

Mariana Miloșescu**MODULUL 2****SGBD**

– Sisteme de gestiune a bazelor de date

B. Programare

Visual FoxPro (Microsoft)

Modul opțional

Adresabilitate

– la specializarea *matematică - informatică*

● Varianta I (BD + SGBD)

– la specializarea *matematică - informatică,*
intensiv informatică

● Varianta I (BD + SGBD + POO și PV)

● Varianta IV (BD + SGBD + Programare web)

Manual pentru clasa a XII-a

Filiera teoretică, profil real,
specializarea: Matematică - informatică

Filiera vocatională, profil militar,
specializarea: Matematică - informatică

(3 ore activități practice) / săptămână



EDITURA DIDACTICĂ ȘI PEDAGOGICĂ S.A.

Cuprins

2. Sisteme de gestiune a bazelor de date	3
2.1. Mediul de programare Microsoft Visual FoxPro	4
2.1.1. Moduri de lucru în Visual FoxPro	6
2.1.1.1. Modul de lucru interactive	6
2.1.1.2. Modul de lucru programat	11
2.1.1.3. Sintaxa comenzilor de configurare a mediului Visual FoxPro	12
2.1.1.3. Configurarea mediului Visual FoxPro pentru tipuri de date	13
2.1.1.4. Redirectarea ieșirii în mediul Visual FoxPro	13
2.1.1.5. Configurarea interactivă a mediului Visual FoxPro	13
2.1.1.6. Comenzi pentru gestionarea fișierelor în Visual FoxPro	14
2.2. Tipuri de date	20
2.2.1. Tipuri numerice	21
2.2.2. Tipuri alfanumerice	22
2.2.3. Tipul logic	23
2.2.3. Tipuri pentru gestionarea timpului	25
2.2.4. Tipul special	25
2.2.5. Crearea variabilelor de memorie	25
2.2.6. Funcții pentru date de tip numeric	28
2.2.7. Funcții pentru date de tip șir de caractere	30
2.2.8. Funcții pentru date folosite pentru gestionarea timpului	36
2.2.9. Funcții pentru conversii între tipuri de date	38
2.2.10. Funcții de uz general	40
2.2.11. Comenzi pentru variabile de memorie	42
2.2.12. Tablourile de memorie	44
2.3. Tabele	50
2.3.1. Crearea tabelului	53
2.3.1.1. Crearea structurii tabelului	54
2.3.1.2. Manipularea structurii tabelului	54
2.3.1.4. Manipularea câmpurilor dintr-un tabel	56
2.3.1.4. Câmpurile memo	56
2.3.2. Poziționarea pe o înregistrare din tabel	63
2.3.2.1. Poziționarea după numărul înregistrării	63
2.3.2.2. Poziționarea după un criteriu logic de căutare	64
2.3.2.3. Controlarea poziționării în tabel	64
2.3.3. Actualizarea tabelului	66
2.3.3.1. Adăugarea de noi înregistrări	67
2.3.3.2. Ștergerea înregistrărilor	67
2.3.3.4. Modificarea înregistrărilor	68
2.3.3.5. Controlarea accesului la înregistrări	70
2.3.4. Consultarea tabelului	72
2.3.5. Sortarea tabelului	75
2.3.6. Indexarea tabelului	76
2.3.6.1. Crearea unui index	76
2.3.6.2. Deschiderea unui index	77
2.3.6.3. Alte operații cu indecși	77
2.3.6.4. Exploatarea unui tabel împreună cu indexul	77
2.4. Baza de date	80
2.4.1. Crearea bazei de date	81

2.4.1.1. Crearea bazei de date vide	81
2.4.1.2. Adăugarea la baza de date a tabelelor vide	81
2.4.1.3. Adăugarea la baza de date a tabelelor noi	81
2.4.1.4. Obținerea informațiilor despre baza de date	81
2.4.1.5. Alte operații cu baza de date	82
2.4.2. Exploatarea bazei de date	82
2.4.2.1. Deschiderea și închiderea bazei de date	82
2.4.2.2. Deschiderea și închiderea tabelelor	82
2.5. Obținerea informațiilor din baza de date	90
2.5.1. Interogările	90
2.5.2. Vizualizările	91
2.5.3. Rapoartele	92
2.5.4. Etichetele	94
2.6. Limbajul SQL	103
2.6.1. Interogarea de selecție	106
2.6.2. Interogarea acțiune	109
2.7. Elemente de programare	111
2.7.1. Construirea programului	111
2.7.1.1. Crearea programului sursă	111
2.7.1.2. Compilarea programului sursă	112
2.7.1.3. Lansarea în execuție a programului	112
2.7.2. Programare structurată	112
2.7.2.1. Declararea datelor	112
2.7.2.2. Structurile de control	113
2.7.3. Modularizarea aplicațiilor	121
2.7.3.1. Controlul execuției modulelor	121
2.7.3.2. Domeniul de vizibilitate al identificatorilor	122
2.7.3.3. Parametrii de comunicație	123
2.7.3.4. Modul de transmitere a parameterilor	123
2.7.3.5. Subprogramele	124
2.7.3. Programare orientată pe obiecte	131
2.7.3.1. Principiile programării orientate pe obiecte	131
2.7.3.2. Clasele	132
2.7.3.3. Obiectele	135
2.7.3.4. Accesul la membrii unui obiect	137
2.7.4. Programare condusă de evenimente	140
2.7.5. Programarea vizuală	141
2.8. Interfața aplicației	147
2.8.1. Proprietățile, metodele și evenimentele obiectelor vizuale	149
2.8.1.1. Proprietățile generale ale obiectelor vizuale	149
2.8.1.2. Evenimente recunoscute de obiectelor vizuale	151
2.8.1.3. Focalizarea	152
2.8.2. Formularele	153
2.8.2.1. Proprietățile generale ale obiectelor vizuale	153
2.8.2.2. Metode și evenimente specifice formularului	155
2.8.2.4. Mediul de date al formularelor	156
2.8.2.4. Metode de construire a formularelor	157
2.8.2.5. Obiecte vizuale folosite pentru construirea formularelor	161
2.8.3. Controalele	175
2.8.3.1. Proprietățile și evenimentele specifice controalelor	175
2.8.3.2. Controale pentru executarea acțiunilor	176
2.8.3.3. Controale pentru introducerea și afișarea datelor	178
2.8.3.4. Controale care permit utilizatorului să aleagă	206
2.8.4. Sistemul de meniuri	224
2.8.5. Aplicația	232
2.8.5.1. Programul monitor	232
2.8.5.2. Gestionarea componentelor unei aplicații	234

2. Sisteme de gestiune a bazelor de date

Competențe

Generale

Specifice bazelor de date

Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

Analizarea unei probleme în scopul identificării și clasificării datelor necesare

Identificarea relațiilor dintre date

Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă

Utilizarea funcțiilor specifice de prelucrare a datelor structurate

Analizarea problemei de gestiune și identificarea datelor necesare pentru a obține informațiile cerute de aplicație.

Gruparea datelor în tabele, identificarea relațiilor care există între tabele și, pe baza acestor relații, gruparea datelor aplicației într-o structură de date de tip bază de date.

Stabilirea corectă a structurii tabelelor și a relațiilor dintre ele astfel încât să se elimine redundanța datelor și să se asigure integritatea datelor.

Folosirea comenzilor pentru actualizarea și consultarea tabelelor libere și a bazelor de date. Folosirea limbajului SQL pentru a obține informații din baza de date.

Elaborarea algoritmilor de rezolvare a problemelor

Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora

Elaborarea strategiei de rezolvare a unei probleme

Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

Folosirea programării conduse de evenimente, a programării vizuale, a interogărilor, a vizualizărilor și a rapoartelor pentru a obține din baza de date informațiile cerute de problemă.

Structurarea aplicației pornind de la informațiile pe care trebuie să le furnizeze baza de date. Identificarea operațiilor de prelucrare a datelor din baza de date pentru obținerea acestor informații. Organizarea elementelor interfeței astfel încât exploatarea ei să însemne o succesiune logică de operații de prelucrare a datelor și de furnizare a informațiilor.

Analizarea rezolvării unei probleme folosind o aplicație cu linie de comandă (realizată prin metoda programării structurate sau prin metoda programării pe obiecte) și o aplicație realizată cu constructorii vizuali. Stabilirea metodei celei mai eficiente.

Implementarea algoritmilor într-un limbaj de programare

Utilizarea instrumentelor de dezvoltare a unei aplicații

Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific

Prezentarea unei aplicații

Utilizarea instrumentelor pentru dezvoltarea unei aplicații vizuale folosind **Visual FoxPro**.

Construirea unei aplicații pentru gestiunea bazelor de date folosind **Visual FoxPro**.

Prezentarea unei aplicații de gestiune pornind de la informațiile pe care trebuie să le furnizeze baza de date.

2.1. Mediul de programare Microsoft Visual FoxPro

Istoricul sistemelor de gestiune a bazelor de date

- **dBase II, III, III+, IV** au fost realizate de firma *Ashton-Tate*. Ele au deținut supremația mulți ani și au impus un standard în domeniul bazelor de date relaționale, astfel încât sistemele dezvoltate ulterior, cum este și **FoxPro**, au preluat și dezvoltat acest standard. Deoarece bazele de date dBase au înmagazinat un volum foarte mare de date, pentru a nu se pierde aceste date și pentru a le putea folosi ulterior cu ajutorul aplicațiilor create cu alte sisteme de gestiune a bazelor de date, toate noile sisteme de gestiune a bazelor de date pentru microcalculatoare, dacă nu folosesc standardul dBase, au implementat facilitatea de import al bazelor de date dBase.
- **Paradox** creat de firma *Borland* a fost ulterior cumpărat de firma *Corel* și integrat în setul software creat de această firmă. Nu respectă standardul dBase.
- **FoxPro** a fost creat de firma *FoxSoftware* și a fost cumpărat ulterior de firma *Microsoft*. Este sistemul de gestiune a bazelor de date care s-a impus cel mai mult în ultima vreme pe microcalculatoare.
- **Access** a fost creat de firma *Microsoft* și integrat în setul **Microsoft Office**. Nu respectă standardul dBase, dar poate importa baze de date dBase.

Baza de date relațională este un instrument complex de organizare a datelor. Ea este formată nu numai din datele care reprezintă informația propriu-zisă (datele din tabele), dar și de multe date ajutătoare necesare pentru descrierea structurii tabelor și a relațiilor dintre tabele, pentru regăsirea și interpretarea datelor, astfel încât să vă poată oferi acces la o mare cantitate de informații care să vă ajute să luați decizii. Pentru administrarea ei este necesar un software specializat numit **sistem de gestiune a bazelor de date**.

Termenul de bază de date se referă la modelul de organizare a datelor pe suportul de memorare, iar termenul de gestiune se referă la acțiunea de stocare și prelucrare a acestor date. Deja ați putut vedea ce tipuri de date se folosesc într-un sistem de gestiune a bazelor de date. Acesta a fost primul pas pentru a înțelege complexitatea unui sistem de gestiune a bazelor de date.

Orice sistem de gestiune a bazelor de date trebuie să asigure **următoarele funcții elementare**:

- definirea bazei de date,
- actualizarea datelor din baza de date prin:
 - adăugarea de noi înregistrări,
 - ștergerea unor înregistrări,
 - modificarea valorii unor câmpuri din înregistrări;
- interogarea bazei de date (extragerea informațiilor din baza de date).

La aceste funcții elementare, pe măsura dezvoltării sistemelor de gestiune a bazelor de date s-au mai adăugat și alte funcții, dintre care cele mai importante sunt:

- generarea formularelor de ecran pentru introducerea datelor,
- generarea rapoartelor,
- noi modalități de interogare a bazelor de date (de exemplu, folosind un limbaj neprocedural cum este SQL).

Visual FoxPro este cea mai nouă versiune a sistemului de gestiune a bazelor de date **FoxPro**. Termenul **visual** înseamnă că aplicația dispune de instrumentele folosite de **programarea vizuală**. Sistemul de gestiune a bazelor de date **FoxPro** este cel mai răspândit produs de acest tip pentru microcalculatoare, deoarece:

- Este **orientat pe standardul dBase**, ceea ce a permis preluarea bazelor de date exploatare prin acest sistem. Sunt baze de date care au fost create acum câțiva ani folosind sistemul de gestiune a bazelor de date dBase. S-a consumat mult timp pentru încărcarea lor și preluarea datelor elementare într-un nou format de bază de date relațională, cum este FoxPro, se poate face simplu, fără pierderea datelor.
- Asigură **compatibilitatea cu versiunile anterioare** și cu produsele dBase.
- Poate fi folosit **sub diverse sisteme de operare: Windows, Unix, System**. Deosebirea de la un sistem de operare la altul apare doar în interfață și structura meniului. Aceste deosebiri sunt ușor de depășit, deoarece înseamnă de fapt readaptarea la interfața noului sistem de operare.
- Permite o adaptare ușoară la **lucrul cu mai mulți utilizatori**, fără să necesite implementarea unei variante speciale pentru lucrul în rețea.
- Are implementat conceptul de **colecție de date definită ca un depozit cu informații despre tabele**, care permite folosirea numelor lungi pentru tabele și pentru câmpuri și implementarea **integrității referențiale** direct în tabelele bazei de date, în loc să fie descrisă în programele de aplicație.
- Permite **comunicarea și cu alte aplicații** folosind mecanisme **DDE (Dynamic Data Exchange)**, un canal de comunicație între procese prin care **Visual FoxPro** poate face schimb activ de date cu alte aplicații și poate să controleze alte aplicații – cum este, de exemplu, comunicarea cu aplicația **Excel**. Orice modificare făcută într-o foaie de calcul *Excel* se va reflecta și în tabelul bazei de date *FoxPro*.
- **Are implementat protocolul OLE (Object Linking and Embedding** – legarea și încapsularea obiectelor) care permite schimbul de date între aplicații diferite. Aplicațiile **FoxPro** sunt **aplicații client** și în ele pot fi legate și încapsulate diferite obiecte produse de aplicațiile server: texte, sunete, imagini, foi de calcul.
- Permite **importul și exportul diferitelor tipuri de date** păstrate în diferite formate de fișiere: **Access, dBase, Paradox, Excel** etc.
- Dispune de un **limbaj procedural** puternic ce permite descrierea datelor și a aplicațiilor pe baza modelului **programării structurate**, în care orice algoritm de rezolvare a unei probleme poate fi descris cu ajutorul a trei structuri de control: liniară, alternativă, repetitivă.
- Are implementat pentru interogarea bazei de date, pe lângă limbajul propriu, atât **limbajul de interogare SQL**, cât și **limbajul QBE** (interogarea prin exemple, exemplele fiind create interactiv prin diferite obiecte puse la dispoziție de interfață). Aceste limbaje ușurează folosirea sistemului de către utilizatorii neinformaticieni.
- Începând cu versiunea 3.0, în limbajul **FoxPro** este implementat și modelul **programării orientate pe obiecte**. Dacă în programarea clasică datele și programele care le prelucrau erau separate, în programarea orientată pe obiecte programul este definit ca o colecție de **obiecte** care înglobează atât ansamblul de date, cât și algoritmii care prelucrează aceste date. Pentru descrierea algoritmilor se folosește în continuare modelul programării structurate.
- Limbajul de programare propriu sistemului de gestiune a bazelor de date FoxPro are implementat și modelul **programării conduse de evenimente**. În cadrul acestui model, programul este considerat un ansamblu de proceduri care nu se execută într-o ordine descrisă de programator, ci numai atunci când apar în sistem anumite evenimente. Un exemplu de astfel de eveniment este apăsarea unei combinații de taste, care se poate produce în orice moment în sistem și nu la momente determinate de programator.
- Permite **programarea vizuală** a aplicațiilor. Acest tip de proiectare oferă programatorilor posibilitatea de a realiza mai simplu interfața aplicației cu ajutorul obiectelor vizuale. Ei nu mai sunt obligați să precizeze *cum trebuie realizată aplicația* (să descrie algoritmul de

rezolvare a problemei), ci *ce trebuie să realizeze aplicația*. Pentru aceasta, FoxPro pune la dispoziția programatorilor diferite instrumente vizuale numite **generatoare** sau **construc-tori**. Acestea sunt programe care permit proiectarea interactivă a bazelor de date, a interogărilor, a rapoartelor, a elementelor de interfață (formulare de introducere a datelor, meniuri, ferestre), a aplicațiilor și a proiectelor. În plus, utilizatorul poate apela și la **procedurile asistent (wizard)** care îl îndrumă interactiv, pas cu pas, în construirea acestor elemente (pe baza datelor furnizate de către programator în cadrul dialogului interactiv).

→ Are o viteză de lucru foarte mare, deoarece folosește o tehnologie specială denumită **tehnologie Rushmore** care reduce timpul de acces la date prin optimizarea interogărilor.

Sistemul de gestiune a bazelor de date **Visual FoxPro** permite două **moduri de lucru**:

- prin intermediul interfeței grafice,
- prin intermediul limbajului de comandă.

2.1.1. Moduri de lucru în Visual FoxPro

Sistemul de gestiune a bazelor de date **Visual FoxPro** permite două **moduri de lucru**:

- modul de lucru interactiv,
- modul de lucru programat

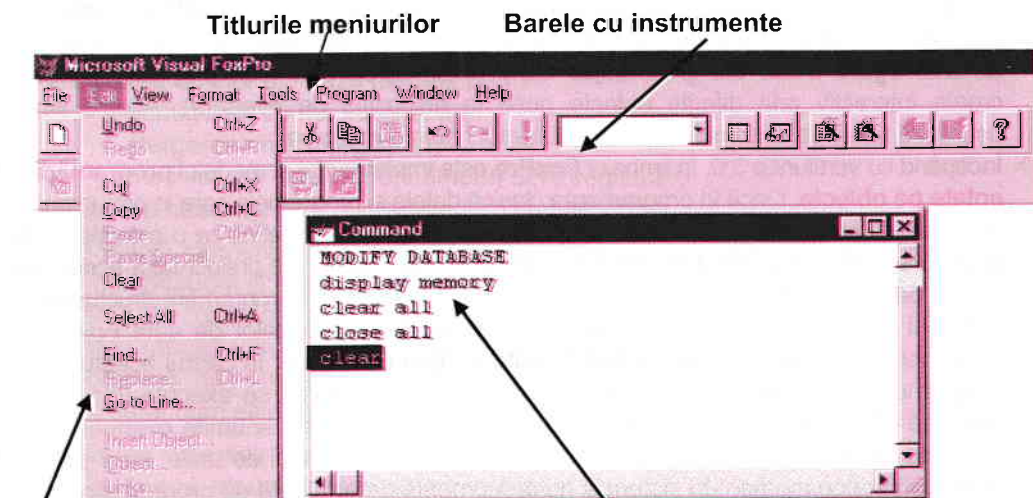
2.1.1.1. Modul de lucru interactiv

Modul de lucru interactiv se poate desfășura și el în două moduri:

- prin intermediul interfeței grafice,
- prin intermediul limbajului de comandă.

Interfața grafică a aplicației Visual FoxPro

Interfața aplicației Visual FoxPro este o interfață grafică de tip MDI (interfață cu mai multe documente), adică în fereastra aplicației pot fi deschise mai multe ferestre document. Ea respectă standardul interfeței de tip **Microsoft Windows** (meniurile, barele cu instrumente, scurtăturile de la tastatură).



Meniul **Edit** conține operațiile pentru editarea textelor, inclusiv a comenzilor din fereastra document **Command**.

În fereastra **Command** pot fi editate comenzile. Pentru editare se pot folosi aceleași tehnici ca și în cazul editoarelor de texte.

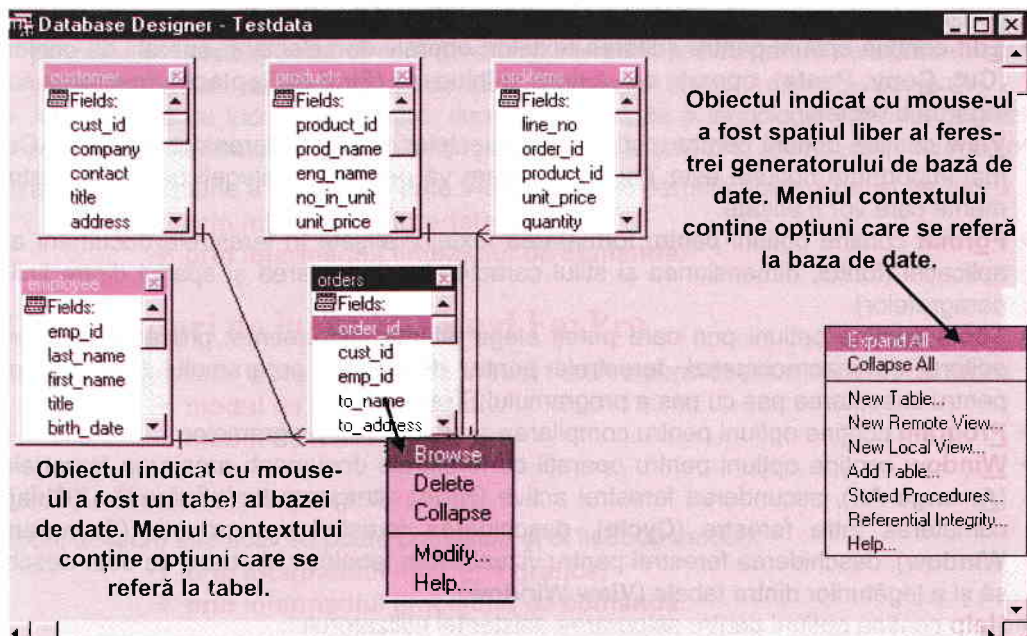
Fereastra aplicației prezintă un sistem de meniuri care pun la dispoziția utilizatorului marea majoritate a comenzilor care formează limbajul de comandă. Comenzile, în funcție de modul în care pot acționa, sunt grupate în meniuri, astfel:

- **File** conține opțiuni pentru operații cu fișiere: creare, deschidere, salvare, tipărire. Mai conține și opțiunea pentru închiderea aplicației **Visual FoxPro: Exit**.
- **Edit** conține opțiuni pentru editarea textelor: operații de selectare, operații de copiere (**Cut**, **Copy**, **Paste**), operații de căutare și înlocuire (**Find** și **Replace**), inserarea sau legarea obiectelor.
- **View** conține opțiuni pentru modificarea obiectelor afișate în fereastra aplicației. Cea mai importantă opțiune este **Toolbars...** care vă permite să alegeți barele cu instrumente care vor fi afișate.
- **Format** conține opțiuni pentru formatarea textelor afișate în ferestrele document ale aplicației (fontul, dimensiunea și stilul caracterelor, indentarea și spațiul dintre liniile paragrafelor).
- **Tools** conține opțiuni prin care puteți alege diferite instrumente: proceduri asistent, editorul de macrocomenzi, ferestrele pentru depanarea programului (**Debug**) sau pentru executarea pas cu pas a programului (**Trace**).
- **Program** conține opțiuni pentru compilarea și executarea programelor.
- **Window** conține opțiuni pentru operații cu ferestrele document: aranjarea ferestrelor (**Arrange All**), ascunderea ferestrei active (**Hide**), ștergerea ferestrei active (**Clear**), comutarea între ferestre (**Cycle**), deschiderea ferestrei de comenzi (**Command Window**), deschiderea ferestrei pentru vizualizarea tabelelor din baza de date deschisă și a legăturilor dintre tabele (**View Window**).
- **Help** conține opțiuni pentru asigurarea asistenței utilizatorului.

Sistemul de meniuri este **dependent de context**. În funcție de operațiile care se execută, mai pot să apară meniurile:

- **Table** conține opțiuni pentru manipularea înregistrărilor din tabel: adăugarea de noi înregistrări – **Append New Record** și **Append Records...**, marcarea pentru ștergere a înregistrărilor (ștergerea logică) – **Delete**, refacerea înregistrărilor șterse logic – **Recall**, ștergerea fizică a înregistrărilor marcate pentru ștergere – **Remove Deleted Records**, poziționarea pe o înregistrare din tabel – **Go to Record...**, înlocuirea valorilor unor câmpuri cu alte valori – **Replace Field...**, actualizarea indexului tabelului pentru a reflecta situația actuală – **Rebuild Indexes** etc. Meniul mai conține și opțiuni pentru stabilirea proprietăților tabelului (**Properties...**) și pentru alegerea fontului, a dimensiunii și a stilului caracterelor cu care se afișează datele din tabel (**Font...**) etc.
- **Database** conține opțiuni pentru manipularea tabelelor din baza de date: crearea unui tabel nou și adăugarea lui la baza de date – **New Table**, adăugarea unui tabel la baza de date – **Add Table**, crearea unei noi vizualizări și adăugarea ei la baza de date – **New Local View**, modificarea structurii tabelului selectat din baza de date – **Modify**, afișarea conținutului tabelului selectat din baza de date într-o fereastră de editare – **Browse**, înlăturarea unui tabel din baze de date – **Remove**, actualizarea indexului tabelului selectat din baza de date – **Rebuild Table Indexes**, ștergerea fizică a înregistrărilor marcate logic pentru ștergere din tabelul selectat din baza de date – **Remove Deleted Records...**, editarea relației selectate dintre două tabele ale bazei de date – **Edit Relationship...**, deschiderea ferestrei generatorului pentru integritatea referențială a bazei de date – **Referential Integrity...** etc.

La fel ca la orice aplicație Microsoft, utilizatorul mai are la dispoziție și **meniul contextului** sau **meniul de comenzi rapide** care se deschide atunci când se execută clic cu butonul din dreapta al mouse-ului pe un obiect din fereastră. Opțiunile din acest meniu depind de context (de obiectul indicat cu mouse-ul).



Bara de stare furnizează informații despre obiectul focalizat (titlu de meniu, opțiune de meniu, buton din bara cu instrumente etc.), despre programul care se execută (numele programului) sau despre tabelul bazei de date selectat. În acest ultim caz, furnizează informații despre identificatorul tabelului (<nume_bază_date> ! <nume tabel>) și aliasul tabelului, despre numărul înregistrării curente x din numărul total de înregistrări y. Afișarea se face sub forma <x>/<y>. Aceste informații sunt afișate în partea stângă a barei de stare:



În partea dreaptă a barei de stare sunt afișate informații despre tastele comutator **Insert (OVR)**, **Num Lock (NUM)** și **Caps Lock (CAPS)**. Dacă numele lor este afișat (ca în exemplu), înseamnă că sunt activate.

Ferestrele document care pot fi deschise în fereastra aplicației **Visual FoxPro** sunt:

- Fereastra **Command**. Se folosește pentru editarea comenzilor.
- Ferestrele pentru **constructori** și **generatoare**. Se folosesc pentru crearea și modificarea diferitelor elemente Visual FoxPro: baze de date (**Database Designer**), tabele ale bazei de date și indecși (**Table Designer**), vizualizări (**View Designer**), interogări (**Query Designer**), formulare pentru introducerea datelor (**Form Designer**), rapoarte (**Report Designer**), meniuri (**Menu Designer**).
- Fereastra pentru **administratorul de proiecte (Project Manager)**. Proiectul este o colecție de fișiere, date, documente și obiecte Visual FoxPro folosite de o aplicație.

- Fereastra pentru **editarea** fișierelor cu texte, a fișierelor de programe și a câmpurilor memo. Numele ferestrei va fi numele fișierului, iar în cazul câmpului memo numele va fi format din numele tabelului și numele câmpului memo, separate prin punct.
- Fereastra pentru editarea unui tabel. Poate să fie de tip **Browse**, când sunt afișate mai multe înregistrări din tabel, câte o înregistrare pe fiecare rând, sau de tip **Edit**, când este afișată o singură înregistrare, câte un câmp pe fiecare rând. Numele ferestrei este numele tabelului.

Deschiderea unei ferestre document se face automat, la alegerea unei opțiuni de meniu sau printr-o comandă. Operațiile cu ferestrele document (redimensionarea, minimizarea, maximizarea, mutarea, defilarea conținutului) se execută la fel ca la orice aplicație Microsoft Windows. Pentru operația de **ascundere a unei ferestre** alegeți opțiunea **Window→ Hide**.

Bare cu instrumente grupează după funcții butoanele scurtături pentru opțiunile din meniuri. Bara **standard** conține aceleași butoane ca orice bară standard din aplicațiile Microsoft Office. Celelalte bare grupează în general butoane care corespund ferestrelor document ale generatoarelor (**Database Designer**, **Form Designer**, **Report Designer**, **View Designer** și **Query Designer**) sau butoane prin care pot fi inserate diferite controale în formulare sau rapoarte (**Form Controls** și **Report Controls**). Controalele care pot fi inserate într-un formular pot fi: etichete, casete de text, deruloare, butoane de comandă, comutatoare, butoane de opțiuni, liste, casete combinate, grile etc.

Limbajul de comandă

Comenzile vor fi editate în fereastra **Command**. Fiecare comandă se scrie pe un rând. Terminarea unei comenzi se marchează prin acționarea tastei **Enter**. Acționarea acestei taste semnifică faptul că interpretorul de comenzi poate prelua această comandă să o analizeze, să o decodifice și să o lanseze în execuție.

Comanda este o succesiune de caractere organizate conform sintaxei limbajului, prin care i se cere calculatorului să execute o acțiune. Sintaxa se referă la structura liniei de comandă. Linia de comandă are următoarea **sintaxă**:



Observații:

1. Cuvintele comenzii (verbe și adverbe) se separă prin spațiu.
2. În general, adverbele din listă pot fi scrise în orice ordine.
3. Cuvintele din comandă pot fi scrise cu litere mari sau mici sau combinație de astfel de litere.
4. Verbul comenzii poate fi prescurtat la primele patru caractere ale cuvântului.
5. Comanda trebuie scrisă pe un singur rând. Dacă se dorește scrierea ei pe mai multe rânduri, înainte de acționarea tastei **Enter** (care semnifică terminarea comenzii), trebuie să se scrie caracterul **;**. Prezența lui înseamnă că prin acționarea tastei **Enter** comanda nu se termină, ci se continuă pe rândul următor.
6. Lungimea maximă a unei linii de comandă este de 1024 caractere.
7. În cadrul sistemului de meniuri, în general, opțiunea de meniu reprezintă verbul comenzii, iar adverbele sunt precizate prin intermediul controalelor din caseta de dialog.
8. Adverbele pot fi:

- nume de câmpuri, nume de variabile de memorie, nume de fișiere, aliasuri, expresii.
- o **listă** – reprezintă un grup de elemente asemănătoare (nume de câmpuri, nume de variabile de memorie etc.). Separarea lor în listă se face prin virgulă.
- un **domeniu** (scope) – reprezintă un grup de înregistrări dintr-un tabel.

Domeniul poate fi precizat prin:

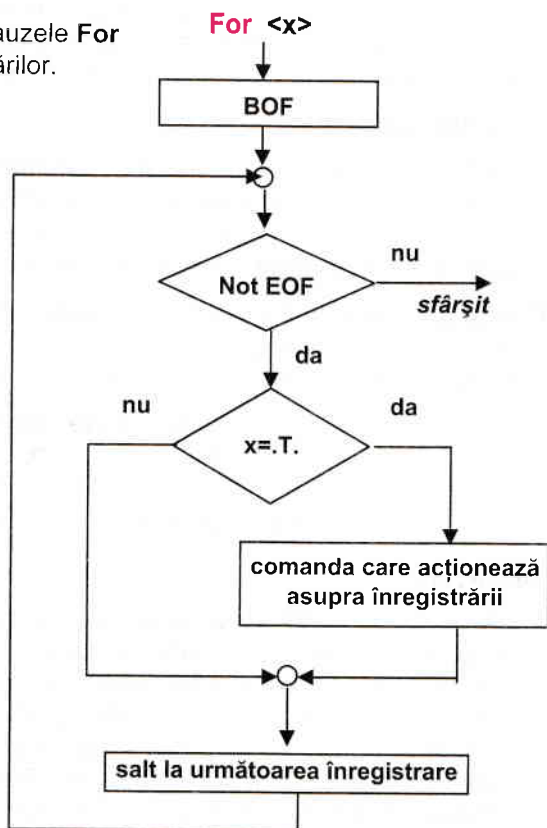
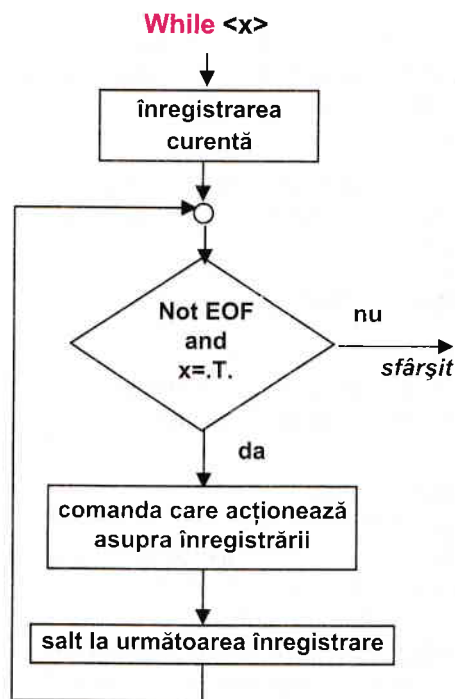
a. cuvinte cheie:

- **Record <n>** – înregistrarea cu numărul n ,
- **Next <n>** – următoarele n înregistrări, începând cu înregistrarea curentă,
- **All** – toate înregistrările,
- **Rest** – toate înregistrările până la sfârșitul tabelului, începând cu înregistrarea curentă.

b. clauze (precizează un anumit tip de acțiune):

- **For <x>** – numai înregistrările care satisfac condiția precizată prin expresia logică x , începând cu prima înregistrare a tabelului;
- **While <x>** – toată secvența de înregistrări care urmează înregistrării curente pentru care expresia logică x este adevărată, până la prima înregistrare pentru care expresia nu este adevărată.

Schemele logice descriu modul în care clauzele **For** și **While** definesc domeniul înregistrărilor.



→ **Alte clauze** care pot fi folosite ca adverbe sunt:

All/Like/Except <șablon> – poate fi folosită pentru a preciza un grup de variabile de memorie, tablouri de memorie sau fișiere, al căror nume corespunde șablonului. Comanda va acționa asupra tuturor (**All**), numai asupra celor care