

Studiu comparativ între modelul de date orientat spre obiecte si modelul de date relational

Asist. Andy STEFANESCU,
Facultatea de Stiinte Economice, Universitatea din Craiova
Asist. Laura STEFANESCU,
Facultatea de Contabilitate - Finante Craiova, Universitatea „Spiru Haret”

It is necessary that data design be achieved properly so that a system offering correct information could be designed, which determines the success and efficiency of the decisional process.

The work aims to analyze to concept of data basis orientated on object and by comparing the relational data basis, to stress out both the drawbacks of the relational data pattern and the advantages the objects design show.

Keywords: objects, data design, connection among objects, referential integrity.

Bazele de date orientate pe obiecte beneficiaza de un model de date orientat pe obiecte ce are la baza notiunea de entitate conceptuala, definita ca un obiect descris printr-o colectie de proprietati. Conceptul de obiect este simplu dar, în acelasi timp, foarte puternic: fiecare obiect poate fi definit si în-tretinut independent de celelalte. Aceasta definitie a unui obiect este foarte asemanatoare definitiei unei entitati. Totusi, un obiect înglobeaza atât starea, cât si comportamentul, în timp ce o entitate modeleaza doar starea. Starea curenta a unui obiect este descrisa de unul sau mai multe **atribute** sau **variabile de instanta**. De exemplu: Agentia de Brokeraj din Pitesti poate avea atributele prezentate în tabelul 1. Atributele pot fi clasificate ca simple sau complexe. Un **atribut simplu** poate sa fie un tip simplu de date (întreg, sir, real) care ia valori literare sau numerice. De exemplu Cod_ Agentie din tabelul 1 este un atribut simplu care cu valoarea literala ABK5. Un **atribut complex** poate contine colectii si/sau referinte. De exemplu, atributul Brokeri_Angajati reprezinta o colectie de obiecte de tip Broker. Un **atribut de referinta** reprezinta o relatie dintre obiecte si contine o valoare sau o colectie de valori, care sunt ele însele obiecte. De exemplu, Brokeri_Angajati este, mai exact o colectie de referinte la obiectele de tip Broker. Din punct de vedere conceptual, un atribut referinta este

similar cheii straine din modelul de date relational.

Tabelul 1. Atributele obiectului pentru instanta Agentii

Cod_Agentie	ABK5
Denumire	Agentia de Brokeraj PitInvest
Localitate	Pitesti
Strada	AI Cuza
Telefon	O48123988
Brokeri_Angajati	Ionele Ion; Popa Paula
Ordine_tranzactionate	Ordin de vânzare; ordin de cumparare
Manager	Dobrin Mihai

Metodologia modelarii logice a bazelor de date se bazeaza pe modelul Entitate-Relatie Extins (EER) si are puncte similare cu modelul de date orientat spre obiecte, asa cum s-a sintetizat în tabelul 2. Abordarea EER singura este insuficienta pentru a desavârsi proiectarea unei baze de date orientate spre obiecte. Abordarea EER trebuie suplinita cu o tehnica care identifica si documenteaza comportamentul fiecărei clase de obiecte. Aceasta implica o analiza detailata a cerintelor de prelucrare. Într-un flux de date conventional, prin utilizarea diagramelor de flux de date (DFD), cerintele de prelucrare ale sistemului sunt analizate separat fata de modelul de date. În analiza orientata spre obiecte cerintele de prelucrare sunt transpuse într-un set de metode, care sunt unice pentru fiecare clasa.

Tabelul 2 Comparatie între modelul relational si modelul obiectelor

Modelul obiectelor	Modelul relational	Diferente
Obiect	Entitate	Obiectul include comportamentul
Atribute	Atribute	Nu
Relatii	Relatii	Asocierile sunt aceleasi, dar în modelarea obiectuala mostenirea include atât starea, cât si comportamentul
Mesaje		Conceptul nu are corespondent pentru modelul de date relational
Clase	Tipuri de entitati	Nu
Instance	Entitati	Nu
Încapsulare		Conceptul nu are corespondent pentru modelul de date relational

Similitudinea dintre cele doua abordari face ca metodologia modelarii logice de date, sa prezinte o baza rezonabila pentru o metodologie de proiectare a bazelor de date orientate spre obiecte. Cu toate ca aceasta metodologie este orientata în principal spre proiectarea bazelor de date relationale, totusi, modelul poate fi transformat relativ simplu pentru modelele în retea si ierarhice. În modelul de date obtinut, relatiile de tip multi-la-multi si relatiile recursive sunt îndepartate. Aceste schimbari nu sunt necesare pentru modelarea orientata spre obiecte si pot fi omise. Modificarile au fost introduse datorita puterii de modelare limitate a modelelor de date traditionale. Utilizarea normalizarii în cadrul metodologiei este înca importanta si nu trebuie omisa în proiectarea bazelor de date orientate spre obiecte. Normalizarea este utilizata pentru a îmbunatatii modelul, în asa fel încât el sa satisfaca diverse constrângeri care evita dublarea inutila a datelor. Faptul ca se creeaza cu obiecte nu înseamna ca redundanta este acceptata. În termenii orientarii spre obiecte, a doua si a treia forma normala trebuie interpretate în felul urmator: „Fiecare atribut dintr-un obiect este dependent de identitatea obiectului.”

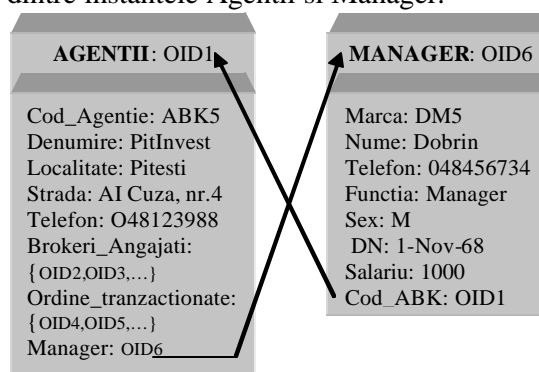
Proiectarea bazelor de date orientate spre obiecte necesita ca schema bazei de date sa includa atât descrierea structurii de date a obiectului si constrângerile, cât si comportamentul obiectului.

Relatiile dintre obiecte si integritatea referentiala

Într-un model de date de obiecte, relatiile sunt reprezentate utilizând atributele de referinta implementate în mod obisnuit prin identificatorii de obiecte (OID). În continuare se va analiza modul de reprezentare a relatiilor dupa cardinalitatea lor: unu-la-unu (1:1), unu-la-multi (1:M) si multi-la-multi (M:M).

Relatii de tip 1:1

O relatie de tip 1:1 dintre obiectele A si B este reprezentata prin adaugarea la obiectul A a unui atribut de referinta si, pentru a mentine integritatea referentiala, adaugarea la obiectul B a unui alt atribut de referinta. De exemplu, în figura 1 este reprezentata relatia de tip 1:1 dintre instantele Agentii si Manager.

**Fig.1.** Relatia de tip 1:1

Relatii de tip 1:M

O relatie de tip 1:M dintre obiectele A si B este reprezentata prin adaugarea la obiectul B a unui atribut de referinta si la obiectul A a unui atribut continând un set de referinte. De exemplu, în figura 2 sunt reprezentate doua relatii de tip 1:M, una dintre Agentii si Bro-

keri_Angajati si alta dintre Brokeri_Angajati si Ordine_Tranzactionate.

Relatii de tip M:M

O relatie de tip M:M dintre obiectele Asi B este reprezentata prin adaugarea la fiecare obiect a unui atribut care contine un set de referinte. De exemplu, în figura 3 este reprezentata relatia de tip M:M dintre entitatile Client si Brokeri_Angajati. În proiectarea bazelor de date relationale, se descompune relatia de tip M:M în doua relatii de tip 1:M, conectate printr-o entitate intermediara. Pornind de la acest model relational, este posibil sa se reprezinte într-un Sistem de Gestiune al Bazelor de Date Orientat Obiect, asa cum este aratat în figura 4.

Integritatea referentiala

Integritatea referentiala pentru modelul relational se realizeaza prin intermediul cheilor

primare si straine. O cheie straina leaga fiecare aparitie din relatia copil de aparitia din relatia parinte, care contine valoarea corespunzatoare cheii candidat. Integritatea referentiala semnifica faptul ca, în cazul în care cheia straina contine o valoare, aceasta trebuie sa se refere la o aparitie existenta în relatia parinte. Pentru a asigura integritatea referentiala, se specifica anumite constrângeri de existenta, care definesc conditiile în care o cheie candidat sau straina poate fi inserata, reactualizata sau stearsa.

Pentru modelul obiectelor integritatea referentiala necesita ca orice obiect la care se face referire sa existe. De exemplu, sa consideram relatia de 1:1 dintre instanta Agentii si instanta Manager, din figura 1, instanta Agentii OID1, se refera la instanta Manager OID6.

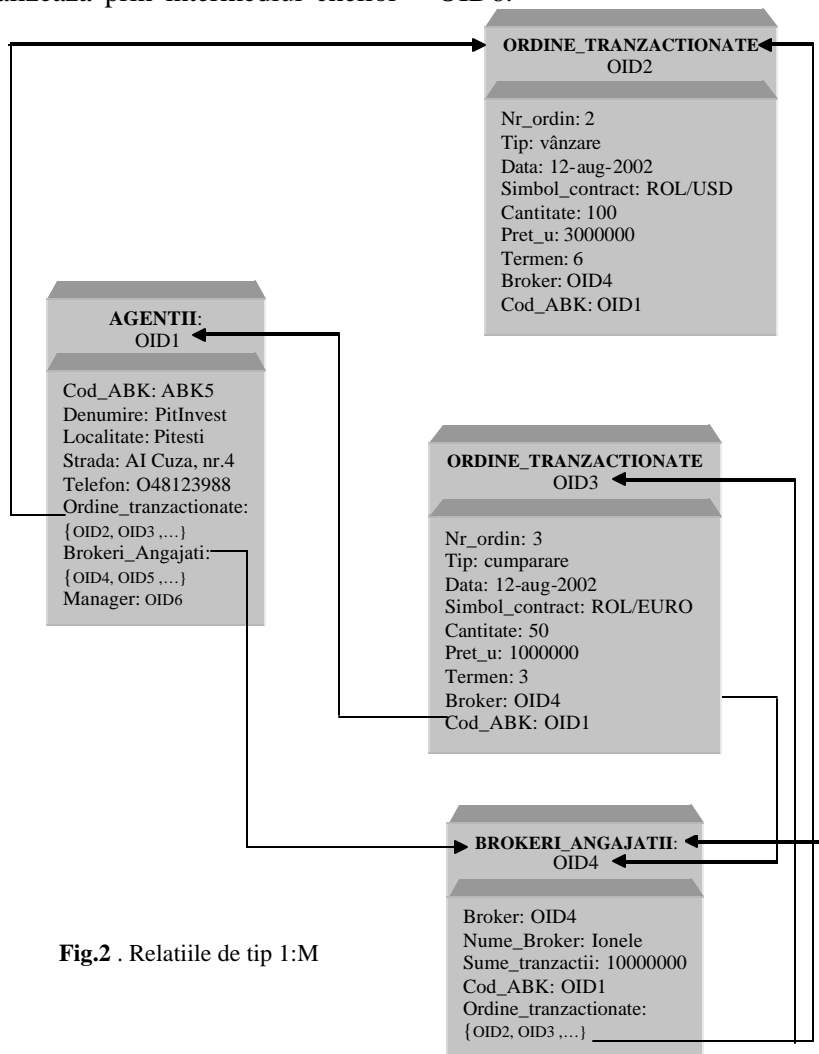


Fig.2 . Relatiile de tip 1:M

Daca utilizatorul sterge aceasta instanta Manager, fara reactualizarea corespunzatoare a instantei Agentii, atunci integritatea referentiala este pierduta. Pentru manipularea integritatii referentiale pot fi utilizate diverse tehnici, si anume:

- Sa se permita utilizatorului sa stearga explicit obiectele. În acest caz, sistemul este responsabil pentru „refacerea spatiului disponibil”. Cu alte cuvinte, sistemul sterge automat obiectele, atunci când ele nu mai sunt accesibile pentru utilizator.
- Sa se permita utilizatorului sa stearga obiectele când acestea nu mai sunt cerute. În acest caz, pentru a întari integritatea

referentiala, sistemul detecteaza automat referintele care nu sunt valabile si le atribuie valoarea NULL sau nu permite stergerea.

- Sa se permita utilizatorului sa modifice si sa stearga obiectele si relatiile, atunci când ele nu mai sunt necesare. În acest caz, sistemul mentine automat integritatea obiectelor prin utilizarea atributelor inverse. De exemplu, în figura 1 exista o relatie între Agentii si Manager si o relatie inversa între Manager si Agentii. Când un obiect de tip Manager este sters, este usor pentru sistem sa utilizeze relatia inversa, pentru a ajusta corespunzator referinta în obiectul Agentii.

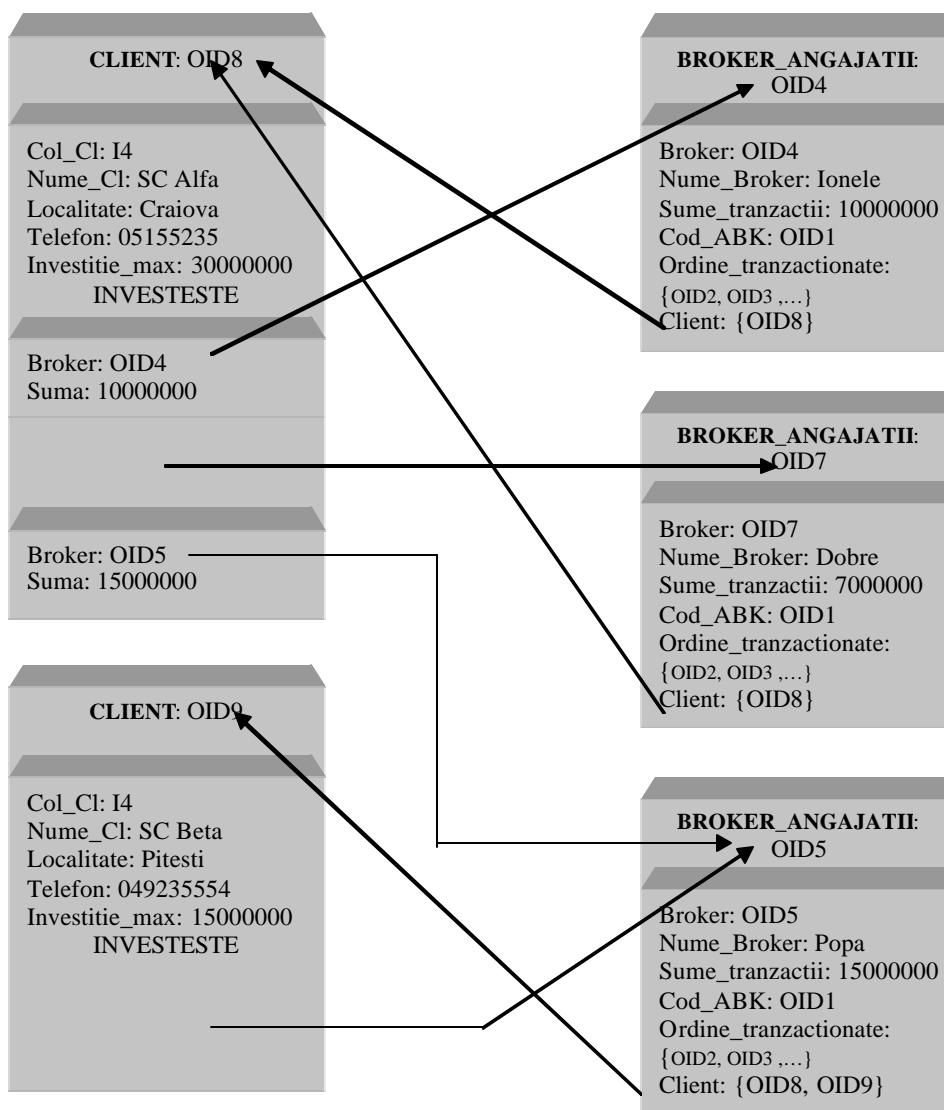


Fig. 3. Relatie de tip M:M

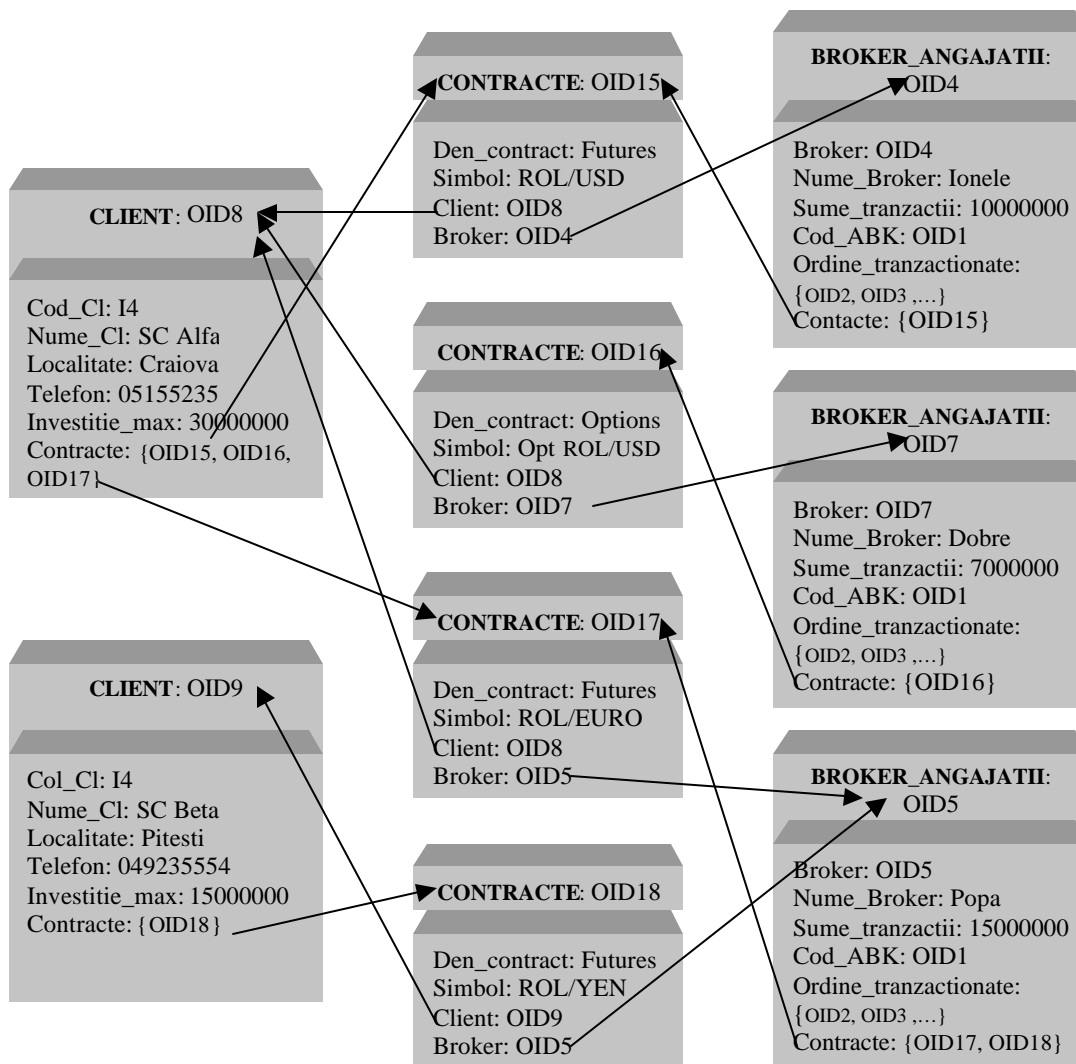


Fig.4. O proiectare alternativa a relatiei de tip M:M

Concluzii

Deși modelul de date relational are un fundament teoretic foarte puternic și câteva puncte tari: simplitate, oportunitate pentru prelucrarea on-line a tranzacțiilor, suport pentru independența datelor, totuși prezintă multe puncte slabe care conduc la o reprezentare insuficientă a entităților din „lumea reală”.

Modelul relational are o singură construcție pentru reprezentarea datelor și relațiilor dintre ele: relația. De exemplu, pentru a reprezenta o relație de tip M:M dintre două entități se creează câte o relație pentru a reprezenta fiecare dintre entitățile A și B și una pentru a reprezenta relația dintre ele. Nu există nici un mecanism prin care să se facă distincție între

entități și relații sau pentru a deosebi diferitele tipuri de relații care există între entități. În schimb modelul de date orientat spre obiecte vine în sprijinul proiectantului pentru eliminarea acestui neajuns prin interpunerea unor clase intermediare care descompun relațiile dintre obiecte.

Problemele de redundanță a informațiilor și anomaliile de reactualizare sunt evitate de identificatorul de obiecte (OID) utilizat ca mecanism pentru identitatea obiectelor. Identitatea obiectelor este cea care garantează că un obiect poate fi întotdeauna identificat în mod unic, prin aceasta asigurându-se automat integritatea instanței. Întrucât identitatea obiectului asigură caracterul unic al întregului sistem, aceasta constituie o constrângere mai

puternica decât integritatea entitatilor din modelul de date relational, care impune numai caracterul unic în cadrul unei relatii. Pentru fiecare identificator OID din sistem la care se face referire, trebuie sa fie întotdeauna prezent un obiect care sa corespunda celui identificator OID – adica nu trebuie sa existe nici o referinta independenta de obiect. În exemplul considerat, se desprinde relatia Broker *Tranzactioneaza* Ordine. Daca se înglobeaza fiecare obiect de tip Ordine în obiectul de tip Broker de care este legat, atunci sunt generate anomalii de actualizare. Daca în loc de aceasta se va îngloba identificatorul OID al obiectului de tip Ordine în obiectul de tip Broker de care este legat, atunci va continua sa existe în sistem numai o singura instanta pentru fiecare obiect, iar coerenta poate fi pastrata mult mai usor. În acest mod obiectele pot fi partajate si identificatorii OID pot fi utilizati pentru a mentine integritatea referentiala.

Printre punctele slabe ale modelului de date relational se mai înscriu: supraîncarcarea semantica, suport insuficient pentru integritate si constrângerile întreprinderii, structura de date omogena, operatii limitate, dificultate în manipularea interogarilor recursive, nepotrivire de impedanta.

Bibliografie

- [1] Davidescu N., *Sisteme informatice financiar-bancar: concepte fundamentale*, Editura All, Bucuresti, 1998
- [2] Embley D., *Object Database Development: Concepts and Principles*, Addison Wesley Longman, 1997
- [3] Oprea D., *Analiza si proiectarea sistemelor informationale economice*, Editura Polirom, Iasi, 1999.
- [4] Spiru C., Lopatan I., *Analiza, proiectarea si programarea orientate spre obiecte*, Editura Teora, Bucuresti, 1995
- [5] Soava G., *Limbaje evaluate*, Editura Reprograph, Craiova, 1999
- [6] Soava G., *Sisteme informatice pentru afaceri*, Editura Reprograph, Craiova, 2000