-relație



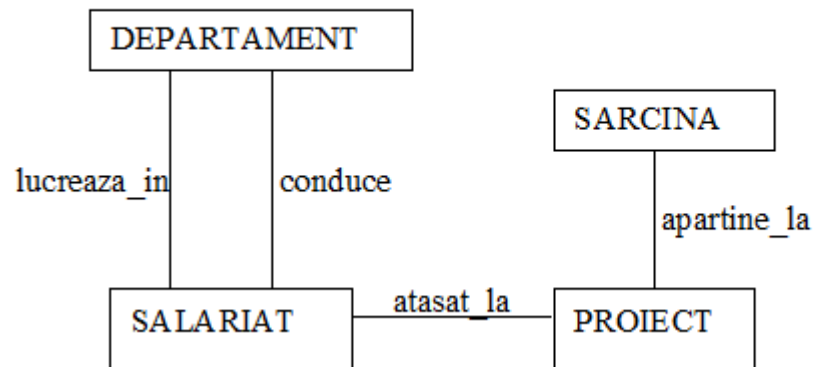
# Diagrame Entitate-Relație

---

- **Diagrama E/R** – model neformalizat pentru reprezentarea unui sistem din lumea reală.
  - Este un model de date conceptual de nivel înalt dezvoltat de Chen (1976).
- **Entitate**: persoană, loc, concept, activitate, eveniment care este semnificativ pentru ceea ce modelăm.

# Diagrame Entitate-Relație

---



# Diagrame Entitate-Relație

---

- **Entitățile** vor deveni **tabele** în modelul relațional.
- În general, entitățile se scriu cu **litere mari**.
- Entitățile sunt **substantive**, dar nu orice substantiv este o entitate.
- Pentru fiecare entitate este obligatoriu să se dea o **descriere detaliată**.
- Nu pot exista, în aceeași diagramă, două entități cu același nume, sau o aceeași entitate cu nume diferite.

# Diagrame Entitate-Relație

---

- **Cheia primară** este un **identificator unic** în cadrul entității, făcând distincție între valori diferite ale acesteia.
- Cheia primară:
  - trebuie să fie **unică** și **cunoscută** la orice moment;
  - trebuie să nu conțină informații descriptive, să fie **simplă**, fără ambiguități;
  - să fie **stabilă**;
  - trebuie să fie controlată de administratorul bazei de date;
  - să fie familiară utilizatorului.

# Diagrame Entitate-Relație

---

- **Relație** (asociere): o **comunicare** între două sau mai multe entități. Existența unei relații este **subordonată existenței entităților** pe care le leagă.
  - În modelul relațional, **relațiile** devin **tabele speciale** sau **coloane speciale** care referă chei primare.
  - Relațiile sunt **verbe**, dar nu orice verb este o relație.
  - Pentru fiecare relație este important să se dea o **descriere detaliată**.
  - În aceeași diagramă pot exista relații diferite cu același nume. În acest caz, le diferențiază entitățile care sunt asociate prin relația respectivă.
  - Pentru fiecare relație trebuie stabilită **cardinalitatea** (**maximă și minimă**) relației, adică numărul de tupluri ce aparțin relației.



# Diagrame Entitate-Relație

---

**poate** (cardinalitate maximă) → **trebuie** (cardinalitate minimă)

- Câți salariați **pot** lucra într-un departament? Mulți!
  - În câte departamente **poate** lucra un salariat? In cel mult unul!
- Relația SALARIAT\_lucraza\_in\_DEPARTAMENT are cardinalitatea maximă *many-one* (n:1).
- Câți salariați **trebuie** să conducă un departament? Cel puțin unul!
  - Câte departamente **trebuie** să conducă un salariat? Zero!
- Relația SALARIAT\_conduce\_DEPARTAMENT are cardinalitatea minimă *one-zero* (1:0).

# Diagrame Entitate-Relație

---

- **Atribut**: **proprietate** descriptivă a unei **entități** sau a unei **relații**.
  - Trebuie făcută distincția între **atribut** (devine coloană în modelele relaționale) și **valoarea acestuia** (devine valoare în coloane).
  - Atributele sunt **substantive**, dar nu orice substantiv este atribut.
  - Fiecărui atribut trebuie să i se dea o **descriere** completă (exemple, contraexemple, caracteristici).
  - Pentru fiecare atribut trebuie specificat **numele**, **tipul fizic** (*integer*, *float*, *char* etc.), **valori posibile**, **valori implicite**, **reguli de validare**

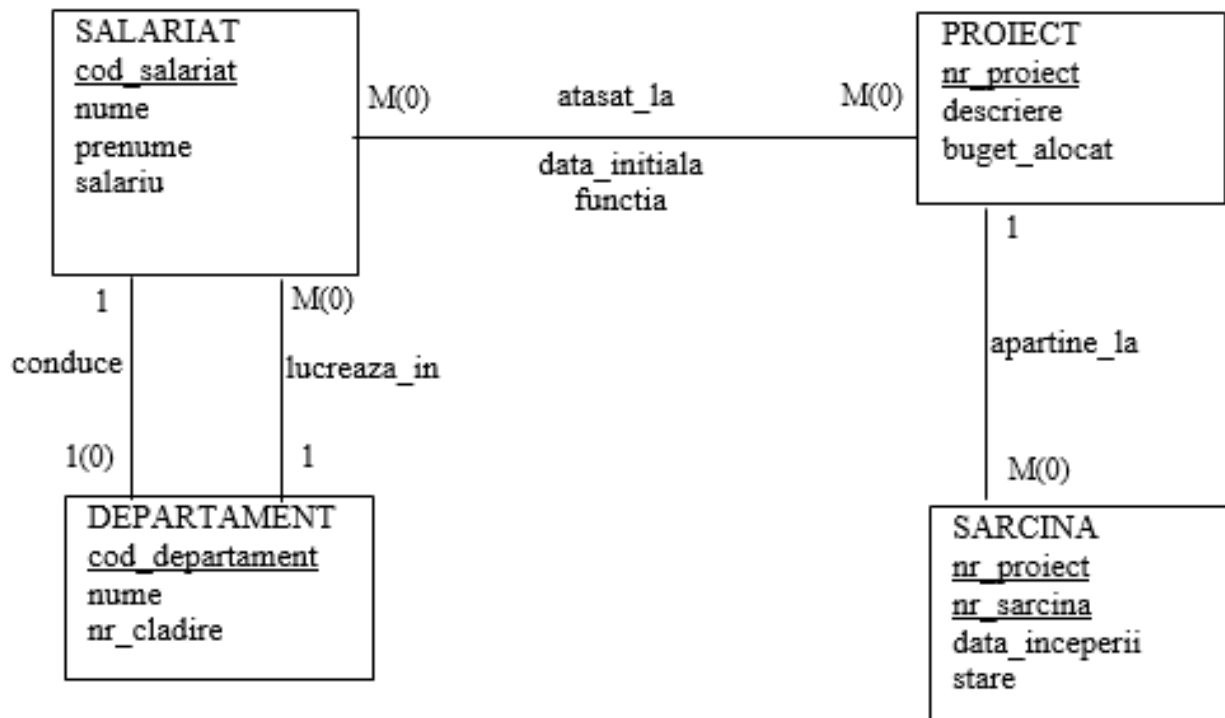
# Diagrame Entitate-Relație

---

## Reguli (neunice) pentru proiectarea unei diagrame E/R:

- **entitățile** sunt reprezentate prin **dreptunghiuri**;
- **relațiile** dintre entități sunt reprezentate prin **arce neorientate**;
- attributele care reprezintă **chei primare** trebuie **subliniate** sau marcate prin simbolul „#”, plasat la sfârșitul numelui acestor attribute;
- **cardinalitatea minimă** este indicată în paranteze, iar cardinalitatea **maximă** se scrie fără paranteze;
- nu este necesar să fie specificate, în cadrul diagramei, toate attributele.

# Diagrame Entitate-Relație



# Diagrame Entitate-Relație

---

## Cazuri speciale de entități, relații, attribute

Dependența

Moștenirea atributelor  
Specializare, generalizare

Relații recursive

Relații de tip 2, 3

Relație sau atribut?

Entitate sau relație?

Vezi curs și bibliografie!

# Diagrame Entitate-Relație

---

## Dependența

- **Entitate dependentă** – nu poate exista în mod independent (SARCINA depinde de PROIECT).
  - Cheia primară a unei entități dependente include cheia primară a sursei (*nr\_proiect*) și cel puțin un atribut caracteristic entității (*nr\_sarcina*).
  - (Entitatea dependentă se desenează prin dreptunghiuri cu linii mai subțiri.)

# Diagrame Entitate-Relație

---

## Moștenirea atributelor Specializare, generalizare

- **Moștenirea atributelor.**
  - **Subentitate** (subclasă) – submulțime a unei alte entități, numită **superentitate** (superclasă) (SALARIAT < — > PROGRAMATOR).
  - Subentitatea se desenează prin **dreptunghiuri incluse în superentitate**.
  - Există o relație între o subentitate și o superentitate, numită **ISA**, care are cardinalitatea maximă 1:1 și minimă 1:0.
  - **Cheile primare, attributele și relațiile unei superentități sunt valabile pentru orice subentitate.** Afirmția reciprocă este falsă.

# Diagrame Entitate-Relație

---

## Moștenirea atributelor Specializare, generalizare

- **Generalizare.**
  - Din entități similare care au mai multe **atribute comune** se pot crea **superentități**.
  - Aceste superentități conțin attributele comune, iar attributele speciale sunt asignate la subentități. Pentru noile superentități se introduc **chei primare artificiale**.
- **Specializare.**
  - După valorile unor attribute clasificatoare se pot determina **clase**.
  - Un grup de **subentități reciproc exclusive** definește o clasă.
  - (Clasele se aliniază în desen vertical. )



# Diagrame Entitate-Relație

---

## Relații recursive

- Într-o diagramă E/R se pot defini **relații recursive**.
- Exemplu?

# Diagrame Entitate-Relație

---

## Relații de tip 2, 3

- Unele relații sunt **relative la două entități** și le numim de **tip 2**, iar dacă relațiile **implică mai mult de două entități**, le vom numi de **tip 3**.
  - Trei relații de tip 2 sunt diferite de o relație de tip 3!
  - Rupând o relație de tip 3 în trei relații de tip 2, pot apărea informații incorecte

# Diagrame Entitate-relație

---

- Trebuie **excluse** din model **relațiile indirecte** deoarece ele pot conduce la redundanță în baza de date.
- **Atributele derivabile** trebuie eliminate și introduse expresii prin care aceste atribute pot fi calculate.
- Exemple?

# Diagrame Entitate-Relație

---

## Relație sau atribut?

- **Relație sau atribut?**

- Dacă un atribut al unei entități reprezintă cheia primară a unei alte entități, atunci el referă o relație (*cod\_departament* în tabelul SALARIAT).

## Entitate sau relație?

- **Entitate sau relație?**

- Se cercetează cheia primară. Dacă aceasta combină cheile primare a două entități, atunci este vorba de o relație.
- (cheia primară a relației *asociat\_la* combină *cod\_salariat* cu *nr\_proiect*, prin urmare, *SALARIAT\_asociat\_la\_PROIECT* va defini o relație și nu o entitate).

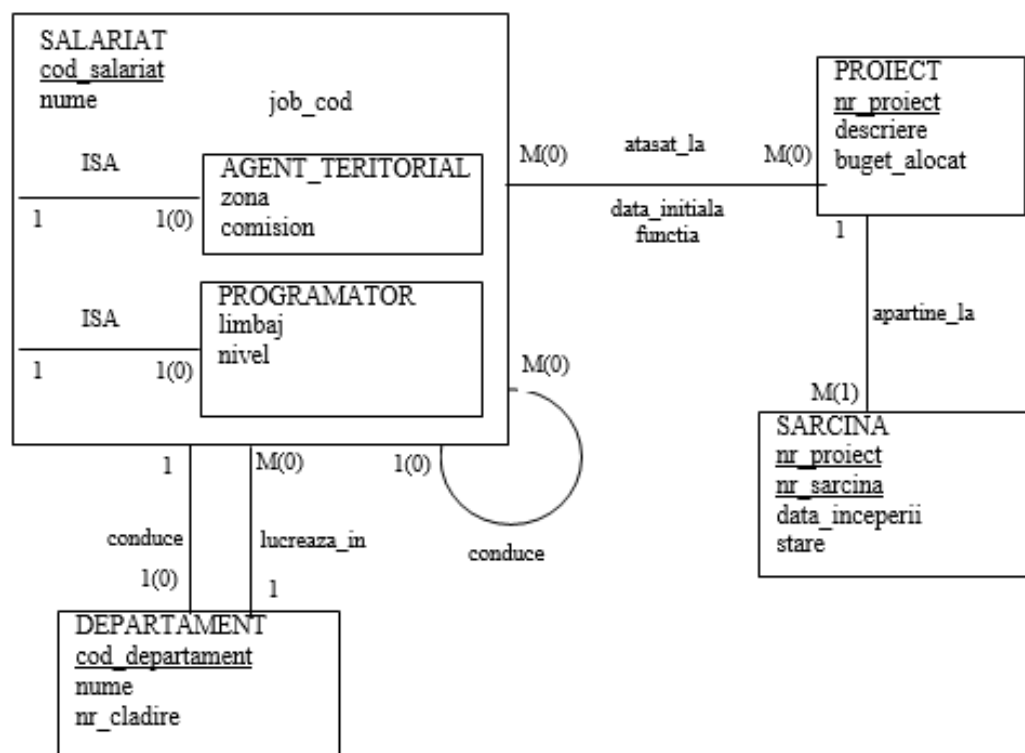
# Diagrame Entitate-Relație

---

## Algoritmul pentru proiectarea diagramei E/R

1. identificarea **entităților** din cadrul sistemului analizat;
2. identificarea **relațiilor** (asocierilor) dintre entități și stabilirea **cardinalității**;
3. identificarea **atributelor** aferente entităților și asocierilor dintre entități;
4. stabilirea atributelor de identificare a entităților, adică stabilirea **cheilor primare**.

# Diagrame Entitate-Relație



# Diagrame Entitate-Relație

---

- Aceeași realitate poate fi percepută diferit de către diverși analiști pentru un același sistem => **modele structurale distincte**.

- Care sunt pașii următori?

- Construirea **diagramei conceptuale**
  - obținerea **schemelor relaționale**
  - **normalizarea** acestora
- } =>

=> un **model relațional** care va elimina anumite **clase de anomalii** ce pot să apară în proiectarea modelului de date.

# Modelul E/R extins

---



# Modelul E/R extins

---

- Conceptele de bază ale modelării E/R nu sunt suficiente pentru a reprezenta cerințe complexe.
- Modelul E/R susținut cu concepte semantice adiționale definește **modelul E/R extins** (EER).
  - include toate **conceptele modelului original**
  - + conceptele adiționale de **subclasă, superclasă, moștenire, specializare, generalizare**.

# Modelul E/R extins

---

- **Superclasa** (superentitatea) este o entitate care include subclase (subentități) distincte, ce trebuie reprezentate în modelul de date.
- **Subclasa** are un rol distinct și, evident, este membră a unei superclase. O subclasă, fiind o entitate, poate să posede propriile subclase.
  - O entitate împreună cu subclasele ei, subclasele acestora și așa mai departe definește o **ierarhie de tip** (**ierarhie de specializare**). De exemplu, ANGAJAT\_TEMP reprezintă o superclasă pentru entitatea MODEL.

# Modelul E/R extins

---

- **Specializarea** este procesul de **maximizare a diferențelor** dintre membrii unei entități, prin identificarea caracteristicilor distinctive ale acestora.
  - Dacă subclasele unei specializări sunt **disjuncte**, atunci o entitate poate fi membră doar a unei subclase a acesteia (**constrângere de disjuncție**).
  - O specializare cu participare **totală** specifică faptul că fiecare entitate din superclasă trebuie să fie membră a unei subclase din specializare (**constrângere de participare**).
  - O specializare cu participare **parțială** specifică faptul că nu este necesar ca o entitate să aparțină vreunei subclase a acesteia. De exemplu, există salariați în PERS\_CONTACT care nu aparțin niciunei subentități ale acesteia.

# Modelul E/R extins

---

- **Generalizarea** este procesul de **minimizare a diferențelor** dintre entități, prin **identificarea caracteristicilor comune** ale acestora.
  - Generalizarea are ca rezultat identificarea unei superclase generalizate din subclasele inițiale.

# Deficiențe ale modelului E/R



# Deficiențe ale modelului E/R

---

- Cauza: interpretare eronată a sensului unei relații => **capcane de conectare.**
- Posibil să necesite restructurarea modelului
- 2 clase de capcane de conectare:
  - **de intrerupere**
  - **în evantai**

# Deficiențe ale modelului E/R

---

- **Capcane de întrerupere:** modelul sugerează existența unei relații între entități, dar **nu există o cale între anumite apariții ale entităților**.
  - Această capcană poate să apară acolo unde există o relație cu **participare parțială** (0 la cardinalitatea minimă), care face parte din calea dintre entitățile ce sunt legate.
- **Capcane în evantai:** modelul ia în considerare o relație între entități, dar **calea dintre anumite apariții ale entităților este ambiguă**.
  - Aceste capcane apar când **două sau mai multe relații *one\_to\_many*** provin din aceeași entitate.

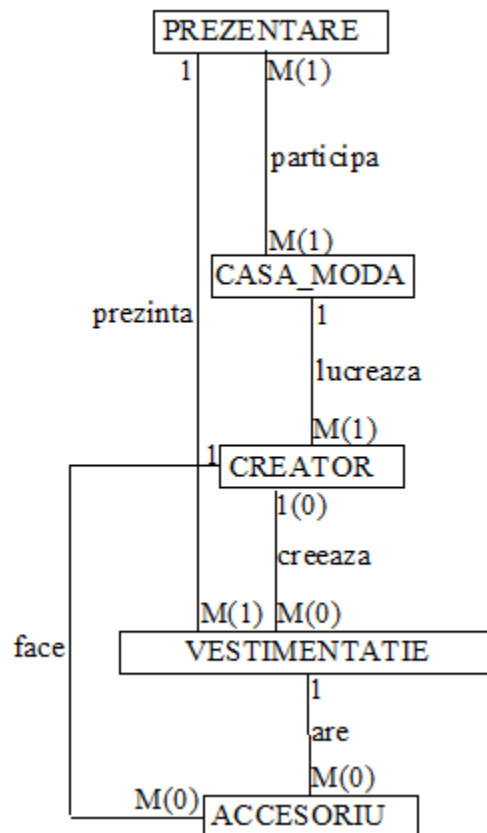
# Deficiențe ale modelului E/R

---

- Aceste capcane generează situațiile în care, așa cum a fost proiectat modelul de date, **el nu poate să răspundă la anumite interogări.**
  - Dați un exemplu de capcană în evantai.
  - Exemplu de capcană de întrerupere: pentru a afla pentru ce prezentare de modă a fost creată o anumită vestimentație, a fost necesară introducerea unei legături între entitățile PREZENTARE și VESTIMENTATIE, care însă a generat redundanță în modelul de date:



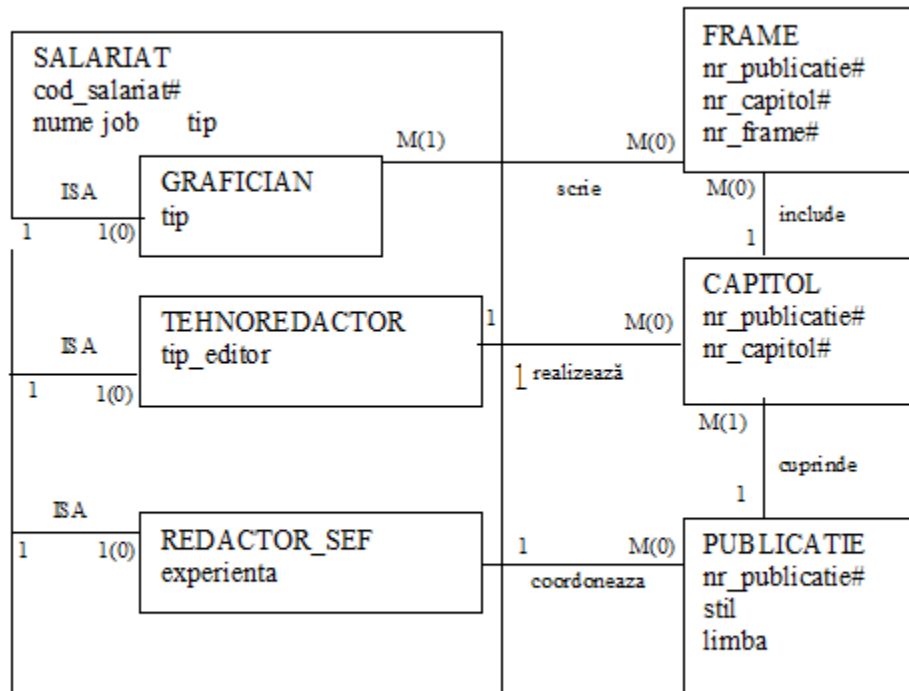
# Deficiențe ale modelului E/R



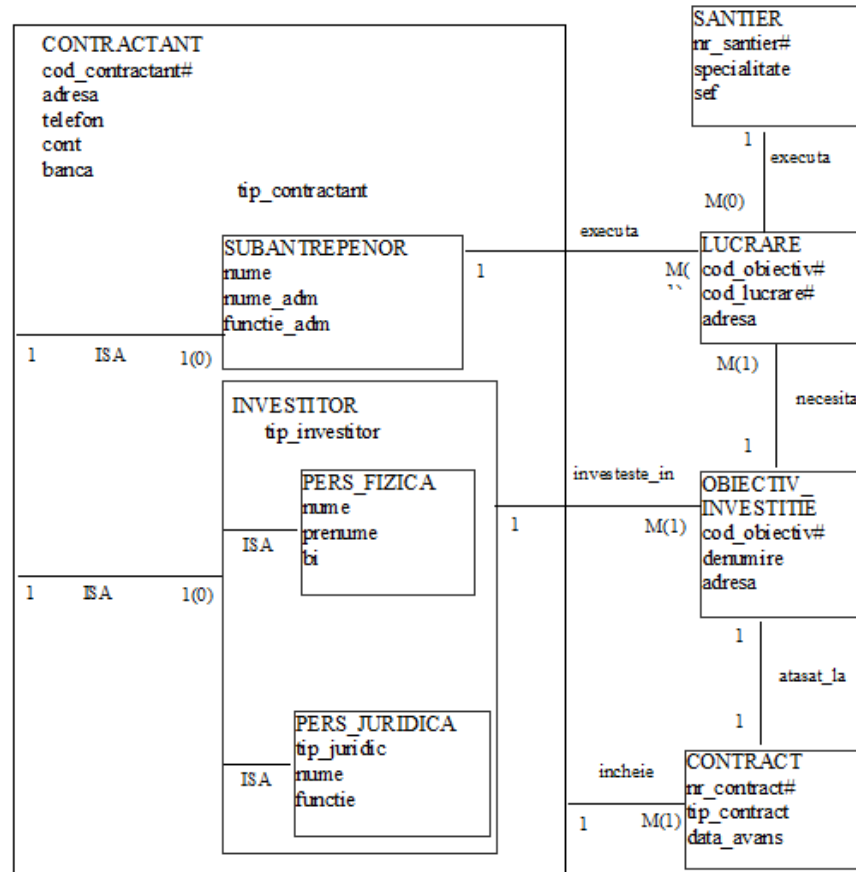
# Exemple de diagrame E/R



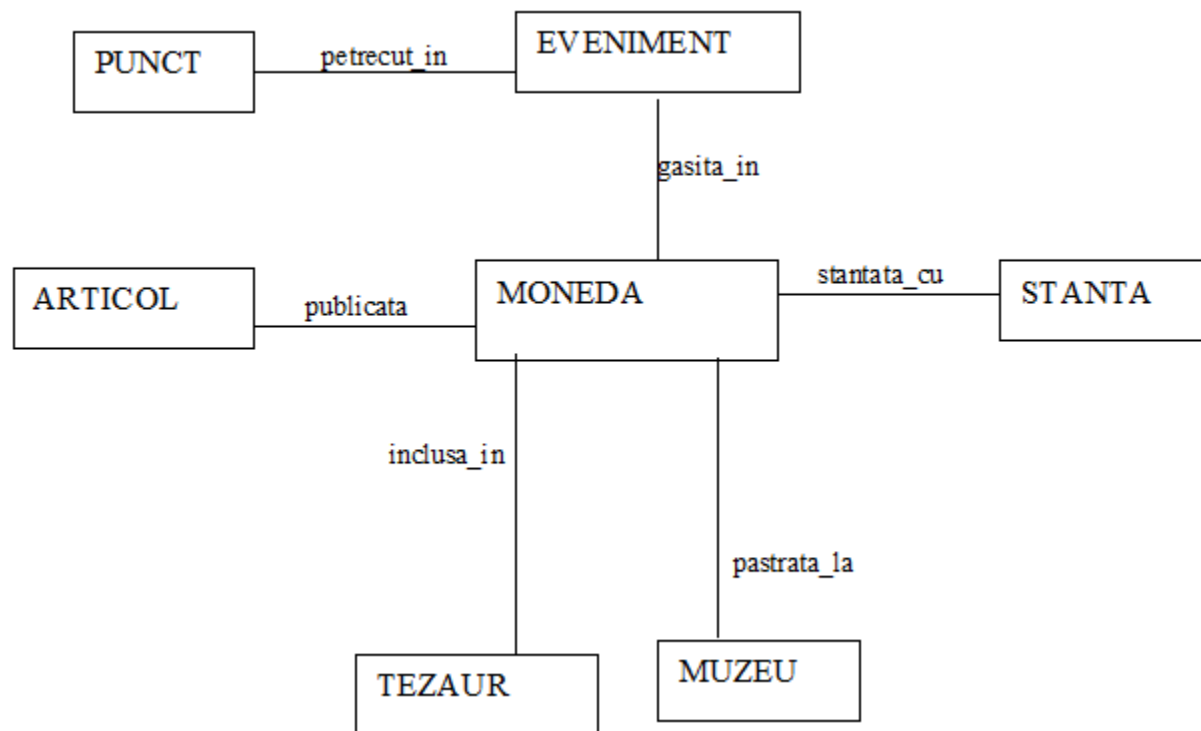
# Exemplul 1: Gestiunea activităților de editare dintr-o editură



# Exemplul 2: Gestiunea unei firme de constructii



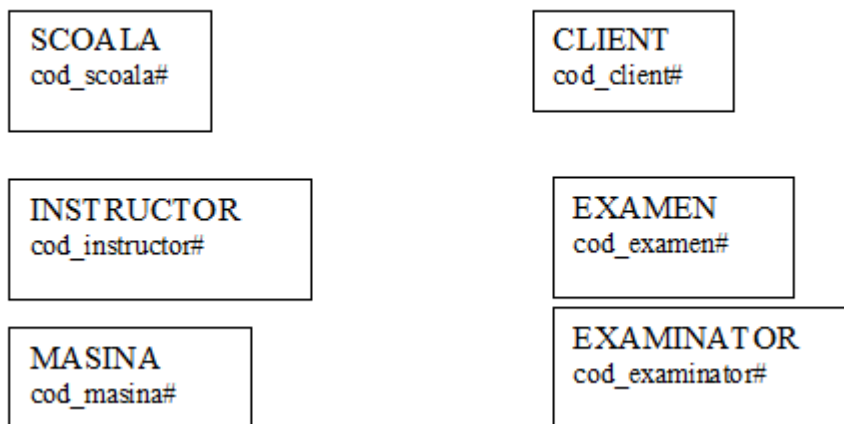
# Exemplul 3: Descoperiri de monede antice din România



Completați cardinalitatea!

STANȚA (nr\_stanță, împărat emitent, valoare nominală, an emitere, monetăria, legenda de pe avers, legenda de pe revers) == > atribute ale entității **STANTA**

# Exemplul 4: Evidența școlilor de șoferi din România



Completați relațiile (*lucreaza\_la*, *conduce*, *sustine*, *asista*, *instruieste*) dintre entități și specificați cardinalitatea!

# Exemplul 5: Campionatele de fotbal ale diferitelor țări

---

- Care este relația dintre entitățile MECI și ECHIPA? Ce cardinalitate are?

# Exemplul 6: Gestiunea activităților dintr-o agenție de turism

---

- Într-o agenție de turism lucrează ghizi, șoferi, agenți de vânzări.
- Din oferta agenției fac parte sejururi și excursii.
- Ghizii conduc excursii, la care sunt atașați șoferi.
- Clienții agenției achiziționează sejururi sau excursii.
- Clienții sunt deserviți de către agenții de vânzări ai agenției.
- Un sejur se desfășoară într-o anumită locație.



# TEMĂ

---

- Proiectați diagrama E/R pentru gestionarea activităților unei universități.  
Lista minimală a entităților care vor apărea în model este următoarea:  
facultate, domeniu de studiu, an universitar, an de studiu, serie, grupă, student, cadru didactic, curs, seminar, laborator, examen).
- Observații:
  - Va fi creat un assingment Teams prin intermediul căruia vor fi trimise soluțiile până pe 16 martie 2024, ora 23:59
  - Diagramele vor respecta modul de reprezentare folosit la curs.

# Bibliografie

---

- Popescu, I., Velcescu, L., ***Proiectarea bazelor de date***, Editura Universității din București, 2008 – Capitolul 2
- Connolly, T.M., Begg, C.E., Database Systems: ***A Practical Approach to Design, Implementation and Management***, 6th edition, Pearson Education, 2015 – Capitolele 11, 12