

Capitolul 8. Proiectarea logică

Scop - construirea unei scheme logice ce reprezintă corect și eficient toate informațiile descrise într-o schemă entitate-relație

Etape:

- *Restructurarea schemei E-R* – fază independentă de modelul logic ales și care se bazează pe criterii de optimizare a schemei și de simplificare a următoarei etape;
- *Translarea în modelul logic* – ține cont de un anumit model logic (modelul relațional spre exemplu) și poate include alte optimizări, bazate pe caracteristicile modelului logic.

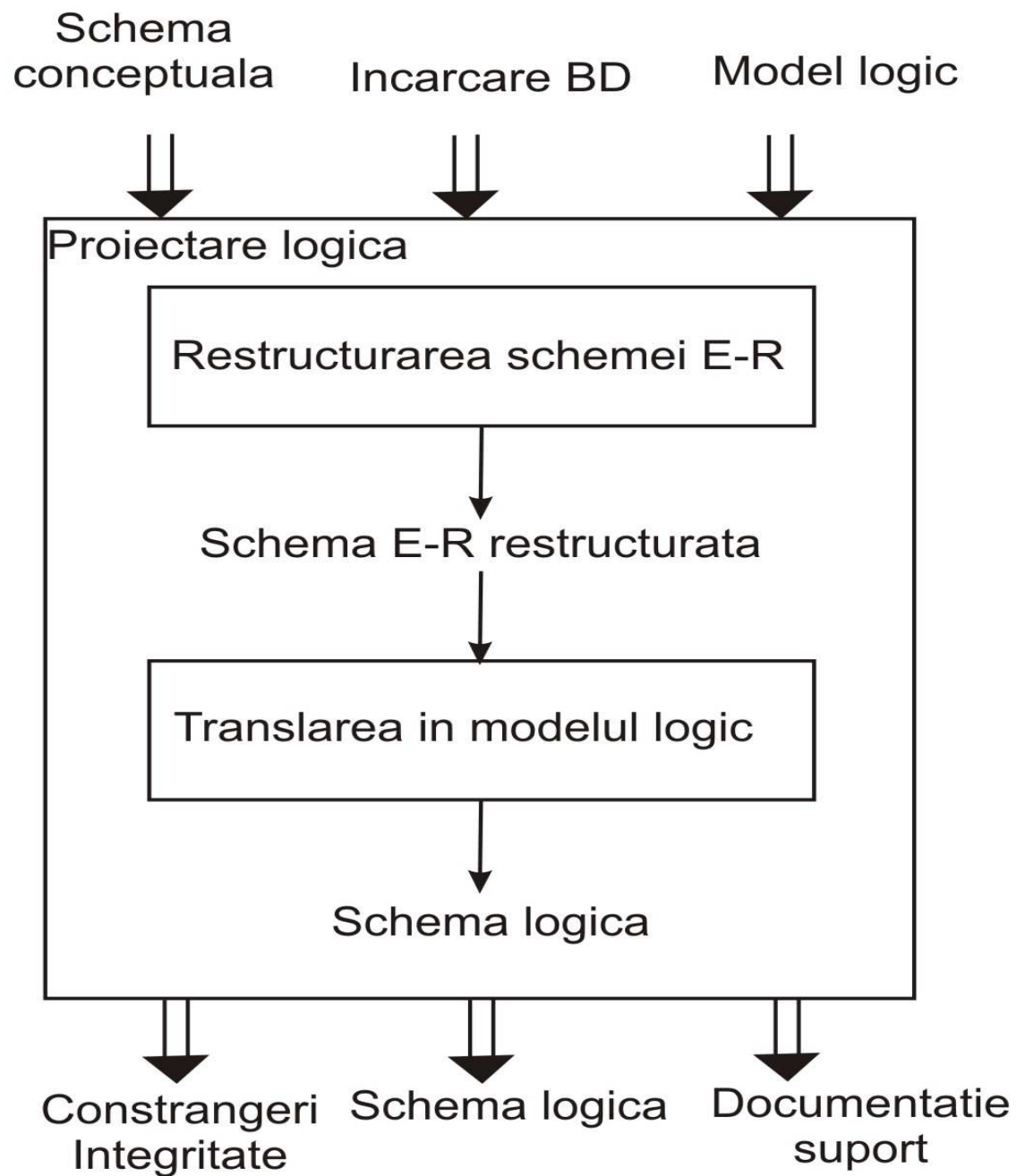


Figura 1. Proiectarea logică a unei baze de date

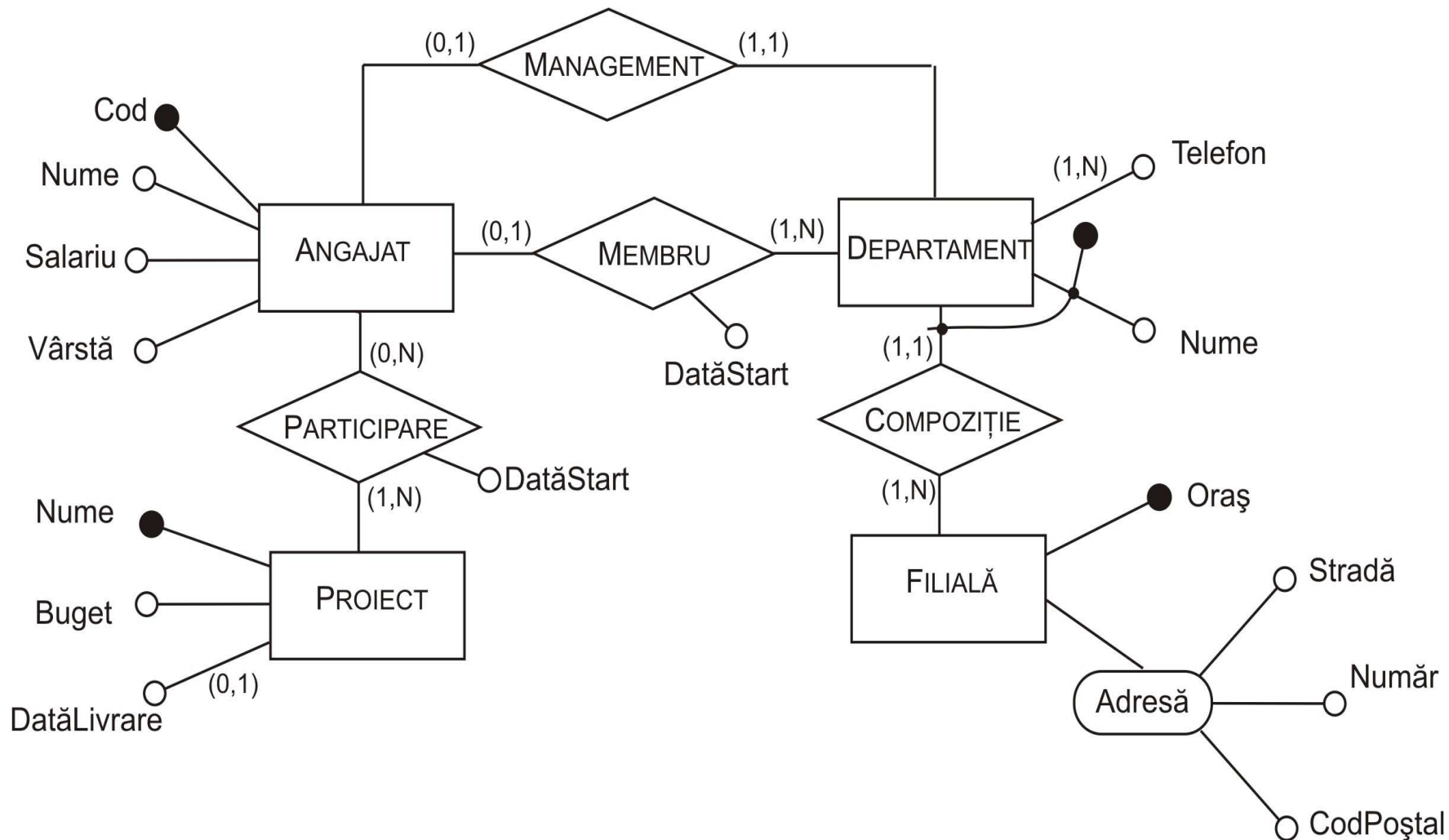
8.1 Analiza performanțelor schemei E-R

Indicatori de performanță:

- *costul unei operații* – evaluat în funcție de numărul de apariții ale entităților și relațiilor ce sunt parcurse pentru a executa o operație asupra unei baze de date
- *cerința de stocare* – evaluată în funcție de numărul de octeți necesari stocării datelor descrise de schemă

Pentru a evalua acești indicatori avem nevoie de următoarele informații:

- *volumul de date*
 - numărul de apariții ale fiecărei entități și relații din schemă
 - dimensiunea fiecărui atribut
- *caracteristicile operațiilor*
 - tipul operației (interactivă sau batch)
 - frecvența operației (numărul mediu de execuții într-un anumit interval de timp)
 - datele implicate (entități și/sau relații)



- **operația 1:** atribuie un angajat unui proiect;
- **operația 2:** găsește datele pentru un angajat, pentru departamentul în care lucrează acesta și pentru proiectele în care este implicat;
- **operația 3:** găsește datele pentru toți angajații unui anumit departament;
- **operația 4:** pentru fiecare filială, găsește departamentele, cu numele managerilor și lista angajaților din fiecare departament.

Tabelul volumelor

<i>Concept</i>	<i>Tip</i>	<i>Volu</i>
Filială	E	10
Departament	E	80
Angajat	E	2000
Proiect	E	500
Compoziție	R	80
Membru	R	1900
Management	R	80
Participare	R	6000

Tabelul operațiilor

<i>Operație</i>	<i>Tip</i>	<i>Frecvență</i>
Op 1	I	50/zi
Op 2	I	100/zi
Op 3	I	10/zi
Op 4	B	2/săptămână

În tabelul volumelor, numărul aparițiilor unei relații depinde de *doi parametri*:

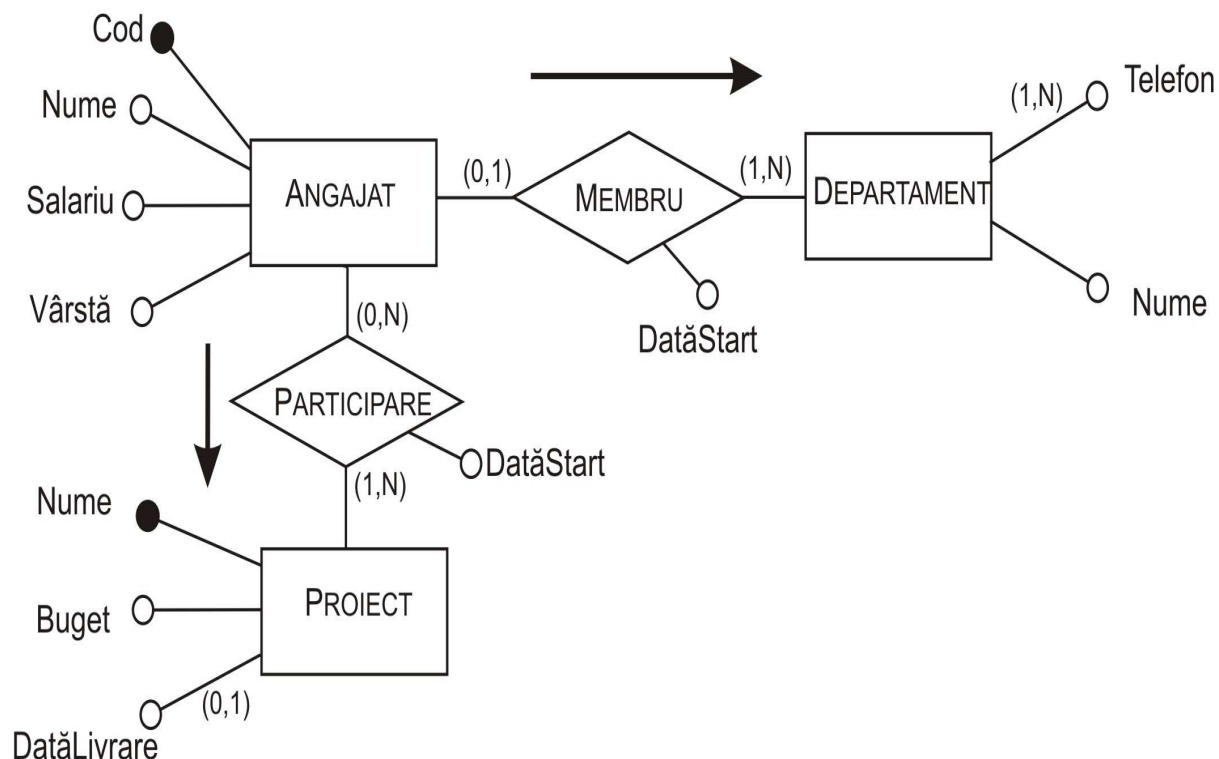
- volumul entităților implicate în relație
- de câte ori o apariție a acestor entități participă, în medie, într-o apariție a relației; acest parametru depinde de cardinalitățile relației

Exemplu:

- numărul de apariții ale relației COMPOZIȚIE este egal cu numărul departamentelor, deoarece cardinalitatea relației indică faptul că fiecare departament aparține doar unei filiale
- numărul de apariții ale relației MEMBRU este mai mic decât numărul angajaților, deoarece există angajați care nu aparțin nici unui departament
- dacă un angajat este implicat în medie în trei proiecte avem 6000 de apariții ale relației PARTICIPARE (și deci $6000/500=12$ angajați în medie pentru fiecare proiect)

Schema de navigare - constă în fragmente ale schemei E-R relevante pentru operație; este utilă desenarea „căilor logice” care trebuie urmate pentru a accesa informațiile cerute.

- **operația 2:** găsește datele pentru un angajat, pentru departamentul în care lucrează acesta și pentru proiectele în care este implicat;



Estimarea costului operației 2:

- trebuie accesată o apariție a entității ANGAJAT pentru a accesa o apariție a relației MEMBRU și, prin intermediul acesteia, a unei apariții a entității DEPARTAMENT
- pentru a obține datele referitoare la un proiect la care angajatul lucrează trebuie să accesăm în medie trei apariții ale relației PARTICIPARE (deoarece am presupus că un angajat lucrează în medie la trei proiecte); apoi, prin această relație accesăm în medie trei apariții ale entității PROIECT

Aceste informații pot fi sumate în *tabelul accesărilor*. În ultima coloană a acestui tabel se menționează tipul accesului (R pentru citire, W pentru scriere)

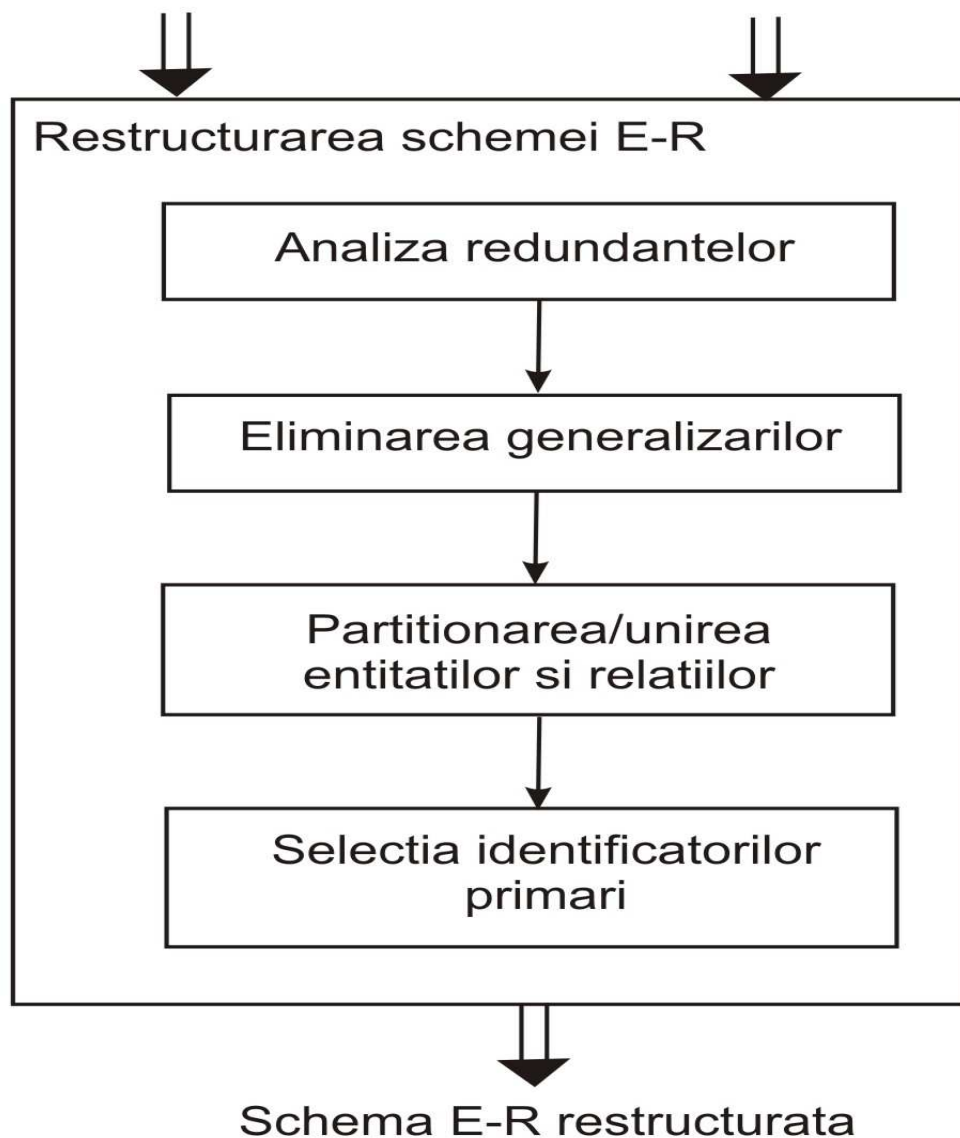
Tabelul accesărilor

<i>Concept</i>	<i>Tip</i>	<i>Accesări</i>	<i>Tip</i>
Angajat	Entitate	1	R
Membru	Relație	1	R
Departament	Entitate	1	R
Participare	Relație	3	R
Proiect	Entitate	3	R

8.2 Restructurarea schemei E-R

Schema E-R

Incarcare BD



decide ștergerea sau păstrarea unor redundanțe din schema E-R;

înlocuiește toate generalizările cu alte construcții

decide dacă este convenabilă partiționarea unui concept în mai multe sau unirea mai multor concepte separate în unul singur

alege un identificator pentru acele entități care au mai mult de un identificator

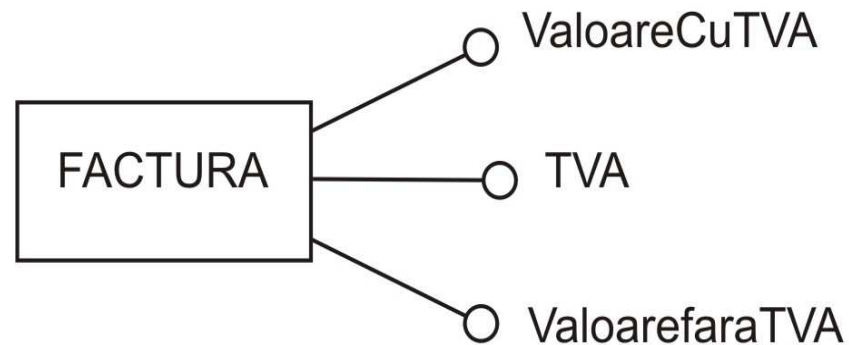
Analiza redundanțelor

Într-o schemă conceptuală o redundanță corespunde unei informații ce poate fi derivată din alte date.

Cele mai frecvente exemple sunt:

- atribute ale căror valori pot fi derivate, pentru fiecare apariție a unei entități/relații, din valorile altor atribute pentru aceeași apariție

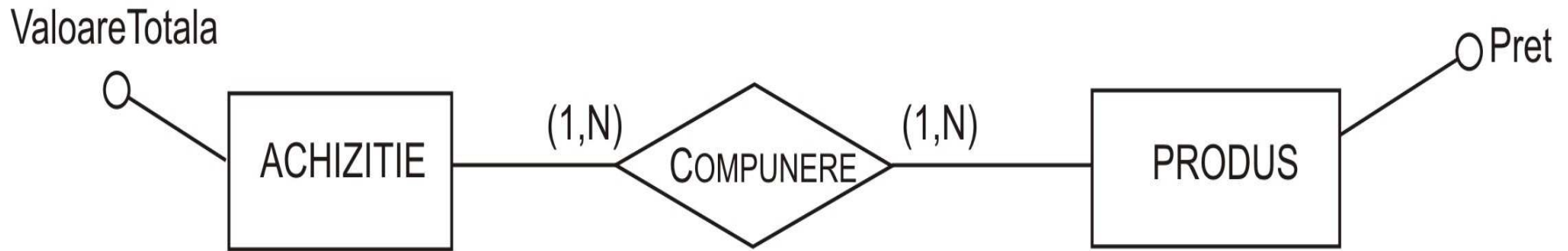
Exemplu



fiecare atribut poate fi dedus din celelalte două prin operații aritmetice

- atribute ce pot fi derivate din alte atribute aparținând altor entități/relații, de obicei prin intermediul funcțiilor agregat

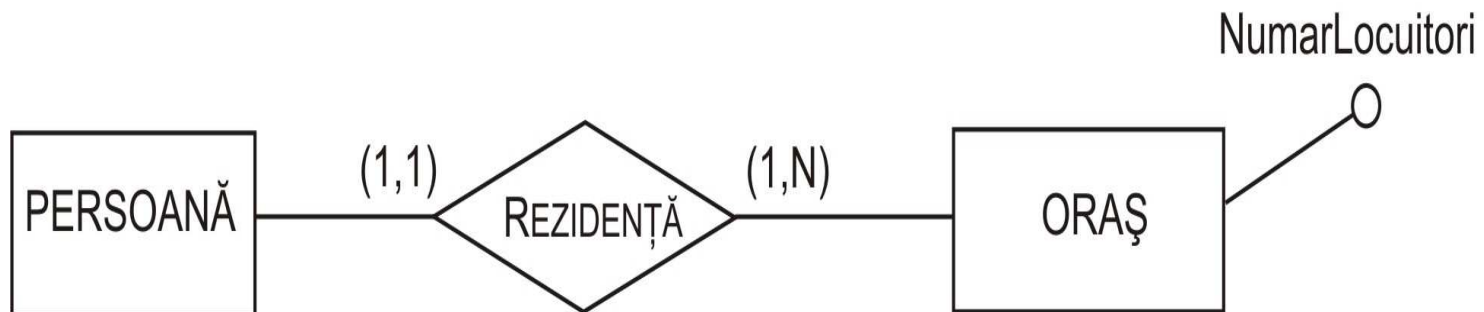
Exemplu



atributul `ValoareTotala` a entității `ACHIZITIE` poate fi calculat din valorile atributului `Pret` a entității `PRODUS` prin sumarea prețurilor produselor achiziționate, după cum se specifică în relația `COMPUNERE`.

- atribute ce pot fi derivate din operații de numărare a aparițiilor

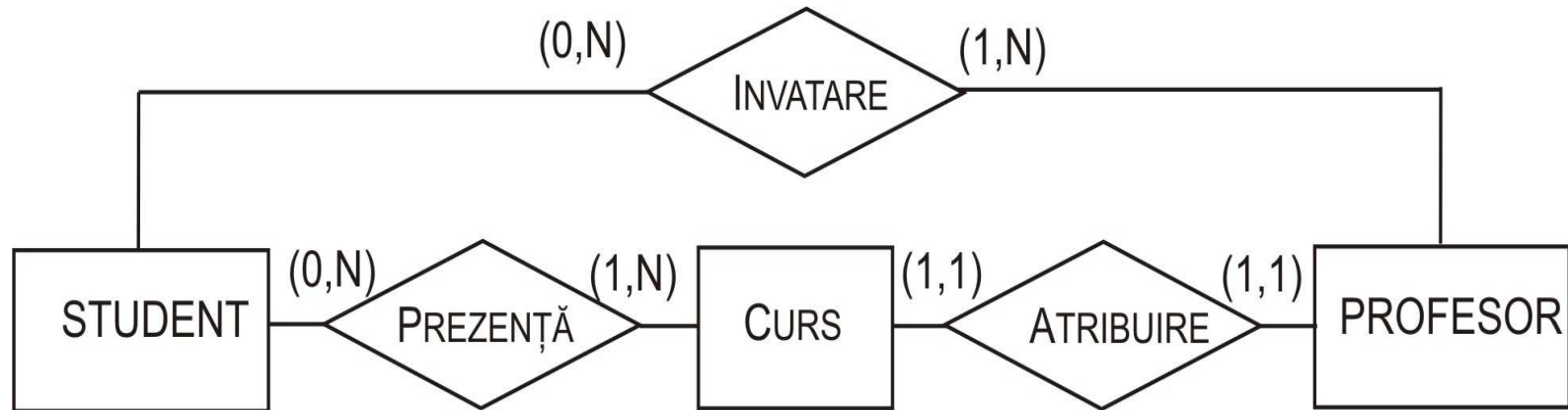
Exemplu



atributul `NumarLocuitori` poate fi obținut prin numărarea aparițiilor relației `REZIDENȚĂ` în care apare un anumit oraș.

- relații ce pot fi derivate din alte relații în prezența ciclurilor

Exemplu



Relația INVATARE dintre studenți și profesori poate fi derivată din relațiile PREZENȚĂ și ATRIBUIRE.

Observație. Prezența ciclurilor nu generează în mod automat redundanțe

Spre exemplu, dacă se înlocuiește relația INVATARE cu relația SUPERVIZARE reprezentând legătura dintre studenți și supervizori, atunci schema nu mai este redundantă.

Prezența informațiilor derivate într-o bază de date prezintă un avantaj și totodată dezavantaje.

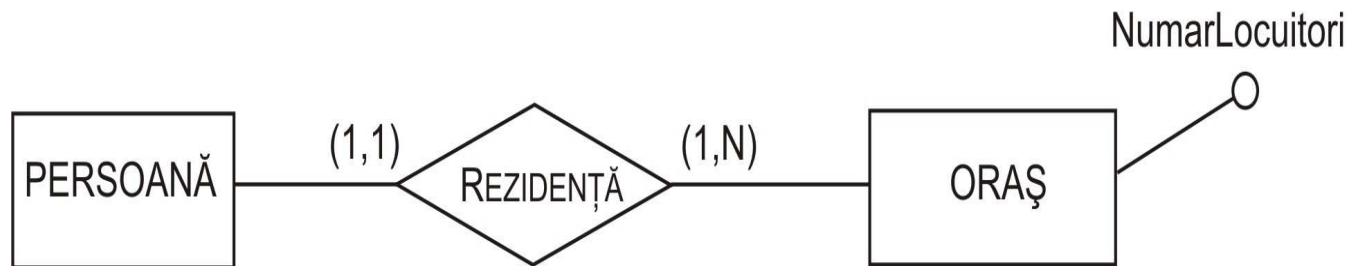
Avantajul este reducerea numărului de accesări necesare obținerii informațiilor derivate.

Dezavantajele ar fi:

- creșterea cererii de stocare;
- necesitatea efectuării de operații suplimentare pentru actualizarea informațiilor derivate.

Decizia menținerii sau eliminării unei redundanțe poate fi luată prin compararea costului operațiilor ce implică informația redundantă și capacitatea de stocare necesară în cazul prezenței, respectiv absenței redundanței.

Exemplu



- *operația 1*: adaugă o persoană nouă cu orașul de rezidență al acesteia;
- *operația 2*: afișează toate datele referitoare la oraș (inclusiv numărul de locuitori).

Presupunem că încărcarea bazei de date este dată în tabelele de mai jos.

Tabelul volumelor

<i>Concept</i>	<i>Tip</i>	<i>Volum</i>
Oraș	E	200
Persoană	E	1000000
Rezidență	R	1000000

Tabelul operațiilor

<i>Operație</i>	<i>Tip</i>	<i>Frecvență</i>
Op 1	I	500/zi
Op 2	I	2/zi

Volumul relației REZIDENȚĂ este egal cu volumul entității PERSONA deoarece cardinalitățile indică faptul că fiecare persoană are rezidența într-un singur oraș.

1. cazul prezenței redundanței (atributul `NumarLocuitori`)

Presupunem că pentru stocarea numărului de locuitori dintr-un oraș sunt necesari 4 octeți \Rightarrow datele redundante necesită $4 \times 200 = 800$ octeți (nesemnificativ).

Costul operațiilor

Pentru operația 1 (adaugă o persoană nouă cu orașul de rezidență al acesteia):

- 1 operație de scriere în entitatea PERSOANĂ (adăugare persoană);
- 1 operație de scriere în relația REZIDENȚĂ (adăugare pereche persoană-oraș);
- 1 operație de citire din entitatea ORAȘ (găsire oraș);
- 1 operație de scriere în entitatea ORAȘ (actualizare număr de locuitori).

Tabelul accesărilor (în prezența redundanțelor)

Operația 1			
Concept	Tip	Accesări	Tip
Persoana	E	1	W
Rezidență	R	1	W
Oraș	E	1	R
Oraș	E	1	W

Operația 1 necesită $3 \times 500 = 1500$ accesări în scriere / zi
 $1 \times 500 = 500$ accesări în citire / zi

Pentru operația 2 (afișează toate datele referitoare la oraș):

- 1 operație de citire din entitatea ORAȘ

Tabelul accesărilor (în prezența redundanțelor)

Operația 2			
<i>Concept</i>	<i>Tip</i>	<i>Accesări</i>	<i>Tip</i>
Oraș	E	1	R

Operația 2 necesită $1 \times 2 = 2$ accesări în citire / zi (neglijabil)

Presupunând că accesul în scriere costă de două ori mai mult decât accesul în citire, numărul de accesări / zi în prezența redundanțelor este $2 \times 1500 + 500 = 3500$.

2. cazul absenței redundanței

Costul operațiilor

Pentru operația 1 (adaugă o persoană nouă cu orașul de rezidență al acesteia):

- 1 operație de scriere în PERSOANA;
- 1 operație de scriere în REZIDENȚĂ.

Tabelul accesărilor (în absența redundanțelor)

Operația 1			
Concept	Tip	Accesări	Tip
Persoana	E	1	W
Rezidență	R	1	W

Operația 1 necesită $2 \times 500 = 1000$ accesări în scriere / zi

Pentru operația 2 (afișează toate datele referitoare la oraș):

- 1 operație de citire din ORAȘ (neglijabil)
- în medie 5000 de operații de citire din REZIDENȚĂ pentru a determina numărul de locuitori ($5000 = 1000000 \text{ persoane} / 200 \text{ orașe}$)

Tabelul accesărilor (în absența redundanțelor)

Operația 2			
Concept	Tip	Accesări	Tip
Oraș	E	1	R
Rezidență	R	5000	R

Operația 2 necesită $2 \times 5000 = 10000$ accesări în citire / zi

Numărul de accesări / zi în absența redundanțelor este de $2 \times 1000 + 10000 = 12000$.

Se observă că în al doilea caz sunt necesare 8500 accesări / zi în plus pentru a „salva” mai puțin de un kilo-octet.

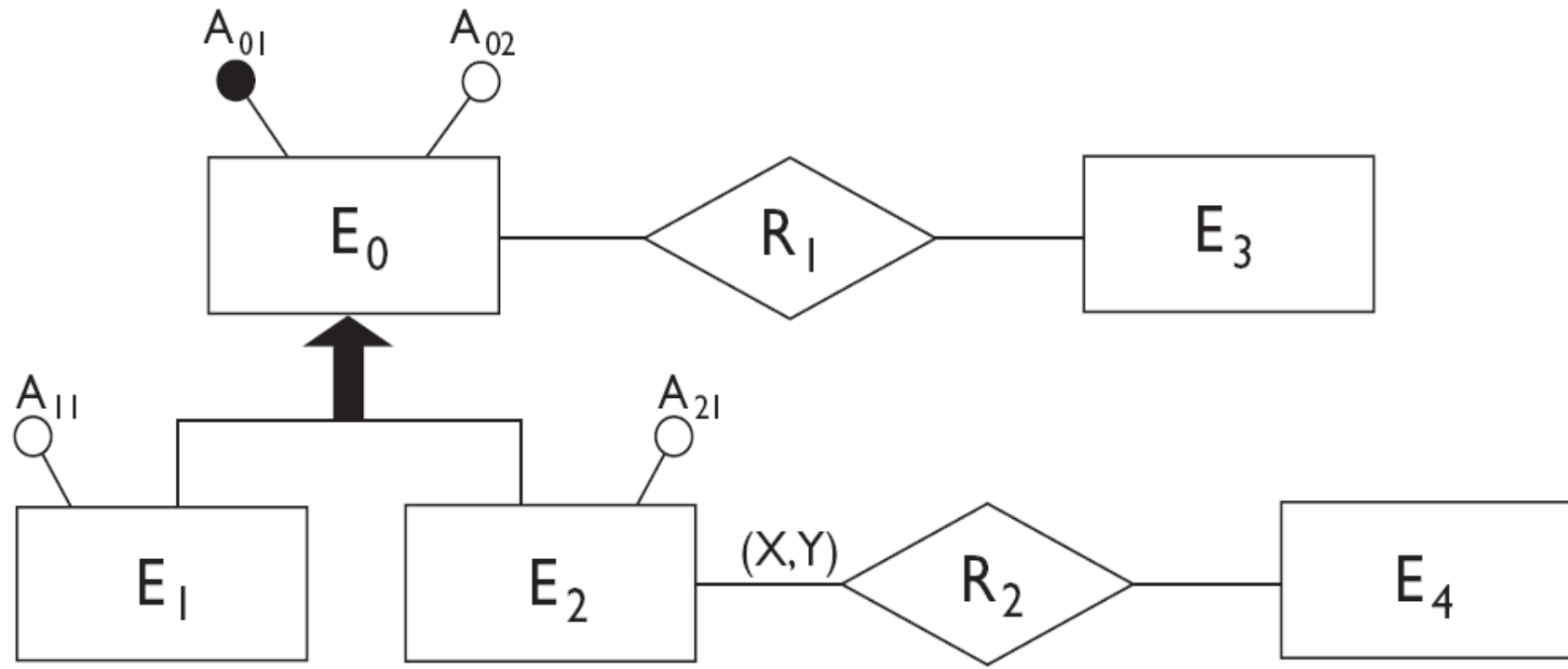
Putem trage concluzia că este mai convenabil să menținem redundanța în cazul acestei probleme.

Eliminarea generalizărilor

Deoarece modelul relațional nu permite reprezentarea directă a generalizărilor din modelul E-R apare necesitatea transformării acestor construcții în alte construcții ce pot fi translatate cu ușurință.

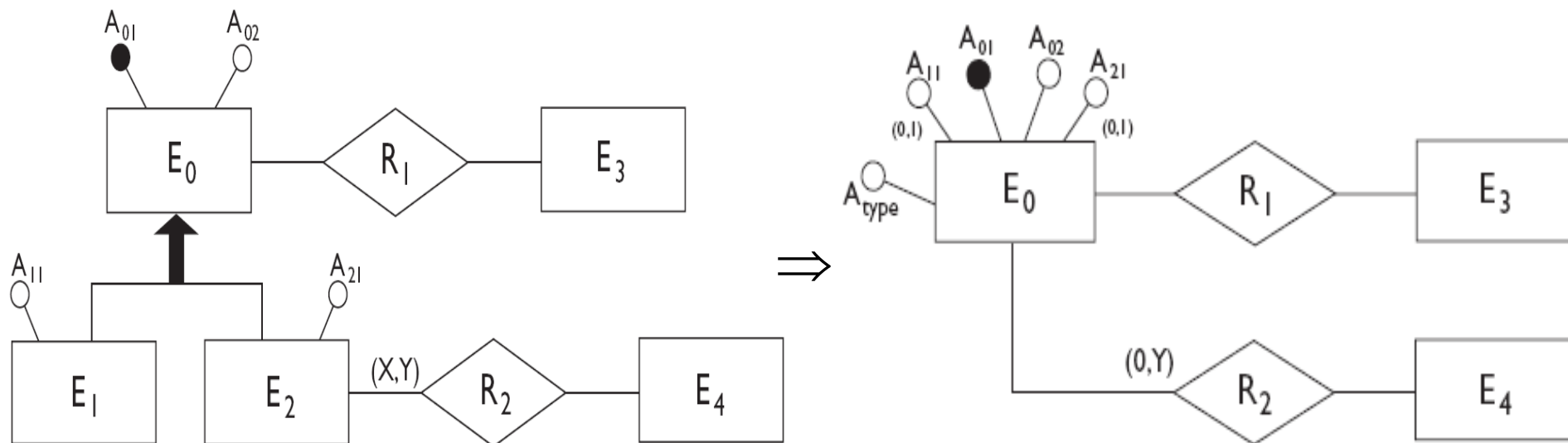
Există **trei posibilități de reprezentare** a unei generalizări cu ajutorul entităților și relațiilor.

Pentru înțelegerea acestor moduri de transformare să considerăm schema următoare



Exemplu de schema cu generalizare

1. Înglobarea entităților copil în entitatea părinte

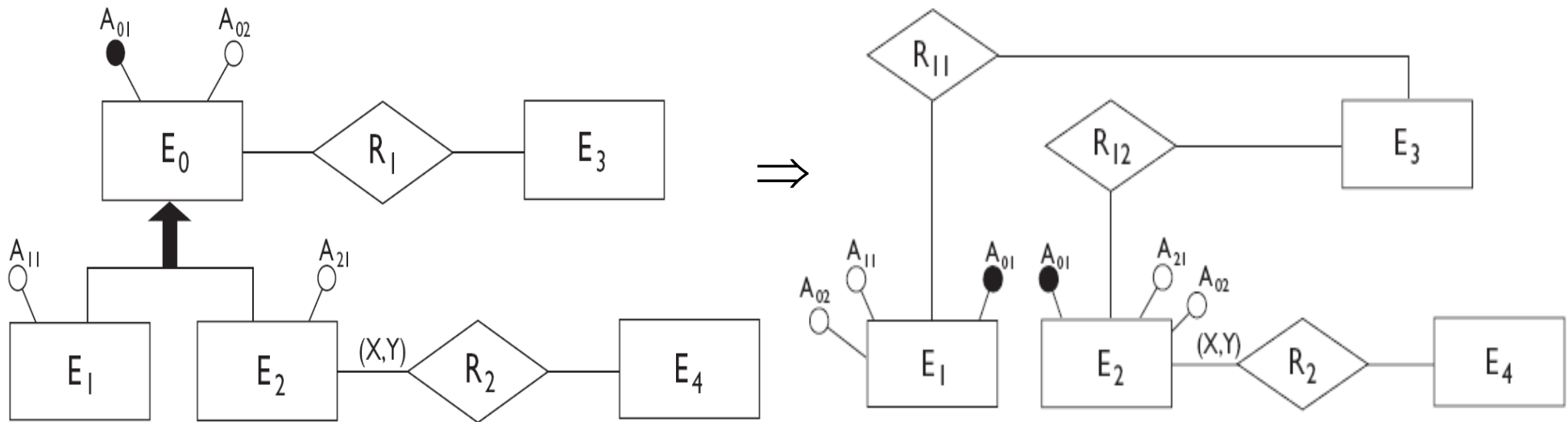


- Se elimină E_1 și E_2 și proprietățile acestora sunt adăugate entității E_0 .
- Se adaugă atributul A_{type} lui E_0 pentru a distinge tipul (E_1 sau E_2) unei apariții a lui E_0 .
- A_{11} și A_{21} pot avea valoarea NULL pentru unele apariții ale entității E_0 .
- Relația R_2 va avea cardinalitatea minimă 0 pentru E_0 deoarece aparițiile lui E_2 formează o submulțime a aparițiilor entității E_0 .

Se folosește când operațiile implică apariții și atribute ale E_0 , E_1 și E_2 într-un mod asemănător.

- În acest caz, deși se stochează valori NULL, alegerea asigură mai puține accesări în comparație cu celelalte variante în care aparițiile și atributele sunt distribuite celorlalte entități.

2. Înglobarea entității părinte în entitățile copil

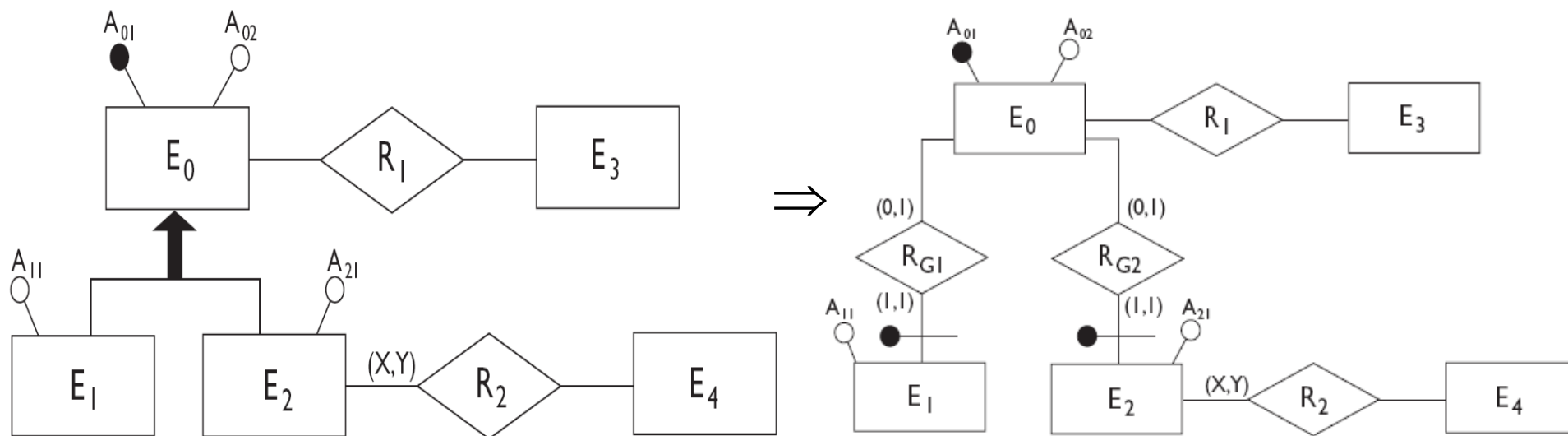


- Entitatea părinte E_0 este eliminată și, prin proprietatea de moștenire, attributele sale, identificatorii și relațiile în care era implicată sunt adăugate entităților copil E_1 și E_2 .
- Relațiile R_{11} și R_{12} reprezintă restricția relației R_1 pe aparițiile entității E_1 , respectiv E_2 .

Este posibilă doar dacă generalizarea este totală, altfel aparițiile lui E_0 care nu sunt apariții nici ale lui E_1 nici ale lui E_2 nu vor putea fi reprezentate.

- Această metodă este utilă când există operații care se referă doar la aparițiile lui E_1 sau ale lui E_2 și astfel se face distincția între aceste entități.
- Nu mai există în principiu attribute ce pot lua valori NULL.
- Numărul de accesări este mai redus în comparație cu a treia metodă, deoarece nu este necesară accesarea lui E_0 pentru a accesa attribute ale E_1 și E_2 .

3. Substituirea unei generalizări cu relații

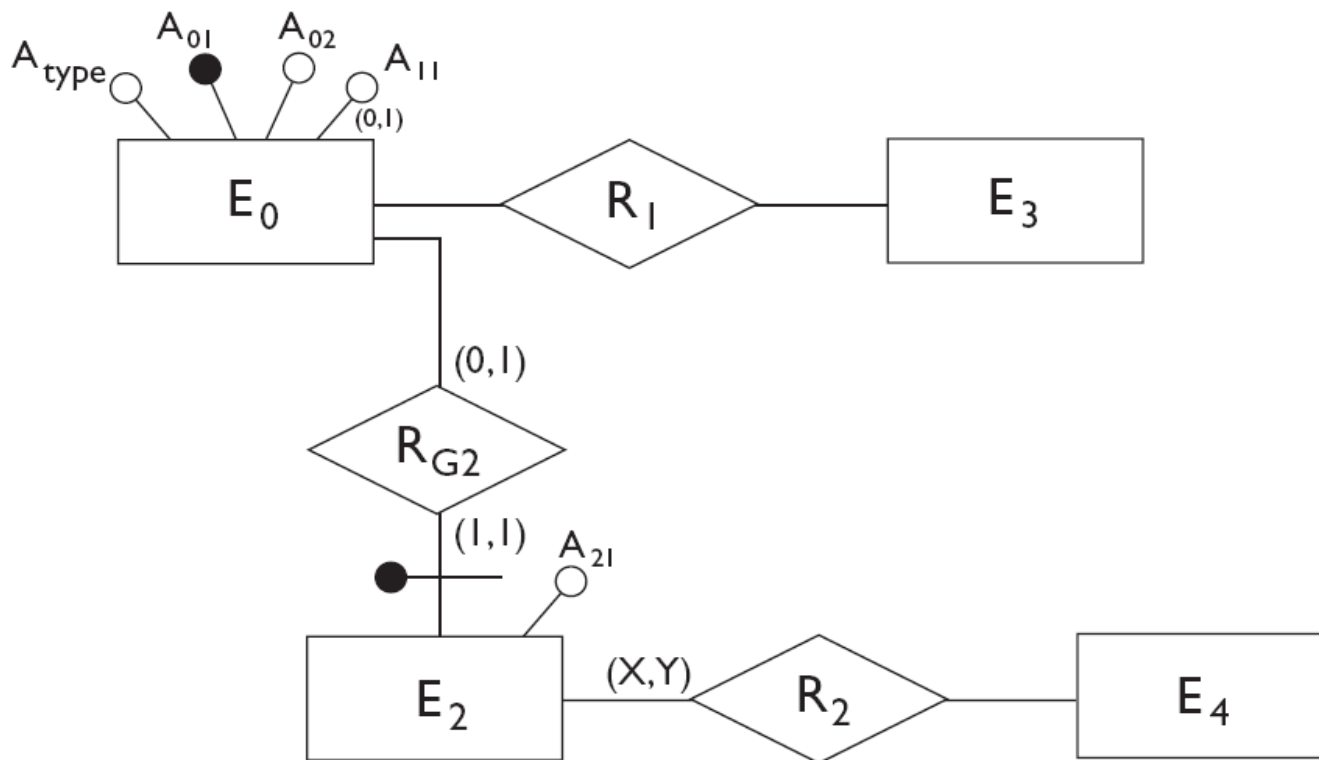


- Generalizarea este transformată în două relații „una la unu”
- Nu există transfer de atribute sau relații și entitățile E_1 și E_2 sunt identificate extern prin entitatea E_0 .
- În noua schemă apar constrângeri în plus: nici o apariție a lui E_0 nu poate participa în ambele relații R_{G1} și R_{G2} .; mai mult, dacă generalizarea este completă, fiecare apariție a lui E_0 trebuie să participe în exact una din relațiile R_{G1} și R_{G2} .

Este utilă atunci când generalizarea nu este totală și operațiile se referă fie la aparițiile și atributele lui E_1 (E_2) fie ale lui E_0

- Stocarea este mai mică în raport cu prima metodă datorită absenței valorilor NULL.
- Crește numărul accesărilor pentru a păstra consistența aparițiilor.

Opțiunile prezentate nu sunt singurele posibile.
 O variantă ar fi cea din figura următoare.



Restructurare posibilă a schemei anterioare

În acest caz, pe baza considerațiilor anterioare, s-a decis înglobarea E_0 și E_1 și lăsarea separată a lui E_2 . Atributul A_{type} este adăugat pentru a distinge aparițiile lui E_0 de cele ale E_1 .

Partiționarea / combinarea entităților și relațiilor

Creșterea eficienței accesului la datele dintr-o bază de date se poate realiza conform următorului :

Principiu.

Operațiile de acces sunt reduse

- ***prin separarea atributelor aceluiași concept (entitate sau relație) ce sunt accesate de operații diferite***

respectiv

- ***prin combinarea atributelor aparținând unor concepte diferite ce sunt accesate de aceleași operații.***

Partiționarea entităților

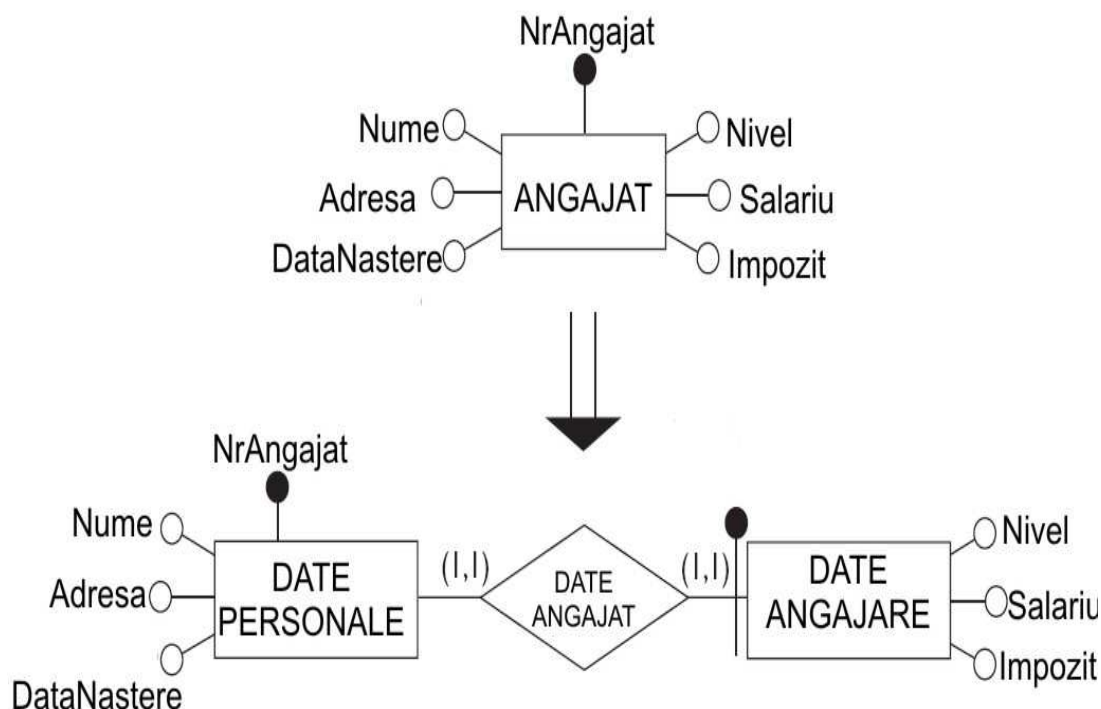


Figura 9. Exemplu de partiționare a entităților

- una din entități descrie informații despre statutul unui angajat
- cealaltă descrie informații personale ale unui angajat.

Este utilă doar dacă operațiile ce implică frecvent entitatea originală necesită, pentru un angajat, ori numai informații personale, ori numai informații legate de angajare.

partiționare verticală - un concept este divizat ținând cont de attributele sale.

Avantajul partiționării verticale este generarea unor noi entități cu număr mai mic de attribute decât entitatea originală \Rightarrow noile entități pot fi translate în structuri fizice din care se poate extrage un volum mare de date printr-un singur acces.

Partiționarea orizontală - divizarea se face în raport cu aparițiile entității.

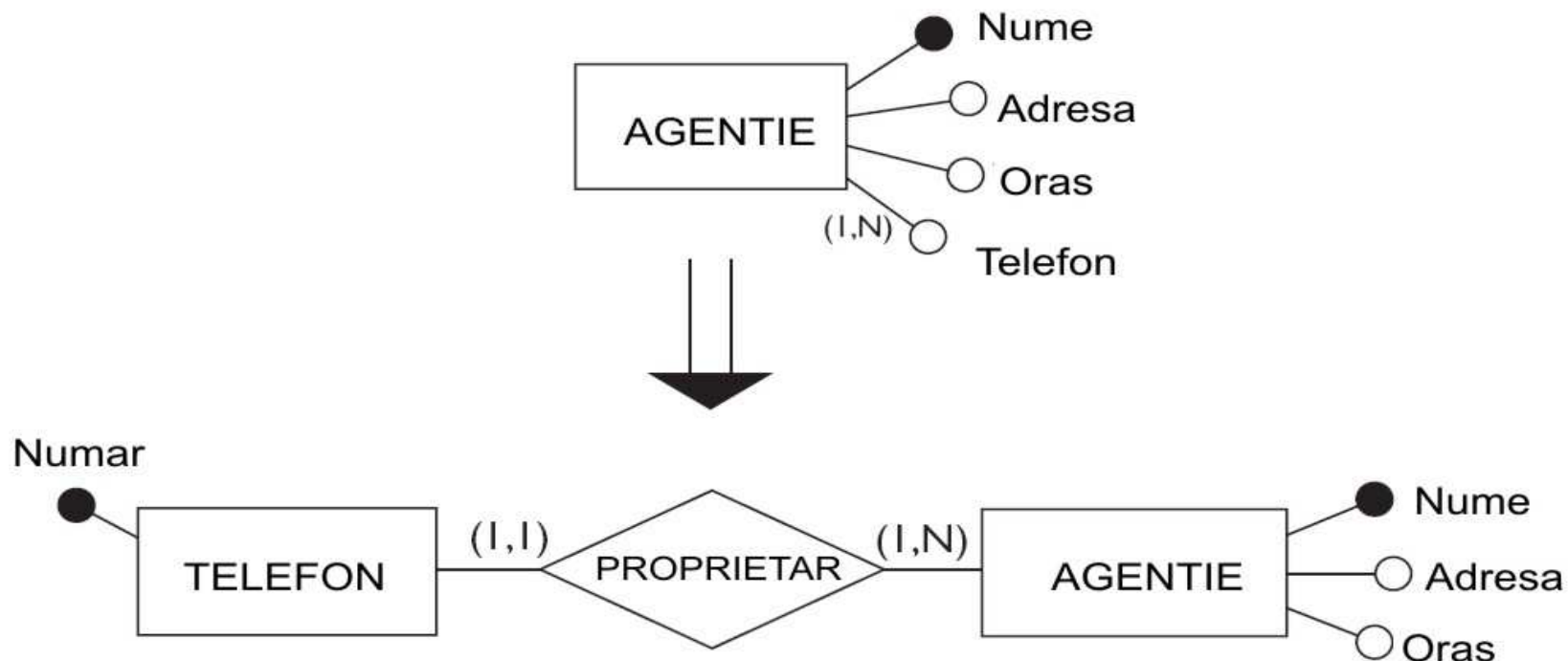
Exemplu:

- Pot exista operații asociate analiștilor și operații asociate angajaților din departamentul vânzări.
- În acest caz se dovedește utilă partiționarea entității ANGAJAT în două entități distincte, ANALIST și VÂNZARE, având aceleași atribute ca entitatea originală.

Partiționarea orizontală are ca efect secundar *introducerea de duplicate* pentru relațiile în care participă entitatea originală ⇒ impact negativ asupra performanțelor bazei de date.

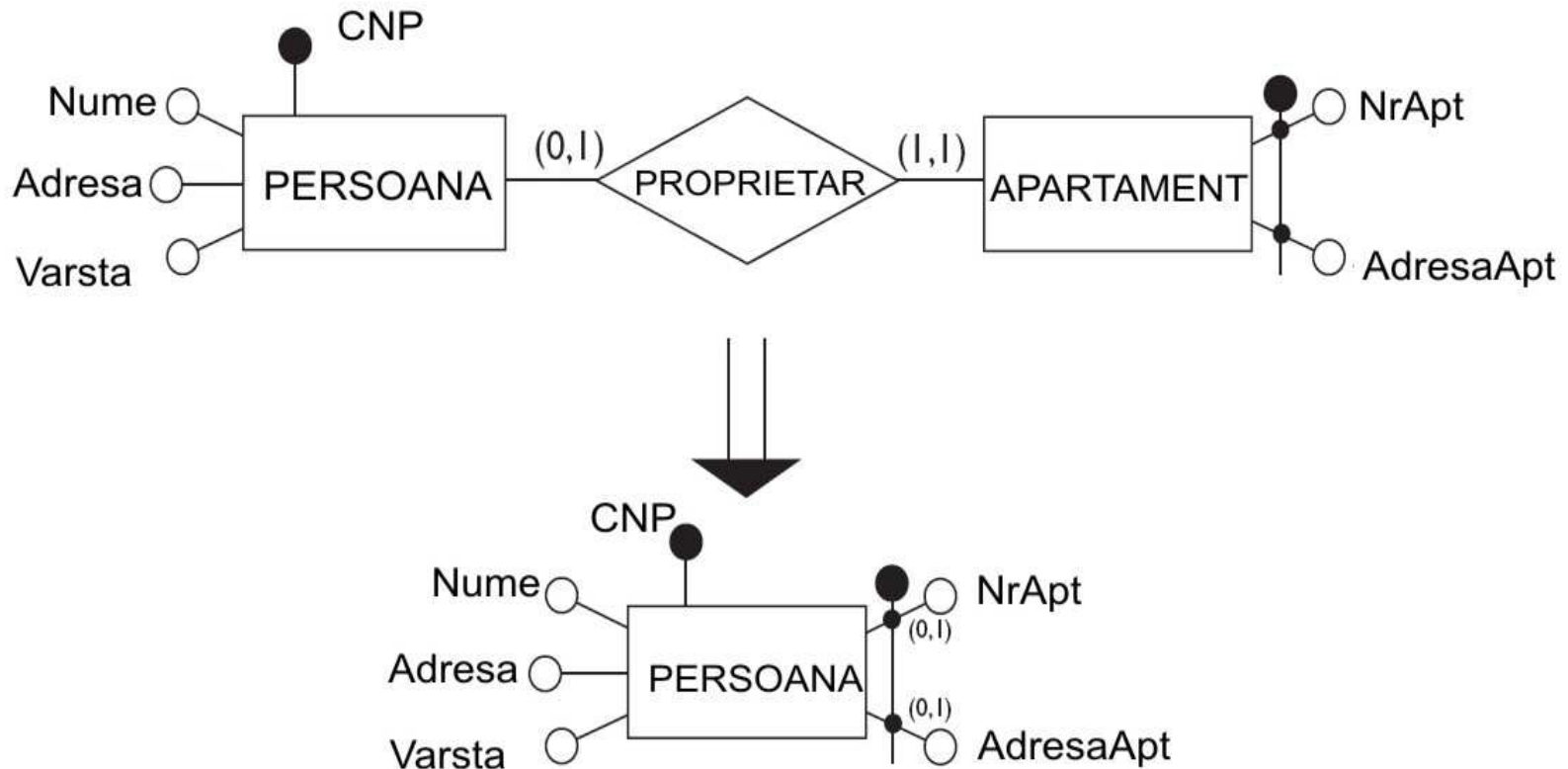
Ștergerea atributelor multi-valoare

Acest tip de restructurare este necesar deoarece modelul relațional nu permite reprezentarea directă a atributelor multi-valoare.



- Entitatea AGENTIE este separată în două entități: una având același nume și aceleași atribute cu entitatea originală, în afară de atributul multi-valoare (Telefon) și o entitate TELEFON cu atributul Numar.
- Entitățile sunt legate de o relație 1:N.
- Evident, dacă atributul este opțional, atunci cardinalitatea minimă pentru entitatea TELEFON trebuie să fie 0.

Combinarea entităților - operația inversă partiționării



Entitățile PERSOANA și APARTAMENT aflate în relația PROPRIETAR sunt combinate într-o singură entitate ce deține attributele ambelor entități.

Această restructurare este sugerată de faptul că majoritatea operațiilor frecvente asupra entității PERSOANA necesită informații în legătură cu apartamentul deținut de persoana respectivă \Rightarrow se dorește evitarea accesărilor necesare extragerii datelor prin intermediul relației PROPRIETAR.

Dezavantaj: posibilitatea apariției valorilor NULL în noua entitate PERSOANA deoarece cardinalitatea entității originale PERSOANA indică faptul că există persoane ce nu dețin un apartament.

Operația de combinare se efectuează în general asupra relațiilor 1:1 și mai rar asupra relațiilor 1:N sau N:N.

Motiv: apariția redundanțelor în atributele non-cheie ale entității având cardinalitatea N.

Alte tipuri de partiționare și combinare

Operațiile de partiționare și combinare pot fi aplicate și relațiilor din următoarele motive:

- pentru a separa aparițiile unei relații ce sunt întotdeauna accesate separat;
- pentru a uni două (sau mai multe) relații între aceleași entități într-o singură relație, atunci când aparițiile lor sunt accesate întotdeauna împreună.

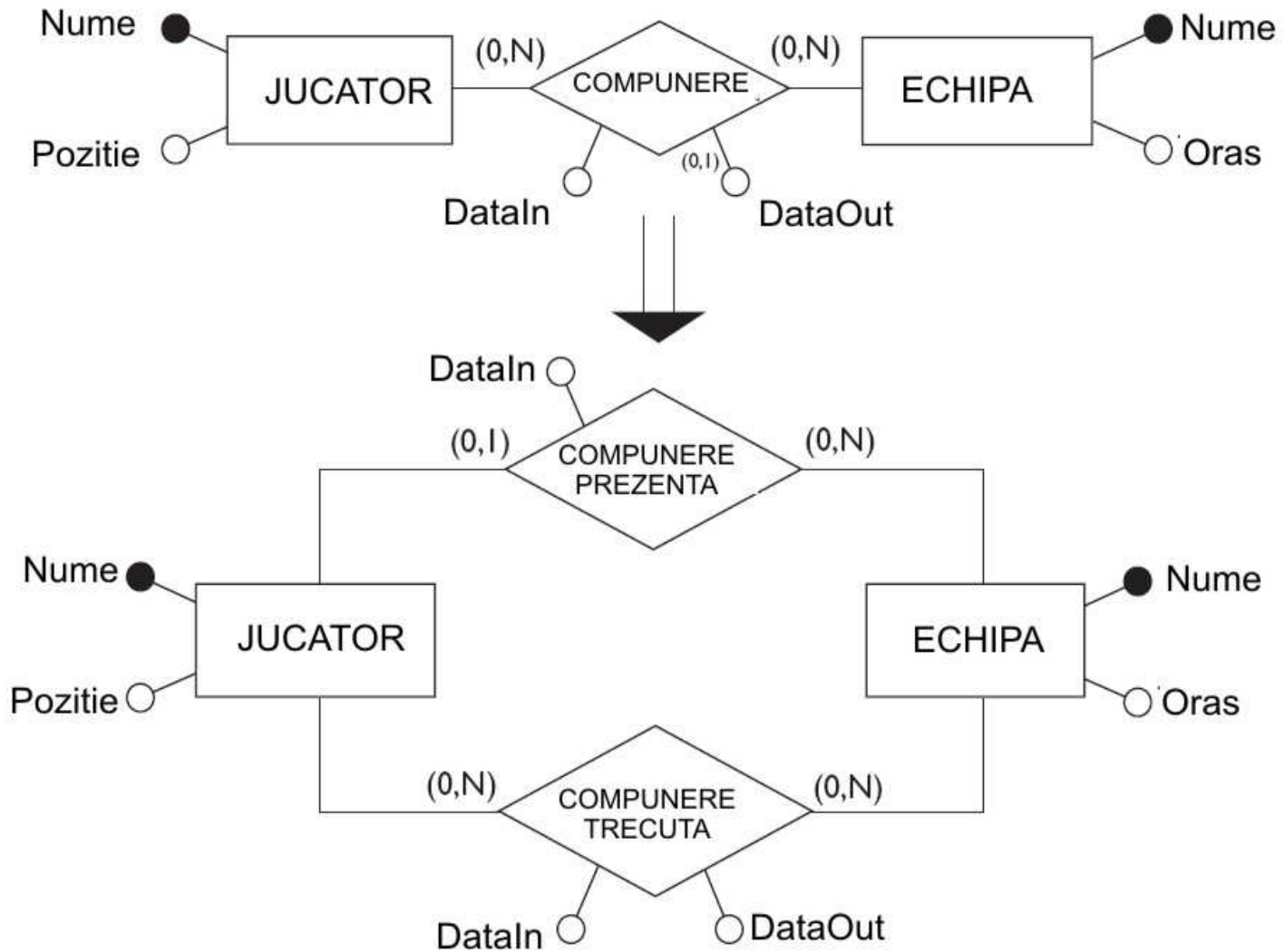


Figura 12. Exemplu de partiționare a unei relații

- se face distincție între jucătorii prezenți ai unei echipe de fotbal și cei din trecut

Selecția identificatorilor primari

În cazul în care există entități ce dețin mai mulți identificatori apare necesitatea stabilirii acelor atribute ce formează *identificatorul primar*.

Identificatorul primar va avea drept corespondent în modelul relațional o cheie primară.

Criteriile de alegere a unui identificator primar sunt:

- Atributele ce pot deține valori NULL nu pot forma un identificator primar.
- Este indicat ca un identificator primar să conțină un atribut sau cât mai puține atribute.

Avantaje:

- indecși de dimensiune redusă (un index este o structură auxiliară pentru acces rapid la date);
- spațiu redus pentru crearea legăturilor logice între relații;
- sunt facilitate operațiile de joncțiune.

- Un identificator intern este de preferat unuia extern care poate implica mai multe entități
 - Motiv: identificadorii externi sunt translați în chei ce conțin identificadorii tuturor entităților implicate în identificadorul extern.
- Este de preferat un identificator ce este utilizat de majoritatea operațiilor pentru a accesa aparițiile entității.
 - operațiile vor fi executate eficient, fiind avantajate de indecșii construiți automat de sistemul de gestiune a bazei de date.

Dacă în această etapă nu există nici un identificator care să satisfacă criteriile anterioare, **se va introduce un atribut suplimentar în entitate**, atribut ce va fi utilizat exclusiv pentru identificarea aparițiilor entității.

8.3 Translarea în modelul relațional

Translarea în modelul relațional este a doua etapă a proiectării logice.

Plecând de la o schemă E-R se construiește schema relațională echivalentă (echivalentă din punct de vedere al capacității de a reprezenta informații echivalente).

Translarea va pleca de la o schemă E-R simplificată, ce nu conține generalizări sau attribute multi-valoare și are numai identificatori primari.

Entități și relații N:N

Translarea în modelul relațional se face urmând pașii:

- Fiecare entitate va fi translată într-o relație M-R (a modelului relațional) cu același nume și având aceleași attribute ca și entitatea;
 - cheia fiecărei relații este dată de identificatorul entității corespunzătoare.
- Fiecare relație E-R dintre entități va avea drept corespondent o relație M-R cu același nume și cu attribute attributele relației E-R și identificatorii entităților implicate;
 - cheia relației M-R va conține identificatorii entităților.
- Dacă entitățile sau relațiile E-R au attribute opționale, attributele corespunzătoare din modelul relațional pot avea valoarea NULL.

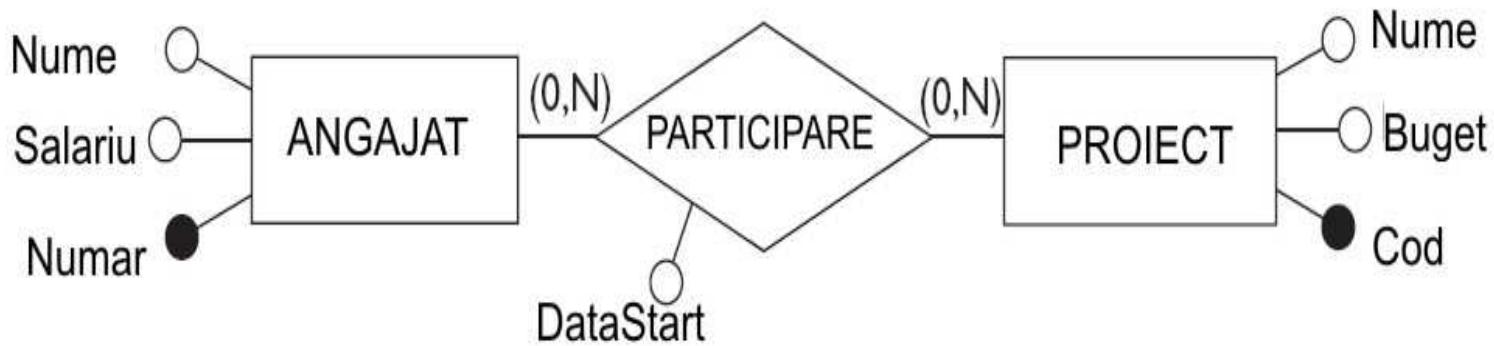


Figura 13. Schemă E-R cu o relație N:N

ANGAJAT(Numar, Nume, Salariu)

PROIECT(Cod, Nume, Buget)

PARTICIPARE(Numar, Cod, DataStart)

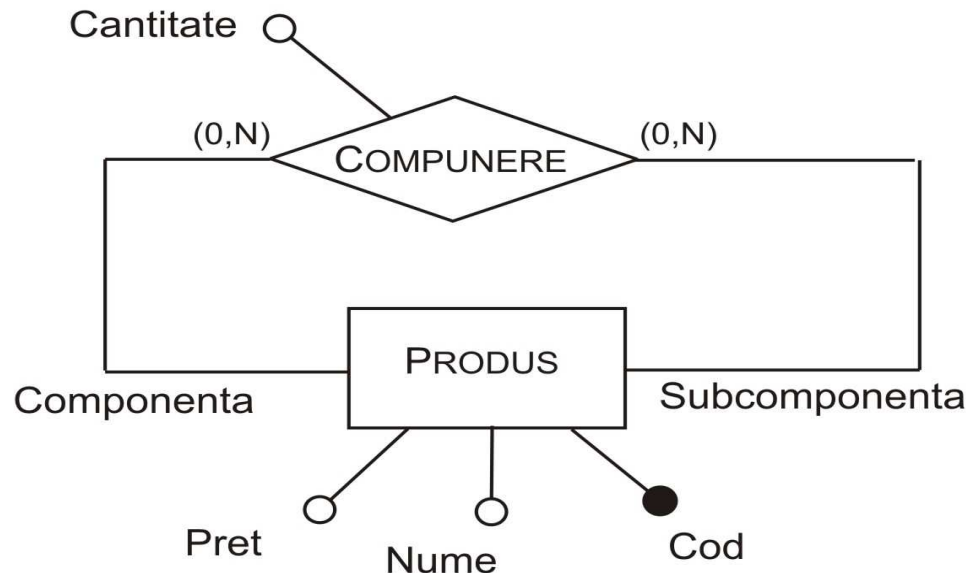
În cazul în care se dorește mărirea clarității schemei obținute se pot opera redenumiri de atribute:

PARTICIPARE(Angajat, Proiect, DataStart)

Domeniul atributului *Angajat* este o mulțime de numere de înregistrare pentru angajați, iar domeniul pentru atributul *Proiect* este o mulțime de coduri de proiecte.

Trebuie impuse constrângeri de referință între aceste atribute și relațiile ANGAJAT, respectiv PROIECT.

Operația de redenumire este esențială în cazul relațiilor recursive.



Prin translare se obține:

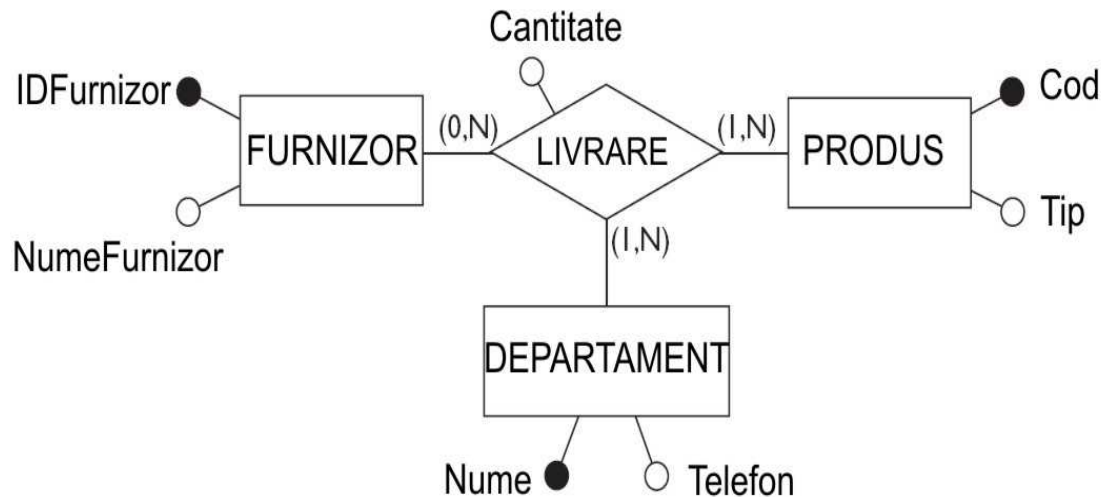
PRODUS(Cod, Nume, Pret)

COMPUNERE(Componenta, Subcomponenta, Cantitate)

Atributele Componenta și Subcomponenta au ca domeniu mulțimea de coduri ale produselor.

Trebuie impuse constrângeri de referință între Componenta și PRODUS și Subcomponentă și PRODUS.

Translarea relațiilor E-R ce implică mai mult de două entități



FURNIZOR(IDFurnizor, NumeFurnizor)
PRODUS(Cod, Tip)
DEPARTAMENT(Nume, Telefon)
LIVRARE(Furnizor, Produs,
Departament, Cantitate)

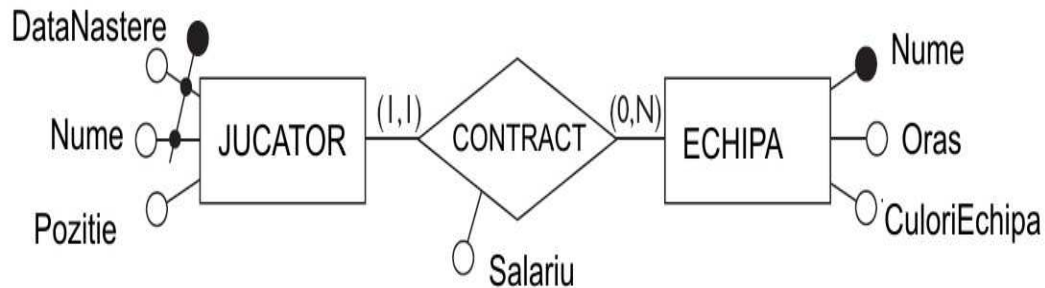
- Se verifică dacă identificatorii entităților, luați împreună, nu formează o cheie sau o super-cheie redundantă pentru relația M-R care reprezintă relația E-R.

Aceasta se poate întâmpla, spre exemplu, dacă există un singur furnizor care livrează un produs dat unui departament dat.

- Rămâne validă cardinalitatea deoarece acest furnizor poate livra mai multe produse acestui departament sau altor departamente.

În acest caz, cheia relației M-R LIVRARE trebuie să fie formată doar din atributele *Produs* și *Departament*, deoarece, fiind dat un produs și un departament, furnizorul este determinat în mod unic.

Relații 1:N



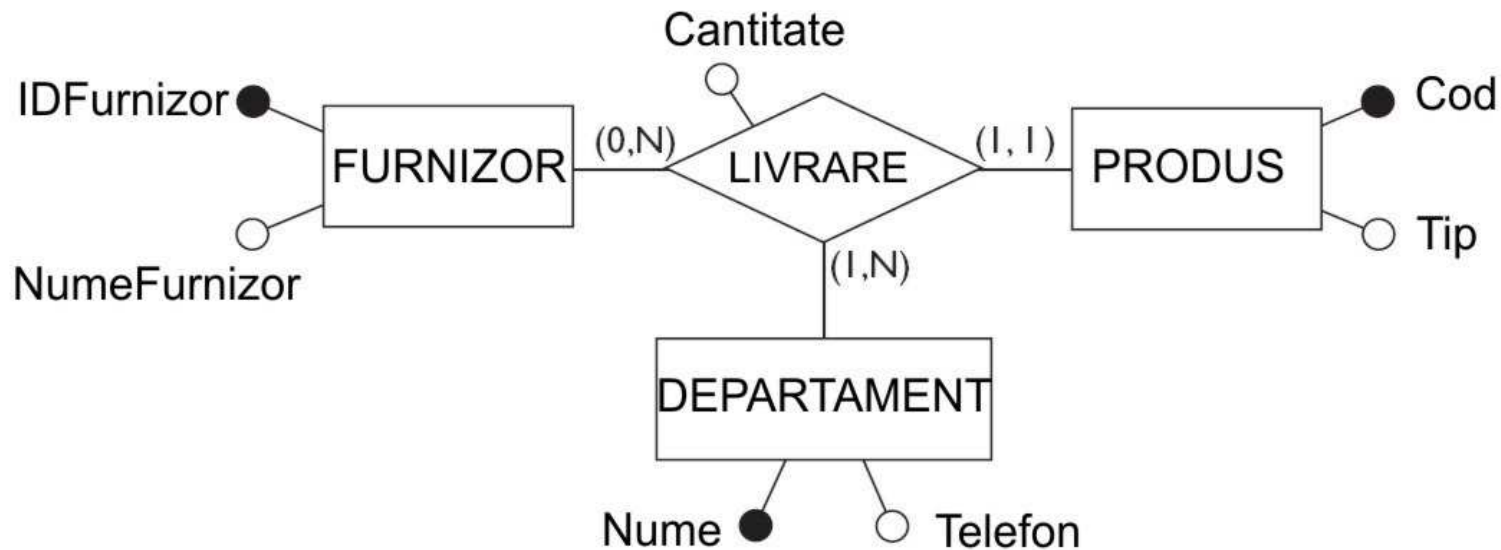
JUCATOR(Nume, DataNastere, Pozitie)
ECHIPA(Nume, Oras, CuloriEchipa)
CONTRACT(NumeJucator,
DataNastereJucator, Echipa, Salariu)

Cheia relației CONTRACT este formată numai din identificatorul entității JUCATOR deoarece cardinalitățile indică faptul că fiecare jucător are contract la o singură echipă.

Observații

- Relațiile JUCATOR și CONTRACT au aceeași cheie \Rightarrow există posibilitatea combinării celor două relații în una singură, fără să existe pericolul introducerii de redundanțe.
- Combinarea este posibilă datorită corespondenței 1:1 între instanțele celor două relații:
JUCATOR(Nume, DataNastere, Pozitie, Echipa, Salariu)
ECHIPA(Nume, Oras, CuloriEchipa)
Trebuie impusă o constrângere de referință între atributul `Echipa` și relația ECHIPA.
- Presupunem că entitatea JUCATOR participă opțional la relația CONTRACT \Rightarrow ambele translări prezentate anterior sunt valide. Ce-a de-a doua prezintă dezavantajul că pot exista valori NULL în relația JUCATOR pentru attributele `Echipa` și `Salariu`.

Regulă: entitatea ce participă la o relație E-R cu cardinalitatea maximă 1 este translată într-o relație M-R ce include identificatorii celorlalte entități implicate în relația E-R și posibilele attribute ale relației E-R \Rightarrow nu mai este nevoie de o relație M-R separată pentru relația din schema E-R.



Presupunem că entitatea PRODUS participă în relația LIVRARE cu cardinalitățile minimă și maximă 1 \Rightarrow pentru fiecare produs există un singur furnizor care livrează produsul respectiv și un singur departament deservit.

Schema este translatată sub forma:

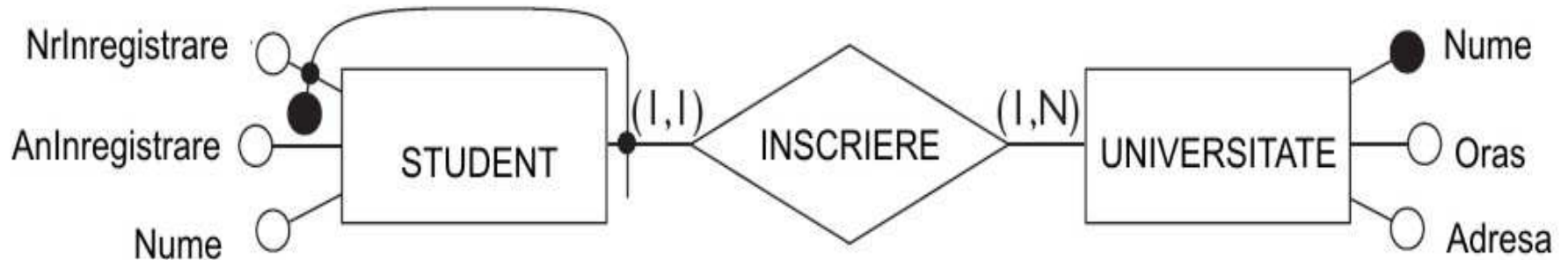
FURNIZOR(IDFurnizor, NumeFurnizor)

DEPARTAMENT(Nume, Telefon)

PRODUS(Cod, Tip, Furnizor, Departament, Cantitate)

Se impun constrângeri de referință între atributul `Furnizor` al relației PRODUS și relația FURNIZOR și între atributul `Departament` al relației PRODUS și relația DEPARTAMENT.

Entități cu identificatori externi



Schema relațională este:

STUDENT(Nrlnregistrare, Universitate, Nume, Anlnregistrare)
UNIVERSITATE(Nume, Oras, Adresa)

Există o constrângere de referință între atributul `Universitate` și relația **UNIVERSITATE**.

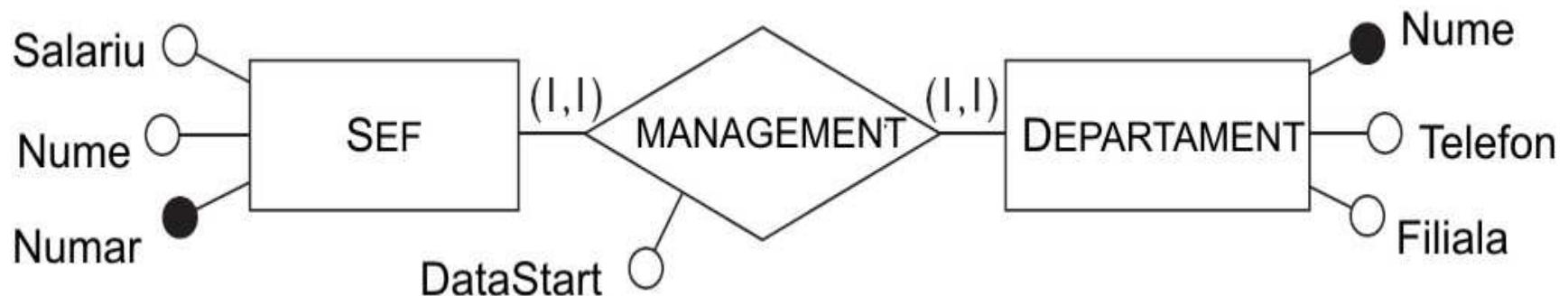
Este de notat faptul că, prin reprezentarea identificatorului extern, a fost reprezentată și relația dintre entități.

Acest tip de translare este valid indiferent de cardinalitatea cu care participă celelalte entități la relație.

Relații 1:1

Pentru relațiile 1:1 există mai multe posibilități de translare.

Să considerăm relația din figura următoare, cu participare obligatorie din partea celor două entități implicate.



Există două posibilități valide de translare:

SEF(Numar, Nume, Salariu, Departament, DataStart)

DEPARTAMENT(Nume, Telefon, Filiala)

cu constrângere de referință între atributul `Departament` al relației SEF și relația DEPARTAMENT, sau

SEF(Numar, Nume, Salariu)

DEPARTAMENT(Nume, Telefon, Filiala, Sef, DataStart)

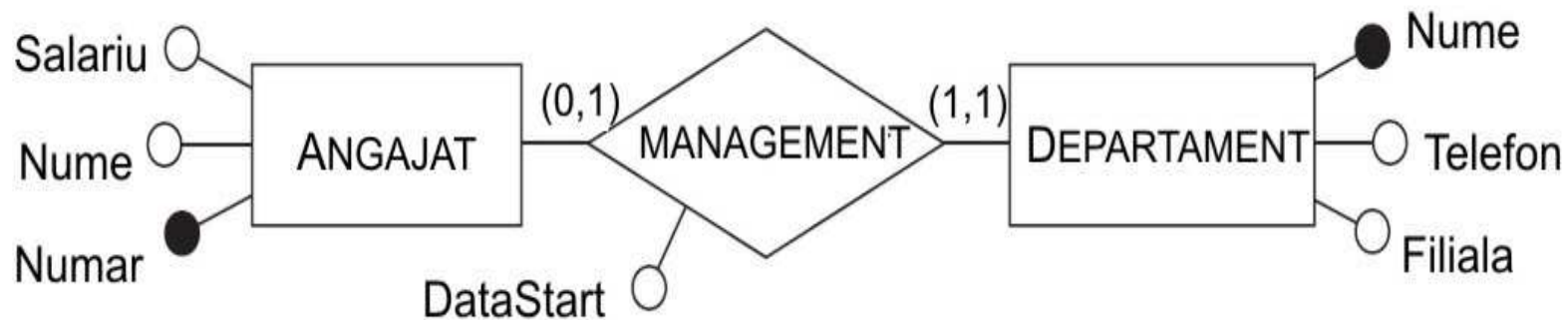
cu constrângere de referință între atributul `Sef` al relației DEPARTAMENT și relația SEF.

Pe lângă aceste două soluții, există posibilitatea obținerii unei singure relații, incluzând toate atributele din schema E-R.

Această soluție va fi eliminată.

Motiv: dacă după faza de restructurare (care precede faza de translare) schema E-R conține două entități legate printr-o relație 1:1, este de dorit ca în faza de translare cele două concepte să rămână separate.

Să considerăm cazul unei relații 1:1 cu **participarea opțională a uneia dintre entități**, ca în schema următoare:



Soluția este:

ANGAJAT(Numar, Nume, Salariu)

DEPARTAMENT(Nume, Telefon, Filiala, Sef, DataStart)

cu constrângere de referință între atributul *Sef* al relației DEPARTAMENT și relația ANGAJAT.

Această opțiune este preferabilă uneia în care relația E-R este reprezentată în relația M-R ANGAJAT prin numele departamentului manageriat, deoarece, pentru acest atribut, ar putea exista valori NULL.

Să considerăm cazul în care **ambele entități au participări opționale**.

Să presupunem că în schema din figura anterioară pot exista departamente fără șef (cardinalitatea minimă a entității DEPARTMENT este 0).

În acest caz se obține prin translare:

ANGAJAT(Numar, Nume, Salariu)

DEPARTMENT(Nume, Telefon, Filiala)

MANAGEMENT(Sef, Departament, DataStart)

Se observă că se poate lua drept cheie a relației MANAGEMENT atributul Departament.

Se impun constrângeri de referință între attributele Sef și Departament ale relației MANAGEMENT și relațiile ANGAJAT și DEPARTMENT.

Avantaj - attributele ce implementează relația din schema E-R nu pot avea valori NULL.

Dezavantaj - relația nouă introdusă.

Această soluție este recomandată în cazul în care numărul de apariții ale relației este mic în comparație cu numărul de apariții ale entităților implicate în relație.