

Tutoriat #1 Modelul Entitate-Relație

Un **model**, în sens larg, este o reprezentare a obiectelor și evenimentelor lumii reale și a asocierilor dintre ele. Modelarea unei baze de date presupune trecerea de la percepția unor **fapte** din viața reală la reprezentarea lor prin **date**. Ideal, încercăm ca modelul obținut să fie o reprezentare cât mai fidelă a realității (să surprindă cât mai multe aspecte, să poată urmări evoluția acestei lumi, să permită "comunicarea" dintre evenimente).

Un model de date presupune 3 componente:

- **regulile după care sunt construite bazele de date (ce deducem din "story", adică din descrierea modelului de date)**
- o mulțime de operații permise asupra datelor (pentru a actualiza/regăsi date)
- regulile de integritate: asigură coerența datelor

În **modelul entitate-relație**, elementele lumii reale sunt împărțite în două categorii de "obiecte": **entități** și **relații** (legături, asocieri) între entități. Diagrama E/R este reprezentarea grafică a acestui model.

Elementele ce alcătuiesc o diagramă E/R:

1. Entitate = lucru, obiect, persoană, eveniment care are semnificație pentru ceea ce modelăm, despre care trebuie să colectăm și memorăm date.

- de cele mai multe ori, căutăm substantivele din descrierea modelului. **Atenție!! Nu orice substantiv reprezintă o entitate!**
- în diagrama E/R, entitățile sunt reprezentate prin dreptunghiuri, numele acestora se va scrie cu litere mari, în limba română (substantivele trebuie să fie la forma de singular)

2. Relație (legătură) = o conexiune între două sau mai multe entități și care are semnificație pentru ceea ce modelăm.

- căutăm printre verbele din descrierea modelului. **Atenție!! Nu orice verb desemnează o relație!**
- putem avea relații cu același nume (în acest caz, le diferențiază entitățile cărora le sunt asociate)
- în diagrama E/R, relațiile sunt reprezentate prin arce neorientate
- trebuie să stabilim **cardinalitatea** (maximă și minimă) a fiecărei relații, adică numărul de tupluri (înregistrări/ linii din tabel) care participă la relație. Cardinalitatea trebuie specificată **pentru fiecare entitate care participă la relație!**

- Cardinalitatea maximă se referă la numărul maxim de înregistrări dintr-o entitate care **pot** fi asociate unei singure înregistrări din cealaltă entitate
- Cardinalitatea minimă (în alte resurse este denumită **opționalitate**) reprezintă obligativitatea existenței unei asocieri între cele două entități. Cu ajutorul ei, definim numărul minim de înregistrări între care **trebuie** să existe o legătură.
- Cardinalitatea se notează la capetele relației, în dreptul fiecărei entități

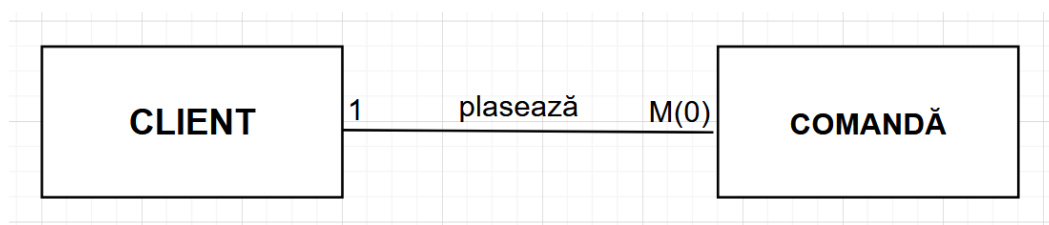
Ex: Avem următoarea specificație/regulă a modelului care descrie funcționarea unei farmacii online: **Un client poate plasa mai multe comenzi.**

→ **cardinalitatea maximă** pentru entitatea **CLIENT** se stabilește punând întrebarea: *Câți clienți pot plasa o comandă? (De către câți clienți poate fi plasată o comandă?)*
R: unul singur!

→ **cardinalitatea minimă** pentru entitatea **CLIENT**: *De către câți clienți trebuie să fie plasată o comandă?*
R: exact unul! (intuitiv, ne putem gândi că o comandă nu poate exista fără să fie inițiată de către un client)

→ **cardinalitatea maximă** pentru entitatea **COMANDĂ**: *Câte comenzi poate plasa un client?*
R: mai multe! (într-un timp mai îndelungat, același client poate avea 2, 3,..., n comenzi)

→ **cardinalitatea minimă** pentru entitatea **COMANDĂ**: *Câte comenzi trebuie să plaseze un client?*
R: niciuna! (de exemplu, pot exista în baza de date clienți care s-au înregistrat în aplicația farmaciei, dar nu au plasat încă nicio comandă)



Vom spune că relația **CLIENT_plasează_COMANDĂ** are **cardinalitatea maximă one-to-many** și cardinalitatea minimă one-zero. În implementare, ne va interesa doar cardinalitatea maximă.

Observație: răspunsurile la întrebările de mai sus (prin care stabilim cardinalitățile) le găsim fie printre specificațiile modelului de date, fie le deducem din context (ne gândim cum ar trebui să fie în realitate și vom modela în acest fel. Adică, în 99.9% din cazuri,

nu ar avea sens să spunem că "o comandă poate fi plasată de mai mulți clienți", pentru că în realitate nu se întâmplă așa ☺).

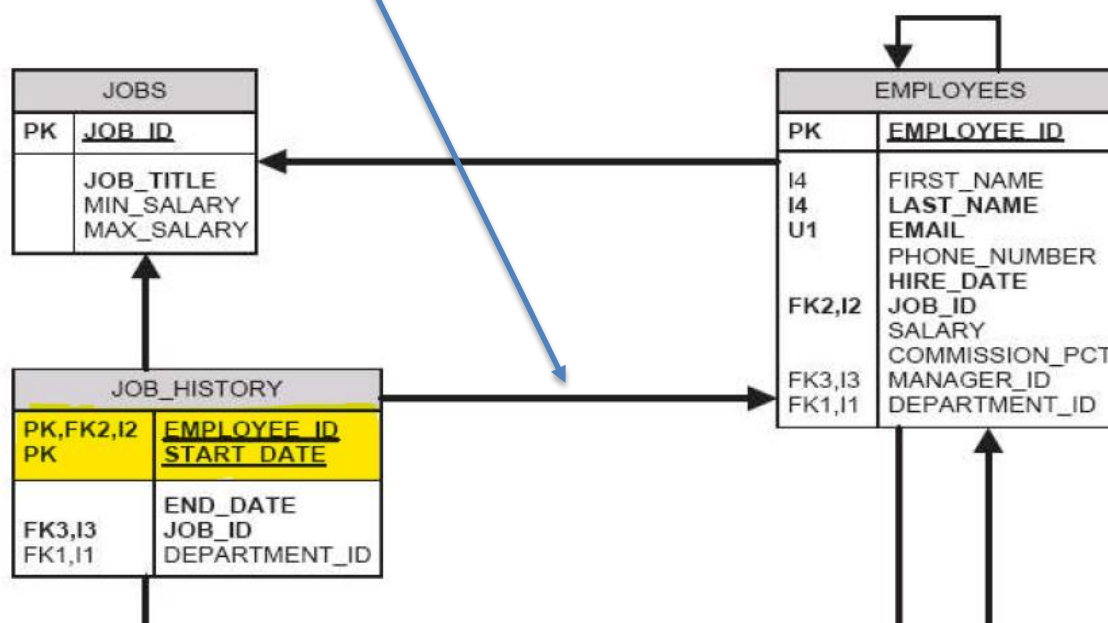
3. Atribut = proprietate descriptivă a unei entități sau relații

- în general, attributele sunt substantive
- pentru fiecare atribut, trebuie specificat numele, tipul fizic, valori posibile, valori implicite (dacă există), reguli de validare (dacă există)
- într-o entitate, trebuie să includem doar acele attribute care sunt caracteristici relevante ale entității (Ex: într-o entitate ANGAJAT, nu vom considera drept atribut culoare_ochi).

4. Cheie primară = un atribut sau o mulțime de attribute, care diferențiază între înregistrările unei entități.

- este **identificator unic** în cadrul entității
- trebuie să fie unică și cunoscută în orice moment
- trebuie să fie cât mai simplă (să conțină numărul minim de attribute care diferențiază între valori diferite ale entității)
- în diagrama E/R, attributele care alcătuiesc cheia primară fie se subliniază, fie se marchează cu un #

Exemplu 1: Avem următoarea relație din diagrama Human Resources (HR) de la laborator: *EMPLOYEE_has_JOB_HISTORY* (Fiecare angajat are un istoric al job-urilor în cadrul firmei).



Observăm că tabelul/entitatea JOB_HISTORY, în care sunt reținute toate job-urile pe care angajații le-au avut în trecut (excluzând job-ul curent), are o cheie primară alcătuită din 2 attribute (*employee_id* și *start_date*). De ce nu era suficient să luăm drept cheie primară doar *employee_id*?

R: Dacă am fi modelat în acel fel, nu am fi putut reține decât un singur job încheiat pentru fiecare angajat, ceea ce nu corespunde neapărat realității, deoarece același angajat ar fi putut avea mai multe job-uri în trecut la aceeași firmă (putem presupune, de exemplu, că cineva a avansat pe scara ierarhică a companiei, și implicit și-a schimbat job-ul, având alte atribuții). De aceea, vom folosi o **cheie primară compusă**. Fiecare înregistrare din tabel este unic identificată de id-ul angajatului și de data la care a început job-ul.

Exemplu 2: Avem mai jos entitatea REZERVARE, care reține informații despre rezervările la un teatru. Bazându-ne doar pe datele pe care le avem, care ar putea fi o cheie primară?

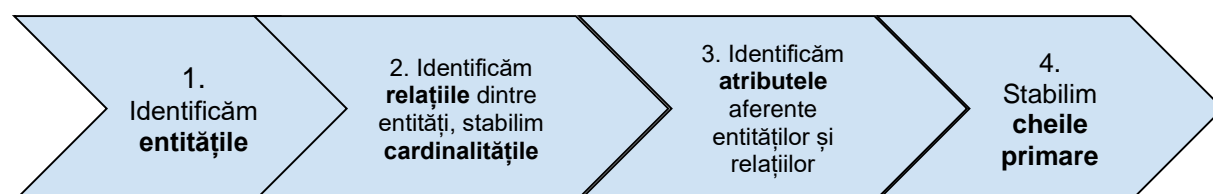
id_cumpărător	preț	id_sală	dată_spectacol	Categorie_bilet
1	150	3	08/03/2025	A
1	150	4	15/03/2025	B
1	200	3	09/03/2025	A
2	200	3	09/03/2025	A
2	140	5	08/03/2025	B
3	150	3	08/03/2025	A

R: Mai întâi ne gândim dacă putem avea o cheie primară doar dintr-un singur atribut. Observăm că nu se poate, întrucât avem repetări ale valorilor pentru fiecare atribut. Dar putem avea în cheia primară două atribute, de exemplu (*id_cumpărător*, *id_sală*)? Nu, avem și în acest caz repetări ale valorilor.

Dar cu trei atribute? Observăm că (*id_cumpărător*, *id_sală*, *dată_spectacol*) este o cheie primară validă, identificând în mod unic o singură linie din tabel.

Obs. Acest exemplu este pur didactic, în general nu va trebui să ne punem atâtea întrebări pentru a stabili o cheie primară, deoarece felul în care modelăm entitățile ne va ajuta să alegem cheile primare cu mai multă ușurință. În realitate, pentru tabelul de mai sus am fi adăugat un atribut *id_rezervare*, unic fiecărei rezervări.

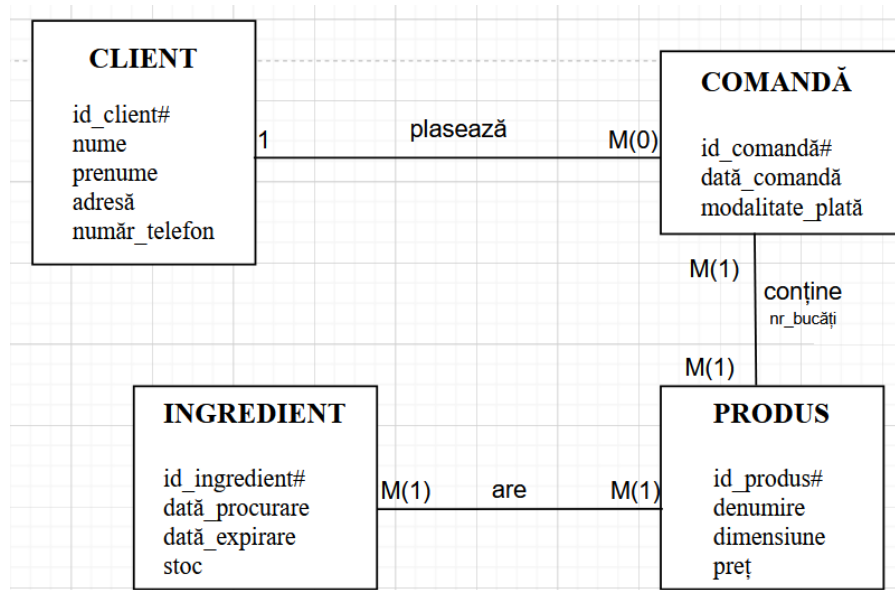
Pași în realizarea diagramei E/R (algoritm)



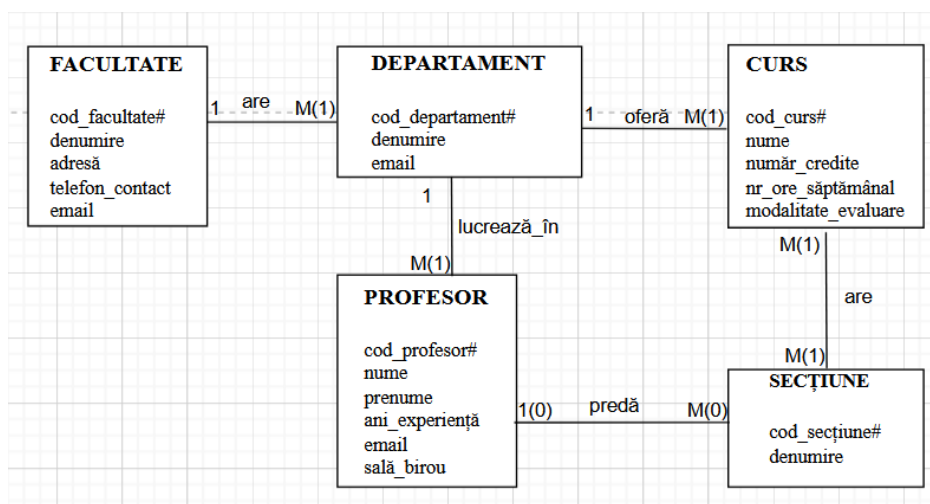
Exemple:

1. La o pizzerie care face și livrare la domiciliu, se ține evidența clienților și a produselor vândute. Fiecare **client** care face comanda online furnizează numele, adresa, numărul de telefon și este identificat unic de către firmă printr-

un ID. **Produsele** au un cod unic, denumire, **ingrediente**, preț, eventual dimensiune (mare, mică, medie). La fiecare **comandă**, clientul specifică tipul produsului, numărul de bucăți din fiecare produs comandat și modalitatea de plată.



- O universitate are în componența sa mai multe **facultăți**, fiecare facultate având mai multe **departamente**. Fiecare departament oferă studenților mai multe **cursuri**. Un **profesor** poate lucra la un singur departament al unei singure facultăți. Fiecare curs are mai multe **secțiuni**, iar o secțiune poate să facă parte din mai multe cursuri. Un profesor poate predă mai multe secțiuni din același curs sau din cursuri diferite, dar o secțiune nu poate fi predată de mai mulți profesori.



Exerciții:

3. O firmă produce mai multe tipuri de mașini, un model fiind caracterizat printr-un nume, mărimea motorului și un sufix care indică gradul de lux al acesteia (de exemplu XL, GL). Fiecare model este construit din mai multe părți. Fiecare parte are o descriere și un cod. Fiecare model de mașină este produs de exact o fabrică a firmei, fabrică ce se poate găsi în una din țările UE. O fabrică poate produce mai multe modele de mașini și mai multe tipuri de părți componente. De asemenea, fiecare tip de parte componentă poate fi produsă de o singură fabrică a firmei.

4. 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. (preluat din curs, Seria 13)