

# **INFORMATIKA III. (FoxPro)**

**Ján Skalka**

## Databázové systémy

Už vieme, že počítač nie je stroj na počítanie (alebo len na počítanie). Povedali sme si, že ide o stroj, ktorý slúži najmä na spracúvanie informácií. Pod slovom spracúvanie sa skrýva evidovanie (zaznamenávanie, uchovávanie), selektovanie (výber vhodných údajov) a výstup (najčastejšie na tlačiareň) údajov.

V počítačovej oblasti je na takúto prácu určená zvláštna skupina programov, ktorým sa hovorí databázové systémy. Slúžia práve na uchovávanie a spracúvanie informácií. Informácie sú uchovávané v zoznamoch (možno si ho v jednoduchom prípade predstaviť ako jeden súbor) - databázach.

Jednoduchým príkladom databázy je napr. telefónny zoznam. Obsahuje údaje o mene, adrese a telefónnom čísle. V prípade, že hľadáme telefónne číslo konkrétnej osoby, nalistujeme jej meno v abecednom zozname a jednoducho zistíme telefónne číslo.

Horšie je, keď poznáme len adresu niektorého zo svojich obchodných partnerov, ale zabudli sme jeho meno (prípadne sme si ho zapísali absolútne nečitateľne). Kým by sme v zozname našli príslušnú adresu, potrebovali by sme pozorne(!) prejsť z neho značnú časť (ak by sme skôr nezošediveli). A práve tu sa ukáže sila databázového systému a rýchlosť spracovania informácií (nájdenie všetkých osôb evidovaných na uvedenej adrese).

Zvolíme si hľadanie podľa adresy, zadáme adresu a skôr než sa stačíme nadýchnuť, výsledok máme pred sebou. V horšom prípade dáme len usporiadať údaje nie podľa mena (ako v klasickom papierovom zozname), ale podľa adresy a hľadanie je opäť veľmi jednoduché.

Ďalším príkladom môže byť často spomínaná knižnica. V bežnej (starým spôsobom mysliacej) knižnici sú tri katalógy - podľa autorov, podľa názvu a systémový podľa oblastí. Zbytočne veľa regálov a zbytočne veľa papiera. Každá kniha je totiž evidovaná trikrát - v každom katalógu raz. A ak sa náhodou stane, že nedisciplinovaný čitateľ poprehadzuje kartotečné lístky, pravdepodobnosť nájdenia hľadanej publikácie poriadne klesne. Ak má knižnica k dispozícii počítač, operátorka (postačí osoba, ktorá rozozná písmená na klávesnici) naplní databázu údajmi (o každej knihe len raz) a to je všetko. Počítač (resp. databázový systém) údaje usporiada podľa želania, vyhľadá požadované knihy, prípadne knihy obsahujúce zvolené kľúčové slová a konečný výstup môže i vytlačiť. Ak sa knižnica nebráni využitiu počítača úplne, eviduje v ňom aj čitateľov, výpožičky a upomienky. V knižnici, ktorú má na starosti počítač, nemusí knihovníčka každé ráno prezerat' zoznam výpožičiek a hľadať darebákov, ktorým treba poslať upomienku. Ak má používaný systém zabudovanú čo len trošku inteligencie, po spustení programu automaticky vyhľadá čitateľov, ktorí k aktuálnemu dátumu mali knihy vrátiť, vytlačí im upomienky a jedinou starosťou knihovníčky je zaniest' ich na poštu. (Je pravdepodobné, že vo veľmi blízkej budúcnosti - technická realizácia je možná už v súčasnosti, no krachuje kvôli neschopnosti ľudí a strachu z používania čohokoľvek nového - bude už upomienky posielat' po sieti samotný počítač.)

V obchodných domoch a veľkoskladoch už prišli na to, že bez počítača sa možno zaobísť len veľmi ťažko. V oblastiach, kde je nutné rýchlo a pružne reagovať na požiadavky zákazníka, človek nestíha. Nie je možné, aby si v hlave udržal informácie o cenách, kvalite a množstve tovaru. Vďaka počítačom netreba - má k dispozícii databázu (ktorá potrebné údaje obsahuje) a databázový systém (ktorý s nimi dokáže pracovať).

Práca s databázovým systémom nie je zložitá. Bežné operácie možno zhrnúť do niekoľkých oblastí:

- vkladanie údajov a prezeranie databázy,
- úpravy (napr. mazanie),
- triedenie - usporiadanie podľa abecedy, podľa číselných hodnôt ap.,

- výber, prehľadávanie (napr. údajov spĺňajúcich zadanú podmienku - zvolená adresa...),
- matematické a štatistické operácie (súčty, priemery, ...),
- výstupy (napr. na tlačiareň).

V praxi nezáleží na tom, s akým systémom pracujeme, ide len o to, aby sme mali prístup k informáciám. Najdôležitejšie sú pre nás údaje, nie spôsob (program - databázový systém), akým sa k nim dostaneme.

Existuje mnoho databázových systémov, ktoré sú schopné vykonávať všetky spomenuté operácie. Medzi najznámejšie a najpoužívanejšie v našich podmienkach patria dBase, FoxPro, Paradox a v súčasnosti sa presadzuje Access pracujúci pod Windows.

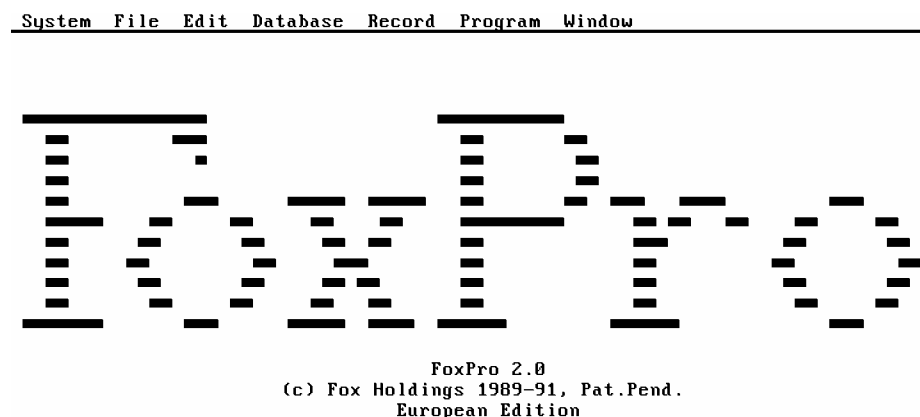
V bežnej praxi sa však nestretnete s databázovým systémom v jeho pôvodnej - surovej podobe. V každom zo spomínaných systémov totiž možno vytvárať aj aplikácie - programy, ktoré sú už prispôbené konkrétnym oblastiam použitia. Uľahčujú a zrýchľujú prácu, dávajú k dispozícii len operácie, ktoré tá-ktorá oblasť využitia potrebuje.

Na Slovensku (v ekonomike a bežnej praxi) je najčastejšie používaným systémom práve FoxPro.

Naučiť sa pracovať s jedným systémom znamená zvládnuť temer všetky, pretože odlišnosti práce s nimi sú naozaj len minimálne.

## Databázový systém FoxPro

Po spustení FoxPro sa dostaneme do základného prostredia systému. Hlavnou a prvou úlohou, ktorú pre prácu so systémom potrebujeme, je vytvoriť zoznam - databázu, do ktorej sa budú vkladať údaje.



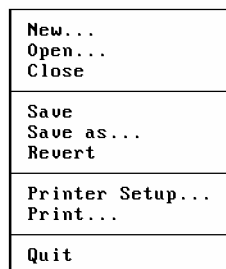
*Vytvorte zoznam svojich známych. Mal by obsahovať ich meno, adresu, vek a zamestnanie.*

Databázu (zoznam) si možno predstaviť ako tabuľku, v ktorej budú uložené údaje (vaši známi). Tabuľka bude rozdelená na stĺpce a riadky. Každý stĺpec bude zodpovedať jednému údaju a každý riadok bude obsahovať údaje o jednej osobe:

MENO	ADRESA	VEK	ZAMESTNANIE
Jožo	Dolná 54	33	robotník
Fero	Mokrovce 11	78	dôchodca
Beáta	Staré Novce 155	12	žiak
Dežo	Mrkvičkovo 999	19	nezamestnaný

Pohybovať sa a ovládať FoxPro možno okrem klávesnice aj myšou. Pri práci s klávesnicou sa po jednotlivých položkách pohybujeme klávesmi *TAB* alebo *Shift+TAB* (prípadne šípkami); ak používame myš, stačí sa na položku nastaviť a stlačiť ľavé tlačidlo. Menu sa vyvoláva stlačením *F10* alebo *Alt* (prípadne *Alt + zvýraznený znak*).

## Vytvorenie databázy



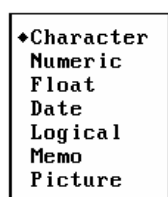
Ak chceme vytvárať novú, neexistujúcu databázu, potrebujeme najprv určiť, z čoho sa bude skladať, aké údaje bude obsahovať - potrebujeme pomenovať stĺpce, vytvoriť akúsi hlavičku - štruktúru zoznamu. Najprv zvolíme *File* → *New*. Ponúknu sa možnosti:

Vyberte *Database* (ideme vytvárať zoznam - databázu) a odsúhlaste *OK*. Po tejto akcii sa spustí dialóg, v ktorom treba určiť jej štruktúru.

Štruktúra databázy hovorí nielen o tom, ako sa budú jednotlivé stĺpce volať, ale aj o type údajov a ich šírke - koľko znakov sa má do nich vojsť.

*Name* - meno (názov) stĺpca (v našom prípade môžeme prvý stĺpec nazvať *MENO*). Napíšeme meno a tabelátorom (alebo myšou) preskočíme do ďalšieho okna.

*Type* - hovorí o type údajov (text, číslo, dátum...),



ktoré bude stĺpec obsahovať. (Ak uvedieme, že v stĺpci budú čísla, počítač nedovolí pri napĺňaní údajmi písať bežné znaky - bude len bezradne pípať.)

Pre bežný text (písmená + čísla + špeciálne znaky) sa používa *Character*. Pre náš stĺpec teda vyhovuje.

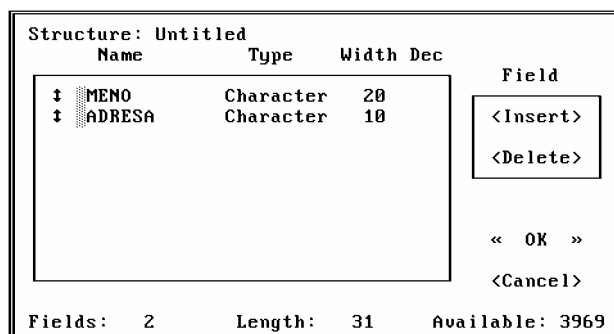
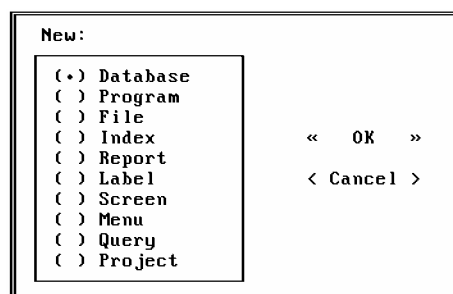
Ak chceme zvoliť iný typ, stačí stlačiť *ENTER* a vybrať si z ponuky.

*Width* - šírka stĺpca = maximálny počet znakov, ktoré môže údaj v stĺpci obsahovať. Pre naše momentálne potreby postačí 20. Na políčko sa stačí nastaviť a písať. Pokiaľ je vysvietené namodro, pôvodný text sa automaticky zmaže, inak sa bude posúvať. Maximálna povolená šírka je 254 znakov.

Po *MENO* rovnakým spôsobom zadáme *ADRESU*. Typ ani šírka sa nemenia.

Položka *VEK*, ktorej údaje vložíme do tretieho riadku, už nebude znakového typu. Bude naplnená číslami. FoxPro ponúka dva číselné typy. (Nastavte sa na *Type* a stlačte *ENTER*). Prvým, častejšie používaným, je *Numeric*, druhým *Float* určený pre nepresné čísla. Nastavíme *Numeric*. Pre šírku poľa postačia 3 miesta, t.j. 0 - 999 rokov.

Po zmene typu na číselný (*Numeric*, *Float*) sa aktivuje aj štvrtý stĺpec - *Dec* - počet desatinných miest. V tomto prípade ponecháme 0.



Pri vytváraní štruktúry ešte nenaplníme databázu, len zadávame, z čoho sa bude skladať.

Údaje o poslednom stĺpci môžu byť rovnaké ako o prvých dvoch.

Názov stĺpca nemusíme písať veľkými písmenami, menia sa automaticky.

Structure: Untitled				Field	
Name	Type	Width	Dec		
↓ MENO	Character	20			
↓ ADRESA	Character	30		<Insert>	
↓ VEK	Numeric	3	0	<Delete>	
↓ ZAMESTNANI	Character	15			

« OK »

<Cancel>

Fields: 4      Length: 64      Available: 3936

Typ stĺpca sme nemenili zo znakového na číselný kvôli zobrazovaniu, pretože aj do stĺpca, ktorý je typu *Character*, môžeme písať čísla. Stĺpec sa stal číselným kvôli operáciám. Zo znakových údajov nemožno urobiť súčet, priemer, ani ich logicky usporiadať (1, 2, 3, 10, 15, 24 - pre čísla by sa pre znaky usporiadalo ako 1, 10, 15, 2, 24, 3 - podľa "abecedy").

Po naplnení a skontrolovaní štruktúry môžeme údaje potvrdiť - *OK*.

Reakciou bude žiadosť o zadanie mena databázy:

Name the new database:	
<input checked="" type="checkbox"/> [ . ] ARCH.DBF FOXHELP.DBF FOXUSER.DBF G.DBF KNIHY.DBF	Drive <input type="text" value="C"/>  Directory <input type="text" value="FOXPRO20"/>
<input type="checkbox"/> All Files	« Save » < Cancel >

Do spodného riadku napíšeme meno súboru, do ktorého budeme údaje vkladat', a po odklepnutí *Save* sa na disk zapíše jeho štruktúra.

Potom dostanete otázka, či vkladat' údaje teraz alebo až neskôr.

Input data records now?	
« Yes »	< No >

V prípade, že zvolíte *No*, dostanete sa do pôvodného stavu, v prípade *Yes*, dostanete k dispozícii prázdnu tabuľku, obsahujúcu názvy stĺpcov, do ktorých môžete vkladat' údaje. Pri vložení prvej položky sa automaticky vytvorí ďalšia sada.

		ZNAMI	≡
Meno	Dezidor	Novak	▲
Adresa	Hrnovce	55	◆
Vek	32		
Zamestnani	krcmar		
<hr/>			
Meno	Mi		
Adresa			
Vek			
Zamestnani			
<hr/>			
»Meno			
»Adresa			
»Vek			
»Zamestnani			
<hr/>			
... automaticky vytvorená sada			

Keď do zoznamu vložíme všetky údaje, môžeme skončiť. Okno s databázou najjednoduchšie zrušíme stlačením *ESC* alebo rozumnejšie *Ctrl+W*.

Ak stlačíme ESC na položke, do ktorej sme práve vložili údaj, prideme oň. Program si totiž naše ESC vysvetlí ako snahu o zrušenie zmien v aktuálnej položke. (Podobne to bolo v T602 pri obnovení riadku - ak sme urobili zmeny, zostali na riadku a chceli sa vrátiť k pôvodnému obsahu, šlo to; ak sme sa posunuli na iný riadok, návrat bol nemožný.) Ak teda chceme končiť prácu stlačením ESC, treba sa predtým presunúť napr. o riadok späť.

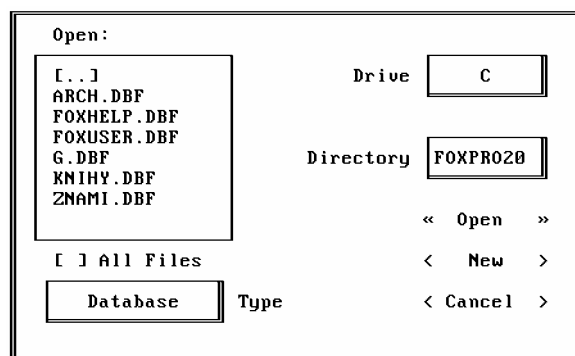
FoxPro potom opustíme voľbou *File*→*Quit*.

Vo FoxPro netreba nahrávať súbor na disk. Údaje sa ukladajú automaticky - hneď pri vkladaní.

Ne vypínajte počítač, kým neopustíte FoxPro. Mohlo by sa stať, že pridete o súbory, s ktorými ste pracovali. Ak chcete počítač vypnúť, zvolíte najprv *File*→*Quit*, čím jednoducho odložíte a zatvoríte všetky používané súbory, jednoducho opustíte FoxPro a vrátite sa do štartovacieho prostredia.

## Práca s existujúcou databázou

### Otvorenie



Spustíme opäť FoxPro a zvolením *File*→*Open* v hlavnom menu sa pokúsme otvoriť databázu s údajmi o svojich známych.

Vybrali sme meno súboru, zvolili *Open* - a nič. Údaje, ktoré sme do databázy vložili, sa nezobrazili. Sú stratené?

Otvorenie databázy ešte neznamená jej zobrazenie. Postupnosťou *File*→*Open* len otvoríme, pripravíme údaje na ďalšie spracúvanie.

Ak chceme údaje aj zobrazit', vyberieme *Record*→*Append*. Zobrazí sa naplnená databáza a môžeme pokračovať napr. pridávaním ďalších položiek.

### Režimy zobrazenia

Vo FoxPro možno s databázou pracovať v dvoch zobrazeniach - v tom, aké sme mali doposiaľ - riadkovom (hovorí sa mu *append* alebo *change*) - alebo v tabuľkovom (*browse* - stĺpcové).

Možno sa medzi nimi ľubovoľne prepínať a záleží len od užívateľa, s ktorým bude pracovať.

Do režimu *browse* sa z režimu *append* dostaneme zvolením *Browse*→*Browse*.

Nazad tým istým spôsobom, ale namiesto položky *Browse* nájdeme v menu *Append* alebo *Change*.

**REŽIM APPEND**

System File Edit Database Record Program Window Browse	
ZNAMI	
Meno Michal Drak Adresa Brnouce 44 Vek 25 Zamestnani starosta	Browse Grid Off Unlink Partitions Change Partition ^H
Meno Juraj Maly Adresa Antily 226 Vek 54 Zamestnani farmar	Size Field Move Field Resize Partitions
Meno Ivan Tretuch Adresa Horny Dolnik 12 Vek 43 Zamestnani ucitel	Goto... Seek... Toggle Delete ^T Append Record ^N

**REŽIM BROWSE**

System File Edit Database Record Program Window Browse				
ZNAMI				
Meno	Adresa	Vek	Zamestnani	Append Grid Off Unlink Partitions Change Partition ^H
Dezidor Novak	Hrnouce 55	32	krcmar	Size Field Move Field Resize Partitions
Michal Drak	Brnouce 44	25	starosta	
Juraj Maly	Antily 226	54	farmar	
Ivan Tretuch	Horny Dolnik 12	43	ucitel	
				Goto... Seek... Toggle Delete ^T Append Record ^N

**Názvoslovie**

Kvôli lepšej zrozumiteľnosti sa (aj v databázových systémoch) zaviedlo jednotné názvoslovie.

Celému súboru údajov možno hovoriť databáza alebo tabuľka (database, table).

Jeden riadok tabuľky sa nazýva záznam alebo veta (record).

Stĺpcu tabuľky patrí pomenovanie pole (field).

Meno	Dezidor Novak
Adresa	Hrnouce 55
Vek	32
Zamestnani	krcmar

Meno	Adresa	Vek	Zamestnani
Dezidor Novak	Hrnouce 55	32	krcmar
Michal Drak	Brnouce 44	25	starosta

Vytvorte databázy:

**AUTOBAZAR** s pol'ami:

ZnačkaCH	15
Cena	N 6
Vek	N 2
Najazdene	N 10

**KRAJE s poľami:**

Kr_mesto	CH	25
Okresov	N	2
Obyvateľov	N	6
Zameranie	CH	15

Pod zameranie uveďte napr. poľnohospodárstvo, strojárstvo, baníctvo, obchod...

**Zmena štruktúry databázy**

Vytvorte databázu TOVAR s poľami:

Nazov	CH	20	
Kusov	N	5	
Cena	N	5	
Dodavateľ	CH	20	a naplňte ju.

Databáza je naplnená, sú v nej všetky druhy tovaru, ale má jednu chybu: ceny sú v celých korunách...

Database: C:\FOXPRO20\TOVAR.DBF

<p>Structure: &lt;Modify&gt;</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>▶NAZOV</td><td>C</td><td>20</td><td>0</td></tr> <tr><td>KUSOV</td><td>N</td><td>5</td><td>0</td></tr> <tr><td>CENA</td><td>N</td><td>5</td><td>0</td></tr> <tr><td>DODAVATEL</td><td>C</td><td>20</td><td>0</td></tr> </table>	▶NAZOV	C	20	0	KUSOV	N	5	0	CENA	N	5	0	DODAVATEL	C	20	0	<p>Indexes:</p>	<p>Index</p> <p>&lt; Add... &gt;</p> <p>&lt;Modify...&gt;</p> <p>&lt; Remove &gt;</p> <p>&lt;No Order &gt;</p>
▶NAZOV	C	20	0															
KUSOV	N	5	0															
CENA	N	5	0															
DODAVATEL	C	20	0															

Fields: 4 Length: 51 Index expr:  
Index filter:

[X] Set Fields...  
( ) On (.) Off

[ ] Filter...  
[ ] Format...

« OK »

Na to, aby sme tento nereálny stav zmenili, nepotrebujeme vytvoriť nový súbor, v ktorom by sme pre stĺpec s cenou nastavili dve desatinné miesta, ale stačí nám zmeniť štruktúru vytvorenej databázy - zmeniť šírku stĺpca a pridať doň dve desatinné miesta. Použijeme *Database*→*Setup*, kde pre naše potreby zatiaľ postačí voľba *Modify*.

Structure: C:\FOXPRO20\TOVAR.DBF

Name	Type	Width	Dec	Field
‡ NAZOV	Character	20		<Insert>
‡ KUSOV	Numeric	5	0	<Delete>
‡ CENA	Numeric	5	0	
‡ DODAVATEL	Character	20		

« OK »

<Cancel>

Fields: 4 Length: 51 Available: 3949



Dostali sme sa do rovnakého okna, v akom sme sa nachádzali pri vytváraní novej databázy. Ďalšie polia môžeme na koniec pridávať bez akýchkoľvek obmedzení. Ak chceme vložiť pole medzi už existujúce, nastavíme sa na riadok, pred ktorý ho chceme pridať, a zvolíme *Insert* (buď priamo v okne - ak máme myš - alebo cez menu *Structure*→*Insert Field*). Ak chceme pole vymazať, použijeme *Delete*.

Našou úlohou je však zmeniť cenu tak, aby zodpovedala bežnému používaniu, t.j. pridať dve desatinné miesta.

Nastavíme sa na *Dec* a jednoducho dopíšeme 2.

**Cena**            **Numeric**        **5**        **2**

Čo sa tým stalo?

FoxPro z určitých dôvodov (známych len programátorom, ktorý ju vytvorili) nereaguje na takúto zmenu bežným spôsobom, t.j. nevyhradí si pre cenu 5 miest + 2 desatinné, ale:

- na celé číslo (celá časť, bodka, desatinná časť) vyhradí 5 miest;
- z tých piatich odtrhne zadané dve na desatinnú časť;
- a jedno na desatinnú bodku.

- a jedno na desatinnú bodku.

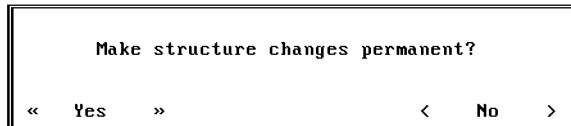
Uvedeným nastavením získame dvojčiferné číslo s dvoma desatinnými miestami. Zadaná šírka poľa hovorí nie o počte cifier celočíselnej časti, ale o celkovom počte miest, do ktorého sa musí vojsť číslo aj s desatinnými miestami.

Ak v našom prípade chceme, aby cena mala 5 celých miest a 2 desatinné, potrebujeme vyhradiť šírku 8 (5+2+des. bodka).

*V prípade, ak zvolíme No, vrátíme sa späť do Modify. Ak by sme chceli zrušiť zmeny, ktoré sme v štruktúre urobili, cesta vedie cez Cancel.*

*Po odklepnutí Cancel dostaneme otázku, či chceme zrušiť zmeny.*

*Ak na túto odpovieme Yes, vrátíme sa k pôvodnej štruktúre databázy.*



Po zmenách štruktúry odsúhlasíme *OK* a na otázku, či ich zachovať natrvalo, odpovieme *Yes*. Zmeny, ktoré sa vykonajú v databáze, nemajú deštruktívny vplyv na údaje v nej obsiahnuté. Všetky údaje zostanú zachované (okrem zvláštnych prípadov, ktoré ešte spomenieme) a v prípade, že sa zmení typ poľa (napr. z *Numeric* na *Character*), zmenia aj svoj typ.

*Pridajte k databáze ďalšie dve polia - Cena\_s\_dph a Clo.*

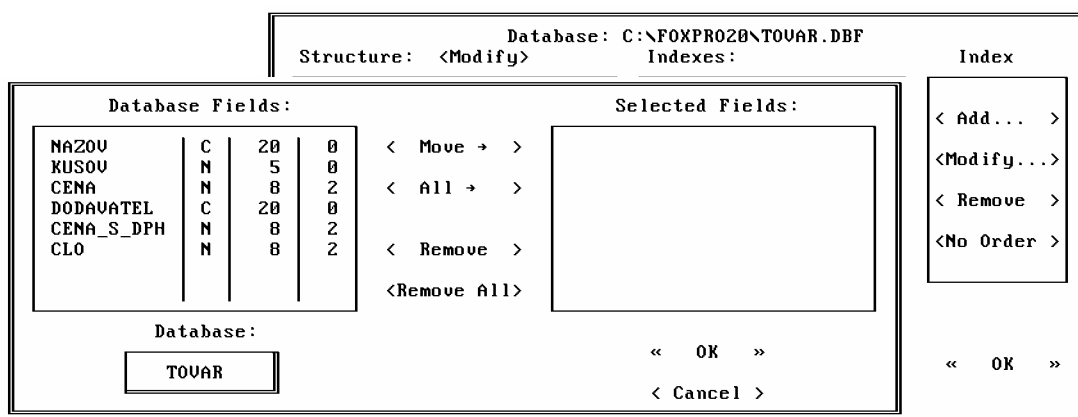
## Výber polí (projekcia)

Často sa stáva, že databáza obsahuje viac stĺpcov, ako momentálne potrebujeme. Niektoré prekážajú a zbytočne zaberajú miesto na monitore. Vtedy je najjednoduchšie skryť ich (resp. vybrať na zobrazenie len potrebné stĺpce).

Túto operáciu umožňuje každý databázový systém a ani FoxPro nie je výnimkou.

Otvorte databázu TOVAR z predchádzajúcej úlohy a nechajte zobrazit' len názov tovaru a jeho cenu.

Otvoriť databázu vieme. Nezávisle na tom, či je zobrazená alebo nie, stačí zvolit' Database→Setup a vybrať položku Set Fields (nastav polia).



Získame okno rozdelené na dve časti. V prvej je zoznam všetkých polí, druhá je prázdna. Ľavá časť obsahuje zoznam všetkých polí, pravá len tie polia, ktoré sa majú zobrazovať. Ak je pravá polovica prázdna, znamená to, že sa budú zobrazovať všetky polia (žiadne nie sú vybraté).

Ak chceme zobrazit' konkrétne polia, treba ich z ľavej časti preniesť do pravej. Existuje na to zasa niekoľko možností:

- na požadované pole dva razy klikneme myšou,
- nastavíme sa na požadované pole a *ENTERom* ho presunieme,
- nastavíme kurzor na požadované pole a klikneme *Move...*

Ak chceme preniesť všetky polia, zvolíme *All*; ak chceme zrušiť zobrazenie poľa, preniesieme ho analogickým spôsobom z pravej časti do ľavej.

Po správnom výbere a odsúhlasení sa vrátíme opäť do okna *Setup*. Vybrané polia svietia normálne, nevybrané sú pohasnuté.

Po ďalšom odsúhlasení a znovuzobrazení databázy máme skutočne k dispozícii len údaje zo zvolených polí.

*V prípade, že máte zobrazené nielen vybrané polia, zrušte okno s databázou a dajte opäť Append alebo Browse. Tento nedostatok sa pomerne často objavuje aj pri ďalších činnostiach, no dá sa naň zvyknúť.*

*Vráťte zobrazenie databázy do pôvodného stavu (t.j. zobrazte všetky polia).*

*Zobrazte len názov tovaru a počet kusov na sklade.*

*Zobrazte názov tovaru, dovozcu a clo.*

*Zobrazte databázu tak, aby nebola viditeľná cena s DPH.*

## Výber riadkov (filtrácia)

Podobne, ako nie vždy potrebujeme zobrazit' všetky stĺpce, sa stáva, že chceme z databázy vybrať len niektoré vety - riadky. Výber riadkov podľa podmienky (zadáваме ju my) sa nazýva filtrácia (údaje filtrujeme - cez sito prejdú len tie, ktoré splnia určenú podmienku).

FoxPro takmer pri všetkých funkciách umožňuje používať filter, no najjednoduchší spôsob výberu dosiahneme použitím *Database* → *Setup* a v otvorenom okne zvolíme *Filter*.

*Vyberte z databázy TOVAR len tovar, ktorého cena je väčšia ako 50,- Sk.*

Zadanú úlohu potrebujeme najprv pochopiť. O čo ide?

Chceme, aby sa zobrazili len tie riadky, ktoré majú v položke *Cena* číslo väčšie ako 50.

Podmienka teda bude *Cena > 50*. Zadať ju možno opäť niekoľkými spôsobmi, no najjednoduchší je:

1. dva razy kliknúť myšou do okna *Field names* na riadok s cenou, čím do okna *Set filter* preniesieme úplný názov poľa (*tovar.cena*),
2. dopísať za cenu zvyšok podmienky: *>50*,

3. odsúhlasiť *OK*.

*Pokiaľ si nie sme celkom istí, či našu podmienku počítač pochopí, môžeme si jej správnosť overiť voľbou *Verify*. Ak je všetko v poriadku, vypíše: **Expression is valid**.*

Po návrate do okna *Setup* vidíme text podmienky pri *Filtri*.

V databáze zostali zobrazené len riadky spĺňajúce podmienku (ak sa v nej vyskytujú iné riadky, zrušte ju a opäť zobrazte).

---

Filter zrušíme opätovným návratom a vymazaním podmienky.

*Vyberte tovar, ktorého je na sklade menej ako 20 kusov.*

*Vyberte tovar, ktorého clo je 200,- Sk.  
(podmienka: **tovar.clo = 200**)*

*Vyberte tovar, ktorého cena s DPH je 100,- Sk a viac.  
(podmienka: **tovar.cena\_s\_dph>99** alebo **tovar.cena\_s\_dph>=100**).*

V databázach možno, samozrejme, používať projekciu a filtráciu (výber stĺpcov a výber riadkov) aj súčasne. Vtedy sa údaje vyberajú podľa podmienky a zobrazia sa len nastavené stĺpce.

*Pole (stĺpec), ktoré je použité v podmienke, musí byť zobrazené.*

V podmienke nestačí uviesť len názov stĺpca, je potrebné pridať aj názov databázy a oddeliť ich bodkou: napr. **tovar.cena**. Písanie môžeme ušetriť klepnutím myši na príslušné pole v okne *Field names*.

V prípade, že podmienke nevyhovuje ani jeden riadok, zobrazí sa prázdna databáza.

*Vytvorte databázu žiakov triedy, v ktorej budete evidovať: meno, vek, priemer známok, známku za správanie, dochádzku (miestny, cezpoľný), počet vymeškaných hodín a počet súrodencov.*

*Zobrazte meno, priemer a známku za správanie.*

*Zobrazte meno, dochádzku a počet vymeškaných hodín.*

*Zobrazte meno, známku za správania a počet vymeškaných hodín.*

*Zobrazte len žiakov, ktorí majú dvoch súrodencov.*

*Zobrazte žiakov, ktorí nemali zo správanie jednotku.*

*Zobrazte vyznamenaných (priemer známok je 1,5 a menej).*

*Zobrazte cezpoľných.*

Tu sa trochu zastavíme. Pri bežných podmienkach (kým sme pracovali s poľami číselného typu) sme podmienku napísali ako názov poľa, znamienko a číslo.

Pri znakových (*Character*) poliach potrebujeme hodnotu (obsah, s ktorým pole porovnáваме) uzavrieť do oddeľovačov (v prípade znakových hodnôt sú to úvodzovky).

Podmienka teda nebude vyzeráť **dochadzka = cezpolny**, ale

***dochadzka = "cezpolny"***

Tento “nedostatok” má svoje opodstatnenie, no zaoberať sa ním nie je naším cieľom.

Zobrazte *cezpolných* žiakov tak, aby sme videli len ich mená a dochádzku. (Nastavíme filter na *dochadzka* = "*cezpolny*" a vyberieme polia *meno*, *dochadzka*).

Zobrazte žiakov s priemerom do 2,0 s tým, že nás zaujíma ich meno, priemer, známka zo správania a počet vymeškaných hodín.

Zobrazte žiakov, ktorí majú viac ako troch súrodencov, a vyberte polia: meno, priemer, počet súrodencov.

## Viacnásobné podmienky

Vyberte žiakov, ktorí sú *cezpolní* a majú *jedného* súrodenca.

Táto úloha nie je rovnakého typu, ako sú predošlé, hoci tak na prvý pohľad vyzerá.

Sú v nej skryté dve podmienky:

1. žiaci majú byť *cezpolní* (*dochadzka* = "*cezpolny*"),
2. majú mať *jedného* súrodenca (*poc\_surod* = 1).

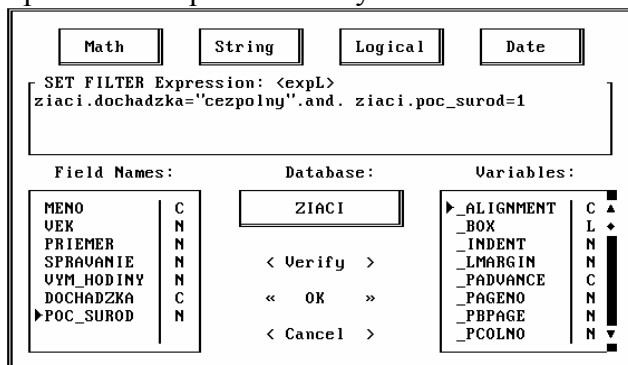
Každú z častí úlohy dokážeme vyriešiť, no ich spojenie, t.j. aby platila aj jedna, aj druhá podmienka, vyžaduje čosi viac.

Čosi viac je spojka "a súčasne" (*and*).

Treba ju vložiť medzi podmienky vo filtri.

Hovorí počítaču, aby vyberal tie riadky, v ktorých platia obe podmienky súčasne. Slovičku *and* sa hovorí logická spojka a od ostatného textu by malo byť oddelené bodkami.

Správne bude podmienka vyzerat' ako:



Vyberte žiakov, ktorí majú viac ako 2 súrodencov a sú vyznamenaní (t.j. priemer je do 1,5).

Vyberte žiakov, ktorých priemer je nad 2,5, alebo majú zo správania známku horšiu ako 1.

V úlohe nemá súčasne platiť, že priemer je horší ako 2,5 a správanie je horšie ako 1. V tomto prípade pre zaradenie žiaka do výberu stačí, aby bola splnená jedna z podmienok.

Namiesto spojky *and* (a), použijeme spojku *or* (alebo). V prípade *and* je podmienka splnená, ak sú splnené obe (pri viacnásobnej - všetky) jej časti; v prípade *or* stačí, aby bola splnená jediná.

Riešenie potom vyzerá:

```
SET FILTER Expression: <expl>  
ziaci.priemer>2.5 .or. ziaci.spravanie>1
```

*Vyberte žiakov, ktorých priemer je pod 1,2, alebo nemajú ani jedného súrodenca.*

### **Podmienky pre Character**

Zvláštnosťou sú podmienky používané pri poliach typu *Character*.

*Vyberte žiakov z prvej polovice abecedy (t.j. od A do K).*

Pri týchto podmienkach si treba uvedomiť, že rovnako ako čísla aj písmená majú v počítači svoje poradie. Ak teda povieme, že chceme vybrať žiakov od *A* do *K*, znamená to, že nás zaujímajú tí, ktorých meno je v abecede za *A* a pred *K*, t.j. ***meno>"A" .and. meno<"K"***, prípadne (ak berieme vrátane *A* a *K*) - ***meno>="A" .and. meno<="K"***.

Podobne, ak by sme mali vybrať žiakov s menom začínajúcim na *A*, použili by sme podmienku ***meno>="A" .and. meno<"B"*** atď.

*Zistite priemer žiakov z druhej časti abecedy.*

## OPAKOVANIE I

Najdôležitejším a najdrahším tovarom sú informácie. Počítače sa už od svojho vzniku zaoberali ich uchovávaním a spracúvaním. Najbežnejšími a najčastejšie používanými informáciami sú rôzne druhy evidencie (telefónny zoznam, skladové karty, katalógy v knižniciach, zoznamy pracovníkov...).

Programy, ktoré sa zaoberajú uchovávaním a spracúvaním informácií, sa nazývajú databázové systémy. Pod spracúvaním informácií treba rozumieť ich prezeranie, triedenie (výber, usporiadanie), tlač ...

Medzi najznámejšie a najpoužívanejšie databázové systémy patria v súčasnosti: dBase (s nadstavbou Clipper), FoxPro, Paradox a Access pod Windows. Okrem nich existuje, samozrejme, množstvo ďalších systémov, ktoré sú dokonalejšie, no menej univerzálne.

Databázové systémy pracujú s údajmi uloženými v súboroch (databázach, databankách), ktoré si možno predstaviť ako tabuľky. V prvom riadku tabuľky sú názvy stĺpcov, v ďalších údaje, ktoré im zodpovedajú.

Všetky systémy majú rovnaký základ, u nás sa najčastejšie používa FoxPro.

### Typy údajov

Dosiaľ sme sa stretli len s

- alfanumerickým (*character*), ktorý môže obsahovať ľubovoľné znaky a
- číselným (*numeric, float*) pre celé alebo desatinné čísla.

### Vytvorenie databázy

- *File* → *New* → (*.*) *Database*,
- definujeme jej štruktúru (názvy polí (stĺpcov), typy a rozmery),
- zadáme meno databázy.

Po týchto operáciách je hotová hlavička (štruktúra) a môžeme sa rozhodnúť, či databázu naplníme, alebo ju zatiaľ ponecháme prázdnu.

### Režimy zobrazenia

- *append* (*Record* → *Append*) - slúži na pridávanie údajov do databázy; položky vety sú zobrazené pod sebou.

- *change* (*Record* → *Change*) - vyzerá rovnako ako *append*, no umožňuje len prezeranie, nie pridávanie.

- *browse* (*Database* → *Browse*) - zobrazí databázu v prehľadnej (tabuľkovej) forme. Prechod cez menu nemusí mať vždy správny účinok (t.j. nie vždy sa pri tomto postupe možno dopracovať k tabuľkovej forme, pretože položka *Database* → *Browse* nesie so sebou prívlastok *browse last* = zobraz podľa posledného. Ak bola databáza pri poslednom otvorení v režime *append*, bude v ňom i teraz).

Medzi režimami sa možno prepínať (a to platí stopercentne) cez poslednú položku v hlavnom menu (podľa voľby buď *Append* alebo *Browse*).

### Pohyb

Pohyb je pre zobrazenia *browse* a *append* trochu odlišný pri použití klávesov *PgUp* a *PgDown*. Pri *browse* sa pomocou nich pohybujete po stránkach, pri *append* vás *PgUp* posunie o record vyššie a *PgDown* o record nižšie.

### Zmena štruktúry

---

*Database*→*Setup*→*Modify* - ponúkne rovnaké okno ako pri definovaní štruktúry a umožní pridávať, mazať, meniť poradie položiek.

### Výber stĺpcov

*Database*→*Setup*→*Set Fields* - umožní vybrať stĺpce, ktoré sa majú pri zobrazovaní databázy ukázať. Ostatné sa nevymažú, zostanú len skryté.

Zobrazené stĺpce sú vysvietené jasne, skryté šedivo.

### Výber riadkov

*Database*→*Setup*→*Filter* - umožňuje nastaviť podmienku, podľa ktorej sa budú do zobrazenia vyberať vety - riadky.

Podmienka má bežne tvar: pole, znamienko (ne-) rovnosti, hodnota.

Čísla píšeme bežným spôsobom, pre hodnoty typu *character* používame ako oddeľovače úvodzovky.

V systéme možno súčasne použiť výber stĺpcov aj výber riadkov, no v podmienke je povolené použiť len zobrazené stĺpce. (Kontrolou sa nemusíte trápiť, FoxPro vám nezobrazené stĺpce jednoducho neponúkne.)

### Viacnásobné podmienky

V prípade, že chceme, aby sa zobrazené riadky vyberali podľa viacerých podmienok, musíme ich od seba oddeliť logickými spojkami **and** alebo **or**.

V prípade **and** sa riadok zobrazí, len ak sú splnené všetky podmienky súčasne, pri **or** stačí splnenie jednej.

Logické spojky sa od ostatných znakov oddeľujú bodkami.

Okrem nich existuje aj **not**, čo znamená opak (negáciu). Napr. pre podmienku vek je rôzny od 25, možno použiť:

**vek<>25**

**vek<25 .and. vek>25**

**.not.vek=25**



## Okná

Vo FoxPro sa pracuje s oknami. Pre každú činnosť, každú akciu sa otvoria príslušné okná. Väčšinu z nich možno presúvať, usporadúvať na obrazovke, prepínať sa medzi nimi ap. Bežné okno (*browse*, *command*...) má okrem svojho obsahu aj niekoľko ovládacích prvkov slúžiacich na manipuláciu s ním.

## Command okno

Okrem menu sa po spustení FoxPro nachádza na obrazovke aj malé okno so záhlavím *Command*. Hovorí sa mu *Command okno* (príkazové okno).

```

■ Command
clear
SET FILTER TO tovar.cena
CREATE Untitled
SET FIELDS TO

```

Toto okno zaznamenáva všetky činnosti, ktoré ste od spustenia FoxPro vykonali, a okrem iného umožňuje aj ovládanie systému pomocou príkazov.

Možno ho v ľubovoľnej situácii vyvolať kombináciou *Ctrl+F2*.

Boli sme zvyknutí napr. na otvorenie databázy používať menu *File→Open*, vybrať súbor a odsúhlasiť.

V *Command okne* stačí použiť príkaz *use meno\_súboru* a súbor sa otvorí.

Rovnako pre zobrazenie namiesto *Record→Append* stačí *append* a databáza sa zobrazí.

Problémy so zobrazením do tabuľky a neustálym prepínaním sa vyrieši príkaz *browse*, ktorý za každých okolností zobrazí tabuľku.

Atd.

Okrem takéhoto uľahčenia sú cez *Command okno* dostupné aj funkcie FoxPro, ku ktorým by sme sa iným spôsobom len ťažko dostali.

Napr. *set status on* zobrazí v dolnej časti obrazovky stavový riadok, ktorý hovorí o názve databázy, počte recordov (riadkov) a aktuálnej pozícii. (K ďalším potrebným príkazom sa dostaneme postupne.)

```
|<C:>|ARCH
```

```
|Rec: 12/17
```

```
|Ins |Num
```

Výhodou *Command okna* je aj možnosť pohybovať sa po vykonaných príkazoch a ľubovoľný z nich odklepnúť (vykonať znova - netreba ho písať).

## Typ Date

Vytvorte databázu *ZAMESTNANCI*, v ktorej budete evidovať meno, dátum narodenia, počet detí, plat, dátum prijatia a stav (slobodny, zenaty, rozvedeny, vdovec).

V úlohe sa vyskytujú dve položky, ktoré majú obsahovať dátum. V databázových systémoch si nemusíme lámať hlavu nad tým, či používať desiatinné čísla alebo vkladať dátum do poľa typu *Character*. Všetky bežne používané programy totiž obsahujú typ *date*. Obsahuje deň, mesiac a rok s tým, že pri definovaní štruktúry si automaticky vyhradí 8 miest.

Structure: Untitled			
Name	Type	Width	Dec
‡ MENO	Character	10	
‡ DAT_NAR	Date	8	

Field  
 <Insert>  
 <Delete>

« OK »

<Cancel>

Fields: 2      Length: 19      Available: 3981

Pri štandardnom nastavení je dátum v poradí *mesiac/deň/rok*, t.j. v americkom formáte. Ak ho chceme zmeniť na európsky, u nás zaužívaný (*deň.mesiac.rok*), musíme do *Command okna* napísať príkaz **set date german**.

Dátumový typ je pomerne "inteligentný". V prípade nesprávneho dátumu vás jednoducho ďalej nepustí, pokiaľ ho neopravíte. Tým dokáže zabrániť napr., aby sa zamestnanec narodil 33.11. alebo aby nastúpil do práce 30.2..

*Zobrazte zamestnancov, ktorí sa narodili od roku 1968.*

Čo znamená narodiť sa od roku 1968? Že dátum narodenia je po 31.12.1967. Do filtra teda napíšeme:

**zamestn.dat\_nar>{31.12.1967}**

Podobne ako úvodzovky pre typ *character* sú zložené zátvorky oddeľovačom pre typ *date*.

*Vyberte zamestnancov, ktorí nastúpili do práce v rozpätí rokov 1989 - 1992. Zobrazte len meno a dátum nástupu do práce.*

## Typ Logical

*Pridajte k údajom o zamestnancoch údaj o tom, či majú okuliare a auto.*

Tento problém by sme mohli vyriešiť tak, že by sme do štruktúry pridali dve polia typu *character* a do nich buď napísali slovom alebo "zakódovali" do jedného písmena (A, N) potrebné údaje. V databázových systémoch je však jednou z najdôležitejších požiadaviek zabrať minimum miesta.

V prípade, ak ide o položku, ktorej obsah možno vyjadriť slovami áno alebo nie, používa sa typ *logical*, ktorý má tú výhodu, že v pamäti (databáze) zaberá len 1 bit (zatiaľčo každý znak typu *character* zaberá 1 byt = 8 bitov).

Structure: C:\FOXPRO20\ZAMESTN.DBF			
Name	Type	Width	Dec
‡ MENO	Character	20	
‡ DAT_NAR	Date	8	
‡ DETI	Numeric	2	0
‡ PLAT	Numeric	10	2
‡ DAT_PRIJAT	Date	8	
‡ STAV	Character	10	
‡ OKULIARE	Logical	1	
‡ AUTO	Logical	1	

Field  
 <Insert>  
 <Delete>

« OK »

<Cancel>

Fields: 8      Length: 61      Available: 3939

Pre typ *logical* sú prípustné len dve hodnoty: **T** (true = pravda = áno) a **F** (false = nepravda = nie). Pri definovaní štruktúry sa pre pole typu *logical* vyhradí automaticky jedno miesto.

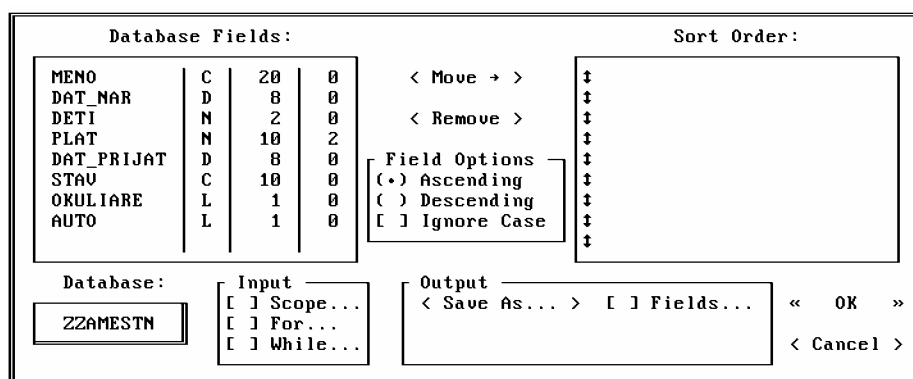
Zobrazte zamestnancov, ktorí majú auto, a vyberte len pole s ich menom a dátumom nástupu do práce (samozrejme, aj s poľom obsahujúcim údaj o aute).

Podmienku vlastníctva auta zadáme ako *zamestn.auto = .T.* Ako oddeľovač sa pre typ *logical* používajú bodky. Ak by sme chceli zamestnancov bez auta, určíme podmienku *zamestn.auto = .F.*

## Triedenie (sortovanie)

Databázy sú určené na evidovanie údajov. Požadujeme od nich, aby boli prehľadné a aby sa v nich rýchlo dala nájsť hľadaná položka. A preto je nevyhnutné mať v zozname aj určitý systém. Najlogickejším a najjednoduchším je usporiadanie databázy, napr. podľa abecedy, podľa veku....

Na triedenie (usporiadanie) sa používa funkcia *Sort* (triediť). V menu sa nachádza *Database* → *Sort*.



Pozostáva z niekoľkých okien:

*Database Fields* - obsahuje zoznam všetkých stĺpcov (polí), podľa ktorých možno triediť.

*Field options* - nastavenie vlastností triedenia:

- *Ascending* - triedenie vzostupne (od najmenšieho po najväčšie),
- *Descending* - triedenie zostupne (od najväčšieho po najmenšie),
- *Ignore case* – ignorovať rozdiel medzi veľkými a malými písmenami.

*Abeceda v počítači je totiž organizovaná trochu inak, než ako sme zvyknutí ju používať. Najprv sú znaky !, ", #, \$, .... potom čísla 0..9, nasledujú veľké písmená A..Z, malé písmená a..z, znaky s diakritikou é, á, í... a napokon špeciálne znaky (čiara, dvojité čiara) atď.*

*V praxi to znamená asi toľko, že slovo Zenit bude podľa abecedy skôr ako babička a oboje predbehne slovo (číslo) 251.*

*Ak zapneme voľbu Ignore case, vyrieši sa aspoň chaos medzi veľkými a malými písmenami, t.j. najprv bude babička a až potom*

Spôsob triedenia (vzostupne, zostupne) treba nastaviť predtým, ako presuniete pole z *Database Fields* do *Sort order*.

Database Fields:						Sort Order:	
MENO	C	20	0	< Move → >	↑	ZAMESTN.MENO	
DAT_NAR	D	8	0	< Remove >	↓	ZAMESTN.DAT_NAR	
DETI	N	2	0		↑		
PLAT	N	10	2		↑		
DAT_PRIJAT	D	8	0		↑		
STAV	C	10	0	Field Options	↑		
OKULIARE	L	1	0	( ) Ascending	↑		
AUTO	L	1	0	(*) Descending	↑		
				[ ] Ignore Case	↑		
Database: ZAMESTN				Input		Output	
				[ ] Scope...		< Save As... > [ ] Fields...	
				[ ] For...		« OK »	
				[ ] While...		< Cancel >	

*Sort order* - hovorí, podľa ktorého stĺpca sa bude triediť. Ak napr. preniesieme do okna (napr. dvakrát klik myšou) stĺpec meno, súbor sa usporiada podľa abecedy (resp. abecedne podľa mena). Ak preniesieme dátum nástupu do zamestnania, dostaneme súbor usporiadaný podľa času, v akom zamestnanci do firmy prichádzali. Vrátiť prenesený stĺpec do prvého okna možno opäť kliknutím myšou.

Databázu možno triediť nielen podľa jedného stĺpca, ale aj podľa viacerých. Predstavme si stav, že máme v zozname okrem iného pätnásť Jožkov Mrkvičkov. Ako zaviesť medzi nich systém?

Možno by bolo najlepšie usporiadanie podľa veku. Vtedy do okna *Sort order* preniesieme ako základné (master) pole *meno* a ako ďalšie *dat\_nar*.

Čo to znamená?

V normálnej situácii bude databáza usporiadaná podľa mena, no keď sa nájdu dve (alebo viac) presne rovnaké mená, počítač ich uloží podľa veku.

MENO	VEK
Kaleráb Ivan	1.1.1956
Mrkvička Jožko	12.8.1980
Mrkvička Jožko	19.5.1976
Mrkvička Jožko	3.8.1955
Mrkvička Jožko	7.11.1948
Niktoš Juraj	2.11.1975

Sorting (sortovanie, triedenie sortovaním) je triedenie databázy do novej databázy, t.j. z pôvodného neusporiadaného súboru sa vytvorí nový, v ktorom sú už recordy uložené podľa zvoleného kritéria.

Skupina *Output* slúži práve na nastavenie niektorých vlastností nového súboru.

*Felds* - rovnako ako pri zobrazovaní databázy nastavuje polia. V tomto prípade ale ide o polia, ktoré sa majú vložiť do novej - usporiadanej - databázy. Bez určenia sa do novej databázy skopírujú všetky stĺpce. Okno pre túto funkciu je rovnaké ako pri *Database* → *Setup* → *Fields*.

*Save as* - povedali sme, že sortovanie je triedenie databázy do nového súboru. V tejto položke sa zadáva meno novej databázy.

Sort Destination File:	
<ul style="list-style-type: none"> <li>▶ [ . . ]</li> <li>ARCH.DBF</li> <li>FOXHELP.DBF</li> <li>FOXUSER.DBF</li> <li>G.DBF</li> <li>KNIHY.DBF</li> <li>TOVAR.DBF</li> <li>ZIACI.DBF</li> <li>[ ] All Files</li> </ul>	Drive <input type="text" value="C"/>  Directory <input type="text" value="FOXPRO20"/>  « Save » < Cancel >

Do okna netreba vchádzať, úplne postačí napísať meno nového súboru do riadku pod voľbou.

```

Output
< Save As... > [ ] Fields...
utriedeny

```

Možnosť potvrdiť triedenie (OK) sa aktivuje, až keď zvolíte pole, podľa ktorého sa má triediť, a názov súboru, do ktorého sa bude usporiadaná databáza ukladať.

Ak zadáte názov cieľového súboru (ten, do ktorého sa bude vytvárať nová databáza) rovnaký, ako je názov práve používaného, FoxPro vám taktne naznačí, že to nejde, lebo súbor sa už používa.

```

File is in use.

```

V prípade, že zadáte meno iného (nie otvoreného) už existujúceho súboru, dostanete výstrahu a otázku, či existujúci súbor prepísať.

```

C:\FOXPRO20\KNIHY.DBF already exists,
overwrite it?
< Yes >                « No »

```

Pole, podľa ktorého triedime sa často nazýva aj **klúčové**.

Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_ABC podľa mena.

### **Postup:**

1. Databáza musí byť, samozrejme, otvorená.
2. Database → Sort.
3. Vybrať pole, podľa ktorého sa bude triediť, a preniesť ho do pravého okna. (Dva razy kliknúť na meno).
4. Zadať meno nového súboru (do riadku v časti Output).
5. Potvrdiť OK.
6. !!! Otvoriť nový súbor - ZAM\_ABC.
7. Dať ho zobrazit' (napr. **browse** do Command okna).

Usporiadajte databázu ZAMESTNANCI do súboru DAT\_NAS zostupne podľa dátumu nástupu do práce.

Najprv treba otvoriť databázu ZAMESTNANCI. V dialógu treba pred prenesením poľa s dátumom nastaviť Field options na Descending. Potom určiť pole s dátumom a ďalší postup je zrejmý.

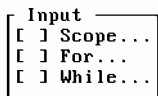
Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_AUTO podľa abecedy, ale vyberte do nej len polia s menom, platom a autom.

Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_DETI podľa počtu detí od najviac po najmenej a vyberte do nej polia s menom, platom a počtom detí.

## Skupina Scope, For, While

Doposiaľ sme taktne obchádzali skupinu *Input*.

Táto trojica sa vyskytuje takmer vo všetkých ďalších funkciách FoxPro, ktorými sa budeme zaoberať. Preto jej venujeme zvýšenú pozornosť.



### For

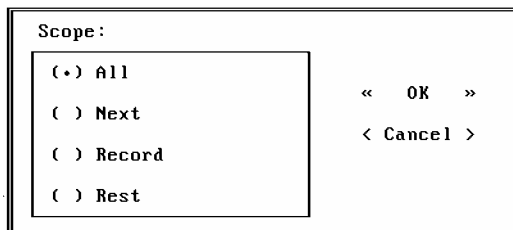
*For* má rovnaký význam (aj výzor) ako *Filter* pri *Database* → *Setup*. Nastavuje podmienku pre výber do novej databázy. T.j., ak chceme do novej databázy len zamestnancov, ktorí majú auto, napíšeme do už známeho okna príslušnú podmienku a skopírujú sa len recordy, ktoré ju spĺňajú.

### While

*While* je podobne ako *For* podmienka, no slúži na zadávanie podmienky typu: **rob pokiaľ**. T.j., ak by sme do tejto podmienky uviedli *zamestn.auto = .T.*, vykonával by sa výber dovtedy, kým by sa nenašiel record spĺňajúci podmienku (zamestnanec s autom).

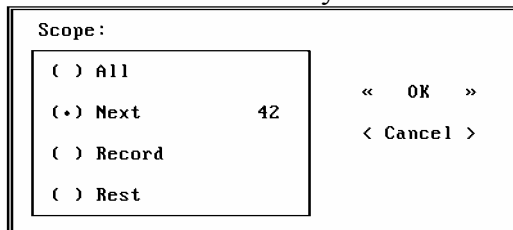
Ľudsky: do novej databázy by sa vyberali recordy. Ak má prvý zamestnanec auto, vyberie sa, ak má druhý - vyberie sa tiež. No ak sa stane, že tretí zamestnanec má v tejto položke **F**, výber sa skončí bez ohľadu na zvyšok tabuľky.

### Scope



Položka *Scope* vymedzuje oblasť použitia. T.j. databázu alebo časť databázy, s ktorou sa bude pracovať.

*All* - štandardné nastavenie. Používa sa v 99% všetkých praktických úloh. Znamená, že výber alebo triedenie sa bude vykonávať v celej databáze.



*Next* - operácia (triedenie alebo čokoľvek iné) si vyberie časť databázy od aktuálneho recordu (od toho, na ktorom je nastavený kurzor) po record o zadané číslo ďalej (ak sme nastavení na 24 recorde a zvolíme číslo 71, FoxPro bude pracovať s recordami 24 - 95) a zvyšok databázy akoby neexistoval.

*Record* - vykoná operáciu s jediným riadkom databázy (s tým, ktorého poradové číslo do kolónky napíšeme). S využitím tejto funkcie sme sa doposiaľ nestretli.

*Rest* - vykoná operáciu pre časť databázy od aktuálnej pozície po koniec.

*Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_OKUL podľa abecedy, ale vyberte do nej len zamestnancov s okuliarami.*

*Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_BEZ zostupne podľa platu a vyberte zamestnancov bez detí.*

*Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_B\_AU podľa platu, vyberte zamestnancov bez auta a polia s menom, autom, platom a počtom detí.*

*Usporiadajte databázu ZAMESTNANCI do súboru ZAM\_POSL podľa počtu detí, v prípade, že počet je rovnaký, nech sa usporiadajú podľa abecedy. Vyberte zamestnancov s autom a s okuliarami súčasne a polia s menom, autom, okuliarami, platom a počtom detí.*

## Matematické a štatistické operácie

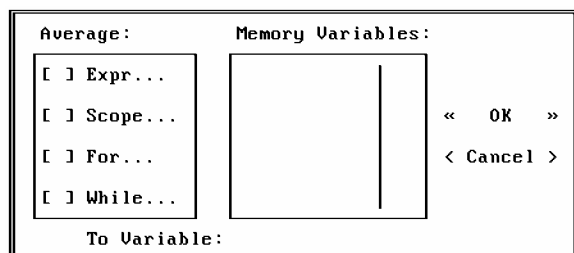
Povedali sme si, že databázové systémy slúžia na uchovávanie, ale hlavne na spracúvanie informácií. Najčastejšie používanými operáciami nad databázami sú matematické a štatistické funkcie slúžiace na vyhodnocovanie číselných údajov v databáze. FoxPro poskytuje v menu niektoré základné, iné sú dostupné len pri programovaní alebo cez *Command okno*.

### Priemer

Pravidelne sa opakujúcou úlohou v praxi je zistiť priemer niektorej z hodnôt uložených v databáze. FoxPro ju rieši jednoducho: *Database* → *Average*.

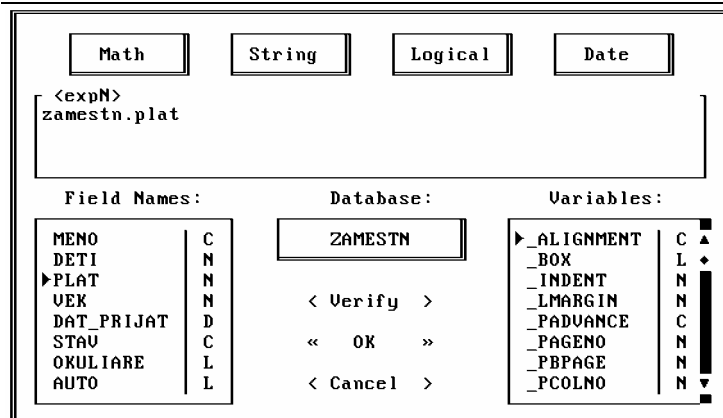
*Zistite priemerný plat zamestnancov z vašej databázy.*

Okno *Average* sa skladá z dvoch častí. Pre nás - ako neprogramátorov - je zaujímavá len jeho ľavá časť. Pravá slúži na určenie premennej, do ktorej by sme mohli vypočítanú hodnotu vložiť.

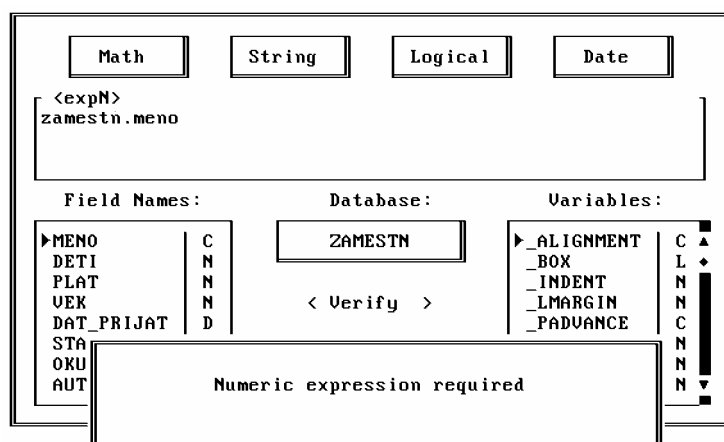


*Scope, For* a *While* už poznáme.

Položka *Expr* (expression) slúži na výber stĺpca, z ktorého chceme priemer vypočítať.



Výzor otvoreného okna je rovnaký ako výzor okna pre *For*, no nevkladáme doň žiadnu podmienku, ale len názov stĺpca, pre ktorý chceme zistiť priemer. V prípade, že vložíme názov stĺpca, ktorý nie je číselného typu, FoxPro nás na chybný krok upozorní.



Ak všetky kroky vykonáme správne (*Scope* je nastavené štandardne na *All*, podmienky zatiaľ nepotrebujeme) a odsúhlasíme *OK*, FoxPro podá informáciu o priemere na pozadí - na pracovnej ploche.

*Pokiaľ výsledok nevidíte, odsuňte všetky okná doľava.*

```
5 records averaged.
ZAMESTN.PLAT
12712.40
```

V prípade, že zabudnete, alebo zámerne nenastavíte *Expr*, získate priemery všetkých číselných stĺpcov v databáze. Riešenie je správne v prípade nastavenia aj nenastavenia. V prvom je výsledok prehľadnejší, v druhom ho získate rýchlejšie.

```
5 records averaged.
DETI      PLAT UEK
2      12712.40  32
```

*Priemer je počítaný na toľko desatinných miest, koľko je vyhradených pre príslušné pole v databáze. V prípade, že nemáte pole s desatinnými miestami, výsledok sa zaokrúhľuje.*

*Vypočítajte priemerný plat zamestnancov s jedným dieťaťom.*



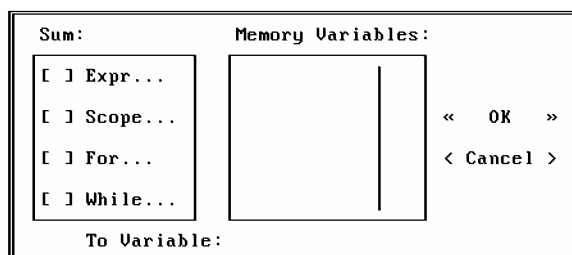
V tomto prípade potrebujeme okrem iného nastaviť aj podmienku *For*.

Počítač pri každom výsledku vypíše nielen priemer, ale aj počet recordov, ktoré do jeho počítania zahrnul.

*Vypočítajte priemerný plat zamestnancov, ktorí vlastnia auto.*

*Vypočítajte priemerný počet detí zamestnancov, ktorí vlastnia auto.*

## Suma - súčet



Táto operácia slúži na zistenie súčtu hodnôt v jednotlivých stĺpcoch. Dostaneme sa k nej *Database* → *Sum*.

Vzhľad sumy, rovnako ako aj všetky položky, zodpovedá oknu *Average*.

*Zistite počet detí, ktoré majú zamestnanci dohromady.*

V prípade, že sa nám nechce špecifikovať stĺpec, ktorý chceme sumarizovať, jednoducho len odklepeme *OK* a vo vypísaných výsledkoch nájdeme súčet pre deti.

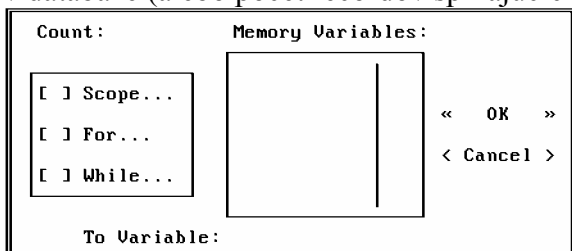
```
5 records
MAX(ZAMESTN.PLAT)
23456.00
```

*Zistite koľko stoja firmu platy zamestnancov, ktorí nemajú deti.*

*Zistite počet detí zamestnancov, ktorí majú auto.*

## Počet recordov spĺňajúcich podmienku

Ak sa stane, že nepotrebujeme ani súčet, ani priemer, ale zaujíma nás len počet recordov v databáze (alebo počet recordov spĺňajúcich podmienku), použijeme *Database* → *Count*.



Vzhľad je opäť takmer zhodný s predchádzajúcimi funkciami, logicky chýba len *Expr*. Možno nastaviť podmienky, oblasť platnosti, alebo pre zistenie celkového počtu recordov v databáze len odsúhlasiť.

Vypísaný výsledok hovorí o príslušnom počte recordov.

5 records

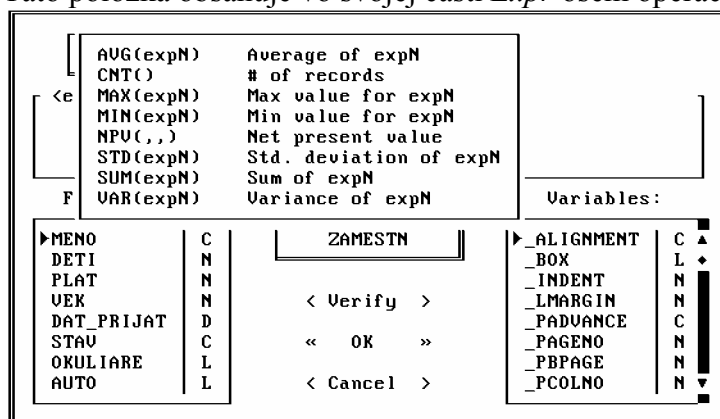
Zistite počet zamestnancov, ktorí boli zamestnaní po roku 1992.

Zistite počet zamestnancov starších ako 45 rokov s menej ako dvoma deťmi.

## Calculate

Niekoľko matematických (alebo skôr štatistických) funkcií je obsiahnutých v položke *Database* → *Calculate*.

Táto položka obsahuje vo svojej časti *Expr* osem operácií:



**AVG** - to isté ako *Average* - vypočíta priemer z vybraného stĺpca.

**CNT** - to isté ako *Count* - zistí počet recordov v databáze.

**MAX** - užitočná funkcia - nájde maximálnu hodnotu zo zvoleného stĺpca.

**MIN** - nájde minimálnu hodnotu zo zvoleného stĺpca.

**STD** - zistí smerodajnú odchýlku.

**SUM** - to isté ako *Sum* - vypočíta súčet vybraného stĺpca.

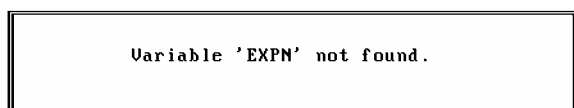
**VAR** - rozptyl hodnôt od priemeru.

Funkcie môžeme napísať aj ručne, no najjednoduchšie je vybrať ich z ponuky *Math*.

Rovnakým spôsobom možno používať príslušné funkcie aj pri zadávaní podmienok, kde ich je podstatne viac a použiteľné sú nielen matematické, ale aj funkcie pre ostatné typy:



Pri použití *Calculate* nesmieme zabudnúť zadať do zátvoriek meno stĺpca, s ktorým operáciu robíme. V prípade, že výraz nevyhovuje jazyku FoxPro, operácia sa nevykoná a vypíše sa upozornenie.



Zistite maximálny plat vašich zamestnancov.

5 records  
MAX(ZAMESTN.PLAT)  
23456.00

## Typ Memo

Vytvorte databázu *VIDEOTEKA*, do ktorej urobíte evidenciu skladu videokaziet. Mala by obsahovať: názov filmu, režiséra, hlavnú úlohu, producenta, dĺžku v minútach, cenu za kus, rok vydania, typ filmu (akčný, dobrodružný, horor, komédia, sci-fi, ...) a stručný obsah.

So všetkými položkami okrem poslednej sme sa už stretli, vieme pre ne vyhradiť príslušný počet znakov a položky naplniť. Obsah je však položka veľmi variabilná a občas si vyžaduje aj veľa slov. Vieme, že typ *Character* je obmedzený na maximálne 254 znakov a ľahko by sa mohlo stať, že pre podrobnejší obsah by nepostačoval.

Preto je pre obsahy (alebo všeobecne pre poznámky alebo typy, ktoré by nemali byť obmedzené - t.j., aby sme do nich mohli písať koľko chceme) vyhradený osobitný typ - *Memo*.

Okrem možnosti naplniť ho ľubovoľným množstvom údajov je charakteristický tým, že v pamäti zaberie iba toľko miesta, koľko údajov doň napíšeme. (Ostatné typy zaberajú toľko miesta, koľko im vyhradíme pri definovaní štruktúry bez ohľadu na to, či ich naplníme, alebo necháme prázdny.)

Pri definovaní štruktúry sa pre *Memo* vyhradí len poznámka, že pôjde o neobmedzený typ (napriek klamlivému rozsahu 10).

Name	Type	Width	Dec	Field
REZISER	Character	30		
HL_ULOHA	Character	30		
PRODUCENT	Character	20		
DLZKA	Numeric	3	0	
CENA	Numeric	3	0	
ROK	Numeric	4	0	
TYP	Character	15		
OBSAH	Memo	10		

Fields: 9      Length: 141      Available: 3859

V databáze (pri naplňaní alebo prezeraní) je v stĺpci s obsahom text *memo* (ak je položka prázdna) alebo *Memo* (ak už obsahuje údaje).

VIDEOTEK				
Dlžka	Cena	Rok	Typ	Obsah
120	230	1996	komedia	memo
100	300	1997	sci-fi	Memo

Ak sa chceme do položky dostať, buď na ňu klikneme myšou, alebo sa nastavíme kurzorom a kombináciou *Ctrl+Home* otvoríme *Memo okno*.

Ak chceme *Memo okno* opustiť, stlačíme *Ctrl+W* alebo myšou klikneme do ľavého horného rohu. V tomto prípade sa zmeny, ktoré sme urobili v obsahu položky, vnesú aj do databázy.

Ak chceme zrušiť posledné zmeny (vrátiť sa k pôvodnému obsahu *Memo* položky), stlačíme *ESC* a na otázku, či zabudnúť na zmeny, odpovieme kladne.

Discard changes?

< Yes >      « No »

---

*Pri zmene štruktúry databázy obsahujúcej Memo položky treba postupovať nanajvyšš opatrne. Ak sa ju budete snažiť zmeniť bežným spôsobom, pravdepodobne prídete o všetky údaje v databáze. FoxPro vyhlási chybu a pri snahe o prezeranie (browse) databázy sa zrúti.*

MEMO file is missing/invalid.

## OPAKOVANIE II

Okrem najzákladnejších typov - *Character* a *Numeric* (prípadne *Float*) - ponúka FoxPro ďalšie:

*Date* - typ, do ktorého sa ukladá dátum. Je vybavený kvapkou inteligencie. Nedovolí písať neexistujúce dátumy. Je preň vyhradených 8 miest.

Štandardne má FoxPro zapnuté americké zobrazovanie dátumu: MM/DD/RR, no príkazom **set date german** ho možno prepnúť na stredo európsky formát DD.MM.RR..

Pre rok sa v základnom nastavení zobrazujú dve miesta a vložený rok sa berie ako 1900+zadané dvojčíslenie. Ak chceme zobrazovať úplný formát, treba použiť príkaz **set century on** (zapni storočie).

*Logical* - najmenej miesta zaberajúci typ slúžiaci pre údaje typu áno/nie, buď/alebo. Vyhradzuje si jediný bit a môže nadobúdať len dve hodnoty **T** (true = áno) a **F** (false = nie).

*Memo* - neobmedzený (poznámkový) typ určený pre rozsiahle položky alebo pre položky, ktorých obsah vopred nie je známy, prípadne je v každom recorde úplne iný (napr. poznámok o jednom zamestnancovi môže byť obrovské množstvo, zatiaľčo o inom neevidujeme nič). Významnou vlastnosťou tohoto typu je aj to, že nezaberá žiadne zbytočné miesto - nevyhradzuje si ho ako iné typy.

Údaje uložené v *Memo* položkách sa neukladajú do súboru s koncovkou *dbf* (kam idú bežné údaje z databázy), ale používajú súbor *fpt*. Preto, keď budete prenášať databázy, nezabudnite skopírovať aj tento.

*Memo* položky sú zobrazené v databáze ako *memo* (pre prázdnu) a *Memo* (pre plnú). Editovať možno buď poklepnutím myšou alebo kombináciou *Ctrl+Home*. Zatvárame najčastejšie *Ctrl+W*.

### Oddeľovače

Každý typ pre prácu so svojimi údajmi používa vlastné oddeľovače, do ktorých ho uzatvárame: *Numeric*, *Float* - nepoužíva žiadne oddeľovače; *Character* - úvodzovky; *Date* - zložené zátvorky - {} ; *Logical* - bodky; *Memo* - úvodzovky rovnako ako *Character*.

### Skupina Scope, For, While

Skupina je charakteristická pre väčšinu funkcií vo FoxPro.

*Scope* - nastavuje rozsah platnosti.

*For* - na spracovanie sa vyberajú len recordy spĺňajúce podmienku.

*While* - na spracovanie sa vyberajú recordy, kým je splnená podmienka.

### Sortovanie

*Database* → *Sort* - triedenie (usporiadanie) do nového súboru podľa zvoleného kritéria. Do súboru možno vyberať stĺpce aj recordy spĺňajúce podmienku. Po vykonaní operácie treba nový súbor otvoriť.

V prípade zmeny poradia (zostupne, vzostupne) treba nastaviť najprv poradie a až potom preniesť stĺpec.

### Matematické a štatistické operácie

**Priemer** - *Database* → *Average*,

**Súčet (suma)** - *Database* → *Sum*,

**Počet** - *Database* → *Count*,

**Maximum, minimum ...** - *Database* → *Calculate*.

Výsledky sa vypisujú na obrazovku - na pracovnú plochu - za otvorené okná. Pri *Average* a *Sum* možno nastaviť výpočet pre jeden alebo pre všetky stĺpce (*Expr*).

**Command okno**

Slúži na ovládanie FoxPro príkazmi a zaznamenáva každú vykonanú akciu. Pomocou neho možno vyvolať funkcie nedostupné alebo len ťažko dostupné cez menu.

**Najpoužívanejšie príkazy v prostredí**

***set status on/off*** - zapnutie/vypnutie stavového riadku,

***set date german*** - prepnutie na európsky formát dátumu,

***browse*** - zobrazenie databázy do tabuľky,

***append*** - zobrazenie databázy pre pridávanie,

***use*** - zatvorí otvorenú databázu alebo s menom otvorí zadanú,

***sum, average, count*** - zistí súčet, priemer (pre všetky číselné stĺpce) a počet recordov v databáze,

***quit*** - ukončí FoxPro.

## Tlačové zostavy - reporty

Vytvorte databázu *SUSEDIA*, v ktorej budete evidovať meno, vek a povolanie svojich susedov.

Evidovanie a spracúvanie informácií je (napriek zdanlivej obsiahlosti) len časťou pracovnej náplne správcov databáz. Oveľa väčšiu časť energie venujú vytváraniu výstupných zostáv (foriem, akými sa budú údaje tlačiť na papier).

Nie je to tým, že by bola náročná na vedomosti, čas alebo zručnosť. Problém je v tom, že informácie z databázy potrebuje každú chvíľu niekto iný. Najprv príde účtovníčka a chce výstupy zamestnancov s platmi a odvodmi do fondov, potom sa ukáže personálny šéf a želá si údaje o rodinách zamestnancov, riaditeľ si zmyslí kontrolovať im pracovné cesty atď.

Pre každého "zákazníka" je potrebné vytvoriť osobitný výstup, vytlačiť ho (najčastejšie na papier) tak, aby sa poskytovali len potrebné údaje a šetrilo sa papierom aj časom príslušných úradníkov. Treba dbať aj na formu, aj na čitateľnosť.

Databázové systémy väčšinou poskytujú aj jednoduchý nástroj na vytlačenie databázy. Výstup je však neprehľadný, chaotický a temer nikdy nezodpovedá požiadavkám zákazníka.

Vyspelejšie (t.j. takmer všetky) systémy ponúkajú možnosť vytvárať výstupné zostavy aranžovaním polí, textov a rôznych prikráslení. Týmto zostavám sa vo FoxPro hovorí **reporty**.

### Vytvorenie reportu

Ak chceme vytvoriť nový report, zvolíme *File* → *New* → (.) *Report*.

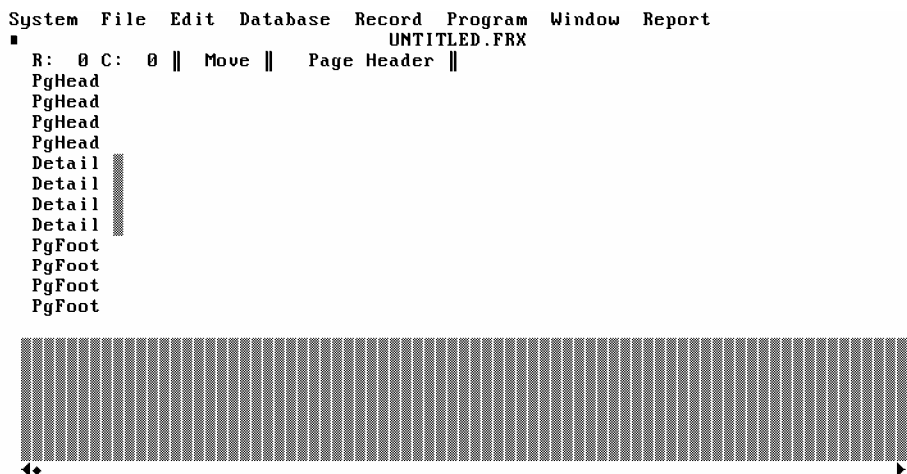
Získali sme okno rozdelené (viac-menej) na tri časti:

*PgHead* - hlavička strany,

*Detail* - obsah strany,

*PgFoot* - päta strany,

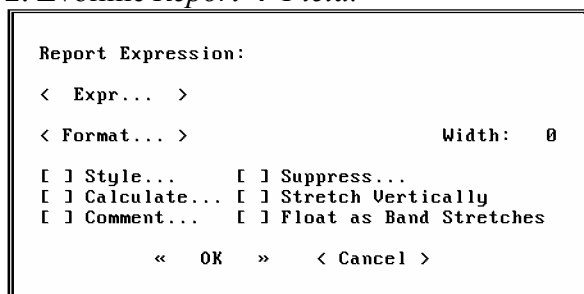
a položku *Report* v menu.



Najskôr sa budeme zaoberať len *Detailom*. Je to časť reportu, ktorá sa na strane opakuje, kým sa vmestí. Túto vetu lepšie pochopíme, keď vytvoríme prvý report.

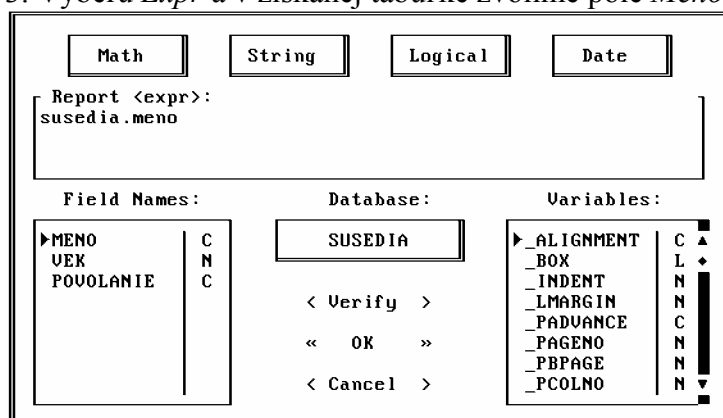
### Postup:

1. Máme otvorenú (nemusí byť zobrazená) databázu *SUSEDIA* a prázdne okno reportu.
2. Zvolíme *Report* → *Field*.



Tabuľka obsahuje spleť parametrov, ktorým sa zatiaľ nebudeme podrobnejšie venovať, ale hneď prikróčíme k:

3. Vyberu *Expr* a v získanej tabuľke zvolíme pole *Meno*.



4. Odsúhlasíme *OK* pre *Expr*, odsúhlasíme *OK* pre *Report expression* a v okne reportu dostaneme inverzne vysvietené pole.

```

■ UNTITLED.FRX
R: 2 C: 26 || Move || Page Header ||
PgHead
PgHead
PgHead      meno
PgHead
Detail
Detail
Detail
Detail
PgFoot
PgFoot
PgFoot
PgFoot

```

5. Uchopíme ho myšou a presunieme do prvého riadku *Detailu*.
6. Rovnako postupujeme pre polia *Vek* a *Povolanie*.
7. Uchopíme *Detail* za posledný riadok a posunieme nahor.

```

PgHead
Detail || meno          vek povolanie
Detail
PgFoot

```

8. Týmto je náš prvý report vytvorený. Cez *Report* → *Page preview* sa pozrieme, ako vyzerá.

```

System File Edit Database Record Program Window Report
Preview
-----
Jano Orech          33 kovac
Miso Maluch         41 zlodej
Dezo Lajos          52 invalid
Anca Fernakova     41 predavacka
« Done » < More > Column: 0

```

Získali sme obraz výstupu (podobne ako v T602 preview). Časť (riadky), ktorá je rozložená v *Detaile*, sa opakuje, kým sa nezobrazí celá databáza (raz pre každý record). Ak sa nevmestí na jednu stranu, rovnaký postup sa opakuje na ďalšej.



Do detailu sme vkladali polia (fields), čím sme systému nariadili, aby na miesta, kde sú uložené, vkladal ich obsah. Ak by sme vložili (napísali) obyčajný text, opakoval by sa rovnako.

*Zväčšite Detail o riadok a napíšte doň svoje meno. Pozorujte, ako sa zmení obraz strany.*

Povedali sme, že report sa skladá z troch častí: hlavičky (*PgHead*), päty (*PgFoot*) a samotnej strany (*Detail*). Hlavička a päta sa na každej vytlačenej strane zobrazia len raz, obsah *detailu* sa opakuje dovtedy, kým na strane nie je celá databáza alebo kým sa údaje vmestia na stranu.

*hlavička*

*detail pre 1. record  
detail pre 2. record  
detail pre 3. record  
detail pre 4. record  
detail pre 5. record  
detail pre 6. record  
detail pre 7. record*

*päta*

V reporte možno používať tri druhy objektov:

**Bežný text** (bude zobrazený rovnako, ako ho napíšete) sa najjednoduchšie vkladá priamo. Kliknite na miesto, kam chcete vložiť text myšou, a potom len jednoducho píšete.

**Polia (fields)** (názvy polí, ktoré vidíte pri vložení, sa nahradia obsahom príslušného poľa) - do reportu sa vkladajú cez menu *Report* → *Field* (v otvorenom okne zvolíte *Expr* a pole, ktoré chcete vložiť).

**Rámčeky (box)** slúžia na zvýraznenie alebo estetické dotvorenie reportu a vkladajú sa cez *Report* → *Box*.

Po vložení box bliká. Chce tým naznačiť, že môžete meniť jeho veľkosť (uchopením myšou za pravý dolný okraj a natáhovaním).

*Objekty sa vždy vkladajú na miesto, kde je v reporte nastavený kurzor.*

Prvotné uloženie objektov nie je, samozrejme, definitívne. Polohu (prípadne rozmery) môžeme meniť. Pre presunutie stačí objekt uchopiť myšou a “preložiť” na požadované miesto.

Ak chcete zmeniť veľkosť boxu, stlačte *Ctrl* a kliknite naň myšou. Rámik začne blikáť a vy môžete meniť jeho rozmery.

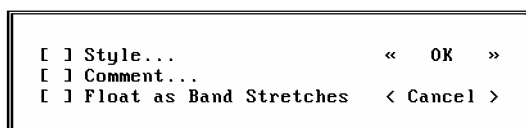
Ak chcete objekt z reportu odstrániť, kliknite naň (vysvieti sa odlišnou - najčastejšie červenou - farbou) a vymažte ho stlačením *Delete*.

Pre úpravu ďalších vlastností postačí dvojité kliknutie na príslušný objekt. Rozbalí sa okno, v ktorom môžete nastaviť dostupné vlastnosti.

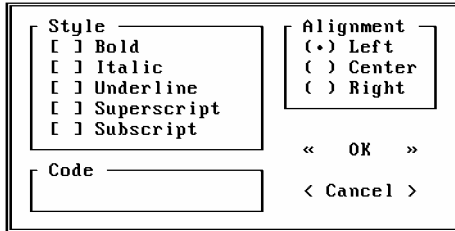
Vzhľad reportu môžeme skontrolovať cez *Report* → *Page preview*.

## Rozšírenie o vlastnosti objektov

### Vlastnosti textu



*Style* - slúži na nastavenie rezu písma a centrovania textu.



Funkčnosť tohto okna je však obmedzená na niekoľko druhov tlačiarňí. V prípade, že chceme nastavené vlastnosti písma využiť, potrebujeme väčšinou nastaviť ovládač (*Printer driver*) v menu *File* → *Printer Setup*.

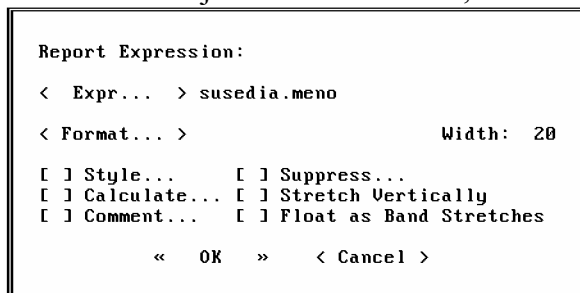
*Comment* - slúži na písanie poznámok o objekte (pri tlači sa nezobrazuje, je len informáciou pre iných užívateľov reportu).

*Float as Band Stretches* - tlačí do stĺpcov.

*Ak chcete text editovať (upravovať), netreba (nemožno) naň kliknúť. Najjednoduchšie je nastaviť sa doň textovým kurzorom (nesmie byť červený) a písať. Ak na text kliknete, zvolíte potom Report → Text (alebo stlačte ENTER).*

## Vlastnosti polí

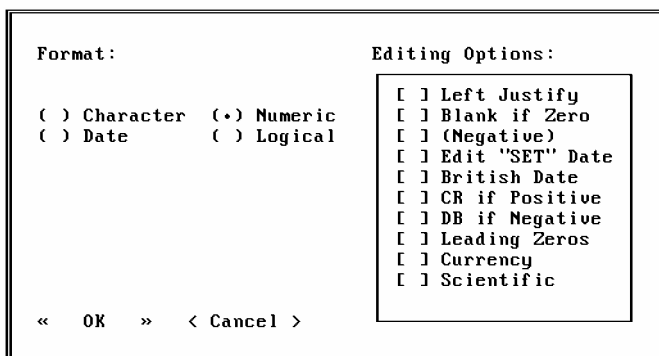
Otvorené okno je rovnaké ako okno, ktoré sa používa pri vkladaní poľa.



*Expr* - slúži na výber poľa.

*Format* - umožňuje nastaviť formát (typ) poľa a niektoré jeho vlastnosti.

Napr. v prípade poľa *Numeric* možno nastavovať:



*Left justify* - číslo sa nezarovná napravo, ako je vo FoxPro zvykom, ale zostane naľavo.

*Blank in Zero* - v prípade, že v číselnom poli sa vyskytne nula, nevypíše sa, ale vynechá sa namiesto nej prázdne miesto.

*Leading Zeros* - doplní prázdne miesta v databáze nulami.

*Scientific* - zobrazí čísla vo vedeckom formáte atď.

Ďalšie vlastnosti poľa:

*Width* - nastaví veľkosť (počet znakov), ktoré bude z daného poľa zobrazovať (ak pre pole s vyhradenými 30 znakmi nastavíme *width* 10, zobrazí sa v bežnom prípade len prvých 10 znakov). Táto voľba nastavuje len zobrazovanie. Nemení štruktúru databázy.

*Style* - nastavuje rez písma.

*Calculate* - možno nastaviť napr. tlač súčtu recordov, čísla recordu... Nastavením vlastností sa stráca pôvodný obsah poľa.

*Nothing* - štandardné nastavenie (nič sa nedeje).

*Count* - namiesto obsahu poľa vloží na dané miesto číslo recordu v databáze.

*Sum* - vypíše súčet numerických hodnôt recordov uložených v poli. Používa sa väčšinou v päte strany a uvádzajú sa doň súčty údajov na strane.

*Average* - podobný ako *Sum*, ale počíta priemer.

*Lowest, Highest* - ako predošlé; nájde minimum a maximum.

*Std. deviation, Variance* - štatistické funkcie.

*Comment* - nezobrazovaný komentár pre iných užívateľov reportu.

*Suppress* – ne/zobrazovať rovnaké hodnoty.

*Stretch Vertically* – dôležitá voľba hovoriaca o zobrazovaní do riadkov pod seba v prípade, že sa obsah položky nevojde do jedného. Najčastejšie sa používa pri tlači *Memo* položiek, ktoré takmer vždy pozostávajú z viacerých riadkov. Ak je vypnutá, z obsahu uloženého v *Memo* položke sa vytlačí len prvý riadok

## Vlastnosti rámov

Toto okno prináša len štyri nové vlastnosti pre vzhľad boxu:

*Single line* - obrysy boxu budú jednoduché (čiara).

*Double line* - obrysy budú dvojité (dve čiary vedľa seba).

*Panel* - čiary v obryse budú nahradené plnými znakmi - panelom. Výsledkom je plný obrys na šírku znaku.

*Character* - rám môžete zložiť z ľubovoľných znakov. Po zvolení tejto položky dostanete zoznam všetkých dostupných znakov, z ktorých môžete rám poskladať.

*Väčšinu znakov je schopná zobrazit' len obrazovka. V prípade, že sa ich pokúsite vytlačiť, môže sa stať, že namiesto srdiečok alebo slniečok nebudete mať nič (tlačiareň nemá znaky definované a nahradí ich medzerami).*

*Vytvorte report z predchádzajúcej databázy s tým, že pridáte popis údajov:*

**Meno: Jano**

**Povolanie: penzista**

**Vek: 95**

**Meno: Ďuro**

**Povolanie: traktorista**

**Vek: 32** *atd'.*

## Tlač reportu

Po vytvorení sa report zvyčajne aj vytlačí (za týmto účelom sme ho vytvárali). Tlač sa nastavuje a spúšťa cez *Database* → *Report*.

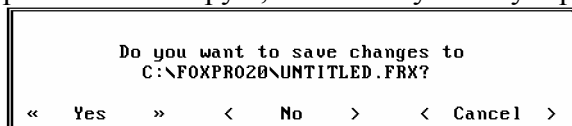
Stačí nám nastaviť v poslednej skupine výstup na tlačiareň - *To Print* a potvrdiť *OK*.

V prípade, že je všetko v poriadku, získame výstup na papieri, inak FoxPro vyhlási chybu.

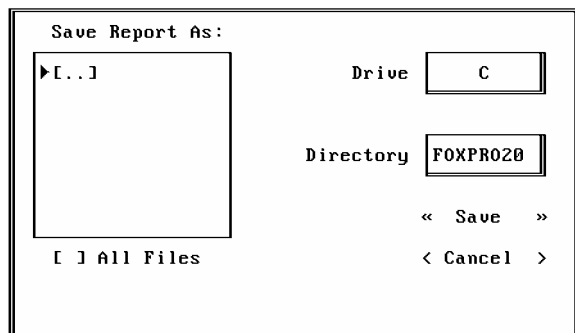
Najčastejšou je vypnuté spojenie počítača s tlačiarňou (*On line*) alebo zapnuté prerušenie (*Pause*), prípadne chýbajúci papier. V každom prípade kolíziu vyriešime a zvolením *Yes* pokračujeme v tlačení.

## Uloženie reportu

Report uzavriete kliknutím na ľavý horný roh okna. Ak ho predtým neuložíte (*File* → *Save*), počítač sa vás opýta, či chcete vytvorený report uložiť.



Ak o tlačový výstup nechcete prísť, odsúhlaste (*Yes*), napíšte meno súboru, v ktorom sa má report uchaovať, zvolíte *Save* a kladne odpovedzte na otázku, či naozaj chcete údaje uložiť.



Vytvorte databázu *DOMY* a naplňte ju. Vytvorte report v tvare:

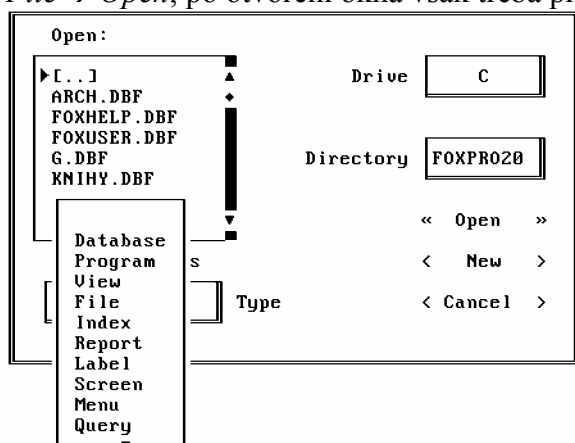
<p><i>typ domu</i></p> <p><i>poloha (adresa)</i></p> <p><i>počet izieb</i></p>	<p><i>rozloha</i></p> <p>..... <i>cena</i></p>
--	--

a orámujte ho. Report vytlačte.

## Otvorenie existujúceho reportu

Reporty sa nevytvárajú na jedno použitie, a preto sme občas nútení otvoriť už existujúci. Otvorenie reportu je takmer totožné s otváraním databázy:

*File* → *Open*; po otvorení okna však treba prestaviť *Type* na *Report*.



Ak neskôr budete otvárať databázu a nezobrazí sa vám, skontrolujte, či nemáte typ nastavený na *Report*.

Vytvorte report pre databázu *VIDEOTEKA* v podobe kartotečných lístkov.

## Hromadná korešpondencia

Vytvorte (otvorte) databázu svojich známych a navrhnete report, ktorým všetkých pozvete na oslavu svojich narodenín.

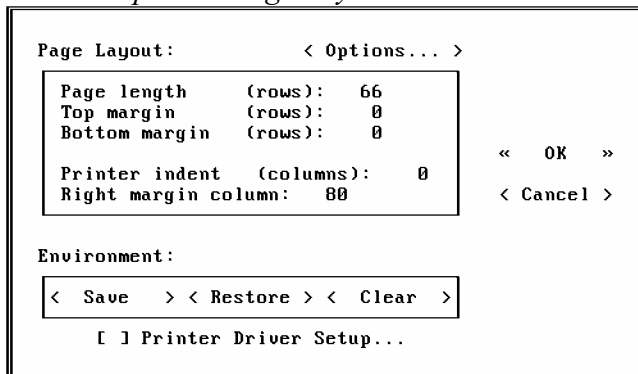
Toto je prípad, keď môžete využiť všetky výhody počítača. Napíšete jeden list, namiesto mena a adresy vložíte príslušné polia a pre každého vytlačíte.

Z estetických dôvodov by ste mali každú pozvánku vytlačiť na osobitný papier.

V bežnom prípade sa na jednu A4 zmestí niekoľko pozvánok. Aby sme povolili na každú stranu len jednu, máme dve možnosti:

1. Natiahneme *Detail* na takú veľkosť (pridáme mnoho prázdnych riadkov), že sa na stranu vojde len raz. Je to správne, ale neelegantné riešenie a v iných prípadoch by nemuselo vyhovovať.

2. Zmenšíme počítačovú stranu, t.j. nastavíme stranu s menším počtom riadkov. Počítač po vytlačení posledného vydáva tlačiarňi povel na vysunutie papiera. Počet riadkov na stranu sa mení v *Report* → *Page Layout*.



*Page length* - počet riadkov na stranu.

*Top margin* - počet riadkov pre horné záhlavie.

*Bottom margin* - počet riadkov pre päť.

*Printer indent* - ľavý okraj - počet medzier, ktoré sa dajú pred každý riadok = vzdialenosť prvého znaku od ľavého okraja.

*Right margin column* - pravý okraj - počet znakov na riadok. Nastavených 80 je pre bežné písmo na tlačiarňi A4 pre papier po výške.

V položke *Options* možno zapnúť/vypnúť vysúvanie (prázdneho) papiera pred a po vytlačení (*Page eject before printing*, *Page eject after printing*).

3. Po správnom nastavení dĺžky strany a preverení cez *Page preview* môžeme pozvánky vytlačiť.

Vytvorte databázu *MENU* a príslušný report s výzorom menu v reštauráciách.

Vytvorte databázu *VYSVEDCENIA* a príslušný report s menom, predmetmi, vymeškanými hodinami, miestom pre podpis a príslušným orámovaním. (Pozn.: pravdepodobne budete potrebovať zmeniť dĺžku strany.)

Pri práci s boxmi sa menej skúseným užívateľom často stáva, že im zmizne objekt bez toho, aby ho vymazali. Najčastejším dôvodom je jeho prekrytie iným objektom. Ak sa vám niečo podobné stane, treba popresúvať objekty myšou a pohľadať "stratenca" pod nimi.

MENO: meno
VEK: vek
POVOLANIE: povolanie

Nepříjemným problémem je aj nedoriešené zobrazovanie boxov (ak sa dva prekrývajú). Boxy sa však napriek nekompaktnému (oddelenému) vzhľadu na obrazovke vytlačia celistvo. Tento nedostatok súvisí so zobrazovaním textových znakov nielen vo FoxPro, ale aj v celom DOSe.

*Ak sa stane, že na výstupe z tlačiarne sú znaky (slová), ktoré na obrazovke (pri Page preview) nevidieť, opäť bude chyba v ich prekrytí inými objektami.*

Ak chcete presúvať viac objektov súčasne (napr. box s celým obsahom), vyberte myšou oblasť, v ktorej ležia (všetky sa vysvietia inou farbou) a uchopením niektorého z nich posuňte.

PgHead	
PgHead	.....
Detail	.....
Detail	.....
Detail	.....
Detail	.....
Detail	.....
Detail	.....
Detail	.....
PgFoot	.....
PgFoot	.....
PgFoot	.....

*Vytvorte databázu DOTAZNIK, ktorá bude obsahovať údaje o zamestnancoch: meno, priezvisko, dátum narodenia, stav, vzdelanie, počet detí, plat, auto, zdravotný stav, poznámky, predchádzajúce zamestnanie atď. Pre údaje vytvorte report, ktorý bude rozdelený na skupiny podľa oblasti, ktorú popisuje (osobné údaje, zdravotný stav, pracovné záznamy...).*

## OPAKOVANIE III

Reporty sú výstupné zostavy, podľa ktorých sa robí výstup na tlačiareň.

### Vytvorenie

*File* → *New* → (.) *Report*

### Uloženie

*File* → *Save (Save as)*

### Otvorenie existujúceho reportu

*File* → *Open* a *Type* nastaviť na *Report*.

Report sa skladá z troch častí: hlavička (*PgHead*), päta (*PgFoot*) a vnútro (*Detail*). Najdôležitejšou časťou je *Detail*, ktorý sa opakuje na výstupe pre každý record databázy.

### Druhy objektov

- text (klikneme myšou do reportu a píšeme),
- pole (field) - *Report* → *Field* - namiesto neho sa vkladá obsah príslušného recordu,
- rám (box) - orámovanie a zvýraznenie textu.

### Zmeny v reporte

- posúva sa uchopením objektu a potiahnutím,
- vlastnosti sa menia dvojklikom na objekt,
- veľkosť (nielen boxov, ale aj polí) sa mení stlačením *Ctrl* a myšou,
- mazanie - po aktivovaní (objekt sčervenie) stlačením *Delete*.

### Súčasná práca s viacerými objektami

- myšou ohraničiť oblasť, v ktorej sa nachádzajú, a uchopením jedného z nich presunúť celú skupinu.

### Zvláštnosť pre Memo

Pre polia typu *Memo* je rozumné nastaviť *Stretch vertically*, čím zabezpečíme tlač obsahu v prípade, ak pozostáva z viacerých riadkov.

### Výzor reportu

*Report* → *Page Preview*.

### Tlač reportu

*Database* → *Report* a v poslednom zoskupení nastaviť výstup na *To Print*.

### Dĺžka strany

*Report* → *Page Layout Page* → *Page Length*

- používa sa najmä v prípadoch, keď na strane má byť konkrétny počet recordov z databázy.

### Záhlavie pre prvú a poslednú stranu

Nastavuje sa vo voľbe *Report* → *Title/Summary* a umožňuje definovať osobitné záhlavie pre začiatok a pre koniec reportu.



Základné operácie, ktoré by mali poskytovať databázové systémy, pozostávajú okrem spomenutých ešte z operácií mazania, pridávania z inej databázy, kopírovania častí do iných databáz, výpočtov na úrovni recordov a hľadania.

## Mazanie recordov

Doteraz sme sa nestretli so situáciou, v ktorej by sme potrebovali z databázy vymazávať údaje. V praxi sa však bez tejto operácie obísť nemožno. Údaje, ktoré prestanú byť aktuálne, sa stávajú zbytočnými, spomaľujú a komplikujú ďalšiu prácu s databázou. V databázových systémoch je mazanie údajov pomerne zložitou a na čas náročnou operáciou. (Našťastie, len na úrovni programátorov, ktorí systém vytvorili; pre užívateľa je rovnako jednoduché ako napr. napĺňanie.)

Samotné mazanie pozostáva z dvoch krokov:

1. *logické vymazanie* - označenie recordov určených na vymazanie,
2. *fyzické vymazanie* - skutočné - nenávratné odstránenie označených recordov.

Takéto “dvojité” mazanie má niekoľko príčin.

Najdôležitejšie sú, samozrejme, bezpečnostné. Pri skutočne dôležitých informáciách si operátor nemôže dovoliť omylom nenávratne vymazať údaj. V tom lepšom prípade by ho opätovné obnovenie stálo len drahý čas, v tom horšom... Ak sa stane, že sa nejaký record omylom označí (vymaže logicky), nič sa nedeje, odznačí sa a práca pokračuje, akoby sa nič nestalo.

Druhým, nemenej podstatným, dôvodom je, že pri rozsiahlych databázach je mazanie na čas veľmi náročnou operáciou. Pokiaľ počas dňa mažeme dvadsať ráz, dvadsať ráz stratíme čas a sme pri práci blokovani. Ale - ak dvadsať ráz len logicky označíme nevyhovujúce recordy a fyzické mazanie spustíme ku koncu pracovnej doby (resp. na noc), všetko je v poriadku.

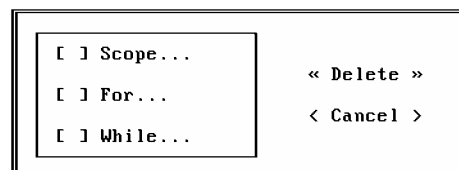
Vo FoxPro možno mazať niekoľkými spôsobmi.

Najjednoduchšie je priame mazanie, keď sa v databáze nastavíme na príslušný record a kombináciou *Ctrl+T* (*Toggle delete*) record označíme.

Označený (logicky vymazaný) record spoznáte podľa značky na ľavom okraji tabuľky. V tomto prípade slúži rovnaká kombinácia (*Ctrl+T*) aj na odznačenie.

Mesto	PSC
Abelová	98513
Abertamy	36235
Abouce	98044
Abrahám	92545
Abramová	03822
Adamov	67904
•Adolfovice	79001
•Alber	37833
Albrechtice (Karviná)	73543
•Albrechtice (Rych. n. Kn.)	51722
Albrechtice nad Útavou	39816
Albrechtice v Jizer. horách	46843
Albrechtiský	74255
•Alešince	95122

Ak chceme označiť naraz viac recordov, použijeme voľbu *Record → Delete*.



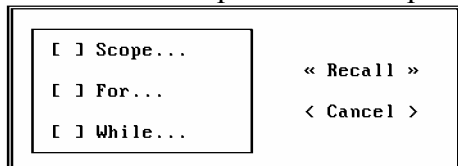
V tomto okne možno nastaviť rozsah platnosti a podmienku, podľa ktorej sa vyberú recordy.

Vzhľadom na to, že operácia mazania patrí medzi “nebezpečné”, je rozsah platnosti štandardne nastavený len na jeden record. Ak chceme aplikovať podmienku na celú databázu, potrebujeme nastaviť Scope na All.

## Rušenie logického vymazania

Na odznačovanie recordov sa používa voľba *Record* → *Recall*.

Pre túto funkciu platia rovnaké pravidlá ako pre *Delete*.

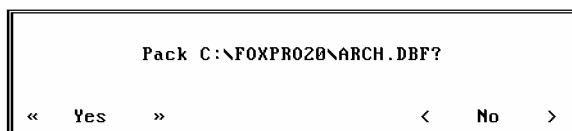


Po všetkých označovaniach, úpravách, odznačovaniach a kontrolách môže konečne dôjsť k fyzickému vymazaniu označených recordov.

Tejto operácii sa hovorí aj **pakovanie databázy**, pretože v podstate ide o nové rozloženie recordov, keď sa na miesta vymazaných dostávajú tie, čo zostanú.

Databáza sa pakuje cez menu voľbou *Databaze* → *Pack* alebo príkazom (cez *Command okno*) **pack**.

Napokon dostanete ešte poslednú možnosť na zrušenie fyzickej likvidácie a po kladnej odpovedi sa pakovanie vykoná.



Otvorte databázu *VIDEOTEKA* a vymažte všetky akčné filmy.

Otvorte databázu *VYSVEDCENIE* a vymažte všetkých žiakov, ktorí mali z matematiky lepšiu známku ako 3.

## Závislosti

Spolu s funkciami *Delete* a *Recall* sa v jednej skupine menu nachádza aj *Replace*.

Táto položka slúži na nahradzovanie hodnôt v stĺpci, hromadné napĺňanie alebo počítanie hodnoty v závislosti na iných stĺpcoch.

Vytvorte databázu *VELKOOBCHOD* s položkami názov tovaru, cena za kus, dph, počet kusov, cena spolu. Naplňte len stĺpce názov, cena za kus a počet kusov.

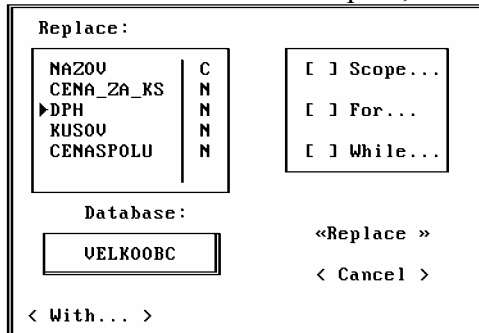
Predpokladajme, že DPH je pre všetky druhy tovaru 23%. Potom pre každú položku v *DPH* platí, že  $DPH = \text{cena za kus} * 0.23$ .

Keby sme databázu naplňali ručne, potrebovali by sme pre každý tovar počítať  $0.23 * \text{cena za kus}$  a po jednom by sme naplňali databázu. Našťastie, existuje funkcia *Replace*, ktorá všetko urobí za nás.

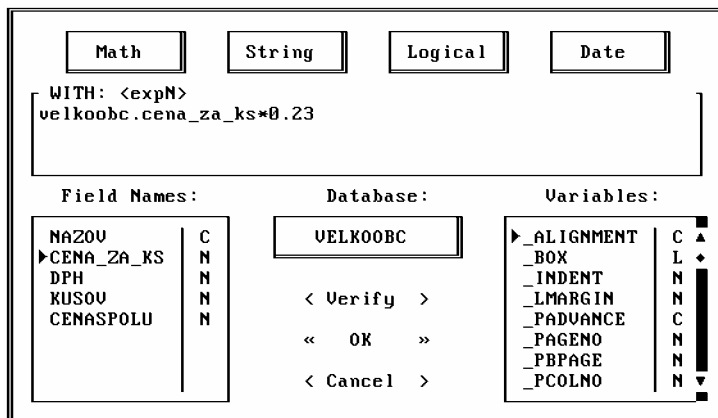
### Postup:

1. *Record* → *Replace*

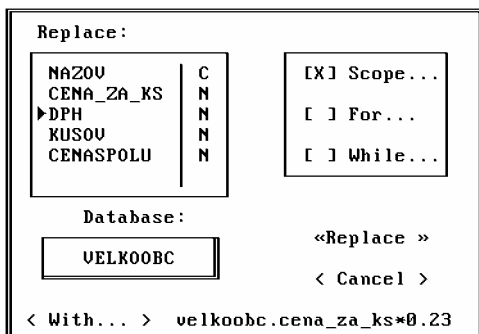
2. Nastavíme sa v okne na pole, ktoré chceme nahradiť - *DPH*.



3. Vojdeme do okna *With*. Prostredie je nám známe, určíme závislosť  $Cena\_za\_ks * 0.23$  a odsúhlasíme.



4. Rovnako ako *Delete* a *Recall*, aj *Replace* patrí medzi funkcie so zvýšeným nebezpečenstvom straty údajov, preto je v oblasti platnosti nastavené *Next*. Prestavíme *Scope* na *All*.



Čo to znamená?

Nastavením v okne *Replace* na pole *DPH* sme povedali, že chceme nahradiť stĺpec *DPH*.

Udaním  $Cena\_za\_ks*0.23$  do položky *With* (s) sme určili, že *DPH* nahradíme 23% z hodnoty tovaru.

A výberom *Scope* → *All* sme prikázali urobiť operáciu s celou databázou.

Po odsúhlasení *OK* sa v databáze nahradí stĺpec prikázanými hodnotami.

Rovnakým spôsobom určte cenu spolu.

$$(cena\_spolu = cena\_za\_ks * kusov)$$

Pridajte položku pre celkovú cenu aj s dph a naplňte ju.

$$(cena\_total = (cena\_za\_ks + dph) * kusov)$$

Z danej databázy vytvorte report a vytlačte ho ako ponuku pre maloobchody.

Vytvorte databázu DAN\_PRIZNANIE s položkami firma, obrat, výdavky, ktoré naplníte, a s položkami hrubý zisk, zdravotné poistenie, fond zamestnanosti, sociálny fond, daňový základ, daň, čistý zisk so závislosťami:

- hrubý zisk = obrat - výdavky

- zdravotné poistenie = hrubý zisk \* 0.135

- sociálny fond = hrubý zisk \* 0.323

- fond zamestnanosti = hrubý zisk \* 0.06

- daňový základ = hrubý zisk - fond zamestnanosti - sociálny fond - zdravotné poistenie

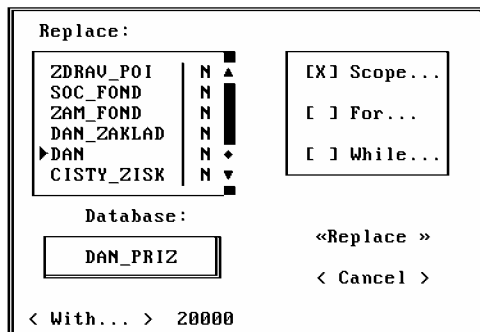
- daň = (daňový základ - 21000) \* 0.20

- čistý zisk = daňový základ - daň.

Vytvorte formulár pre takéto daňové priznanie.

Pomocou Replace môžete nielen počítať závislosti, ale aj naplňovať stĺpce konkrétnymi hodnotami.

V predchádzajúcej databáze nahraďte položku daň paušálnou sumou 20000.



Určíte ste si všimli, že po tejto operácii sa zmenili len údaje v stĺpci daň a údaje v stĺpci čistý zisk zostali pôvodné. FoxPro si nedokáže zapamätať vykonané operácie a použité závislosti, preto v takýchto prípadoch musíte pôvodné operácie zopakovať (môžete použiť *Command okno* a odklepnúť už vykonané operácie).

Vypočítajte čistý zisk pre zmenené údaje.

Rovnako ako pri iných operáciách možno aj pri *Replace* používať podmienku. V skutočnosti percento pre zdanenie závisí od hrubého zisku.

Vypočítajte daň podľa platných tabuliek, napr. pre daňový základ do

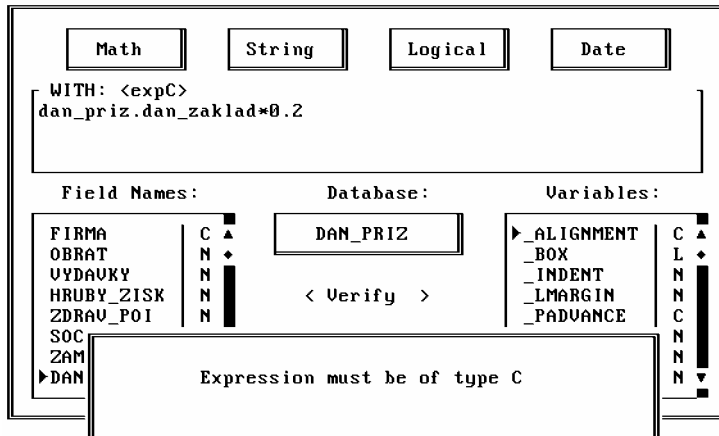
60000	- 15 %
120000	- 20 %
180000	- 25 %
540000	- 32%

1080000            - 40%  
 viac                - 42%

Pozn.: Uvedené údaje nemusia zodpovedať skutočnosti.

Takže okrem nahradenia hodnotou zadáme aj podmienky (napr. pre *hrubý zisk*  $\leq 100000$  and *hrubý zisk*  $> 75000$  dáme nahradiť daň hodnotou  $\text{hrubý zisk} * 0.25$ ).

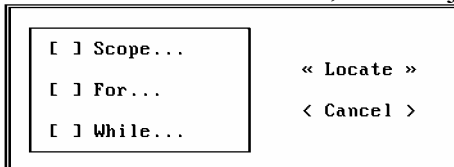
Častou chybou je nesprávne nastavenie kurzora v okne *Replace*. Ak sme nastavení na poli typu *Character* a chceme ho nahrádzať numerickými hodnotami, vyvoláme chybu.



## Vyhľadávanie

Predstavte si telefónny zoznam s tisíckami položiek, v ktorom chcete nájsť osobu nachádzajúcu sa kdesi v strede databázy.

Môžete sa pohybovať ručne (šípkami), no oveľa hospodárnejší a rýchlejší spôsob je použiť funkciu *Record* → *Locate*, ktorá nájde hľadanú položku (resp. nastaví na ňu kurzor).



Zadaním podmienky *For* v *Locate* sa kurzor nastaví na prvý record, ktorý podmienku spĺňa.

Zvolením *Record* → *Continue* sa nastavíte na ďalší podmienku spĺňajúci record alebo (ak už žiaden nie je) na koniec databázy.

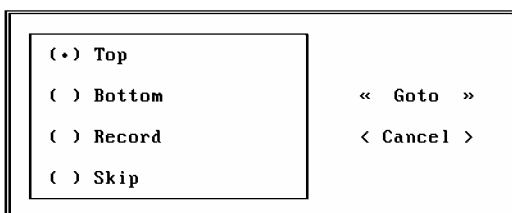
Po databáze sa možno zrýchlene pohybovať aj ďalším spôsobom - *Record* → *Goto*.

*Top* - kurzor skočí na začiatok databázy.

*Bottom* - kurzor skočí na koniec databázy.

*Record* - kurzor skočí na record s udaným poradovým číslom.

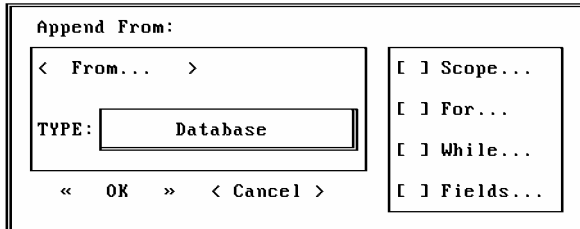
*Skip* - kurzor preskočí zadaný počet recordov.



## Spájanie databáz

Ak sa údaje zbierajú od viacerých “objektov”, riešenie problému si vyžaduje spojiť všetky údaje do jednej databázy.

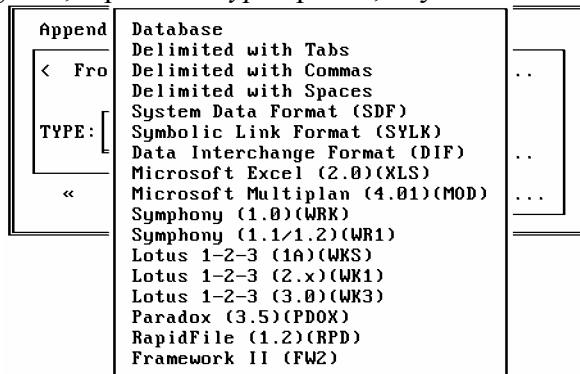
Na tento účel slúži vo FoxPro funkcia *Database* → *Append from*. Pridáva do otvorenej databázy údaje zo zvolenej.



Databázy spojíte do jednej tak, že vo *From* vyberiete databázu, ktorej obsah chcete pridať na koniec otvorenej. Pridávaná databáza zostane v pôvodnom tvare, len údaje z nej sa skopírujú do otvorenej.

Výhodou funkcie je aj možnosť pridávať nedatabázové súbory ako databázu.

Napr. v T602 vytvoríte tabuľku, ktorú chcete neskôr spracovať ako databázu. Zvolíte *Append from*; v položke *Type* spôsob, akým sú v texte oddelené stĺpce, a odsúhlasíte.



*Database* - klasická databáza.

*Delimited with Tabs* - stĺpce sú oddelené tabelátormi.

*Delimited with Commas* - stĺpce sú oddelené čiarkami.

*Delimited with Spaces* - stĺpce sú oddelené medzerami.

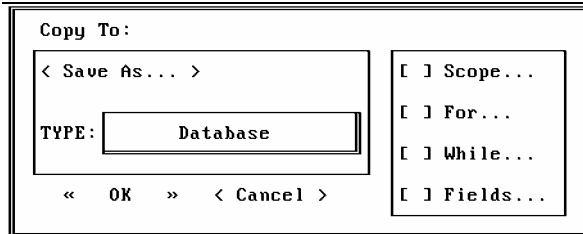
*Ostatné typy* - prislúchajú iným databázovým systémom, z ktorých môžeme údaje importovať do FoxPro.

Pri spájaní databáz môžeme určiť podmienku (*For*) aj polia (*Fields*), ktoré chceme pridávať.

*V praxi sme sa stretli (dosť často vo verzii 2.0) s prípadmi, že funkcia Append from jednoducho nefungovala. Z pridávanej databázy si povyberala len niektoré stĺpce, ostatné nahradila prázdnyimi napriek tomu, že polia boli rovnakých typov aj rozmerov.*

## Kopírovanie

Opačnou funkciou k *Append from* je *Database* → *Copy to*. Ako už názov napovedá, ide o funkciu, ktorá kopíruje údaje z otvorenej databázy do inej.



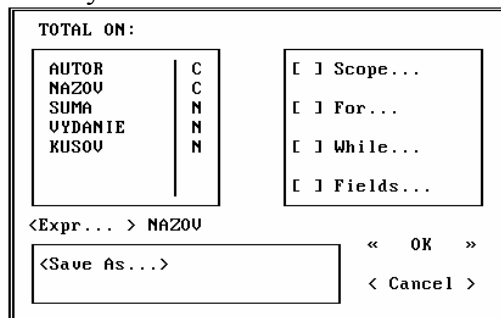
Funguje na rovnakom princípe ako *Sort*, no výsledkom nie je usporiadaná databáza, ale recordy v rovnakom poradí (niektoré možno vynechať), ako boli v pôvodnej databáze.

Rozšírením oproti *Sort* je možnosť exportovať údaje do iných formátov (napr. do textu, prípadne iných databázových systémov). Typy sú rovnaké ako *Append from*.

## Total

Funkcia *Database* → *Total* je prostriedkom na sčítanie numerických položiek recordov, ktoré majú niektorú hodnotu rovnakú.

Ľudsky: Máme niekoľkoročnú databázu kníh. Knihy nevozia naraz, prinášajú ich postupne.



Vždy, keď privezú knihy, zaeviduje sa názov a zodpovedajúci počet kusov. Stane sa, že raz privezú 50 kníh s rovnakým titulom a o týždeň 23 takých istých. Knihy sú potom evidované v dvoch recordoch, aj keď by úplne stačil jeden. Všetko je rovnaké – názov, autor, cena, vydanie; dátum neevidujeme. Časom sa v databáze nahromadí viac takýchto položiek (hovoríme o redundancii).

Voľba *Total* slúži práve na riešenie podobných problémov. Vytvorí novú databázu, v ktorej bude názov knihy len raz a zodpovedať mu bude počet všetkých kusov s rovnakým názvom.

Kliknutím do *Total on* vyberieme pole, podľa ktorého sa bude sumarizovať (t.j. názov knihy). Podmienkou úspešného vyriešenia je usporiadanie databázy podľa položky, pre ktorú sa sumarizovanie vykonáva. (Rovnaké knihy musia nasledovať za sebou, inak sa sčítovanie nevykoná správne.)

Najrozumnejšie je teda pred samotnou operáciou databázu utriediť alebo zindexovať (viď ďalej).

Možno, samozrejme, nastavovať podmienky a oblasť platnosti.

Nepříjemné je, že nemožno vybrať položku, pre ktorú sa má súčet urobiť. FoxPro totiž sčíta všetky numerické položky. Problém možno obísť len tak, že do výslednej databázy vyberieme pole s údajom (napr. názov knihy) a pole s jeho hodnotou (napr. počet kníh z jedným názvom).

*Vytvorte databázu automobilov, v ktorej budete evidovať počet kilometrov, ktoré najazdili. Vygenerujte databázu, v ktorej bude informácia o tom, ktorý typ koľko najazdil.*



Vytvorte databázu dodávok kníh a z nej vygenerujte:

- databázu, v ktorej budú počty kníh pre rovnaké názvy,
- databázu, v ktorej bude počet kníh privezených v rovnaké dni.

Nezabudnite, že databáza musí byť pred použitím Total utriedená.

## Operácie s tabuľkou

*Zmena poradia stĺpcov* - sa vykoná uchopením za hlavičku tabuľky (názov stĺpca) a ťahaním.

*Novú šírku stĺpca* - nastavíte uchopením za okraj (hranicu) stĺpca a ťahaním.

*Zdvojenie tabuľky* - dva pohľady na tabuľku (v každom môžete mať zobrazený iný stĺpec a iný riadok, prípadne môžete pracovať v inom režime) - dosiahnete uchopením a posunutím dvojšípky v spodnej časti okna s databázou.

SUSEDIA		
Meno	Vek	Povolanie
Jano Orech	33	kovac
Miso Maluch	41	zlodej
Dezo Lajos	52	invalid
Anca Fernakova	41	predavacka

## OPAKOVANIE IV

### Mazanie

Mazanie recordov v databáze prebieha v dvoch fázach:

1. *logické (označenie)*,
2. *fyzické (pakovanie)*.

Deje sa tak z bezpečnostných a časových dôvodov, lebo mazanie je v rozsiahlych databázach náročné na čas.

Mazať možno po jednom recorde (*Ctrl+T*) alebo hromadne (*Record → Delete*). Rovnako možno logické mazanie aj zrušiť: po jednom recorde - *Ctrl+T*, naraz - *Record → Recall*.

Na fyzické mazanie slúži *Database → Pack*.

### Závislosti (hromadné napĺňanie)

Databázy možno napĺňať ručne alebo (ak existujú medzi stĺpcami matematické vzťahy, prípadne sú v stĺpci rovnaké hodnoty) hromadne (*Record → Replace*).

V okne zadávame, čo (*Replace*), čím (*With*), za akých podmienok (*For*) a v akej oblasti (*Scope*) nahrádzať.

### Vyhľadávanie a presun

Na tieto operácie slúži *Record → Locate*, keď sa kurzor nastaví na prvý record spĺňajúci podmienku; *Record → Continue*, keď skočí na najbližší ďalší vyhovujúci nastaveniu a *Record → Goto*, keď kurzor možno nastaviť na začiatok, koniec alebo na poradovým číslom zvolený record.

### Spájanie databáz

*Database → Append from*

- kopíruje zvolenú databázu (alebo textový súbor) na koniec otvorenej.

### Kopírovanie databáz

*Database → Copy to*

- kopíruje aktuálnu databázu do inej (alebo textového súboru).

### Sumarizovanie

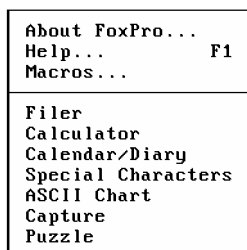
alebo sčítovanie numerických položiek rovnakých recordov - *Database → Total*.

## Menu FoxPro

Základné funkcie FoxPro sú zhrnuté v menu, ktoré okrem myši možno aktivovať stlačením *F10* alebo *Alt*.

Pre FoxPro je charakteristické, že v ponuke sú vždy len tie funkcie, ktoré majú zmysel. (Menu sa mení pri každej skupine akcií - napr. iné menu je pri bežnom používaní; iné, keď navrhujeme štruktúru, prípadne vytvárame report. Rovnako v prípade, ak nie je otvorená žiadna databáza, nemožno aktivovať položku *Record*.)

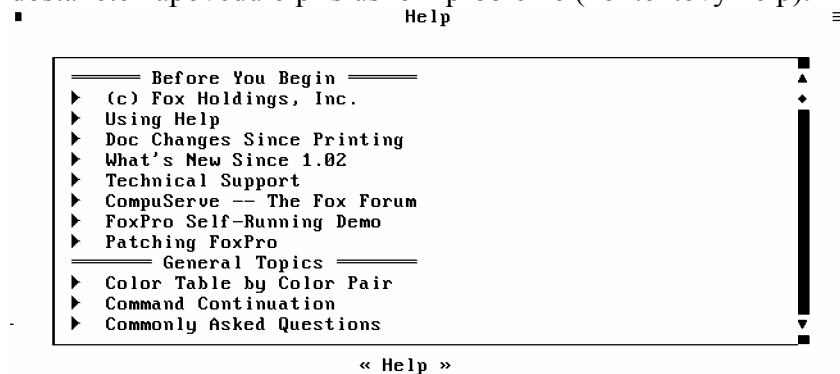
## Podmenu System



Obsahuje základné informačné a užívateľské položky, bez ktorých by sme sa vo FoxPro obišli, no svojou prítomnosťou spríjemňujú a (aspoň trochu) uľahčujú prácu.

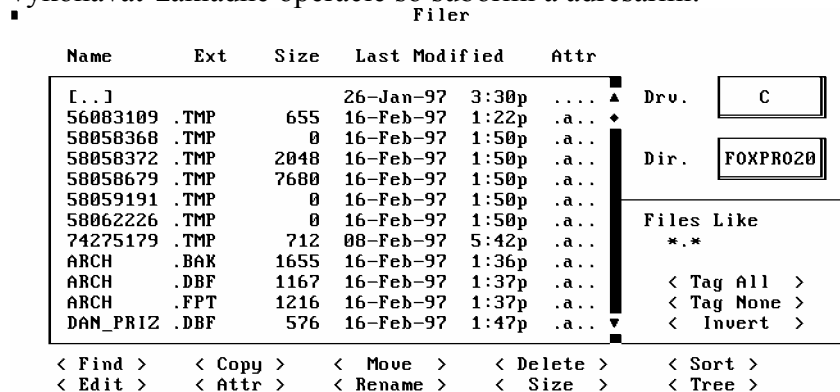
**About FoxPro** - podáva údaje o programe a počítači.

**Help** - pomoc. Po stlačení *F1* alebo aktivovaní cez menu sa otvorí okno, ktoré obsahuje zoznam hesiel, ktoré sú k dispozícii. V prípade, že stlačíte *F1* v niektorej z položiek menu, dostanete nápovedu o príslušnom probléme (kontextový help).



**Macros** - umožňujú definovať makrá (podobne ako v T602).

**Filer** - správca súborov. Ide o ďalšiu oklieštenú kópiu Norton Commandera, v ktorej možno vykonávať základné operácie so súbormi a adresármi.



**Calculator** - kalkulačka so základnými operáciami.

**Calendar/Diary** - kalendár ponúkajúci pohľad na aktuálny mesiac s možnosťou listovať o mesiac (rok) dopredu i dozadu a nastavovať sa na aktuálny dátum. V prípade, že chcete použiť *Diary* a písať si poznámky pre konkrétny deň, tabelátorom sa prepnete na diár a *Shift+Tab* späť na kalendár.

Calendar/Diary

February 1997						
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

< < M > < M > > < < Y > < Y > > < Today >

**ASCII chart** - obsahuje všetkých 255 zobraziteľných znakov, ktoré sú k dispozícii a rovnakým spôsobom ako zo *Special Characters* ich umožňuje vybrať. Tabuľka okrem toho obsahuje aj číselný kód znaku v desiatkovej a šestnástkovej sústave.

**Special Characters** - špeciálne znaky. Vybraná časť zo všetkých znakov, ktoré možno vhodne používať najmä pri tvorbe reportov. Znak na pozíciu kurzora dostanete dvojnásobným kliknutím myšou.

Special Characters

¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ à á â ã
ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö × ø ù ú û ü ý þ
ÿ

**Capture** - umožňuje kopírovanie častí obrazovky (pri práci s databázami prakticky nevyužiteľné).

**Puzzle** - logická hra, ku ktorej sa možno utiekať, keď ste zúfali zo všetkého naokolo. Cieľom je poukladať zamiešané štvorce od 1 do 15 za pomoci jedného prázdneho. Mieša sa *Shuffle*, posúva sa kliknutím na štvorec.

Puzzle

	6	3	12
5	15	2	9
10	14	13	7
1	4	8	11

< Shuffle >

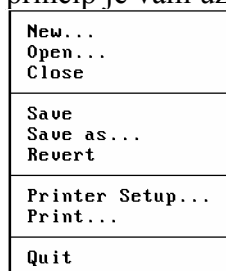
Puzzle

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

< Shuffle >

## Podmenu File

**New** - vytvorenie nového súboru zvoleného typu: *.Database* - vytvorí novú databázu. Tento princíp je vám už dokonale známy.



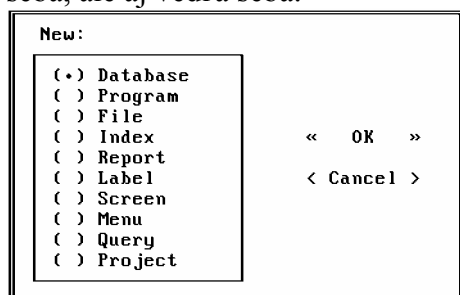
**Program** - otvorí prázdne okno, do ktorého možno písať program v jazyku FoxPro. Ten pozostáva zo slov, ktorých časť vám je už tiež známa (príkazy písané do *Command okna*).

**File** - vytvorí obyčajný súbor. Výsledok je rovnaký ako *Shift +F4* v Norton Commanderi.

**Index** - vytvorí súbor pre usporiadanie (budeme sa ním zaoberať ďalej).

**Report** - vytvorí výstupnú zostavu.

**Label** - tiež vytvorí výstupnú zostavu, ale navyše umožní ukladať údaje z recordov nielen pod seba, ale aj vedľa seba.



**Screen** - vytvorí užívateľskú obrazovku. Využívajú ju najmä programátori, ktorí navrhnu vzhľad obrazovky a nechajú si vyrobiť program, ktorý ju nakreslí.

**Menu** - podobný ako *Screen*. Programátor si nakreslí, rozloží a definuje menu na obrazovke a FoxPro mu naň vytvorí program.

**Query** - slúži na spájanie databáz a vytváranie dotazov.

**Project** - slúži (okrem iného) na vytváranie *exe* súborov z programov FoxPro.

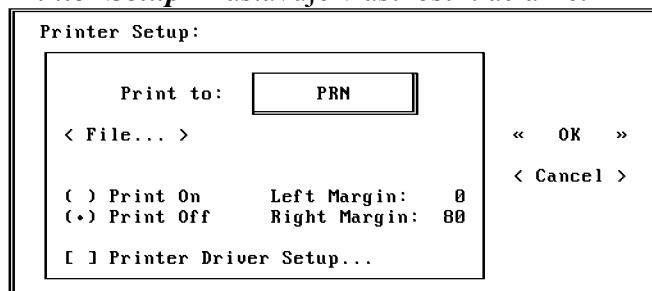
**Open** - otvorí súbor. Pri otváraní treba myslieť aj na správny typ súboru.

**Close** - zatvorí aktívne okno. V prípade, že chcete zavrieť databázu, použite príkaz *use*.

**Save, Save as** - slúžia na uloženie súboru. Pri otvorenej databáze nie sú aktívne, pretože databáza sa ukladá priamo na disk. Využívali sme ich najmä pri ukladaní reportov.

**Revert** - slúži na obnovenie súboru z disku. Pokiaľ sme v nejakom súbore urobili zmeny, neuložili ich na disk a chceme sa vrátiť k pôvodnému obsahu, zvolíme túto funkciu. Nefunguje pre databázy, ktoré sa na disk zapisujú priebežne a automaticky.

**Printer Setup** - nastavuje vlastnosti tlačiarne.



Umožňuje nastaviť tlač nielen na tlačiareň, ale aj do súboru (vtedy sa text, ktorý by normálne išiel na tlačiareň, ukladá v rovnakom formáte na disk - do súboru) po nastavení *Print to* na *File*.

*Print on/off* zapína alebo vypína komunikáciu s tlačiarňou.

*Printer Driver Setup* umožňuje nastaviť pre FoxPro inteligentnú komunikáciu s neznámou tlačiarňou (šikmé, tučné písmo...).

**Print** - vytlačí zvolené okno (*Windows*) alebo vybraný súbor. Neslúži na tlač databáz v tabuľkovej forme.

**Quit** - ukončí FoxPro.

## Podmenu Edit

Podmenu slúži na prácu s textom a používa sa skôr pri iných činnostiach ako pri práci s databázami. V tomto poňatí nie je pre nás zaujímavé.

Undo	^U
Redo	^R
Cut	^X
Copy	^C
Paste	^V
Clear	
Select All	^A
Goto Line...	
Find...	^F
Find Again	^G
Replace And Find Again	^E
Replace All	
Preferences...	

## Podmenu Database

Setup...
Browse
Append From...
Copy To...
Sort...
Total...
Average...
Count...
Sum...
Calculate...
Report...
Label...
Pack
Reindex

**Setup** - umožňuje zmeny v štruktúre databázy (*Modify*), filtráciu - výber recordov spĺňajúcich podmienku (*Filter*), projekciu - výber stĺpcov (*Set Fields*) a zobrazenie poľa podľa formátového súboru (*Format*).

**Browse** - zobrazí databázu v režime, v akom bola zobrazená naposledy.

**Append from** - pridáva k otvorenej databáze údaje z inej databázy alebo iného (textového) súboru.

**Copy to** - kopíruje recordy (alebo časť z nich) do inej databázy, prípadne z nej vytvára textový súbor.

**Sort** - vytvorí nový súbor, v ktorom je databáza usporiadaná podľa zvoleného poľa.

**Total** - spočíta numerické hodnoty pre rovnaké položky. Databáza musí byť usporiadaná.

**Average** - počíta priemer numerických položiek.

**Count** - zisťuje počet recordov v databáze.

**Sum** - vypočíta súčet stĺpcov.

**Calculate** - obsahuje osem matematických a štatistických funkcií.

**Report** - slúži na tlač reportov. Pre výstup na tlačiareň má byť v poslednej skupine aktivizovaná voľba *To Print*.

**Label** - slúži na tlač labelov (menoviek, vizitiek).

**Pack** - pakuje databázu - stará sa o fyzické odstránenie logicky vymazaných recordov.

**Reindex** - reindexuje databázu (vid' ďalej).

## Podmenu Record

Append Change
Goto... Locate... Continue ^K Seek...
Replace... Delete... Recall...

**Append** - zobrazí databázu v režime *append* a umožní vkladať ďalšie údaje.

**Change** - funguje rovnako ako *Append*, ale umožňuje len upravovať (nie pridávať) recordy.

**Goto** - nastaví kurzor na začiatok, koniec alebo na požadovaný record zadaný číslom.

**Locate** - nastaví kurzor na record spĺňajúci zadanú podmienku.

**Continue** - nastaví kurzor na najbližší ďalší record spĺňajúci podmienku.

**Seek** - prehľadáva zindexovaný súbor.

**Replace** - umožňuje naplňať závislé stĺpce.

**Delete** - logicky maže (označuje) zvolené recordy.

**Recall** - ruší označenie logicky vymazaných recordov.

## Podmenu Program

Do... ^D
Cancel Resume ^M
Compile... Generate... FoxDoc FoxGraph...

Slúži na spúšťanie a kontrolu správnosti programu. Okrem iného umožňuje aj prekladať vytvorené obrazovky a menu do programovacieho jazyka. Nachádza sa mimo okruhu nášho záujmu.

## Podmenu Window

Hide Clear
Move ^F7 Size ^F8 Zoom ↑ ^F10 Zoom ↓ ^F9 Cycle ^F1 Color...
Command ^F2 Debug Trace View

**Hide** - skryje aktívne okno bez toho, aby sa zrušilo.

**Clear** - vymaže obsah aktívneho okna.

**Move** - umožní presun okna.

**Size** - dovoľí meniť rozmery okna.

**Zoom ↑** - zväčší okno na celú obrazovku.

**Zoom ↓** - zmenší okno na pôvodnú veľkosť.

**Cycle** - preskočí na ďalšie otvorené okno.

**Color** - umožní zmeniť farby prostredia FoxPro.

**Command** - aktivizuje *Command okno*.

**Debug, Trace** - určené pre testovanie programov.

**View** - obsahuje nastavenia FoxPro a umožňuje meniť ich.

\*\*\*\*\*

Pod týmito voľbami sa občas môžu nachádzať ďalšie riadky, ktoré obsahujú zoznam všetkých otvorených okien. V prípade, že zvolíte niektoré z nich, FoxPro vás doň prepne.

## Podmenu Browse

Change Grid Off Unlink Partitions Change Partition ^H
Size Field Move Field Resize Partitions
Goto... Seek... Toggle Delete ^T Append Record ^N

Podmenu sa aktivuje v prípade zobrazenia databázy.

**Browse (Change)** - prepne do jedného z dvoch režimov zobrazenia databázy.

**Grid Off (Grid On)** - vypne (zapne) zobrazenie čiar, ktorými sú v režime *browse* oddelené stĺpce.

**Link/Unlink Partitions** - sú aktívne v prípade, ak tabuľku databázy rozdelíme na dve časti, a umožňujú pohybovať sa buď po oboch naraz, alebo len po jednej z nich.

**Change Partition** - v zdvojenej tabuľke preskakuje medzi pohľadmi.

**Size Field** - zmení šírku stĺpca.

**Move Field** - zmení poradie stĺpcov.

**Resize Partitions** - zdvojí tabuľku.

**Goto** = Record → Goto

**Seek** = Record → Seek

**Toggle Delete** = Ctrl+T (logicky zmaže, resp. odznačí aktuálny record).

**Append Record** - pridá prázdny record.

## Podmenu Report

Page Layout... Page Preview... ^I
Data Grouping... Title/Summary... Variables...
Box ^B Field... ^F Text ^T
Add Line ^N Remove Line ^O
Bring to Front ^G Send to Back ^J Center
Group Ungroup
Quick Report...

**Page Layout** - obsahuje vlastnosti strany (počet riadkov, stĺpcov, záhlavia, informácie o vysúvaní papiera - pred a po tlačení - atď.).



**Page Preview** - ukáže obraz stránky.

**Data Grouping** - zoskupenie údajov podľa spoločného znaku (podobne ako *Total*).

**Title/Summary** - pridáva osobitné "záhlavie" na začiatok a na koniec celého reportu.

**Variables** - práca s premennými.

**Box** - vloží do reportu rám.

**Field** - vloží do reportu pole.

**Text** - umožní upravovať aktívny text.

**Add Line** - vloží riadok nad pozíciu kurzora.

**Remove Line** - vymaže riadok, na ktorom je nastavený kurzor.

**Bring to Front, Send to Back** - súvisí s polohou objektov, ktoré sú na jednom mieste a prekrývajú sa - *Bring to Front* presunie do popredia, *Send to Back* presunie do pozadia.

**Center** - vycentruje objekt do stredu riadku.

**Group** - "zlepí" aktívne objekty a umožní pracovať s nimi (presúvať) ako s jedným.

**Ungroup** - rozlepi objekty spojené *Group*.

**Quick Report** - vygeneruje report v "surovej" forme.

## Podmenu Run

Je vymoženosťou verzie 2.5 a obsahuje najpoužívanejšie položky dostupné aj z iných častí menu.

## Triedenie indexovaním

Ak sme doposiaľ potrebovali utriediť súbor, použili sme funkciu *Database* → *Sort*, ktorou sme vytvorili nový súbor obsahujúci recordy v predpísanom poradí.

Takéto triedenie má tri nevýhody:

- pre rozsiahle databázy môže trvať veľmi dlho, pretože táto funkcia jednak vyhľadáva správny record, jednak ho kopíruje do nového súboru,
- vytvára sa nový súbor (často rovnako veľký), čo je opäť pre prácu s obrovskými databázami veľmi riskantné (ľahko sa môže stať, že pre nový súbor nebude na disku dostatok miesta),
- výsledkom *sortovania* je databáza obsahujúca rovnaké recordy ako pôvodná, len v inom poradí - údaje sú zbytočne zdvojené.

Z týchto dôvodov sa v praxi *sortovanie* bežne nepoužíva, nahrádza ho ***indexovanie***.

Pri *indexovaní* sa nevytvára súbor obsahujúci údaje v inom poradí, vytvára sa len súbor tzv. ukazovateľov - indexov, ktoré (zjednodušene) určujú poradie, v akom sa majú recordy vypisovať.

Napr.: nech sú v databáze *MENA* uložené recordy:

1. Roman
2. Andrej
3. Michal
4. Mária
5. Zuzana
6. František

Pri *sortovaní* získame nový súbor s údajmi:

1. Andrej
2. František
3. Mária
4. Michal
5. Roman
6. Zuzana

Pri *indexovaní* len “poradie”, v akom sa majú recordy vypísať:

- 2.
- 6.
- 4.
- 3.
- 1.
- 5.

Pri zobrazení databázy sa údaje vypisujú v tomto poradí. Ak máme porovnať oba druhy triedenia vzhľadom na miesto, ktoré zaberajú na disku, vyjde jednoznačne víťazne *indexovanie* (čím viac položiek obsahuje record, tým viac sa ušetrí). To isté platí aj pre rýchlosť.

Okrem triedenia je možné pri *indexovaní* používať aj podmienky. Ak by sme si dali vypísať ženské mená (podľa abecedy), indexový súbor by obsahoval položky:

- 4.
- 5.

*Samotné vytváranie indexových súborov prebieha vo FoxPro trochu inak, ako bolo popísané, no princíp zostáva.*

## Vytvorenie indexu

Usporiadajte (indexovaním) databázu ZAMESTNANCI podľa abecedy.

Ak chceme vytvoriť index, použijeme *File* → *New* → *(.) Index*.  
Získame okno, v ktorom nastavíme jeho vlastnosti.

Vľavo je zoznam všetkých polí, podľa ktorých možno indexovať, vpravo sú už vytvorené indexové súbory (žiadne). Ak chceme vytvoriť index, zvolíme pole, podľa ktorého sa bude vytvárať, a preniesieme ho do pravého okna.

Database Fields:		Options	Index On:
MENO	C	<input checked="" type="checkbox"/> Ascending	
DETI	N	<input type="checkbox"/> Descending	
PLAT	N	<input type="checkbox"/> Unique	
UEK	N	<For...>	
DAT_PRIJAT	D	Tag Name	
STAU	C	< Move → >	
OKULIARE	L	<Remove/Edit >	
AUTO	L		
Index Expression <Expr...>		Output	
		<input type="checkbox"/> IDX <input checked="" type="checkbox"/> Compact	
		<input checked="" type="checkbox"/> CDX <input type="checkbox"/> Structural	
		<Save As...>	
		C:\FOXPRO20\ZZAMESTN.C	
			<< OK >>
			< Cancel >

Pri tejto akcii sa meno poľa objavilo aj v položke *Tag name*. Táto položka udáva meno, pod akým bude index vytvorený - volaný (možno ho meniť, ale je to zbytočné).

V skupine *Options* sa nachádzajú ďalšie položky:

*Ascending, Descending* - usporiadanie vzostupne alebo zostupne.

*For* - pre zadanie podmienky.

*Unique* - pokiaľ je zapnutá táto vlastnosť indexového súboru, tak v prípade, že existuje viac recordov s rovnakým obsahom položky, podľa ktorej sa indexuje, zobrazí sa len jedna (napr.: ak máte v zozname štyroch Jožkov Mrkvičkov, určíte indexovanie podľa mena a zapnete *Unique*, zobrazí sa len jeden z nich).

V okne *Index Expression* možno nastaviť výraz, podľa ktorého sa bude indexovať (s použitím znakových funkcií by sme mohli nastaviť, aby sa neindexovalo podľa abecedy, ale podľa druhého písmena alebo podľa dĺžky mena atď.). Pre bežné potreby vystačíme aj bez tejto voľby.

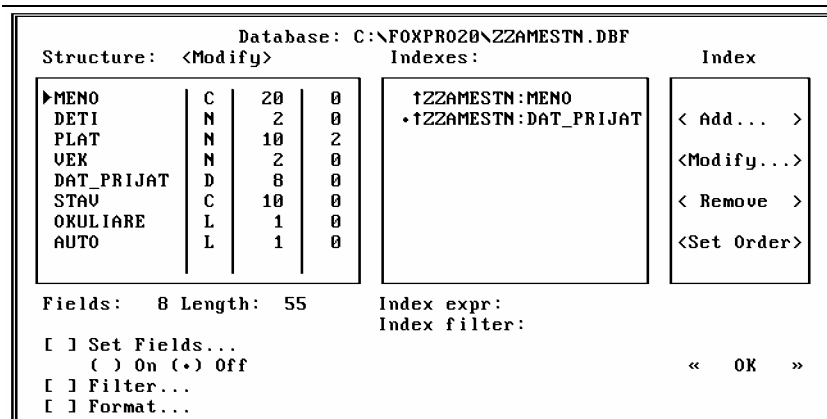
Okno *Output* hovorí o spôsobe, akým sa budú ukladať indexy na disk.

*IDX* - pre každý index vytvorí nový súbor,

*CDX* - indexy patriace k jednej databáze sa ukladajú do jedného súboru.

Pre zadanú úlohu stačí presunúť pole s menom do pravého okna a odsúhlasiť. Všetko ostatné sa vykoná samo.

Pre databázu možno mať vytvorených aj viac indexov. Ten, podľa ktorého je aktuálne usporiadaná sa nazýva *Master index*. Najčastejšie sa indexy prepínajú v *Database* → *Setup*, kde je *Master index* označený bodkou. Ak ho chceme prestaviť, nastavíme na nového "mastera" kurzor a zvolíme *Set order*.



Pri ďalšom otvorení databázy sa spolu s ňou otvárajú aj indexové súbory.

Pri vytváraní nových indexov je najrozumnejšie všetky existujúce preniesť z pravého okna do ľavého a začať odznova. Zabráňte tým častým zmätkom.

Vytvorte index podľa počtu detí a vyberte tých zamestnancov, ktorí sa narodili po roku 1961.

## Query

*Query* alebo po česky dotazy (slovenský ekvivalent ťažko nájsť) slúži na výber a spájanie databáz.

Majme niekoľko databáz o svojich známych. V jednej budú ich osobné údaje, v druhej pracovné, v tretej údaje o príbuzných atď.

V každej databáze by mal byť identifikátor, podľa ktorého ich spoznáme (evidenčné číslo alebo jednoducho meno a priezvisko).

Pomocou *Query* dokážeme z týchto databáz vytvoriť jednu. (Tento príklad je len na pochopenie, praktické využitie v ňom zatiaľ nehľadajte.)

Vytvorte databázy *OSOBNÉ* a *PRACOVNE*, v ktorých budete evidovať údaje o svojich známych.

OSOBNÉ	PRACOVNE
Meno	Meno
Adresa	Zamestnávateľ
Vek	Plat

Tieto dve databázy spojíme do jednej.

Funkcia *File* → *New* → (.) *Query* nám poskytne okno, v ktorom môžeme databázy spojiť.

RQBE - UNTITLED

<b>Databases</b> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">PRACOUNE</div> <p style="text-align: center;">&lt; Add &gt; &lt; Clear &gt;</p>	<b>Output Fields</b> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">       ↓ MENO        ↓ ZAMESTNAVÅ        ↓ PLAT        ↓        ↓        ↓     </div> <p style="text-align: center;">[X] Select Fields... [ ] Order By... [ ] Group By... [ ] Having...</p>	<b>Output To</b> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px; text-align: center;">Browse</div> <p style="text-align: center;">[ ] Options...</p> <p style="text-align: center;">&lt; See SQL &gt; &lt;&lt; Do Query &gt;&gt;</p>		
<b>Field Name</b>	<b>NOT</b>	<b>Example</b>	<b>Up=Lo</b>	<b>Select Criteria</b>
				<Insert> <Remove> < Or >

**Postup:**

1. Do okna *Databases* potrebujeme vložiť zoznam databáz, s ktorými budeme pracovať. Voľbou *Add* databázu pridáme, *Clear* ju zo zoznamu vymažeme.

2. Pri pridávaní druhej databázy sa proces preruší RQBE dialógom, v ktorom udávame spojenie databáz (ktoré pole považovať za spoločné).

RQBE Join Condition:

OSOBNÉ.MENO	[ ] NOT	Like	PRACOUNE.MENO
[ ] Ignore Upper/Lower Case			<< OK >>
			< Cancel >

V prvom okne sú položky prvej databázy - vyberieme *meno*. Druhé okno obsahuje "vzťah" (*Like* - ako (znamená, že jedno zo slov je časťou druhého), *Exactly Like* - presne ako (slová sú absolútne rovnaké), *More Than* - viac ako, *Less Than* - menej ako), podľa ktorého sa spojenie vytvorí. My chceme vytvoriť databázu, v ktorej spojíme recordy s rovnakým menom, teda ponecháme nastavenie *Like* (prípadne prestavíme na *Exactly Like*). V poslednom okne vyberieme položku *meno* z druhej databázy a odsúhlasíme. (Vzťah sa zobrazí aj v dolnej časti hlavného okna.)

Nastavili sme spojenie databáz podľa mena, t. j. ak sa nájde v oboch databázach rovnaké meno, výsledkom bude riadok získaný spojením údajov z jednej aj druhej databázy.

3. Zostáva nám ešte nastaviť zobrazenie polí, ktoré majú byť v novej databáze - *Select Fields*.

*Polia budú v novej databáze usporiadané v poradí, v akom ich zadáme.*

4. Zvolíme *Do Query* a databáza je na svete.

QUERY				
Meno	Adresa	Vek	Zamestnava	Plat
Miso Zlatny	Jaskynna 31	22	Drevian, s.r.o.	4578.90
Jaroslav Kaciak	Rybniouce 67	43	Rybarsky zväz	12600.00
Jan Mafera	Silouce 333	58	Obecný úrad	3123.80

Spojenie sa vykonalo len do okna *Browse* (podľa nastavenia v *Output to*). Ak chceme vytvoriť aj fyzický súbor, prestavíme túto voľbu na *Table/DBF*.

V tomto príklade sme spojením databáz získali len spojenie databáz. *Query* sa ale väčšinou používajú najmä kvôli šetreniu miestom. Najčastejším prípadom je použitie v knižniciach.

V knižnici (alebo ľubovoľnej inej požičovni) je potrebné evidovať najmä výpožičky.

Malo by sa vedieť kto, kedy a čo si vypožičal, t.j.: *číslo čitateľského preukazu, meno, adresa, dátum výpožičky, evidenčné číslo knihy, názov, autor, cena, vydanie, počet strán, kľúčové slová*.

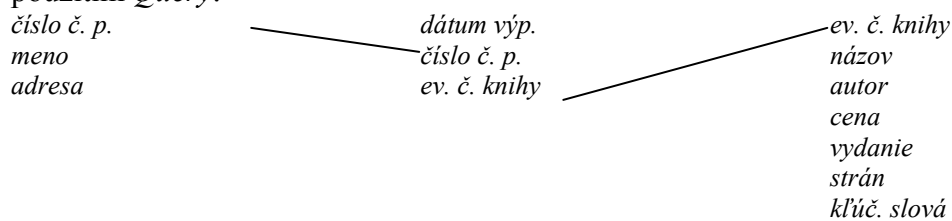
Databáza s takýmto obsahom zaberie “dost” miesta. A ak si jeden čitateľ vypožičia 10 kníh, alebo ak sa kniha vypožičia viac ráz (čo je úplne normálne), údaje v databáze sa začínajú opakovať a jej objem katastroficky narastá.

A vtedy nastupuje možnosť rozdeliť takúto obrovskú databázu na niekoľko menších, v ktorých sa údaje opakovať nebudú.

Systém by mohol pozostávať z troch databáz:

- čitateľ (*číslo čitateľského preukazu, meno, adresa*),
- kniha (*evidenčné číslo knihy, názov, autor, cena, vydanie, počet strán, kľúčové slová*),
- výpožičky.

Vo výpožičkách by stačilo evidovať dátum a jednoznačné identifikátory knihy a čitateľa (*číslo čitateľského preukazu, evidenčné číslo knihy*). V tomto prípade by nedochádzalo k extrémnej redundancii a kompletne údaje o výpožičke by bolo možné získať jednoduchým použitím *Query*.



Vytvorte databázy a na základe ich prepojenia cez číslo čitateľského preukazu a evidenčné číslo knihy zobrazte kompletne údaje o výpožičkách.

Výsledný dotaz môže vyzeráť takto:

Field Name	NOT	Example	Up=Lo	Select Criteria
‡ CITATEL.CCP	<input type="checkbox"/> Like	UYPOZICK.CCP	<input type="checkbox"/>	<Insert>
‡ KNIHA.ECK	<input type="checkbox"/> Like	UYPOZICK.ECK	<input type="checkbox"/>	<Remove>
				< Or >

Pri práci s *Query* môžu nastať problémy s otváraním súborov (pracuje vo viacerých oblastiach). Ak sa vám príslušný súbor odmieta otvoriť, použite príkaz **close all**, čím zatvoríte všetky databázy vo všetkých oblastiach.

## Súčasné otvorenie (zobrazenie) viacerých databáz

Zvykli sme si, že pri otváraní novej databázy sa stará vždy automaticky zatvorí a nech sme sa čo ako snažili, nepodarilo sa nám zobraziť naraz dve browse okná pre dve rozličné databázy. FoxPro túto operáciu cez menu ani nedokáže (veď načo je bežnému užívateľovi otvorenie niekoľkých databáz naraz - aj tak môže čítať v jednom okamihu len jednu).

V prípade, že potrebujeme napr. porovnávať obsahy dvoch databáz, môžeme použiť *Query* alebo postupovať nasledovne:

1. Zatvoríme databázu príkazom *use* (len preistotu - nie je podmienkou),
2. Otvoríme prvú databázu (a môžeme ju zobraziť),
3. Príkazom *select 2* prepne FoxPro do druhej pracovnej oblasti,
4. V novej pracovnej oblasti otvoríme druhú databázu (a zobrazíme ju),
5. Prepínaním sa medzi oknami môžeme súčasne pracovať s oboma tabuľkami.

System File Edit Database Record Program Window Browse

OSOBNÉ		
Meno	Adresa	Vek
Miso Zlatny	Jaskynna 31	22
Jaroslav Kaciak	Rybnikovce 67	43
Jan Mafera	Silovce 333	58

PRACOVNE		
Meno	Zamestnava	Plat
•Miso Zlatny	Drevian, s.r.o.	4578.90
■ Jaroslav Kaciak	Rybarsky zväz	12600.00
■ Jan Mafera	Obecný úrad	3123.80

Príkaz *select* slúži na prepínanie medzi (štandardne) 25 oblast'ami, pričom v každej z nich môže byť otvorená iná databáza (v dvoch oblastiach sa tá istá otvoriť nedá). Pre prácu s konkrétnou databázou sa najprv musíme do príslušnej oblasti prepnúť (kliknutím na okno s otvorenou databázou alebo použitím *select*).

Ak chceme databázy vo všetkých oblastiach zatvoriť, použijeme príkaz *close all*, pre zavretie databázy v aktívnej oblasti stačí *use*.

Štandardne (po spustení FoxPro) je nastavená pracovná oblasť 1.

## OPAKOVANIE V

### Indexovanie

- triedenie pomocou indexového súboru. Oproti klasickému *sortovaniu* je rýchlejšie a zaberá menej miesta. Indexový súbor pozostáva z ukazovateľov, ktoré systému určujú poradie, v akom údaje vypísať.

### Query

- slúži na spájanie databáz a vytváranie "zostáv" z viacerých súborov. Šetrí miesto a zvyšuje prehľadnosť.

### Súčasné otvorenie viacerých databáz

- možno ho realizovať len vo viacerých oblastiach. FoxPro dáva k dispozícii 25 oblastí, medzi ktorými sa prepína príkazom *select*. Štandardne je nastavená oblasť s číslom 1. Na uzavretie všetkých otvorených databáz (aj okien) sa používa príkaz *close all*.



**OBSAH**

<b>DATABÁZOVÉ SYSTÉMY .....</b>	<b>1</b>
Databázový systém FoxPro .....	3
Vytvorenie databázy .....	4
Práca s existujúcou databázou .....	6
Otvorenie .....	6
Režimy zobrazenia .....	6
Názvoslovie .....	7
Zmena štruktúry databázy .....	8
Výber polí (projekcia) .....	9
Výber riadkov (filtrácia) .....	11
Viacnásobné podmienky .....	13
Podmienky pre <i>Character</i> .....	14
OPAKOVANIE I .....	15
<b>OKNÁ .....</b>	<b>17</b>
Command okno .....	17
Typ Date .....	17
Typ Logical .....	18
Triedenie (sortovanie) .....	19
Skupina Scope, For, While .....	22
For .....	22
While .....	22
Scope .....	22
Matematické a štatistické operácie .....	23
Priemer .....	23
Suma - súčet .....	25
Počet recordov spĺňajúcich podmienku .....	25
Calculate .....	26
Typ <i>Memo</i> .....	27
OPAKOVANIE II .....	29
<b>TLAČOVÉ ZOSTAVY - REPORTY .....</b>	<b>31</b>
Vytvorenie reportu .....	31
Rozšírenie o vlastnosti objektov .....	33
Vlastnosti textu .....	33
Vlastnosti polí .....	34
Vlastnosti rámov .....	35
Tlač reportu .....	36

---

Uloženie reportu .....	36
Otvorenie existujúceho reportu .....	37
Hromadná korešpondencia .....	38
OPAKOVANIE III.....	40
Mazanie recordov .....	42
Rušenie logického vymazania .....	43
Závislosti .....	43
Vyhľadávanie .....	46
Spájanie databáz.....	47
Kopírovanie .....	47
Total .....	48
Operácie s tabuľkou .....	49
OPAKOVANIE IV .....	50
<b>MENU FOXPRO.....</b>	<b>51</b>
Podmenu System.....	51
Podmenu File.....	53
Podmenu Edit .....	54
Podmenu Database .....	54
Podmenu Record .....	55
Podmenu Program .....	55
Podmenu Window .....	55
Podmenu Browse.....	56
Podmenu Report.....	56
Podmenu Run .....	57
<b>TRIEDENIE INDEXOVANÍM.....</b>	<b>58</b>
Vytvorenie indexu .....	59
Query .....	60
Súčasné otvorenie (zobrazenie) viacerých databáz .....	63
OPAKOVANIE V .....	64