

## 11. IMPLEMENTAREA STRUCTURILOR IERARHICE

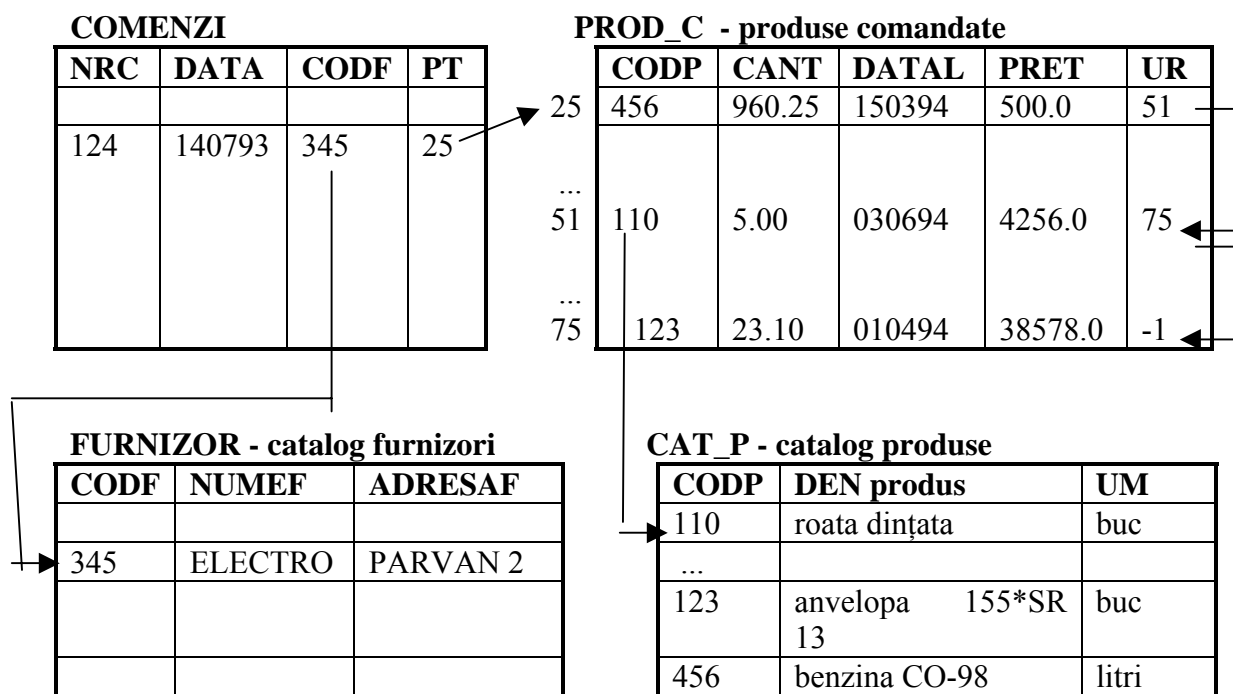
Daca un cod se regăsește în mai multe înregistrări și aceste înregistrări au o parte din câmpuri identice, (determinate de cod) pentru a reduce redundanță, se vor folosi doua fișiere legate între ele. Primul va conține setul de câmpuri ce au valori identice pentru un grup de înregistrări, iar al doilea câmpurile care diferă. Legătura dintre ele se poate realiza în doua moduri:

a) Adăugând în fișierul 2 codul înregistrării părinte din fișierul 1 și indexând fișierul 2 după acest cod;

b) Utilizând în fișierul 1 (master) un câmp pointer PT, care va indica numărul primei înregistrări asociate din fișierul 2. Toate înregistrările din fișierul 2 asociate aceleiași înregistrări părinte vor fi înlanțuite între ele printr-un pointer UR, care specifică numărul de ordine în fișier a următoarei înregistrări. Ultima înregistrare din lanț va avea UR=-1.

Ultima metoda este foarte rapida, eliminând cutarea prin fișierul index a articolelor asociate unui articol master, care poate dubla sau tripla timpul de acces la informații.

Pentru exemplificare vom considera o baza de date pentru evidenta comenzi, care va conține fișierele COMENZI, PROD\_C (produse comandate), CAT\_P (catalog produse) și FURNIZOR având structura:



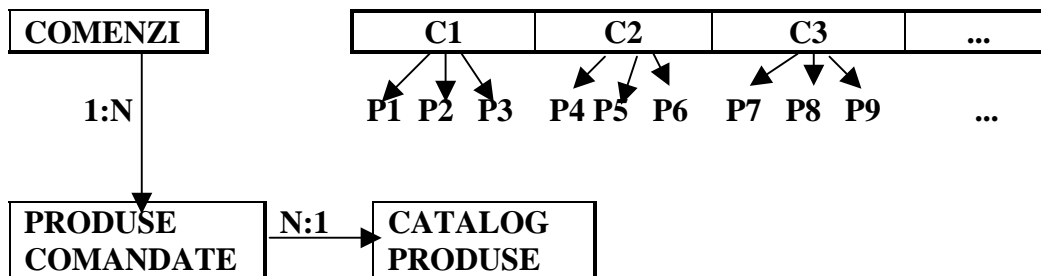
S-a notat: **NRC** - nr. comandă, **DATA** - data lansării comenzii, **CODF** - cod furnizor  
**NUMEF** - nume furnizor, **ADRESAF** - adresă furnizor  
**CODP** - cod produs, **CANT** - cantitate,  
**DATAL** - data livrării, **PRET** - preț unitar  
**DEN** - denumire produs, **UM** - unitate de măsură

Intre fişierul **PROD\_C** si **CAT\_P** se realizează o legătura automata prin cimpul **CODP** utilizând comanda SET RELATION TO, iar intre **COMENZI** si **FURNIZOR** prin câmpul **CODF**.

Pentru fişierul **PROD\_C** vom realiza recuperarea spaţiului eliberat prin ştergerea înregistrărilor si alocarea dinamica a acestui spaţiu noilor înregistrări adăugate. Pentru aceasta utilizam pointerul **UR** din prima înregistrare, care va fi cap de lanţ pentru înregistrările eliberate prin ştergere. Pentru toţi pointerii folosim -1 ca indicator sfârşit de listă. Înregistrările corespunzătoare produselor şterse, vor fi eliminate din lanţul asociat comenzii respective si vor fi adăugate in lanţul de libere. Adăugările de noi produse pentru o comanda se fac la începutul listei după algoritmul :

**SELECT PROD\_C** - selecţie fişier produse  
**APPEND BLANK** - adăugare produs nou la sfârşit  
 \* - se memorează pointerul cap de lanţ din comanda în **UR** din  
 \* înregistrarea adăugată (sistem stiva)  
**REPLACE UR WITH COMENZI->PT**  
 \* - pointerul **PT** din **COMENZI** va indica ultimul produs adăugat  
**REPLACE COMENZI->PT WITH RECNO()**

Această BD este de tip ierarhic (arborescent) cu legaturi 1:N intre **COMENZI** si **PRODUSE\_COMANDATE** si de tip referinţă (N:1) prin cheie spre **CATALOGUL PRODUSE**. Structura se poate reprezenta figurativ astfel:



Programul va conţine :

- Programul principal, care defineşte si activează meniurile
- Procedurile: **COMAND** - selectează comanda indicata prin număr (NRC)  
**ADAUGP** - adaugă produse la comanda selectata reutilizând spaţiul lanţului înregistrărilor şterse (LIBER)  
**AFISP** - afişează produsele de pe comanda selectata  
**STERGP** - şterge produsele indicate prin **CODP** de pe c-da recuperând spaţiul eliberat in lanţul LIBER

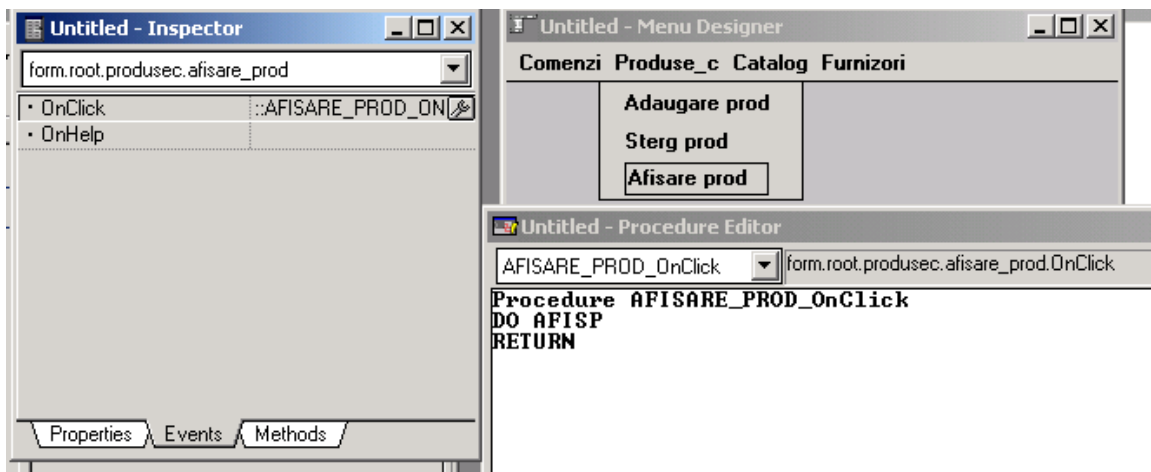
Restul procedurilor din meniuri referitoare la fişierele **COMENZI** si **CAT\_P**, sunt simple si nu s-au prezentat existând exemple de acest tip discutate anterior.

Pentru o mai buna înţelegere se dau meniurile, structurile si conţinuturile fişierelor utilizate, rezultatele obţinute din diferite proceduri pe ecran.

Se poate proiecta un meniu care sa conțină operațiile necesare grupate:

- Comenzi:** Adăugare comandă  
Modificare comandă  
Ștergere comandă  
Afișare date din comandă (mai puțin produsele comandate)
- Produse\_C** Adăugare produs pe comanda selectată  
Ștergere produs de pe comandă dat prin Codp  
Afișare produse de pe comanda selectată
- Catalog** Adăugare, Modificare, Ștergere, afișare produs dat prin Codp  
**Furnizori** Adăugare, Modificare, Ștergere, afișare furnizor dat prin CodF

Fiecărui element de meniu ise va asocia o procedură.



Meniul va fi asociat unui form **CForm** care va conține obiecte Text și EntryField pentru a afișa elementele comenzii selectate și unui produs adăugat pe comandă.

Form-ul va avea atașat pe evenimentul OnOpen o procedură care deschide toate fișierele bazei de date cu indecși corespunzători și stabilește legăturile între fișiere. Tot în această procedură se vor specifica câmpurile asociate din diferite fișiere pentru fiecare EntryField de pe Form.

S-au folosit fonturi de 10 și Bold pentru a se vedea clar.

Procedurile AdaugP, StergP, AfisP se vor modifica la afișare pentru a se folosi interfața grafică proiectată.

Pentru afișarea unei comenzi cu toate produsele de pe comandă se va folosi un form separat, în care se vor afișa datele comenzii. Produsele de pe comandă se vor copia într-un fișier de temporar și se vor afișa cu BROWSE.

Comenzi

5

Data:

NR.Comanda:  CodF:

Furnizor:

10

Cod produs  Cantitatea

Nume Produs

Pret

Untitled - Procedure Editor

Form\_OnOpen form.OnOpen

```

Procedure Form_OnOpen
Clear
Use Comenzi in 1 index Inrc alias CD
Use Prod_c in 2 alias PC
Use Cat_p in 3 index Icodp alias CP
Use Furnizor in 4 index Icodf alias FR
Set Relation To CodF in FR
select 2
Set RelationTo Codp in CP
select 1
form.Entryfield1='CD->data'
form.Entryfield2='CD->NRC'
form.Entryfield3='CD->CODF'
form.Entryfield4='FR->NumeF'
form.Entryfield5='PC->CodP'
form.Entryfield6='CP->Den_p'
form.Entryfield7='PC->Cant'
form.Entryfield8='PC->Pret'
Return

```

```

*****
* Program evidenta comenzi inlantuind produsele de pe comanda
* FISIERE : Comenzi, Prod_c (produse comandate)
* Cat_p (catalog produse)
*****

clear
set talk off
Clear All ; inchide toate fisierele sterge toate ferestrele
* definire popup produse pp *
defi popup pp from 5,5 to 15,35 mess mes2
defi bar 1 of pp prompt 'PRODUSE COMANDATE' SKIP
defi bar 2 of pp prompt repl(=',20) SKIP
defi bar 3 of pp prompt space(10) SKIP
defi bar 4 of pp prompt 'adaugare produse pe c-da'
defi bar 5 of pp prompt 'afisare produse de pe c-da'
defi bar 6 of pp prompt 'stergere produse din c-da'
defi bar 7 of pp prompt 'terminat'
ON SELECT POPUP pp do pselect
ACTIVATE Popup pp
Close Database
RETURN

* PCSELECT ** Procedura selectie popup produse *
PROCED pselect
do case
    case bar()=4
    do comand
    case bar()=5
    do comand
    case bar()=6
    do comand
    case bar()=7
    deact popup
    return

*****
* COMAND **Procedura cautare comanda dupa NRC -nr comanda
*****
PROCEDURE COMAND
clear
use comenzi in 1 index inrc alias cd
use prod_c in 2 alias pc
use cat_p in 3 index icodp alias cp
defi window fc from 0,0 to 3,79 && fereastra comenzi
defi window fp from 4,0 to 19,79 && fereastra produse comandate

```

```

activ window fc      && activare fereastra comenzi
r1='d'
DO while r1$'DdYy'
clear
vnrc=space(5)
@ 0,1 say 'Nr.comanda: ' get vnrc mess 'Dati nr.comanda'
read
seek vnrc      && cauta comanda in fisierul de comenzi
if eof()
@ 1,0 say 'Nr.C-DA inexistent!! ' + Vnrc
wait 'Continuati ? d/n ' to r1
loop
endif
@ 1,1 say 'Furnizor: ' + codf
@ 1,25 say 'Data: ' + dtoc(data)

* Selectie operatie ceruta
activ window fp      && fereastra produse
do case
case bar()=4          && adaugare produse pe comanda
do adaugp
case bar()=5          && afisare produse de pe comanda
do afisp
case bar()=6
do stergp
other
@ 0,3 say 'functie inexistent!'
wait 'Continuati?d/n ' to r1
loop
endcase
***** Revenire din proceduri produse
select 1              && zona fis comenzi
deactiv window fp    && sterge fereastra produse
clear
@ 0,3 say 'Continuati cu alta comanda ' get r1
read
ENDDO
deactiv wind fc      && sterge fereastra comanda
close all            && inchide toate fisierele
RETURN
***sfirsit procedura comand
*****

```

**\*ADAUGP \* procedura adaugare produse pe comanda selectata**

\* se aloca dinamic peste inregistrarile sterse

\*\*\*\*\*

**PROCED ADAUGP**

sele 2

**go 1**                    && pointer libere

**liber=ur**

r2='d'

**DO WHILE r2 \$ 'dDYy'**

clear

vcdp=space(5)

@ 1,1 say 'Cod produs: ' **get vcdp**

read

**da=seek(vcdp,3)**      && verificare existenta produs

  if .not. da

? 'Nu exista codul produs '+ vcdp styl 'i'

? 'Actualizati catalogul de produse ' styl 'i'

wait 'Continuati ? d/n ' to r2

loop

**endif**

@ 1,30 say cp->den                    && se ia numele produsului din catalog

  if liber #-1                    && exista inregistrari libere

**go liber**                    && alocare dinamica pe o inregistrare libera

**liber=ur**                    && corectie lant de inregistrari libere

  repl cant with 0, pret with 0      && sterge vechile valori

  else                    && nu exista libere

**APPEND BLANK**                    && adaugare inregistrare noua daca nu exista libere

**endif**

**repl cdp with vcdp, datal with date()**      && completare inregistrare

@ 2,1 say 'Cantitate comandata:' **get cant**

@ 3,1 say 'Pret unitar: ' **get pret**

@ 4,1 say 'Data livrarii: ' **get datal**

read

\* \*actualizare pointeri

**repl ur with cd->pt**

**repl cd->pt with recno()**

wait 'Mai adaugati produse? d/n ' to r2

**ENDDO**

**go 1**                    && memorare pointer libere pe prima inregistrare

**repl ur with liber**

return

\*\*sfirsit procedura ADAUGP

\*\*\*\*\*

**\* AFISP \*\* procedura afisare produse de pe comanda**

\*\*\*\*\*

**PROCEDURE AFISP**

sele 2

**set relation to codp into cp** && spre catalog

n=cd->pt

if n=-1

wait 'Nu exista produse comandate'

return

endif

clear

nl=2 && contor linii afisare

\* Afisare cap de tabel

@ 0,0 say 'CODP DENUMIRE CANTITATE PRET VALOARE '

@ 1,0 say repl('=',79)

t=0 && valoare totala pe comanda

**DO while n # -1**

**go n** && pozitionare pe inregistrare (cd->pt prima)

\* S-a pozitonat si pe produsul din catalog cu codul CODP

if nl<15

\* Afisare o linie de produs de pe comanda si cu datele de catalog

@ nl,0 say 'I '+codp+' I '+cp->den+' I '+str(cant,7,2)+' I '+str(pret,7,2)+;  
' I '+str(pret\*cant,10,2)+' I '

nl=nl+1

**n= ur** && adresa urmatorului produs de pe comanda

else

wait

nl=2 && ecran nou

@ 0,2 clear && sterge ecran fara cap tabel

endif

**t=t+pret\*cant** && total comanda

**ENDDO**

@ nl,0 say repl('=',78)

@ nl+1,29 say 'TOTAL VALOARE COMANDA '+str(t,12,2)

wait

return

\*\* sfirsit procedura AFISP

\*\*\*\*\*



**\* STERGP \* Stergere produse de pe comanda \***  
 \* inregistrările sterse se atasează listei de libere (UR din record 1)  
 \*\*\*\*\*

**PROCEDURE STERGP**

**SELECT 2**

```

go 1
liber=ur      && cap liste inregistrari libere
n=cd->pt
  if n=-1
? 'Nu exista produse pe comanda'
wait
return
  endif
r2='d'
  DO WHILE r2 $ 'DdYy'
c=1           && indicator prima din lant
gasit =0     &&indicator produs negasit
n=cd->pt     && inceput lant
clear
vcodp=space(5)
@ 1,1 say 'Cod produs de sters' get vcodp
read
  DO WHILE n # -1
go n         && urmatorul articol
  IF .not. codp=vcodp
ultim=n     && nr. articol anterior in lant
n=ur
c=c+1       && nu e primul produs
loop
  ELSE
gasit=1     && s-a gasit produsul cu vcodp
disp
? 'Inregistrarea s-a sters'
m=ur       && salvare pointer
repl ur with liber && se adauga la lantul de libere
liber=recno()
gasit =1
  exit
  ENDIF
  ENDDO
  IF gasit =0
? 'Nu exista produsul pe comanda'
wait ' Continuati stergerile? d/n ' to r2
loop
  ENDIF

```

```
* Produsul s-a gasit
IF C=1      && articolul sters a fost primul
repl cd->pt with m && se modifica capul de lant
  ELSE
go ultim      && revenire la precedentul articol din lant
repl ur with m && actualizare lant
  ENDIF
wait 'Continuati stergeri? d/n' to r2
  ENDDO
go 1      && memorare cap lista libere
repl ur with liber
return
** sfirsit STERGP
*****
```