

### 13 .IMPLEMENTAREA STRUCTURILOR DE TIP RETEA SIMPLA

In structurile arborescente, fiecare nod părinte poate avea mai mulți fii, iar fiecare nod fiu are un singur părinte. Din acest motiv modelele de BD ierarhice nu permit legături orizontale între entitățile A și B de pe același nivel ierarhic (fig.1), sau referințe de la doua entități A și B spre aceeași entitate C (fig.2).



Consideram gestiunea unei biblioteci universitare, care conține un fișier de CARTI si unul de STUDENTI (cititori) (fig.3)

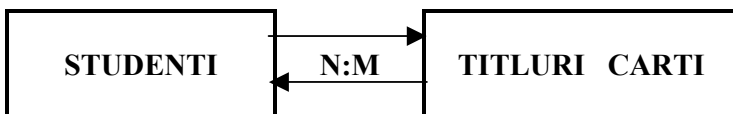


Fig.3

Un student poate împrumuta de la biblioteca mai multe cărți. Fiecare titlu de carte poate exista în mai multe exemplare, cu număr de inventar distinct (NRI), care pot fi împrumutate la studenți diferiți. Aceste legături sunt de tipul rețea simplă (N:M).

Pentru implementarea acestei BD vom folosi structura din Fig.4, care transforma legaturile N:M in 2 legaturi mai simple 1:N, care se pot implementa prin referințe si liste înlănțuite.

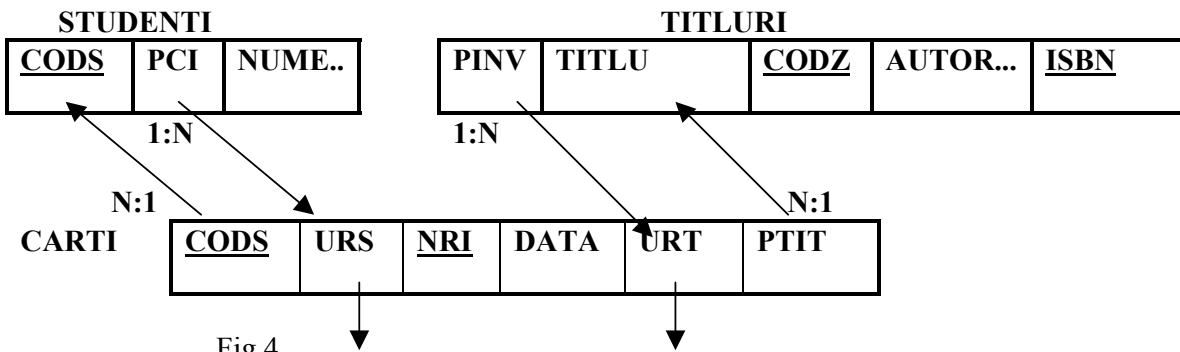


Fig.4

Fișierul STUDENTI este indexat după CODS și NUME. El refera prin pointerul PCI (pointer cărți împrumutate) prima carte din lanțul de cărți împrumutate de un student, având pointerul de înlănțuire URS. Cărțile se identifica prin numărul de inventar NRI, după care fișierul este indexat. Cărțile ne împrumutate, au câmpurile CODS si DATA completate cu spatii, iar cele împrumutate conțin codul studentului și data când s-a împrumutat cartea.

Fișierul TITLURI conține datele comune pentru toate cărțile cu același titlu (titlu, autor, cod ISBN, nr. pagini, cod limba, ..). Pointerul PINV (pointer nr. inventar) referă prima carte din lanțul cărților cu acel titlu, care se înlănțuie între ele prin pointerul URT. Pointerul PTIT face o referință de la o carte identificată prin NRI, spre înregistrarea care conține titlul

cărții. Toți pointerii folosesc numărul înregistrării referite din fișier, și utilizează **-1 pentru a indica sfârșitul de lanț.**

**Codul zecimal de clasificare CODZ** servește la identificarea titlurilor după domeniu și poate fi cheie de indexare a fișierului pentru a găsi toate cărțile dintr-un domeniu mai larg sau mai îngust. Acest cod consideră 10 domenii mari fiecare împărțit în 10 subdomenii, care la rândul lor se împart în 10 sub-subdomenii, s.a.m.d.

**Codul ISBN** servește la și cod unic de identificarea unui titlu de carte și conține: Codul țării, codul editurii, codul colecției, codul cărții în colecție, etc. Pentru a comanda un titlu de carte la o librărie electronică este suficient să se specificie ISBN-ul, fără alte amănunte.

In fig.5 se prezintă structura BD si modul de înlănțuire:

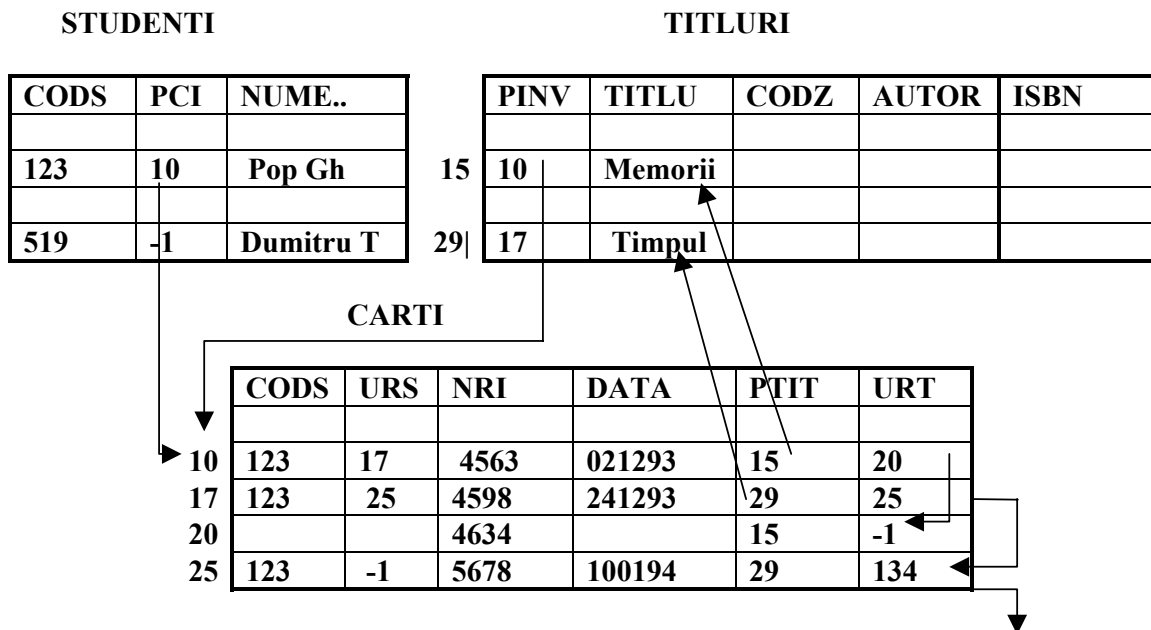


Fig.5

Prezentam mai jos programul principal si procedurile:

```
*****
* PBIBL * Program gestiune biblioteca
* I.Jian 2004
*****
set exact on
set talk off
clear
mes2='Selectati cu sageti sau prima litera '
defi popup b1 from 5,5 to 15,40 mess mes2
defi bar 1 of b1 prompt 'OPERATII CU CARTI' skip
defi bar 3 of b1 prompt 'Adaugare titluri '
defi bar 4 of b1 prompt 'Imprumutare carti '
defi bar 5 of b1 prompt 'Carti imprumutate '
defi bar 6 of b1 prompt 'Restituire carti '
defi bar 7 of b1 prompt 'Terminat '
on select popup b1 do pselect
```

```

activ popup b1
return                                && **sfirsit program principal**
*****
PROCEDURE PSELECT
DO CASE
    case bar()=3
    do atitl
        case bar()=4
        do cauts                                && Imprumut carti unui student
            case bar()=5
            do cauts                                && Afisare carti imprumutate
                case bar()=6
                do cauts                                && Restituire carti
                    case bar()=7
                    close all
DEACTIV POPUP
return
ENDCASE
RETURN

```

```

*****
* CAUTS * Cautare un student dupa Cods
*****

```

```

PROCEDURE CAUTS
SELECT 1
use stud index icods alias st
use carti index inri in 2 alias ca
use titluri in 3 alias tl
SELECT 2
set relation to ptit into tl
SELECT 1
r1='d'
    DO WHILE r1 $ 'DdYy'
clear
vcod =space(5)
@ 1,1 say 'Cod student ' get vcod
read
vcod =trim(vcod)
seek vcod
if eof()
    ?'Studentul cod: '+vcod+' nu exista '
    wait
    loop
endif
DO CASE
case bar()=4
set exact on
do imprumut                                && imprumut carti
case bar()=5
set exact off

```

```

do afis_c                                && afisare carti imprumutate
  case bar()=6
  set exact on
do rest_c                                && restituiere carti
  ENDCASE
SELECT 1
wait 'Continuati pt. alt student ?d/n ' to r1
  ENDDO

clear
close all
set exact off
RETURN                                && * sfirsit CAUT_S

* *****
* IMPRUMUT * imprumut carti
* *****

PROCEDURE IMPRUMUT
R2='d'
  DO WHILE r2 $ 'DdYy'
clear
* n=pci                                && pointer carti imprumutate
select 2                                && fisier carti (volume)
vnri=0
@ 2,2 say 'Nr.inventar carte: ' get vnri
read
* seek vnri                            && cautare carte
* if eof()
* Se considera numarul de inventar NRI egal cu numarul inregistrarii din fisierul Carti
  if vnri >recc() .or.vnri <0
  ? 'Nr.inventar: ', vnri, ' inexistent'
wait
return
  endif
go vnri
  if cods # space(5)
  ? 'Cartea e imprumutata studentului cods: ',cods
d1=seek(cods,1)                        && cautare student
  ? st->nume,' adresa: ',st->adresa
wait
return
  endif
repl urs with st->pci,cods with st->cods,data with date()
repl st->pci with recno()              && actualizare pointeri
  ? 'S-a inregistrat ',nri,tl->titlu,tl->autor
  ? ' pt.studentul: ',st->nume,' cods:',st->cods
wait 'Mai sint carti pentru acelasi student? d/n ' to r2
  ENDDO
RETURN                                && * sfirsit imprumut carti

```

```

*****
* AFIS_C * afisare carti imprumutate
*****
PROCEDURE AFIS_C
    DO WHILE cods =vcod      && se pot afisa pe grupuri
clear
@ 2,0 say 'Studentul '+Nume +'cods: '+cods +' adr: '+adresa
    if pci=-1
@ 3,5 say 'Nu are carti imprumutate! '
wait
skip
loop
    endif
@ 3,5 say 'A imprumutat cartile: '
n=pci                      && pointer spre carti
select 2                    && fisierul carti
    DO WHILE n # -1
go n                        && pozitionare pe carte
* afisare titluri carti imprumutate
? nri, ' ,tl->titlu, ' ,tl->autor
n=urs                      && urmatoarea carte ENDDO
select 1
wait
skip                        && urmatorul student din grup
    ENDDO
RETURN                      && * sfirsit AFIS_C

```

```

*****
* ATITL * adaugare titluri carti
*****
PROCEDURE ATITL
use carti in 2              && fisier carti
use titluri in 3           && fisier titluri carti
SELECT 3
r1='d'
    DO WHILE r1 $ 'DdYy'
clear
append blank
@ 1,10 say 'ADAUGARE TITLURI CARTI'
@ 3,2 say 'Titlul cartii: ' get titlu
@ 4,2 say 'Autori: ' get autor
@ 5,2 say 'Cod zecimal: ' get codz
@ 6,2 say 'Cod limba: ' get codl
read
n= recno()                 && Nr.inregistrare titlu adaugat
SELECT 2
* Se genereaza nr. de exemplare in fis. carti

```

```

go bottom
p=-1                && indicator urmatoarea inregistrare
ni=nri+1           && nr.inventar urmatore dupa ultima carte
ne=0               && contor nr.exemplare
@ 8,5 say 'Nr. exemplare: ' get ne valid ne>0
read
  DO WHILE ne>0    && generare numere inventar
append blank
repl nri with ni   && nr inventar urmatore
repl ptit with recno(3) && pointer spre titlul cartii
repl urt with p    && urmatoarea carte cu acelasi titlu
? 'Nr.inventar: ',nri && nr inventar generat
ni=ni+1
ne=ne-1           && contor cicluri -nr exemplare
p=recno()         && adresa inregistrare (precedenta)
  ENDDO
* repl tl->pinv with p && pointer cap de lant in fis.titluri
SELECT 3
repl pinv with p
wait 'Mai adaugati titluri? d/n' to r1
  ENDDO
wait 'Reindexati fisierul titluri? d/n ' to r1
  if r1$'DdYy'
index on codz to icodz
  endif
close all
clear
return
* sfirsit * ATITL

*****
* REST_C * Restituire carti imprumutate
*****
PROCEDURE REST_C
r2='d'
n=pci              && pointer prima carte imprumutata
  SELECT 2        && fisier carti
    DO WHILE r2 $ 'DdYy'
clear
  if st->pci=-1
? 'Studentul ',st->cods,' nu mai are carti imprumutate '
wait
return
  endif
input 'Nr.inventar carte: ' to vnri
n=st->pci          && prima carte
n1=1              && este primul din lant
gasit =.n.        && indicator articol gasit

```

```

DO WHILE n # -1          && parcurgere lant
go n
IF nri # vnri
preced=n                && articol precedent in lant
n1= n1+1                && nu e primul
n=urs                   && urmatorul articol
loop
ELSE
gasit =.y.
? ' Cartea ',Nri: ',nri,' RESTITUITA '
? tl->titlu,' autor: ',tl->autor
m=urs                   && salvare pointer urmator
repl cods with space(5), urs with 0  && initializari cimpuri
exit                    && terminare cautare
ENDIF
ENDDO
IF .not. gasit
? ' Nr. inventar: ',vnri,' gresit '
wait 'Continuati restituire pt. acelasi student? d/n ' to r2
loop
ENDIF
* cartea s-a gasit
IF n1=1                 && articolul sters este primul in lant
repl st->pci with m
ELSE
go preced
repl urs with m        && refacere lant
ENDIF
wait 'Continuati restituiri pt. acelasi student?d/n ' to r2
ENDDO
RETURN
* sfirsit REST_c
*****

```