

3. NORMALIZAREA DATELOR

- Anomaliile care pot apărea la o bază de date
- Definiția normalizării
- Care sunt formele normale
- Care sunt regulile pe care trebuie să le respecte o entitate pentru a se afla în una din formele normale 1NF, 2NF, 3NF
- Cum poate fi adus un ERD la a treia formă normală

3.1. CE ESTE NORMALIZAREA?

- **Normalizarea** este o tehnică de proiectare a bazelor de date prin care se elimină (sau se evită) anumite anomalii și inconsistențe a datelor.
- O baza de date bine proiectată nu permite astfel ca datele să fie **redundante**, adică aceeași informație să se găsească în locuri diferite, sau să memorezi în baza de date, informații care se pot deduce pe baza altor informații memorate în aceeași bază de date.



- Anomaliile care pot să apară la o bază de date nenormalizată sunt următoarele:
 - **anomalii la actualizarea datelor:** la o bibliotecă se înregistrează într-o tabelă următoarele date despre cărți: ISBN, titlu, autor, preț, subiect, editura, adresa editurii. Dacă o editură își schimbă adresa, bibliotecara va trebui să modifice adresa editurii respective, în înregistrările corespunzătoare tuturor cărților din bibliotecă apărute la respectiva editură. Dacă această modificare nu se face cu succes, unele dintre înregistrări rămânând cu vechea adresă, apare din nou o inconsistență a datelor.
 - **anomalii de inserare** – în exemplul anterior, nu putem memora adresa unei edituri, lucru inacceptabil dacă dorim să avem informații și despre edituri a căror cărți nu le avem în bibliotecă, eventual de la care dorim să facem comenzi.
 - **anomalii de ștergere** – să presupunem că într-o tabelă memorăm următoarele informații: codul studentului, codul cursului, codul profesorului. La un moment dat, nici un student nu mai dorește să participe la un anumit curs. Ștergând toate înregistrările corespunzătoare cursului, nu vom mai putea ști niciodată cine preda acel curs.
- Edgar Codd a definit primele trei forme normale 1NF, 2NF și 3NF. Ulterior s-au mai definit formele normale 4NF, 5NF, 6NF care însă sunt rar folosite în proiectarea bazelor de date.

3.2. PRIMA FORMĂ NORMALĂ

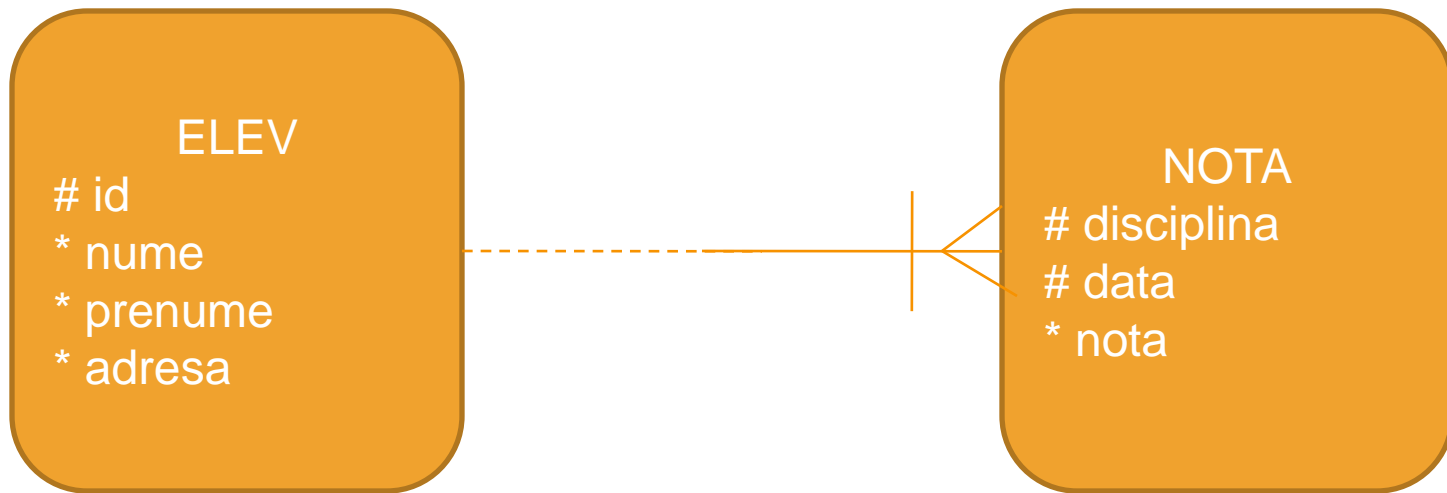
- O entitate se găsește în prima formă normală dacă și numai dacă:
 - nu există attribute cu valori multiple;
 - nu există attribute sau grupuri de attribute care se repetă.
- Toate attributele trebuie să fie atomice, adică să conțină **o singură informație**. Dacă un atribut are valori multiple, sau un grup de attribute se repetă, atunci trebuie să creați o entitate suplimentară pe care să o legați de entitatea originală printr-o relație de 1:m. În noua entitate vor fi introduse attributele sau grupurile de attribute care se repetă.



```
ELEV
# id
* nume
* prenume
* adresa
o disciplina1
0 nota1
0 disciplina2
0 nota2
...
o disciplinan
o notan
```

- Să considerăm entitatea din figură alăturată referitoare la notele elevilor unei clase.
- Câteva observații referitoare la această entitate:
 - Câte discipline are un elev?
 - Câte perechi (disciplina, nota) va trebui să aibă entitatea Elevi?
- Să spunem că știm exact câte discipline maxim poate studia un elev.
 - Ce se întâmplă dacă în anul viitor școlar acest număr de discipline va fi mai mare?
- La o materie un elev poate avea mai multe note.
 - Câte note poate avea un elev?
 - Cum memorăm aceste note?
 - Le punem în câmpul corespunzător disciplinei cu virgulă între ele?

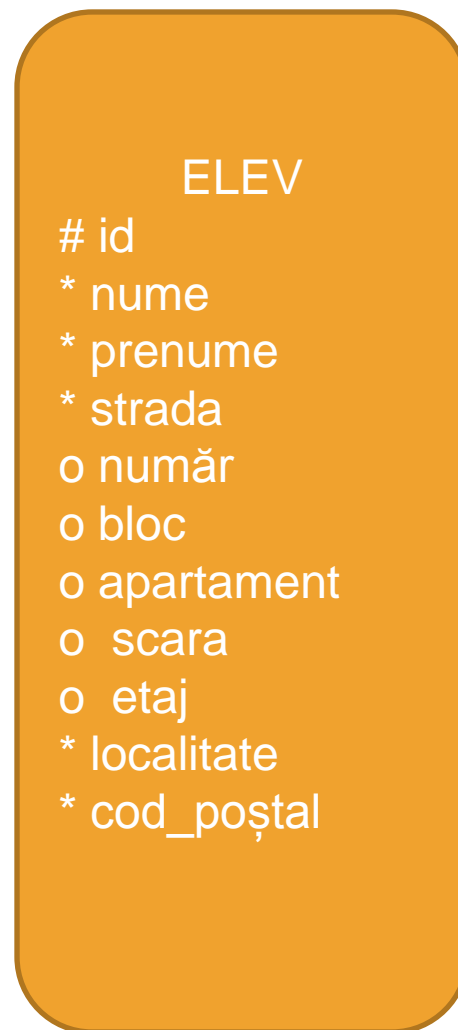
- Pentru a rezolva această problemă, vom crea o nouă entitate în care vom introduce disciplina și nota la disciplina respectivă.



- În acest fel fiecărui elev îi pot corespunde oricâte note, iar la o disciplină poate avea oricâte note, singura restricție conform acestui model fiind că un elev nu va putea primi în aceeași zi la aceeași materie mai multe note.



- Un alt exemplu de încălcare a regulilor primei forme normale, puțin mai "ascuns", este prezentat în figura alăturată pentru că adresa este de forma "str. Florilor, bl. 45, sc. A, ap. 28, etaj 3, Brașov, cod 123123", formă care de fapt conține mai multe informații elementare.
- Acest atribut trebuie "spart" în mai multe attribute.



- Noile atributele introduse sunt opționale întrucât dacă elevul locuiește la casă, probabil atributele bloc, apartament, scara, etaj, nu au sens.
- Deoarece datele din interiorul unei adrese este puțin probabil să se modifice, modificându-se cel mult adresa completă a unui elev, se poate decide să nu se opereze modificarea anterioară.
- Dacă însă aceste informații s-ar modifica frecvent, (de exemplu denumirile străzilor) atunci probabil modificarea este de dorit.



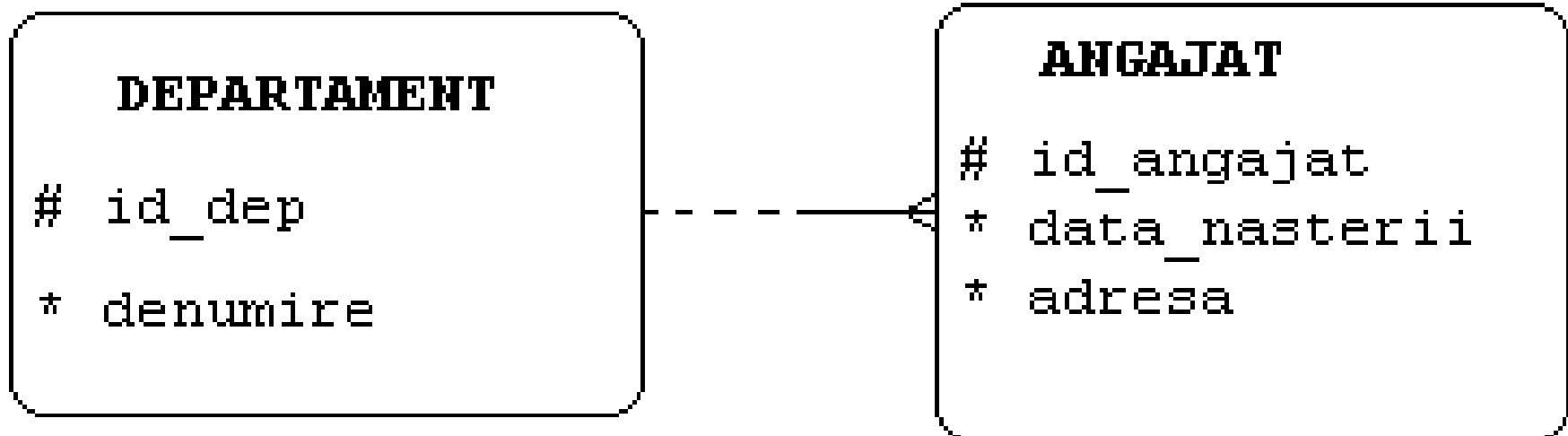
3.3. A DOUA FORMĂ NORMALĂ

- O entitate se găsește în a doua formă normală **dacă și numai dacă se găsește în prima formă normală și în plus orice atribut care nu face parte din UID (unique identifier) va depinde de întregul UID nu doar de o parte a acestuia.**
- De exemplu dacă memorăm angajații unui departament într-o entitate ca mai jos:

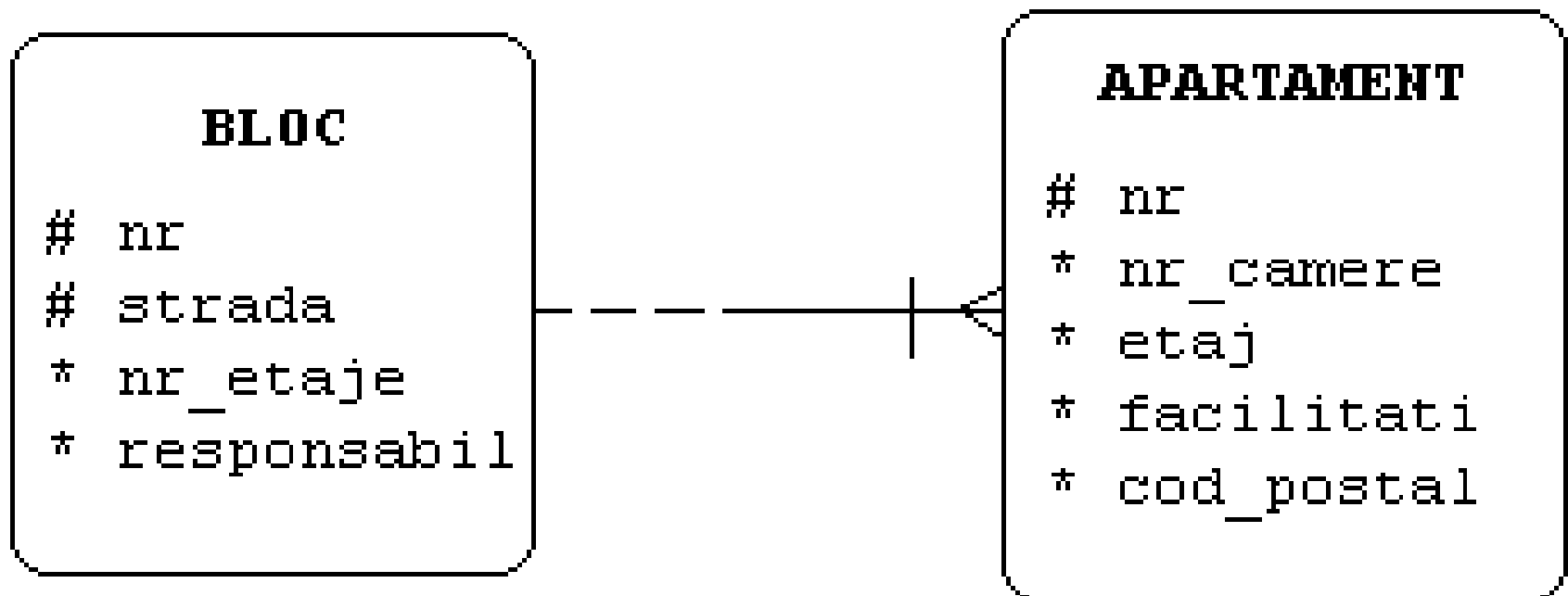
```
DEPARTAMENT
# id_dep
# id_angajat
* data_nasterii
* adresa
```

Entitatea DEPARTAMENT

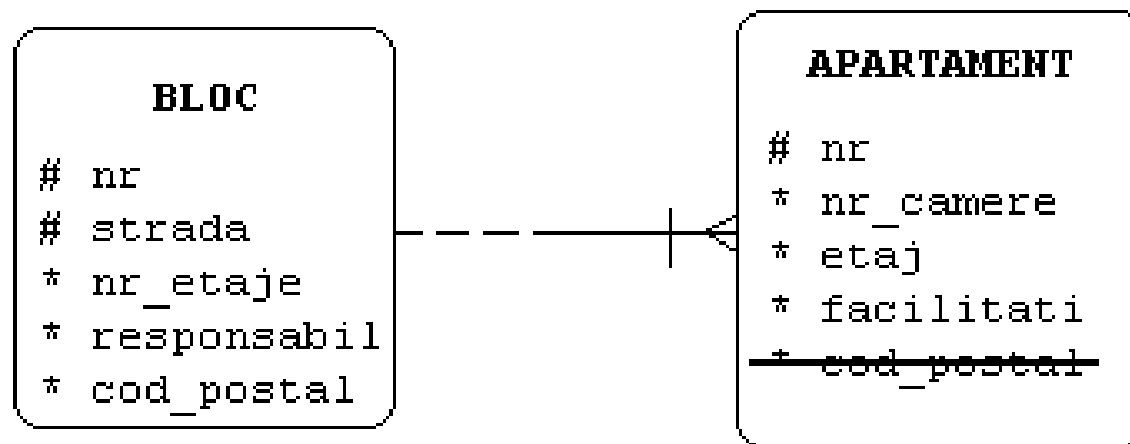
- Se observă că **data_nasterii** și adresa sunt două attribute care depind doar de id-ul angajatului nu de întregul UID care este combinația dintre attributele **id_dep** și **id_angajat**.
- Această situație se rezolvă prin crearea unei noi entități **ANGAJAT**, pe care o legăm de entitatea **DEPARTAMENT** printr-o relație **1:M**



- O situație mai specială este în cazul relațiilor barate, când trebuie ținut seama că UID-ul unei entități este compus din attribute din entitatea respectivă plus un atribut sau mai multe attribute provenite din relația barată. Să considerăm următorul exemplu:



- Se observă că UID-ul entității **APARTAMENT** este compus din combinația a trei atribute: numărul apartamentului, numărul blocului și strada.
- Deci toate atributele din entitatea **APARTAMENT** care nu fac parte din UID, trebuie să depindă de întregul UID. Dar se știe că atributul `cod_postal` depinde doar de strada și de numărul blocului, nu și de numărul apartamentului. Acest lucru ne spune că acest atribut nu este memorat la locul potrivit. Deoarece depinde doar de combinația (strada, nr_bloc), înseamnă că de fapt depinde de UID-ul entității **bloc**. Așadar vom muta atributul `cod_postal` în entitatea **BLOC**.



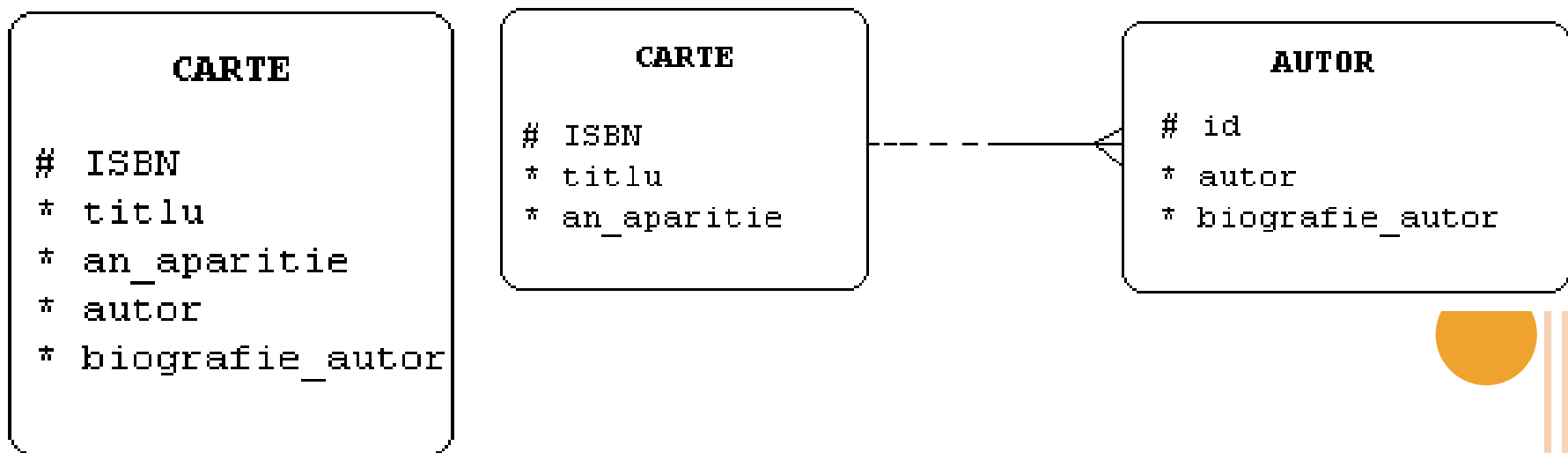
- **Observație.** Dacă o entitate se găsește în prima formă normală și UID-ul său este format dintr-un singur atribut atunci ea se găsește automat în a doua formă normală.

3.4. A TREIA FORMĂ NORMALĂ

- O entitate se găsește în a treia formă normală dacă și numai dacă se găsește în a doua formă normală și în plus nici un atribut care nu este parte a UID-ului nu depinde de un alt atribut non-UID. Cu alte cuvinte nu se acceptă dependențe tranzitive, adică un atribut să depindă de UID în mod indirect.



- Luăm ca exemplu entitatea **CARTE** Atributul **biografie_autor** nu depinde de **ISBN** ci de atributul **autor**. Nerezolvarea acestei situații duce la memorarea de date redundante, deoarece biografia unui autor va fi memorată pentru fiecare carte scrisă de autorul respectiv. Rezolvarea acestei situații este să creăm o nouă entitate **AUTOR**, pe care o legăm de entitatea **CARTE** printr-o relație **1:M**
- Acest exemplu este corect doar dacă o carte are doar un singur autor.



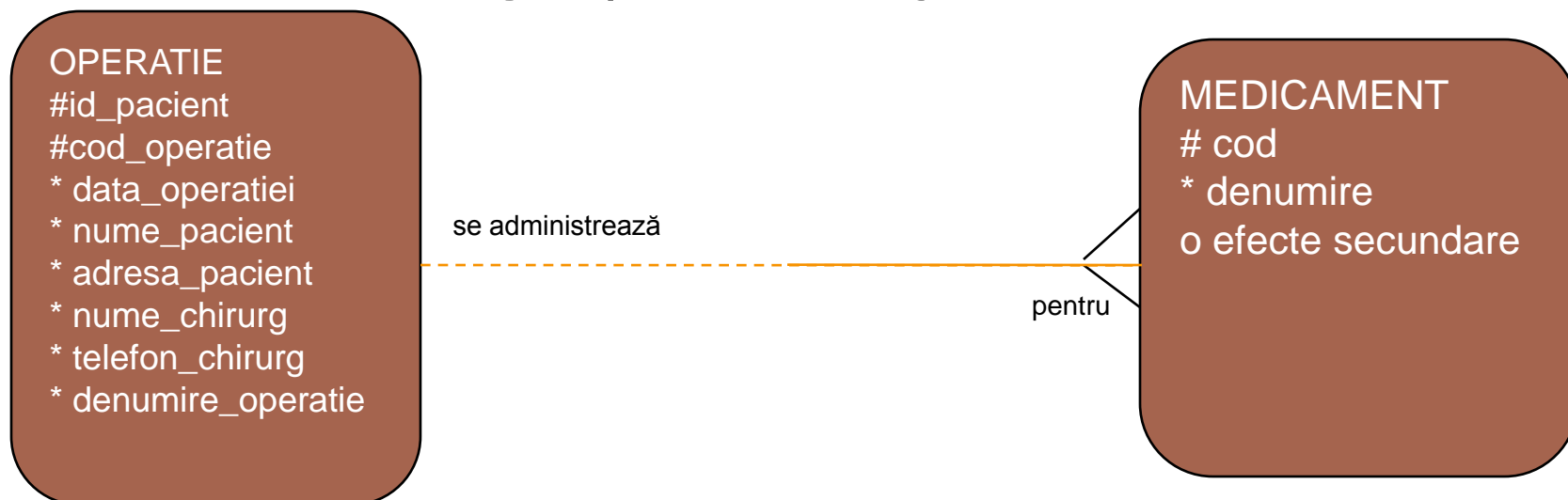
3.5. EXEMPLU DE NORMALIZARE

- Într-o clinică trebuie memorate toate datele despre operații.
- Pentru fiecare operație se memorează codul pacientului, numele și adresa, numele chirurgului, codul operației efectuate, data la care a avut loc, denumirea operației și tratamentul administrat după operație.
- Într-o primă fază se creează o singură entitate cu toate aceste informații:

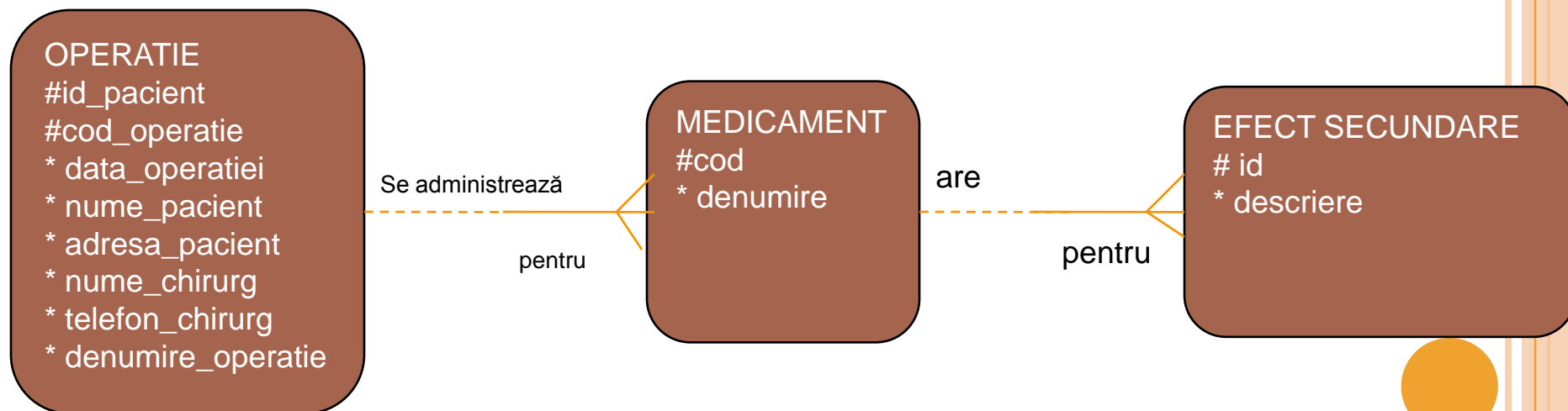
OPERATIE

```
# id_pacient
# cod_operatie
* data_operatiei
* nume_pacient
* adresa_pacient
* nume_chirurg
* telefon_chirurg
* tratament
o efecte_secundare
* denumire_operatie
```

- **Prima formă normală**
- Fiecare tratament constă în unul sau mai multe medicamente, fiecare cu efectele lui. Pentru aceasta se creează o nouă entitate numită **MEDICAMENT**, legată de entitatea **OPERAȚIE** printr-o relație 1:M.

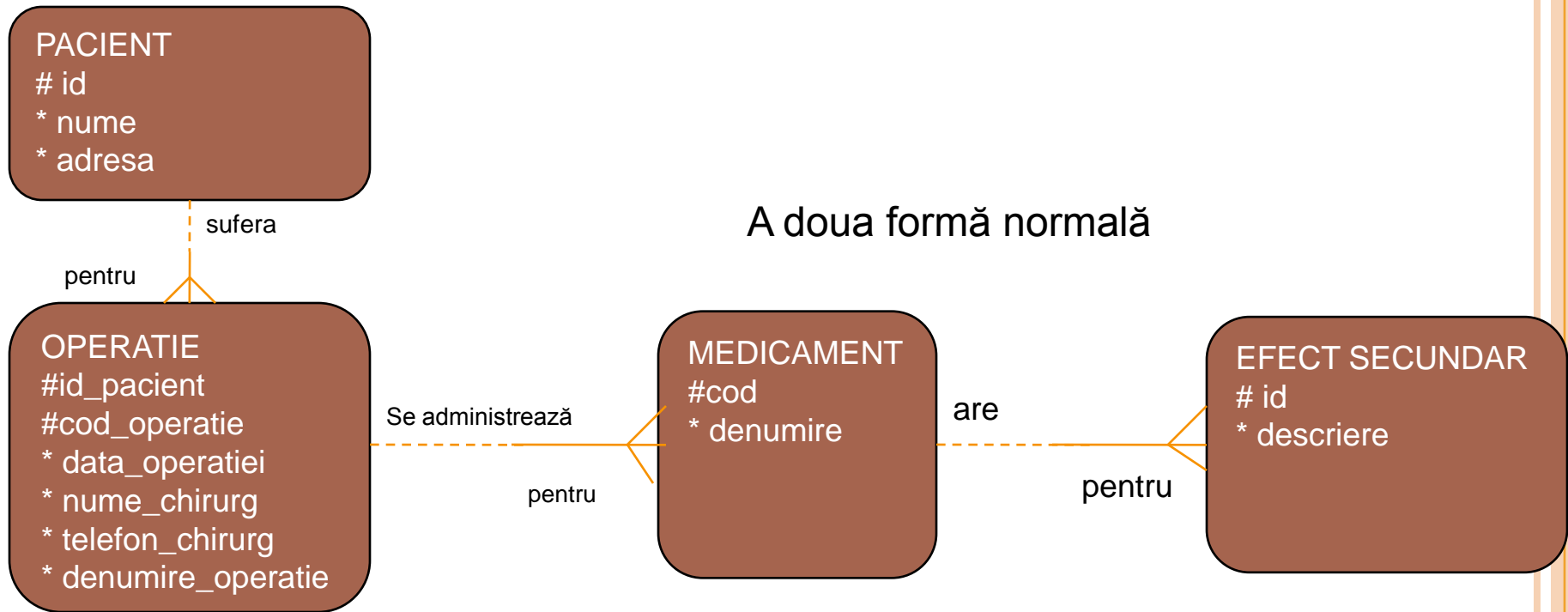


- Nici acest model nu este în prima formă normală pentru că fiecare medicament poate avea unul sau mai multe efecte secundare. De aceea se creează o entitate care memorează **efectele secundare** ale medicamentelor:



Prima formă normală

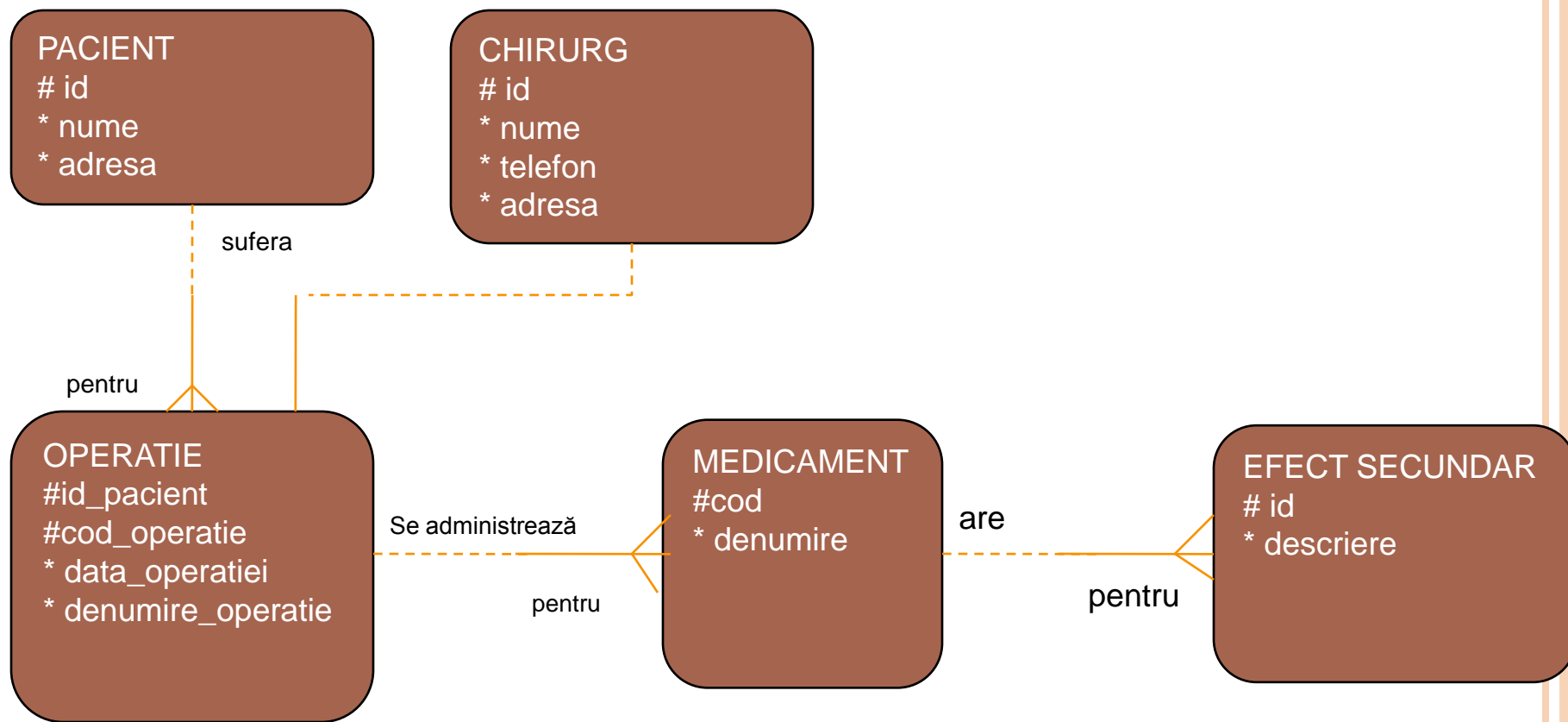
- **A doua formă normală.** Pentru a aduce entitățile în a doua formă normală, trebuie rezolvate problemele:
 - numele și adresa pacientului nu depind de întregul UID (id_pacient + cod_operatie) ci doar de o parte a acesteia și anume de id_pacient
 - denumirea operației depinde doar de cod_operatie și nu de întregul UID.
- De aceea se creează o nouă entitate numită PACIENT în care se vor memora toate datele despre pacient.



- **A treia formă normală**

- În ERD-ul de mai sus de observă că **telefon_chirurg** nu depinde de **cod_operatie** ci doar de **nume_chirurg**.
Datele despre chirurg se vor memora într-o nouă entitate numită **CHIRURG**.

A treia formă normală



În această lecție am învățat despre:

- Anomaliile care pot apărea la o bază de date
- Definiția normalizării
- Care sunt formele normale
- Care sunt regulile pe care trebuie să le respecte o entitate pentru a se afla în una din formele normale 1NF, 2NF, 3NF
- Cum poate fi adus un ERD la a treia formă normală

SFÂRȘIT

