

LECTIA 1

Organizarea datelor în baze de date Entități, instanțe, atribute, articole

Date și informații. Colecții de date

Prelucrarea datelor în vederea obținerii informațiilor este o activitate care a apărut cu mult timp înainte de inventarea calculatorului electronic.

Date = aspecte fundamentale a unor activități sau evenimente, nesupuse unor prelucrări, neevaluate din punct de vedere al utilității.

Informații = ansambluri de date corelate și elaborate într-un scop bine determinat, pentru satisfacerea cerințelor celui care le utilizează.

Datele sunt aspecte primare ale realității înconjurătoare, care sunt percepute prin intermediul unor receptori, sub diverse forme. De exemplu, datele culese dintr-o stație meteorologică provin din diverse surse: măsurători ale factorilor atmosferici: (temperatură, umiditate, presiune atmosferică) .

informație = produs al procesului de prelucrare a datelor.

Datele sunt organizate sub forma colecțiilor și a structurilor de date un exemplu de colecție de date organizată după criterii calitative este *fișierul*.

Colecția de date reprezintă un ansamblu de date organizat după anumite criterii

Structura de date constituie o colecție de date între care s-au stabilit o serie de legături care conduc la un anumit mod de identificare și selectare a componentelor

- Actualizarea
- Ordonarea
- Interclasarea
- Copierea etc.

Conceptul de bază de date

O *baza de date* poate fi privită ca o colecție de date ce modelează un univers. Acest univers este format din mai multe *obiecte* care interacționează, obiectele de același tip constituind o *entitate*. O entitate este un element al lumii reale.

Un *model* reprezintă o abstractizare a unui sistem sau proces din lumea reală, o descriere matematică formală a sistemului. Modelul se constituie într-un set de notații și o terminologie necesară exprimării ideilor referitoare la sistemul sau procesul descris. Modelul este folosit fie pentru studierea unui sistem existent, fie pentru construirea unui sistem nou.

Modelarea este procesul prin care obținem un model. Procesul de modelare al unui sistem/proces din lumea reală se desfășoară în trei etape, așa cum ilustrează figura 1.1.

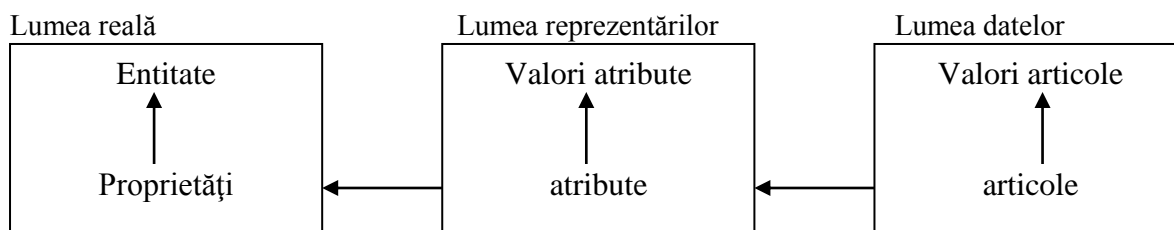


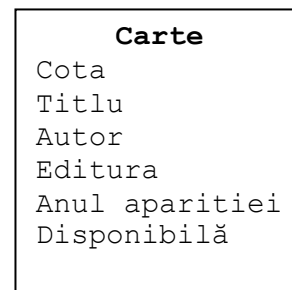
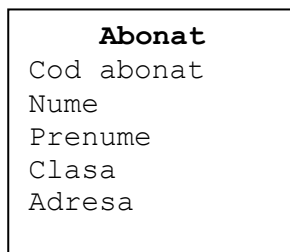
Fig. 1.1: Etapele de modelare

Fiecărui atribut i se asociază un articol.

Articolul este unitatea elementară de date, care poate fi identificată printr-un nume și poate lua valori într-un anumit domeniu de valori.

Să presupunem că dorim să realizăm gestionarea unei biblioteci școlare. Un model simplificat al activităților din cadrul bibliotecii poate conține entitățile abonat și carte, care au următoarele proprietăți relevante:

Exemplul 1.1



Cod abonat, Nume, Prenume, clasa și adresa sunt atributele entității abonat, în timp ce Cota, titlu, autor, editura, anul apariției și disponibilă sunt atribute ale entității carte.

Entitatea este o descriere generică a unei realități înconjurătoare, care grupează mai multe obiecte cu proprietăți comune, numite instanțe ale entității. De exemplu, entitatea carte descrie proprietățile generice ale unei clase de obiecte de tip carte. Două instanțe ale entității carte de deosebesc prin valorile asociate atributelor (una sau mai multe valori distincte)

Instanțele unei entități pot fi reprezentate sub forma unui tabel, având capul de tabel format din numele articolelor asociate atributelor entității.

Figura 1.2 prezintă structura posibilă a tabelului asociat entității carte. Memorarea tabelului pe un suport de memorie externă se va realiza într-un fișier special numit tabelă de date (table). Linia hașurată din tabel reprezintă o înregistrare, iar coloana hașurată indică toate valorile articolului An_apar pentru diversele instanțe ale entității carte.

Cota	Titlu	Autor	Editura românească	An apar	Disponibilă
10322	Ion	Liviu Rebreanu	Cartea românească	1970	1
10238	Mara	Ioan Slavici	RAO	2002	0
10232	Frații Jderi	Mihail Sadoveanu	Cartea românească	1980	1

Fig. 1.2 Tabela de date

Din punctul nostru de vedere nu contează modul de organizare a datelor în fișierul ce stochează tabelul, importantă fiind structura conceptuală a acestuia.

Identificator (cheie) = atribut sau mulțime de atribute care identifică fiecare instanță a unei entități.
Identificator unic (cheie primară) = atribut al unei entități, care are valori unice pentru fiecare instanță a entității.

De exemplu, mulțimea formată din atribute Autor și Editură constituie o cheie pentru entitatea carte, dar nu reprezintă o cheie primară (pot exista mai multe cărți scrise de același autor, publicate la aceeași editură.)

Este posibil sa existe 2 atribute cu valori unice pentru o anumită entitate , oricare dintre acestea putând fi alese cheie primară. Aceste atribute sunt numite chei candidat, însă numai una din acestea va fi aleasă cheie primară.

De exemplu, dacă entitatea abonat ar fi conținut și atributul *buletin_identitate* , acesta ar fi fost, alături de atributul *cod abonat*, o cheie candidat.

În practica definirii structurii conceptuale a bazei de date, fiecare entitate va avea obligatoriu o cheie primară.

Terminologie specifică domeniului bazelor de date

În domeniul bazelor de date există mai multe grupuri specializate în dezvoltarea standardelor, cum ar fi: CODASYL, DBTG, ANSI/X3/SPARC.

Definirea termenilor care urmează este realizată conform standardului CODASYL:

Articol = unitate elementară de date, care poate fi identificată prin nume
Tipul (schema) unui articol = ansamblu format din identificatorul folosit, descrierea multimii valorilor acceptabile și descrierea formatului de memorare
Înregistrare = colecție de articole
Set de date = colecție de înregistrări
Zona = subdiviziune a memoriei interne, identificabilă printr-un nume, care poate memora un set de date
Schema bazei de date = ansamblu format din lista identificatorilor folosiți și descrierea articolelor. Schema bazei de date definește un tip de bază de date.
Baza de date = totalitatea exemplarelor înregistrărilor, seturilor de date și zonelor de memorie organizate sub o anumită schemă. Schema bazei de date are un caracter preponderent static, în timp ce datele administrate au un caracter dinamic.
Dicționarul datelor = mulțimea descrierii obiectelor unei baze de date .
Administratorul bazei de date = persoană sau grup de persoane responsabilă de analiza, proiectarea, implementarea, exploatarea și întreținerea unei baze de date.
Sistem de gestiune a bazelor de date (SGBD) = ansamblu de programe destinat creării, întreținerii și exploatării unei baze de date.

Mai multe detalii privind cerințele impuse unei baze de date vor fi prezentate în capitolul 2.

Lectia 2

Diagrama entitate-relatie (I)

o baza de date este produsul unui process de modelare a unei realitati, a unui system sau process din lumea reala.

Baza de date reuneste o colectie de entitati intre care se stabilesc relatii.

Relatia intre doua entitati este gandita ca fiind o asociere, o interdependenta intre acestea.

Diagrama entitate-relatie reprezinta un model de descriere a unui sistem sau proces din lumea reala, permintand studierea acestuia si elaborarea unui model adecvat.

Diagrama entitate-relatie utilizeaza o descriere grafica pentru doua categorii de elemente:

- Entitati;
- Legaturi (asocieri intre entitati);

Entitati.

Entitate = persoana, loc, activitate sau eveniment care este semnificativ pentru modelul realizat.

Sa presupunem ca avem de rezolvat problema gestionarii datelor referitoare la elevii dintr-o institutie scolara. La o prima analiza, putem indentifica doua entitati absolut necesare modelului nostru:

- Entitatea elev
- Entitatea clasa

Pentru entitatea elev vom retine acele caracteristici necesare modelului nostru, cum ar fi: numele, prenumele, data nasterii, scoala de provenienta etc. Nu ne intereseaza insusiri ale entitatii elev cum ar fi inaltimea, culoarea ochilor sau genul de muzica preferat, deoarece sunt caracteristici irelevante pentru scopul nostru.

O entitate din cadrul modelului trebuie sa se conformeze regulilor urmatoare:

- Numele unei entitati trebuie sa fie un substantiv, dar nu orice substantiv poate fi utilizat ca nume pentru o entitate. Se vor utiliza nume sugestive, care sa permita o prima descriere a entitatii.
- De exemplu, nu putem crea o entitate cu numele *Seful clasei*, deoarece seful clasei este elev al unei clase si are caracteristicile identificate la entitatea elev. Cel mult putem introduce un nou atribut pentru entitatea elev si anume atributul *Sef*, avand valoarea 1 daca elevul este seful clasei si zero in caz contrar.
- Nu pot exista doua entitati cu acelasi nume sau o entitate cu doua nume diferite;
- Pentru fiecare entitate se va da o descriere completa a atributelor sale (semnificatie, domeniu de valor);
- **Entitatea va avea obligatoriu un atribut sau o combinatie de attribute, care sa permita indentificarea in mod unic a fiecarei instante a entitatii, adica o cheie primara.** Cheia primara trebuie sa satisfaca urmatoarele cerinte:

1. sa fie stabila – modificarea frecventa a cheii primare duce la modificari in intreg modelul realizat si, implicit, in programele de aplicatie care gestioneaza baza de date;
2. sa fie suficient de simpla si fara ambiguitati;
3. sa fie familiara utilizatorilor bazei de date, astfel incat acestia sa o poata utiliza in cadrul operatiunilor efectuate asupra bazei de date.
De exemplu, o cheie primara potrivita pentru entitatea elev este atributul *CNP* (codul numeric personal), deoarece este unic pentru orice persoana si reprezinta o modalitate de identificare folosita pe scara larga.

Diagrama entitate-relatie utilizeaza o descriere grafica a entitatilor, folosind urmatoarele conventii:

- descrierea entitatii este realizata intr-o zona dreptunghiulara bine delimitata;
- numele entitatii este scris pe prima linie din cadrul zonei dreptunghiulare, cu litere mari

- atributele sunt scrise cu litere mici pe liniile urmatoare

Atribute

Atribut = proprietate descriptiva a unei entitati, caracteristica a unei entitati.

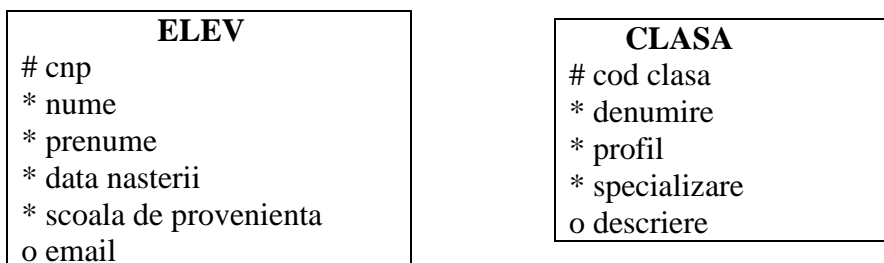
Nume, prenume, data nasterii sun atribute ale entitatii *ELEV*. Numele unui atribut este un substantiv, care descrie cu exactitate o caracteristica a entitatii si exprima o insusire calitativa sau cantitativa a entitatii.

Un atribut trebuie sa aiba o singura valoare, de un anumit tip (intreg, real, sir decaractere, data calendaristica etc.). Exista atribute a caror valoare este obligatorie pentru a descrie corect si complet o entitate. Aceste atribute le vom numi *attribute obligatorii*, in timp ce atributele a caror valoare poate lipsi vor fi numite *attribute optionare*.

Reprezentarea atributelor in cadrul diagramei entitate-relatie se realizeaza conform cu urmatoarele reguli:

- atributele sunt scrise cu litere mici, fiecare pe o linie;
- atributele care sunt chei primare sunt precedate de caracterul '#';
- atributele obligatorii sunt precedate de caracterul '*';
- atributele optionale sunt precedate de caracterul 'o'.

Exemplu 1.2 prezinta entitatile *ELEV* si *CLASA*, reprezentate grafic, conform convetiilor enuntate mai sus.



Exemplul 1.2

Atributul *email* al entitatii *ELEV* si atributul *descriere* al entitatii *CLASA* sunt atributele optionale, deoarece nu reprezinta o caracteristica esentiala a entitatilor respective. Valoarea atributului

Email poate ramane nedefinita, deoarece nu orice elev are in mod obligatoriu o adresa de posta electronica.

In acest caz, spunem ca valoare atributului email poate fi valoare *NULL*.

NULL = constanta predefinita care semnifica o valoare nedefinita sau inaplicabila.

Valoarea *NULL* nu trebuie confundata cu *sirul vid* ori cu valoarea *0*. Atributele obligatorii nu pot avea valoarea *NULL*.

Relatii

Relatie = asociere, mod de comunicare, interdependenta stabilita (de

obicei) între două entități.

Relatia reprezintă un raport care există între entități și este exprimată prin utilizarea unor verbe care să descrie cu exactitate interdependența dintre acestea. Pot exista mai multe relații cu același nume, dar entități diferite.

Diagrama entitate-relatie reprezintă relațiile printr-o linie care unește cele două entități, la fiecare dintre cele două capete (să presupunem stanga și dreapta) scriindu-se numele relației. Numele relației este format dintr-un cuvânt sau grup de cuvinte, care conține obligatoriu un verb, și exprimă modul în care entitatea din acea parte este asociată cu entitatea din cealaltă parte a relației.

Să identificăm pentru început o relație între entitățile *ELEV* și *CLASA*. Evident, fiecare clasă *poate conține* un număr oarecare de elevi și fiecare elev *trebuie să aparțină* unei clase. Chiar dacă, în realitate, o clasă conține în mod obligatoriu cel puțin 20 de elevi, în cadrul unei baze de date ce modelează această realitate, datele despre o anumită clasă pot fi introduse în baza de date înainte de a insera datele referitoare la elevii aveau clase. Astfel, relația dintre entitățile *ELEV* și *CLASA* poate fi exprimată prin următoarele două enunțuri:

- I. Fiecare *CLASA* **poate conține** 0 sau mai mulți *ELEVI*.
- II. Fiecare *ELEV* **trebuie să aparțină** unei singure *CLASE*.

Enunțurile corespund celor două părți ce compun o relație. Cuvintele “cheie” din exemplul nostru sunt cele îngroșate: **poate** și **trebuie**. Aceste două cuvinte exprimă *optionalitatea relației*. Cuvintele scrise aldin exprimă *cardinalitatea relației* sau *gradul relației*.

Cardinalitatea relației = proprietate a celor două părți ale unei relații dintre entitățile A și B, care exprimă **câte** instanțe ale entității A sunt asociate cu instanțele entității B și reciproc.
Optionalitatea relației = proprietate a celor două părți ale unei relații dintre entitățile A și B, care exprimă **câte** dintre instanțele entității A **pot** sau **trebuie** să se asocieze instanțelor entității B și reciproc.

Pe baza enunțurilor I și II putem deduce cardinalitatea relației dintre entitățile *ELEV* și *CLASA*: **many-to-one** (mai multi-la unu sau *n:1*).



Să considerăm entitățile *SALARIAT* și *DEPARTAMENT*, identificate în cadrul modelului de gestiune a datelor unei firme oarecare, și relația următoare:



Incercam sa determinam optionalitatea si cardinalitatea relatiei prin enunturile:

Fiecare SALARIAT **poate conduce** 0 sau *un singur* DEPARTAMENT.
Fiecare DEPARTAMENT **trebuie condus** de un *unic* SALARIAT.

Relatia dintre cele doua entitati are cardinalitatea **one-to-one** (unu-la-unu sau 1:1).

Optionalitatea relatiei este exprimata prin:

- *nu este obligatoriu* ca fiecare salariat sa fie seful unui departament, dar acest lucru *este posibil* (un salariat *poate conduce* un departament);
 - un departament *trebuie* sa aiba un sef (*trebuie sa fie condus*), dar acesta este unic.
- Sa consideram acum entitatile CONCURENT si PROBA, care intervin in gestionarea datelor

unui concurs atletic. Vom determina cardinalitatea si optionalitatea relatiei:



Incercam sa determinam optionalitatea si cardinalitatea relatiei prin enunturile:

Fiecare CONCURENT **poate participa** la *una sau mai multe* PROBE.
Fiecare PROBA **poate fi sustinuta** de 0 sau *mai multi* CONCURENTI.

Relatia dintre cele doua entitati are cardinalitatea **many-to-many** (mai multi-la-mai multi sau n:m).

In privinta optionalitatii relatiei putem face urmatoarele aprecieri:

- un concurent nu este obligat sa participe la mai multe probe, dar poate face acest lucru;
- proba de concurs *poate* reuni la start *mai multi* concurenti, asa cum este firesc (*poate fi sustinuta*) sau exista posibilitatea ca niciun concurent sa nu participe la acea proba.

LECTIA 3

DIAGRAMA ENTITATE-RELATIE (II)

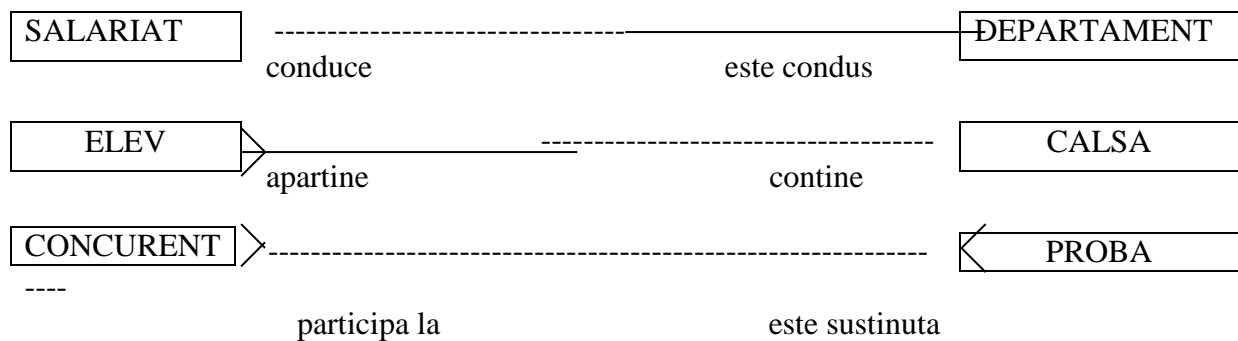
In lectiile precedente au fost prezentate componentele diagramei entitate-relatie (entitati, attribute, relatii), precum si cateva conventii privind modul de reprezentare grafica a acestora. In aceasta sectiune vom prezenta conventiile privind reprezentarea grafica a relatiilor dintre entitati si vom modela o problema de gestiune a datelor din viata de zi cu zi.

Reprezentarea grafica a unei relatii stabilite dintre doua entitati A si B se va realiza prin respectarea urmatoarelor reguli:

- Entitatile sunt reprezentate printr-un dreptunghi, avand numele entitatii scris in interior, cu majuscule; eventual, dreptunghiul va contine si descrierea atributelor entitatii;
- Relatia va fi reprezentata printr-o linie ce uneste cele doua entitati(segment de dreapta); portiunea dintre entitatea A si mijlocul segmentului reprezinta latura/partea

stanga a relatiei, iar portiunea dintre entitatea B si mijlocul segmentului reprezinta latura/partea dreapta a relatiei

- De fiecare parte a liniei ce reprezinta relatia, se va nota, cu litere mici, numele relatiei;
- Optionalitatea relatiei va fi reprezentata, pe fiecare latura a relatiei, prin utilizarea unei linii punctuate (**poate fi**) sau continue (**trebuie**);
- Cardinalitatea relatiei este reprezentata de partea terminala a liniei din fiecare parte a relatiei; daca avem cardinalitatea n pe o latura a relatiei, atunci linia ce descrie relatia se va desparti in trei linii (sub forma de trident).
- Reluam exemplele de relatii definite in lectia precedenta, reprezentandu-le conform regulilor enuntate:



Exemplul 1.3

Exemplul 1.3 ne arata cum putem reprezenta graphic, in cadrul diagramei entitate-relatie, relatiile stabilite intre doua entitati,. Cum putem citi corect aceste relatii?

Simplu, respectand totdeauna urmatoarea succesiune de pasi:

- Pas 1. **fiecare** entitate A
- Pas 2. **poate/trebuie** (optionalitate)
- Pas 3. **nume relatie**
- Pas 4. **unul si numai unul/mai multe** (cardinalitatea)
- Pas 5. entitate B

Regula de mai sus se aplica pentru ambele parti ale unei relatii, asa cum ilustreaza exemplul 1.4

- Pas 1. **fiecare** SALARIAT
- Pas 2. **poate** (optionalitatea)
- Pas 3. **conduce**
- Pas 4. **unul si numai unul** (cardinalitatea)
- Pas 5. DEPARTAMENT (E)

Exemplul 1.4 (a)

- Pas 1. **fiecare** DEPARTAMENT

Pas 2. **trebuie (optionalitatea)**
 Pas 3. **condos de**
 Pas 4. **unul si numai unul (cardinalitatea)**
 Pas 5. **SALARIAT**

Exemplul 1.4 (b)

Exemplul 1.4 (a) ilustrat aplicarea exacta a regulii de citire a unei relatii, dar practica ne obliga sa ajustam aceste enunturi conform cu normele exprimarii corecte si coerente ale limbii romane. Exemplul 1.4(b) poate fi exprimat, fara a pierde din semnificatie astfel:

Pas 1. **fiecare DEPARTAMENT**
 Pas 2. **trebuie (optionalitatea)**
 Pas 3. **condos de**
 Pas 4. **un unic (cardinalitatea)**
 Pas 5. **SALARIAT**

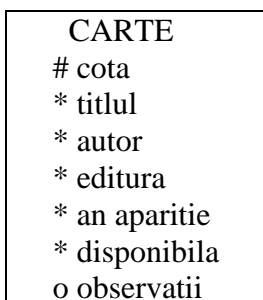
Exemplul 1.5

Gestionarea datelor dintr-o biblioteca. Studio de caz

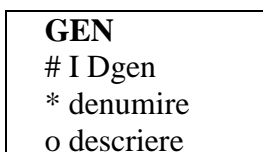
In aceasta sectiune ne propunem sa realizam modelul conceptual al unei probleme concrete de gestiune a datelor: gestionarea datelor unei biblioteci scolare.

Vom identifica entitatile care apar si atributurile acestora, vom defini relatii intre entitati si vom reprezenta modelul folosind diagrama entitate-relatie

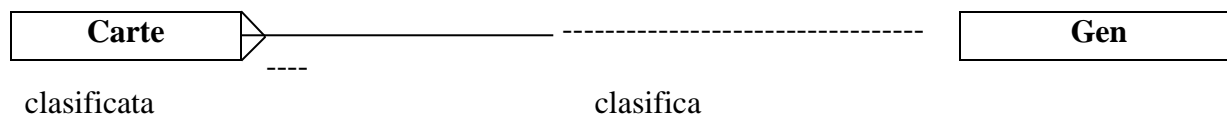
Pas 1. deoarece ratiunea existentei unei biblioteci este tocmai manipularea cartilor, incepem constructia modelului cu entitatea care apare in mod evident; entitatea CARTE



Pas 2. fiecare carte este incadrata de catre criticii literari intr-un anumit current literar sau intr-un anumit gen literar. Astfel, putem defini entitatea GEN, care este utilizata la clasificarea cartilor din biblioteca in functie de curentul literar caruia ii apartine.



Pas 3. stabilim relatia dintre entitatile CARTE si GEN; este many-to-one:



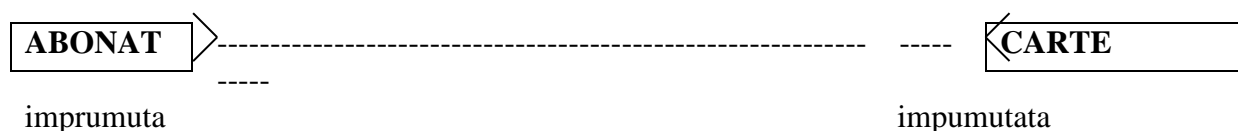
Relatia, de tip many-to-one, poate fi citita astfel:

Fiecare CARTE **trebuie** clasificate printr-un unic GEN.
 Fiecare GEN **poate** clasifica *una* sau *mai multa* CARTI.

Pas 4. Cartile, clasificate sau nu, trebuie imprumutate elevilor sau profesorilor din cadrul unitatii de invatamant la care ne referim, deci este nevoie de o noua entitate: ABONAT.

ABONAT
 # cnp
 * nume
 * prenume
 * adresa

Pas 5. Identificam relatiile care se pot stabili intre entitatea ABONAT si entitatile existente, CARTE si GEN. ratiunea existentei abonatilor sau clientilor bibliotecii este aceea de a imprumuta carti, deci vom defini o relatie intre entitatile ABONAT si CARTE:



relatia este de tip many-to-many, deoarece:

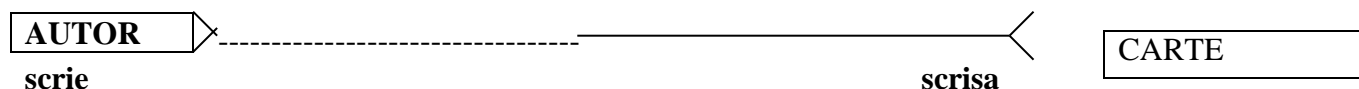
Fiecare ABONAT **poate** imprumuta una sau mai multe CARTI
 Fiecare CARTE **poate** fi imprumutata de unul sau mai multi ABONATI

Pas 6. sa presupunem ca biblioteca noastra doreste sa puna la dispozitie publicului date referitoare la autorii cartilor. Vom defini o noua entitate, AUTOR:

AUTOR
 # I Dautor
 * nume
 * prenume
 * data nasterii
 o data decesului
 * locul nasterii
 * nationalitatea

Pas 7. Stabilim relatiile intre ultima entitate introdusa in model si celelalte entitati. Abonatii bibliotecii vor dori, probabil, fie date referitoare la autorii unei carti, fie date referitoare la

cartile scrise de anumiți autori. Prin urmare, vom defini o asociere între entitățile CARTE și AUTOR.



Relația este de tip many-to-many, deoarece:

Fiecare AUTOR **poate** scrie una sau mai multe CARTI

Fiecare CARTE **trebuie** scrisă de unu sau mai mulți AUTORI

Pas 8. Bazându-ne pe entitățile și relațiile identificate la pașii anteriori, desenăm diagrama entitate-relație:

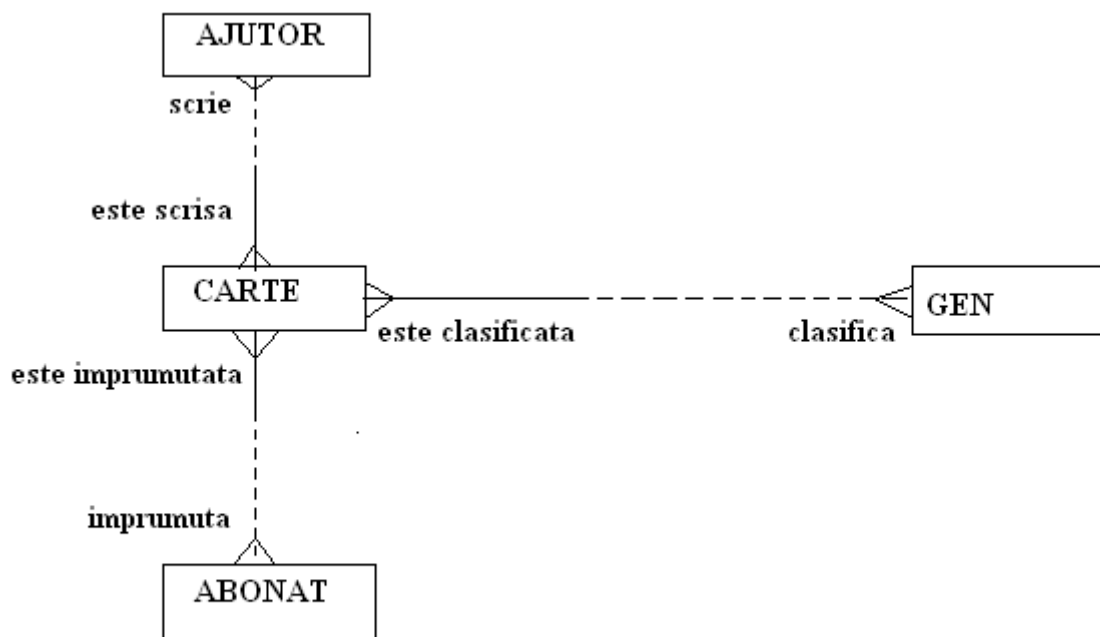


Figura 1.3 Diagrama entitate-relație

LECTIA 4

Rezolvarea relațiilor many-to-many

După realizarea primei variante a diagramei entitate-relație, avem deja o privire de ansamblu asupra structurii conceptuale a datelor din problema concretă de gestiune pe care o modelăm. Etapele următoare ale procesului de modelare constau în rafinarea pas cu pas a modelului inițial, vizând îmbunătățirea acestuia.

În practica procesului de proiectare a modelului conceptual nu sunt acceptabile relațiile many-to-many care apar dese ori între entități. Primul pas către rafinarea modelului constă în eliminarea relațiilor many-to-many și înlocuirea acestora cu relații one-to-many.

Pentru a vedea de ce este necesar sa eliminam relatiile many-to-many, sa analizam cu atentie relatia stabilita intre entitatile AUTOR si CARTE.

Sunt posibile doua situatii:

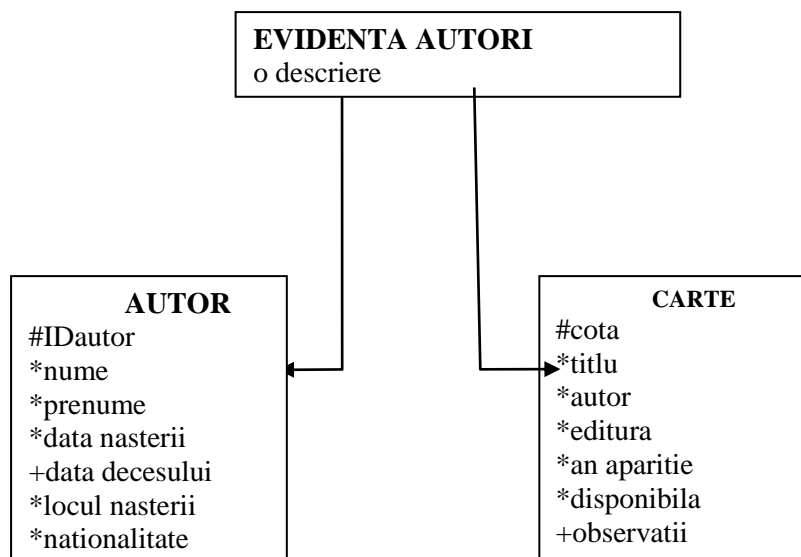
- Daca dorim sa memoram, pentru fiecare carte, doar numele autorului, fara sa oferim si alte detalii legate de aceste, atunci existenta entitatii AUTOR nu mai are sens, problema rezolvandu-se prin definirea atributului autor in cadrul entitatii CARTE si eliminarea din model a entitatii AUTOR.
- Daca dorim sa oferim abonatilor informatii despre autorii cartilor din biblioteca, sau sa permitem cautarea cartilor scrise de un autor precizat, atunci relatia many-to-many nu poate fi evitata.

Consideram ca ne aflam in a doua situatie, deci trebuie sa gasim o cale de a elimina relatia many-to-many dintre cele doua entitati. De ce, totusi, trebuie sa facem acest lucru? Sa efectuam o scurta analiza:

- Pentru relatia considerata stim ca un scriitor este autorul uneia sau a mai multor carti, dar nu putem sti cum se realizeaza aceasta asociere. Pentru autorul Mihai Eminescu putem determina, folosind cele doua entitati si relatia dintre ele, care sunt cartile scrise de acesta si sunt disponibile in biblioteca ? Raspunsul este NU!
- Reciproc, exista vreo modalitate de a afla care sunt autorii unei carti date? Raspunsul este din nou negativ.

In concluzie, necesitatea eliminarii relatiilor many-to-many din cadrul modelului este data de gradul mare de ambiguitate generat de acestea.

Relatia many-to-many va fi inlocuita in cadrul modelului prin doua relatii one-to-many, stabilite intre entitatile initiale si o a treia entitate, nou introdusa, numita entitate de legatura sau entitate de intersectie.



Inlocuirea unei relatii many-to-many existente intre entitatile A si B se va realiza astfel:

- se introduce o noua entitate C, aceasta avand identificatorul unic format din cheile primare ale celor doua entitati, plus alte attribute suplimentare, nu este obligatoriu ca unicul identificator al entitatii de intersectie sa fie compus din unicul identificator al relatiei originale (se poate introduce un identificator propriu unic);
- relatia dintre entitatile A si C este o relatie one-to-many;
- relatia dintre entitatile B si C este o relatie one-to-many;

- relatiile one-to-many dintre entitatile initiale si entitatea de intersectie C se reprezinta graphic prin adaugarea unei bare orizontale la capatul dinspre entitatea C.

Barele orizontale trasate in cadrul diagramei entitate-relatie arata faptul ca entitatea de intersectie C nu va contine explicit identificatori unici ai entitatilor din relatia initiala, acestia fiind dedusi din relatie. In momentul trecerii de la modelul conceptual la modelul fizic, tabela corespunzatoare entitatii C va contine campurile id autor si cota, care vor fi denumite chei straine.

Entitatea de intersectie introdusa in exemplul nostru, EVIDENTA AUTORI, constituie in fapt un fel de “jurnal”, care are ca instante perechi (AUTOR, CARTE), pastrand astfel asocierea dintre fiecare autor si cartile pe care le-a scris, singur sau impreuna cu alti autori.

Instantele entitatii nu retin decat cheile primare din cele doua entitati, pentru identificarea caracteristicilor cartilor si autorilor si, eventual, cateva attribute suplimentare ce descriu asocierea respectiva.

Nu este permisa mutarea anumitor attribute din entitatile AJUTOR sau CARTE in entitatea EVIDENTA AUTORI, deoarece modelul va deveni redundant (aceleasi date vor fi memorate, inutil, in mai multe inregistrari ale bazei de date). De exemplu, mutarea atributului *nationalitate* din entitatea AUTOR in entitatea EVIDENTA AUTORI determina memorarea rationalitatii unui autor pentru toate cartile pe care le-a scris.

Reanalizand modelul construit pentru problema gestionarii datelor unei biblioteci scolare (fig. 1.3) si inlocuind cele doua relatii many-to-many, obtinem un model nou (vz. Fig. 1.5).

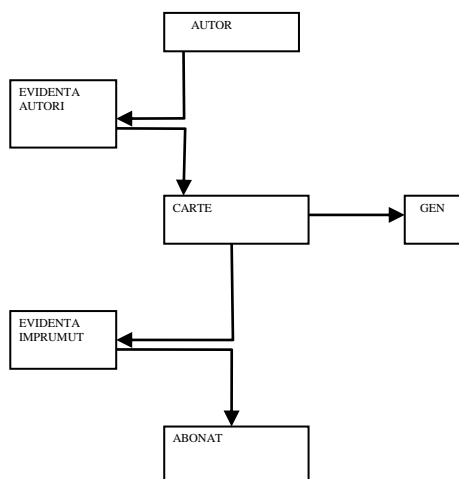


Fig 1.5

Scopul procesului de modelare a unei probleme de gestiune este acela de a proiecta o structura conceptuala a datelor flexibila, care sa poata satisface necesitatile informationale ale unei categorii largi de utilizatori.

Structura conceptuala este bine proiectata daca permite memorarea datelor in “locul potrivit” si intr-un singur exemplar.

Redundanta(a datelor) = caracteristica a unui model conceptual/baza de date de a contine aceleasi date in mai multe exemple.

Unul dintre cele mai importante scopuri ale unei baze de date este acela de a elimina, pe cat posibil, fenomenul de redundanta a datelor.

Sa analizam oportunitatea existentei atributului autor in cadrul entitatii CARTE. Acest atribut a fost introdus in prima faza a realizarii modelului, in ipoteza ca orice carte are un singur autor, si ca nu vom memora decat numele acestuia. Odata cu introducerea entitatii AUTOR si relationarea ei cu entitatea CARTE, atributul autor devine inutil si trebuie eliminate din cadrul modelului.

O alta problema ce trebuie evitata intr-o structura conceptuala este posibilitatea ca un anume gen de informatie sa fie memorata in mai multe locuri. De exemplu, san e imaginam ca o parte din lista adreselor de contact ale furnizorilor unei firme este memorata in entitatea FURNIZOR, iar alta parte in entitatea CONTRACTE.

Eliminarea deficientelor constatate in cadrul unui model conceptual se poate realize prin *procesul de normalizare*.

Normalizare = process de ameliorare progresiva (in etape) a unei scheme conceptuale. La finalul fiecarei etape schema conceptuala atinge un anumit “grad de perfectiune”, pe care convenim sa-l numim **forma normala**.

Normalizarea porneste de la o structura conceptuala de baza, care contine entitati si relatiile dintre acestea, si incearca obtinerea unei scheme conceptuale ce satisface anumite standarde (reguli). Exista mai multe seturi de reguli prin aplicarea carora modelul nostrum de baza este transformat intr-un model ce se conformeaza standardelor. Dupa aplicarea fiecarui set de reguli, spunem ca modelul nostrum se afla intr-o anumita forma normala.

Teoria matematica a normalizarii a fost elaborata de F.Codd si se bazeaza pe aplicarea a 5 seturi de principii, ducand la trecerea structurii conceptuale, pe rand, prin 5 forme normale, notate FN1, FN2, FN3, FN4, FN5.

Modelul de baza este studiat in vederea identificarii incalcarii principiilor primei forme normale. Dupa eliminarea deficientelor constatate spunem ca *modelul este in prima forma normala* sau *se conformeaza primei forme normale*. Se aplica acelasi procedeu pentru urmatoarele forme normale.

In practica, se utilizeaza frecvent doar primele trei forme normale.

Prima forma normala.

Structura conceptuala se afla in prima forma normala daca:

- fiecarui atribut ii corespunde o valoare indivizibila
- oricare entitate nu contine attribute sau grupuri de attribute repetitive

Regulile primei forme normale sunt reguli minimale, pe care trebuie sa le respecte orice structura conceptuala.

Sa luam ca exemplu entitatea ANGAJAT, care memoreaza datele referitoare la salariati unei firme ce produce software, definite astfel:



Modul in care este definita entitatea ANGAJAT incalca ambele reguli ale primei forme normale, astfel:

- atributul *nume* si *prenume* nu este indivizibil, el avand o valoare compusa (o valoare pentru nume si alta pentru prenume);
- attributele *email1*, *email2* si *email3* au valori care se vor repeta pentru mai multe instante ale entitatii ANGAJAT (cu siguranta exista mai multi salariati care au o singura adresa e-mail, iar utilizare celorlalte attribute este inutila).

Aducerea entitatii ANGAJAT la prima forma normala se realizeaza prin aplicarea urmatorului algoritm general:

Algoritm pentru aducerea structurii conceptuale la prima forma normala:

Pas 1. Pentru fiecare atribut care are valoare compusa se introduc noi attribute, cate unul pentru fiecare componenta a valorii initiale.

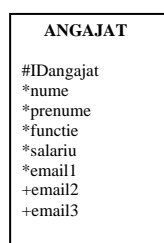
Pas 2. Fiecare atribut sau grup de attribute repetitive se plaseaza intr-o entitate noua.

Pas 3. Se alege pentru fiecare entitate obtinuta la pasul 2 o cheie primara si se stabilesc attributele ce descriu entitatea (attributele repetitive identificate la structura de baza plus alte attribute ce descriu noua entitate).

Pas 4. Se stabilesc relatiile de asociere intre entitatile initiale si entitatile nou introduse.

Pentru exemplul anterior, care incalca ambele reguli ale primei forme normale, procedam astfel:

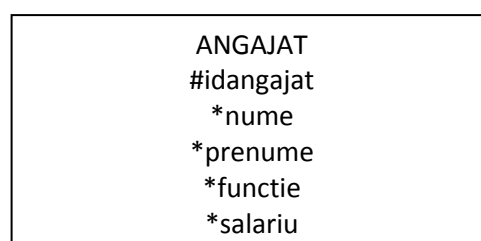
- atributul nume si prenume va fi descompus in doua attribute separate: nume, prenume;



- Attributele email1, email 2, email3 vor fi atasate unei entitati noi numite CONTACT



- Entitatea ANGAJAT va fi modificata astfel:



- Stabilim o relatie one-to-many intre entitatile ANGAJAT SI CONTACT, deoarece:

Fiecare angajat poate sa posede unul sau mai multe adrese de CONTACT.
Fiecare CONTACT trebuie sa identifice un unic ANGAJAT.

- Relatia este in prima forma normal si se poate reprezenta astfel:



A doua forma normala

Structura conceptuala se afla in a doua forma normal daca:

- Structura este in prima forma normal
- Fiecare atribut care nu este cheie ester dependent de intreaga cheie primara.

Structura conceptuala se afla in a doua forma normala daca:

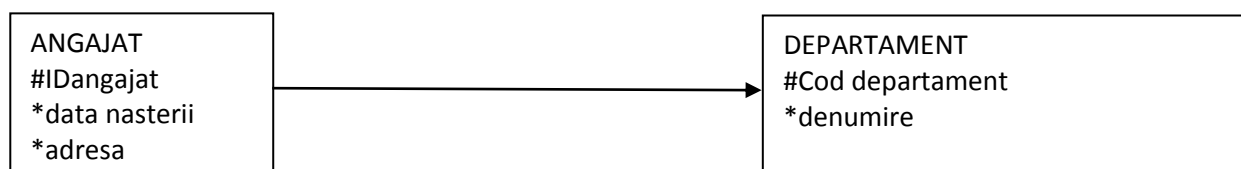
- Structura este in prima forma normala
- Fiecare atribut care nu este cheie este dependent de intreaga cheie primara.

Sa consideram entitatea department, folosita pentru memorarea informatiilor referitoare la angajatii si departamentele unei firme:

DEPARTAMENT
#cod departament
#idangajat
*data nasterii
*adresa

Cheia primara a entitatii DEPARTAMENT este o combinatie intre attributele cod department si idangajat. Observati ca attributele data_nasterii si adresa sunt dependente doar de o parte a cheii primare (IDangajat) si nu de intreaga cheie primara.

Rezolvarea acestei problem se realizeaza prin introducerea unei entitati noi, entitatea ANGAJAT, legata de entitatea DEPARTAMENT printr-o relatie one-to-many.



Lucreaza

compus

Sa consideram al doilea exemplu, pentru o mai buna intelegere a dependentei unui atribut de cheia primara a entitatii.

Consideram ca, in cadrul unei firme de software, un angajat lucreaza la unul sau mai multe proiecte, indeplinind una sau mai multe sarcini in cadrul fiecaruia. Structura conceptual a ar putea fi aceea ilustrata in figura 1.6.

Observati ca attributele beneficiar si email din cadrul entitatii de legatura SARCINA nu sunt dependente de intreaga cheie primara. Atributul email este dependent de atributul idangajat, iar atributul beneficiar depinde de IDproiect, deci sunt caracteristici ale entitatilor ANGAJAT, respective PROIECT.

Rezolvarea incalcarii principiilor celei de-a doua forme normale se realizeaza prin plasarea atributelor email si beneficiar in entitatile corespunzatoare acestora.

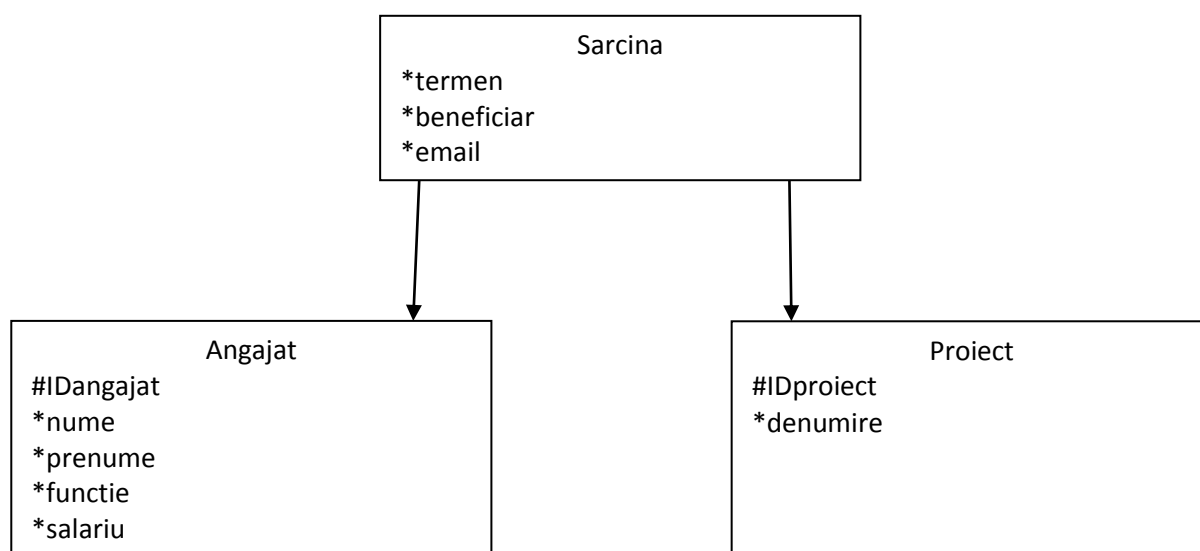


Figura 1.6 Structura conceptual in a doua forma normal

A treia forma normala

Structura conceptuala se afla in a doua forma normala daca:

- Structura este in a doua forma normala
- Fiecare atribut care nu este cheie este dependent direct de cheia primara (nu exista dependente tranzitive de cheia primara);

In general, spunem ca un atribut A este dependent tranzitiv de atributul K, daca exista un atribut B astfel incat:

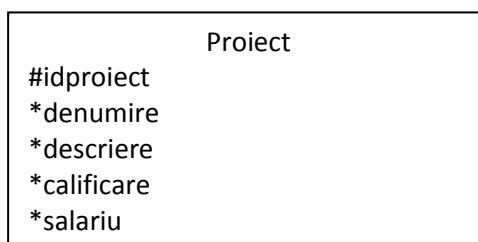
B este dependent de K;

A este dependent de B.

Structura conceptual se afla in cea de-a treia forma normal, daca nu exista niciun atribut care sa depinda tranzitiv de o cheie primara.

Sa exemplificam prin considerarea entitatii PROIECT, care memoreaza date referitoare la proiectele derulate de o firma. Presupunem ca fiecare proiect necesita o anumita calificare

pentru angajatul care-l coordoneaza, fiecare calificare avand acelasi nivel de salarizare, indiferent de proiect.



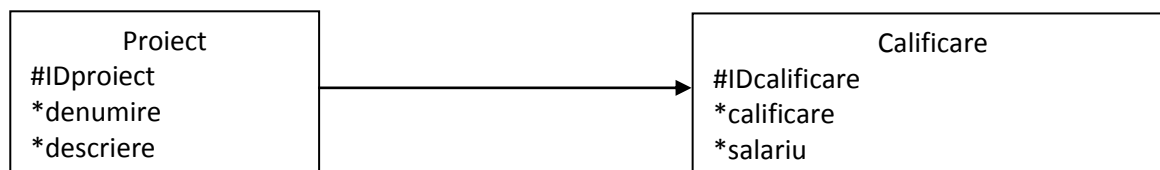
Tabelul urmator contine cateva instante ale entitatii PROIECT:

IDProiect	Denumire	Calificare	Salariu
10001	Program pentru gestiunea stocurilor	MBA	4000
10002	Aplicatie pentru arhivarea documentelor	MASTER	3000
10003	Aplicatie pentru gestiunea documentelor	MASTER	3000
10004	Magazin virtual	MBA	4000

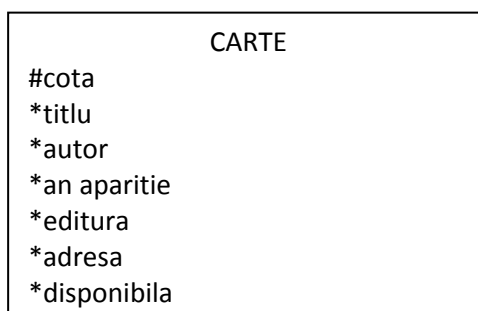
Studiind tabelul de mai sus descoperim o redundanta a datelor, deoarece atributul salariu are valori identice pentru toate instantele corespunzatoare unor proiecte cu acelasi nivel de calificare. In plus, atributul salariu este dependent de atributul calificare, acesta fiind dependent de cheia primara a entitatii. Astfel, am descoperit o dependent tranzitiva a unui atribut de cheia primara a entitatii PROIECT.

Rezolvarea dependentelor transitive se realizeaza prin introducerea unei entitati noi in cadrul modelului, care va contine atributul dependent si, eventual, alte attribute ce descriu noua entitate.

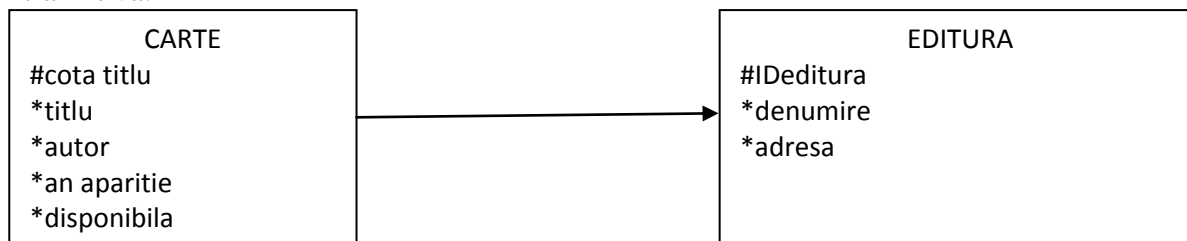
Figura 1.7 prezinta schema conceptual a exemplului considerat, aflata in a treia forma normal.



Sa consideram un alt exemplu pentru a ilustra dependenta tranzitiva. Fie entitatea CARTE, care memoreaza informatii despre cartile existente intr-o biblioteca. Așa cum se observa in descrierea entitatii, atributul adresa nu depinde de cheia primara, nefiind o caracteristica a entitatii CARTE, ci este dependent de atributul editura. Cum atributul editura este dependent de cheia entitatii CARTE, rezulta o dependent tranzitiva a atributului editura de cheia primara, deci schema conceptual nu se conformeaza principiilor celei de-a treia forme normale.



Entitatea carte, descrisa in continuare, poate fi trecuta in a treia forma normal prin introducerea unei entitati noi, EDITURA, unde mutam attributele implicate in dependent tranzitiva.



LECTIA 6: MODELAREA UNEI PROBLEME DE GESTIUNE. STUDIU DE CAZ. ETAPELE UNUI PROCES DE MODELARE

Modelarea unei probleme de gestiune se realizeaza printr-un process compus din urmatoarele etape:

PAS1: identificarea si definirea cerintelor utilizatorilor finali

In urma unei serii de intalniri dintre echipa de analisti si reprezentantii beneficiarului proiectului, sunt stabilite principalele cerinte pe care trebuie sa le indeplineasca modelul conceptual. Identificarea cerintelor nu poate fi realizata fara o buna cunoastere a tuturor detaliilor procesului de gestiune ce trebuie modelat.

Strangerea datelor necesare poate fi realizata prin observarea sistematica a procesului ce trebuie modelat sau prin intalniri directe cu beneficiarul. In multe cazuri, mai ales cand beneficiarul nu poseda cunostiinte in domeniul informatics, se alege varianta unui interviu/ chestionar, cerintele stabilindu-se in functie de raspunsurile date de beneficiar.

Succesul acestui demers depinde in mare masura de abilitatea analistului de a formula intrebarile potrivite si de claritatea cu care beneficiarul isi reprezinta mental ceea ce doreste de la acel proiect.

Finalitatea acestui pas consta in realizarea unor documente care descriu amanuntit problem ce trebuie medelata, precum si cerintele pe care trebuie sa le indeplinseasca modelul.

PAS 2 Proiectarea structurii conceptuale a modelului:

- Identificarea entitatilor;
- Stabilirea atributelor fiecarei entitati
- Stabilirea relatiilor dintre entitati

Finalitatea acestei etape este reprezentata de schema conceptuala a datelor, in general aceasta fiind descrisa prin diagrama entitate-relatie.

Pas 3. Aceasta etapa realizeaza ameliorarea schemei conceptuale, rezultate la finalul etapei precedente, prin:

- identificarea si corectarea abaterilor schemei conceptuale de la principiile primelor trei forme normale;
- eliminarea relatiilor many-to-many.

Daca schema conceptuala mai prezinta anomalii, atunci pasul 3 se aplica iterativ, pana la eliminarea acestora.

Intregul process de proiectare poate fi reluat daca, pe parcurs, apar modificari majore la cerintele proiectului sau daca sunt identificate anomalii ce necesita reproiectarea modelului.

Modelarea unei probleme concrete. Studiu de caz

Descrierea problemei

Editura Poesis S.R.L. doreste sa realizeze o antologie de poezie romaneasca, de la Enachita Vacarescu si pana la poetii contemporani. Antologia va putea fi accesata prin intermediul retelei internet, fara nicio restrictie.

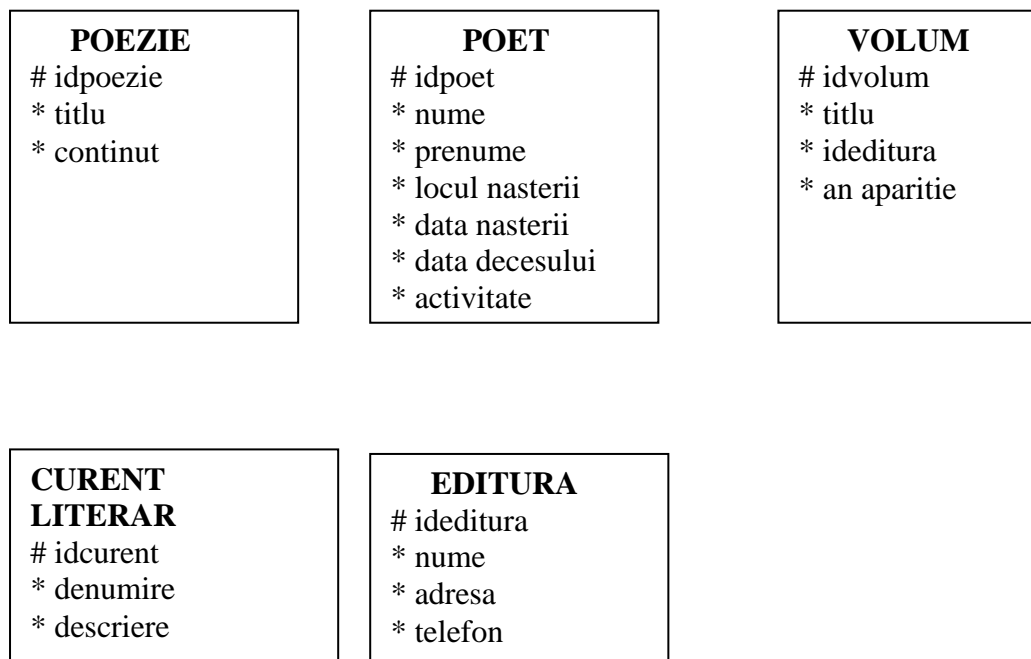
In urma intalnirilor cu reprezentantii editurii Poesis S.R.L. au rezultat urmatoarele cerinte privind modelul conceptual:

- aplicatia va memora date biografice referitoare la poetii inclusi in antologie, date despre fiecare poezie publicata, precum si despre volumele de versuri tiparite antum sau postum;
- poeziile vor fi clasificate in functie de curentul literar din care fac parte; se vor furniza si informatii privind editurile care au publicat volumele de versuri.

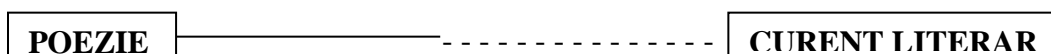
Identificarea entitatilor, atributelor si relatiilor

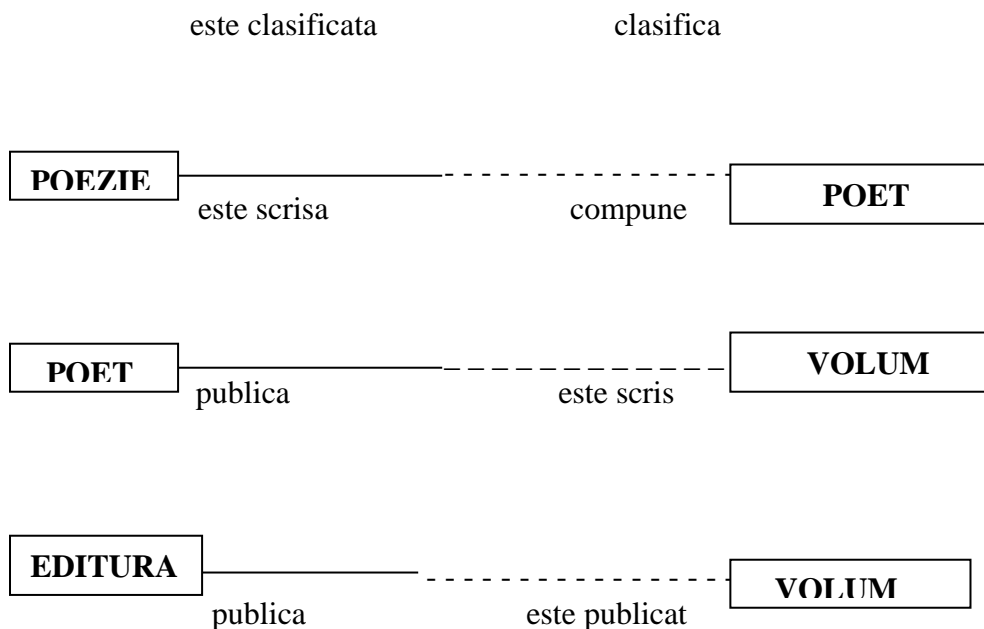
Pe baza descrierii problemei, identificam urmatoarele entitati necesare modelului nostru conceptual: POEZIE, POET, VOLUM, EDITURA, CURENT LITERAR.

Entitatile vor avea urmatoarele atribute:



Identificam relatiile dintre entitati:





Inlocuim cele doua relatii de tip many-to-many cu relatii one-to-many, stabilite prin intermediul a doua entitati de legatura si reprezentam grafic diagrama entitate-relatie. (vezi figura 1.8)

Structura conceptuala din figura de mai jos este in a treia forma normala.

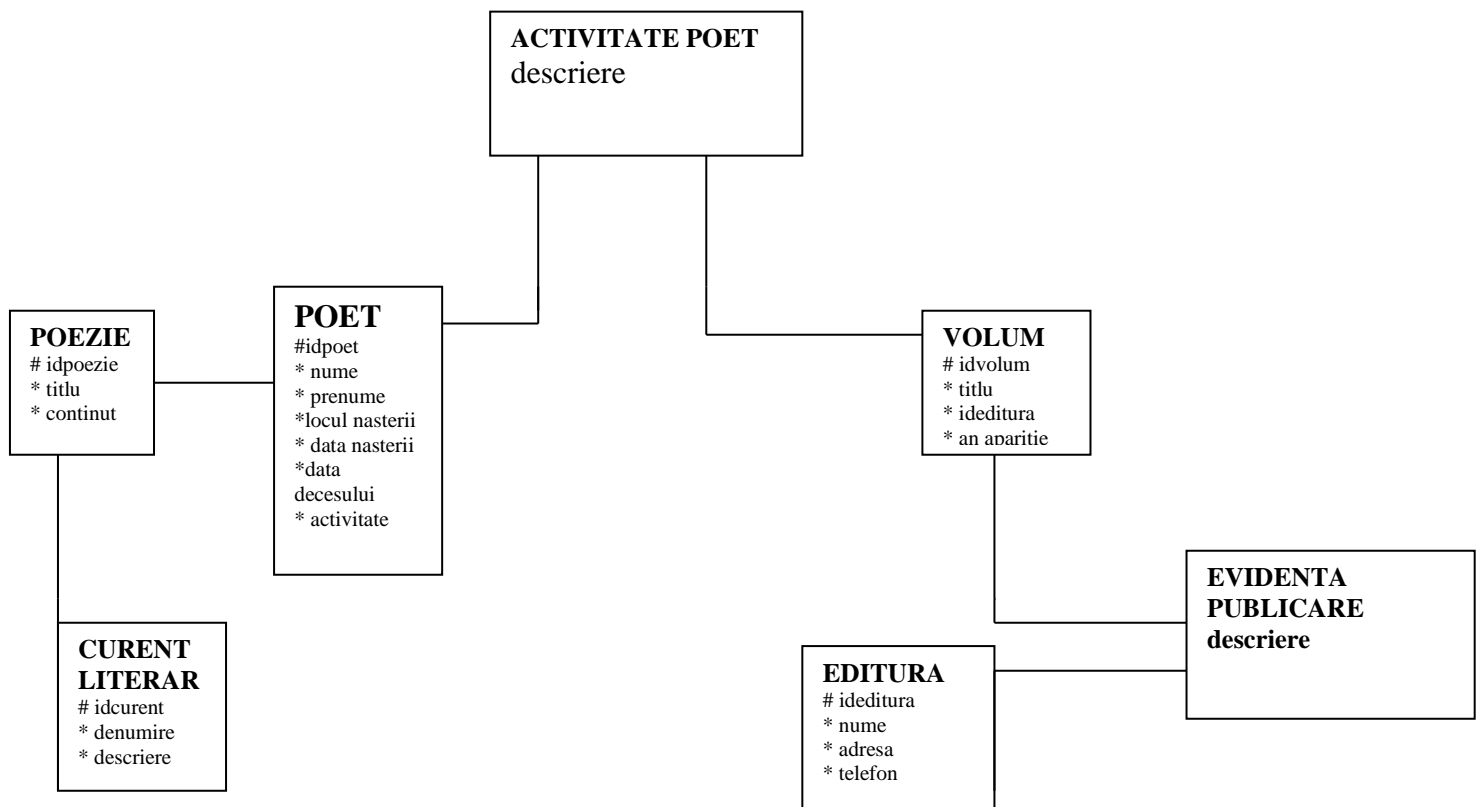


Figura 1.8 Diagrama entitate-relatie.

APLICATII

A.Itemi de tip grila

1. Orice atribut al unei entitati trebuie sa aiba o valoare unica.
 - a. adevarat
 - b. fals
2. O entitate nu trebuie sa aiba, obligatoriu, o cheie primara.
 - a. adevarat
 - b. fals
3. Orice entitate trebuie sa aiba cel putin un atribut optional.
 - a. adevarat
 - b. fals
4. Valoarea NULL reprezinta o valoare 0 sau un sir de caractere vid.
 - a. adevarat
 - b. fals
5. Un identificator unic poate fi o combinatie de attribute.
 - a. adevarat
 - b. fals
6. Care dintre urmatoarele substantive reprezinta o entitate?
 - a. culoare
 - b. autoturism
 - c. consum
 - d. an fabricatie
7. Care dintre urmatoarele cuvinte reprezinta un atribut?
 - a. denumire
 - b. placa de retea
 - c. mouse
 - d. hard-disc
8. Care dintre urmatoarele attribute pot fi alese drept unic identificator pentru entitatea STUDENT?
 - a. nume
 - b. prenume
 - c. CNP (cod numeric personal)
 - d. Profil
 - e. Facultate
 - f. CI (seris si numarul cartii de identitate)
9. Considerand entitatea PRODUS, descrisa de attributele idprodus, denumire si prît, ce reprezinta tripletul de valori (1202, televizor, 1200)?
 - a. tabela de valori
 - b. instanta a entitatii PRODUS
 - c. o relatie
 - d. o entitate.

B.Probleme

1. Identificati entitatile si attributele caracteristice pentru urmatoarele probleme de gestiune a datelor:

- a. Activitatea unui centru de inchiriere a masinilor.
- b. Gestionarea datelor referitoare la pacientii unui cabinet de medicina a familiei.
- c. Gestionarea datelor la examenul de bacalaureat.
- d. Activitatea unui deposit de produse agro-alimentare.

2. Identificati relatiile care se pot stabili intre urmatoarele perechi de entitati:
 - a. ORAS si JUDET
 - b. DIRIGINTE si CLASA
 - c. MATERIE si CLASA
 - d. PRODUS si FURNIZOR
 - e. CLIENT si CONT(cont bancar)
 - f. CLIENT si COMANDA.
3. Desenati diagrama entitate-relatie pentru fiecare din relatiile de la exercitiul precedent.
4. Pentru fiecare din relatiile de la exercitiul 2, aplicati algoritmul din exemplul 1.4, pentru "citirea" acestora de la ambele capete ale relatiei. Remarcati optionalitatea si cardinalitatea fiecarei relatii.
5. Pentru fiecare relatie de tip many-ti-many identificata la exercitiu 2, determinati entitatea de intersectie si redesenati diagrama entitate-relatie.
6. Consideram urmatoarea diagrama entitate-relatie(incompleta), care modeleaza datele dintr-o agentie de turism pentru tineret.

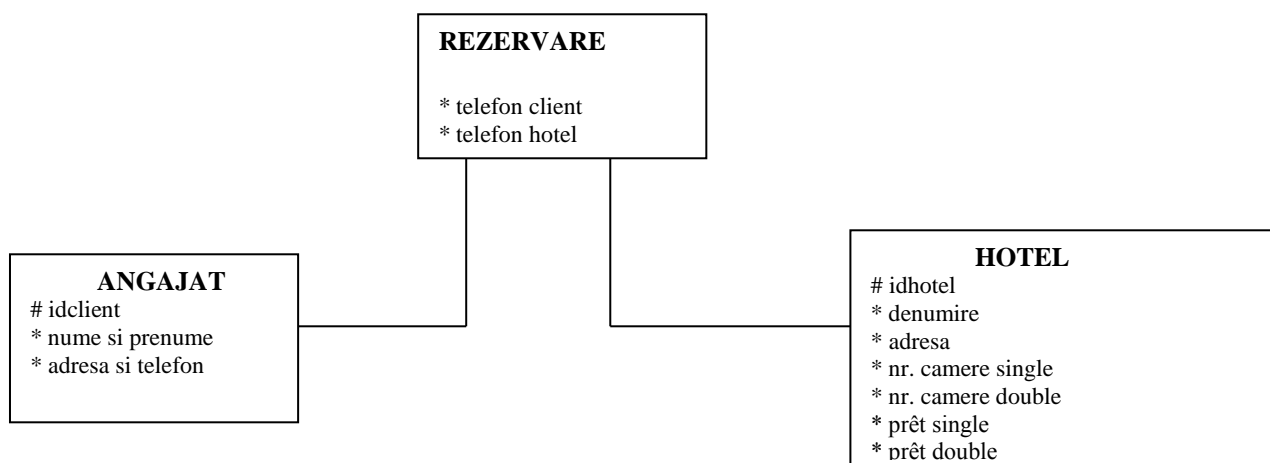


Diagrama din figura prezinta un model simplificat al datelor vehiculate intr-o agentie de turism, modelul nostru dorind sa surprinda activitatea de realizare a rezervarilor camerelor pentru clientii agentiei.

Precizari

- se doreste memorarea datelor referitoare la clienti(numa, prenume, adresa si alte date de identificare), la hotelurile partenere(denumire, adresa, preturi pentru diverse tipuri de camere), precum si memorarea rezervarilor realizate;
- Presupunem ca hotelurile partenere au numai camere de doua tipuri (single-cu un pat si double-cu doua paturi), dar preturile acestora sunt diferite in sezonul de vara fata de sezonul de iarna;
- Pentru rezervari se vor memora datele referitoare la clientul care face rezervarea, hotelul ales, perioada si numarul de zile.

Cerinte:

- Identificati attributele care lipsesc fiecarei entitati;

- Identificati cardinalitatea si optionalitatea relatiei dintre entitati;
- Identificati si eliminate abaterile modelului de la primele trei forme normale;
- Reprezentati diagrama entitate-relatie care satisface a treia forma normal.

C.

Miniproiect

Liceul vostru doreste realizarea unui catalog virtual care sa permita gestionarea urmatoarelor date:

- Datele personale ale elevilor si cadrelor didactice;
- Date referitoare la clasele, profilurile si specializarile din liceu;
- Memorarea notelor obtinute de fiecare elev la fiecare materie din planul de invatamant corespunzator profilului clasei din care face parte elevul;
- Posibilitatea calcularii si memorarii mediilor, precum si obtinerea unor statistici la nivel de elev, clasa, an de studiu, scoala.

Construiti modelul conceptual al problemei de gestiune descries prin scenariul de mai sus, urmand pasii:

- Identificati entitatile si attributele care le descriu;
- Identificati relatiei dintre entitati;
- Reprezentati modelul utilizand diagram entitate-relatie;
- Eliminati relatiile many-to-many si redesenati diagrama entitate-relatie;
- Identificati si corectati abaterile de la prima forma normala;
- Identificati si corectati abaterile de la a doua forma normala;
- Identificati si corectati abaterile de la a treia forma normala;
- Redesenati diagrama entitate-relatie in a treia forma normala.