

1. Tabele de date

FoxPro este un mediu de dezvoltare integrat, care, pe lângă instrumentele de programare, oferă și un sistem de tabele de date relaționale care simplifică procesele de gestionare și organizare eficientă a datelor. Datele sunt organizate tabelar, pe *linii (records, articole înregistrări)* și *coloane (fields, câmpuri)*, mai multe tabele, între care există legături, alcătuind o bază de date relațională.

🗐 1.1. Crearea unui tabel de date – definirea structurii

Crearea unui tabel de date implică specificarea numelui tabelului (fișier cu extensia implicită DBF) și este inițiată prin comanda

CREATE [<fişier>]

sau prin alegerea opțiunii *<u>F</u>ile, <u>N</u>ew* (în acest caz în caseta <u>**N**ew</u> se selectează <u>*T*</u>able și <u>**N**</u>ew file). Ca urmare pe ecran este deschisă fereastra constructorului de tabele, în care vor fi precizate caracteristicile noului tabel (nume, tip, lungime, etc.) și a indecșilor folosiți la ordonarea datelor din listă.

Fiecare **câmp** (field, coloană) al unui tabel este caracterizat prin:

- nume (Name);
- **tip** (*Type*) șir de caractere, numeric, dată calendaristică sau moment de timp, logic, memo;

 lăţime (Width) - numărul de caractere pe care îl ocupă fiecare dată memorată în acel câmp;

 numărul de zecimale (Decimal) - pentru datele de tip r 	numeric:
--	----------

b	🖉 Т.	able Designer	- personal.dbf						×
	Fiel	ds Indexes	Table						
		Name	Туре		Width	Decimal	Index	NULL	
		nume	Character		25		t	<u> </u>	ОК
		marca	Character		8				Canaal
		functia	Character		20				
		salariu	Numeric		8	0			
		datanast	Date		8				
		sex	Logical		1				<u>D</u> elete
	ŧ	adresa	Memo	•	4		-		
	Г							_	
			_					-	

- fanionul de indexare (Index) indică dacă se stabileşte un index pe câmpul respectiv (indecşii reprezintă o tehnologie cu ajutorul căreia datele din tabele sunt văzute într-o anumită ordine, dată de un anumit criteriu numit criteriu de indexare);
- fanionul de valoare nulă (NULL) arată dacă în câmpul respectiv poate fi memorată o valoare nulă, pentru a se face diferenţierea între un câmp lăsat necompletat şi unul completat cu valoarea 0.

1.2. Deschiderea (activarea) tabelelor de date

Pentru iniţierea unei sesiuni de lucru cu un tabel de date, acesta trebuie să fie activat (deschis). Deschiderea unui tabel de date se poate realiza prin **meniu** (*<u>F</u>ile, <u>O</u>pen, Files of <u>t</u>ype: Table (*.dbf))* sau prin comanda USE:

```
USE [<fişier>] [IN <zona de lucru>] [TAG <nume etichetă>]
[ASCENDING | DESCENDING]
[ALIAS <alias>] [EXCLUSIVE] [NOUPDATE]
```

- <fișier> numele tabelului de date care va fi deschis (extensia implicită DBF);
- *IN* permite deschiderea unui tabel de date într-o altă zonă de lucru decât cea curentă;
- TAG face precizarea unei variante de indexare;
- ASCENDING | DESCENDING precizează tipul ordonării (implicit ascendentă) atunci când se foloseşte indexarea;
- *ALIAS* stabileşte un nou *alias*;
- EXCLUSIVE asigură exclusivitatea folosirii unui tabel de date atunci când se lucrează într-o rețea de calcul;
- NOUPDATE protejează tabelul de date la scriere (este deschis doar pentru consultare).
- 👃 Observații
- Derularea unei sesiuni de lucru în FoxPro se realizează în directorul de lucru implicit. Acest director poate fi schimbat prin comanda:

SET DEFAULT TO <path> unde *path* defineste noul director.

 Lista tabelelor de date aflate în directorul de lucru poate fi obţinută prin lansarea comenzii:

DIR

 Se poate limita, temporar, accesul la înregistrările tabelului de date prin folosirea comenzii:

SET FILTER TO <cond>

Zone de lucru

Pentru manipularea simultană a mai multor tabele de date, FoxPro folosește zone speciale de memorie numite *"zone de lucru"*, în care depune informațiile necesare efectuării operațiilor cu tabelele de date respective.

Pentru identificarea unei zone de lucru se folosesc *alias-uri* (literele A, B, C ... sau cifrele 1, 2, 3, ...) care permit identificarea tabelelor de date și altfel decât prin numele lor. În afară de aceste alias-uri utilizatorul poate defini și propriile sale alias-uri.

👃 Observații

 La pornirea FoxPro-ului zona de lucru implicită va fi zona 1 (cu alias 1 sau A). Aceasta se poate schimba prin comanda :

SELECT <expN> | <expC>

Comanda este urmată de o cifră, literă sau de un alias definit de utilizator. Dacă alias-ul este 0 se va selecta prima zonă de lucru neselectată.

 Pentru a obţine informaţii despre zonele de lucru folosite şi tabelele de date deschise se pot folosi două funcţii:

USED([<alias>])

şi **DBF([<alias>])**

Funcția *USED* returnează *.T.* sau *.F.* dacă, în zona de lucru curentă sau precizată prin alias, este sau nu prelucrat un tabel de date. Funcția *DBF* returnează numele tabelului de date deschis în zona de lucru.

Exemple:

```
use personal in 1
use nomenclator in B alias nomen
select nomen
?dbf()
C:\PROGRAM
FILES\DEVSTUDIO\VFP\JOB\NOMENCLATOR.DBF
?used(1)
.T.
```

1.3. Modificarea structurii unui tabel de date

Modificarea caracteristicilor unui tabel de date poate fi făcută prin comanda:

MODIFY STRUCTURE

Această comandă permite modificarea structurii tabelului de date activ prin deschiderea unei ferestre de dialog, aceeaşi ca la crearea tabelului de date. Modificările care pot fi aduse structurii unui tabel de date sunt :

- schimbarea numelor, tipurilor şi lăţimii câmpurilor deja existente;
- ştergeri, inserări de câmpuri, prin plasarea cursorului pe zona de modificare (prima coloană din fereastra de dialog) și acționarea butoanelor **Delete** și **Insert**.

După efectuarea modificărilor, FoxPro copiază conţinutul tabelului de date vechi în tabelul de date nou, cu efectuarea schimbărilor respective.

Caracteristicile unui tabel de date (structura) pot fi vizualizate prin comanda:



🗐 1.4. Închiderea prelucrării tabelelor de date

Finalizarea operației de prelucrare a tabelelor se poate face prin comenzile:

<u> </u>			
USE	sau	CLOSE ALL	
		•	

Prima comanda închide prelucrarea tabelului de date din zona de lucru curentă iar a doua închide și eliberează toate zonele de lucru (în acest caz zona curentă devine zona de lucru 1).

Un grup de înregistrări (domeniu) poate fi:

- All semnifică toate înregistrările tabelei;
- Next următoarele înregistrări;
- Record înregistrarea cu numărul de ordine specificat;
- Rest restul înregistrărilor din poziția curentă până la sfârșitul tabelului.

Prelucrarea tabelelor de date se realizează la nivel de înregistrare. La un moment dat este activ un singur tabel de date și o singură înregistrare din acest tabel. Fiecare înregistrare are un număr de ordine, asociat acesteia la introducerea ei în tabelul de date. Numărul de ordine al înregistrării curente poate fi modificat prin comenzile:

```
GOTO TOP | BOTTOM | RECORD <expN>
```

(salt la prima, ultima sau înregistrarea precizată din tabel);

SKIP [±<expN>]

(salt înainte sau înapoi cu un număr de înregistrări);

```
LOCATE [<domeniu>] [FOR <cond1>] [WHILE <cond2>]
```

(căută prima înregistrare care respectă condiția <cond1>, domeniul înregistrărilor fiind dat de clauzele <domeniu> și *WHILE*).

Poziția curentă în tabelul de date poate fi stabilită cu ajutorul funcțiilor:

- EOF() returnează .T. sau .F. după cum este detectat sau nu marcajul de sfârşit de fişier (aflat după ultima înregistrare);
- BOF() returnează .T. sau .F. pentru început de fișier;
- RECNO() furnizează numărul de ordine al înregistrării curente.

Exemple:

use personal
?recno()
1
goto record 7
?recno()
7
goto top
?recno()
1
skip –1
?bof()
.т.
skip 3
?recno()
3
go bottom
skip
?eof()
.т.
locate for nume='ALEX'

Introducerea înregistrărilor noi într-un tabel de date

Numărul de înregistrări dintr-o bază de date, furnizat de **funcția RECCOUNT()**, poate fi modificat, prin introducerea unor înregistrări vide (adăugări, inserări):

APPENDBLANKsau**INSERT**BLANK [BEFORE]

Inserarea se execută implicit după înregistrarea curentă, dacă nu se folosește opțiunea *BEFORE*, iar următoarele înregistrări sunt translatate spre sfârșitul tabelului de date și renumerotate.

🗊 Ştergerea înregistrărilor dintr-un tabel

Eliminarea unor înregistrări din tabelele de date se efectuează la nivel *logic* sau *fizic* prin utilizarea comenzilor *DELETE, RECALL, PACK, ZAP*.

 a) Ştergerea logică (marcare pentru ştergere). Prin marcarea pentru ştergere înregistrările din tabelele de date nu sunt eliminate definitiv. Pentru ştergerea logică este utilizată comanda:

DELETE [<domeniu>] [FOR <cond1>] [WHILE <cond2>]

Această comandă (precum și altele) folosește clauza *FOR* pentru selectarea tuturor înregistrărilor care corespund condiției logice <cond1>. Clauza *WHILE* întrerupe testarea înregistrărilor când găsește o înregistrare ce nu respectă condiția dată, <cond2>.

Accesul la înregistrările marcate pentru ștergere este controlat de comanda:

SET DELETED ON | OFF

Când se alege opțiunea *ON*, înregistrările marcate pentru ștergere nu vor fi accesibile celorlalte comenzi, iar când se selectează opțiunea *OFF*, înregistrările sunt accesibile indiferent de marcajul de ștergere (starea implicită este *OFF*). Această comandă este utilă atunci când se lucrează pe grupuri de înregistrări.

b) Demarcarea ștergerii logice. Marcajele de ștergere logică pot fi anulate prin folosirea comenzii:

RECALL [<domeniu>] [FOR <cond1>] [WHILE <cond2>]

c) Ştergerea fizică. Prin folosirea acestui tip de ştergere înregistrările sunt eliminate definitiv din tabelul de date (ştergerile fizice nu mai pot fi refăcute). Ştergerile fizice au efect asupra înregistrărilor marcate logic sau asupra tuturor înregistrărilor din tabelul de date prelucrat prin utilizarea uneia din comenzile:

PACK

sau

ZAP

```
🗊 Modificarea conținutului înregistrărilor
```

Modificarea conţinutului înregistrărilor dintr-un tabel de date presupune înlocuirea valorilor din unul sau mai multe câmpuri cu altele noi. Comanda *REPLACE* permite actualizarea tabelelor de date cu valori precizate în comandă și are următoarea sintaxă:

```
REPLACE <câmp1> WITH <exp1> [, <câmp2> WITH <exp2> ...]
[<domeniu>] [FOR <cond1>] [WHILE <cond2>]
```

REPLACE înlocuiește vechea valoare din câmpul <câmp1> cu valoarea expresiei <exp1>, valoarea din câmpul <câmp2> cu <exp2> și așa mai departe.

Exemple:

use personal
<pre>?reccount()</pre>
25
append blank
<pre>?reccount()</pre>
26
goto 7
insert blank before
<pre>?recno()</pre>
7
dele next 3
skip -1
recall
pack
?reccount()
24
replace for nume='ALEX' nume with 'ALEXANDRU

Fereastra BROWSE

Principala comandă, pentru realizarea diferitelor operațiuni asupra tabelelor de date, prin deschiderea unei ferestre de lucru, este comanda *BROWSE* care poate fi editată în următoarea sintaxă:

```
BROWSE [FIELDS <listă de câmpuri>] [FOR <cond>]
[FREEZE <câmp>]
[NOAPPEND] [NODELETE] [NOEDIT | NOMODIFY]
```

- *FIELDS* permite ca numai câmpurile precizate să intre în editare (selecție pe verticală);
- FOR precizează condiţiile conform cărora înregistrărilor vor fi afişate în fereastra BROWSE;
- FREEZE permite modificări doar pe câmpul precizat;
- NOAPPEND invalidează posibilitatea efectuării adăugărilor;
- NODELETE elimină marcarile pentru ștergere;
- *NOEDIT* | *NOMODIFY* interzice modificările.

Ca efect al lansării comenzii *BROWSE,* în meniul principal, apare o nouă opțiune, *T<u>a</u>ble*, care permite:

 adăugarea unei noi înregistrări la un tabel de date: Append <u>N</u>ew Record (Ctrl+Y);

💓 М	icros	oft Vis	ual Fox	ro					
<u>F</u> ile	<u>E</u> dit	⊻iew	<u>T</u> ools	<u>P</u> rogram	T <u>a</u> ble	<u>W</u> indow	<u>H</u> elp		
					<u>P</u> rope <u>F</u> ont.	erties 			
					<u>G</u> o to	o Record		+	<u>I</u> op
					Appe	end <u>N</u> ew Ri	ecord	Ctrl+Y	<u>B</u> ottom
					Togg	gle Deletion	Mark	Ctrl+T	<u>N</u> ext Providuo
					<u>A</u> ppe <u>D</u> elet R <u>e</u> ca Re <u>m</u> o	end Record te Records all Records, ove Delete	ls d Record	ts	<u>R</u> ecord # Locate
					<u>R</u> epla <u>S</u> ize I Mo <u>v</u> e	ace Field Field e Field			
					Resi <u>z</u> ✓ Link I C <u>h</u> an	ze Partition: Partitions nge Partition	s 18	Ctrl+H	
					Rebu	uild Inde <u>x</u> es	;		

- stabilirea înregistrării curente: <u>Go to Record;</u>
- precizarea unor criterii de poziţionare, <u>L</u>ocate..., la fel ca prin folosirea comenzii cu acelaşi nume:

	💓 Loca	ate Record		×
Domeniul	_ <u>S</u> cope	All		· · ·
Clauza FOR	<u> </u>			
Clauza <i>WHILE</i>	_ <u>W</u> hile			
			Locate	Cancel

- stergerea datelor din tabele:
 - marcare (demarcare) pentru ştergere: <u>T</u>oggle Deletion Mark (Ctrl+T) sau clic cu mouse-ul pe fanionul de ştergere din dreptul înregistrării:

Fanionul de stergere

Ë	📺 Personal						
		Nume	Marca				
		Vasiliu Alina	202				
	Π	Pop Stefan	157				
		Musat Luminita	204				
₽		Popescu Constantin	159				
		Petriu Stelian	177				

- stergere fizică: Re<u>m</u>ove Deleted Records;
- stergeri conform sintaxei comenzilor DELETE şi RECALL:
 <u>D</u>elete Records, R<u>e</u>call Records;
- modificări ale valorilor înscrise în câmpurile tabelelor conform sintaxei comenzii REPLACE: <u>R</u>eplace Field.

Câmpuri MEMO

Câmpurile **MEMO** definesc **câmpuri de lungime variabilă** de tip text. Un tabel de date, cu cel puţin un câmp MEMO, are asociat un **fişier suplimentar** în care sunt depuse informaţiile legate de acest câmp. Pentru a putea identifica datele din fişierul MEMO asociat, care

Fişierul MEMO asociat

Л

Tabelul de date

Nume	Marca	Functia	Adresa	
Vasiliu Alina	202	Secretara	memo	
Pop Stefan	157	Director general	memo	
Musat Luminita	204	Secretara	memo	

aparțin unei anumite înregistrări, în tabelul de date, pe poziția corespunzătoare a înregistrării, se memorează o adresă spre fișierul MEMO (poziția primului caracter al câmpului respectiv).

Modificarea câmpurilor MEMO se poate face în fereastra *Browse* sau prin comenzi specifice:

```
MODIFY MEMO <câmp memo1>[,<câmp memo2>,....]
```

 se deschide câte o fereastră de editare pentru fiecare câmp memo;

```
CLOSE MEMO <câmp memo1>[,<câmp memo2>,....] | ALL
```

- se vor închide ferestrele de editare corespunzătoare câmpurilor *memo* specificate în listă (sau toate - *ALL*);

```
APPEND MEMO <câmp memo> FROM <fișier> [OVERWRITE]
```

- conținutul unui fișier extern, de tip text, va fi adăugat la sfârșitul câmpului *memo* precizat (dacă se folosește clauza OVERWRITE vechiul conținut al câmpului se pierde);

```
COPY MEMO <câmp memo> TO <fișier> [ADDITIVE]
```

-conținutul unui câmp *memo* se copiază într-un fișier extern de tip text. Dacă se folosește clauza ADDITIVE copierea se va face la sfârșitul unui fișier deja existent.

🗐 1.6. Ordonarea și actualizarea tabelelor de date

Înregistrările apar, în tabelul de date, în ordinea în care au fost introduse, dar nu întotdeauna aceasta este și ordinea lor de utilizare. Ordonarea unui tabel presupune stabilirea unui criteriu după care să fie parcurse înregistrările acestuia. Acest criteriu poate fi un câmp sau o combinație de câmpuri ale tabelului de date numit **cheie de ordonare**. O cheie de ordonare este o expresie, alcătuită din câmpuri ale tabelului, care este evaluată pentru fiecare înregistrare, ordinea de prelucrare a înregistrărilor fiind dată de valorile acestei expresii. Se folosesc două metode de ordonare a bazelor de date: *fizică* și *logică*.

Ordonarea fizică (sortarea)

Sortarea constă în rearanjarea fizică a datelor din tabel într-o altă ordine, dată de criteriul de sortare. Astfel rezultă un tabel de date nou cu aceleași informații ca și cel original, dar ordinea fizică (naturală) a înregistrărilor este schimbată.



- <fișier> numele tabelului de date nou creat;
- o <câmp1>, <câmp2>,.... lista cheilor de sortare;
- /A , /D sortare ascendentă, descendentă, relativă la câmpul asociat;
- /*C* sortare în ordine alfabetică, necontând literele mari sau mici;
- ASCENDING | DESCENDING ordonare ascendentă sau descendentă cu specificarea ordinii pentru toate câmpurile;
- *FIELDS* selecție pe verticală (filtrare).

Ordonarea logică (indexarea)

Indexarea este o metodă mult mai eficientă de ordonare a tabelelor de date, care permite ca operațiile de consultare și căutare a datelor în tabel să fie rapide. Prin indexare se creează un **fișier index**, asociat bazei de date, de dimensiuni mult mai mici decât dimensiunea bazei de date. Fără a detalia structura fișierului index, trebuie menționat că fișierul index este o listă simplu înlănțuită a numerelor articolelor din baza de date aflate în ordinea lor naturală.

La definirea structurii bazei de date (prin *CREATE* sau *MODIFY STRUCTURE*), în *coloana Index*, se pot defini unul sau mai multe criterii de indexare care sunt identificate prin etichete (numele unei etichete este același, în acest caz, cu numele câmpului). Activarea unuia sau altuia din criteriile de indexare se realizează în două moduri:

a) la deschiderea tabelului de date:

USE <fișier> **ORDER** <nume etichetă>

b) în timpul lucrului prin comanda:

```
SET ORDER TO TAG < nume etichetă >
```

Stabilirea altor criterii de indexare, diferite de cele definite în structura tabelului de date, se poate face prin comanda:

INDEX ON <exp></exp>	TAG <nume etichetă=""></nume>
[FOR < cond>]
[ASCENDING	DESCENDING]
[UNIQUE]	

- TAG identifică un criteriu de indexare;
- UNIQUE clauză care nu admite dubluri pe câmpul de indexare

Ştergerea unei etichete (criteriu) de indexare se realizează utilizând comanda:

```
DELETE TAG < nume etichetă > | ALL
```

Principalul *avantaj* al indexării rezultă din *posibilitatea localizării rapide* a unor înregistrări, mult mai eficientă decât cea furnizată prin *LOCATE*. Acest lucru este realizat prin comanda:

SEEK <exp>

Comanda *SEEK* caută prima înregistrare pentru care valoarea expresiei calculate cu valori din câmpurile acesteia este egală cu <exp>, pe criteriul de indexare selectat.

🔶 Observații

- la un moment dat poate fi activ un singur criteriu de indexare, numele etichetei corespunzătoare fiind furnizat de funcția TAG();
- efectul unei comenzi de căutare (SEEK sau LOCATE) poate fi prelungit prin folosirea comenzii CONTINUE;
- funcţia FOUND() determină (.T. sau .F.) dacă s-a reuşit sau nu localizarea.

Comanda:

```
SET EXACT ON | OFF
```

stabilește modalitatea de căutarea pe subșiruri de caractere. Clauza implicită este *OFF*.

 Dacă <exp> este un şir de caractere, acesta trebuie pus între apostrofuri.

Ē		1.7.	Introd	lucerea	formatată	а	datelor
	≣µ –	±. /.	THUOU	uccica	Ioimatata	a	uateiui

FoxPro oferă posibilitatea introducerii datelor sub forma unor machete sau formate prin asocierea la clauza GET a clauzelor PICTURE și FUNCTION:

```
@<linie, coloană> [GET<variabilă>]
    [PICTURE <expC>] [FUNCTION <expC>]
[READ]
```

- @ plasează cursorul pe poziția specificată a ecranului.
- *GET* stabilește porțiunea de pe ecran unde vor fi introduse valorile ce vor fi atribuite variabilei <variabilă>.

- PICTURE şi FUNCTION permit formatarea afişării datelor. Ambele clauze sunt urmate de un şir de caractere care conţine coduri referitoare la modul de afişare.
- Codurile *PICTURE* se referă doar la un anumit caracter din formatul de afişare și anume la cel de pe aceeaşi poziție cu cea a codului.
- Codurile *FUNCTION* acționează asupra tuturor caracterelor din formatul de afișare.
- Clauzele *PICTURE* și *FUNCTION* pot acționa împreună, dar *FUNCTION* este prioritară.
- *READ* determină citirea conținutelor zonelor de editare pentru introducerea valorilor de la tastatură, create prin GET.

Codurile de formatare PICTURE

- **X** Permite orice caracter
- **9** Cifre și semnele + . E (date numerice)
- # Cifre, spatii și semnele + . E
- A Numai litere
- W Numai date logice
- N Numai litere și cifre
- Y Permite doar caracterele logice Y, y, N şi n
- ! Literele sunt convertite în majuscule
- * Afişează asterix-uri în locul zerourilor inițiale
- Specifică poziția punctului zecimal.

Codurile de formatare FUNCTION

- A Permit numai caractere alfabetice (litere)
- B Aliniere la stânga
- **I** Centrare
- J Aliniere la dreapta
- **C** După un număr pozitiv se afișează *CR* (credit)
- **D** Format dată calendaristică stabilit de SET DATE
- **E** Format dată calendaristică european (britanic)
- T Elimină spatiile de început și sfârșit
- **X** După un număr negativ se afișează *DB* (debit)
- **Z** Afişează spatii dacă valoarea câmpului este 0
- (Numerele negative sunt incluse între paranteze
- ! Literele sunt convertite în majuscule.

```
Exemple:
                         cu două zecimale pe câmp de 7
 * citirea unui număr
 caractere:
 a=0
 @10,10 GET a PICTURE '9999.99'
 READ
 * citirea oricăror caractere pe câmp de lungime 10:
 b=space(10)
  @10,10 GET b PICTURE REPLICATE ('X',10)
 READ
 * introducere cu majuscule:
 b=space(10)
  @10,10 GET b PICTURE REPLICATE ('!',10)
 READ
  * introducerea unei majuscule și a două cifre cu centrare:
         .....PICTURE 'A99' FUNCTION 'I!'
  * următoarele machete de editare sunt echivalente:
         ..... PICTURE 'AAAAA' FUNCTION '!'
         ..... PICTURE 'XXXXX' FUNCTION 'A!'
```

1.8. Actualizarea tabelelor de date

Actualizarea tabelelor de date înseamnă *modificarea structurii* tabelelor de date (vezi comanda *MODIFY STRUCTURE*) sau *modificarea conținutului* acestora. În ceea ce privește modificarea conținutului unui tabel de date se au în vedere posibilitățile de ștergere sau adăugare a unor înregistrări precum și modificarea conținutului celor existente (vezi comenzile *DELETE, PACK, RECALL, REPLACE*). Pentru editarea de la tastatură a tabelelor de date se poate utiliza comanda *BROWSE* sau se pot alcătui secvențe program.

Exemplu.

Fie fişierul Agenda.dbf, având structura:

Nume	Caracter	30
Data_N	Date	8
Telefon	Caracter	10

Pentru introducerea, de la tastatură, a unor date în acest fișier se poate concepe următoarea procedură:

```
procedure introducere
clear
@10,10 say 'Numele' get nume function '!'
@12,10 say 'Data nasterii' get data_n
@14,10 say 'Telefonul' get telefon
read
endproc
```

Folosind această procedură se poate edita următoarea secvență pentru consultarea și actualizarea fișierului *Agenda.dbf*:

clear
use agenda
TEXT
Alegeti optiunea:
1. Afisare
2. Adaugare
3. Modificare
0. Terminare
ENDTEXT
opt=0
@10,20 say 'Optiunea:' get opt picture '9' range 0,3
read
clear
do case
case opt=1
do afisare
case opt=2
do adaugare
case opt=3
do modificare
case opt=0
cancel
endcase
cancel
procedure afisare
browse
endproc
procedure adaugare
append blank

```
do introducere
endproc

procedure modificare
numele = space(30)
@10,0 say 'Numele cautat' get numele function '!'
read
locate for nume=numele
if found()
do introducere
else
WAIT 'Nume negasit!'
endif
endproc
```

😓 Observații

- Pentru afişarea meniului de opţiuni s-a utilizat instrucţiunea
 TEXT ... ENDTEXT
- Pentru afişarea mesajului de eroare "Nume negasit!" s-a utilizat instrucţiunea WAIT.

1.9. Ciclul SCAN

Ciclul *SCAN* este un ciclu cu număr finit de pași, specializat în lucrul cu tabele de date. Această structură repetitivă realizează parcurgerea tabelului și executarea unui grup de instrucțiuni pentru fiecare înregistrare care îndeplinește condițiile specificate. Ciclul este utilizat, deseori, pentru actualizarea conținutului tabelelor de date, conform unor anumite condiții, sau pentru obținerea informațiilor statistice.

Sintaxa este:

```
SCAN [<domeniu>] [FOR <cond1>] [WHILE <cond2>]
[<instrucţiuni>]
[LOOP]
[EXIT]
ENDSCAN
```

1.10. Extragerea unor informații statistice

Comenzile care permit extragerea unor informații din tabelele de date se aplică asupra unor câmpuri sau asupra unor expresii care conțin numele unor câmpuri.

COUNT [TO <var>] [FOR <cond>] [<domeniu>] [WHILE <cond>]

- numără înregistrările din tabelul de date, numărul de înregistrări putând fi memorat într-o variabilă de lucru;

```
SUM <listă exp>[TO<listă var>]
[FOR <cond>] [<domeniu>] [WHILE <cond>]
```

- însumează valorile expresiilor din lista de expresii(<listă exp>)
 și le atribuie, eventual, unei liste de variabile (<listă var>);

AVERAGE <listă exp>[TO <listă var>] [FOR <cond>] [<domeniu>] [WHILE cond>]

- la fel ca SUM având ca rezultat media aritmetică.

CALCULATE <listă funct>[TO<listă var>] [FOR <cond>] [<domeniu>] [WHILE <cond>]

- calculează anumite expresii folosind următoarele funcții (<listă funct>):

SUM(<expN>) - suma; $AVG(\langle expN \rangle)$ - media aritmetică: STD(<expN>) deviaţia standard; VAR(<expN>) – abaterea medie pătratică; - val maximă, nu neapărat numerică; MAX(<exp>) - val minimă, nu neapărat numerică; MIN(<exp>) NPV(<expN1>,<expN2>)[,<expN3>]) - valoarea prezentă netă a unei investiții (vezi functia de calcul financiar); - numărul de înregistrări din tabelul de date. CNT()

🗗 Aplicații

1º Se crează tabelul (fișierul) de date **Personal.dbf** având

structura:

Câmp	Semnificație	Tip de dată
• nume	Numele angajatului	Şir de caractere
• marca	Cod numeric angajat	Şir de caractere
functia	Funcția angajatului	Şir de caractere
• salariu	Salariul brut	Numeric
 datanast 	Data nașterii	Dată calendaristică
• sex	Sex	Logic
• adresa	Adresa angajatului	Memo
2º Se introduc în	tabel următoarele	înregistrări:

88	Personal							×
	Nume	Marca	Functia	Salariu	Datanast	Sex	Adresa	*
	Vasiliu Alina	202	Secretara	2900000	13/01/1970	F	memo	
	Pop Stefan	157	Director general	13000000	12/12/0195	Т	memo	
	Musat Luminita	204	Secretara	2855000	26/03/1973	F	memo	
	Popescu Constantin	159	Contabil sef	11000000	11/02/1958	Т	memo	
	Petriu Stefan	177	Inginer sef	10550000	02/02/1958	Т	memo	
	Toma Dumitru	163	Referent	3700000	06/06/1960	Т	memo	
Þ	Dumitrache Florina	161	Secretara	2850000	25/06/1971	F	memo	-
-							Þ	

Ŧ

în câmpul sex se înscriu valorile .T. (masculin) și .F. (feminin)

- valorile din câmpul adresa (text de lungime variabilă) nu pot fi înscrise direct în tabel, dar sunt stocate într-un fişier asociat tabelului de date numit fişier memo. Trimiterea spre acest fişier se poate face printr-un dublu clic, cu mouse-ul pe câmpul memo corespunzător
 3º Informații privind:
 - fişierele tabel de date prezente pe disc, în directorul de lucru:
 dir
- numărul de înregistrări ale tabelului de date şi numărul de ordine al înregistrării curente:

?reccount()
?recno()

4⁰ Operații asupra tabelului de date:

 limitarea accesului în tabel la înregistrările care conţin angajaţi de sex masculin:

set filter to sex=.T.

browse

• anularea filtrării:

set filter to

browse

• localizarea înregistrării cu marca '177' :

locate for marca='177'

browse

• marcarea pentru ștergere a înregistrărilor 2 și 4:

delete record 2 delete record 4

browse

 marcarea pentru ştergere a înregistrărilor corespunzătoare persoanelor cu vârste mai mari de 40 ani:

```
delete all for (year(date())-year(datanast))>40
browse
```

• demarcarea tuturor înregistrărilor:

recall all

browse

 ştergerea fizică a înregistrărilor pentru care salariul este cuprins între 3500000 şi 12000000:

dele all for salariu >=3500000 and salariu <=12000000 pack

browse

• înlocuirea, în câmpul *nume*, a literelor mici cu litere mari:

replace all nume with upper(nume)

browse

înlocuirea valorică a salariului, la toţi salariaţii, cu 12%:

```
replace all salariu with salariu*1.12
```

browse

5^o Editarea tabelului de date selectiv (selecții pe orizontală și pe verticală) utilizând comanda BROWSE:

• afişarea pe câmpurile nume, functia, salariu :

browse fields nume, functia, salariu

 editarea înregistrărilor corespunzătoare persoanelor de sex masculin și a câmpurilor nume, salariu:

browse fields nume, salariu for sex=.T.

• editarea tabelului de date doar pe câmpul salariu:

brow freeze salariu

6° Introducerea adresei persoanelor având marca '204' și '161' : locate for marca='204' modify memo adresa locate for marca='161' modify memo adresa

7 Se crează, prin sortare, un tabel de date **Persoane.dbf**, ordonarea făcându-se după câmpurile **Functia** (cheie primară) și **Salariu** (cheie secundară), ascendent:

sort on functia, salariu to Persoane use Persoane <u>br</u>owse

- **8**⁰ Se definesc următoarele criterii de indexare:
- criteriul nume+functia, num: index on nume+functia tag num
- criteriul functia+salariu, func: index on functia+str(salariu,8) tag func
- criteriul *salariu*, descendent:
 - index on salariu tag salariu descending
- criteriul *marca* **index on marca tag marca**

90 Vizualizarea tabelului de date conform celor patru criterii de indexare definite anterior: set order to tag num browse set order to tag func browse set order to tag salariu browse ?tag() set order to tag marca browse **10**⁰ Folosind criteriul marca, se localizează înregistrarea de cod '204' și se face, pentru aceasta, o mărire de salariu de 8%: set order to tag marca seek '204' replace salariu with salariu*1.08 **11**⁰ Informații despre tabelul de date: • numărul total de înregistrări: count to tot ?tot • numărul de angajați 'Secretara': count for functia='Secretara' to a ?a sau (prin program): a=0 scan for functia='Secretara' a=a+1endscan ?a • media salarială pentru funcția 'Secretara': average salariu for functia='Secretara' to b sau: calculate avg(salariu) for functia='Secretara' to b **?h** • salariul maxim: calculate max(salariu) to m **?m**

```
sau (prin program):

maxim= salariu

scan

if salariu >maxim

maxim=salariu

endif

endscan

?maxim
```

2. Utilizarea bazelor de date şi a vederilor

2.1. Crearea unei baze de date (Database)

Prin crearea unei *baze de date* mai multe tabele de date sunt reunite într-o singură colecție. Fiecare bază de date are asociat un *dicționar de date* care oferă mai multă flexibilitate în proiectarea și modificarea bazei de date și reduce sarcina programatorului de a scrie cod pentru validările la nivel de câmp și de înregistrare sau pentru a asigura unicitatea valorilor din câmpurile reprezentând chei primare. Fișierele bază de date, conținând tabele de date și informațiile asociate lor sunt fișiere având **extensia implicită dbc**.

Dicționarul de date Visual FoxPro permite crearea sau specificarea unor elemente cum ar fi:

- Cheile primare și secundare;
- Relațiile dintre tabelele bazelor de date;
- Valorile implicite ale câmpurilor;
- Măștile de intrare și a formatelor de afișare a câmpurilor;
- Regulile la nivel de câmp și de înregistrare;
- Procedurile stocate.

Crearea unei baze de date poate fi inițiată prin comanda:

MODIFY DATABASE [<fişier>]

sau prin alegerea opțiunii *Eile, New* din **meniu** (în acest caz în caseta New se selectează *Database* și *New file*). Când se creează o nouă bază de date, aceasta nu conține nimic (tabele asociate sau alte obiecte).

Pentru iniţierea unei sesiuni de lucru cu o bază de date, aceasta trebuie să fie activată (deschisă). Deschiderea unei baze de date se poate realiza prin **meniu** (*<u>F</u>ile, <u>O</u>pen, Files of <u>t</u>ype: Database* (*.*dbc*)) sau prin **comanda** MODIFY DATABASE.

Observație

Lucrul cu bazele de date se realizează utilizând **Constructorul bazei de date** în fereastra **Database Designer**, in meniul principal fiind activată opțiunea Database care oferă instrumentele de lucru cu bazele de date.

🖑 Sfat

Lucrați cu bara de instrumente *Database Designer*. Dacă aceasta nu este afișată utilizați meniul: <u>V</u>iew, <u>T</u>oolbars...Database Designer.



Bara de instrumente *Database Designer* permite efectuarea unor operațiuni asupra bazelor de date cum ar fi:

- **New Table** Crearea unui nou tabel de date și includerea lui
- în baza de date;
- Add Table Includerea unui tabel deja creat în baza de date;
- Remove Table Eliminarea unui tabel de date din baza de date;
- Modify Table Modificarea structurii unui tabel;

 Browse Table – Deschiderea ferestrei de vizualizarea pentru un tabel;

• New Local View – Crearea unei vederi (View).

🗐 2.2. Adăugarea tabelelor la baza de date

Fiecare tabel Visual FoxPro poate exista în una din următoarele două stări: ca **tabel liber**, adică sub formă de fișier *.dbf* neasociat vreunei baze de date, sau ca tabel dintr-o bază de date, adică sub formă de **tabel asociat** unei baze de date. Tabelele asociate unei baze de date au unele proprietăți specifice, care nu există în cadrul tabelelor externe, cum ar fi regulile la nivel de câmp și de înregistrare, declanșatoarele și relațiile permanente.

Tabelele pot fi asociate cu o bază de date prin crearea lor în cadrul unei baze de date deschise în fundal sau prin adăugarea tabelelor existente la o bază de date.

Observație

Un tabel existent poate fi asociat cu o singură bază de date.

2.3. Eliminarea tabelelor din baza de date

Prin adăugarea unui tabel la o bază de date, Visual FoxPro modifică înregistrarea din antetul fișierului corespunzător tabelului pentru a actualiza calea și numele fișierului asociat bazei de date care include acum tabelul. Procesul de eliminare a unui tabel dintr-o bază de date (*Remove Table*) nu numai că șterge tabelul și dicționarele de date din fișierul bazei de date, ci actualizează informația de relație pentru a reflecta noul statut al tabelului, cel de tabel liber. Eliminarea unui tabel dintr-o bază de date nu determină ștergerea automată a fișierului asociat tabelului. Pentru ștergerea fișierului .*dbf* se poate utiliza comanda DELETE.

2.4. Crearea relațiilor permanente între tabele

Pentru stabilirea unor relații între tabele, trebuie ținut cont că acestea nu sunt relații de egalitate, ci de subordonare. Unul dintre tabele va fi **părinte** (*Parent Table*), iar celălalt **copil** (*Child Table*).

În Visual FoxPro, sunt utilizați **indecșii** pentru a stabili relații permanente între tabelele bazei de date. Indecșii sunt constituiți pe baza unor expresii (în particular indecșii pot fi nume de câmpuri ale tabelelor) prin utilizarea procedeului de **indexare**. Tipul etichetei de index sau al cheii determină tipul de relație permanentă care se poate crea. Este necesară folosirea unei **chei primare** (*Primary*) pentru relații *"una la mai multe"*. Pentru partea *"mai multe"*, cheile de indexare sunt **chei obișnuite** (*Regular*).

Exemplu

Baza de date *Furnizor.dbc* include tabelele *clienti.dbf, comenzi.dbf, produse.dbf*. Aceste tabele de date sunt indexate după câmpurile *codc* (cod client) și *codp* (cod produs). Tabelele *clienti.dbf* și *produse.db*f sunt tabele părinte iar tabelul *comenzi.dbf* este tabel copil, stabilindu-se relații "una la mai multe": *clienti.dbf– comenzi.dbf*



şi *produse.dbf* – *comenzi.dbf*. În tabelul *clienti.dbf* respectiv *produse.dbf* **cheile** *codc*, *codp* sunt **primare** (unice), valorile corespunzătoare acestora neputând fi scrise decât o singură dată în tabel – nu pot exista mai mulți clienți sau mai multe produse cu același cod:

💓 Table Designer -	clienti.dbf				X
Fields Indexes Ta	ble				
Order Name	Туре	Expression	Filter		
t CODC	Primary	codc			ок
				Ca	ancel
				Fia	ura2.2

În tabela copil, *comenzi.dbf*, pot apare de mai multe ori acelaşi coduri de clienți sau produse. Se impune stabilirea cheilor *codc* și *codp* drept **chei normale**:

Ø	Table I	Designer	- comenzi.dbf				×
F	ields 🕕	ndexes	Table				
	Order	Name	Туре	Expression	Filter		
	t	CODP	Regular	codp		A	ОК
	t	CODC	Regular	codc			Cancel



Crearea unei relații permanente între tabele:

În fereastra *Database Designer*, se alege numele indexului primar (cum ar fi *codc*) și apoi se trage spre numele indexului din tabela legata (cum ar fi *codc* din *comenzi.dbf*).

Stergerea unei relații permanente dintre tabele:

În fereastra *Database Designer*, se execută clic pe linia de relație dintre două tabele. Ca urmare, lățimea liniei de relație crește pentru a indica faptul că s-a selectat relația respectivă. Pentru eliminarea relației se apasă tasta *DELETE*.

Crearea integrității referențiale

Generatorul de Integritate Referențială (*Referential Integrity Builder*) se lansează prin **meniu**: <u>D</u>atabase, Edit Referetial <u>Integrity</u>. Generarea *integrității referențiale* presupune crearea unui set de reguli care să păstreze relațiile definite între tabele atunci când se inserează sau se șterg înregistrări.

Ø	Referential Integ	rity Builder						X
R	ules for Updating] Rules for Delet	ing Rules	s for Insertii	ng			
١	Which rule do you want to apply when the key value in the parent table is modified?							
(🖲 <u>C</u> ascade: upda	tes all related reco	ords in the (child table v	vith the nev	v key value.		
(C <u>R</u> estrict: prohib	its the update if th	ere are rela	ted records	; in the child	i table.		
(O <u>Ig</u> nore: allows t	he update and lea	ives related	records in	the child ta	ble alone.		
	Parent Table	Child Table	Update	Delete	Insert	Parent Tag	Child Tag	
	clienti	comenzi	Cascade	Cascade	Restrict	codc	codc	
	produse	comenzi	Ignore	Ignore	Ignore	codp	codp	
								_
╞								-
┢								
L								
	<u>H</u> elp					<u>0</u> K	Cancel	

Figura2.4

Prin întărirea integrității referențiale, Visual FoxPro nu permite:

- Să se adauge înregistrări la un tabel copil atunci când nu există nici o înregistrare asociată în tabelul părinte.
- Să se modifice valori în tabelul părinte care ar duce la apariția de înregistrări orfane în tabelul copil.
- Să se șteargă înregistrări din tabelul părinte atunci când există înregistrări asociate în tabelul copil.

Generatorul de integritate referențială gestionează mai multe niveluri de ștergere și actualizări în cascadă și este instrumentul cel mai potrivit pentru generarea integrității referențiale.

🗕 Observații

- Se recomandă lansarea generatorului de integritate referențială imediat după includerea unui nou tabel și stabilirea relațiilor cu alte tabele deja incluse.
- Se recomandă realizarea relațiilor de integritate în cascadă.
- Dacă se intervine în arhitectura bazei de date, de pildă, prin modificarea tabelelor sau a indecşilor folosiţi într-o relaţie permanentă, se impune relansarea generatorului înainte de a se folosi baza de date astfel încât acesta să reflecte noua arhitectură. Altfel, pot fi generate rezultate imprevizibile.

🗐 2.5. Închiderea unei baze de date

Închiderea unei baze de date se poate realiza prin comenzile:

-sau-

CLOSE DATABASE

CLOSE ALL

2.6. Crearea unei vederi (View)

Vederile reprezintă un tip special de tabele, construite pe baza datelor din una sau mai multe tabele sau vederi legate între ele prin relații. Vederile sunt folosite atunci când se dorește o altă structură a tabelelor (bazelor de date), structură construită pe baza unor tabele și vederi deja existente.

😓 Observații

- Vederile sunt incluse în bazele de date, ele neputând fi folosite decât atunci când baza de date care le conține este deschisă. Chiar memorarea efectivă a vederilor se face în fişierul bazei de date (.dbc).
- Vederile sunt văzute și prelucrate ca orice alt tabel.
- Vederile sunt tabele logice, nestocând fizic date.

Crearea unei vederi în cadrul unei baze de date se poate realiza prin **meniu**, <u>*File*</u>, <u>*New*</u> (în acest caz în caseta <u>New</u> se selectează <u>*View*</u> și <u>*New*</u> *file*), sau folosind **butonul** *New Local View* din bara de instrumente *Database Designer*. În meniul principal este activată opțiunea *Query* care oferă instrumentele de lucru pentru vederi și interogări. Pentru construirea vederilor se folosește un utilitar special al SGBD-ului Visual FoxPro numit *View Designer* (*"Constructorul de vederi"*).

Modul de utilizare a acestui utilitar este asemănător celui folosit pentru construirea interogărilor (*Query Designer*).

Construirea unei vederi implică parcurgerea următoarelor etape principale:

- dacă baza de date care urmează să conţină vederea nu este deschisă, se deschide, eventual se creează;
- apoi se pornește Constructorul de vederi;
- urmează precizarea sursei de date, adică a tabelelor din care se preiau datele pentru construirea vederii respective:

😽 Add Table or View	×
<u>D</u> atabase:	
Furnizor 💌	Add
Tables in database:	- Close
clienti Comenzi	Other
produse	
	Select
	• T <u>a</u> bles
	-

Figura 2.5

apoi se stabilesc câmpurile care vor intra în componența vederii, câmpuri selectate dintre cele ale tabelelor sursă precizate anterior;

 în continuare, se specifică o serie de parametri precum: legăturile între tabelele specificate, eventualele condiții de filtrare (selecție), ordonare, grupare a datelor în vedere, modul de actualizare a datelor sursă pe baza datelor din vedere etc.

🗗 Aplicații

1º Pentru evidența produselor livrate de un furnizor se creează baza de date **Furnizor.dbc**, având în componență următoarele tabele de date:

a) clienti.dbf (Client C, 35 / Codc C, 4 / Adresa - memo)

Client	Codc	Adresa
Maximo SRL	MX13	memo
Conex Imp SRL	CX13	memo
Livcom SRL	LC01	memo

b) comenzi.dbf (Data D / Cantitate N, 10 / Codc C, 4 / Codp C, 5)

Data	Cantitate	Codc	Codp
12/03/2001	25	MX13	TC005
12/03/2001	150	MX13	VL002
12/03/2001	200	PM33	VE002
13/03/2001	250	LC01	VL002
14/03/2001	20	CX13	TC005
15/03/2001	120	CX13	FI001

c) produse.dbf (Denumire C, 35 / Pret N, 10 /Um C, 6 / Codp C, 5 / Stoc N, 12)

Denumire	Pret	Um	Codp	Stoc
TEVI CUPRU 20 MM	540000	BUC	TC005	55
VOPSEA EMAIL ALBA DYO	75000	LITRU	VE002	570
VOPSEA LAVABILA ALBA KOBBER	76000	LITRU	VL002	620
FAIANTA IMPORT ITALIA	120000	MP	FI001	400

unde **Codc** este codul atribuit firmei, iar **Codp** este codul produsului și vor constituii cheile de legătură între tabele.

2⁰ Să se indexeze tabelele de date:

- clienti.dbf după câmpul Codc, căruia i se va atribuii statutul de cheie primară (vezi figura 2.2);
- comenzi.dbf după câmpurile Codc și Codp cărora li se va atribuii statutul de **chei normale** (vezi figura 2.3);
- produse.dbf după câmpul Codp cheie primară.

3º Să se realizeze relațiile de tip "una la mai multe" (vezi figura 2.1):

- a) între tabelele de date clienti.dbf și comenzi.dbf (clic cu mouse-ul pe cheia Codc din tabelul clienti.dbf și tragere peste cheia Codc din tabelul comenzi.dbf).
- b) între tabelele de date **produse.dbf** și **comenzi.dbf**.

4⁰ Să se lanseze Generatorul de Integritate Referențială (prin meniu: **D**atabase, Edit Referetial **I**ntegrity) pentru realizarea relațiilor de integritate în cascadă (Cascade) – vezi figura 2.4. Să se verifice stabilirea integrității referențiale în următoarele situații:

- o stergere logică (DELETE) în tabelul părinte, clienti.dbf, se reflectă ca ștergere logică în tabelul copil comenzi.dbf.
- orice modificare a cheii primare din tabelul părinte, clienti.dbf, se reflectă ca modificare în tabelul copil **comenzi.dbf**.



5^o Să se includă în baza de date un tabel vedere, livrari:

Indicație

Se va acționa **butonul** *New Local View* din bara de instrumente *Database Designer* și se vor include în vedere cele trei tabele de date (*vezi figura 2.5*), precum și câmpurile necesare. La închiderea ferestrei View *Designer*, acestei vederi, i se va asocia numele **livrari**.

6º Să se obțină situația livrărilor, afișând conținutul tabelului **livrari** (Browse):

Client	Denumire	Um	Pret	Cantitate	Data
Maximo SRL	TEVI CUPRU 20 MM	BUC	540000	25	12/03/2001
Conex Imp SRL	TEVI CUPRU 20 MM	BUC	540000	20	14/03/2001
Livcom SRL	VOPSEA LAVABILA ALBA KOBBER	LITRU	76000	250	13/03/2001
Maximo SRL	VOPSEA LAVABILA ALBA KOBBER	LITRU	76000	150	12/03/2001
Conex Imp SRL	FAIANTA IMPORT ITALIA	MP	120000	120	15/03/2001

3. Interogarea bazelor de date

O *interogare* a unei baze de date reprezintă extragerea - în funcție de anumite criterii și într-un anumit format - unor date din tabele de date care alcătuiesc baze de date. Componenta *Visual FoxPro*-ului care realizează acest obiectiv se numește **Query Designer** ("Constructorul de interogări"), componentă asemănătoarea cu **View Designer** ("Constructorul de vederi"). Datele extrase printr-o interogare pot fi prezentate sub diverse forme (*Browse*, rapoarte, grafice, etc,).

🗐 3.1. Crearea, deschiderea și rularea unei interogări

QPR) poate fi inițiată prin comanda:

MODIFY QUERY [<nume interogare>]
sau prin alegerea opțiunii <u>*F*</u>ile, <u>N</u>ew din **meniu** (în acest caz în caseta New se selectează <u>*F*</u>orm și <u>N</u>ew file).

Activarea (deschiderea) unui formular se poate realiza prin meniu (*<u>F</u>ile, <u>O</u>pen, Files of <u>type</u>: Query (*.qpr))* sau prin comanda MODIFY QUERY.

O interogare este o secvență executabilă care poate fi rulată în următoarele moduri:

din meniul principal (<u>Query</u>, <u>R</u>un Query),

prin acţionarea butonului
 din linia de instrumente standard,

prin comanda
 DO <nume interogare>.

Observație

Prin lansarea **Constructorului de interogări**, în fereastra **Query Designer**, meniul principal va include opțiunea *Query* care oferă instrumentele de lucru necesare.

🖑 Sfat

Lucrați cu bara de instrumente *Query Designer*. Dacă aceasta nu este afișată utilizați meniul: <u>V</u>iew, <u>T</u>oolbars...Query Designer.

Remove Table



Bara de instrumente *Query Designer* permite efectuarea unor operațiuni cum ar fi:

• Add Table – Includerea unui tabel de date în interogare;

Remove Table – Eliminarea unui tabel;

 Add Join – Stabilirea sau modificarea unor legături între tabelele de date incluse în interogare;

Show the SQL window – transpunerea secvenţei de interogare în instrucţiuni SQL;

 Query Destination – stabilirea destinaţiei unei interogări: o fereastră *Browse*, un tabel permanent, o listă la imprimantă (report), un grafic, etc.

 Maximize the table View – modificarea formei de afişare a conţinutului ferestrei Query Designer.

Fereastra **Query Designer** (figura 3.1) este constituită din două zone, în care sunt afișate tabelele surse de date (împreună cu legăturile dintre ele), respectiv paginile în care se vor preciza parametrii de definire a interogării și anume:

- Fields (câmpuri) specificarea câmpurilor sursă ce vor fi incluse în tabelul destinație;
- Join (legătură) legăturile între tabelele sursă;
- Filter (filtru) –condițiile de selecție (filtrare) a datelor din tabelele de date;
- Order By (ordonare) specifică ordinea de prelucrare;
- Group By (grupare) criteriile de grupare a datelor;
- Miscellaneous (altele).

📰 Query Designer - interogare	
Discipline Note * inota cod inota cod inota cod inota grupa	▲
Fields Join Filter Order By Group By Miscellaneous	
Available fields: Student: matricola Student: matricola Add > Student: grupa Add > Note nota Add All >> Note nota Note.cod Functions and expressions:	ia V

3.2. Fereastra Query Designer

Specificarea câmpurilor interogării

Pentru ca un câmp să fie preluat într-o interogare se activează pagina **Fields**, după care acesta se selectează din lista *Availabile fields* (câmpuri disponibile) și se acționează butonul *Add*. Ca urmare numele câmpului va fi copiat în lista *Selected fields* (vezi figura 3.1). Dacă se dorește introducerea unui câmp calculat se va introduce formula de calcul corespunzătoare în caseta *Functions and expresions*.

Specificarea legăturilor între tabelele de date

Modelul relațional al bazelor de date permite stabilirea unor legături între tabelele de date. Aceste relații pot fi *permanente*, memorate în fișierul bazei de date, sau stabilite *dinamic* în cadrul unor secvențe program. *Stabilirea legăturilor dinamice* se poate realiza cu ajutorul comenzii:

SET RELATION TO	<etichetă1> INTO <tabel1>,</tabel1></etichetă1>
	<etichetă2> INTO <tabel2>,</tabel2></etichetă2>
1	

unde:

- tabelul părinte este cel activ;
- TO stabilește criteriile de ordonare;
- INTO precizează tabelele copil.

La fel ca în cazul definirii unei vederi (cu *View Designer*), pentru stabilirea unei legături între două tabele de date, este necesar ca cele două tabele să aibă cel puțin un câmp comun prin intermediul căruia se realizează corespondența. Relația stabilită între două tabele legate este o relație de subordonare: unul din tabele va fi **tabelul părinte**, iar celălalt va fi **tabelul copil**. Mutarea indicatorului de înregistrări al tabelului părinte pe o anumită înregistrare determină mutarea indicatorului de înregistrăre al tabelului copil pe înregistrarea corespunzătoare, dar nu și invers.

🔒 Observații:

- Este obligatorie indexarea tabelului copil după criteriul care stabilește legătura (de obicei acesta este un câmp comun cu al tabelului părinte, dar poate fi și o expresie oarecare).
- Se pot stabilii două tipuri de legături între tabelul părinte și tabelul copil:
 - *"legături unu la unu",* când în tabelul copil valorile cheii de indexare sunt unice;
 - "legături unu la mai multe", când înregistrările tabelului copil pot lua și valori egale pe cheia de indexare.

Legăturile permanente necesare pentru stabilirea unei interogări se aleg folosind pagina **Join** (figura 3.2), prin specificarea câmpurilor care fac obiectul legăturii și a tipului de legătură:

- Inner Join legătură interioară, între înregistrările din tabelul părinte care au corespondent în tabelul copil şi invers;
- Left Join legătura la stânga, când vor fi luate în considerare și înregistrările din tabelul părinte (din stânga) care nu au corespondent în tabelul copil;
- Right Join legătura la dreapta, când vor fi luate în considerare şi înregistrările din tabelul copil (din dreapta) care nu au corespondent în tabelul părinte;
- **Full Join** vor fi considerate toate legăturile.

Fields Join	Filter	Order By	Group By) Mis	scellaneous		
Type ↔ Inner ↓ t ↔ Inner ↓	oii S oii D	ield Name tudenti.matri riscipline.com	icola 1	<u>Not</u>	Criteria = =	Value Note.matricola Note.cod	Logical
						Insert	<u>R</u> emove



Condiţiile de selecţie a înregistrărilor

Principala funcție a unei interogări este aceea de selecție. Criteriile de selecție sunt precizate în pagina *Filter*.

Fields Join Filter Order By	Group By Mis	cellaneous		
Field Name Studenti.grupa	Not Criteria =	Example FC01	Case	Logical <none></none>
			Insert	Bemove
			Insen	

Figura 3.3

Ordonarea afişării rezultatelor interogării

Specificarea ordinii de afişare se face cu ajutorul paginii **Order By**. În figura 3.4 s-a selectat ordonarea după câmpul *nume* din tabelul *Studenti.dbf*:

Fields Join Filter Order B	9 Group By Miscellaneous	
Sele <u>c</u> ted fields: Studenti.nume Studenti.grupa Discipline.materia Note.nota	Add >	Ordering criteria:
	<u>Ascending</u> <u>D</u> escending	





Query Designer (pagina **Group By**) permite ca mai multe înregistrări să fie cumulate după mai multe criterii, pentru fiecare grup de înregistrări fiind furnizată în tabelul de destinație o singură înregistrare. Gruparea poate fi realizată după unul sau mai multe câmpuri care vor forma cheia de grupare. Înregistrările având aceleași chei de grupare vor furniza în tabela de destinație o singură înregistrare (câmpurile numerice pentru un grup vor conține valori obținute prin însumarea valorilor corespunzătoare din grup).



Destinația interogării

Figura3.5

Datele extrase prin interogare pot fi trimise spre o anumită destinație (folosind butonul *Query Destination* din bara de butoane a Constructorului de interogări) cum ar fi:

- fereastra *Browse* implicit;
- un tabel permanent (Table) acesta putând fi folosit într-o nouă prelucrare;
- ecranul (Screen);
- o listă la imprimantă (*Report*);
- un grafic (Graph) construit cu ajutorul componentei Microsoft Graph.

🗗 Aplicații

1⁰ Să se creeze următoarele tabele de date:

Studenti.dbf (nume C, 30 / matricola N,4 /grupa C,4)

- Discipline.dbf (materia C, 30 / cod C,4)
- Note.dbf (nota N,2 / matricola N, 4 / cod C,4):

Nume	Matricola	Grupa
IURIE RAZVAN	205	FC01
CONTANTINESCU PAUL	203	FC01
VASILIU ELENA	201	FC01
POPA DUMITRU	254	FC02
DUMITRU ENE	255	FC02
ANDREESCU VIRGIL	421	TSO1
MUSAT ELENA	432	TSO1
POPESCU IOANA	475	TSO2
TOMA ALEXANDRU	477	TSO2

Materia	Cod	Nota	Matricola	Cod
MACROECONOMIE	MC01	7	203	IF01
INFORMATICA I	IF01	10	255	MT01
INFORMATICA II	IF02	8	203	IF02
MATEMATICA	MT01	6	421	MC01
MANAGEMENT	MN01	10	475	IF02
		9	255	MC01
		10	432	MT01

2⁰ Să se indexeze tabelele de date Studenti.dbf, Discipline.dbf după câmpurile Matricola respectiv Cod.

3º Să se creeze și să se ruleze fișierul de interogare a tabelelor de date Studenti.dbf, Discipline.dbf, Note.dbf, pentru afişarea notelor studenților incluși în baza de date (vezi figura 3.1 și figura 3.2). Să se stabilească ca destinație a interogării fișierul **Catalog.dbf** (vezi figura 3.5) și să se vizualizeze conținutul acestui tabel.

4º Să se realizeze interogarea bazei de date pentru afişarea notelor în ordinea alfabetică a numelor studenților (vezi figura 3.4).

5⁰ Să se ruleze fișierul de interogări pentru afișarea notelor studenților din grupa 'FC01' (vezi figura 3.3).

6° Să se ruleze fișierul de interogări pentru toate cele patru tipuri de legături (Inner Join, Left Join, Right Join, Full Join) – vezi figura 3.2.

4. Utilizarea formularelor

Formularul (forma) reprezintă o fereastră pentru introducerea sau extragerea datelor utilizatorului, în mod interactiv, cu ajutorul unor *obiecte de interfață* cărora le sunt asociate *proprietăți* și *metode* (secvențe de instrucțiuni – secvențe de cod). Definirea unui formular se realizează prin utilizarea tehnologiei **POO** (**P**rogramare **O**rientată **O**biect) care permite modificare proprietăților obiectelor incluse în formular, ceea ce se reflectă în aspectul formularului și a comportamentului acesteia în diferite situații.

Form Designer ("Constructorul de formulare") este o componentă a *Visual FoxPro* - ului dedicată creării și modificării formularelor precum și generării secvenței de cod necesare pentru execuția acestora (fișierele corespunzătoare au extensia implicită **scx**).

🗐 4.1. Crearea, deschiderea și rularea unui formular

Crearea unui formular (fișier cu **extensia implicită SCX**) poate fi inițiată prin comanda:

MODIFY FORM [<nume formular>]

sau prin alegerea opțiunii <u>F</u>ile, <u>N</u>ew din **meniu** (în acest caz în caseta New se selectează <u>F</u>orm și <u>N</u>ew file).

Activarea (deschiderea) unui formular se poate realiza prin meniu (*<u>F</u>ile, <u>O</u>pen, Files of <u>type:</u> Form (*.scx))* sau prin comanda MODIFY FORM.

Scopul creării unui formular este **obținerea unei secvențe executabile** care poate fi rulată în următoarele moduri:

din meniul principal (For<u>m</u>, <u>R</u>un Form),

- prin acţionarea butonului
 din linia de instrumente standard,
- prin comanda DO FORM <nume formular>.

Observație

Prin lansarea **Constructorului de formulare**, în fereastra **Form Designer**, meniul principal va include opțiunea Form care oferă instrumentele de lucru necesare.

4.2. Elementele constructorului de formulare

Ferestrele de lucru

- Form Designer fereastra principală care conţine obiectele formularului.
- Properties fereastra de proprietăți având ca rol modificarea proprietăților obiectelor formularului (metode, nume, titluri, dimensiuni, fonturi, culori, etc.).
- Data Environment mediul de date al formularului, fereastra pentru precizarea tabelelor de date care vor fi deschise automat în momentul rulării formularului.
- Ferestrele pentru cod succesiunile de instrucţiuni care urmează a fi executate la apariţia diferitelor evenimente din cadrul formularului cum ar fi: acţionare unui buton, selectare unui obiect, etc.

Barele utilitare

Form Designer Toolbar - bara de butoane standard a formularului:



Butoanele din această bară de instrumente permit deschiderea/închiderea ferestrelor: *Data Environment, Properties, Code* (afișarea secvenței de cod) sau a barelor de lucru:

- Form Control Toolbar butoanele de control;
- **Color Palette Toolbar** stabilirea atributelor de culoare;
- Layout Toolbar alinierea obiectelor în formular, suprapunerea obiectelor, etc.

😽 Microsoft Visual FoxPro	
<u>File Edit View Format Form Iools Program Window H</u> elp	
	m Controls Toolbar
	a obiectelor din formular
🖩 Form Designer - actualizare.sc:	
Form Designer Toolbar	🖀 Properties - actualizare. 🚛 💷 🛛 🖂 Walizar
Denumirea componentei. [txtComponenta ··· 1 Dolar = Dolar ··· Lei	Command3
Codul txtCod >> << Valoarea	All Data Metho Layout Other
proprietății	X V Jx Adauga
Tipul Stop	AutoSize .F False (Defa A BaseClass Commandbuttor
Pretul in delari (fana TVA)	Cancel .F False (D
edtObservatii	Class Commandbullto
Pretul in lei (cu TVI)	Click Event Proprietate
Valoare TVA FORMULARUL	ColorScheme
	ColorSource 4 - Windows Co Comment (None)
Salt la codul cod Adaugă	Default .F False (Defa
	DisabledForeCo 128,128,128
Oblect	
<u> </u>	
	NUM

🖑 Sfat

Lucrați cu **barele de instrumente**. Dacă acestea nu sunt afișate utilizați meniul: <u>V</u>iew, <u>T</u>oolbars...

4.3. Mediul de date al formularului (Data Environment)

Constructorul de formulare permite configurarea mediului de date în care lucrează formularul, adică a zonelor de lucru și a tabelelor de date deschise în ele. Tabelele de date sunt deschise automat (fără comenzi speciale), dacă acestea sunt incluse în *Data Environment,* înainte de execuția formularului (este de la sine înțeles că aceste tabele de date au fost deja create).

👃 Observații

- Deschiderea ferestrei Data Environment poate fi făcută prin meniu (<u>V</u>iew, Data <u>Environment...</u>) sau folosind butonul corespunzător din bara de butoane Form Designer. Astfel în meniul principal este inclusă opțiunea <u>D</u>ataEnvironment.
- Includerea (Add) unui tabel în Data Environment este realizată prin meniu: <u>D</u>ataEnvironment, <u>A</u>dd sau prin clic cu butonul dreapta al mouse-ului.

Mediul de date al formularului este considerat un obiect, deci are asociate metode și proprietăți. Cele mai importante metode, legate de folosirea datelor sunt:

- Metoda OpenTables este folosită la deschiderea tabelelor de date necesare (dacă nu este utilizată, tabelele asociate sunt deschise automat).
- Metoda Load Event poate fi folosită, de asemenea, la deschiderea tabelelor. Aceasta este iniţiată după metoda OpenTables.
- Metoda Init Event este o metodă apelată după metoda Load, și face anumite inițializări ale obiectelor din formular, inclusiv cu datele din tabelele care sunt deja deschise.
- **Metoda CloseTables** are ca scop închiderea tabelelor de date din mediul de date al formularului la inițiativa utilizatorului.
- Metodele Unload (descărcare) şi Destroy (distrugere a formularului) acţionează asupra mediului de date şi sunt activate după metoda CloseTables.

🗐 4.4. Utilizarea controalelor

Includerea controalelor în cadrul unui formular se realizează prin introducerea unor obiecte (*Label*, *Text Box*, etc.) în acest formular. Pentru aceasta, se acționează butonul corespunzător din *Form Controls Toolbar* și se stabilește porțiunea dreptunghiulară (zona) din formular unde va fi afișat obiectul. Practic, obiectul va fi desenat (se poziționează cursorul mouse-ului în colțul stânga-sus al zonei de afișare, se ține apăsat butonul stâng și se trage cursorul spre colțul dreapta-jos).

🗊 Proprietăți și metode

Un formular poate conține mai multe obiecte, el însuși fiind considerat un obiect. Fereastra **Properties** permite alegerea obiectului și a proprietății a cărei valoare se schimbă, ca de exemplu:

- **Proprietatea** *Caption* este destinată stabilirii unui titlu (denumire) pentru formular sau obiect.
- Proprietatea Name permite atribuirea unor nume obiectelor. Visual FoxPro stabileşte valori implicite proprietăţilor Caption şi Name cum ar fi:
 - Form1, Form2... pentru formulare
 - *Label1, Label2...* pentru obiectele *Label*
 - o *Command1, Command2...* pentru butoane
 - *Text1, Text2...* pentru obiectele *Text Box*
 - *Edit1, Edit2...* pentru obiectele *Edit Box*

Aceste nume ale obiectelor pot fi folosite așa cum au fost definite de *Visual FoxPro* însă utilizatorul are posibilitatea stabilirii unor denumiri proprii.

- **Proprietatea FontSize** stabilește atributele de scriere privind dimensiunea fontului folosit.
- Proprietatea FontBold permite scrierea bold.
- Proprietatea BorderStyle oferă posibilitatea stabilirii atributelor de bordare.

- **Proprietatea Format** permite introducerea formatată a datelor conform codurilor *Function*.
- **Proprietatea Input Mask** permite introducerea formatată a datelor conform codurilor *Picture*.

Observație

O categorie specială de proprietăți o reprezintă metodele. **Metodele** sunt secvențe de cod (instrucțiuni) asociate diferitelor obiecte la activarea/dezactivarea acestora.

Form Designer oferă posibilitatea modificării unei proprietăți prin comenzi de atribuire, referirea la proprietatea respectivă făcându-se prin construcția:

<nume formular>.<nume proprietate>

Referirea la un obiect din cadrul unei formulare se realizează printr-o construcție asemănătoare, adică:

<nume formular>.<nume object>.<nume proprietate>

Există construcții speciale, mai scurte, cum ar fi:

- **This** care se referă la obiectul curent;
- ThisForm care indică formularul ce conţine obiectul curent;
- *Parent* care se referă la obiectul care conține obiectul curent.

Exemple

Form1.Caption='Actualizare'

⇒ modifică proprietatea Caption (titlu) a formularului Form1

Form1.Vizualizare

- ⇒ execută metoda *Vizualizare* ataşată formularului *Form1.Label1.Caption='Numele persoanei'*
 - \Rightarrow modifică proprietatea *Caption* (titlu) a obiectului *Label1*

ThisForm.Caption='Adăugare'

⇒ modifică proprietatea *Caption* (titlu) a formularului curent
 ThisForm.Vizualizare ⇒ execută metoda *Vizualizare* ataşată formularului curent
 This.Caption='Numele persoanei'

⇒ modifică proprietatea *Caption* (titlu) a obiectului

Obiecte din formular

Obiecte simple

A Label – afişează un text informativ

Text Box – definește un câmp de editare în care utilizatorul poate introduce un text.

Edit Box – pentru editarea variabilelor șir de caractere, tablou, câmpuri ale tabelelor de date, câmpuri memo.

Command Button – este tipic pentru startul evenimentelor, pentru căutarea și închiderea formularelor, mutarea diferitelor înregistrări, imprimarea rapoartelor, etc.

Check Box (casete de validare). Obiectele Check Box permit utilizatorului să specifice o stare de tip logic (true sau false), astfel încât la executarea formularului să se poată alege între două stări. Principalele proprietăți care le sunt asociate sunt:

- Proprietatea Value furnizează valorile numerice 0 sau 1 după cum caseta este neselectată sau selectată (bifată).
- Proprietatea Style schimbă stilul de afişare, având valoarea 0-Standard sau 1-Graphical, caseta de validare primind aspectul unui buton neapăsat/apăsat:



Spinner (casete de incrementare/decrementare). Obiectele Spinner permit utilizatorului să-și stabilească opțiunile parcurgând incremental o plajă de valori sau introducând direct valorile în casetă. Pentru aceasta, câmpul de editare al casetei are în dreapta sa două butoane: unul pentru incrementare și altul pentru decrementare:



Proprietăți principale:

- **Proprietatea** *Value* stabilește valoarea inițială înscrisă în casetă și furnizează valorile editate la execuția formularului.
- Proprietățile KeyboardHighValue şi KeyboardLowValue stabilesc limita superioară și inferioară a valorilor introduse în câmpul de editare direct de la tastatură (depășirea acestor limite generează mesaj de eroare).
- Proprietățile SpinnerHighValue și SpinnerLowValue stabilesc limita superioară și inferioară a valorilor introduse cu ajutorul butoanelor de incrementare/decrementare (depășirea acestor limite nu conduce la eroare dar valoarea aflată în editare nu mai este mărită/micşorată).

Combo Box - List Box (casete cu listă - listă derulantă). Listele sunt obiecte care oferă utilizatorului posibilitatea selectării unor elemente dintr-o mulțime finită, afișată total sau parțial pe ecran. Se utilizează două tipuri de liste: *liste simple - List Box* și *liste derulante sau expandabile - ComboBox* care afișează doar elementul curent al listei.

Listele simple, *List Box* afişează la un moment dat mai multe elemente (nu neapărat pe toate):

Vasiliu Alina 📃
Pop Stefan
Musat Luminita
Popescu Constantin
Petriu Stelian 📃

Listele expandabile, *Combo Box*, afişează, în stare neactivată, doar un element al listei:

Pop Stefan 🔹	Pop Stefan 💌
	Vasiliu Alina
	Pop Stefan
	Musat Luminita
	Popescu Constantin
	Petriu Stelian

Proprietăți ale listelor:

- Proprietatea RowSourceType stabileşte sursa de date a elementelor unei liste cum ar fi:
 - 0 None (sursă nespecificată), când specificarea elementelor listei se face în mod dinamic, la rularea formularului;
 - 1 Value (valoare), care presupune că elementele listei sunt enumerate manual în proprietatea *RowSource* (listă de elemente separate prin virgulă);
 - 2 Alias, când elementele listei sunt preluate dintr-o tabelă de date specificată în proprietatea RowSource;
 - 3 SQL Statement (instrucţiuni SQL);
 - o 4 Query (interogare);
 - 5 Array (tablou) elementele listei sunt preluate dintr-un tablou;
 - *etc.*

• Proprietatea RowSource specifică sursa efectivă de date.

Numărul implicit de coloane dintr-o listă este 1 însă pot fi construite și **liste multicoloană**. O variantă simplă de a obține o listă multicoloană dintr-o listă unicoloană o oferă proprietățile *RowSourceType* și *RowSource*. De exemplu, dacă proprietatea *RowSourceType* este 2 – *Alias* proprietatea *RowSource* poate avea valoarea *Personal.Marca+Personal.Nume* (pentru baza de date *Personal.dbf*) astfel încât rezultă o listă de forma:

•
159 Popescu Constantin
157 Pop Stefan
177 Petriu Stefan
159 Toma Dumitru
161 Dumitrache Florina
204 Musat Luminita
201 Vasiliu Alina

Obiecte compuse (container)

Obiectele compuse sunt constituite ca un grup de obiecte simple. Construirea (<u>Builder...</u>) unui obiect container (de exemplu includerea unor obiecte simple) sau editarea (<u>Edit</u>) unui obiect inclus

într-un container acționarea

mouse-ului:



presupune butonului drept al **Option Group** (butoane radio). Butoanele radio sunt un tip special de obiecte de interfață care oferă posibilitatea alegerii unei singure opțiuni din mai multe posibile (sunt obiecte container). Proprietățile cele mai utilizate ale butoanelor radio sunt:

- **Proprietatea** *ButtonCount* stabilește numărul de componente ale obiectului.
- Proprietatea AutoSize determină calcularea automată a dimensiunii grupului de butoane în funcție de conținut, dacă are valoarea .T. – True.
- Proprietatea Value furnizează valorile numerice 1, 2, 3, ... reprezentând numărul de ordine al butonului selectat (implicit 1).
- Proprietatea Style a fiecărui obiect din container schimbă stilul de afişare, având valoarea 0-Standard sau 1-Graphical :

Standard	Graphical	
C Optiune 1	Optiune 1	
Optiune 2	Optiune 2	
C Optiune 3	Optiune 3	

Command Group (butoane de comandă). Grupurile de butoane de comanda au roluri asemănătoare cu ale butoanelor simple (*Command Button*) doar că ele sunt constituite ca obiecte container, deci pot fi construite și editate ca acest tip de obiecte. Cele mai importante proprietăți (**ButtonCount**, **AutoSize**, **Value**) sunt identice cu ale butoanelor radio.

Grid (Grile). Grila este un obiect container, variantă îmbunătățită a ferestrei *Browse*, care conține mai multe obiecte de interfață, plasate matricial, cum ar fi: coloane, antete (header), câmpuri de editare, fiecare cu propriul set de proprietăți și metode. Elementele componente ale unei grile sunt cele prezente și în mediul ferestrei *Browse*:

- liniile, coloanele și celulele;
- barele de derulare folosite atunci când dimensiunea grilei depăşeşte dimensiune zonei vizibile;
- antetul (header) capul de tabel;
- coloana de marcare a liniilor;
- coloana de stergere a liniilor.

Obiectul *Grid* are o organizare internă arborescentă, fiind dispus pe nivele: la primul nivel sunt situate coloanele apoi antetele coloanelor și obiectele de interfață:



Grila și elementele din grilă au anumite proprietăți cum ar fi:

- Proprietatea RecordSourceType stabileşte sursa de date a grilei:
 - 0 Table şi 1 Alias, sursa de date este un tabel de date care va fi deschis automat la iniţializarea grilei. În acest caz, înregistrările tabelei vor fi linii în grilă, iar câmpurile tabelului de date vor fi coloane;
 - 2 Prompt –sursa de date (tabelul de date) se specifică la execuția formularului;
 - 3 Query (interogare);
 - *4 SQL Statement* (instrucțiuni SQL).
- Proprietatea RecordSource specifică denumirea exactă a sursei de date.

🗗 Aplicații

1⁰ Să se editeze următorul formular:

🔰 Calcule financiare	
	Funcția
a1	● FV
a2	- C PV C Payment
3	
	- Calculează
rezultat	

Indicaţii

- Se lansează Constructorul de formulare.
- Se stabilesc **proprietățile obiectelor** în felul următor:

Obiect	Tip	Proprietate	Valoare proprietate	Observații
Form1	Form	Caption	Calcule financiare	
Command1	Button Command	Caption	Calculează	Color Pallete : Verde
		FontBold	.TTrue	
A Label1	Label	Caption	Funcția:	Color Pallete : Roşu
⊙ _{FV}	Optiongroup	Name	Opt	
PV		Caption	FV	Color Pallete : Roşu
Payment		Caption	PV	Color Pallete : Roşu
		Caption	Payment	Color Pallete : Roşu
		FontBold	.TTrue	
abl a1	Text Box	Name	a1	Color Pallete : Albastru
		FontSize	11	
		FontBold	.TTrue	Primul argument
abl a2	Text Box	Name	a2	Color Pallete : Albastru
	•	FontSize	11	
		FontBold	.TTrue	Al doilea argument

din fereastra **Properties**:

abl a3	Text Box	Name	a3	Color Pallete : Albastru
		FontSize	11	
		FontBold	.TTrue	Al treilea argument
abl rezultat	Text Box	Name	Rezultat	Color Pallete : Roşu
		FontSize	12	
		FontBold	.TTrue	Afişează rezultatul

• Se stabilesc **metodele** (secvențele de cod) folosind pagina **Method**

Obiect	Metoda	Secvența de cod
Calculează	Click Event (se poate executa și dublu click pe obiect)	a1=val(ThisForm.a1.Value) a2=val(ThisForm.a2.Value) a3=val(ThisForm.a3.Value) Opt=ThisForm.Opt.Value do case case Opt=1 ThisForm.rezultat.Value=FV(a1,a2,a3) case Opt=2 ThisForm.rezultat.Value=PV(a1,a2,a3) case Opt=3 ThisForm.rezultat.Value=Payment(a1,a2,a3) endcase Observație: Deoarece valorile introduse în Text Box-uri sunt de tip șir de caractere, este necesară convertirea lor în valori numerice, utilizând funcția val.

• Se salvează și se execută formularul astfel creat.

2⁰ Să se editeze un formular pentru actualizarea datelor din tabelul de date **Personal.dbf**.

Indicații

 Se include în mediul de date al formularului, Data Environment, tabelul Personal.dbf şi se importă în formular, ca obiecte, câmpurile tabelului de date, prin tragere, cu mouse-ul în zona de editare a formularului. Aceste obiecte vor fi considerate de Constructorul de formulare ca fiind de tip Text Box (txtNume, txtMarca, txtFunctia, txtSalariu, txtDatanast), de tip Edit Box (pentru câmpul Adresa care este Memo) sau de tip Check Box (pentru câmpul Sex).

👹 Actualiza	аге	
Numele	txtNume	Contor 🔺
Marca Funcția	txtMar Salt la marca M	Vizualizare
Salariu	Adresa Adresa	Adaugă
Data nașterii	edtAdresa 🔺	Şterge
	⊤ Sex	Stop

 Se etichetează denumirile câmpurilor din tabelul de date (Numele, Marca, etc.) folosind obiecte de tip Label.

Obiect	Tip	Proprietate	Valoare	Observații
			proprietate	
Form1	Form	Caption	Actualizare	
Command1	Button Command	Caption	Vizualizare	Color Pallete : Mov
Command2	Button Command	Caption	Adaugă	Color Pallete : Verde
Command3	Button Command	Caption	Şterge	Color Pallete : Albastru
Command4	Button Command	Caption	Stop	Color Pallete : Roşu
abl _M	Text Box	Name	м	Poziționare pe o anumită marcă
		FontBold	.TTrue	
Contor	Spinner	Name	Contor	Contorul înregistrărilor
		Value	1	
		FontBold	.TTrue	

• Se stabilesc **proprietățile obiectelor** în felul următor:

 Se stabilesc metodele (secvenţele de cod) folosind pagina Method din fereastra Properties:

Obiect	Metoda	Secvența de cod
Vizualizare	Click Event	browse
Adaugă	Click Event	append blank ThisForm.Refresh ThisForm.Contor.Value=recno()
Şterge	Click Event	delete pack ThisForm.Refresh ThisForm.Contor.Value=recno()
Stop	Click Event	ThisForm.Release
ab) M	Lost Focus Event	locate for This.Value=Marca if found() ThisForm.Refresh ThisForm.Contor.Value=recno() endif Metoda Lost Focus permite ca la ieșirea din câmpul de editare să se execute secvența care permite: - poziționare pe câmpul Marca - introducerea, în Text Box-uri, a valorilor din tabelul de date (refresh) - afisarea numărului înregistrării curente - recno()
Contor	LostFocus Event	s=This.Value if s>=1 and s= <reccount() goto s ThisForm.Refresh else ThisForm.Contor.Value=recno() endif - poziționare pe valoarea înscrisă (goto s)</reccount()
	DownClick Event	- la fel ca la Contor.LostFocus
	UpClick Event	- la fel ca la Contor.LostFocus

- Se stabilește ordinea de activare a obiectelor prin acționarea butonului **Set Tab Order.**
- Se salvează și se execută formularul astfel creat.

3^o Să se editeze un formular pentru evidența unor calificative. În mediul de date al formularului se vor introduce (**în această ordine!)** tabelele de date:

a) Tabelul de date Calificative.dbf, având structura:

Nume	Character	30
Nota1	Numeric	2
Nota2	Numeric	2
Media	Numeric	5,2

b) Tabelul de date **Personal.dbf**.

1	Aco	rdare calificative				$ \times $
	N	umele				
	Co	mbo1	•	Adaugă	Stop	
	С	alificative				
		Marca/Numele persoanei	Nota 1	Nota 2	Media	-
		abi	abl	abl	abl	
		1				-

Indicaţii

- Obiectul GrdCalificative poate fi creat în formular prin tragerea tabelului Calificative.dbf, cu mouse-ul, din fereastra Data Environment.
- Se stabilesc proprietățile obiectelor active în felul următor:

Obiect	Tip	Proprietate	Valoare proprietate	Observații
Form1	Form	Caption	Acordare calificative	
Adaugă	Button Command	Caption	Adaugă	
	•	FontBold	.TTrue	

C Stop	Button Command	Caption	Stop	
		FontPold	T True	
Combo1	Combo Box	FontBold	.TTrue	Permite selectarea unei componente din fișierul Personal.dbf
	•	RowSourceType	2 - Alias	
		RowSource	Personal.marca+ Personal.nume	Afişarea listei se face pe două coloane
GrdCalificative	Grid	FontSize	10	Liniile listei se adaugă prin acționarea butonului <i>Adaug</i> ă
	Header1	Caption	Numele persoanei	
		Alignment	2 – Middle Center	La fel și pentru celelalte câmpuri
	Text1	Enabled	.F False	La fel pentru coloana 4

 Se stabilesc metodele (secvenţele de cod) folosind pagina Method din fereastra Properties:

Obiect	Metoda	Secvența de cod	
ab GrdCalificative,	Lost Focus	repl media with (nota1+nota2)/2	
Column2,			
TEXLI		-câmpul media rezultă din calcul	
		- la fel pentru GrdCalificative, Column3, Text1	
	Click	append blank	
Adaugă	Event	repl nume with personal.marca+personal.nume	
		thistorm.retresh	
		-introduce în lista marca și numele persoanel	
	Click	ThisForm.Release	
Stop	Event	-dezactivează (închide formularul)	

- Se stabilește ordinea de activare a obiectelor prin acționarea butonului **Set Tab Order.**
- Se salvează și se execută formularul astfel creat.

🗐 5. Proiectarea meniurilor

Meniurile sunt obiecte de interfață care pun la dispoziția utilizatorului o serie de opțiuni care permit alegerea și declanșarea unor operații de prelucrare. Aproape orice interfață grafică conține, într-o formă sau alta, un meniu. Un meniu apare, de obicei, în partea superioară a ferestrelor de aplicații și are o structură standard, iar în general, orice operație care poate fi executată în cadrul unei aplicații trebuie să aibă un corespondent în meniul acesteia, pentru a putea fi executată. Utilizatorul își construiește un model mental al structurii aplicației prin simpla examinare a meniurilor și a articolelor de meniu, meniul reprezentând un mijloc prin care utilizatorul comunică sistemului ce operație să efectueze la un moment dat.

Meniu orizontal		Opțiune selectată din meniul orizontal							
(Meniu Bar)					Opţi	iune s	electată	din subme	niu
Firm <u>a F</u> isiere <u>C</u> umpar		<u>C</u> umparari	<u>V</u> anzari	Avi <u>z</u> e	Ajutor				
		<u>F</u> acturi	definitiv	e prin le	ai	CTRL+L			
			Facturi definitive prin <u>v</u> aluta			CTRL+V			
	Submeniu Vertical (Submenu)			Facturi Bonuri <u>I</u> ncasa	in <u>n</u> ume de casa ri clienti	rar		CTRL+B	
				<u>L</u> ista d	ocument	e vanz	ari		

Un meniu este format, în general, dintr-o bară (sau un submeniu orizontal) care conține mai multe opțiuni. Fiecare dintre acestea are asociat un submeniu vertical, care este activat numai la alegerea opțiunii respective. Alegerea de către utilizator a opțiunilor meniurilor verticale poate declanșa o operație de prelucrare sau poate determina afișarea unui nou submeniu vertical.

🗐 5.1. Crearea și deschiderea unui meniu

Dente fi inițiată prin comanda:

MODIFY MENU [<nume meniu>]

sau prin alegerea opțiunii <u>F</u>ile, <u>N</u>ew din **meniu** (în acest caz în caseta <u>New</u> se selectează <u>M</u>enu și <u>N</u>ew file). Din fereastra deschisă pe ecran se alege butonul **Menu** (meniu), dacă se dorește crearea unui meniu cu bară și submeniuri verticale, respectiv butonul **Shortcut** (scurtătură), dacă se dorește construirea unui submeniu vertical, având subordonate alte meniuri verticale. Astfel va fi lansat în lucru **Constructorul de meniuri**.

Activarea (deschiderea) unui meniu care a fost deja generat se poate realiza prin selectarea acestuia (*File, Open, Files of type: Menu (*.mxx))* sau prin comanda MODIFY MENU.

Înainte de a stabili componentele unui meniu și caracteristicile acestora trebuie precizate o serie de opțiuni, cum ar fi poziția față de meniul sistemului Visual FoxPro. Opțiunile generale ale unui meniu pot fi precizate în fereastra de dialog **General Options** (opțiuni generale): <u>V</u>iew, <u>G</u>eneral Options... din meniul Visual FoxPro-ului.

🗐 5.2. Definirea componentelor unui meniu

Butoanele din secțiunea *Location* (locație) stabilesc poziția noului meniu față de meniul sistem:

- <u>R</u>eplace înlocuirea meniului sistem;
- <u>Append</u> adăugare;

ad

Sfat

- <u>B</u>efore plasarea noului meniu înaintea submeniului sistemului specificat în lista derulantă alăturată (care este afişată numai la selectarea butonului);
- <u>A</u>fter noul meniu va fi plasat după submeniul specificat.

👹 General Options		×
Procedure:		
	×	OK Cancel Edi <u>t</u>
Location <u>B</u> eplace <u>Append</u> <u>B</u> efore <u>C</u> After	Menu code Setup Cleanup	

Se pot specifica secvențe de cod (*Meniu code*) la inițializarea meniului (crearea sa) și ștergerea meniului (eliminarea sa din memorie). Secvențele de cod respective se specifică în ferestre deschise prin activarea comutatoarelor **Setup** (inițializare) și *Cleanup* (ștergere), urmată de acționarea butonului OK.

În fereastra *Meniu Designer* sunt enumerate și descrise opțiunile meniului creat și o serie de obiecte de interfață folosite la

 Prompt	Result		Options	Menu level:
\$ Firm\ <a< th=""><th>Submenu 💌</th><th>Edit</th><th></th><th>Menu Bar 📃 💌</th></a<>	Submenu 💌	Edit		Menu Bar 📃 💌
\ <fisiere< th=""><th>Submenu</th><th></th><th></th><th>ltere</th></fisiere<>	Submenu			ltere
\ <cumparari< th=""><th>Submenu</th><th></th><th></th><th>I tem</th></cumparari<>	Submenu			I tem
\ <vanzari< th=""><th>Submenu</th><th></th><th></th><th>Insert</th></vanzari<>	Submenu			Insert
Avi\ <ze< th=""><th>Submenu</th><th></th><th></th><th>Insert Bar</th></ze<>	Submenu			Insert Bar
A\ <jutor< th=""><th>Submenu</th><th></th><th></th><th></th></jutor<>	Submenu			
				Delete
				Proview
			-	LIGNEW

manipularea acestor opțiuni. La un moment dat, în fereastra de lucru, este afișat numai grupul de opțiuni al unui subsistem (orizontal sau vertical) al meniului proiectat. O opțiune poate fi folosită pentru accesul la un alt submeniu sau poate fi dedicată declanșării unei acțiuni. Pentru a reveni la meniul imediat superior se folosește lista derulantă **Menu level** (nivel meniu), în care sunt disponibile toate meniurile superioare ierarhic meniului curent.

În fereastra *Constructorului de meniuri,* pe coloane, se specifică caracteristicile opțiunilor, astfel:

 Coloana *Prompt* cuprinde textele informative ale opţiunilor în care se pot introduce şi combinaţii speciale, prin intermediul cărora se obţin diferite efecte:

Combinația	Efect
\<	Dacă este plasată înaintea unuia dintre caracterele șirului
	face ca acel caracter să fie folosit ca o tastă pentru
	meniul este activat Caracterul respective, atunci canu
	în textul opțiunii
Λ	Determină dezactivarea opțiunii respective. Aceasta
	apare pe ecran în culori șterse, iar alegerea sa nu
	declanșează nici o acțiune (chiar dacă în cazul său au fost
	specificate operații de executat)
\-	Trasarea unei bare delimitatoare între mai multe grupuri
	de optiuni ale aceluiasi submeniu

 Coloana Result este destinată precizării operaţiunilor care vor fi executate la selectarea acestora:

- Submenu activarea unui nou submeniu conţinând propriul grup de opţiuni. Coloana a treia va conţine un buton (Create sau Edit) la a cărei acţionare se va intra în editarea submeniului respectiv;
- Command permite lansarea unei comenzi FoxPro specificată în câmpul de editare din dreapta listei (coloana a treia);
- **Procedure** executarea mai multor comenzi grupate într-o procedură. Butonul din dreapta listei permite intrarea în editarea procedurii respective.

3. Coloana a treia specifică operațiunea de executat la alegerea opțiunii.

4. Coloana **Options** (opțiuni) este utilizată pentru deschiderea ferestrei de dialog **Prompt Options**, pentru specificarea unor parametri suplimentari ai opțiunii respective cum ar fi:

😽 Prompt Options		×
Shortcut Key L <u>a</u> bel: CTRL+L Key <u>I</u> ext: CTRL+L		Negotiate None C Left C Middle C Right
Skip For:		
Message:		
<u>B</u> ar #:		
<u>C</u> omment:		
	OK	Cancel

• **Shortcut** (scurtătură) - cale directă de alegere. Un *Shortcut* conține:

- Câmpul de editare *Key Label* (etichetă tastă), pentru a indica faptul că opțiunea poate fi aleasă direct prin utilizarea unei combinații de taste. Combinația de taste dorită este introdusă în caseta *Key Label* prin apăsarea combinației de taste respective.
- Câmpul Key Text (text tastă) este un text suplimentar ce poate fi ataşat opțiunii. În mod implicit Visual FoxPro repetă tastele de comandă rapidă din caseta Key Label în caseta Key Text, dar acesta poate fi modificat (de exemplu combinația CTRL+L poate fi modificată în ^L).
- Skip For (sărită când...) permite impunerea unei condiții prin care să se controleze accesul la o opțiune. În acest câmp este introdusă o expresie logică. Dacă valoarea ei este adevărat, atunci opțiunea va fi dezactivată, deci nu va fi disponibila pentru utilizator (ea apare cu culori şterse, pentru a indica această stare). Dacă valoarea expresiei este *fals,* opțiunea va putea fi aleasă de utilizator. De exemplu, pentru ca o opțiune să

fie accesibilă numai dacă un fișier există pe disc condiția introdusă poate fi: *NOT(FILF('Furnizor.dbf)).*

• **Message** (mesaj) - are ca efect apariția unui mesaj explicativ în bara de stare a Visual FoxPro-ului (plasată în partea inferioară a ferestrei mediului de programare).

🗐 5.3. Previzualizarea meniului

Este recomandabil ca, în faza de editare a unui meniu, să se facă afișarea de probă a acestuia, chiar dacă el nu a fost salvat sau programul corespunzător nu a fost încă generat. Acest procedeu se numește **previzualizare.** La previzualizare, alegerea unei opțiuni a meniului nu este însoțită de execuția comenzii corespunzătoare, ci doar de afișarea acesteia. Previzualizarea meniului în curs de editare se realizează prin acționarea butonului de meniuri sau utilizând meniul principal al sistemului: <u>M</u>enu, <u>P</u>review. În starea de previzualizare, meniul sistemului este înlocuit de meniul în curs de editare, iar pe ecran este afișată o fereastră care indică opțiunea selectată din acest meniu.

🗐 5.4. Generarea și rularea meniurilor

Un meniu editat poate fi salvat ca un fișier cu extensia implicită **.MNX.** Pentru a putea folosi meniul, trebuie generat un program la a cărui rulare meniul să fie activat. Această operație poate fi realizată din mediul Visual FoxPro: <u>M</u>enu, <u>G</u>enerate... În fereastra deschisă pe ecran se precizează numele fișierului ce va fi generat:

👹 Generate	Menu	×
Output <u>F</u> ile:	C:\program Files\devstudio\vfp\job\meniuri.mpr	
	Generate Cancel	

În mod implicit, acesta are același nume cu cel al fișierului în care este memorat meniul, dar extensia **.MPR.**

O dată generat meniul, acesta poate fi rulat printr-o comandă

DO. Deoarece comanda *DO* presupune extensia implicită **PRG** (de la programe), este necesară specificarea extensiei **.MPR**:

DO <meniu>.MPR

Observație

Revenirea la meniul standard al mediului de programare Visual FoxPro se realizează prin comanda:

SET SYSMENU TO DEFAULT

Aplicație

Să se editeze un meniu pentru **crearea** și **modificarea** structurii unui fișier cât și pentru **editarea** conținutului lui:

Fisier Stop	
<u>⊂</u> reare <u>M</u> odificare	CTRL+C CTRL+M
<u>E</u> ditare	CTRL+E

Indicaţii

- Se lansează Menu Designer.
- Se editează opțiunile meniului principal.

Prompt	Result	Options Menu level:
\ <fisier< th=""><th>Submenu</th><th>📩 Menu Bar 📃</th></fisier<>	Submenu	📩 Menu Bar 📃
\ <stop< th=""><th>Procedure</th><th></th></stop<>	Procedure	

• Opțiunea Stop are asociată următoarea procedură:

Procedure
close all
set sysmenu to default

Se editează opțiunea **Fisier** din meniu.

	Prompt	Result	Op	tions	Menu level:	
1	Creare	Command 💌	create	✓	Fisier	•
	\ <modificare< th=""><th>Command</th><th>modify structure</th><th>√</th><th></th><th></th></modificare<>	Command	modify structure	√		
	٨.	Submenu				
Γ	\ <editare< th=""><th>Command</th><th>browse</th><th>\checkmark</th><th></th><th></th></editare<>	Command	browse	\checkmark		

Se previzualizează, se generează și se rulează meniul creat.

6. Generarea rapoartelor

Raportul reprezintă un set de informații, constituite pe baza datelor introduse anterior în bazele de date, care urmează a fi tipărit la imprimantă sau afișat pe display-ul calculatorului. Definirea unui raport se realizează, ca și în cazul formularelor, prin utilizarea tehnologiei **POO** (**P**rogramare **O**rientată **O**biect) care permite definirea și modificare proprietăților obiectelor introduse în raport.

Componenta Visual FoxPro-ului care permite crearea și modificarea rapoartelor precum și generarea secvenței de cod necesare pentru execuția acestora este **Report Designer**.

🗐 6.1. Crearea și deschiderea unui raport

Crearea unui raport (fișier cu **extensia implicită .FRX**) poate fi inițiată prin comanda:

CREATE REPORT [< nume raport>]

sau prin alegerea opțiunii <u>F</u>ile, <u>N</u>ew din **meniu** (în acest caz în caseta New se selectează <u>R</u>eport și <u>N</u>ew file).

Activarea (deschiderea) unui raport se poate realiza prin meniu (*<u>F</u>ile, <u>O</u>pen, Files of <u>type: Report (*.frx</u>)) sau prin comanda:*

MODIFY REPORT < nume raport>

le Observație

Prin lansarea Constructorului de rapoarte, în fereastra

<u>R</u> eport	
<u> </u>	/Summary
Data	a <u>G</u> rouping
<u>V</u> aria	ables
Defa	ault <u>F</u> ont
<u>P</u> riva	ate Data Session
Quic	k Report
<u>R</u> un	Report

Report Designer, meniul principal va include opțiunea **Report** care conține opțiuni referitoare la diferite operații.

🗍 6.2. Imprimarea și afișarea rapoartelor

Un raport este un fișier în care sunt memorate elementele sale precum și caracteristicile acestora. **Imprimarea (Print) unui raport** poate fi realizată în unul din următoarele moduri:

- din meniul principal (<u>Report, Run Report</u>),
- prin acţionarea butonului
 sau a butonului
 din linia
 de instrumente,

prin comanda:

REPORT FORM < nume raport > TO PRINTER [PROMPT]

(clauza PROMPT este folosită pentru specificarea parametrilor de tipărire)

Pentru **afişarea sau previzualizarea (Preview)** pe ecran se poate utiliza:

- butonul din linia de instrumente standard,
- comanda:

REPORT FORM < nume raport> PREVIEW

Vizualizarea informațiilor din raport se realizează utilizând bara utilitară de mai jos:



占 Observație

Se poate obține trimiterea raportului spre un fișier (în loc de imprimantă) cu ajutorul clauzei TO FILE a comenzii REPORT FORM:

REPORT FORM < nume raport > TO FILE < fişier > [ASCII]

Clauza ASCII determină modul de scriere în fișier: în prezența clauzei formatul fișierului va fi text, iar în absența acesteia formatul va fi unul specific imprimantei (informațiile se depozitează în fișierul respectiv, exact cum s-ar trimite la imprimantă).

6.3. Constructorul de rapoarte

- Report Designer fereastra principală care conține obiectele raportului.
- Data Environment mediul de date al raportului, fereastra pentru precizarea tabelelor de date care vor fi deschise automat în momentul rulării raportului.
- Barele utilitare
 - Report Designer Toolbar bara de butoane standard a raportului:





Butoanele din această bară de instrumente permit deschiderea/închiderea ferestrelor: *Data Environment*, *Data Grouping* (gruparea datelor) sau a barelor de lucru:

- Report Control Toolbar butoanele de control;
- **Color Palette Toolbar** stabilirea atributelor de culoare;
- Layout Toolbar alinierea obiectelor în formular, suprapunerea obiectelor, etc.

🖑 Sfat

Lucrați cu **barele de instrumente** asociate Constructorului de rapoarte. Dacă acestea nu sunt afișate utilizați meniul: <u>V</u>iew, <u>T</u>oolbars...

O dată pornit Constructorul de rapoarte, pe ecran este afișată fereastra de editarea a raportului.

Un raport este constituit din benzi care nu redau exact ceea ce va fi în raportul final, dar codifică modul în care se determină ceea ce se afișează în raport. Un raport poate conține următoarele benzi:

- Detail (detalii) generează rândurile de detalii (conţinutul de bază al raportului). Pentru un raport care preia datele dintr-o tabelă, banda de detalii a raportului ar putea corespunde înregistrărilor tabelei;
- Title (titlu) include o zonă care se afişează o singură dată, pe prima pe pagină a raportului. Această bandă conţine de obicei titlul raportului şi eventual, capul de tabel, dacă acesta nu trebuie repetat pe fiecare pagină în parte);
- Summary (rezumat) specifică o zonă care este afişată pe ultima pagină a raportului, la sfârşit, o singură dată;
- Page Header (antet de pagină) generează textul care va fi afişat în partea superioară a fiecărei pagini a raportului;
- Page Footer (subsol de pagină) generează conţinutul părţii inferioare a fiecărei pagini a raportului;
- **Group Header** *n* (antetul grupului de nivel *n*) și **Group Footer**

📓 Report Designer - Report1
▲ Title
T.
A Page Header
🔺 Detail
A Page Footer
🔺 Summary 📃 🖓
n (subsolul grupului de nivel *n*) - generează în raportul final mai multe grupuri, în funcție de niște valori numite chei de grupare, putându-se preciza conținutul benzii antet de grup, respectiv subsol de grup;

Un raport conține cel puțin trei benzi: antetul de pagină, detaliile și subsolul de pagină. Dacă se dorește inhibarea afișării lor, trebuie reduse la dimensiunea zero (chiar dacă apar în fereastra raportului, nu generează nimic în raportul final). Benzile de titlu și de concluzii sunt afișate opțional, afișarea lor putând fi validată sau invalidată prin meniu: **<u>R</u>**eport, <u>**T**</u>itle/Summary</u>.

Fiecare dintre aceste două benzi are asociat un comutator, *New page* (pagină nouă), la activarea căruia banda corespunzătoare va începe pe pagină nouă:

😽 Title/Summary	×
Report title	OK Cancel
Report summary Summary band New page	

Mediul de date al raportului

Un raport preia date din una sau mai multe tabele și le aranjează în pagină. Deci la construirea raportului este necesară specificarea tabelelor sursă. Mediul de date al unui raport se configurează în fereastra mediului de date deschisă prin meniu: <u>V</u>iew, Data <u>E</u>nvironment sau prin acționarea butonului corespunzător din bara de instrumente standard a raportului. O dată deschisă această fereastră, la meniul sistemului este adăugat un nou submeniu, numit <u>D</u>ataEnvironment, care conține opțiuni referitoare la configurarea mediului de date.

Fontul implicit al raportului

La construirea unui raport, trebuie stabilit ca font implicit cel care predomină printre obiectele raportului, pentru celelalte obiecte fontul urmând a fi specificat explicit. Fiecărui obiect nou definit în raport îi va fi atribuit fontul implicit al raportului, care poate fi schimbat ulterior.

Stabilirea fontului implicit al raportului se face prin meniu: <u>**R**</u>eport, Default <u>**F**</u>ont, iar caracteristicile fontului sunt specificate în caseta de dialog deschisă pe ecran.

Tipuri de obiecte incluse în rapoarte

Editarea unui raport trebuie să înceapă prin introducerea unor obiecte în acesta. Pentru aceasta, se acționează butonul corespunzător din *Report Controls Toolbar* și se selectează obiectul care va fi inclus în raport:



A Label – Texte informative.

Textele informative reprezintă un șir de caractere de lungime variabilă. Pentru a stabili poziția textului informativ se poziționează cursorul în punctul dorit din raport, după care se execută clic cu mouse-ul. Când se termină introducerea textului, se execută un clic în afara obiectului nou creat.

Un obiect *Label* creat poate fi repoziționat, prin tragere cu ajutorul mouse-ului, pe suprafața raportului. Se pot efectua și trageri dintr-o bandă în alta, nu numai în cadrul aceleiași benzi.

Fontul folosit la afișarea textului este stabilit în ferestrea Font,

care se deschide prin meniu: Format, Format, Format, Astfel, pot fi specificate toate caracteristicile fontului: setul caractere (**F**ont), dimensiunea (**S**ize), stilul (Font style), culoarea (**C**olor), etc.



Câmpurile reprezintă elementele variabile ale unui raport, putând genera în raportul final mai multe valori, valori care vor fi determinate de sursa de date precizată: *o variabilă a raportului, o expresie de calcul, un câmp al unui tabel de date.* Caracteristicile legate de *fontul* în care se realizează afișarea valorii din câmp se stabilesc la fel ca pentru obiectele de tip *Label*.

Definirea caracteristicilor unui obiect de tip *Field* se realizează în fereastra de dialog **Report Expression** (deschisă prin dublu clic pe obiect).

Poziția câmpului (*Field position*) relativă la banda din raport în care este definită poate fi: *F***I**oat (poziție variabilă, în funcție de obiectele definite deasupra lui), *Fix relative to* **t**op of band (poziție fixă relativ la marginea superioară a benzii), *Fix relative to* **b**ottom of band (poziție fixă relativ la latura inferioară a benzii).

În caseta <u>Expression</u> se înscrie expresia care va furniza valoarea ce trebuie afișată. De multe ori, această expresie este un simplu câmp al unei tabele sursă sau o variabilă de lucru.

💓 Report E	xpression	×
<u>E</u> xpression:	mfix.denumire	ОК
<u>F</u> ormat:		Cancel
Field positio	n	
	🔿 Fjoat	<u>C</u> alculations
	• Fix relative to top of band	Print When
	○ Fix relative to <u>b</u> ottom of band	
□ <u>S</u> tretch w	ith overflow	

Formatul de afișare al câmpului (caseta *<u>F</u>ormat*) reprezintă un șir de caractere (macheta de formatare). Acest format se poate stabili și în mod interactiv, prin acționarea butonului din dreapta casetei, care determină deschiderea ferestrei **Format**.

Asupra valorilor asociate obiectelor *Field* se pot aplica diferite funcții pentru a obține unele informații sintetice: suma (*Sum*), media aritmetică (*Average*), valoarea maximă și minimă (*Highest*, *Lowest*), deviația standard (*Standard deviation*), varianța (*Variance*). Funcția care se aplică asupra unui câmp al unui raport este precizată în fereastra de dialog **Calculate Field**, deschisă prin acționarea butonului *Calculations:*

Reset: End of Report	
Calculate Car © Nothing © Count © Sum © Average © Lowest © Highest © Standard deviation © Variance	K



În cadrul unui raport pot fi desenate linii, chenare sau pot fi introduse imagini. În legătură cu aceste obiecte pot fi precizate anumite caracteristici cum ar fi **Pen** (peniță) – *Format Pen,* **Fill** (umplere) – *Format Fill*, etc.

Aplicație

1º Să se creeze tabelulul de date, **mfix.dbf**

Denumire	Valinv	Valamor
Calculator Pentium III 1,2 GHz	15650000	7042500
Calculator Pentium IV 1,5 GHz	17850000	8032500
Calculator AMD_Athlon_XP 1600+	14200000	6390000
Imprimanta A4 HP 656	1950000	877500
Fax Modem ROBOTICS 56k	1277000	574650

având structura: *Denumirea mijlocului fix* DENUMIRE C 30 *Valoarea de inventar* VALINV N 10 *Valoarea amortizată* VALAMOR N 10

2º Să se genereze, pe baza tabelului de date definit anterior următorul raport:

Situatia mijloacelor fixe				
	Data curenta	18/06/2003		
Denumire Val. inventar Val. amorti:				
Calculator Pentium III 1,2 GHz	15650000	7042500		
Calculator Pentium IV 1,5 GHz	17850000	8032500		
Calculator AMD Athlon XP 1600+	14200000	6390000		
Imprimanta A4 HP 656	1950000	877500		
Fax Modem ROBOTICS 56k	1277000	574650		
Tota	50927000	22917150		

Indicaţii

- Se lansează Constructorul de rapoarte.
- În mediul de date al raportului se introduce tabelul de date definit anterior.
- Se includ în raport obiectele necesare și anume:
 - În banda **Page Header**:
 - *textele informative (denumirea raportului, capul de tabel),*
 - elementele semigrafice (linii, chenare),
 - câmpul care furnizează data calendaristică.

- În banda Detail se trag, cu mouse-ul, din mediul de date al raportului, câmpurile: denumire, valinv, valamor.
- În banda Summary se introduc câmpurile valinv, valamor. Pentru ca acestea să afişeze valori totalizatoare se execută câte un dublu clic pe ele pentru a deschide fereastra Report Expresion. Aici, se acţionează butonul Calculations şi se selectează funcţia Sum.

	Report Designer - mf.frx 0 1 2 3 4 5 6 7 8 9	10 11 12	13 14 15 16 ·
0.	Situatia mijloacelor fixe		
2	Data curenta	a date()	
3 .	Denumire	Val. inventar	Val. amortizata
	A Page Header		
0	⁰ denumire va	alinv	valamor
	Letail		
	A Page Footer		
0 -	Total Va	alinv	valamor
	Summary		_ _
•			▶ <i> </i> .

3⁰ Să se previzualizeze și să se salveze raportul editat.

7. Crearea și editarea proiectelor

Proiectul reprezintă un fișier cu extensia implicită **.PJX** care permite gruparea într-o formă unitară a componentelor unui sistem informatic cum ar fi baze de date, tabele, programe, formulare, meniuri, rapoarte, interogări etc., pentru a ține evidența acestora. Proiectarea unei aplicații înseamnă, în general, crearea unui fișier proiect la care sunt adăugate pe rând componentele sale. Mediul de programare Visual FoxPro permite ca toate operațiile care se efectuează asupra elementului unui proiect să poată fi declanșate din interiorul proiectului, prin intermediul resursei numită **Gestionar de proiecte**.

Prin generarea proiectelor se creează posibilitatea desfășurării unor serii de operații care permit:

- actualizarea automată a proiectului, atunci când între forma sursă și cea compilată se constată desincronizări;
- o crearea fișierelor de aplicații și a programelor executabile.

🗐 7.1. Crearea și deschiderea unui proiect

Crearea sau deschiderea unui proiect este declanşată prin comanda:

MODIFY PROJECT [< nume proiect>]

sau prin alegerea opțiunii <u>F</u>ile, <u>N</u>ew din **meniu** (în acest caz în caseta <u>New</u> se selectează <u>P</u>roject și <u>N</u>ew file). Astfel este deschisă fereastra Gestionarului de proiecte:

Project I	Manager -	Conturi				×
All	Data	Documents	Classes	Code	Other	+
	Data Document Forms B Repor Class Libra Class Libra Code Other	ts ontract ts s aries			<u>N</u> ew <u>A</u> dd <u>M</u> odify R <u>u</u> n Remo <u>v</u> e Buil <u>d</u>	
Descrip Path:	tion: c:\pi	rogram files\devstu	udio/vfp/job/co	ontract.scx		

Fereastra include o listă cu elementele componente ale proiectului, grupate pe categorii (aceste categorii apar și în paginile alternative ale ferestrei):

Categoria	Tipul elementului
Data Databases Free Tables	 baze de date tabele izolate interogări
⊡ I Documents ⊕ III Forms III Reports IIII Labels	 formulare rapoarte etichete
Code Programs API Libraries 01 Applications	 programe biblioteci API aplicaţii

7.2. Editarea proiectelor

Editarea proiectelor este realizată cu ajutorul grupului de butoane din partea dreaptă a ferestrei *Project Manager*. Astfel, după selectarea din categoria sa a tipului elementului se acționează unul din butoanele:

- New adăugarea unui element nou la proiect (va fi pornit Constructorul asociat tipului de element specificat).
- Add dacă elementul este deja creat şi trebuie doar adăugat la proiect. Din fereastra de dialog deschisă pe ecran se selectează fişierul corespunzător.
- Modify editarea (modificarea) unui element al proiectului (din Gestionarul de proiecte se porneşte direct Constructorul asociat unui anumit tip de element).
- **Run** rularea unui element (program, formular, meniu etc.).
- Remove eliminarea unui element din proiect din lista Gestionarului de proiecte (înlăturarea unui element dintr-un proiect nu înseamnă și ștergerea fizică a acestuia de pe disc, ci doar ștergerea indicatorului corespunzător din tabela proiectului).
- Build (construire) compilarea componentelor prin stabilirea unor opţiuni:

Compilarea elementelor unui proiect are ca efect crearea a două tipuri de fișiere:

- fișiere de tip aplicație care pot fi rulate direct, sub controlul mediului de programare (<u>B</u>uild Application);
- fişiere executabile independent de mediul Visual FoxPro (Build <u>Executable</u>).

👹 Build Options	×
Action <u> R</u> ebuild Project	OK)
C <u>B</u> uild Application	Cancel
Build <u>E</u> xecutable	Help
O Build OLE DLL	
C Options	I
🗖 Recompile <u>A</u> ll Files	
Display Errors	
🗖 R <u>u</u> n After Build	<u>V</u> ersion

Vor fi compilate componentele noi sau cele care au suferit modificări de la ultima lor compilare. Pot fi recompilare toate componentele proiectului, atunci când se activează comutatorul *Recompile* <u>**A**</u>*II Files* din secțiunea *Options*.

🗐 7.3. Generarea fișierelor aplicație

Aplicațiile reprezintă o forma a proiectului, generată într-un singur fișier, de tip aplicație, care poate fi executat direct de sistem și care conține programele, formularele, rapoartele și celelalte elemente ale proiectului. Acest fișier are extensia implicită **.APP**.

Stabilirea fișierelor modificabile

Din punctul de vedere al incorporării efective în fișierul aplicație, elementele unui proiect se împart în două grupe:

- fişiere incluse (elemente care fac parte din proiect şi sunt incluse în fişierul aplicaţie);
- fişiere excluse (elemente care fac parte din proiect, dar nu sunt incluse în fişierul aplicație).

Fişierele excluse au în stânga numelor lor simbolul \emptyset :



Marcarea unui fișier ca *"exclus" ("inclus")* se realizează prin meniu: *Project, Exclude (Include)*.

Fişierele incluse în proiect sunt asamblate într-un singur fişier aplicație. După generarea proiectului, fișierele marcate ca *"incluse"* sunt protejate la scriere. Unele fișiere care fac parte dintr-un proiect, cum ar fi tabelele, pot fi modificate de către de utilizator. În acest caz, fișierele trebuie adăugate la proiect, dar vor fi marcate ca *"excluse".* Fișierele excluse sunt parte a proiectului, dar ele nu sunt compilate în fișierul aplicație, așa încât se pot actualiza la utilizare.

占 Observație

În mod implicit, tabelele sunt marcate ca excluse, deoarece Visual FoxPro presupune că acestea vor fi modificate în cadrul aplicației. Ca regulă generală, elementele unui proiect care au ca rezultat generarea de cod (formulare, rapoarte, interogări, meniuri și programe) trebuie incluse în fișierul aplicație, pe când fișierele de date trebuie excluse. Stabilirea fișierelor incluse, respectiv excluse, se realizează pe baza cerințelor aplicației. De pildă, o tabelă care cuprinde informații de sistem importante sau informații folosite pentru căutare pot fi incluse în fișierul aplicație pentru a le proteja împotriva modificărilor neavenite. Invers, un fișier raport (*.frx*) poate fi exclus pentru a permite modificarea lui dinamică în timpul rulării.

Punctul de pornire al aplicaţiei

Înainte de executarea unei aplicații trebuie stabilit fișierul principal al acesteia, adică acea componentă care va fi lansată atunci când se execută comanda de rulare a aplicației. Stabilirea fișierului principal (program, formular, meniu) se face prin selectarea acestuia din lista Gestionarului de proiecte și alegerea opțiunii <u>Set Main</u> a meniului *Project*. În listă, programul principal va fi afișat cu caractere aldine:



🗕 Observație

Fișierul desemnat ca fișier principal al aplicației este marcat ca parte a proiectului, astfel că după compilarea aplicației el va fi protejat la scriere (read-only). Într-un proiect nu poate exista decât un singur fișier principal.

🗐 Rularea unei aplicații

O aplicație poate fi rulată dacă este activat comutatorul *R<u>u</u>n After Build* din fereastra *Build Options* sau prin comanda **Do** (în acest caz este necesară precizarea explicită a extensiei fișierului):

DO <nume aplicație>.APP

7.4. Generarea fişierelor executabile

Prin facilitatea de construire a programelor executabile se obține o independență a programelor față de mediul Visual FoxPro și distribuirea lor independentă. Generarea unui fișier executabil (*Build* **E**xecutable) pe baza unui proiect se face la fel ca în cazul aplicației, fiind necesară construirea proiectului corespunzător în care trebuie introduse toate elementele acestuia (fișierele incluse și excluse, fișierul principal).

Un fişier executabil (cu extensia **.EXE)** poate fi rulat independent de mediul Visual FoxPro, fiind totuşi necesare o serie de biblioteci de proceduri ale sistemului, acesta funcţionând în cooperare cu două biblioteci de legături dinamice - Vfp6r.dll (rVfp500.dll în Visual FoxPro 5) şi Vfp6enu.dll - care trebuie asociate aplicaţiei pentru a oferi un mediu Visual FoxPro complet şi coerent.

Controlul ciclului de evenimente

Controlul ciclului de evenimente se referă la prelucrarea evenimentelor generate de utilizator, cum ar fi clicurile de mouse și apăsările de taste și este realizat prin comenzile:

READ EVENTS și CLEAR EVENTS

Plasarea corectă a comenzii READ EVENTS în cadrul fișierului principal prezintă importanță, deoarece prelucrarea datelor în fișierul principal este oprită din momentul executării comenzii READ EVENTS până la lansarea unei comenzi CLEAR EVENTS. Astfel se poate lansa READ EVENTS ca ultimă comandă într-o procedură de inițializare, executată după ce mediul de dezvoltare a fost inițializat și interfața cu utilizatorul a fost afișată. Dacă nu se include comanda READ EVENTS, după rulare, aplicația va returna controlul sistemului de operare (aplicația va fi rulată corect din fereastra *Command* a mediului de dezvoltare, în schimb, dacă ea va fi lansată independent de mediul Visual FoxPro, aplicația va apărea pentru scurt timp, apoi va dispărea de pe ecran). De asemenea, aplicația trebuie să ofere o modalitate de a încheia ciclul de evenimente.

Comanda CLEAR EVENTS suspendă prelucrarea evenimentelor de către Visual FoxPro și cedează controlul către programul care a lansat comanda READ EVENTS ce a declanșat ciclul de evenimente.

🕑 Sfaturi

- Includeți comanda READ EVENTS în fişierul principal. Dacă acesta este
 - un formular: în metoda Activate Event;
 - un meniu: în procedura <u>C</u>leanup (meniu: <u>V</u>iew, <u>G</u>eneral Options...).
- Asigurați-vă că interfața are un mecanism (cum ar fi un buton *Exit* sau o comandă de meniu) pentru a lansa comanda CLEAR EVENTS.

Rularea unui fişier executabil

Un fișier .EXE poate fi rulat dacă este activat comutatorul *R*<u>u</u>*n After* <u>Build</u> din fereastra <u>Build</u> *Options* sau prin comanda **Do**:

```
DO <nume aplicație>.EXE
```

Aplicație

1⁰ Să se editeze un proiect, pentru evidența studenților înscriși la o facultate care să includă:

a) fișierul INSCRISI.DBF cu structura:

•	Număr legitimație	LEG	N 3
•	Numele candidatului	NUME	C 15
	Nota proba 1	N1	N 5,2
•	Nota proba 2	N2	N 5,2
	Media	Media	N 5,2

b) următorul meniu (marcat ca fișier principal):

<u>F</u> isier	<u>N</u> ote	lesi	ire
<u>A</u> ctualizare	Note proba <u>1</u>	CTRL+A	
<u>L</u> istare	Note proba <u>2</u>	CTRL+B	
	<u>M</u> edia	CTRL+M	

c) un formular, pentru actualizarea tabelului de date (opțiunea *E*isier, <u>A</u>ctualizare);

d) un raport (*Report*), pentru listarea tabelului de date (opțiunea *F*isier, *L*istare);

e) O interogarea (*Query*), pentru afişarea, în ordine descrescătoarea, a notelor obținute la proba 1 (opțiunea <u>N</u>ote, Note proba<u>1</u>);

f) O interogarea, pentru afişarea, în ordine descrescătoarea, a notelor obţinute la proba 2 (opţiunea <u>N</u>ote, Note proba<u>2</u>);

g) O interogare pentru afişarea descendentă a mediilor candidaților admiși (opțiunea <u>N</u>ote, <u>M</u>edia);

Indicație

Se va include comanda **SET SYSMENU TO DEFAULT** în procedura atașată opțiunii <u>*Iesire*</u>

3⁰ Să se genereze și să se ruleze fișierul executabil.

Indicații

- a) Se va include comanda **READ EVENTS** în procedura <u>*C*</u>*leanup* asociată meniului (<u>*V*</u>*iew*, <u>*G*</u>*eneral Options...*).
- b) Se va include comanda **CLEAR EVENTS** în procedura ataşată opțiunii <u>I</u>esire

²º Să se genereze și să se ruleze fișierul aplicație.