

5. ELEMENTE AVANSATE DE PROIECTARE A BAZELOR DE DATE

- Modelarea clasificărilor obiectelor cu ajutorul subtipurilor și a supertipurilor
- Variante de mapare a subtipurilor și supertipurilor
- Maparea relațiilor exclusive
- Maparea relațiilor nontransferabile
- Modelarea datelor care se modifică în timp

5.1. TIPURI ȘI SUBTIPURI

- În lumea reală obiectele sunt de obicei clasificate. Astfel vorbim despre animale vertebrate și nevertebrate, despre licee teoretice, colegii, grupuri școlare etc. E normal ca în modelarea bazelor de date să putem modela și astfel de clasificări.
- Un **subtip** sau o **subentitate** este o clasificare a unei entități care are caracteristici comune cu entitatea generală, precum atribute și relații. Subtipurile se reprezintă în cadrul hărții relațiilor ca entități în interiorul altei entități. Atributele și relațiile comune tuturor subtipurilor se vor reprezenta la nivelul **supertipului**, sau **superentității**. Atributele și relațiile supertipului vor fi moștenite de către subtipuri.
- Un subtip poate avea la rândul său alte subtipuri incluse.



Exemplu de folosire a subtipurilor și supertipurilor

ANGAJAT

id
* nume
* adresa
* data_nasterii

SECRETARA

MANAGER

ALTUL

REPREZENTANT_VÂNZĂRI

* zona_vanzări
* permis_conducere

DEPARTAMENT


conduce

condus de

lucrează la

are



- Subtipurile trebuie să respecte două reguli importante:
 - **trebuie să acopere toate cazurile posibile de instanțe ale supertipului**, cu alte cuvinte, orice instanță a supertipului trebuie să aparțină unui subtip. De multe ori ERD-urile includ un subtip "ALTUL" pentru a acoperi toate situațiile, și pentru a permite viitoare dezvoltări ale modelului.
 - subtipurile **trebuie să se excludă reciproc**. Această regulă se traduce pe exemplul de mai sus în faptul că un angajat nu poate fi, de exemplu, și manager și secretară în același timp. 

5.2. MAPAREA TIPURILOR ȘI SUBTIPURILOR

- Nici un sistem de gestiune a bazelor de date nu suportă în mod direct supertipurile și subtipurile. Putem adopta mai multe soluții ale acestei probleme. Vom exemplifica aceste variante pentru schema din [figura anterioară](#), în care, pentru simplitate, vom presupune că nu avem nevoie de subentitatea **ALTUL**.



VARIANTA 1 . vom crea o tabelă pentru supertip și câte o tabelă pentru fiecare subtip. diagramele de tabelă în acest caz vor fi:

Tabela **ANGAJAȚI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Id_angajat	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Data_nasterii	Date		*
Id_departament	Number	Fk	*

Tabela **SECRETARE**

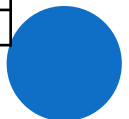
Numele coloanei	Tip	Tip cheie	Opționalitatea
Id_angajat	Number	Pk	*

Tabela **MANAGERI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Id_angajat	Number	Pk	*
Bonus	Number		*
Id_depart_conducus	Number	Fk	o

Tabela **REPREZENTANȚI_VÂNZĂRI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Id_angajat	Number	Pk	*
Zona_vanzari	Varchar2		*
Permis_conducere	Varchar2		*



- Am notat cu **Id_depart_condus** codul departamentului pe care îl conduce un manager, iar cu **Id_departament** codul departamentului în care lucrează un anumit angajat.
- Cheia primară a supertipului va fi inclusă în toate tabelele corespunzătoare subtipurilor și va deveni cheia primară a acelei tabele.
- Atributele și cheile străine provenite din relațiile de la nivelul supertipului vor fi memorate în tabela corespunzătoare supertipului. Atributele și relațiile de la nivel de subtip, se vor memora doar în tabela corespunzătoare subtipului respectiv.
- Acest model este cel mai natural dar poate crea multe probleme privind eficiența întrucât sunt necesare multe operații de interogare din tabele multiple, pentru a obține informații suplimentare despre toți angajații.



VARIANTA 2. Vom crea câte o tabelă pentru fiecare subtip. Atributele și cheile străine provenite din relațiile de la nivelul supertipului vor fi introduse în fiecare tabelă astfel obținută, acestea fiind moștenite de către fiecare subtip.

Tabela **SECRETARE**

Numele coloanei	Tip	Tip cheie	Opționalitate
Id_angajat	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Id_departament	Number	Fk	*
Data_nasterii	Date		*

Tabela **MANAGERI**

Numele coloanei	Tip	Tip cheie	Opționalitate
Id_angajat	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Data_nasterii	Date		*
Bonus	Number		*
Id_depart_condus	Number	Fk	o
Id_departament	Number	Fk	*

Tabela **REPREZENTANȚI_VÂNZĂRI**

Numele coloanei	Tip	Tip cheie	Opționalitate
Id_angajat	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Data_nasterii	Date		*
Id_departament	Number	Fk	*
Zona_vanzari	Varchar2		*
Permis_conducere	Varchar2		*



VARIANTA 3. Vom crea o singură tabelă pentru supertip. Această tabelă va conține toate coloanele corespunzătoare atributelor de la nivelul supertipului, dar și toate coloanele corespunzătoare tuturor atributelor din toate subtipurile. Atributele de la nivelul supertipului își vor păstra opționalitatea, însă atributele de la nivelul subtipurilor, vor fi toate introduse în tabelă, dar vor fi toate opționale. Relațiile de la nivelul supertipului se transformă normal. Relațiile de la nivelul subtipurilor se vor implementa cu ajutorul cheilor străine opționale.

Tabela **ANGAJAȚI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Id_angajat	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Id_departament	Number	Fk	*
Data_nasterii	Date		*
Bonus	Number		o
Id_depart_condus	Number	Fk	o
Zona_vanzari	Varchar2		o
Permis_conducere	Varchar2		o
Tip_angajat	Numeric		*



- Am introdus un atribut suplimentar **Tip_angajat**, cu ajutorul căruia vom codifica dacă un angajat este manager, secretară sau reprezentant de vânzări. Deoarece attributele de la nivelul subtipurilor sunt obligatorii pentru subtipul respectiv, va trebui să stabilim o regulă de integritate la nivel de înregistrare, care să verifice că pentru o înregistrare de un tip anume sunt completate câmpurile corespunzătoare. De exemplu, la adăugarea unui nou manager în tabela **ANGAJAȚI**, trebuie să verificăm dacă este completat câmpul bonus.
- Se observă că vor fi multe câmpuri cu valoarea null, ceea ce înseamnă o risipă de spațiu de memorie.
- În acest tabel am codificat managerii cu 1, reprezentanții de vânzări cu 2, iar secretarele cu 3. Așadar această variantă de implementare este convenabilă când există puține attribute și relații la nivelul subtipurilor.

Tabela **ANGAJAȚI**

Id_angajat	Bonus	Id_departament_condus	Zona_vanzari	Permis_conducere	Tip_angajat	...
10	125	5	(null)	(null)	1	
121	(null)	(null)	Transilvania	568147	2	
245	(null)	(null)	(null)	(null)	3	
...						

5.3. RELAȚII EXCLUSIVE (ARCE)

- În unele situații relațiile se pot exclude reciproc adică dintr-un grup de relații la un moment dat doar una dintre ele poate avea loc. De exemplu, un cont anume la o bancă este deținut fie de o persoană fizică fie de o firmă, nu de ambele la tipuri de clienți simultan.
- Un grup de relații exclusive este reprezentat în ERD printr-un arc peste relațiile care fac parte din grupul respectiv. Toate relațiile ce fac parte din grupul de relații exclusive trebuie să aibă aceeași opționalitate. Un arc aparține unei singure entități adică va include doar relații care pleacă de la aceeași entitate. O entitate poate avea mai multe arce dar o relație nu poate face parte decât dintr-un singur arc.

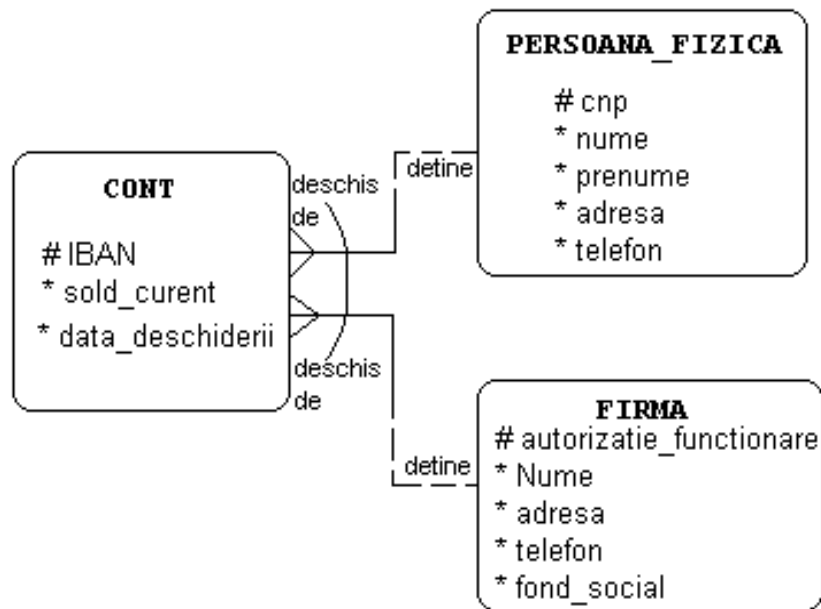


○ Există două tipuri de relații exclusive:

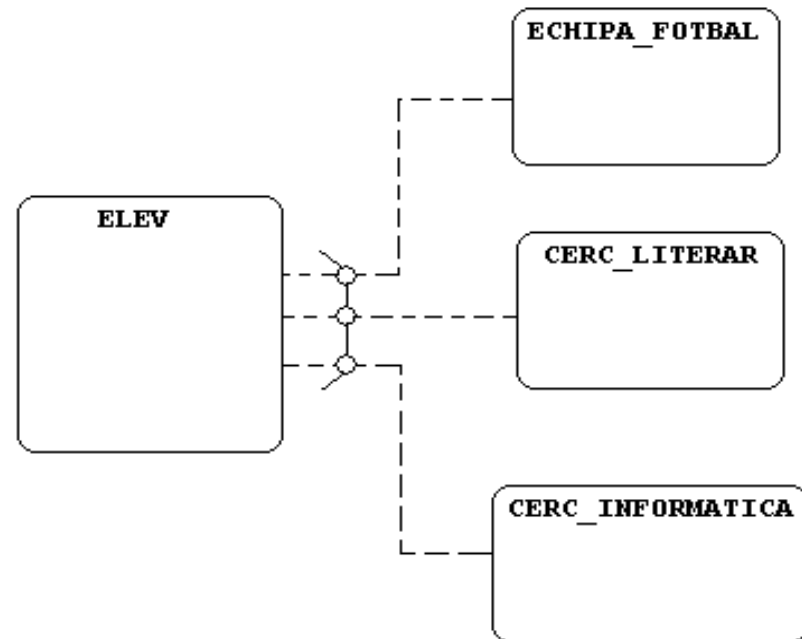
- **relații exclusive obligatorii** – în care toate relațiile ce fac parte din arcul respectiv sunt obligatorii ceea ce înseamnă că de fiecare dată una din relații are obligatoriu loc
- **relații exclusive opționale** – în care toate relațiile ce face parte din arc sunt opționale. În acest caz de fiecare dată are cel mult una din relații existând varianta ca pentru o instanță a entității arcului să nu aibă loc niciuna dintre relațiile din grupul respectiv.



Relații exclusive obligatorii



Relații exclusive opționale



- Pentru a mapa un arc vom crea atâtea chei străine câte relații există în arcul respectiv. Pentru modelul ce conține relații exclusive obligatorii, vom obține următoarele tabele:

Tabela **CONTURI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
IBAN	Number	Pk	*
Sold_curent	Number		*
Data_deschiderii	Date		*
Cnp	Number	Fk1	o
Autorizatie_functionare	Number	Fk2	o

Tabela **PERSOANE_FIZICE**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Cnp	Number	Pk	*
Nume	Varchar2		*
Prenume	Varchar2		*
Adresa	Varchar2		*
Telefon	Number		*

Tabela **FIRME**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Autorizatie_functionare	Number	Pk	*
Nume	Varchar2		*
Adresa	Varchar2		*
Telefon	Number		*
Fond_social	Number		*



- Deși relațiile din arc sunt obligatorii, cheile străine corespunzătoare au fost setate ca fiind opționale, deoarece pentru fiecare înregistrare trebuie să avem completată una din cele două chei străine, iar cealaltă cheie străină trebuie să rămână necompletată (principiul exclusivității). Va trebui să implementăm o condiție de integritate care să verifice această condiție.

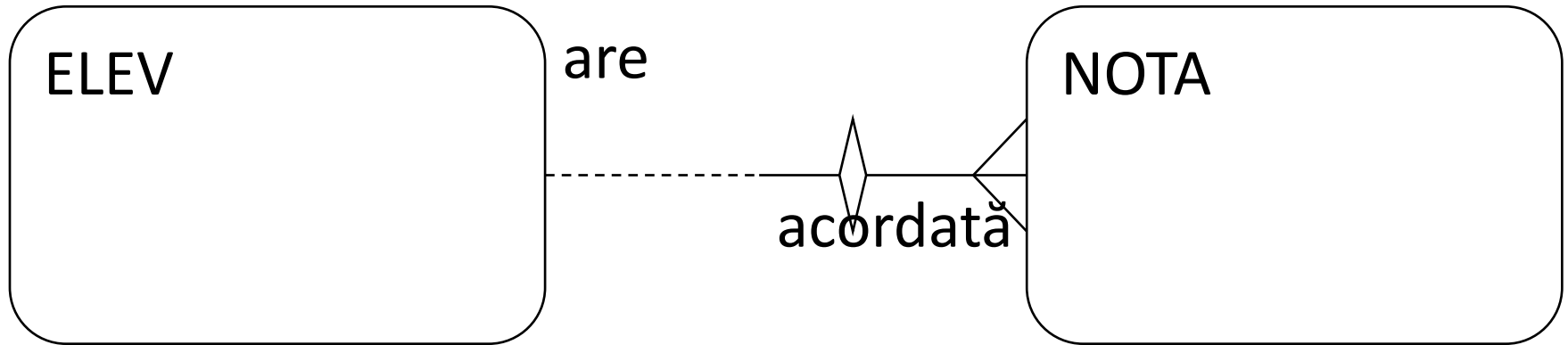


5.5. NOSTRANSFERABILITATEA

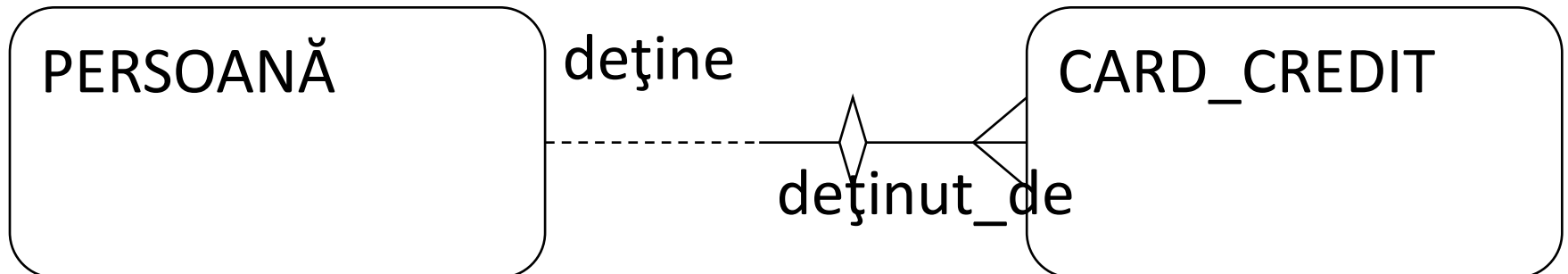
- O relație este netransferabilă dacă o asociație între două instanțe ale celor două entități odată stabilită nu mai poate fi modificată. Nontransferabilitatea unei relații se reduce la faptul că valorile cheii străine corespunzătoare relației respective nu pot fi modificate.
- Nontransferabilitatea anumitor relații poate proveni uneori din reguli speciale ale afacerii modelate. Relațiile ce se referă la informații financiare sau la informații ce pot fi expuse fraudelor pot fi setate ca nontransferabile.
- În ERD, o relație nontransferabilă se notează cu un romb pe linia corespunzătoare relației, înspre entitatea a cărei cheie-străină nu este permis să o modificăm (adică în partea cu manz a unei relații one to many)



Exemplu de relație nontransferabilă: este vorba de notele date elevilor. Este normal ca o notă acordată unui elev să nu fie transferată altui elev



Exemplu de relație nontransferabilă: proprietarul unui card bancar nu se poate modifica

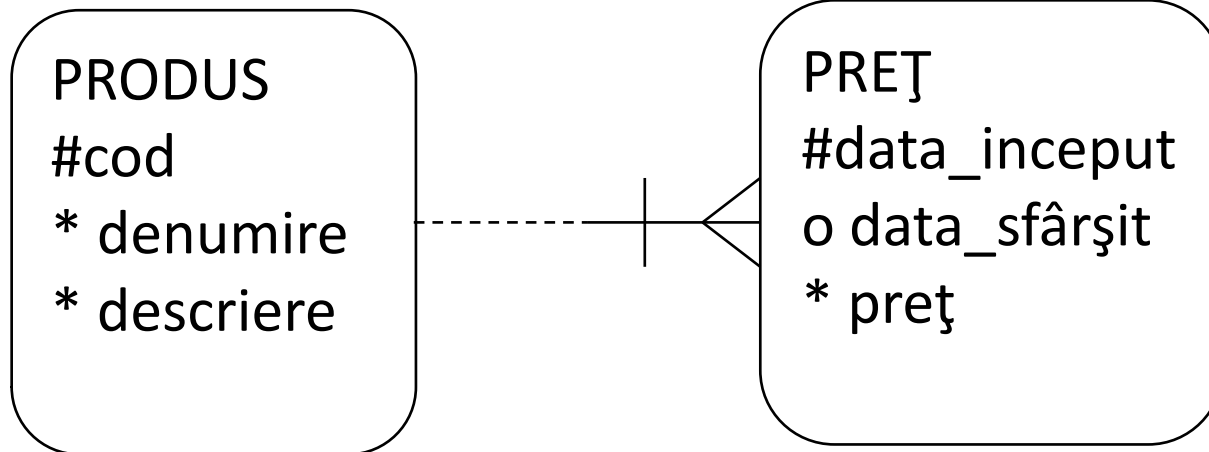


5.6. MODELAREA DATELOR ISTORICE

- Orice lucru se schimbă în decursul timpului, chiar și relațiile dintre obiecte. Datele dintr-o bază de date pot suferi modificări.
- Exemplu 1: un investitor care dorește să cumpere acțiuni, studiază care acțiuni se vând mai bine, care sunt rezultatele diferitelor firme de pe piață pe o anumită perioadă de timp și pe baza acestor date va încerca să facă o previziune în timp privind viitorul firmelor și acțiunilor respective, lucru bazat pe istoricul firmelor și acțiunilor respective.
- Exemplu 2: într-o echipă de fotbal, la achiziționarea unor jucători noi este nevoie de date legate de echipele la care a jucat un anumit jucător și ce performanțe a avut acolo, ce accidentări etc.
- Exemplu 3: prețul produselor suferă modificări des. Factorii care duc la aceste modificări sunt: inflația, anotimpul etc. Atributul preț din entitatea produs își modifică valoarea în permanență de-a lungul timpului.

PREȚ
cod
* denumire
* preț
* descriere

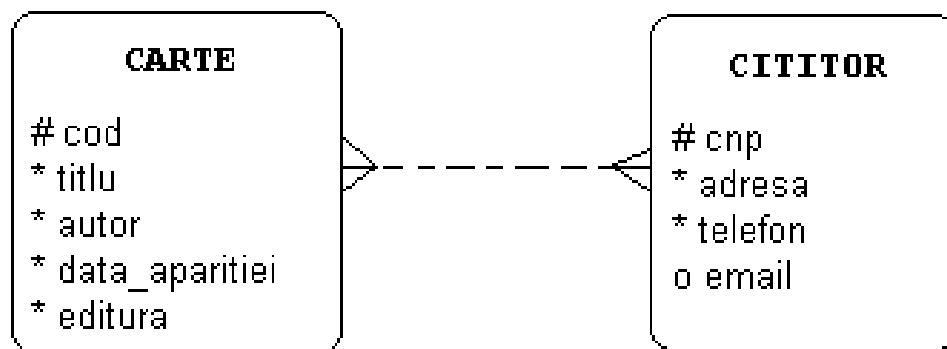
- Exemplu de model simplu:
- Dacă este important să reținem un istoric al prețurilor pentru fiecare produs, atunci atributul preț se va transforma într-o nouă entitate:



Relația dintre două entități

Atributul `data_sfârșit` este opțional deoarece data până la care este valabil prețul curent nu este de obicei cunoscut

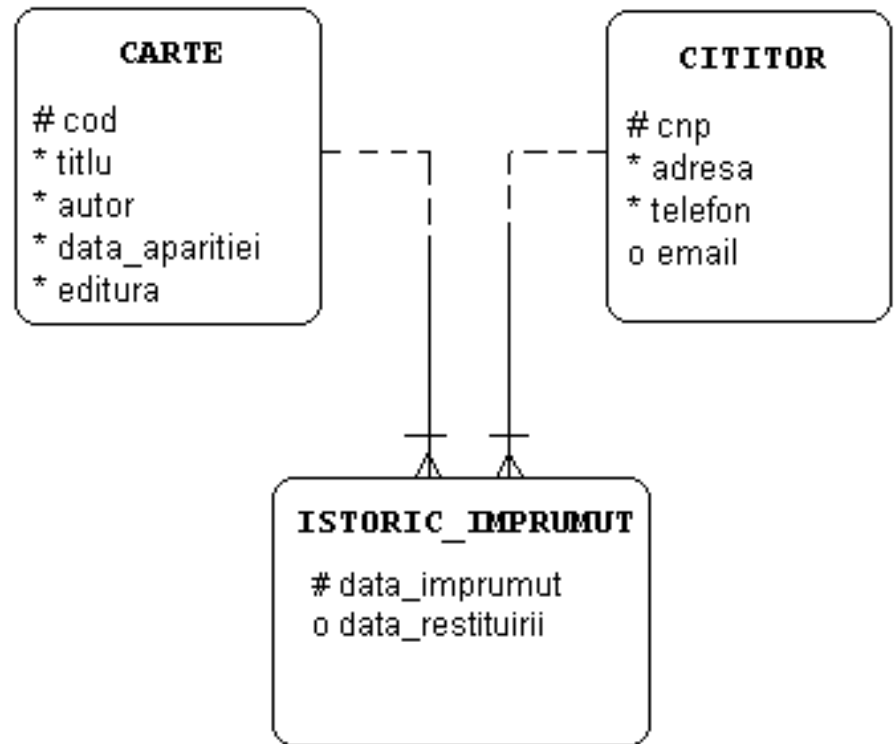
- Pentru o bibliotecă, memorarea datelor legate de împrumuturi are nevoie de păstrarea unui istoric al împrumuturilor deoarece pe baza lor se pot afla domeniile de interes ale cititorilor și se vor face achiziții de carte în viitor, se poate determina uzura cărților ce trebuie înlocuite etc.



Exemplu de relație M:M



- Relația M:M se rezolvă prin crearea unei entități de intersecție **ISTORIC_IMPRUMUT**



Cazul 1 de barare incorectă a relațiilor



- Dacă se mapează acest ERD, tabela corespunzătoare diagramei va fi tabela **ISTORIC_IMPRUMUTURI**:

Numele coloanei	Tip	Tip_cheie	Opționalitate
Cod_carte	Number	Pk,fk	*
Cnp	Number	Pk,fk	*
Data_imprumut	Date	pk	*
Data_restituire	date		0

- Cheia primară din tabela este formată din combinația coloanelor **cod_carte**, **cnp cititor** și **data_imprumut**. Combinația celor trei trebuie să fie unică pentru fiecare înregistrare în parte.
- Bararea celor două relații dintre entitatea de intersecție nu este totdeauna o soluție corectă.



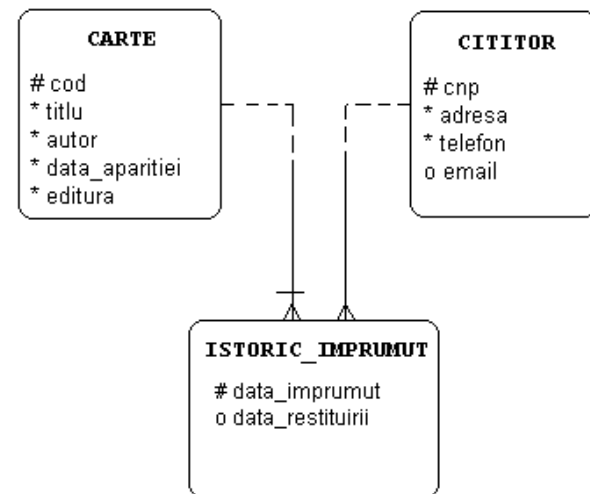
- Cheia primară nu este corect aleasă. Se poate încerca bararea relației dintre **ISTORIC_IMPRUMUTURI** și **CITITOR**
- În această situație, cheia primară este formată din câmpurile **data_împrumut** și **cnp-ul** cititorului. Aceasta înseamnă că un cititor nu poate împrumuta la o anumită dată decât o singură carte – lucru nerealist. Sau doi cititori pot împrumuta în aceeași zi, aceeași carte.
- Deci nici această soluție nu este corectă.



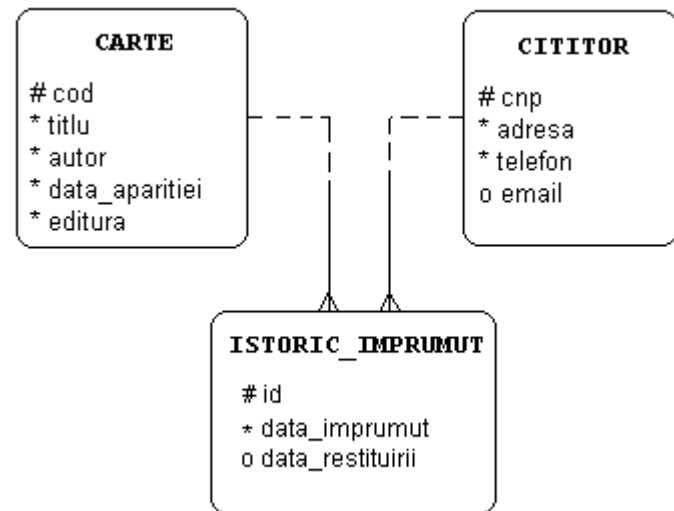
- Dacă nu se barează nici una dintre relații, situația este și mai rea deoarece aceasta înseamnă că biblioteca poate împrumuta într-o singură zi o singură carte.



- Cazul corect este cel al barării relației dintre CARTE și ISTORIC_IMPRUMUTURI.
- În această situație, cheia primară este combinația coloanelor **cod_carte** și **data_imprumut**. Un cititor poate împrumuta două sau mai multe cărți la aceeași dată, combinația celor două coloane fiind unică.
- Doi cititori nu pot împrumuta în aceeași zi aceeași carte, pentru că este încălcată regula de integritate a cheii primare.
- Un cititor poate împrumuta aceeași carte la date diferite



- Bararea automată a relațiilor dinspre entitatea de intersecție nu este totdeauna o soluție corectă.
- Pentru a evita aceste complicații, se poate recurge la introducerea unei chei artificiale în entitatea de intersecție ce face ca pentru fiecare împrumut să se completeze o fișă separată care are un număr unic.



Introducerea unei chei artificiale



În această lecție am învățat despre:

- Modelarea clasificărilor obiectelor cu ajutorul subtipurilor și a supertipurilor
- Variante de mapare a subtipurilor și supertipurilor
- Maparea relațiilor exclusive
- Maparea relațiilor nontransferabile
- Modelarea datelor care se modifică în timp

SFÂRȘIT

