

Extinderea modelului relațional

Lect.dr. Mioara UDRICĂ
Universitatea Spiru Haret, București

Aplicațiile privind gestiunea unei unități economice trebuie să răspundă necesităților de informație pentru luarea deciziilor imediate, dar și unor decizii pe termen lung, care presupun corelarea informațiilor din diferite domenii, manipularea datelor netradiționale. Modelului relațional, care utilizează structuri simple, construite pornind de la tipuri atomice (întreg, real, șir de caractere), nu mai poate răspunde complexității problemelor cerute de contextul actual. S-au dezvoltat astfel două tendințe, vizând fie îmbunătățirea modelelor clasice existente, fie definirea unor noi modele de date.

Cuvinte cheie: agregare, generalizare, moștenire

Prezentarea bazelor de date astfel încât să suporte noțiunile abstracte de generalizare și agregare, constituie o încercare de extindere a bazelor de date relaționale. Pentru a reduce complexitatea unor aplicații, modelul lumii reale este descompus într-o ierarhie de abstracții, o combinație de agregări și generalizări. Astfel, o entitate care aparține unei ierarhii rezultate din generalizare poate aparține în același timp unei ierarhii obținute prin agregare, fără a avea importanță pentru o entitate dacă este rădăcină sau este nod intermediar. Ierarhia abstracțiilor permite utilizatorului să acceseze modelul la diferite nive-

luri de detaliu, altfel spus, la diferite niveluri de abstractizare.

Agregarea este o formă de abstracție unde entități de diferite tipuri sunt combinate pentru a forma o alta, de grad înalt de abstractizare. Modelul relațional furnizează suport pentru modelarea agregării. Relațiile între entități sunt considerate și modelate ca entități de sine stătătoare, agregarea corespunzând noțiunii matematice de produs cartezian. De exemplu, relația *asamblează*, între *Personal*, care supraveghează *Imobilizări* pentru obținerea de *Produce* cu ajutorul unor *Materiale* se reprezintă ca în figura 1.

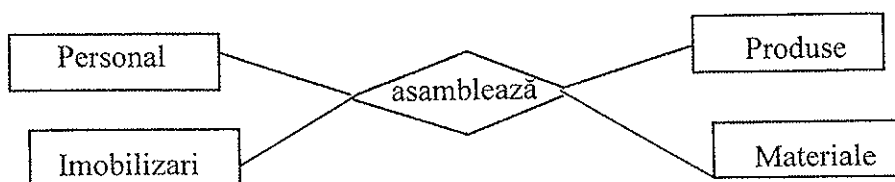


Fig. 1.

Ea poate fi modelată (figura 2) ca o entitate agregată numită *Asamblare*, cu atributele cost și timp, care participă la relație alături de celelalte entități. Un caz particular de agregare constituie relația "parte a", relație existentă între o entitate și părțile ei componente, care pot fi

entități de diferite tipuri. În modelarea lumii reale, se pot forma ierarhii de agregări.

Generalizarea este o abstracție care permite unei clase de entități individuale să fie gândită generic ca o singură entitate. Este folosită pentru a grupa

împreună entități similare care pot fi descrise de un tip generic de entitate, care

omite detaliile nerelevante ale entităților individuale.

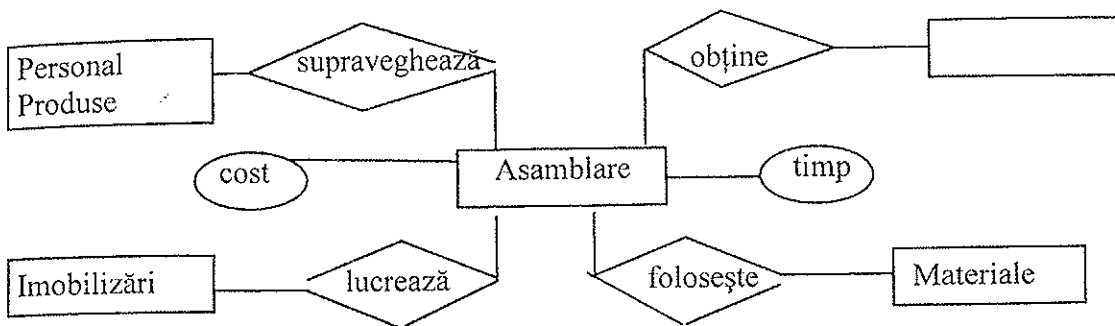


Fig. 2.

Într-o generalizare, colecția de entități individuale $E^1_{ind}, \dots, E^n_{ind}$ cu condiția $E^i_{ind} \cap E^j_{ind} = \emptyset$ pentru $1 \leq i, j \leq n, i \neq j$

este generic descrisă de o entitate $E_{generic}$. Fiecare entitate E^i_{ind} poate fi văzută ca fiind o entitate de tip $E_{generic}$.

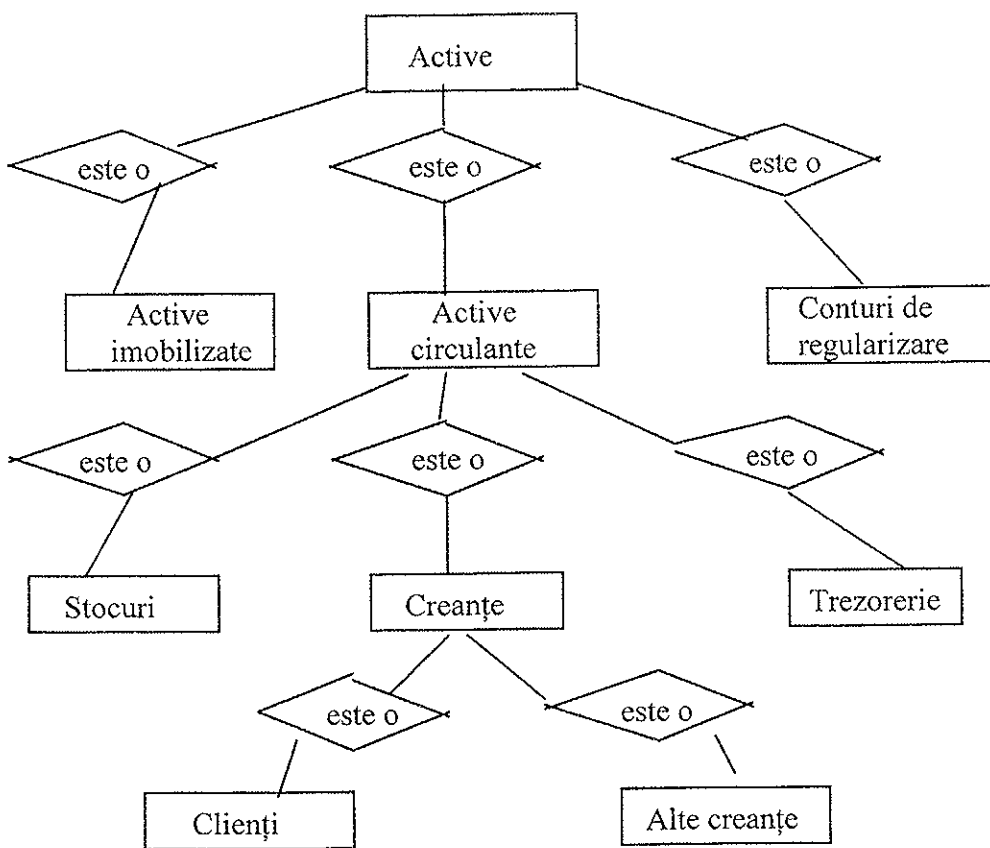


Fig. 3.

ților de g, care datelor pornind flexității șind fie

diferite ție un-combi- id înalt ial fur- i agre- onside- e stătă- oțiunii De e- Perso- ri pen- il unor t 1.

elarea nii de care iduale i enti- grupa

Generalizarea pe mai multe niveluri între entități similare legate printr-o relație de tipul "este o", dă naștere la o ierarhie de generalizări. În figura 3 se prezintă un exemplu de rezolvare cu ajutorul generalizării a unei situații des întâlnite în activitatea de analiză a situației patrimoniale, de interpretare a documentelor de sinteză.

Moștenirea este o parte integrantă a generalizării, în sensul că tipurile de

entități specializate moștenesc proprietățile tipurilor care le generalizează sau, privind în sens opus, proprietățile comune ale celor mai specializate entități sunt factorizate și asociate tipului de entitate general. Astfel, entitatea $E_{generic}$ cuprinde toate proprietățile caracteristice ale entităților individuale relevante pentru nivelul de abstractizare E_{ind}^i $1 < i < n$, fapt care poate fi formalizat astfel:

$$\text{Proprietățile}(E_{generic}) := \bigcap \text{Proprietățile}(E_{ind}) \quad E_{ind} \in \{E_{ind}^1, \dots, E_{ind}^n\}$$

Tipul E_{ind}^i , cu $1 < i < n$ se numește *specializare* a tipului generic $E_{generic}$. Schemele entităților E_{ind}^i descriu acele proprietăți care sunt specifice entităților individuale și, deci, nu sunt acoperite de descrierea entității generice. Generalizarea și specializarea nu sunt acceptate de modelele relaționale pure. Există însă încercări de extindere a lor, pentru a accepta aceste concepte.

Se consideră un tip de entitate generică R care este specializat în tipurile R_{11}, \dots, R_{mpm} . Cele m dimensiuni de specializare sunt s_{k1}, \dots, s_{km} . Ele formează o

submulțime a tuturor proprietăților s_1, \dots, s_n care descriu entitățile tipului R și fiecare specifică o proprietate a entităților de tip R pe baza căreia se pot grupa în submulțimi specializate R_{11}, \dots, R_{ip} . Mulțimile R_{11}, \dots, R_{ip} sunt mutual disjuncte și fiecare entitate de tip R aparține numai unei mulțimi R_{11}, \dots, R_{ip} , în funcție de proprietățile sale s_{ki} . Astfel, fiecare entitate de tipul R este specializată în m diferite dimensiuni și deci aparține la m diferite tipuri de entități specializate, așa cum arată figura 4.

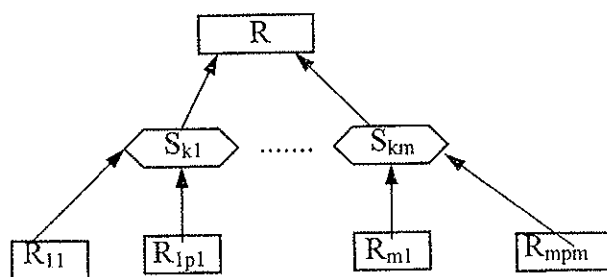


Fig. 4.

Pentru modelarea ierarhiilor abstracte din exemplul precedent în literatura de specialitate se folosește următoarea sintaxă:

```
var R [ : generic
    s_k1 = ( R_11, ..., R_1p1);
    ...
```

```
s_km = (R_m1, ..., R_mpm);
of ]
aggregate [keylist]
s_1:[key] R_1;
.....
s_n:[key] R_n;
end
```

Elementele care descriu schema au în plus următoarele semnificații:

- tuplurile de tip R constituie agregări reprezentate de atributele s_1, \dots, s_n . Fiecare din aceste atribute constituie referiri la alte tupluri;
- R_i ($1 \leq i \leq n$) este fie identificator generic (cheia trebuie să apară în definiție), fie un identificator de tip (cheia nu trebuie să apară);
- lista cheilor este o submulțime a mulțimii $\{s_1, \dots, s_n\}$;
- fiecare R_{ij} ($1 \leq i \leq m$, $1 \leq j \leq p_i$) este un identificator generic cu domeniile cheilor aceleași ca pentru R ;
- fiecare s_{ki} ($1 \leq i \leq m$) este același ca oricare s_j ($1 \leq j \leq n$);
- dacă s_{ki} este același cu s_j atunci cheia de tip R_j este șirul (R_{i1}, \dots, R_{ipi})

• un atribut s_i este fie o referință spre alt obiect în care caz cheia trebuie să apară, fie o valoare atomică de tip R_i , în care caz R_i este un identificator de tip. Fiecare atribut s_{ki} ($1 \leq i \leq m$) constituie o partiție disjunctă de obiecte aparținând lui R în clase de obiecte R_{i1}, \dots, R_{ip} , unde R_{ij} însuși reprezintă o definiție de tip. Exemplu (figura 5): Pentru a înregistra în contabilitate obținerea unui produs, trebuie să evidențiem pentru fiecare componentă informații despre manoperă și materialele. Modelul următor folosește agregarea pentru a-i descrie componentele particulare și generalizarea pentru a prezenta modul în care este construit.

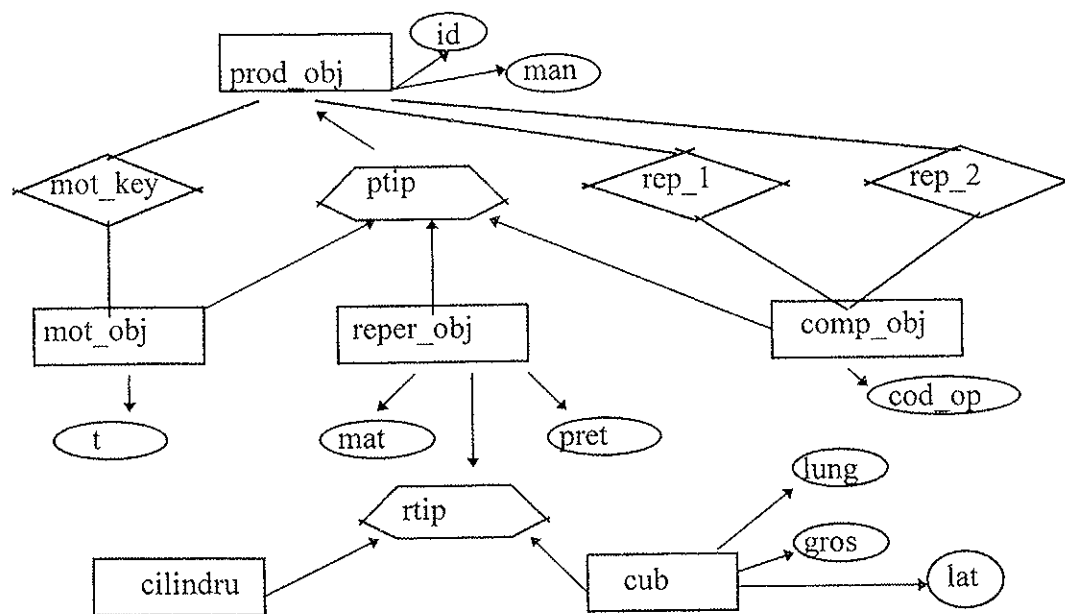


Fig. 5.

```

var prod_obj: generic
PTIP = (reper_obj, mot_obj, comp_obj);
of
aggregate [ID]
  ID: identificator;
  MAN: manopera;

```

```

PTIP: str_comp;
end
Reper_obj este specializare a tipului
prod_obj. Urmând regulile moștenirii,
atributele, cu excepția celui de speciali-
zare, sunt duplicate.

```

```

var reper_obj: generic
RTIP = (cub, paralelipiped, cilindru...);
  of
  aggregate [ID]
  ID: identificator ;
  MAN: manopera;
  MAT: material;
  PRET: real;
  RTIP: geom_form;
end
Tipul cub nu este generic, fiind constituit
din agregarea mai multor atribute.
var cub
aggregate [ID]
  ID: identificator ;
  MAN: manopera;
  MAT: material;
  PRET: real;
  LUNG: real ;
  LAT: real ;
  GROS: real ;
end
Mot_obj conține un atribut mot_key care
asigură referirea unui tuplu din relația

```

```

prod_obj, și atributul propriu T, unde
sunt înscrise caracteristicile motorului,
sub forma unei matrice. Mot_key are
ace-leași proprietăți ca "foreign key" de
la modelele relaționale.
var mot_obj
aggregate [ID]
  ID: identificator;
  MAN: manopera;
  MOT_KEY: key prod_obj;
  T : matrice ;
end
Comp_obj conține două atribute pentru
referirea tuplului prod_obj.
var comp_obj
aggregate [ID]
  ID: identificator ;
  MAN: manopera ;
  REP_1: key prod_obj ;
  REP_2: key prod_obj ;
  COD_OP: (strunjire, taiere....)
end.
Extensia schemei relaționale este pre-
zentată în figura 6.

```

PROD OBJ		
ID	MAN	PTIP
subans-1	atelier-3	comp_obj
reper-1	atelier-1	reper_obj
reper-2	terti	reper_obj
.....

COMP OBJ				
ID	MAN	REP 1	REP 2	COD OP
.....
subans-1	atelier-3	reper-1	reper-2	imbinare
.....

REPER OBJ				
ID	MAN	MAT	PRET	RTIP
.....
reper-1	atelier-1	fier	130.50	cub
reper-2	terti	aluminu	50	cub
.....

de
ui,
are
de

CUB						
ID	MAN	MAT	PRET	LUNG	LAT	GROS
....
reper-1	atelier-1	fier	130.50	10	5	7
reper-2	terti	aluminium	50	20	12	3
.....

Fig. 6.

Prod_obj este de tip generic, împărțit în submultimi disjuncte de atributul *PTIP*, și obținut din mai multe agregări. Descrierea pentru *subans-1* în *prod_obj* reprezintă cel mai înalt nivel de abstractizare. Detalierea descrierii se face în relația *comp_obj*, unde informațiile sunt duplicate, și deci se definesc două attribute noi, care identifică în *prod_obj* două tupluri. *Cub_obj* reprezintă cea mai specializată descriere.

Deși fiecare nivel de abstracție moștenește attributele nivelului superior, mărindu-se astfel redundanța, teoretic modelul definește variabile generice obți-

nute prin agregarea unor attribute și oferă o soluție pentru situații în care attributele sunt ele însele relații.

Bibliografie

* Dupuy Y. Le comptable, la comptabilite et la conception des systemes d'information. Revue francaise de comptabilite nr.215/1990.

* Huges J. D. Object-oriented databases management. (Application in Engineering and Computer Science), Prentice Hall, 1994.

tru

re-