

Katona Viktor

MÁSODIK PC-KÖNYVEM

ADATBÁZIS-KEZELÉS KEZDŐKNEK



Második PC-könyvem

Számítástechnikai ABC

Katona Viktor

Második PC-könyvem

Adatbázis-kezelés kezdőknek

Sors Bona

1995

A Katolikus Iskolák Főhatósága
ezt a tankönyvsorozatot saját iskolái számára
elfogadta

Lektorálta
Adamát Lajos Sch. P.

Kiadó: Lakatos Éva — 6001 Kecskemét, Pf. 496
Tipográfia: Adamát Lajos
Fedélterv: Ulrich Gábor
Fotó: Fellegi Jenő
ISBN 963 85400 0 1

Készítette: Házinyomda Kft. — Kecskemét
Felelős vezető: Fehér Dezső

Előszó

Az Első PC-könyvem-ben a számítógép használatának alapjaival ismerkedhetett meg az olvasó. Ebben a könyvben arról lesz szó, hogy nagy mennyiségű adatból hogyan lehet a számítógép segítségével az éppen szükséges — az adatokból kibányászható — ismeretekre szert tenni.

Korunk az információrobbanás kora. Ebben az adatsdömpingben eligazodásra csak akkor van esélyünk, ha az adatainkból megfelelő módszerekkel ki tudjuk választani a számunkra fontos ismereteket. Erre a célra már több programot, ún. adatbázis-kezelőt fejlesztettek ki. Ebben a könyvben a Ashton-Tate cég dBASE III+ nevű szoftvertermékén keresztül sajátíthatjuk el az adatbázis-kezelés rejtjelmeit. Mivel ez az adatbázis-kezelő terjedt el a legjobban, ezért „szabvánnyá” vált. Ez azzal az előnnyel jár, hogy az olvasó például a Windows alatt futó Microsoft FoxPro adatbázis-kezelőn is szinte változatlan alakban használhatja az itt tárgyalt parancsokat.

A könyvnek nem célja a dBASE III+ teljes körű ismertetése, csak arra vállalkozik, hogy az alapokat a megszokott alapossággal elsajátíttassa. Így például szándékosan kimaradt minden olyan utasítás, mely a programírással kapcsolatos. Hasonlással élve: csak csónakázni (mégcsak nem is hajózni) tanulunk az adattengeren, a csónaképítést (pláne a hajóépítést) rábízzuk szakemberekre.

A könyvben talált példákhoz, feladatokhoz szükséges adatbázisfájlok megtalálhatók a könyv mellékleteként kapott lemezen. Ezek tartalma megtalálható a könyv szövegében is, csak a könnyebbség kedvéért készen adjuk. Mivel gyakorló feladatokból sosincs elég, ezért a könyv feladatain kívül további feladatok is találhatóak a lemezmellékleten a gyakorolni vágyók számára.

Reméljük, hogy a könyvsorozatnak ez a része is elnyeri a felhasználók tetszését, elsajátítják belőle az adatbázis-kezelés alapvető elemeit, és ennek előnyeit napi munkájuk során is élvezni fogják.

Két olyan dolog van, amely nem fogy el, ha másokkal megosztják: a szeretet és az ismeret. Reméljük mindkettőben növekedni fognak az olvasók.

Kecskemét, 1995. április 16.

Adamát Lajos

Tartalomjegyzék

1. Adatnyilvántartás papíron.....	13
2. Az adatbázis-kezelés alapfogalmai.....	15
a) Az egyed	15
b) A rekord.....	15
c) A mező.....	15
d) Az adatbázisfájl.....	16
e) Az adatbázis	16
f) Az adatbázis-kezelő program	16
3. A dBASE fogalma	17
4. Az adatbázisfájlok azonosítása	18
5. A mező neve	18
6. Adattípusok	19
a) Szám.....	19
b) Dátum.....	19
c) Karakter sorozat.....	19
d) Logikai adat.....	20
7. A mező típusa	20
a) Numerikus mező	20
b) Dátum jellegű mező	20
c) Karakteres mező.....	21
d) Logikai mező	21
8. A mező hossza	22
9. A rekordok hossza	23
10. Adatbázisfájl tervezése	23
11. Az adatbázisfájl szerkezete	24
12. Mely fájlok szükségesek a dBASE működéséhez?	25
13. Hová helyezzük a készülő fájlokat?.....	26
14. Hogyan indítjuk el a dBASE programot?	27
15. Mi látható a képernyőn a dBASE indulásakor?	27
a) A parancssor	28
b) A státusz sor.....	28
1) A parancs-cella	28
2) A meghajtó-cella	28

3) Az adatbázisfájl-cella.....	28
4) A rekord-cella	28
5) Az üzemmód-cella.....	28
6) A lámpa-cella	29
16. Hogyan adunk a dBASE-nek parancsot?.....	29
17. Hogyan töröljük le a képernyőt?	29
18. Hogyan lépünk ki a dBASE programból?	30
19. Adatbázisfájl létrehozása.....	30
a) Az adatbázisfájl létrehozásának kezdeményezése	30
b) Az adatbázisfájl szerkezetének begépelése.....	31
c) Az adatbázisfájl feltöltése adatokkal.....	32
20. A gépelési hibák kijavítása	33
21. Az adatbázisfájlok számbavétele.....	37
22. Adatbázisfájl megnyitása	38
23. Adatbázisfájl lezárása	38
24. Az adatbázisfájl szerkezetének kiíratása.....	39
25. Az aktuális rekord fogalma	40
26. Aktuális rekord kijelölése	41
a) Kijelölés a GO paranccsal.....	41
b) Kijelölés a SKIP paranccsal.....	42
27. Rekordok hozzáfűzése az adatbázisfájl végéhez	43
28. Rekord betoldása az adatbázisfájlba	44
29. Adatbázisfájl tartalmának kiíratása	45
30. Adatbázisfájl tartalmának módosítása	46
a) Módosítás az APPEND paranccsal.....	47
b) Módosítás az EDIT paranccsal.....	47
c) Módosítás a CHANGE paranccsal	47
d) Módosítás a BROWSE paranccsal	48
31. Rekordok törlése az adatbázisfájlból	49
a) A törlés elvi kérdései	49
b) A DELETE parancs használata.....	49
c) A RECALL parancs használata.....	50
d) A PACK parancs használata.....	50
32. Az adatbázisfájl összes rekordjának törlése.....	51
33. A konstans fogalma	52
34. Konstanstípusok	52
35. A konstansok megadása.....	53

a) Numerikus konstans megadása.....	53
b) Karakteres konstans megadása.....	53
c) Logikai konstans megadása.....	53
36. A művelet fogalma.....	54
37. A művelet eredményének kiírása.....	55
38. Számítási műveletek.....	56
39. Naptári műveletek.....	58
40. Összefűző műveletek.....	59
a) Összefűzés szóközmeghagyással.....	60
b) Összefűzés szóközát helyezéssel.....	60
41. Logikai műveletek.....	61
a) Az ÉS művelet.....	61
b) A VAGY művelet.....	61
c) A TAGADÁS művelet.....	62
42. Összehasonlító műveletek.....	63
43. Karakterkereső művelet.....	65
44. A kifejezés fogalma.....	66
45. A kifejezés értékének kialakulása.....	67
46. A kifejezés értékének kiírása.....	69
47. Numerikus kifejezések.....	70
48. Dátum jellegű kifejezések.....	71
49. Karakteres kifejezések.....	72
50. Logikai kifejezések.....	73
51. A hatáskör fogalma.....	77
52. A hatáskör megadása.....	77
a) RECORD.....	77
b) NEXT.....	78
c) ALL.....	78
d) REST.....	79
53. A hatáskör alapértelmezése.....	80
54. A feltétel fogalma.....	81
55. Hatáskör és feltétel együttes alkalmazása.....	82
56. A kifejezéslista fogalma.....	83
57. Kifejezéslista alkalmazása.....	84
58. A rekordsorszám letiltása.....	87
59. Kiírás a nyomtatóra.....	88

60. A mezőlista fogalma	88
61. Mezőlista alkalmazása	89
62. Összefoglalás	90
63. A parancs formája	90
64. A mezők tartalmának automatikus módosítása	93
65. Összegzés, átlagolás, számlálás	95
a) Összegzés és átlagolás	95
b) Számlálás	96
66. Adatbázisfájl szerkezetének megváltoztatása	98
67. Fájl másolása	100
68. Fájl átnevezése	100
69. Fájl törlése	101
70. Adatbázisfájl másolása	101
71. Keresés az adatbázisfájlban	103
72. A rekordok kezelési sorrendje	105
73. Rendezés	106
74. Az indexelés fogalma	111
75. Az indexfájlok azonosítása	113
76. Indexfájl létrehozása	114
77. Indexfájlok megnyitása és lezárása	116
78. Főindexfájl kijelölése	117
79. A főindexfájl módosító szerepe	118
a) A GO és a SKIP parancs módosulása	119
b) A hatáskör módosulása	120
c) A BROWSE parancs módosulása	121
80. Újrindexelés	121
81. A rendezés és az indexelés összehasonlítása	122
82. Gyorskeresés az adatbázisfájlban	124
83. A munkaterület fogalma	125
84. A munkaterület azonosítója	125
85. Az aktuális munkaterület fogalma	126
86. Munkaterület váltása	126
87. Az aktív adatbázisfájl fogalma	127
88. Egyszerű és összetett mezőnév	128
a) Egyszerű mezőnév	128
b) Összetett mezőnév	128

89. Adatbázisfájlok kapcsolata	130
90. Két adatbázisfájl összekapcsolása.....	132
91. Adatbázisfájlok együttes tervezése	134
92. Több adatbázisfájl összekapcsolása	134
93. Az összekapcsoltság megszüntetése	136
94. A függvény fogalma	137
95. A függvény típusa	138
96. Függvény a kifejezésben	138
97. A függvény formája	139
98. A függvény értékének kiírása	140
99. Matematikai függvények	141
a) Az abszolútérték-függvény	141
b) Az egészrész-függvény	141
c) A négyzetgyök-függvény	142
d) A kerekítő függvény	142
e) A maximum-függvény	142
f) A minimum-függvény	143
100. A matematikai függvények alkalmazása.....	143
101. Karakterkezelő függvények.....	145
a) Balrész-függvény.....	145
b) Jobbrész-függvény	145
c) Középrész-függvény	145
d) Sokszorozó függvény.....	146
e) Kisbetű-függvény	146
f) Nagybetű-függvény	146
g) Bal oldali szóközöket levágó függvény	147
h) Jobb oldali szóközöket levágó függvény.....	147
i) Hossz-függvény.....	147
j) Belső kód-függvény.....	148
k) Karaktert adó függvény	148
l) Karakterkereső függvény	148
m) Szóközöket adó függvény	149
n) Betűvizsgáló függvény	149
102. A karakterkezelő függvények alkalmazása.....	149
103. Dátumkezelő függvények.....	152
a) A mai dátumot megadó függvény	152
b) Évszám-függvény	153

c) Hónap-függvény	153
d) Nap-függvények	153
104. A dátumkezelő függvények alkalmazása	154
105. Választó függvény	155
106. Típusváltató függvények	156
a) A STR függvény	157
b) A VAL függvény	157
c) A CTOD függvény	158
d) A DTOC függvény	158
107. A típusváltató függvények alkalmazása	159
108. DOS-parancs végrehajtása	160
109. A dBASE működésének szabályozása	161
a) A képernyő színeinek beállítása	161
b) A dátumok formájának beállítása.....	162
c) A kiírt tizedesjegyek számának beállítása.....	163
d) A funkcióbillentyűk beállítása	165
e) A fájlkeresés szabályozása	167
f) Szabályozás kapcsolókkal.....	168
1) A BELL kapcsoló	169
2) A CENTURY kapcsoló.....	169
3) A CONFIRM kapcsoló	170
4) Az EXACT kapcsoló	170
5) A FIXED kapcsoló	171
6) A HEADINGS kapcsoló	171
7) A HELP kapcsoló	171
8) A HISTORY kapcsoló	172
9) A MENUS kapcsoló	172
10) A SAFETY kapcsoló	172
11) A STATUS kapcsoló	173
g) A tárolt parancsok számának beállítása	173
110. A CONFIG.DB fájl szerepe.....	174

1. fejezet

Adatnyilvántartás papíron

A számítógépek megjelenése előtt az adatokat papíron tartották nyilván.

Példa

Papíron fogunk adatnyilvántartást készíteni az osztály tanulóiról. Mindenkiről az alábbi adatokat jegyezzük fel:

- neve
- születési helye
- születési dátuma
- testvéreinek száma
- tanulmányi átlaga
- kollégiumban lakik-e?

A nyilvántartáshoz egy ilyesféle táblázatot fogunk használni:

A tanuló neve	Születési helye	Születési dátuma	Testvéreinek száma	Tanulmányi átlaga	Kollégiumban lakik-e?

Felmerül a kérdés: mely rovat mekkora helyet igényel. Ez attól függ, hogy az egyes rovatok tartalmát hány karakterrel tudjuk megadni.

— A tanuló neve megadható 25 karakterrel, mert valószínűleg nincs olyan tanuló, akinek a neve 25-nél több karakterből állna. Ha mégis lenne ilyen, akkor nevének a 25. karakter utáni részét elhagyjuk.

— A születési hely megadható 20 karakterrel. 20 karakter még az olyan hosszú településnév megadásához is elegendő, mint pl. Kiskunfélegyháza.

— A születési dátum megadásához 8 karakter szükséges: 4 számjegy az évnek, 2 a hónapnak és 2 a napnak. Az évet a hónaptól, valamint a hónapot a naptól ponttal fogjuk elválasztani. Ha valaki például 1965. február 21-én született, akkor azt így adjuk meg: 1965.02.21

— A testvérek száma esetleg kétjegyű szám is lehet, ezért a megadásához 2 karakter szükséges.

— A tanulmányi átlag megadásához 4 karakter szükséges: 1 karakter az egészjegynek, 1 a tizedespontnak és 2 a tizedesjegyeknek.

— Annak megadásához, hogy a tanuló kollégiumban lakik-e, elegendő 1 karakter. Megállapodunk ugyanis abban, hogy ha kollégiumban lakik, akkor Y-t, ha pedig nem, akkor N-t írunk ebbe a rovatba (Y=yes, N=no).

Ha az elmondottakat figyelembe vesszük, akkor a táblázat rovatainak szélességét így célszerű módosítani:

	A tanuló neve	Születési helye	Születési dátuma	T. sz.	Tan. átl.	K.
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						
15.						
16.						
17.						

A rovatok feliratát esetenként helyszűke miatt rövidíteni kell.

Például a testvérek számát T.sz-szel rövidítettük.

Feladat

Töltsd ki az előző táblázatot valós adatokkal! (A tanulók sorban menjenek ki a táblához és írják fel adataikat, a többiek pedig mindezt jegyezzék be a táblázatukba!)

Megjegyzés

Aki ezt a könyvet önállóan tanulmányozza, a fenti táblázatot elképzelt adatokkal töltsd ki!

2. fejezet

Az adatbázis-kezelés alapfogalmai

Azt a nyilvántartást, amelyet az imént papíron készítettünk, rövidesen számítógépbe fogjuk táplálni. Ismerkedjünk meg tehát az ehhez szükséges alapfogalmakkal!

a) Az egyed

A nyilvántartás szereplői mindig jól meghatározott egyedek (például személyek vagy tárgyak). Példánkban az egyedek az osztály tanulói voltak.

A nyilvántartás célja határozza meg, hogy az egyedekről milyen adatokat tartunk nyilván. Egy cipőkereskedő például biztos, hogy nyilvántartaná a tanulók lábméretét, és nemigen érdekelné őt a tanulmányi átlaguk.

b) A rekord

Az ugyanazon egyedről nyilvántartott adatok összességét rekordnak nevezzük. A rekord valójában a nyilvántartás egy sorát jelenti.

A rekordokat sorszámmal látjuk el: az első sor az 1. rekord, a második sor a 2. rekord, és így tovább.

A tanulókról készített nyilvántartásban rekordot alkotnak például az alábbi adatok: a harmadik sorban szereplő tanuló neve, születési helye, születési dátuma, testvéreinek száma, tanulmányi átlaga, és hogy kollégiumban lakik-e?

c) A mező

A nyilvántartás minden oszlopa egy-egy mező.

A tanulókról készített nyilvántartásban hat mező található. Ezek közül például a „Születési dátum” föliratú mező a tanulók születési dátumát tartalmazza.

d) Az adatbázisfájl

A számítógépes adatnyilvántartást mágneslemezen, fájlban fogjuk elhelyezni.

A rekordokra és mezőkre osztott fájl adatbázisfájlnak nevezzük.

Az adatbázisfájl így szemléltethető:

	1. mező	2. mező	3. mező	4. mező	...	m. mező
1. rekord						
2. rekord						
3. rekord						
4. rekord						
...						
n. rekord						

e) Az adatbázis

Több összetartozó (vagy egyetlen) adatbázisfájl alkot egy adatbázist. Az adatbázis-hoz soroljuk azokat az egyéb fájlkat is, amelyek szorosan kapcsolódnak az adatbázis-fájlokhoz.

f) Az adatbázis-kezelő program

Az adatbázis kezeléséhez (használatához) programra van szükség. Ilyen programok készen kaphatók.

Minden adatbázis-kezelő program képes az alábbi teendők ellátására:

1. Az adatbázisfájl létrehozása

A létrehozás azt jelenti, hogy az eddig papíron vezetett nyilvántartást betápláljuk a számítógépbe. Ettől kezdve a papíron vezetett nyilvántartás feleslegessé válik.

2. Az adatbázisfájl kiegészítése újabb rekordokkal

Kiegészítésre akkor van szükség, amikor újabb egyedekkel akarjuk bővíteni a nyilvántartást. (A tanulókról készített nyilvántartásban kiegészítésre akkor kerül sor, amikor új tanuló érkezik az osztályba, vagy amikor kiderül, hogy valakit előzőleg véletlenül kifelejtettünk.)

3. A rekordok tartalmának módosítása

Módosításra akkor kerül sor, amikor valamely adat megváltozik, vagy amikor kiderül róla, hogy tévesen van begépelve. (A tanulókról készített nyilvántartásban a „Tanulmányi átlag” feliratú mező tartalmát minden bizonyítványosztáskor módosítani kell.)

4. Rekordok törlése

Törlésre akkor van szükség, amikor kiderül, hogy valamely egyedet a továbbiakban nem kell már nyilvántartani. (A tanulókról készített nyilvántartásban akkor kell rekordot törölni, amikor egy tanuló évisméltésre kényszerül, vagy iskolát változtat.)

5. Válogatás a rekordok között

A rekordok között néha válogatni kell. A válogatás alapját egy vagy több mező tartalma képezi. (A tanulókról készített nyilvántartásban a „Testvérek száma” feliratú mező alkalmat nyújt arra, hogy kiválogassuk például azokat a tanulókat, akiknek 2-nél több testvérük van.)

6. Műveletek elvégzése az adatbázisfájlban szereplő adatokkal

Az adatokkal sokféle műveletet lehet végezni. (A tanulókról készített nyilvántartásban a „Tanulmányi átlag” feliratú mező alapján ki lehet számítani az osztály tanulmányi átlagát.)

7. A rekordok sorrendjének megváltoztatása

A rekordok az adatbázisfájlban a begépelés sorrendjében helyezkednek el. Ezen a sorrenden néha változtatni akarunk. (A tanulókról készített nyilvántartásban szükség lehet a név vagy a tanulmányi átlag által meghatározott sorrendre.)

8. Több adatbázisfájl egyidejű használata

Célszerű, ha a naponta változó adatokat és a ritkán változó adatokat más-más adatbázisfájlban helyezük el. Ilyenkor szükségessé válik két vagy több adatbázisfájl egyidejű használata.

9. Az adatok kiírása

Gyakran van szükség arra, hogy az adatbázisfájlban szereplő adatokat, vagy a műveletek eredményeképpen kapott adatokat kiírassuk a képernyőre vagy a nyomtatóra.

3. fejezet A dBASE fogalma

Az adatbázis-kezelő programok közül világviszonylatban az egyik legismertebb a dBASE III PLUS. A továbbiakban ezzel, pontosabban ennek 1.0-ás verziójával foglalkozunk, és röviden csak dBASE-nek mondjuk.

4. fejezet

Az adatbázisfájlok azonosítása

Az adatbázisfájlokat (mint általában a fájlokat) azonosítóval kell ellátni.

Az adatbázisfájl nevét annak kell meghatározni, aki az adatbázisfájlt létrehozza.

Az adatbázisfájl kiterjesztése azonban nem lehet tetszőleges, mert a dBASE megköveteli, hogy DBF legyen. Ezért a .DBF a fájlazonosítóból elhagyható.

Példa

Adatbázisfájl azonosítója lehet például TANULO.DBF vagy UZLET.DBF, de nem lehet TANULO.DAT vagy UZLET.TXT.

5. fejezet

A mező neve

Annak érdekében, hogy az adatbázisfájl mezőit egymástól megkülönböztessük, a mezőket névvel kell ellátni. A mezőnév maximum 10 karakterből állhat. Az alábbi karakterek szerepelhetnek benne:

- az angol ABC nagybetűi
- számjegyek
- aláhúzásjel.

A mezőnév legelső karakterének betűnek kell lenni.

A mezők nevét az adatbázisfájl létrehozója határozza meg. A névadásnál törekedni kell arra, hogy a név utaljon a mező tartalmára.

Példa

A tanulókról készített nyilvántartás mezőit az alábbi nevekkel látjuk el:

<i>A mező jelentése</i>	<i>A mező neve</i>
A tanuló neve	NEV
Születési helye	SZUL_HELY
Születési dátuma	SZUL_DATUM
Testvéreinek száma	TESTVER
Tanulmányi átlaga	ATLAG
Kollégiumban lakik-e?	KOLLEGIUM

6. fejezet

Adattípusok

Az adatbázisfájl mezőit adatokkal fogjuk feltölteni, és az adatokkal műveleteket fogunk majd végezni.

Az adatoknak négy típusa különböztethető meg:

- szám
- dátum
- karaktersorozat
- logikai adat.

a) Szám

A szám csak számjegyeket, tizedespontot és előjelet tartalmazhat.

Példa

Számok:

-18.25
1965
+34.22

b) Dátum

A dátum 8 karakterből áll: az év 4 számjegye, a hónap 2 számjegye és a nap 2 számjegye. Az évet a hónaptól és a hónapot a naptól határolójellel választjuk el. Határolójelként általában pontot használunk.

Példa

Dátumok:

1965.02.21
2004.08.11

c) Karaktersorozat

A karaktersorozat tetszőleges karaktereket tartalmazhat.

Példa

Karaktersorozatok:

Bartók Béla
76-324616
76324616

d) Logikai adat

Az olyan adatot, amely „igen vagy nem?” jellegű kérdésre ad választ, logikai adatnak nevezzük.

Megállapodunk abban, hogy

- az igenlő választ az Y vagy a T betű jelenti
- a nemleges választ az N vagy az F betű jelenti.

Ezek szerint a logikai adat egyetlen karakterrel adható meg:

- az Y vagy a T azt jelenti, hogy ‘igen’
- az N vagy az F azt jelenti, hogy ‘nem’.

Az ‘igen’ helyett azt szoktuk mondani, hogy ‘**igaz**’.

A ‘nem’ helyett azt szoktuk mondani, hogy ‘**hamis**’.

Példa

Logikai adatok:

Y	(Jelentése ‘igen’, más szóval ‘ igaz ’)
T	(Jelentése ‘igen’, más szóval ‘ igaz ’)
N	(Jelentése ‘nem’, más szóval ‘ hamis ’)
F	(Jelentése ‘nem’, más szóval ‘ hamis ’)

7. fejezet

A mező típusa

Egy mező vagy számokat, vagy dátumokat, vagy karaktersorozatokat, vagy logikai adatokat tartalmaz.

A mező típusát az határozza meg, hogy milyen típusú adatokat tartalmaz.

a) Numerikus mező

Az olyan mezőt, amely számokat tartalmaz, numerikus mezőnek nevezzük.

A tanulókról készített nyilvántartásban két numerikus mező van:

- a testvérek száma
- a tanulmányi átlag.

b) Dátum jellegű mező

Az olyan mezőt, amely dátumokat tartalmaz, dátum jellegű mezőnek nevezzük.

A tanulókról készített nyilvántartásban dátum jellegű mező a születési dátumokat tartalmazó mező.

c) Karakteres mező

Az olyan mezőt, amely karaktersorozatokat tartalmaz, karakteres mezőnek nevezzük.

A tanulókról készített nyilvántartásban két karakteres mező van:

- a tanuló neve
- születési helye.

d) Logikai mező

Az olyan mezőt, amely logikai adatokat tartalmaz, logikai mezőnek nevezzük.

A tanulókról készített nyilvántartásban logikai mező az a mező, amely arra a kérdésre ad választ, hogy „Kollégiumban lakik-e?”

Megjegyzés

A mezőtípusokra az alábbi rövidítéseket használjuk:

Mezőtípus	Rövidítés
numerikus	N
dátum jellegű	D
karakteres	C
logikai	L

Példa

A tanulókról készített nyilvántartás mezőinek típusát az alábbi táblázat foglalja össze:

<i>A mező neve</i>	<i>A mező típusa</i>
NEV	C
SZUL_HELY	C
SZUL_DATUM	D
TESTVER	N
ATLAG	N
KOLLEGIUM	L

8. fejezet

A mező hossza

Bármilyen típusú is egy mező, a tartalma mindig karakterekből tevődik össze.

Azt a számot, amely megmutatja, hogy a mező tartalma hány karakterből tevődik össze, a mező hosszának nevezzük.

Példa

A tanulókról készített nyilvántartás mezőinek a hosszát az alábbi táblázat szemlélteti:

<i>A mező neve</i>	<i>A mező hossza</i>
NEV	25
SZUL_HELY	20
SZUL_DATUM	8
TESTVER	2
ATLAG	4
KOLLEGIUM	1

Numerikus mezőknél a mező hosszán kívül még az is fontos dolog, hogy mennyi a tizedesjegyek száma. A tanulókról készített nyilvántartásban a „Tanulmányi átlag” 2 tizedesjegyet tartalmaz, a „Testvérek száma” viszont nem tartalmaz tizedesjegyet.

Megjegyzések

1. Logikai mező hossza mindig 1.
2. Dátum jellegű mező hossza mindig 8.
3. Karakteres mező hossza nem lehet több 254-nél.
4. Numerikus mező hossza nem lehet több 19-nél.
5. Numerikus mezőnél a tizedesjegyek száma nem lehet több 15-nél.
6. Numerikus mezőnél a mező hosszának legalább kétszeresének kell lenni a tizedesjegyek számánál.
7. Dátum jellegű mező hossza ezért csak 8, és nem 10, mert a határolójel nem tárolódik az adatbázisfájlban.

9. fejezet

A rekordok hossza

Minden egyes rekord ugyanolyan hosszú. A rekordok hosszát úgy kapjuk meg, hogy a mezők hosszait összeadjuk, és még hozzáadunk 1-et. (Később magyarázzuk meg, hogy miért kell 1-et hozzáadni.)

Példa

A tanulókról készített nyilvántartásban a mezők hosszainak összege:

$$25+20+8+2+4+1 = 60,$$

tehát mindegyik rekord hossza 61.

10. fejezet

Adatbázisfájl tervezése

Az adatbázisfájl tervezése a következő lépésekből áll:

1. Eldöntjük, hogy kik vagy mik lesznek az adatnyilvántartás egyedei. (Más szóval: eldöntjük, hogy kiktől vagy mikről akarunk adatokat nyilvántartani.)
2. Eldöntjük, hogy milyen adatokat kívánunk nyilvántartani. (Más szóval: eldöntjük, hogy mely adatok alkossák a nyilvántartás mezőit.)
3. Nevet adunk a mezőknek.
4. Meghatározzuk a mezők típusát.
5. Eldöntjük, hogy a karakteres és a numerikus mezők hossza mennyi legyen.
6. Eldöntjük, hogy a numerikus mezők hány tizedesjegyet tartalmazzanak.

Feladatok

1. Tervezz adatbázisfájlt otthoni könyveid nyilvántartása céljából!

Útmutatás

A nyilvántartás egyedei most a könyveid lesznek. Célszerű a könyveidről az alábbi adatokat nyilvántartani:

- szerzője
- címe
- kiadásának éve
- vannak-e benne képek?
- ára.

2. Tervezz adatbázisfájlt az ismerőseidről annak érdekében, hogy ne felejts el megemlékezni születésnapjukról!

Útmutatás

Célszerű az ismerőseidről az alábbi adatokat nyilvántartani:

- neve
- születési dátuma
- címe (irányítószám, utca, házszám).

11. fejezet

Az adatbázisfájl szerkezete

Az adatbázisfájl szerkezete azt fejezi ki, hogy milyen nevű mezők szerepelnek benne, milyen típusúak ezek a mezők, mennyi a hosszuk, és a numerikus mezők hány tizedesjegyet tartalmaznak.

Példa

A tanulókról készítendő adatbázisfájl szerkezete:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tizedesjegyek száma</i>
NEV	C	25	-
SZUL_HELY	C	20	-
SZUL_DATUM	D	8	-
TESTVER	N	2	0
ATLAG	N	4	2
KOLLEGIUM	L	1	-

Megjegyzés

A „Tizedesjegyek száma” feliratot a továbbiakban így rövidítjük: „Tjsz.”

Feladatok

1. Add meg az otthoni könyveidről készítendő adatbázisfájl szerkezetét!
2. Add meg az ismerőseidről készítendő adatbázisfájl szerkezetét!

12. fejezet

Mely fájlok szükségesek a dBASE működéséhez?

A dBASE működéséhez az alábbi fájlok szükségesek:

DBASE.EXE
DBASE.OVL
DBASEINL.OVL
DBASE.MSG

A DBASE.EXE fájl magát a dBASE programot tartalmazza.

A DBASE.OVL és a DBASEINL.OVL fájl a dBASE program kiegészítését tartalmazza.

A DBASE.MSG fájl azokat az angol nyelvű üzeneteket tartalmazza, amiket a munka során a dBASE program a gépkezelőnek küld.

Jó, ha megvannak a következő fájlok:

HELP.DBS
CONFIG.DB

Ha rendelkezésünkre áll a HELP.DBS fájl, akkor angol nyelvű segítség kérhető a dBASE programtól.

A CONFIG.DB fájl a dBASE program működésének körülményeit szabályozza. Ezt a fájlt magunknak kell elkészítenünk!

A dBASE program működéséhez szükséges fájlok általában a C: meghajtó DBASE katalógusában vannak elhelyezve.

Feladatok

1. Annak érdekében, hogy a dBASE úgy működjön, ahogyan ebben a könyvben le van írva, hozd létre a COPY CON paranccsal C: meghajtó DBASE katalógusában a CONFIG.DB fájlt! Tartalma ez legyen:

```
BELL = OFF  
CENTURY = ON  
CONFIRM = ON  
EXACT = ON  
HELP = OFF  
MENUS = OFF  
SAFETY = OFF  
STATUS = ON  
COMMAND = SET DATE ANSI
```

Ha színes monitorod van, és azt akarod, hogy a dBASE színes változatban működjön, akkor a CONFIG.DB fájlban rendelkez a színbeállításról is, például ennek a sornak a szerepeltetésével:

```
COLOR = W/B+,GR+/R
```


Útmutatás

A feladat megoldása érdekében ezt a parancsot kell kiadnod a DOS-nak:

```
COPY CON C:\DBASE\CONFIG.DB
```

Ezt követően kell begépelned a megadott sorokat. Végezetül le kell ütnöd az F6 funkcióbillentyűt, majd az Enter-t.

2. A DIR parancs segítségével nézd meg, hogy a C: meghajtó DBASE katalógusában megvannak-e a dBASE működéséhez szükséges fájlok!

Megjegyzés

A CONFIG.DB fájl sorainak értelmét később ismerjük meg.

Külön kell szólnunk a CONFIG.SYS fájlról. Ennek a fájlnek a C: meghajtó főkatalógusában kell elhelyezkednie. Számos program, így a dBASE is bizonyos kívánalmakat állít a CONFIG.SYS tartalmát illetően. A dBASE zavartalan működése érdekében a CONFIG.SYS az alábbi sorokat kell hogy tartalmazza:

```
FILES = 20  
BUFFERS = 15
```

(Az nem baj, ha a 20 helyén nagyobb szám szerepel!)

Feladat

A DIR parancs segítségével nézd meg, hogy a C: meghajtó főkatalógusában van-e CONFIG.SYS fájl!

Ha nincs, akkor a COPY CON paranccsal hozd létre! Tartalma ez legyen:

```
FILES = 20  
BUFFERS = 15
```

Ha a CONFIG.SYS fájl már létezik, akkor a TYPE paranccsal nézd meg, hogy tartalmazza-e a fenti két sort. Ha nem, akkor valamely szövegszerkesztő programmal tedd bele ezeket a sorokat!

13. fejezet

Hová helyezzük a készülő fájlokat?

A tanulás során szükség lesz egy hajlékony gyakorlólemezre.

A gyakorlólemez főkatalógusában szerepelni kell egy KM katalógusnak, mert ebben fogjuk elhelyezni a készülő fájlokat.

A könyvben feltételezzük, hogy a gyakorlólemez az A: meghajtóban van. Ha van B: meghajtó, akkor természetesen lehet ott is.

Feladat

A gyakorlásra szánt hajlékony lemezt tedd be a meghajtóba, és hozd létre a főkatalógusában a KM katalógust!

Megoldás

Ezt a parancsot kell kiadni a DOS-nak:

```
MD A:\KM
```

Fontos megjegyzés

Ne feledkezz meg a későbbiek folyamán sem arról, hogy amikor szükséges, a gyakorlólemez benn legyen a meghajtóban!

14. fejezet

Hogyan indítjuk el a dBASE programot?

A dBASE program indítása a következő lépésekből áll:

1. Ha nem a C: az aktuális meghajtó, akkor azt kell aktuálissá tenni:

```
C:
```

2. Ha ékezetes magyar betűket is akarunk használni, akkor végre kell hajtatnunk az ehhez szükséges fájlt. Ha ez a fájl például a C: meghajtó főkatalógusában található, és az azonosítója CWIHUN1.COM, akkor az alábbi parancsot kell kiadni:

```
\CWIHUN1.COM
```

3. Ha a C: meghajtón nem a DBASE az aktuális katalógus, akkor azt kell aktuálissá tenni:

```
CD \DBASE
```

4. El kell indítani magát a dBASE programot:

```
DBASE.EXE
```

Feladat

Indítsd el a dBASE programot!

15. fejezet

Mi látható a képernyőn a dBASE indulásakor?

A dBASE indulásakor az angol nyelvű tájékoztató szövegen kívül két fontos dolog látható: a parancssor és a státuszsor.

a) A parancssor

A dBASE parancssora abból ismerhető fel, hogy a legelején egy pont látható. Ez a pont a dBASE készenléti jele. Azt jelzi, hogy parancsot lehet adni a dBASE-nek.

b) A státuszsor

A parancssor alatt látható a dBASE státuszszora. Ez induláskor valahogy így néz ki:

Command Line	<C:>			Ins	NumCaps
Parancs	Meghajtó	Adatbázisfájl	Rekord	Üzem mód	Lámpa

A státuszsor hat cellára van osztva.

1) A parancs-cella

A parancs-cella azt mutatja, hogy milyen parancs végrehajtása van folyamatban. Egyelőre az a felirat olvasható itt, hogy

Command Line

Ez arra utal, hogy még nincs végrehajtás alatt álló parancs.

2) A meghajtó-cella

A meghajtó-cella az aktuális meghajtót mutatja (most éppen C:).

3) Az adatbázisfájl-cella

Az adatbázisfájl-cella annak az adatbázisfájlnak a nevét mutatja, amellyel éppen dolgozunk. (Ez a cella most még természetesen üres.)

4) A rekord-cella

Ha majd már dolgozunk valamely adatbázisfájllal, akkor a rekord-cella két dolgot fog mutatni:

- az adatbázisfájl aktuális (kitüntetett) rekordjának sorszámát
- az adatbázisfájl összes rekordjának számát.

5) Az üzemmód-cella

Ha az üzemmód-cella bal oldali részén az „Ins” felirat látható, akkor a billentyűzet betoldó üzemmódban van, ellenkező esetben felülíró üzemmódban.

Az üzemmód-cella jobb oldali részével később foglalkozunk.

6) A lámpa-cella

A lámpa-cella a Num Lock és a Caps Lock lámpák állapotát mutatja.

Ha világít a Num Lock lámpa, akkor a cella bal oldali részében a „Num” felirat látható.

Ha világít a Caps Lock lámpa, akkor a cella jobb oldali részében a „Caps” felirat látható.

16. fejezet

Hogyan adunk a dBASE-nek parancsot?

A dBASE-nek szánt parancsot a parancssorba, a készenléti jel mellé kell begépelni.

Kis- és nagybetűket egyaránt használhatunk.

Egy parancs maximálisan 254 karakterből állhat.

Miután a parancs végét jelző Enter-t leütöttük, a parancs végrehajtásra kerül (feltéve, hogy helyesen gépeltük be). Ha a készenléti jel ismét megjelenik, akkor újabb parancsot adhatunk.

Minden parancsnak van neve. Vannak olyan parancsok, amelyeknek a neve egyetlen szó, de vannak olyan parancsok is, amelyeknek a neve több szóból áll.

Számos olyan parancs van, amelynek a működése érdekében a nevéen kívül még egyéb dolgokat is be kell gépelni.

A gépelés alatt álló parancsot a kurzormozgató billentyűk segítségével javíthatjuk.

Az előzőleg kiadott parancsok között a ↑ vagy a ↓ nyílbillentyűvel közlekedünk.

Hibás parancs kiadását követően hibajelzést kapunk, és újra megjelenik a készenléti jel, vagyis lehetővé válik a parancs újbóli begépelése. Gépelési munkát takaríthatunk meg, ha a parancs újbóli begépelése helyett a javítás mellett döntünk. Ez esetben a következőket kell tennünk:

1. Üssük le a ↑ nyílbillentyűt, erre visszakapjuk az előző (hibás) parancsot.
2. Végezzük el a szükséges javításokat.
3. A javított parancs végrehajtása érdekében üssük le az Enter-t.

17. fejezet

Hogyan töröljük le a képernyőt?

A képernyő törlése a CLEAR paranccsal történik.

Ha a CLEAR szó begépelése után leütjük az Enter-t, akkor máris törlődik a képernyő. (A készenléti jel és a státuszsor természetesen nem törlődik.)

Megjegyzés

A CLEAR parancs hasonlít a DOS CLS parancsához.

Feladat

Töröld le a képernyőt, de a parancsot először tévesen add ki, így:

```
KLEAR
```

Ezután javítsd ki a parancsot!

18. fejezet

Hogyan lépünk ki a dBASE programból?

A dBASE programból a QUIT paranccsal kell kilépni.

Ha a QUIT szó begépelése után leütjük az Enter-t, akkor már ki is léptünk a dBASE programból.

Feladat

Lépj ki a dBASE-ből, majd indítsd el újra!

19. fejezet

Adatbázisfájl létrehozása

a) Az adatbázisfájl létrehozásának kezdeményezése

Adatbázisfájl létrehozását a CREATE paranccsal kell kezdeményezni.

A CREATE szó begépelése után egy szóközt kell gépelni, majd meg kell adni a létrehozandó adatbázisfájl teljes azonosítóját (vagyis meg kell adni, hogy melyik meghajtó melyik katalógusában kívánjuk létrehozni az adatbázisfájlt, és mi legyen az azonosítója). Mivel az adatbázisfájl kiterjesztése kötelezően DBF, ezért a .DBF elhagyható. A parancsot az Enter leütésével kell zárni.

Példa

Ha olyan adatbázisfájlt akarnánk létrehozni a gyakorlólemez KM katalógusában, amelynek azonosítója WORK.DBF lesz, akkor a létrehozást ezzel a paranccsal kellene kezdeményezni:

```
CREATE A:\KM\WORK.DBF
```

vagy rövidebben:

```
CREATE A:\KM\WORK
```

Feladat

Adatbázisfájlt kell létrehozni a tanulókról. Megállapodásunk szerint az adatbázisfájlt a gyakorlólemezünk KM katalógusába tesszük majd. Állapodjunk meg abban is, hogy az adatbázisfájl neve TANULO lesz.

Add ki azt a parancsot, amellyel kezdeményezni kell ennek az adatbázisfájlnak a létrehozását!

Megjegyzés

Mihelyt az adatbázisfájl létrehozását kezdeményező parancsot kiadtuk, a készülő adatbázisfájl neve megjelenik az adatbázisfájl-cellában.

b) Az adatbázisfájl szerkezetének begépelése

A CREATE parancs végrehajtása során megjelenik a képernyőn egy olyan munkafelület, amely lehetővé teszi az adatbázisfájl szerkezetének begépelését. Minden mezővel kapcsolatban az alábbi rovatokat kell kitölteni:

- Field Name (a mező neve)
- Type (típusa)
- Width (hossza)
- Dec (a tizedesjegyek száma).

A mező nevének megadása után mindig Enter-t kell ütni.

A mező típusaként a dBASE minden esetben felajánlja a karakteres típust. Ha a mező valóban ilyen típusú, akkor elég leütni az Enter-t. Egyébként a mező típusának rövidítését kell megadni (L, D, N), ilyenkor azonban nem kell leütni az Enter-t.

Karakteres mezőnél és numerikus mezőnél meg kell még adni a mező hosszát, majd Enter-t kell ütni.

Numerikus mezőnél a tizedesjegyek számát is meg kell adni, majd Enter-t kell ütni.

Feladat

Gépelj be a tanulókról készülő adatbázisfájl szerkezetét! Használd az alábbi táblázatot:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>
NEV	C	25	-
SZUL_HELY	C	20	-
SZUL_DATUM	D	8	-
TESTVER	N	2	0
ATLAG	N	4	2
KOLLEGIUM	L	1	-

A begépelés során a mezők között a ↑ vagy a ↓ nyílbillentyűvel közlekedünk.

Ha a szerkezetet teljes egészében begéveltük, akkor alkalmazni kell a Ctrl End billentyűkombinációt. Ennek hatására a szerkezet tárolódni fog. (A dBASE a szerkezetet magában az adatbázisfájlban tárolja.)

A Ctrl End billentyűkombináció alkalmazása után az alábbi üzenetet kapjuk:
Press ENTER to confirm. Any other key to resume.

Magyarul:

Üss Enter-t, ha valóban befejezted. Egyébként üss bármit.

Ha tehát bármilyen más billentyűt ütünk le, akkor folytathatjuk a szerkezet begépelését.

Miután az Enter-rel jeleztük, hogy valóban befejeztük a szerkezet begépelését, a dBASE a következő kérdést teszi fel:

Input data records now? (Y/N)

Magyarul:

Viszel be most adatokat az adatbázisfájlba? (Y/N)

Két választásunk van:

— Ha most nincs időnk adatok bevitelére, akkor üssünk N-t. Ez esetben csak a szerkezet kerül bele az adatbázisfájlba, az adatokkal való feltöltésére később (de már nem a CREATE paranccsal) kerülhet sor.

— Ha még van időnk, akkor üssünk Y-t. Ez esetben elkezdődhet az adatbázisfájl feltöltése adatokkal.

Feladat

Mentsd el az adatbázisfájl szerkezetét! Az ezt követő kérdésre úgy válaszolj, hogy azonnal lehetővé váljon az adatbázisfájl feltöltése adatokkal!

c) Az adatbázisfájl feltöltése adatokkal

Ha az adatbázisfájl szerkezetének megadását követően az

Input data records now? (Y/N)

kérdésre Y-nal válaszoltunk, akkor azonnal elkezdhetjük az adatbázisfájl feltöltését adatokkal.

A feltöltés rekordonként történik.

Minden adat begépelése után Enter-t kell ütni.

A gépelést megkönnyíti, hogy a mezők neve állandóan látható a képernyőn, és azt is látjuk, hogy a mezők megadásához mennyi hely áll rendelkezésünkre.

További könnyítések:

- logikai mezőnél kérdőjel van írva a mezőbe
- dátum jellegű mezőnél a határolójel már be van írva a mezőbe
- numerikus mezőnél a tizedespont már be van írva a mezőbe (föltéve, hogy a mező tartalmaz tizedesjegyet).

Példa

A tanulókról készülő adatbázisfájl feltöltésekor ilyen a képernyő:

NEV	
SZUL_HELY	
SZUL_DATUM	. .
TESTVER	
ATLAG	.
KOLLEGIUM	?

Ha a rekordok begépelésével végeztünk, akkor alkalmazni kell a Ctrl End billentyűkombinációt. Ennek hatására kerülnek be ugyanis az adatok az adatbázisfájlba.

Ha a Ctrl End billentyűkombináció alkalmazásakor üres rekord látható a képernyőn, akkor ez az üres rekord is belekerül az adatbázisfájlba. Ezt úgy kerülhetjük el, hogy a Page Up (Pg Up) billentyűvel visszalépünk egy nem üres rekordba, s csak ezt követően alkalmazzuk a Ctrl End billentyűkombinációt.

Feladat

Töltsd fel a tanulókról készülő adatbázisfájlt a papíron készített nyilvántartás alapján!

Megjegyzés

Az adatbázisfájl feltöltése során minden rekord kap egy sorszámot.

20. fejezet

A gépelési hibák kijavítása

Az adatok begépelésekor elkövetett hibák kijavításához a következőket kell tudni:

1. Az Insert (Ins) billentyűvel lehet váltani a billentyűzet felülíró és betoldó üzemmódja között.
2. A Backspace billentyűvel a kurzortól balra eső karaktert lehet kitörölni.

3. A Delete (Del) billentyűvel azt a karaktert lehet kitörölni, amelyiken a kurzor áll.
4. A → vagy a ← nyílbillentyűvel a mezőn belül egy karakterrel lehet a kurzort mozgatni.
5. Az End billentyű a mezőn belül a következő szó elejére viszi a kurzort.
6. A Home billentyű a mezőn belül az előző szó elejére viszi a kurzort.
7. A ↑ vagy a ↓ nyílbillentyűvel a mezők között lehet közlekedni.
8. A Page Down (Pg Dn) billentyűvel a következő képernyőoldalra vihetjük a kurzort, a Page Up (Pg Up) billentyűvel pedig az előzőre.
9. Ha alkalmazzuk a Ctrl Y billentyűkombinációt, akkor törlődik a mezőnek a kurzortól jobbra eső része.

Fontos megjegyzések

1. A legkorábbi dátum, amit megadhatunk, a Krisztus születése utáni 100. év januárjának első napja. A legkésőbbi megadható dátum a 9999. év decemberének 31. napja. E két dátum között minden tényleges dátum megadható.

Ha érvénytelen dátumot adunk meg, akkor hibajelzést kapunk, és a gépelést csak a szököz leütése után lehet folytatni.

2. Ha az adatok begépelése során olyan hibát követünk el, aminek hatására akaratunk ellenére visszajutunk a parancssorba, akkor az adatok begépelése úgy folytatható, hogy a parancssorba begépeljük az APPEND szót, majd leütjük az Enter billentyűt.

Feladatok

1. Hozz létre olyan adatbázisfájlt, amely otthoni könyveid nyilvántartását teszi lehetővé. Minden könyvedről az alábbi adatokat tartsd nyilván:

- szerzője
- címe
- kiadásának éve
- vannak-e benne képek?
- ára.

Az adatbázisfájl szerkezete ez legyen:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Típsz.</i>	<i>Jelentés</i>
SZERZO	C	25	-	A könyv szerzője
CIM	C	30	-	Címe
KIADAS	N	4	0	Kiadásának éve
KEPEK	L	1	-	Vannak-e benne képek?
AR	N	6	2	Ára

Az adatbázisfájlt a gyakorlólemez KM katalógusába tedd, a neve KONYV legyen!
Ha valós adatok nem állnak rendelkezésedre, akkor az adatbázisfájlt az alábbi adatokkal töltsd fel:

SZERZO	CIM	KIADAS	KEPEK	AR
Dosztojevszkij	Fehér éjszakák	1956	N	32.50
Tolsztoj	Háború és béke	1956	N	77.00
Csehov	Színművek	1955	N	23.00
Szaltikov-Scsedrin	A Galavljov család	1954	Y	16.00
Puskin	Anyegin	1953	Y	36.00
Turgenyev	Apák és fiúk	1954	N	19.50
Gorkij	Az Artamonovok	1953	N	19.50
Gogol	Holt lelkek	1974	Y	36.00
Herzen	Ki a bűnös?	1956	N	32.00

2. Hozz létre adatbázisfájlt a közeli ismerőseidről! Mindegyikükről ezeket az adatokat tartsd nyilván:

- neve
- születési dátuma
- címe (irányítószám, utca, házsám).

Az adatbázisfájl szerkezete ez legyen:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>
NEV	C	20	-
SZUL_DATUM	D	8	-
IRSZ	N	4	0
UTCA	C	12	-
HAZSZAM	C	6	-

Az adatbázisfájlt a gyakorlólemez KM katalógusába tedd, a neve ISMEROS legyen.

Ha valós adatok nem állnak rendelkezésedre, akkor az adatbázisfájlt ezekkel az adatokkal töltsd fel:

NEV	SZUL_DATUM	IRSZ	UTCA	HAZSZAM
Nagy Anna	1964.12.26	6133	Kossuth	12/A
Antal László	1936.11.30	6000	Diófa	9-11.
Göröcs József	1956.10.21	6456	Csóka	8.
Kovács Annamária	1982.12.24	6000	Petőfi	13.
Kiss Anett	1984.01.01	6000	Hunyadi	20.
Gidai Anikó	1967.03.14	6000	Báthory	33/B
Berkecz Balázs	1965.11.13	1191	Cserepes	18-22.
Homonnai Gyula	1967.12.06	1052	Váci	18.

3. Hozd létre azt az adatbázisfájlt, amely az ISMEROS.DBF adatbázisfájlban szereplő irányítószámokat és a nekik megfelelő településneveket tartalmazza! (Ebben az adatbázisfájlban minden irányítószám csak egyszer szerepeljen!)

Az adatbázisfájl szerkezete ez legyen:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tízs.</i>
SZAM	N	4	0
TELEPULES	C	20	-

Az adatbázisfájlt a gyakorlólemez KM katalógusába tedd, a neve POSTA legyen!

Ha az ISMEROS.DBF adatbázisfájlt a könyvben megadott adatokkal töltötted fel, akkor a POSTA adatbázisfájl ezekkel az adatokkal töltsd fel:

SZAM	TELEPULES
6133	Jászszentlászló
6456	Madaras
6000	Kecskemét
1191	Budapest
1052	Budapest

21. fejezet

Az adatbázisfájlok számbavétele

Ha tudni akarjuk, hogy egy katalógusban milyen adatbázisfájlok vannak, akkor a DIR parancsot kell használni.

A DIR szó begépelését egy szököz követi, majd hivatkozni kell a szóban forgó katalógusra, ezután (ha nem a főkatalógusról van szó) egy \ következik, végül le kell ütni az Enter-t, és már látható is a képernyőn a katalógusban található valamennyi adatbázisfájl

- azonosítója
- rekordjainak száma
- készítésének dátuma
- mérete byte-ban kifejezve,

valahogy így:

Database Files	# Records	Last Update	Size
LAKOS.DBF	5	1994.07.25	1025
TELEFON.DBF	8	1994.07.25	1732
VASARLAS.DBF	16	1994.07.24	1141

Példa

Ha a C: meghajtó DBASE katalógusában lenne egy ADAT nevű alkatalógus, és az abban található adatbázisfájlokat akarnánk számba venni, akkor ezt a parancsot kellene kiadni:

```
DIR C:\DBASE\ADAT\
```

Megjegyzés

Ezt a DIR parancsot ne tévesszük össze a DOS DIR parancsával!

A kettő között a lényeges különbség az, hogy a DOS DIR parancsa a katalógusban levő valamennyi fájlt mutatja (meg az alkatalógusokat is), a dBASE DIR parancsa viszont (ha másképpen nem rendelkezünk) csak a DBF kiterjesztésű fájlokat.

Ha azt akarnánk, hogy a dBASE DIR parancsa is mutassa az összes fájlt, akkor a parancs végére még ezt kellene begépelnünk:

```
*.*
```

Feladat

Nézd meg, hogy milyen adatbázisfájlok vannak a gyakorlólemez KM katalógusában!

22. fejezet

Adatbázisfájl megnyitása

Ha egy adatbázisfájlt használni akarunk, akkor azt előzőleg meg kell nyitni.

A megnyitás az USE paranccsal történik.

Az USE szó begépelése után egy szóközt kell gépelni, majd meg kell adni a megnyitandó adatbázisfájl teljes azonosítóját, végül Enter-t kell ütni.

Példa

Ha a C: meghajtó DBASE katalógusában lenne egy ADAT nevű alkatalógus, és abban lenne egy UZLET nevű adatbázisfájl, akkor azt ezzel a paranccsal nyitnánk meg:

```
USE C:\DBASE\ADAT\UZLET.DBF
```

vagy rövidebben:

```
USE C:\DBASE\ADAT\UZLET
```

Megjegyzések

1. Egy adatbázisfájlt akkor is meg lehet nyitni, ha adat még nincs benne, csupán a szerkezete van begépelve.

2. Ha olyan adatbázisfájlt akarunk megnyitni, amelynek előzőleg még a szerkezetét sem gépeltük be, akkor a következő hibaüzenetet kapjuk:

```
File does not exist.
```

Magyarul:

Nincs ilyen adatbázisfájl.

23. fejezet

Adatbázisfájl lezárása

Ha már nem akarjuk használni a nyitott adatbázisfájlt, akkor le kell zárunk. A lezárás úgy valósítható meg, hogy begépeljük az USE szót, majd Enter-t ütünk.

Megjegyzések

1. A nyitott adatbázisfájl automatikusan lezáródik akkor, ha egy másik adatbázisfájlt nyitunk meg.

2. A létrehozásakor az adatbázisfájl automatikusan megnyílik, és mindaddig nyitott marad, míg le nem zárjuk, vagy ki nem lépünk a dBASE-ből, vagy meg nem nyitunk egy másik adatbázisfájlt.

Feladat

A gyakorlólemez KM katalógusában jelenleg négy adatbázisfájl található:

TANULO.DBF

KONYV.DBF

ISMEROS.DBF

POSTA.DBF

a) Nyisd meg a TANULO.DBF adatbázisfájlt!

b) Nyisd meg a KONYV.DBF adatbázisfájlt!

c) Zárd le a KONYV.DBF adatbázisfájlt!

24. fejezet

Az adatbázisfájl szerkezetének kiírása

A nyitott adatbázisfájl szerkezetét a DISPLAY STRUCTURE paranccsal írathatjuk ki a képernyőre.

Ha a DISPLAY STRUCTURE szavakat begépeztük, majd leütöttük az Enter-t, akkor a nyitott adatbázisfájlról az alábbi információk jelennek meg a képernyőn:

- neve
- rekordjainak száma
- létrehozásának dátuma
- szerkezete
- a rekordok hossza.

Előfordul, hogy az adatbázisfájl szerkezete nem fér el egyszerre a képernyőn. Ilyenkor ha már megtelt a képernyő, akkor az alábbi üzenet látható:

Press any key to continue...

Magyarul:

Üss le egy billentyűt a folytatáshoz...

Példa

A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl szerkezetét a következőképpen írathatjuk ki a képernyőre:

a) Az adatbázisfájl megnyitása:

```
USE A:\KMTANULO.DBF
```


b) A szerkezet kiírása:

DISPLAY STRUCTURE

Ekkor a képernyőn ezt látjuk:

Structure for database: a:\km\tanulo.dbf

Number of data records: 17

Date of last update : 1994.07.24

Field	Field Name	Type	Width	Dec
1	NEV	Character	25	
2	SZUL_HELY	Character	20	
3	SZUL_DATUM	Date	8	
4	TESTVER	Numeric	2	
5	ATLAG	Numeric	4	2
6	KOLLEGIUM	Logical	1	
**	Total	**	61	

Feladatok

1. Írasd ki a képernyőre a gyakorlólemez KM katalógusában lévő KONYV.DBF adatbázisfájl szerkezetét!
2. Írasd ki a képernyőre a gyakorlólemez KM katalógusában lévő ISMEROS.DBF adatbázisfájl szerkezetét!
3. Írasd ki a képernyőre a gyakorlólemez KM katalógusában található POSTA.DBF adatbázisfájl szerkezetét!

Megjegyzések

1. A DISPLAY STRUCTURE parancs különösen hasznos akkor, ha elfelejtettük az adatbázisfájl szerkezetét.
2. A LIST STRUCTURE parancs ugyanígy működik, azzal a különbséggel, hogy ha a képernyő betelt, akkor a kiírás nem függesztődik föl.

25. fejezet

Az aktuális rekord fogalma

A nyitott adatbázisfájlban egy rekord a többivel szemben általában ki van tüntetve, ezt nevezzük aktuális rekordnak.

Célszerű azt a rekordot tenni aktuálissá, amelyben levő adatokra éppen szükségünk van.

Mint tudjuk, a rekord-cella az aktuális rekord sorszámát és az összes rekord számát mutatja. Ha ebben a cellában az „EOF” felirat látható, ez arra utal, hogy egyik rekord

sem aktuális. Ha pedig ebben a cellában a „None” felirat látható, ez azt jelzi, hogy az adatbázisfájlban egyetlen rekord sincs.

26. fejezet

Aktuális rekord kijelölése

Ha a nyitott adatbázisfájlban ki akarjuk jelölni az aktuális rekordot, akkor a GO vagy a SKIP parancsot használjuk.

a) Kijelölés a GO paranccsal

A GO parancs arra szolgál, hogy a nyitott adatbázisfájlban a sorszám alapján jelöljük ki az aktuális rekordot.

A GO szó után szóközt kell gépelni, majd meg kell adni annak a rekordnak a sorszámát, amelyet aktuálissá akarunk tenni, végül le kell ütni az Enter-t.

Ha olyan nagy sorszámot adunk meg, amilyen sorszámú rekord nincs is az adatbázisfájlban, akkor egyik rekord sem lesz aktuális.

Példa

A nyitott adatbázisfájl 7. rekordja ezzel a paranccsal tehető aktuálissá:

```
GO 7
```

Megjegyzések

1. Az első rekordot aktuálissá tehetjük így is:

```
GO TOP
```

2. Az utolsó rekord így is aktuálissá tehető:

```
GO BOTTOM
```

3. A parancs használatakor a GO elhagyható, vagy helyette GOTO is írható. Tehát az alábbi parancsok hatása ugyanaz:

```
GO 15
```

```
15
```

```
GOTO 15
```

Feladatok

1. Nyisd meg a gyakorlólemez KM katalógusában található TANULO.DBF adatbázisfájl!

2. Mondd meg, hogy a megnyitást követően mely rekordja vált aktuálissá!

3. Tedd aktuálissá a 3. rekordját!

4. Tedd aktuálissá a legutolsó rekordját!
5. Tedd aktuálissá a legelső rekordját!
6. Zárd le az adatbázisfájlt!

b) Kijelölés a SKIP paranccsal

A SKIP paranccsal az aktuálishoz képest valamennyivel nagyobb vagy kisebb sorszámú rekord jelölhető ki aktuális rekordnak.

A SKIP szó után egy szóközt kell gépelni, majd meg kell adni azt a pozitív vagy negatív számot, amely megmutatja, hogy az aktuálishoz képest mennyivel nagyobb vagy kisebb sorszámú rekordot akarjuk aktuálissá tenni.

Példák

1. Ha jelenleg a 6. rekord az aktuális, akkor a
SKIP 4
parancs végrehajtása után a 10. rekord válik aktuálissá.

2. Ha jelenleg a 9. rekord az aktuális, akkor a
SKIP -3
parancs végrehajtása után a 6. rekord válik aktuálissá.

Megjegyzések

1. Ha a SKIP parancsban nem adunk meg számot, akkor a hatása azonos a
SKIP 1
parancs hatásával.

2. Ha a SKIP parancsban olyan nagy számot adunk meg, hogy ezáltal olyan sorszámú rekord válna aktuálissá, amilyen nincs is az adatbázisfájlban, akkor nem lesz aktuális rekord.

3. Ha a SKIP parancsban negatív szám megadásával próbálunk aktuálissá tenni nem létező sorszámú rekordot, akkor az 1. rekord lesz az aktuális.

Példák

1. Ha jelenleg a 7. rekord az aktuális, akkor a
SKIP
parancs hatására a 8. rekord válik aktuálissá.

2. Ha jelenleg a 9. rekord az aktuális, és az adatbázisfájl mindössze 18 rekordot tartalmaz, akkor a

SKIP 16

parancs kiadása után nem lesz aktuális rekord.

3. Ha jelenleg a 9. rekord az aktuális, akkor a

SKIP -15

parancs kiadása után az 1. rekord válik aktuálissá.

Feladatok

1. Nyisd meg a gyakorlólemez KM katalógusában található KONYV.DBF adatbázisfájlt!

2. Tedd aktuálissá a legutolsó rekordot, majd tedd aktuálissá egymás után mindig a kettővel kisebb sorszámú rekordot mindaddig, míg a legelső rekord nem válik aktuálissá!

3. Zárd le az adatbázisfájlt!

Megjegyzés

Gyakran előfordul, hogy egy parancs végrehajtása után már nem az a rekord az aktuális, amely a végrehajtása előtt aktuális volt.

Érdemes tehát odafigyelni a rekord-cellára, mert abból mindig megtudhatjuk, hogy melyik az aktuális rekord.

27. fejezet

Rekordok hozzáfűzése az adatbázisfájl végéhez

Ha a nyitott adatbázisfájl végéhez újabb rekordokat akarunk hozzáfűzni, akkor az APPEND parancsot kell használni.

Az APPEND szó begépelését követően Enter-t kell ütni, és már kezdődhet is a hozzáfűzés.

A hozzáfűzés során az adatok begépelése ugyanúgy történik, mint ahogyan eredetileg az adatbázisfájlt feltöltöttük adatokkal.

A hozzáfűzést a Ctrl End billentyűkombinációval kell befejezni.

Példa

A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl végéhez így fűzünk hozzá újabb rekordokat:

a) Megnyitás:

USE A:\KMTANULO.DBF

b) Hozzáfűzés:

APPEND

c) Lezárás:

USE

Feladatok

1. Fűzd hozzá a gyakorlólemez KM katalógusában található KONYV.DBF adatbázisfájl végéhez az alábbi rekordokat:

SZERZO	CIM	KIADAS	KEPEK	AR
Gogol	A köpönyeg	1951	Y	12.00
Dosztójevszkij	Bűn és bűnhődés	1973	N	30.00
Tolsztoj	Anna Karenina	1958	N	48.00
Csehov	A kutyás hölgy	1960	Y	28.00
Tolsztoj	Feltámadás	1950	N	23.00
Goncsarov	Oblomov	1953	N	28.00

2. Fűzz hozzá a gyakorlólemez KM katalógusában található ISMEROS.DBF adatbázisfájl végéhez még két rekordot!

Gondoskodj róla, hogy a hozzáfűzött rekordokban szereplő irányítószámok (a hozzátartozó településnévvel együtt) benne legyenek a POSTA.DBF adatbázisfájlban is!

28. fejezet

Rekord betoldása az adatbázisfájlba

Lehetőség van arra is, hogy az adatbázisfájl két rekordja közé toldjunk be egy új rekordot.

Az INSERT paranccsal az aktuális rekord után vagy elé toldhatunk be egy rekordot.

Ha csupán az INSERT szót gépeljük be, majd leütjük az Enter-t, akkor az aktuális rekord után toldhatunk be egy rekordot.

Ha az INSERT BEFORE szavakat gépeljük be, és azután ütjük le az Enter-t, akkor az aktuális rekord elé toldhatunk be egy rekordot.

A betoldás során az adatok begépelése ugyanúgy történik, mint ahogyan eredetileg az adatbázisfájlt feltöltöttük adatokkal.

Példa

Ha a gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl 4. és 5. rekordja közé akarnánk egy rekordot betoldani, akkor ezeket a parancsokat adnánk ki:

a) Megnyitás:

```
USE A:\KMITANULO.DBF
```

b) A 4. rekord aktuálissá tétele:

```
GO 4
```

c) Betoldás a 4. rekord után:

```
INSERT
```

d) Lezárás:

```
USE
```

Megjegyzés

A betoldást követően a rekordok átsorszámozódnak.

Feladat

Toldj be egy rekordot a gyakorlólemez KM katalógusában levő ISMEROS.DBF adatbázisfájl 3. rekordja elé!

Gondoskodj róla, hogy a betoldott rekordban szereplő irányítószám (a hozzá tartozó településnévvel együtt) benne legyen a POSTA.DBF adatbázisfájlban is!

29. fejezet

Adatbázisfájl tartalmának kiíratása

Ha látni akarjuk a képernyőn a nyitott adatbázisfájl tartalmát, akkor a DISPLAY vagy a LIST parancsot kell használni.

A két parancs közötti lényeges különbség az, hogy ha másképpen nem rendelkezünk, akkor a DISPLAY parancs csak az aktuális rekord tartalmát írja ki, a LIST viszont valamennyi rekord tartalmát.

Miután begépetük a DISPLAY vagy a LIST szót, majd leütöttük az Enter-t, már látható is a képernyőn a kívánt tartalom.

Példák

1. Ha a gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl teljes tartalmát akarjuk kiíratni, akkor ezeket a parancsokat adjuk ki:

a) Megnyitás:

USE A:\KMITANULO.DBF

b) Kiíratás:

LIST

c) Lezárás:

USE

2. Ha a gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl 3. rekordjának tartalmát akarjuk kiíratni, akkor ezeket a parancsokat adjuk ki:

a) Megnyitás:

USE A:\KMITANULO.DBF

b) A 3. rekord aktuálissá tétele:

GO 3

c) Kiíratás:

DISPLAY

d) Lezárás:

USE

Feladatok

1. Írasd ki a gyakorlólemez KM katalógusában található KONYV.DBF adatbázisfájl összes rekordjának tartalmát!

2. Írasd ki a gyakorlólemez KM katalógusában található ISMEROS.DBF adatbázisfájl 2. rekordjának tartalmát!

Megjegyzés

A DISPLAY és a LIST parancs tárgyalását ezzel még nem fejeztük be, további lehetőségeikre a későbbiekben folyamatosan visszatérünk.

30. fejezet

Adatbázisfájl tartalmának módosítása

Ha módosítani akarjuk a nyitott adatbázisfájl tartalmát, akkor négy parancs közül választhatunk: a módosítást az APPEND, az EDIT, a CHANGE és a BROWSE parancsok bármelyikével elvégezhetjük.

a) Módosítás az APPEND paranccsal

Az APPEND parancsot eddig arra használtuk, hogy rekordokat fűzzünk a nyitott adatbázisfájl végéhez.

Az APPEND parancs azonban azt is lehetővé teszi, hogy módosítsuk a nyitott adatbázisfájl tartalmát.

Ennek érdekében a következőket kell tenni:

1. Az APPEND szó begépelését követően Enter-t kell ütni, majd a Page Up (Pg Up) és a Page Down (Pg Dn) billentyűk segítségével a módosítandó rekordhoz kell lapozni.
2. El kell végezni a szükséges módosításokat.
3. A módosítások befejeztével alkalmazni kell a Ctrl End billentyűkombinációt.

Feladat

A gyakorlólemez KM katalógusában található TANULO.DBF adatbázisfájlban az APPEND parancs használatával végezd el a következő módosításokat:

1. Az utolsó előtti rekordban a testvérek számát növeld meg 1-gyel!
2. A 3. rekordban a születési dátumot változtasd meg két nappal későbbi időpontra!

b) Módosítás az EDIT paranccsal

Ha az EDIT szót begépeljük, majd Enter-t ütünk, akkor lehetővé válik a nyitott adatbázisfájl aktuális rekordjának módosítása.

Ha még egyéb rekordokat is módosítani akarunk ugyanezzel a paranccsal, akkor sorban egymásután a módosítandó rekordokhoz kell lapozni a Page Up (Pg Up) és a Page Down (Pg Dn) billentyűk segítségével, és el kell végezni a szükséges módosításokat.

A módosítások befejeztével alkalmazni kell a Ctrl End billentyűkombinációt.

Feladat

A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájlban az EDIT parancs használatával végezd el a következő módosításokat:

1. A köpönyeg című könyv árát javítsd ki 14 Ft-ra!
2. Kiderült, hogy az Oblomov című könyvben vannak képek. Gondoskodj róla, hogy az adatbázisfájl is ezt mutassa!

c) Módosítás a CHANGE paranccsal

Ha a CHANGE szót begépeljük, majd Enter-t ütünk, akkor a nyitott adatbázisfájlban éppen úgy végezhetjük el a módosítást, mint az EDIT paranccsal.

d) Módosítás a BROWSE paranccsal

Ha az APPEND , az EDIT vagy a CHANGE paranccsal végezzük el az adatbázisfájl módosítását, egyszerre mindig csak egy rekord látható a képernyőn.

Ha viszont a BROWSE paranccsal végezzük a módosítást, akkor egyszerre sok rekord látható a képernyőn, ezért a rekordok között könnyebb eligazodni.

Ha a BROWSE szó begépelése után Enter-t ütünk, akkor máris lehetővé válik a nyitott adatbázisfájl bármely adatának módosítása, csak éppen előbb a módosítandó adatot meg kell keresni a kurzormozgató billentyűkkel.

A BROWSE parancs lehetővé teszi azt is, hogy az adatbázisfájl végéhez új rekordokat fűzzünk hozzá. Ezt a következőképpen kell csinálni:

1. A legutolsó rekordra kell vinni a kurzort, majd le kell ütni a ↓ nyílbillentyűt.
2. Ekkor a dBASE ezt a kérdést teszi fel:

====>Add new records? (Y/N)

Magyarul:

Akarsz hozzáfűzni új rekordokat? (Y/N)

Ha erre a kérdésre Y-nal válaszolunk, majd Enter-t ütünk, akkor lehetővé válik új rekordok hozzáfűzése az adatbázisfájl végéhez.

A módosítások és az esetleges hozzáfűzések befejeztével alkalmazni kell a Ctrl End billentyűkombinációt.

Megjegyzés

Ha a parancs használatakor olyan mezőket akarunk láthatóvá tenni, amelyek jelenleg nem láthatók, akkor a Ctrl → vagy a Ctrl ← billentyűkombinációt kell alkalmazni.

Feladat

A gyakorlólemez KM katalógusában található ISMEROS.DBF adatbázisfájl végéhez fűzz hozzá két rekordot a BROWSE parancs használatával!

Gondoskodj róla, hogy a hozzáfűzött rekordokban szereplő irányítószámok (a hozzátartozó településnévvel együtt) benne legyenek a POSTA.DBF adatbázisfájlban is!

31. fejezet

Rekordok törlése az adatbázisfájlból

a) A törlés elvi kérdései

A feleslegessé vált rekordokat az adatbázisfájlból rendszeresen törölni kell. A törlés veszélyes dolog. Mindig fennáll annak a veszélye, hogy véletlenül olyan rekordot törölünk ki, amelyre még szükség van. A dBase ezért úgy rendelkezik, hogy a törlést több lépésben kell elvégezni:

— Először a DELETE paranccsal ki kell jelölni a törlésre szánt rekordokat.

— Ha rájövünk, hogy bizonyos rekordokat mégsem akarunk törölni, akkor ezek kijelöltsége a RECALL paranccsal megszüntethető.

— Ha már biztosak vagyunk abban, hogy valamennyi törlésre kijelölt rekord törölhető, akkor azok visszavonhatatlan törlését a PACK paranccsal kell elvégezni.

A törlésre kijelölt rekordok még benne vannak az adatbázisfájlban. Kiíratáskor a rekordsorszám melletti * jelzi, hogy a rekord törlésre ki van jelölve. Ha a kijelöltséget megszüntetjük, akkor a * eltűnik.

Megjegyzés

Már említettük, hogy a rekordok hossza 1-gyel nagyobb, mint a mezők hosszának összege. Ennek oka az, hogy a fent említett * számára minden rekordban fõnn van tartva 1 byte.

b) A DELETE parancs használata

Ha másképpen nem rendelkezünk, akkor a DELETE parancs az aktuális rekordot jelöli ki törlésre. Ennek érdekében csupán a DELETE szót kell begépelni, majd le kell ütni az Enter billentyűt.

Példa

Ha a nyitott adatbázisfájl 5. és 9. rekordját akarjuk törlésre kijelölni, akkor ezeket a parancsokat adjuk ki:

```
GO 5  
DELETE  
GO 9  
DELETE
```


c) A RECALL parancs használata

Ha másképpen nem rendelkezünk, akkor a RECALL parancs az aktuális rekord kijelöltségét szünteti meg. Ennek érdekében csupán a RECALL szót kell begépelni, majd le kell ütni az Enter billentyűt.

Példa

Ha a nyitott adatbázisfájl 9. rekordjának kijelöltségét akarjuk megszüntetni, akkor ezeket a parancsokat adjuk ki:

```
GO 9  
RECALL
```

d) A PACK parancs használata

Ha végleg el akarjuk távolítani az adatbázisfájlból a törlésre kijelölt rekordokat, akkor a PACK szót kell begépelni, majd le kell ütni az Enter billentyűt.

A PACK paranccsal óvatosan kell bánni, mert az eltávolított rekordok visszaállítására a dBASE már nem ad lehetőséget. A parancs végrehajtása után megmaradó rekordok átsorszámozódnak.

Példa

Ha ki akarnánk törölni a gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl 7. és 10. rekordját, akkor a következő parancsokat adnánk ki:

a) Az adatbázisfájl megnyitása:

```
USE A:\KM\tanulo.dbf
```

b) A 7. és a 10. rekord kijelölése törlésre:

```
GO 7  
DELETE  
GO 10  
DELETE
```

c) A kijelölt rekordok eltávolítása:

```
PACK
```

d) Az adatbázisfájl lezárása:

```
USE
```

Feladat

Töröld ki a gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl 2. rekordját! Győződj meg róla, hogy a törlés sikeres volt!

Megjegyzés

Rekordoknak törlésre való kijelölése az APPEND, az EDIT, a CHANGE vagy a BROWSE paranccsal is elvégezhető.

— Az APPEND, az EDIT vagy a CHANGE parancs használatakor a Pg Dn és a Pg Up billentyűvel a törlésre szánt rekordhoz kell lapozni, majd alkalmazni kell a Ctrl U billentyűkombinációt. Ennek hatására az üzemmód-cella jobb oldali részén a „Del” felirat jelenik meg annak jelzésére, hogy a rekord ki van jelölve törlésre.

— A BROWSE parancs használatakor a ↑ és ↓ nyílbillentyűk segítségével a kiválasztó sávot a törlésre szánt rekordra kell vinni, majd alkalmazni kell a Ctrl U billentyűkombinációt. Ennek hatására feltűnik a „Del” felirat az üzemmód-cella jobb oldali részében.

Mind a négy parancsra elmondható:

— Ha a Ctrl U billentyűkombinációt olyan rekordra alkalmazzuk, ami törlésre van kijelölve, akkor a kijelöltsége megszűnik.

— A parancsból a Ctrl End billentyűkombinációval kell kilépni.

— A kilépés után a rekordok végleges eltávolítása érdekében ki kell adni a PACK parancsot.

32. fejezet

Az adatbázisfájl összes rekordjának törlése

Ha az adatbázisfájl összes rekordját törölni akarjuk, akkor érdemes használni a ZAP parancsot.

A ZAP parancs ugyanis az adatbázisfájl összes rekordját azonnal és visszavonhatatlanul kitörli, függetlenül attól, hogy mely rekordok voltak törlésre kijelölve.

Mindössze a ZAP szót kell begépelni, majd le kell ütni az Enter billentyűt.

Az adatbázisfájl szerkezetét a ZAP parancs nem számolja fel, tehát a végrehajtása után azonnal lehet új rekordokat bevinni a rekordjaitól megfosztott adatbázisfájlba.

Példa

Ha a házi könyvtárunk valamennyi kötetét eladtuk volna, akkor a gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl valamennyi rekordját ki kellene törölnünk. Ezt legegyszerűbben így tehetnénk meg:

a) Az adatbázisfájl megnyitása:

```
USE A:\KMKONYV.DBF
```

b) Az összes rekord törlése:

```
ZAP
```


c) Az adatbázisfájl lezárása:
USE

33. fejezet

A konstans fogalma

A parancsban konkrétan megadott adatot konstansnak nevezzük.

Példa

Már eddig is használtunk konstansokat, például a

```
GO 9
```

parancsban szereplő 9 egy konstans volt.

34. fejezet

Konstanstípusok

Három konstanstípus létezik:

- numerikus konstans
- karakteres konstans
- logikai konstans.

Numerikus konstansról akkor beszélünk, ha számot adunk meg egy parancsban konkrétan.

Karakteres konstansról akkor beszélünk, ha karaktersorozatot adunk meg egy parancsban konkrétan.

Logikai konstansról akkor beszélünk, ha logikai adatot adunk meg egy parancsban konkrétan.

Megjegyzés

Mivel dátum jellegű konstans nem létezik, ezért dátumot nem adhatunk meg konkrétan egyetlen parancsban sem. Majd látni fogjuk, hogy ez a megszorítás miként hidalható át.

35. fejezet

A konstansok megadása

a) Numerikus konstans megadása

Numerikus konstans megadásakor egyszerűen be kell gépelni a számot.

Példa

Numerikus konstansok:

-18.25

1965

+34.22

b) Karakteres konstans megadása

Karakteres konstans megadásakor a karaktersorozatot idézőjelbe vagy aposztrófok közé kell tenni.

Példa

Karakteres konstansok:

“Bartók Béla”

‘76-324616’

‘76324616’

Megjegyzés

Az idézőjel belső kódja 34, az aposztrófé pedig 39.

c) Logikai konstans megadása

Logikai konstans megadásakor a logikai adat (Y, N, T vagy F) elé is, meg mögé is egy pontot kell tenni.

Példa

Logikai konstansok:

.Y. (‘igaz’)

.N. (‘hamis’)

.T. (‘igaz’)

.F. (‘hamis’)

36. fejezet

A művelet fogalma

Néhány parancsban az adatokkal műveleteket lehet végezni.

A műveletben részt vevő adatokat operandusoknak nevezzük.

A műveleti jel azt jelzi, hogy az operandusokkal milyen műveletet kell elvégezni.

Az alábbi műveleti jelek léteznek:

+ - * / ** ^ .AND. .OR. .NOT.
< > = <= >= <> # \$

A művelet eredményeként a kívánt új adatot kapjuk.

Példa

Vegyük a szorzási műveletet. Ennek * a műveleti jele.

Ha azt írjuk le, hogy

3*6

akkor egy olyan szorzási művelet áll előttünk, amelyben az első operandus 3, a második pedig 6. A művelet eredménye 18.

Minden műveletnél elő van írva az operandusok típusa, és eleve tudni lehet az eredmény típusát is.

Példa

A szorzás esetében mindkét operandusnak számnak kell lenni, és az eredmény is szám.

A példaként felhozott szorzási műveletben mindkét operandust konstanssal adtuk meg. Az operandus azonban mezőnévvel is megadható (olyan mezőnévvel, amely szerepel a nyitott adatbázisfájlban). A mezőnév azt az adatot jelöli, ami az aktuális rekordban az illető mezőben szerepel.

Példa

Tegyük fel, hogy a nyitott adatbázisfájlban szerepel egy ORA nevű numerikus mező. A

100*ORA

egy olyan szorzási művelet, amelyben az első operandus egy konstans, a második operandus pedig egy mezőnév.

Ha az aktuális rekordban az ORA nevű mező tartalma 4, akkor a művelet eredménye 400.

37. fejezet

A művelet eredményének kiírása

A ? parancs segítségével kiíratható a képernyőre a művelet eredménye.

A ? begépelése után egy szóközt kell tenni, majd be kell gépelni magát a műveletet (vagyis az operandusokat és a műveleti jelet), végül Enter-t kell ütni.

Példák

1. Ha ki akarjuk írni a $2*5$ művelet eredményét, akkor ezt a parancsot adjuk ki:

? 2*5

2. Egy vállalat raktárában levő anyagokról készített adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tíjz.</i>	<i>Jelentés</i>
MEGNEVEZES	C	20	-	Az anyag megnevezése
MENNYISEG	N	9	0	Raktári mennyiség
EGYSEGAR	N	9	0	Egységár

Az adatbázisfájl teljes azonosítója C:\DBASE\ADAT\RAKTAR.DBF

Az adatbázisfájl első három rekordjának a tartalma:

MEGNEVEZES	MENNYISEG	EGYSEGAR
Tégla	10000	12
Csempe	15000	50
Cserép	60000	80

Ki akarjuk írni a téglá, a csempe és a cserép raktári értékét, vagyis az egységárunknak a mennyiségükkel való szorzatát.

A kiíratást ezekkel a parancsokkal végezzük el:

a) Az adatbázisfájl megnyitása:

USE C:\DBASEVADAT\RAKTAR.DBF

b) A téglá raktári értékének kiíratása:

GO 1

? MENNYISEG*EGYSEGAR

c) A csempe raktári értékének kiíratása:

GO 2

? MENNYISEG*EGYSEGAR

d) A cserép raktári értékének kiíratása:

GO 3

? MENNYISEG*EGYSEGAR

Feladatok

1. A ? parancs felhasználásával írasd ki a 65 és a 49 szorzatát!

2. Minden tanuló kap testvérenként 500 Ft szociális juttatást. A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl felhasználásával írasd ki, hogy a 4. és a 7. rekordban szereplő tanuló mennyi juttatásban részesül!

38. fejezet

Számtani műveletek

Hat számtani művelet létezik: az összeadás, a kivonás, a szorzás, az osztás, az előjelváltás és a hatványozás:

A művelet megnevezése	A művelet jele
Összeadás	+
Kivonás	-
Szorzás	*
Osztás	/
Előjelváltás	-
Hatványozás	** vagy ^

Ezek a műveletek számokkal végezhetők el, és eredményül is számot kapunk.

Az operandusok tehát numerikus konstansok vagy numerikus mezőnevek lehetnek.

Az előjelváltásnak egyetlen operandusa van, a többinek pedig kettő.

Példák

1.

a) Összeadás:

65+23 Az eredmény 88

b) Kivonás

34-45 Az eredmény -11

c) Szorzás

9*6 Az eredmény 54

d) Osztás

45/4 Az eredmény 11.25

e) Előjelváltás

-65 Az eredmény mínusz 65

f) Hatványozás

9**2 Az eredmény 9-nek a második hatványa, vagyis 81

2^3 Az eredmény 2-nek a harmadik hatványa, vagyis 8

2. A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A dolgozó neve
ALAPBER	N	6	0	Havi alaphére
POTLEK	N	6	0	Havi bérpótléka

Az aktuális rekord tartalma:

NEV Deák Éva

ALAPBER 20000

POTLEK 2000

Számtani műveletek:

a) ALAPBER*1.2

Az eredmény 24000, mert az aktuális rekordban az ALAPBER tartalma 20000, ez szorozódik meg 1.2-del.

b) ALAPBER+POTLEK

Az eredmény 22000, mert az aktuális rekordban az ALAPBER tartalma 20000, és ehhez adódik hozzá a POLTEK tartalma, vagyis 2000.

39. fejezet

Naptári műveletek

Három olyan művelet van, amely a naptárral kapcsolatos:

A művelet megnevezése	A művelet jele
Dátum növelése	+
Dátum csökkentése	-
Dátumok különbözete	-

Mindhárom műveletnek két operandusa van.

a) Dátum növelése

Ez a művelet egy dátumot egy számmal növel meg, és eredményül az annyi nappal későbbi dátumot adja.

Ebből következik, hogy

- az első operandusnak dátum jellegű mezőnévnek kell lenni
- a második operandus numerikus konstans vagy numerikus mezőnév lehet.

b) Dátum csökkentése

Ez a művelet egy dátumot egy számmal csökkent, és eredményül az annyi nappal korábbi dátumot adja.

Ebből következik, hogy

- az első operandusnak dátum jellegű mezőnévnek kell lenni
- a második operandus numerikus konstans vagy numerikus mezőnév lehet.

c) Dátumok különbözete

Ez a művelet két dátumot von ki egymásból, és eredményül a két dátum között eltelt napok számát adja.

Ebből következik, hogy mind a két operandusnak dátum jellegű mezőnévnek kell lenni.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZULETES	D	8	-	Mikor született?
OVODA	D	8	-	Mikor ment először óvodába?
ISKOLA	D	8	-	Mikor ment először iskolába?

Az aktuális rekord tartalma:

NEV	Nagy Zoltán
SZULETES	1947.09.04
OVODA	1951.09.05
ISKOLA	1954.09.01

a) Dátum növelése:

ISKOLA+7

Az eredmény 1954.09.08, mert az aktuális rekordban az ISKOLA mező tartalma 1954.09.01, és ehhez adódik hozzá 7 nap.

b) Dátum csökkentése:

ISKOLA-3

Az eredmény 1954.08.29, mert az aktuális rekordban az ISKOLA mező tartalma 1954.09.01, és ebből vonódik le 3 nap.

c) Dátumok különbözete:

OVODA-SZULETES

Az eredmény 1462, mert az aktuális rekordban az OVODA mező tartalma 1951.09.05, a SZULETES mező tartalma pedig 1947.09.04, és e két dátum között 1462 nap telt el. Ez azt jelenti, hogy Nagy Zoltán születése és óvodába járása között 1462 nap telt el.

40. fejezet

Összefűző műveletek

Két összefűző művelet létezik:

A művelet megnevezése	A művelet jele
Összefűzés szóközmeghagyással	+
Összefűzés szóközathelyezéssel	-

Ezek a műveletek karaktersorozatokkal végezhetőek el, és eredményül is karaktersorozatot kapunk.

Az operandusok tehát karakteres konstansok vagy karakteres mezőnevek lehetnek. Mindkét műveletnek két operandusa van.

a) Összefűzés szóközmeghagyással

Ennek a műveletnek a hatására a két karaktersorozat oly módon fűződik össze, hogy a bennük szereplő szóközők a helyükön maradnak.

Példa

„Ady ” + „Endre ” Az eredmény „Ady Endre ”

b) Összefűzés szóközathelyezéssel

Ennek a műveletnek a hatására a két karaktersorozat oly módon fűződik össze, hogy az első karaktersorozat végén szereplő szóközők az eredmény legvégére kerülnek.

Példa

„Ady ”-„Endre ” Az eredmény „AdyEndre ”

Összefoglaló példa

A nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Tartalom</i>
VEZNEV	C	12	-	Kocsis
KERNEV	C	12	-	Emese
HAVIDIJ	N	6	0	22000

Összefűző műveletek:

- a) VEZNEV+KERNEV Az eredmény „Kocsis Emese ”
b) VEZNEV-KERNEV Az eredmény „KocsisEmese ”

41. fejezet

Logikai műveletek

Három logikai művelet létezik:

A művelet megnevezése	A művelet jele
ÉS	.AND.
VAGY	.OR.
TAGADÁS	.NOT.

Ezek a műveletek logikai adatokkal végezhetőek el, és eredményül is logikai adatot kapunk.

Az operandusok tehát logikai konstansok vagy logikai mezőnevek lehetnek.

A TAGADÁS műveletnek egyetlen operandusa van, a többinek pedig kettő.

a) Az ÉS művelet

Az ÉS művelet eredménye csak akkor 'igaz', ha mindkét operandus értéke 'igaz'.

Példák

- 1) .T. .AND. .T. Az eredmény 'igaz'
- 2) .T. .AND. .F. Az eredmény 'hamis'
- 3) .F. .AND. .T. Az eredmény 'hamis'
- 4) .F. .AND. .F. Az eredmény 'hamis'

b) A VAGY művelet

A VAGY művelet eredménye csak akkor 'hamis', ha mindkét operandus értéke 'hamis'.

Példák

- 1) .F. .OR. .F. Az eredmény 'hamis'
- 2) .T. .OR. .T. Az eredmény 'igaz'
- 3) .T. .OR. .F. Az eredmény 'igaz'
- 4) .F. .OR. .T. Az eredmény 'igaz'

c) A TAGADÁS művelet

A TAGADÁS művelet eredménye akkor **'hamis'**, ha az operandus értéke **'igaz'**; és akkor **'igaz'**, ha az operandus értéke **'hamis'**.

Példák

- 1) .NOT. .T. Az eredmény **'hamis'**
- 2) .NOT. .F. Az eredmény **'igaz'**

Összefoglaló példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
KOLLEGISTA	L	1	-	Kollégiumban lakik-e?
SPORTOLO	L	1	-	Sportol-e?
ENEKKAROS	L	1	-	Énekkaros-e?
MENZAS	L	1	-	Menzás-e?

Az aktuális rekord tartalma:

NEV	Tóth Benedek
KOLLEGISTA	N
SPORTOLO	Y
ENEKKAROS	N
MENZAS	Y

Logikai műveletek:

- a) KOLLEGISTA .AND. SPORTOLO

Az eredmény **'hamis'**, mert a az aktuális rekordban a KOLLEGISTA mező tartalma **'igaz'**, a SPORTOLO mező tartalma pedig **'hamis'**. (A művelet eredménye azoknál a tanulóknál **'igaz'**, akik kollégiumban laknak, és sportolnak is.)

- b) KOLLEGISTA .OR. MENZAS

Ez eredmény **'igaz'**, mert az aktuális rekordban a KOLLEGISTA mező tartalma **'hamis'**, a MENZAS mező tartalma pedig **'igaz'**. (A művelet eredménye azoknál a tanulóknál **'igaz'**, akik kollégiumban laknak vagy menzások.)

c) .NOT. ENEKKAROS

Az eredmény 'igaz', mert az aktuális rekordban az ENEKKAROS mező tartalma 'hamis'. (A művelet eredménye azoknál a tanulóknál igaz, akik nem énekkarosok.)

42. fejezet

Összehasonlító műveletek

Hat összehasonlító művelet létezik:

A művelet megnevezése	A művelet jele
Kisebb	<
Nagyobb	>
Egyenlő	=
Kisebb vagy egyenlő	<=
Nagyobb vagy egyenlő	>=
Nem egyenlő	#

Ezek a műveletek két azonos típusú adatot hasonlítanak össze, és az eredmény egy olyan logikai adat, amely az összehasonlítás eredményét tükrözi.

Egyetlen megkötés van: az összehasonlítandó adatok nem lehetnek logikaiak.

Ebből következik, hogy valamennyi műveletnek két operandusa van, és ezek lehetnek numerikusok, dátum jellegűek vagy karakteresek, de azonos típusúaknak kell lenniük.

a) Számok összehasonlításakor a számtanban megszokott eredményt kapjuk.

b) Karaktorsorozatok összehasonlításakor az eredmény a következő szabályok szerint alakul ki:

— Két karakter közül az a kisebb, amelynek kisebb a belső kódja.

— Két karaktorsorozat összehasonlítása balról jobbra történik oly módon, hogy ha az első karaktereik egyenlőek, akkor a második karakterek kerülnek összehasonlításra, s ha azok is megegyeznek, akkor a harmadikok következnek, és így tovább. Mihelyt valahányadik karakterük eltérő, kialakul a megfelelő eredmény. A karaktorsorozatok végén szereplő szóközök azonban nem befolyásolják az eredményt.

Két karaktorsorozat tehát csak akkor egyenlő, ha a végükön levő szóközöktől eltekintve minden karakterük megegyezik.

c) Dátumok összehasonlításakor a korábbi dátum a kisebb.

Példák

1. Számok összehasonlítása:

- a) $8 > -12$ Az eredmény **'igaz'**
b) $9 < 9$ Az eredmény **'hamis'**

2. Karaktorsorozatok összehasonlítása:

- a) "ANDAY" > "ANDAI"

Az eredmény **'igaz'**, mert a karaktorsorozatok első négy karaktere egyenlő, az ötödik karakterek közül pedig az „Y”-nak nagyobb a belső kódja, mint az „I”-nek.

- b) "GYÖRGYI" < "GYÖNGYI"

Az eredmény **'hamis'**, mert az „R” belső kódja nagyobb, mint az „N” belső kódja.

3. A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
NEV	C	25	-	A dolgozó neve
SZUL_HELY	C	20	-	Születési helye
ALAPBER	N	6	0	Havi alapbére
POTLEK	N	6	0	Havi bérpótléka
NAGYKORU	D	8	-	Mikor lett nagykorú?
MUNKA	D	8	-	Mikor kezdett el dolgozni?

Az aktuális rekord tartalma:

NEV	Kiss Leizer Anna
SZUL_HELY	Székesfehérvár
ALAPBER	25000
POTLEK	1500
NAGYKORU	1985.03.21
MUNKA	1984.05.02

Összehasonlító műveletek:

a) ALAPBER \geq 20000

A művelet eredménye **'igaz'**, mert az aktuális rekordban az ALAPBER mező tartalma 25000, és ez nagyobb 20000-nél. (A művelet eredménye azoknál a dolgozóknál igaz, akiknek a havi alapbére nagyobb vagy egyenlő, mint 20000 Ft.)

b) POTLEK = 2000

A művelet eredménye **'hamis'**, mert az aktuális rekordban a POTLEK mező tartalma 1500, és ez nem egyenlő 2000-rel. (A művelet eredménye azoknál a dolgozóknál **'igaz'**, akiknek a havi bérpótléka 2000 Ft.)

c) SZUL_HELY = 'Székesfehérvár'

A művelet eredménye **'igaz'**, mert az aktuális rekordban a SZUL_HELY mező tartalma „Székesfehérvár”, és a végén levő szóközök nem számítanak. (A művelet eredménye azoknál a dolgozóknál igaz, akik Székesfehérváron születtek.)

d) NEV = 'Kis Leizer Anna'

A művelet eredménye **'hamis'**, mert az aktuális rekordban a NEV mező tartalma „Kiss Leizer Anna”, a vezetéknévben a Kiss két s-sel van írva.

e) MUNKA < NAGYKORU

A művelet eredménye **'igaz'**, mert az aktuális rekordban a MUNKA mező korábbi dátumot tartalmaz, mint a NAGYKORU mező. (A művelet eredménye azoknál a dolgozóknál **'igaz'**, akik nagykorúságuk elérése előtt kezdtek el dolgozni.)

43. fejezet

Karakterkereső művelet

Egyetlen karakterkereső művelet létezik:

A művelet megnevezése	A művelet jele
Bennefoglaltatik	\$

Ebben a műveletben két karaktersorozat vesz részt, és eredményül logikai adatot kapunk.

Ebből következik, hogy a műveletnek két operandusa van, és mindkettő karakteres konstans vagy karakteres mezőnév lehet.

A művelet eredménye csak akkor **'igaz'**, ha az első karaktersorozat bennefoglaltatik a második karaktersorozatban.

Példák

1.

a) „Judith” \$ „Szabó Judit”

Az eredmény **‘hamis’**, mert a „Judith” nem foglaltatik benne a „Szabó Judit”-ban.

b) „Judit” \$ „Szabó Judit”

Az eredmény **‘igaz’**, mert a „Judit” bennefoglaltatik a „Szabó Judit”-ban.

2. A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
LAKCIM	C	40	-	Lakcíme
SZUL_DATUM	D	8	-	Születési dátuma

Karakterkereső műveletek:

a) ‘Erika’ \$ NEV

Ennek a műveletnek az eredménye az Erika nevű tanulóknál **‘igaz’**.

b) ‘Szombathely’ \$ SZUL_HELY

Ennek a műveletnek az eredménye a szombathelyi születésű tanulóknál **‘igaz’**.

c) ‘Budapest’ \$ LAKCIM

Ennek a műveletnek az eredménye a Budapesten lakó tanulóknál — illetve azoknál, akiknél az utca nevében szerepel a Budapest — **‘igaz’**.

44. fejezet A kifejezés fogalma

A kifejezés nem más, mint

— konstansoknak

— mezőneveknek

— műveleti jeleknek

olyan összekapcsolása, aminek van értelme.

A kifejezésben ezen kívül gömbölyű zárójeleket is használhatunk. A gömbölyű zárójelpárok egymásba helyezhetők.

Fogadjuk el, hogy

- egy önmagában álló konstans is kifejezés
- egy önmagában álló mezőnév is kifejezés.

Példák

1. Ha a nyitott adatbázisfájlban a TESTVER egy numerikus mező neve, a SZUL_HELY pedig egy karakteres mező neve, akkor

TESTVER+SZUL_HELY

nem kifejezés, mert értelmetlen dolog numerikus és karakteres mezőnév közé + jelet tenni.

2. Ha a TESTVER a nyitott adatbázisfájl egy numerikus mezőjének a neve, akkor az alábbiak mindegyike egy kifejezés:

a) TESTVER

Ez a kifejezés mindössze egy mezőnévből áll.

b) 1000

Ez a kifejezés mindössze egy konstansból áll.

c) TESTVER*1000

Ez a kifejezés egy mezőnévből, egy műveleti jelből és egy konstansból áll.

d) (TESTVER*1000)/2

Ez a kifejezés zárójelet is tartalmaz.

e) ((TESTVER*1000)/2+1)*3

Ebben a kifejezésben egy zárójelpár a másikba van helyezve.

45. fejezet

A kifejezés értékének kialakulása

A kifejezés értékének kialakulását az úgynevezett szintszámok és a zárójelek határozzák meg.

Minden művelet rendelkezik szintszámmal.

Az alábbi táblázatban az összes művelet szintszáma fel van tüntetve:

A művelet megnevezése	A művelet jele	Szintszáma
Hatványozás	** vagy ^	1
Összefűzés szóközmeghagyással	+	1
Összefűzés szóközát helyezéssel	-	1
Bennefoglaltatás	\$	1
Előjelváltás	-	2
Szorzás	*	3
Osztás	/	3
Összeadás	+	4
Kivonás	-	4
Dátum növelése	+	4
Dátum csökkentése	-	4
Dátumok különbsége	-	4
Kisebbség	<	5
Nagyobb	>	5
Egyenlőség	=	5
Kisebbség vagy egyenlőség	<=	5
Nagyobb vagy egyenlőség	>=	5
Nem egyenlőség	#	5
TAGADÁS	.NOT.	6
ÉS	.AND.	7
VAGY	.OR.	8

Elsőként az 1-es szintszámú műveletek hajtódnak végre, ezt követik a 2-es szintszámúak, és így tovább. Az egyforma szintszámmal rendelkező műveletek balról jobbra hajtódnak végre.

Ezt a végrehajtási sorrendet zárójelek alkalmazásával módosítani lehet. A zárójelben levő műveletek ugyanis elsőbbséget élveznek. Ha a zárójelben több művelet van, akkor azok végrehajtási sorrendjét ugyancsak a szintszámok határozzák meg.

Példa

a) $11+100*9 \leq 920$.AND. $6^2 = 36$

Legelőször végrehajtódik a 6^2 művelet, mert a hatványozásnak 1-es a szintszáma. Az eredmény 36.

Ezt követi a $100*9$ művelet, mert a szorzásnak 3-as a szintszáma. Az eredmény 900.

Ezután következik a $11+900$ művelet, mert az összeadásnak 4-es a szintszáma. Az eredmény 911.

Most jönnek (balról jobbra haladva) a $911 \leq 920$ és a $36 = 36$ műveletek, mert ezeknek a szintszáma 5-ös. Mindkettőnek az eredménye **'igaz'**.

Végül következik az .AND. művelet, mert annak a szintszáma 7.

A végeredmény **'igaz'**.

b) $(11+100)*9 \leq 920$.AND. $6^2 = 36$

Legelőbb a zárójelben levő művelet hajtódik végre. Ennek eredménye 111.

Ezután végrehajtódik a 6^2 művelet, aminek az eredménye 36.

Ezt követi a $111*9$ művelet, ennek eredménye 999.

Most jönnek (balról jobbra haladva) a $999 \leq 920$ és a $36 = 36$ műveletek. Az egyiknek az eredménye **'hamis'**, a másiké pedig **'igaz'**.

Végül következik az .AND. művelet.

A végeredmény **'hamis'**.

46. fejezet

A kifejezés értékének kiírása

A ? parancs segítségével nem csupán egy művelet eredménye, hanem tetszőleges kifejezés értéke is kiíratható.

A ? begépelése után egy szóközt kell tenni, majd be kell gépelni magát a kifejezést, végül Enter-t kell ütni.

Példa

Ha ki akarjuk írni az első öt négyzetszám összegét, akkor ezt a parancsot adjuk ki:

? $1^2+2^2+3^2+4^2+5^2$

47. fejezet

Numerikus kifejezések

Az olyan kifejezést, amelynek az értéke szám, numerikus kifejezésnek nevezzük.

Példa

Numerikus kifejezések:

- a) $((6+8)/2+1)^2$ A kifejezés értéke 64
 b) $(2^2+4^2+6^2)/4$ A kifejezés értéke 14

Feladatok

1. Számítsd ki fejből az alábbi kifejezések értékét:

- a) $(7*8-7*7)^2$
 b) $((10+9)*(10-3^2))/2$

A ? parancs felhasználásával nézd meg, hogy jól számoltál-e?

2. Tegyük fel, hogy a nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma a következő:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Tartalom</i>
NEV	C	25	-	Kovács Annamária
TARTOZAS	N	8	0	2500
KOVETELES	N	8	0	100
KAMATLAB	N	5	2	30.00
ENGEDMENY	N	8	0	300

Számítsd ki számítógép használata nélkül a következő kifejezések értékét:

- a) $TARTOZAS*(1+KAMATLAB/100)^2$
 b) $(TARTOZAS-KOVETELES)*(1+KAMATLAB/100)^2-ENGEDMENY$
 c) $(TARTOZAS/2-ENGEDMENY)*(1+KAMATLAB/100)**2$

A kapott eredményeket ellenőrizd a számítógéppel!

Megoldás

Az ellenőrzés érdekében a következőket kell tenned:

Mindenekelőtt hozd létre a fenti szerkezettel rendelkező adatbázisfájlt! (Az adatbázisfájlt a gyakorlólemez KM katalógusába tedd, neve pedig ADOS legyen!) Az adatbázisfájlbba vidd be Kovács Annamária rekordját a fenti tartalommal!

Ezután gondoskodj róla, hogy Kovács Annamária rekordja legyen az aktuális rekord!

Végül a ? parancs alkalmazásával sorban egymás után határozd meg a kifejezések értékét!

48. fejezet

Dátum jellegű kifejezések

Az olyan kifejezést, amelynek az értéke dátum, dátum jellegű kifejezésnek nevezzük

Feladat

Tegyük fel, hogy a nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tízs.</i>	<i>Tartalom</i>
NEV	C	25	-	Csikai Klára
ISK_KEZDET	D	8	-	1956.09.01
ISK_VEG	D	8	-	1974.06.12

Határozd meg számítógép használata nélkül a következő kifejezések értékét:

- ISK_KEZDET-1
- ISK_VEG+30
- ISK_VEG-ISK_KEZDET

Ellenőrizd magadat a számítógéppel! (Az ellenőrzéshez szükséges adatbázisfájlt a gyakorlólemez KM katalógusába tedd, és a neve DATUM legyen!)

49. fejezet

Karakteres kifejezések

Az olyan kifejezést, amelynek az értéke karaktersorozat, karakteres kifejezésnek nevezzük.

Példa

A nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Tartalom</i>
NEV	C	24	-	Kada Erika
VAROS	C	20	-	Kecskemét
UTCA	C	20	-	Matkói
HAZSZAM	C	6	-	18.

Karakteres kifejezések:

a) `UTCA+"út"+ " "+HAZSZAM`

A kifejezés értéke „Matkói út 18. ”

b) `(UTCA-"út")-(" "+HAZSZAM)`

A kifejezés értéke „Matkói út 18. ”

Feladat

Tegyük fel, hogy a nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Tartalom</i>
NEV	C	25	-	Kovács Gabriella
ANYJA_NEVE	C	25	-	Kis Gabriella
CIM	C	40	-	Baja Fő út 1.

Határozd meg számítógép használata nélkül a következő kifejezések értékét:

a) `"A tanuló neve: "+NEV`

b) `"A tanuló adatai: "+NEV-ANYJA_NEVE-CIM`

Ellenőrizd magadat a számítógéppel! (Az ellenőrzéshez szükséges adatbázisfájlt a gyakorlólemez KM katalógusába tedd, és a neve SZEMELY legyen!)

50. fejezet

Logikai kifejezések

Az olyan kifejezést, amelynek az értéke logikai adat, logikai kifejezésnek nevezzük.

Példák

1. Logikai kifejezések:

a) $\text{NOT}(\text{T} \text{ OR } \text{F})$

A kifejezés értéke **'hamis'**.

b) $(\text{T} \text{ OR } \text{F}) \text{ AND } (\text{T} \text{ AND } \text{F})$

A kifejezés értéke **'hamis'**.

c) $\text{NOT}(\text{T} \text{ OR } \text{F})$

A kifejezés értéke **'hamis'**.

d) $5 < 6 \text{ AND } \text{"KATI"} < \text{"Kati"}$

A kifejezés értéke **'igaz'**, mert az is igaz, hogy $5 < 6$, és az is igaz, hogy $\text{"KATI"} < \text{"Kati"}$.

e) $4 \geq 6 \text{ OR } 8 = 9$

A kifejezés értéke **'hamis'**, mert az is **'hamis'**, hogy $4 \geq 6$, és az is **'hamis'**, hogy $8 = 9$.

2. Egy vállalat dolgozóiról készített adatbázisfájl szerkezetét és a mezők jelentését az alábbi táblázat szemlélteti:

Mezőnév	Típus	Hossz	Tj.sz.	Jelentés
NEV	C	20	-	A dolgozó neve
SZUL_DATUM	D	8	-	Születési dátuma
MUNKA_KEZD	D	8	-	Munkaviszonyának kezdete
ALAPBER	N	6	0	Havi alapbére
POTLEK	N	6	0	Havi bérpótléka
LEVONAS	N	6	0	Havi levonása
EGYETEM	L	1	-	Végzett-e egyetemet?
VEZETO	L	1	-	Vezető beosztásban dolgozik-e?

Az adatbázisfájl tartalma:

Nev	Szul_Datum	Munka_Kezd	Alapber	Potlek	Levonas	Egyetem	Vezeto
Kiss Gyula	1954.03.30	1969.09.01	55000	0	0	Y	Y
Herczeg András	1938.01.22	1962.09.03	45000	2000	500	Y	Y
Fekete Béla	1948.08.23	1973.09.03	43000	4000	2500	Y	Y
Sólyom Melinda	1965.04.05	1983.01.03	47000	500	0	N	N
Kis András	1965.04.12	1987.03.16	20000	1000	600	N	Y
Babai Attila	1963.06.03	1987.05.15	20000	1000	0	Y	Y
Módos Andor	1956.12.07	1974.11.01	18000	1000	0	N	N
Falus Róbert	1965.09.17	1983.10.31	18000	1000	0	N	N
Sutus Csabáné	1967.07.25	1987.04.30	16000	500	0	N	N
Kertai Sándor	1966.11.20	1964.06.01	16000	500	1000	N	N
Kun András	1965.06.01	1983.12.01	15000	500	0	N	N
Rimóczi Renáta	1975.07.07	1994.10.03	15000	0	0	N	N
Csabai Antal	1956.02.08	1985.08.15	15000	0	0	N	N
Kada Erika	1966.01.18	1984.06.15	14000	0	0	N	N
Pfliegel Péter	1966.05.12	1985.05.01	14000	0	0	N	N

Logikai kifejezések:

a) ALAPBER \geq 45000

A kifejezés értéke azoknál a dolgozóknál **'igaz'**, akiknek a havi alapbére legalább 45000 Ft. (Három ilyen dolgozó van: Kiss Gyula, Herczeg András és Sólyom Melinda.)

b) EGYETEM

A kifejezés értéke azoknál a dolgozóknál **'igaz'**, akik egyetemet végeztek. (Négy ilyen dolgozó van: Kiss Gyula, Herczeg András, Fekete Béla és Babai Attila.)

Megjegyzés

Bár az EGYETEM = .T. is logikus lenne, de ez a forma nem használható. Ezzel ugyanis azt feszegetnénk, hogy igaz-e az, hogy a „van egyetemi végzettsége” **'igaz'**. Mivel ezzel teljesen egyenértékű a rövidebb, az, hogy „van egyetemi végzettsége”, így az előző forma felesleges, ezért tilos.

c) EGYETEM .AND. ALAPBER >= 45000

A kifejezés értéke azoknál ez egyetemet végzett dolgozóknál 'igaz', akiknek a havi munkabére legalább 45000 Ft. (Két ilyen dolgozó van: Kiss Gyula és Herczeg András.)

d) VEZETO .AND. LEVONAS > 2000

A kifejezés értéke azoknál a vezető beosztású dolgozóknál 'igaz', akiknél a havi levonás meghaladja a 2000 Ft-ot. (Egy ilyen dolgozó van: Fekete Béla.)

e) EGYETEM .AND. ALAPBER+POTLEK-LEVONAS < 22000

A kifejezés értéke azoknál az egyetemet végzett dolgozóknál 'igaz', akiknek a havi járandósága 22000 Ft-nál kevesebb. (Egy ilyen dolgozó van: Babai Attila.)

f) .NOT. EGYETEM .AND ALAPBER > 30000

A kifejezés értéke azoknál az egyetemet nem végzett dolgozóknál 'igaz', akiknek a havi alapbére meghaladja a 30000 Ft-ot. (Egy ilyen dolgozó van: Sólyom Melinda.)

g) .NOT. VEZETO .AND. MUNKA_KEZD-SZUL_DATUM > 23*365

A kifejezés értéke azoknál a nem vezető beosztású dolgozóknál 'igaz', akik már betöltötték a 23. életévüket akkor, amikor munkába álltak. (Egy ilyen dolgozó van: Csabai Antal.)

Feladatok

1. Határozd meg számítógép használata nélkül az alábbi kifejezések értékét:

- a) .NOT.(.F. .AND. .F.)
- b) (.T. .OR. .F.) .OR. (.NOT.(.NOT. .T.))
- c) 3*4 < 10
- d) 3*4 <= 12
- e) 3*4 < 10 .AND. 3*4 <= 12
- f) "Eva" < "EVA" .OR. "ALMA" # "alma"
- g) .NOT. "ABC" < "ABc"

A ? parancs felhasználásával ellenőrizd magadat!

2. Tegyük fel, hogy a nyitott adatbázisfájl szerkezete és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Tartalom</i>
NEV	C	25	-	Kovács Szilvia
FIU	L	1	-	N
OKOS	L	1	-	Y
SZORGALMAS	L	1	-	N
OPTIMISTA	L	1	-	Y

Határozd meg számítógép használata nélkül a következő kifejezések értékét:

- .NOT. FIU .AND. OKOS
- .NOT. (FIU .AND. OKOS)
- OKOS .OR. .NOT. SZORGALMAS
- .NOT. (OPTIMISTA .OR. OKOS)

Ellenőrizd magadat a számítógéppel! (Az ellenőrzéshez szükséges adatbázisfájlt a gyakorlólemez KM katalógusába tedd, és a neve JELLEM legyen!)

3. Tegyük fel, hogy a nyitott adatbázisfájl szerkezete és a mezők jelentése :

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
ATLAG	N	4	2	Tanulmányi átlaga
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

Mondd el, hogy mely tanulók esetében 'igaz' az alábbi kifejezések értéke:

- TESTVER ≥ 2 .AND. ATLAG > 3.5
- .NOT. KOLLEGIUM .OR. TESTVER = 0
- KOLLEGIUM .AND. ATLAG > 4

4. Egy uzsorás az adósairól olyan adatbázisfájlt készített, melynek szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	Az adós neve
LAKOHELY	C	20	-	Lakóhelye
ADOSSAG	N	6	0	Adóssága
JOVEDELEM	N	6	0	Havi jövedelme
SZOKOTTE	L	1	-	Szokott-e rendszeren fizetni?

Mondd el, hogy mely adósok esetében 'igaz' az alábbi kifejezések értéke:

- ADOSSAG ≥ 10000 .AND. ADOSSAG ≤ 20000
- ADOSSAG > 50000 .AND. .NOT. SZOKOTTE
- ADOSSAG/JOVEDELEM > 2 .AND. "Budapest" \$ LAKOHELY

51. fejezet

A hatáskör fogalma

Emlékezzünk vissza a DISPLAY, a DELETE, a RECALL, a LIST, az EDIT és a CHANGE parancsokra.

Láttuk, hogy ha másképpen nem rendelkezünk, akkor a DISPLAY, a DELETE és a RECALL parancs a nyitott adatbázisfájl aktuális rekordjára vonatkozik (a DISPLAY kiírja a tartalmát, a DELETE törlésre jelöli ki őt, a RECALL pedig megszünteti a kijelöltségét).

Láttuk azt is, hogy ha másképpen nem rendelkezünk, akkor a LIST, az EDIT és a CHANGE parancs a nyitott adatbázisfájl valamennyi rekordjára vonatkozik (a LIST valamennyinek a tartalmát kiírja, az EDIT és a CHANGE paranccsal pedig valamennyinek a tartalma módosítható).

A hatáskör azon rekordok összességét jelenti, amelyekre a parancs vonatkozik.

Elmondhatjuk tehát, hogy ha másképpen nem rendelkezünk, akkor

— a DISPLAY, a DELETE és a RECALL parancs hatáskörébe csupán az aktuális rekord tartozik bele

— a LIST, az EDIT és a CHANGE parancs hatáskörébe valamennyi rekord beletartozik.

Felmerül a kérdés: hogyan szabályozhatjuk mi magunk az egyes parancsok hatáskörét. Például hogyan írhatjuk elő a DISPLAY parancs számára azt, hogy a hatáskörébe ne csupán az aktuális rekord tartozzon bele, hanem mondjuk az aktuális rekordtól kezdve 12 rekord. Vagy hogyan írhatjuk elő a LIST parancs számára, hogy a hatáskörébe mondjuk csak a 6. rekord tartozzon bele.

52. fejezet

A hatáskör megadása

A hatáskör megadása négyféleképpen lehetséges:

a) RECORD

A parancsba kiegészítésképpen ezt gépeljük be:

RECORD p

(itt **p** egy pozitív egész értékkel rendelkező numerikus konstanst jelöl). Ilyenkor a parancs hatáskörébe egyedül a **p**-edik rekord tartozik bele.

Példák

1. Ha kiadjuk a

LIST RECORD 5

parancsot, akkor a parancs hatáskörébe az 5. rekord tartozik bele, tehát csupán az 5. rekord tartalma fog a képernyőn megjelenni.

2. Ha kiadjuk a

DELETE RECORD 7

parancsot, akkor a parancs hatáskörébe a 7. rekord tartozik bele, tehát a 7. rekord lesz törlésre kijelölve (függetlenül attól, hogy melyik az aktuális rekord).

b) NEXT

A parancsba kiegészítésként ezt gépeljük be:

NEXT p

(itt **p** egy pozitív egész értékkel rendelkező numerikus konstanst jelöl). Ilyenkor a parancs hatáskörébe **p** darab rekord tartozik bele: az aktuális rekord és az azt követő **p-1** darab rekord.

Példák

1. Ha a nyitott adatbázisfájlban jelenleg a 7. rekord az aktuális, és kiadjuk a

RECALL NEXT 3

parancsot, akkor a parancs hatáskörébe a 7. rekordtól kezdve három rekord tartozik bele, tehát a 7., a 8. és a 9. rekord kijelöltsége fog megszűnni.

2. Ha a nyitott adatbázisfájlban jelenleg a 6. rekord az aktuális, és kiadjuk a

LIST NEXT 4

parancsot, akkor a parancs hatáskörébe a 6. rekordtól kezdve négy rekord tartozik bele, tehát csupán a 6., a 7. a 8. és a 9. rekord tartalma íródik ki.

c) ALL

A parancsba kiegészítésként ezt gépeljük be:

ALL

Ilyenkor a parancs hatáskörébe az összes rekord beletartozik.

Példák

1. Ha kiadjuk a

DISPLAY ALL

parancsot, akkor a parancs hatáskörébe valamennyi rekord beletartozik, tehát valamennyinek a tartalma ki lesz írva a képernyőre.

2. Ha kiadjuk a
DELETE ALL

parancsot, akkor a parancs hatáskörébe valamennyi rekord beletartozik, tehát valamennyi ki lesz jelölve törlésre.

d) REST

A parancsba kiegészítésképpen ezt gépeljük be:

REST

Ilyenkor a parancs hatáskörébe az aktuális rekord és az azt követő összes rekord beletartozik.

Példák

1. Tegyük fel, hogy a nyitott adatbázisfájlban 21 rekord van, és jelenleg a 18. rekord az aktuális. Ha kiadjuk a

LIST REST

parancsot, akkor a parancs hatáskörébe a 18., a 19., a 20. és a 21. rekord tartozik bele, tehát ezeknek a tartalma lesz kiírva a képernyőre.

2. Tegyük fel, hogy a nyitott adatbázisfájlban 110 rekord van, és jelenleg a 108. rekord az aktuális. Ha kiadjuk a

RECALL REST

parancsot, akkor a parancs hatáskörébe a 108., a 109. és a 110. rekord tartozik bele, tehát ezeknek a kijelöltsége fog megszűnni.

Feladatok

1. Tegyük fel, hogy a nyitott adatbázisfájl összesen 11 rekordból áll, és közülük jelenleg a 6. rekord az aktuális. Mi fog történni a következő parancs hatására:

a)

DELETE NEXT 4

b)

DISPLAY NEXT 3

c)

RECALL ALL

d)

EDIT RECORD 8

e)

LIST REST

f)

CHANGE RECORD 3

2. A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájlban végezd el az alábbi teendőket:

- a) Írasd ki a 3. rekordot követő rekordok tartalmát!
- b) Jelöld ki törlésre a 7. rekordot és az azt követő rekordokat!
- c) Növeld meg a 4., az 5., a 6. és a 7. rekordban szereplő könyv árát egységesen 10 Ft-tal!
- d) Szüntesd meg az összes rekord kijelöltségét!

Megjegyzés

Sok olyan parancs van, amelyben nincs értelme (és ezért nem is lehet) hatáskört megadni. Az eddig tanult parancsok közül ilyen a CLEAR, a QUIT, a CREATE, a DIR, az USE, a DISPLAY STRUCTURE, a LIST STRUCTURE, a GO, a SKIP, az APPEND, az INSERT, a BROWSE, a PACK és a ZAP.

53. fejezet

A hatáskör alapértelmezése

Ha egy parancsban megadható ugyan hatáskör, de mégsem adjuk meg, akkor az úgynevezett alapértelmezés lép érvénybe.

A DISPLAY, a DELETE és a RECALL parancsnál a hatáskör alapértelmezése az aktuális rekord.

A LIST, az EDIT és a CHANGE parancsnál a hatáskör alapértelmezése az összes rekord.

Példák

1. Ha a nyitott adatbázisfájlban jelenleg a 12. rekord az aktuális, akkor a
DISPLAY RECORD 12

és a

DISPLAY

parancsok ugyanazt jelentik, mivel a DISPLAY parancsban a hatáskör alapértelmezése az aktuális rekord.

2. Mivel a LIST parancsban a hatáskör alapértelmezése ALL, ezért a
LIST ALL

és a

LIST

parancsok ugyanazt jelentik.

54. fejezet

A feltétel fogalma

Bizonyos parancsokba bele lehet gépelni a FOR szót és ezt követően egy logikai kifejezést. Ilyenkor a parancs a hatáskörébe tartozó rekordok közül is csak azokra a rekordokra vonatkozik, amelyeknél a logikai kifejezés értéke 'igaz'.

A FOR szót követő logikai kifejezést feltételnek nevezzük.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
TESTVER	N	2	0	Testvéreinek száma
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

a) Ha egy parancsba belegépeljük azt, hogy

```
FOR TESTVER > 3
```

akkor a parancs hatása azokra a rekordokra szűkül, amelyekben a TESTVER mező tartalma nagyobb 3-nál, vagyis azokra a tanulókra, akiknek 3-nál több testvérük van.

b) Ha egy parancsba belegépeljük azt, hogy

```
FOR KOLLEGIUM
```

akkor a parancs hatása azokra a rekordokra szűkül, amelyekben a KOLLEGIUM mező tartalma 'igaz', vagyis azokra a tanulókra, akik kollégiumban laknak.

c) Ha egy parancsba belegépeljük azt, hogy

```
FOR TESTVER > 3 .AND. KOLLEGIUM
```

akkor a parancs hatása azokra a rekordokra szűkül, amelyekben a TESTVER mező tartalma nagyobb 3-nál, és a KOLLEGIUM mező tartalma 'igaz'; vagyis azokra a kollégiumban lakó tanulókra, akiknek 3-nál több testvérük van.

Megjegyzés

Sok olyan parancs van, amelyben nincs értelme (és ezért nem is lehet) feltételt megadni.

Vannak olyan parancsok, amelyeknek az alapértelmezett hatásköre az aktuális rekord. Ilyen például a DISPLAY parancs. Ezeknél a parancsoknál a feltétel megadása az alapértelmezett hatáskört ALL-ra változtatja.

55. fejezet

Hatáskör és feltétel együttes alkalmazása

Tudjuk, hogy ha egy parancsba hatáskört és feltételt is belegépelünk, akkor a parancs csak azokra a rekordokra vonatkozik, amelyek a hatáskörbe is beletartoznak, meg a feltétel is teljesül rájuk.

Az eddig tanult parancsok között hat olyan van, amelybe hatáskör és feltétel egyaránt belegépelhető. Ezek a következők: DISPLAY, LIST, EDIT, CHANGE, DELETE, RECALL.

Példa

Egy autókereskedő az eladásra kínált autókról adatbázisfájlt készített. Az adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Típus</i>	<i>Jelentés</i>
GYARTMANY	C	12	-	Az autó gyártmánya
TIPUS	C	12	-	Típusa
BENZIN	L	1	-	Benzinüzemű-e?
HENGER	N	4	0	A hengerek lökettérfogata (cm ³)
TARTALY	N	2	0	Az üzemanyagtartály térfogata (l)
TÖMEG	N	5	0	Tömege (kg)
SEBESSEG	N	3	0	Legnagyobb sebessége (km/óra)
FOGYASZTAS	N	4	1	Üzemanyagfogyasztása 100 km-en (l)

Az alábbiakban felírunk néhány parancsot, és elmondjuk, hogy mely rekordokra vonatkozik.

a)

EDIT ALL FOR .NOT. BENZIN .AND. SEBESSEG < 180

A parancs az összes olyan nem benzinüzemű autóra vonatkozik, amelynek a legnagyobb sebessége nem éri el a 180 km/óra-t. (Azoknak az adatai módosíthatók.)

b)

LIST REST FOR TARTALY > 40 .AND. HENGER > 1300

A parancs az aktuális rekordtól kezdve azokra az autókra vonatkozik, amelyeknek a tartályába 40 liternél több üzemanyag fér el, és a hengerük lökettérfogata meghaladja az 1300 köbcentimétert. (Azoknak az adatait írja ki.)

c)

DISPLAY NEXT 25 FOR "Skoda" \$ GYARTMANY .AND. TOMEG > 850

A parancs az aktuális rekordban és az azt követő 24 rekordban elhelyezkedő olyan Skoda gyártmányú autókra vonatkozik, amelyeknek a tömege meghaladja a 850 kg-ot. (Azoknak az adatait írja ki.)

d)

DISPLAY FOR "Pick Up" \$ TIPUS

A FOR miatt az alapértelmezett hatáskör az ALL, ezért a parancs az összes Pick Up típusú autóra vonatkozik. (Azoknak az adatai íródnak ki.)

56. fejezet

A kifejezéslista fogalma

Kifejezéslistát úgy kapunk, hogy tetszőleges típusú kifejezéseket felsorolunk, és egymástól vesszővel választjuk el őket.

Fogadjuk el, hogy egyetlen kifejezés is kifejezéslistát alkot.

A ? parancs segítségével egy kifejezéslista értékeit is kiírathatjuk.

Példa

A nyitott adatbázisfájl szerkezete:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tíjz.</i>
NEV	C	25	-
SZUL_HELY	C	20	-
SZUL_DATUM	D	8	-
TESTVER	N	2	0
KOLLEG	L	1	-
SPORTOL	L	1	-

Kifejezéslista például a következő:

NEV+SZUL_HELY, TESTVER*10, KOLLEG .AND. SPORTOL, SZUL_DATUM+1

Ez a kifejezéslista négy kifejezésből áll.

Az első egy karakteres kifejezés:

NEV+SZUL_HELY

A második egy numerikus kifejezés:

TESTVER*10

A harmadik egy logikai kifejezés:
KOLLEG .AND. SPORTOL

A negyedik egy dátum jellegű kifejezés:
SZUL_DATUM+1

A fenti kifejezések értékeit ezzel a paranccsal íratjuk ki:

? NEV+SZUL_HELY, TESTVER*10, KOLLEG .AND. SPORTOL, SZUL_DATUM+1

Feladat

Nyisd meg a gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájlt!

Írasd ki a 4., majd a 7. rekordból az alábbi kifejezéslista kifejezéseinek értékét:
NEV-(“ “+SZUL_HELY), KOLLEGIUM .AND. TESTVER < 3

57. fejezet

Kifejezéslista alkalmazása

A DISPLAY és a LIST parancsba bele lehet gépelni kifejezéslistát is.

Tudjuk, hogy ezek a parancsok az adatbázisfájl tartalmának kiírását végzik.

Ha nem adunk meg kifejezéslistát, akkor a kiírásra kerülő rekordokból valamennyi mezőnek a tartalma kiíródik. Ha viszont megadunk kifejezéslistát, akkor a kiírásra kerülő rekordokból a kifejezéslistában szereplő kifejezések értékei kerülnek kiírásra.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
SZUL_DATUM	D	8	-	Születési dátuma
MUNKA_KEZD	D	8	-	Munkaviszonyának kezdete
ALAPBER	N	6	0	Havi alaphére
POTLEK	N	6	0	Havi bérpótléka
LEVONAS	N	6	0	Havi levonása
EGYETEM	L	1	-	Végzett-e egyetemet?
VEZETO	L	1	-	Vezető beosztásban dolgozik-e?

a) Ha az összes dolgozó nevét és havi alapbérét akarjuk kiíratni, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV, ALAPBER
```

vagy

```
LIST NEV, ALAPBER
```

(Célszerűbb a DISPLAY parancsot használni, mert az lehetővé teszi a lapozást!)

b) Ha a 114. rekordtól kezdődően akarjuk kiíratni a dolgozók nevét és éves alapbérét, akkor ezeket a parancsokat adjuk ki:

```
GO 114
```

```
DISPLAY REST NEV, ALAPBER*12
```

c) Ha a 105. rekordtól kezdve tizenkét dolgozó nevét és havi járandóságát akarjuk kiíratni, akkor ezeket a parancsokat adjuk ki:

```
GO 105
```

```
DISPLAY NEXT 12 NEV, ALAPBER+POTLEK-LEVONAS
```

d) Ha az összes dolgozó közül az egyetemet végzett vezető beosztású dolgozók nevét és éves járandóságát akarjuk kiíratni, akkor ezt a parancsot adjuk ki:

```
LIST NEV, (ALAPBER+POTLEK-LEVONAS)*12 FOR EGYETEM .AND. VEZETO
```

e) Ha mindazoknak a dolgozóknak a nevét és havi alapbérét akarjuk kiíratni, akiknek a havi alapbére 15000 Ft-nál kevesebb, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV, ALAPBER FOR ALAPBER < 15000
```

f) Ha az összes dolgozó közül azoknak a nevét akarjuk kiíratni, akik nem vezető beosztásúak, de a havi alapbérük meghaladja a 16000 Ft-ot, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV FOR .NOT. VEZETO .AND. ALAPBER > 16000
```

g) Ha az összes dolgozó közül azoknak a nevét és havi alapbérét akarjuk kiíratni, akik 23. életévüket már betöltötték, amikor munkába álltak, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV, ALAPBER FOR MUNKA_KEZD-SZUL_DATUM > 23*365
```

Megjegyzés

Az eddig tanult parancsok közül csak a DISPLAY és a LIST engedi meg kifejezéslista használatát.

Feladatok

1. A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl szerkezete:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>
NEV	C	25	-
SZUL_HELY	C	20	-
SZUL_DATUM	D	8	-
TESTVER	N	2	0
ATLAG	N	4	2
KOLLEGIUM	L	1	-

- Töröld le a képernyőt!
- Nyisd meg az adatbázisfájlt!
- Írasd ki az 5. rekord összes mezőjének tartalmát!
- Írasd ki a 2. rekordtól kezdve három rekord minden mezőjének tartalmát!
- Írasd ki az összes rekord összes mezőjének tartalmát!
- Írasd ki a 4. rekordtól kezdve az összes rekord NEV és SZUL_HELY mezőjének tartalmát!
- Írasd ki a 3. rekord NEV és KOLLEGIUM mezőjének tartalmát!
- Írasd ki az 5. rekordtól kezdve három rekord NEV és SZUL_DATUM mezőjének tartalmát!
- Írasd ki a NEV, a TESTVER és az ATLAG mezők tartalmát azokból a rekordokból, ahol a KOLLEGIUM mező értéke **'igaz'**, és az ATLAG mező értéke kisebb 4.5-nél!
- Írasd ki a NEV, a TESTVER és az ATLAG mezők tartalmát azokból a rekordokból, ahol a KOLLEGIUM mező értéke **'igaz'**, vagy az ATLAG mező értéke kisebb 4.5-nél!
- Írasd ki a NEV mező tartalmát azokból a rekordokból, ahol a KOLLEGIUM mező értéke **'hamis'**.

2. A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjz.</i>	<i>Jelentés</i>
SZERZO	C	25	-	A könyv szerzője
CIM	C	30	-	Címe
KIADAS	N	4	0	Kiadásának éve
KEPEK	L	1	-	Vannak-e benne képek?
AR	N	6	2	Ára

a) Nyisd meg az adatbázisfájl!

b) Írasd ki a szerkezetét!

c) Írasd ki az összes olyan könyv címét és árát, amelyben vannak képek, és 1959 előtt adták ki!

d) Írasd ki az összes olyan könyv szerzőjét és kiadásának évszámát, amelynek az ára 25 Ft-nál több, de 35 Ft-nál kevesebb!

58. fejezet

A rekordsorszám letiltása

Ha a DISPLAY vagy a LIST parancs használatakor nem akarjuk, hogy a rekordsorszám megjelenjen a kiíratás során, akkor a parancsba bele kell gépelni az OFF szót.

Példa

A

```
DISPLAY ALL
```

és a

```
DISPLAY ALL OFF
```

parancsok között csupán az a különbség, hogy az első kiírja a rekordsorszámokat is, a második viszont nem.

59. fejezet

Kiírás a nyomtatóra

Ha a DISPLAY, a LIST, a DISPLAY STRUCTURE vagy a LIST STRUCTURE parancsnál azt akarjuk, hogy a kiírás a nyomtatón is megtörténjen, akkor a parancsba bele kell gépelni a TO PRINT szavakat.

Példa

A
LIST
és a
LIST TO PRINT

parancsok között csupán az a különbség, hogy az első csak a képernyőre írja ki az adatokat, a második viszont a nyomtatóra is.

Megjegyzés

Mielőtt a papírt kivennénk a nyomtatóból, általában ki kell adni az utolsó sor kiírását (és a lapemelést) eredményező EJECT parancsot.

60. fejezet

A mezőlista fogalma

A mezőlista mezőneveknek olyan felsorolását jelenti, ahol az egyes mezőneveket vesszővel választjuk el egymástól.

Példa

A nyitott adatbázisfájl szerkezete:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjz.</i>
NEV	C	25	-
SZUL_HELY	C	20	-
TESTVER	N	2	0
KOLLEGIUM	L	1	-

Mezőlisták:

- NEV, TESTVER
- NEV, SZUL_HELY, KOLLEGIUM

61. fejezet

Mezőlista alkalmazása

Vannak olyan parancsok, amelyek megengedik mezőlista alkalmazását.

Ha mezőlistát alkalmazunk, akkor a parancs csupán azokra a mezőkre vonatkozik, amelyek szerepelnek a mezőlistában.

A parancsban a mezőlista elé mindig ki kell írni a FIELDS szót.

Az eddig tanult parancsok közül a BROWSE, az EDIT és a CHANGE parancs engedi meg mezőlista alkalmazását.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjtz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
ATLAG	N	4	2	Tanulmányi átlaga
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

1. Ha kiadjuk a

```
BROWSE FIELDS NEV, ATLAG
```

parancsot, akkor csupán a tanulók nevét és tanulmányi átlagát módosíthatjuk.

2. Ha kiadjuk az

```
EDIT RECORD 4 FIELDS NEV, SZUL_HELY
```

parancsot, akkor csupán a 4. rekordban szereplő tanuló nevét és születési helyét módosíthatjuk.

3. Ha kiadjuk a

```
CHANGE FIELDS NEV, SZUL_DATUM, TESTVER FOR KOLLEGIUM
```

parancsot, akkor csupán a kollégiumban lakó tanulók nevét, születési dátumát és testvéreinek a számát módosíthatjuk.

62. fejezet

Összefoglalás

Az eddigiek során számos parancsot úgy tanultunk meg, hogy azonnal megismerkedtünk az összes lehetőségével. Ilyen volt a CLEAR, a QUIT, a CREATE, a DIR, az USE, a GO, a SKIP, az APPEND, az INSERT, a PACK és a ZAP parancs.

Ezzel szemben a DISPLAY, a DELETE, a RECALL, a LIST, az EDIT, a CHANGE, a BROWSE, a DISPLAY STRUCTURE és a LIST STRUCTURE parancsokat lépcsőzetes tanultuk meg:

- Mindenekelőtt megtanultuk, hogy melyik parancs mire használható.
- A parancsokat először a lehető legegyszerűbb formájukban alkalmaztuk: csupán a parancs nevét gépeltük be, majd azonnal leütöttük az Enter billentyűt.
- Később megismerkedtünk a hatáskör és a feltétel fogalmával, s megtanultuk, hogy ezek alkalmazása miként befolyásolja a DISPLAY, a DELETE, a RECALL, a LIST, az EDIT és a CHANGE parancsok működését.
- Megtanultuk, hogy kifejezéslista alkalmazása miként befolyásolja a DISPLAY és a LIST parancs működését.
- Megtanultuk, hogy az OFF szó miként befolyásolja a DISPLAY és a LIST parancs működését.
- Megtanultuk, hogy a TO PRINT szavak miként befolyásolják a DISPLAY, a LIST, a DISPLAY STRUCTURE és a LIST STRUCTURE parancs működését.
- Megtanultuk, hogy a mezőlista alkalmazása hogyan befolyásolja a BROWSE, az EDIT és a CHANGE parancsok működését.
- Megtanultuk, hogy a ? paranccsal nem csak egy művelet eredménye, hanem több kifejezés értéke is kiíratható.

63. fejezet

A parancs formája

A parancs formája egy olyan képlet, amely szemléletesen mutatja azt, hogy a parancsba milyen szavakat és fogalmakat kell vagy lehet belegépelni.

A parancs formáját leíró képletben

- nagybetűvel jelöljük azokat a szavakat, amiket betű szerint kell begépelni
- kisbetűkkel jelöljük azokat a fogalmakat (hatáskör, feltétel, kifejezéslista stb.), amiket értelemszerűen kell begépelni
- szögletes zárójelbe tesszük azt, amit nem kötelező begépelni
- | jellel jelöljük a választás lehetőségét
- egy fogalom ismétlődésekor indexeket használunk
- három ponttal jelöljük a további ismétlődés lehetőségét.

Példák

1. Láttuk, hogy a DISPLAY parancsba bele lehet írni hatáskört is, kifejezéslistát is, feltételt is, az OFF szót is meg a TO PRINT szavakat is. (A feltételt a FOR szó előzi meg.) Ezek közül egyedül a DISPLAY szó szerepeltetése kötelező.

Tehát a DISPLAY parancs formája:

DISPLAY [hatáskör] [kifejezéslista] [FOR feltétel] [OFF] [TO PRINT]

2. Láttuk, hogy a LIST parancsba bele lehet írni hatáskört is, kifejezéslistát is, feltételt is, az OFF szót is meg a TO PRINT szavakat is. (A feltételt a FOR szó előzi meg.) Ezek közül egyedül a LIST szó szerepeltetése kötelező.

Tehát a LIST parancs formája:

LIST [hatáskör] [kifejezéslista] [FOR feltétel] [OFF] [TO PRINT]

3. Láttuk, hogy a DELETE parancsba bele lehet írni hatáskört is meg feltételt is. (A feltételt a FOR szó előzi meg.) Ezek közül egyedül a DELETE szó szerepeltetése kötelező.

Tehát a DELETE parancs formája:

DELETE [hatáskör] [FOR feltétel]

4. Láttuk, hogy a RECALL parancsba bele lehet írni hatáskört is meg feltételt is. (A feltételt a FOR szó előzi meg.) Ezek közül egyedül a RECALL szó szerepeltetése kötelező.

Tehát a RECALL parancs formája:

RECALL [hatáskör] [FOR feltétel]

5. Láttuk, hogy a BROWSE parancsba bele lehet írni mezőlistát is. (A mezőlistát a FIELDS szó előzi meg.) Ezek közül egyedül a BROWSE szó szerepeltetése kötelező.

Tehát a BROWSE parancs formája:

BROWSE [FIELDS mezőlista]

6. Láttuk, hogy az EDIT parancsba bele lehet írni hatáskört is, mezőlistát is meg feltételt is. (A mezőlistát a FIELDS szó, a feltételt pedig a FOR szó előzi meg.) Ezek közül egyedül az EDIT szó szerepeltetése kötelező.

Tehát az EDIT parancs formája:

EDIT [hatáskör] [FIELDS mezőlista] [FOR feltétel]

7. Láttuk, hogy a CHANGE parancsba bele lehet írni hatáskört is, mezőlistát is meg feltételt is. (A mezőlistát a FIELDS szó, a feltételt pedig a FOR szó előzi meg.) Ezek közül egyedül a CHANGE szó szerepeltetése kötelező.

Tehát a CHANGE parancs formája:

CHANGE [hatáskör] [FIELDS mezőlista] [FOR feltétel]

8. Láttuk, hogy a CREATE parancs alkalmazásakor meg kell adni a létrehozandó adatbázisfájl teljes azonosítóját.

Tehát a CREATE parancs formája:

CREATE teljes_fájlazonosító

9. Láttuk, hogy ha az USE parancsot megnyitásra akarjuk használni, akkor meg kell adni a megnyitandó adatbázisfájl teljes azonosítóját. Ha viszont az USE parancsot lezárásra akarjuk használni, akkor nem kell megadni teljes fájlazonosítót.

Tehát az USE parancs formája:

USE [teljes_fájlazonosító]

10. Láttuk, hogy a DIR parancs alkalmazásakor a katalógusra való hivatkozással kell megadnunk, hogy mely katalógus adatbázisfájljait akarjuk számba venni. Ha nem a főkatalógusról van szó, akkor ezt egy \ követi. Ha az egyéb fájlokra is kíváncsiak vagyunk, akkor ezt követően még ezt is bele kell írni a parancsba: *.*

Tehát a DIR parancs formája:

DIR katalógushivatkozás[\] [*.*]

11. Láttuk, hogy a GO parancs alkalmazásakor meg kell adni egy numerikus konstans, vagy a TOP szót, vagy a BOTTOM szót.

Tehát a GO parancs formája:

GO numerikus_konstans|TOP|BOTTOM

12. Láttuk, hogy a SKIP parancs alkalmazásakor meg lehet adni egy numerikus konstans.

Tehát a SKIP parancs formája:

SKIP [numerikus_konstans]

13. Láttuk, hogy az INSERT parancs alkalmazásakor meg lehet adni a BEFORE szót.

Tehát az INSERT parancs formája:

INSERT [BEFORE]

14. Láttuk, hogy a DISPLAY STRUCTURE és a LIST STRUCTURE parancsban meg lehet adni a TO PRINT szavakat is.

Tehát ennek a két parancsnak a formája:

DISPLAY STRUCTURE [TO PRINT]

LIST STRUCTURE [TO PRINT]

15. Láttuk, hogy a CLEAR, a QUIT, az APPEND, a PACK és a ZAP parancsoknál csak a parancs nevét kell megadni.

Tehát ezeknek a parancsoknak a formája:

CLEAR
QUIT
APPEND
PACK
ZAP

16. Láttuk, hogy a ? paranccsal egy kifejezéslista értékeit írathatjuk ki.

Tehát a ? parancs formája:

? kifejezéslista

Megjegyzések

1. Ha egy fogalom csak több szóval határozható meg, akkor a félreértések elkerülése érdekében a szavakat aláhúzásjellel kötjük össze, így hangsúlyozva ki, hogy egyetlen fogalomról van szó.

Ezért írtuk a teljes fájlazonosítót így: **teljes_fájlazonosító**, a numerikus konstans pedig így: **numerikus_konstans**.

2. Azokat a szavakat, amelyek a parancs formájában nagybetűvel vannak írva, nem kötelező végig begépelni: elég, ha csak az első négy betűjüket gépeljük be. Ha viszont további betűiket is begépeljük, akkor azoknak helyeseknek kell lenniük.

3. Természetesen nem elég a parancs formáját ismernünk, hanem azt is tudnunk kell, hogy mire használható és hogyan működik a parancs.

64. fejezet

A mezők tartalmának automatikus módosítása

Láttuk, hogy a nyitott adatbázisfájl mezőinek tartalma módosítható az EDIT, a CHANGE, a BROWSE vagy az APPEND parancsok bármelyikével. Ezekkel a parancsokkal minden adatot csak egyedileg módosíthatunk.

Ezzel szemben a REPLACE paranccsal a nyitott adatbázisfájl mezőinek tartalmát több rekordban is automatikusan módosíthatjuk.

A parancs formája:

REPLACE [hatáskör] **mezőnév₁** WITH kifejezés₁
[,mezőnév₂ WITH kifejezés₂ ...] [FOR feltétel]

A parancs csak a hatáskör és a feltétel által meghatározott rekordokra vonatkozik. Mezőnévként a nyitott adatbázisfájl mezőneveit adhatjuk meg.

A WITH előtt megadott mező tartalma a WITH után megadott kifejezés értékével lesz egyenlő.

A WITH előtt megadott mező típusának meg kell egyeznie az utána megadott kifejezés típusával.

A három pont a további ismétlődés lehetőségét fejezi ki, tehát egy paranccsal sok mező tartalmát módosíthatjuk.

A hatáskör alapértelmezése az aktuális rekord.

Példa

A nyitott adatbázisfájl egy kereskedelmi vállalat által forgalmazott termékekről az alábbi adatokat tartalmazza:

- a termék neve
- kódja
- beszerzési egységára
- haszonkulcsa.

A termék kódja olyan egybetűs adat, amely a termék fajtáját fejezi ki, az alábbi táblázatnak megfelelően:

A termék fajtája	Kódja
Élelmiszer	E
Vegyianyag	V
Ruhanemű	R

A haszonkulcs százalékban van megadva, és azt fejezi ki, hogy a beszerzési árhoz képest hány százalékkal magasabb a termék eladási ára.

Az adatbázisfájl szerkezete:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjisz.</i>
TERMEKNEV	C	25	-
KOD	C	1	-
BESZERZ_AR	N	6	0
H_KULCS	N	2	0

a) A vegyi anyagok beszerzési ára egységesen 20 százalékkal nőtt, a haszonkulcs pedig mostantól kezdve 30 százalék lesz.

A változásokat ezzel a paranccsal vezetjük át az adatbázisfájlon:

```
REPLACE BESZERZ_AR WITH BESZERZ_AR*1.2, H_KULCS WITH 30 FOR KOD='V'
```

b) Az élelmiszerek kódját E betűről L betűre szeretnénk változtatni. A változtatást ezzel a paranccsal végezzük el:

```
REPLACE KOD WITH "L" FOR KOD = 'E'
```

Feladat

A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl AR nevű numerikus mezőjének a tartalmát növeld másfélszeresére azoknál a rekordoknál, ahol a KEPEK nevű logikai mező értéke 'igaz'!

65. fejezet

Összegzés, átlagolás, számlálás

Eddig csak az adatbázisfájl egyes rekordjain belül végeztünk műveleteket.

Lehetőség van arra is, hogy a rekordok között végezzünk összegzést, átlagolást vagy számlálást.

a) Összegzés és átlagolás

Ha egy vagy több kifejezés értékét a rekordok vonatkozásában akarjuk összegezni, akkor a SUM parancsot kell használni; ha pedig átlagolni akarjuk, akkor az AVERAGE parancsot kell alkalmazni.

A SUM parancs formája:

```
SUM [hatáskör] [numerikus_kifejezéslista] [FOR feltétel]
```

A numerikus kifejezéslistában szereplő kifejezések értéke kiszámításra kerül a hatáskör és a feltétel által meghatározott rekordoknál, majd az így kapott értékek összegződnek. Eredményül annyi számot kapunk, mint ahány numerikus kifejezést megadtunk.

Ha nem adunk meg egyetlen numerikus kifejezést sem, akkor az adatbázisfájl valamennyi numerikus mezője összegződik.

A hatáskör alapértelmezése: ALL.

Az AVERAGE parancs formája:

```
AVERAGE [hatáskör] [numerikus_kifejezéslista] [FOR feltétel]
```

A numerikus kifejezéslistában szereplő kifejezések értéke kiszámításra kerül a hatáskör és a feltétel által meghatározott rekordoknál, majd az így kapott értékek átlago-

lódnak. Eredményül annyi számot kapunk, mint ahány numerikus kifejezést megadtunk.

Ha nem adunk meg egyetlen numerikus kifejezést sem, akkor az adatbázisfájl valamennyi numerikus mezője átlagolódik.

A hatáskör alapértelmezése: ALL.

b) Számlálás

Ha meg akarjuk tudni, hogy valamilyen feltételnek hány rekord felel meg, akkor a COUNT parancsot kell használni.

A parancs formája:

COUNT [hatáskör] [FOR feltétel]

Eredményképpen megjelenik a képernyőn, hogy a nyitott adatbázisfájlban mennyi a hatáskör és a feltétel által meghatározott rekordok száma.

A hatáskör alapértelmezése: ALL.

Példák

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tízs.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
ALAPBER	N	6	0	Havi alaphére
POTLEK	N	6	0	Havi bérpótléka
LEVONAS	N	6	0	Havi levonása
EGYETEM	L	1	-	Végzett-e egyetemet?
VEZETO	L	1	-	Vezető beosztásban dolgozik-e?

a) Ha ki akarjuk számítani a dolgozóknak egy hónapban fizetendő összes alaphért és az összes bérpótlékot, akkor ezt a parancsot adjuk ki:

SUM ALAPBER, POTLEK

Eredményül két számot kapunk. Az első a dolgozók havi alaphérének összegét mutatja, a második pedig a havi bérpótlékaik összegét.

b) Ha az

AVERAGE ALAPBER, POTLEK

parancsot kiadjuk, akkor eredményül a dolgozók havi alapbérének átlagát, valamint havi bérpótlékuk átlagát kapjuk.

c) Ha ki akarjuk számítani a dolgozók számára az egész év folyamán kifizetendő teljes pénzösszeget, akkor ezt a parancsot adjuk ki:

```
SUM 12*(ALAPBER+POTLEK-LEVONAS)
```

Eredményül most egyetlen számot kapunk, mert csak egy numerikus kifejezést adtunk meg.

d) Ha kiadjuk az

```
AVERAGE 12*(ALAPBER+POTLEK-LEVONAS)
```

parancsot, akkor eredményül a dolgozók évi járandóságának átlagát kapjuk.

e) Ha kiadjuk a

```
SUM
```

parancsot, akkor három számot kapunk:

- a dolgozók havi alapbérének összegét
- a dolgozók havi bérpótlékának összegét
- a dolgozók havi levonásának összegét.

f) Ha kiadjuk az

```
AVERAGE
```

parancsot, akkor három számot kapunk:

- a dolgozók havi alapbérének átlagát
- a dolgozók havi bérpótlékának átlagát
- a dolgozók havi levonásának átlagát.

g) Ha ki akarjuk számítani az egyetemet végzett vezető beosztású dolgozók havi járandóságának átlagát, akkor ezt a parancsot adjuk ki:

```
AVERAGE (ALAPBER+POTLEK-LEVONAS) FOR EGYETEM .AND. VEZETO
```

h) Ha tudni akarjuk, hogy hány olyan egyetemet végzett dolgozó van, akinek a havi alapbére nem éri el a 20000 Ft-ot, akkor ezt a parancsot adjuk ki:

```
COUNT FOR EGYETEM .AND. ALAPBER < 20000
```

Feladatok

1. A gyakorlólemez KM katalógusában található TANULO.DBF adatbázisfájl felhasználásával számítsd ki, hogy

- a) a tanulóknak összesen hány testvérük van
- b) mennyi az osztály tanulmányi átlaga

c) hány tanulónak jobb a tanulmányi átlaga 3.75-nál?

2. A gyakorlólemez KM katalógusában található KONYV.DBF adatbázisfájl felhasználásával számítsd ki, hogy

a) mennyi a könyvek összértéke

b) egy könyv átlagosan mennyibe került

c) hány olyan könyv van, amely 1965 előtt jelent meg?

66. fejezet

Adatbázisfájl szerkezetének megváltoztatása

Az adatbázisfájl szerkezetét még akkor is megváltoztathatjuk, amikor már fel van töltve adatokkal.

A szerkezetben bármilyen változtatást el lehet végezni.

A nyitott adatbázisfájl szerkezetét a MODIFY STRUCTURE paranccsal lehet megváltoztatni.

A parancs formája:

MODIFY STRUCTURE

A parancs kiadásakor a képernyőn megjelenik a nyitott adatbázisfájl szerkezete, s máris lehetővé válik annak megváltoztatása.

A mezők egyike kiválasztó sávval van megkülönböztetve a többitől. Mindig azon a mezőn lehet változtatást végezni, amelyiken rajta van a kiválasztó sáv. A kiválasztó sávot a ↓ vagy a ↑ nyílbillentyűvel mozgatjuk. A mezőn belül a többi kurzormozgató billentyűvel mozgunk.

Meg lehet változtatni a mező nevét, típusát, hosszát és a tizedesjegyek számát is.

Ha egy mezőt meg akarunk szüntetni, akkor a kiválasztó sávot arra a mezőre kell vinni, majd alkalmazni kell a Ctrl U billentyűkombinációt.

Ha új mezőt akarunk betoldani, akkor a kiválasztó sávot arra a mezőre kell vinni, amely elé be akarjuk a mezőt toldani, majd alkalmazni kell a Ctrl N billentyűkombinációt. Ennek hatására hely keletkezik az új mező számára, s meg lehet adni a mező nevét, típusát, hosszát és a tizedesjegyek számát.

A változtatásokat a Ctrl End billentyűkombinációval véglegesítjük. Ennek hatására megjelenik az alábbi felirat:

Press ENTER to confirm. Any other key to resume.

Ez azt jelenti, hogy ha valóban befejeztük a szerkezet megváltoztatását, akkor Enter-t kell ütnünk, egyébként bármilyen más billentyűt. Ez utóbbi esetben folytatható a szerkezet megváltoztatása.

Ha a szerkezet megváltoztatását az Esc leütésével fejezzük be, akkor a végrehajtott változtatások érvénytelenítődnek.

* * *

Vegyük sorra, hogy a változtatások milyen hatással vannak az adatbázisfájl tartalmára.

a) Ha a szerkezet megváltoztatása során mezőnevet is változtatunk, akkor a következő kérdést kapjuk:

Should data be COPIED from backup for all fields? (Y/N)

Magyarul:

A megváltoztatott nevű mezők tartalma megmaradjon-e? (Y/N)

Ha az Y-nal válaszolunk, akkor a tartalom megmarad, ha viszont az N-nel, akkor a megváltoztatott nevű mezők tartalma minden rekordban üres lesz. Bárhogyan is döntünk, döntésünket az Enter leütésével meg kell erősíteni.

b) Ha a mező típusát megváltoztatjuk, akkor előfordulhat, hogy a változást követően a mező tartalma minden rekordban üressé válik. Ez akkor fordul elő, ha például karakteres mező típusát dátum jellegűre változtatjuk.

c) Ha a mező hosszát csökkentjük, akkor előfordulhat, hogy egyes rekordokban a mező tartalma megcsonkul.

d) Ha a mező hosszát megnöveljük, akkor a mező addigi tartalma megmarad, és a végén még helykitöltő karakterek keletkeznek.

A mező hosszának növelése arra jó, hogy ebbe a mezőbe a későbbiekben hosszabb adatot is begépelhessünk.

e) Ha egy mezőt megszüntetünk, akkor annak addigi tartalma természetesen elvész valamennyi rekordból.

f) Ha új mezőt hozunk létre, akkor annak tartalma minden rekordban üres mindaddig, míg föl nem töltjük adatokkal.

Új mezőt akkor hozunk létre, ha az egyedekről még újabb adatot is tárolni akarunk.

g) Ha a mező nevét és hosszát egyszerre változtatjuk, akkor a mező tartalma elvész.

Feladat

A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl szerkezetében hajtsd végre az alábbi változtatásokat:

a) A CIM mező hosszát növeld meg 2-vel!

b) Az AR mező hosszát csökkentsd 3-ra, a tizedesjegyeinek számát pedig csökkentsd 0-ra!

c) A KEPEK mező után toldj be egy új mezőt! A mező azt fogja majd tartalmazni, hogy a könyvet olvastad-e már. A mező neve OLVAS legyen, a típusa pedig logikai!

- d) A most létrehozott OLVAS nevű mezőt töltsd fel valós adatokkal!
- e) Írasd ki az adatbázisfájl mostani szerkezetét!
- f) Írasd ki a képernyőre az adatbázisfájl mostani tartalmát!

67. fejezet

Fájl másolása

A COPY FILE paranccsal másolatot készíthetünk egy tetszőleges fájlról (kivéve a nyitott adatbázisfájlról).

A parancs formája:

COPY FILE teljes_fájlazonosító TO teljes_fájlazonosító

A TO szó előtt a másolandó fájl teljes azonosítóját, a TO szó után pedig a másolat teljes azonosítóját kell megadni. Ha a másolandó fájlnek vagy a másolatnak nincs kiterjesztése, akkor a név után a pontot ki kell tenni.

Példa

Ha a C: meghajtó DBASE katalógusának ADAT alkatalógusában levő ELAD.TXT fájlról olyan másolatot akarnánk készíteni, amely a gyakorlólemez főkatalógusában fog elhelyezkedni ELAD.DAT azonosítóval, akkor ezt a parancsot adnánk ki:

```
COPY FILE C:\DBASE\ADAT\ELAD.TXT TO A:\ELAD.DAT
```

Megjegyzés

A COPY FILE parancsot ne keverjük össze a DOS COPY parancsával!

A két parancs között az a lényeges különbség, hogy a DOS COPY parancsában alkalmazhatunk globális fájlhivatkozást, a COPY FILE parancsban viszont nem.

68. fejezet

Fájl átnevezése

A RENAME paranccsal átnevezhetünk egy tetszőleges fájlt (kivéve a nyitott adatbázisfájlt).

A parancs formája:

RENAME teljes_fájlazonosító TO teljes_fájlazonosító

A TO szó előtt az átnevezendő fájl teljes azonosítóját, a TO szó után pedig az új teljes azonosítóját kell megadni. Ha az eredeti fájlnek vagy az új fájlnek nincs kiterjesztése, akkor a név után a pontot ki kell tenni.

Példa

Ha a C: meghajtó DBASE katalógusának ADAT alkatalógusában levő ELAD.TXT fájlt ELADOTT.DAT-ra akarnánk átnevezni, akkor ezt a parancsot adnánk ki:

```
RENAME C:\DBASE\ADAT\ELAD.TXT TO C:\DBASE\ADAT\ELADOTT.DAT
```

69. fejezet Fájl törlése

Az ERASE paranccsal törölhetünk egy tetszőleges fájlt (kivéve a nyitott adatbázis-fájlt).

A parancs formája:

ERASE teljes_fájlazonosító

A törlendő fájl teljes azonosítóját kell megadni. Ha a törlendő fájlnek nincs kiterjesztése, akkor a név után a pontot ki kell tenni.

Példa

Ha a C: meghajtó DBASE katalógusának ADAT alkatalógusában levő ELAD.DAT fájlt ki akarnánk törölni, akkor ezt a parancsot adnánk ki:

```
ERASE C:\DBASE\ADAT\ELAD.DAT
```

Megjegyzés

Az ERASE parancs hasonlít a DOS DEL parancsához.

A két parancs között az a lényeges különbség, hogy a DEL parancsban alkalmazhatunk globális fájlhivatkozást, az ERASE parancsban viszont nem.

70. fejezet Adatbázisfájl másolása

A dBASE COPY parancsa a nyitott adatbázisfájlról készít másolatot.

Ha egy adatbázisfájlról a DOS COPY parancsával vagy a dBASE COPY FILE parancsával készítünk másolatot, akkor valamennyi rekord és valamennyi mező átkerül a másolatba.

Ha viszont a dBASE COPY parancsával készítjük a másolatot, akkor az eredeti adatbázisfájlból nem feltétlen fog minden rekord és minden mező átmásolódni az új adatbázisfájlba.

A dBASE COPY parancsának formája:

COPY [hatáskör] [FIELDS mezőlista] TO teljes_fájlazonosító [FOR feltétel]

A készülő fájl teljes azonosítóját a TO szó után kell megadni.

A másolatba csak a hatáskör és a feltétel által meghatározott rekordok kerülnek át.

Ha megadjuk a FIELDS szót, és azt követően egy mezőlistát, akkor a másolatba csak a mezőlistában szereplő mezők kerülnek át.

A hatáskör alapértelmezése: ALL.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_DATUM	D	8	-	Születési dátuma
LAKCIM	C	40	-	Lakcíme
IRODALOM	N	1	0	Osztályzata irodalomból
MATEMATIKA	N	1	0	Osztályzata matematikából
FIZIKA	N	1	0	Osztályzata fizikából
KEMIA	N	1	0	Osztályzata kémiából

a) Ha olyan másolatot akarunk készíteni az adatbázisfájlról, amely a tanulók nevén kívül csak a matematikából, fizikából és kémiából szerzett érdemjegyüket tartalmazza, és azt akarjuk, hogy az új adatbázisfájl teljes azonosítója A:\KM\TERM.DBF legyen, akkor ezt a parancsot adjuk ki:

```
COPY FIELDS NEV, MATEMATIKA, FIZIKA, KEMIA TO A:\KM\TERM.DBF
```

vagy rövidebben:

```
COPY FIELDS NEV, MATEMATIKA, FIZIKA, KEMIA TO A:\KM\TERM
```

b) Ha olyan másolatot akarunk készíteni az adatbázisfájlról, amely csak az irodalomból 5-ös osztályzattal rendelkező tanulók nevét és matematika érdemjegyet tartalmazza, és azt akarjuk, hogy a másolat teljes fájlazonosítója A:\KMMAT.DBF legyen, akkor ezt a parancsot adjuk ki:

```
COPY FIELDS NEV, MATEMATIKA FOR IRODALOM = 5 TO A:\KMMAT.DBF
```


Feladatok

1. A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájlról készíts olyan másolatot, amely csak a 3.75-nál jobb tanulmányi átlaggal rendelkező tanulók adatait tartalmazza. A testvérek számát ne másold át az új adatbázisfájlba! A másolat teljes azonosítója A:\KM\JOBBAK.DBF legyen!

2. A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájlról készíts olyan másolatot, amely csak azokat a könyveket tartalmazza, amelyekben vannak képek. A másolat teljes azonosítója A:\KM\KEP.DBF legyen!

71. fejezet

Keresés az adatbázisfájlban

Ha a nyitott adatbázisfájlban a tartalma alapján akarunk megkeresni egy rekordot, akkor a LOCATE és a CONTINUE parancsot kell használni.

A LOCATE parancs formája:

LOCATE [hatáskör] [FOR feltétel]

A CONTINUE parancs formája:

CONTINUE

Nézzük, hogyan működnek ezek a parancsok.

A LOCATE parancsban olyan feltételt kell megadnunk, amely teljesül a keresett rekordra.

A LOCATE parancs a hatáskör és a feltétel által meghatározott rekordok közül a legelsőket találja meg, és azonnal aktuálissá is teszi.

Ha a hatáskör és a feltétel több rekordot határoz meg, akkor elképzelhető, hogy ezek közül nem azt keressük, amelyet a LOCATE parancs aktuálissá tett.

Ez esetben alkalmaznunk kell a CONTINUE parancsot.

A CONTINUE parancs ugyanis a következő olyan rekordot teszi aktuálissá, amit a LOCATE parancs hatásköre és feltétele meghatároz.

A CONTINUE parancsot annyiszor kell kiadni, míg célhoz nem érünk.

Megjegyzések

1. A DISPLAY parancsot felhasználhatjuk annak eldöntésére, hogy a kívánt rekord vált-e már aktuálissá.

2. A LOCATE parancsban a hatáskör alapértelmezése: ALL.

3. Ha már nincs olyan rekord, amelyet a LOCATE parancs hatásköre és feltétele meghatároz, akkor a képernyőn ez az üzenet jelenik meg:

End of locate scope.

Példa

A nyitott adatbázisfájl a vállalat dolgozóiról többek között az alábbi adatokat tartalmazza:

- neve
- havi alapbére.

A dolgozó neve a NEV nevű karakteres mezőben, a havi alapbére pedig az ALAPBER nevű numerikus mezőben van elhelyezve.

Az adatbázisfájl NEV és ALAPBER mezőinek tartalma:

	NEV	ALAPBER
1.	Gyapai Henrik	90000
2.	K. Szabó Pál	52000
3.	Gyulai Lajos	50000
4.	Kerek Ferenc	35000
5.	Nagy Zoltán	36000
6.	Annus Judit	39000
7.	Judik Erzsébet	40000
8.	Szabó Pálné	48000
9.	Homonnai Anikó	25000
10.	Szarvas Katalin	26000
11.	Duba Gyula	26000
12.	Kukorelli Géza	27000
13.	Anka Judit	29000
14.	Gyulai Beáta	15000
15.	Szász Károly	20000

	NEV	ALAPBER
16.	Kutas Angéla	19500
17.	Kiss Gyula	17500
18.	Fekete András	23000
19.	Csabai Antal	20000
20.	Gerőházi Tihamér	15000
21.	Gömöri Judit	17500
22.	Andai Géza	16000
23.	Cser Tibor	12000
24.	Csurgai Klára	13000
25.	Kovács Anett	11500
26.	Árvay Katalin	15500
27.	Hemző Etelka	18000
28.	Lelei Ambrus	14000
29.	Kelemen Judit	13000

Gömöri Judit rekordját szeretnénk aktuálissá tenni, de vezetéknevére nem emlékszünk pontosan. Ilyenkor a következőképpen járunk el:

Kiadjuk a

LOCATE ALL FOR "Judit" \$ NEV

parancsot. Ennek hatására először Annus Judit rekordja válik aktuálissá. Mi azonban nem az ő rekordját akartuk aktuálissá tenni, ezért ki kell adnunk a

CONTINUE

parancsot. Ennek hatására most a következő Judit, vagyis Anka Judit rekordja válik aktuálissá. Csalódottan megyünk tovább: ismételten kiadjuk a

CONTINUE

parancsot. Ekkor az újabb Judit, vagyis Gömöri Judit rekordja válik aktuálissá. Itt fellélegzünk, mert e nevet látván eszünkbe jut, hogy valóban Gömöri Juditnak hívják azt a dolgozót, akinek a rekordját aktuálissá akartuk tenni.

Megjegyzés

Kevesebb kínlódással értünk volna célba, ha többet tudtunk volna a keresett dolgozóról. Ha például tudtuk volna azt, hogy 17500 Ft a havi alapbére, akkor kiadhattuk volna a

```
LOCATE ALL FOR "Judit" $ NEV .AND. ALAPBER = 17500
```

parancsot, s ennek hatására azonnal Gömöri Judit rekordja vált volna aktuálissá, hiszen a Juditok közül egyedül ő az, akinek az alapbére pontosan 17500 Ft.

72. fejezet

A rekordok kezelési sorrendje

Gyakran előfordul, hogy a rekordokat a sorszámuktól eltérő sorrendben akarjuk kezelni.

Példák

1. Ha a tanulókról készült adatbázisfájl rekordjait névsor szerint akarnánk kiíratni, akkor az lenne a jó, ha a névsorban legelső helyen álló tanuló rekordja lenne az 1. rekord, a második helyen álló tanulóé a 2. rekord, és így tovább.

Más szóval: jó lenne, ha a rekordok a NEV mező tartalma szerint növekvő sorrendben helyezkednének el az adatbázisfájlban.

2. Ha a tanulókról készült adatbázisfájl rekordjait a tanulmányi átlag csökkenő sorrendjében akarnánk kiíratni, akkor az lenne a jó, ha a legjobb tanulmányi átlaggal rendelkező tanuló rekordja lenne az 1. rekord, a rangsorban második helyen álló tanulóé a 2. rekord, és így tovább.

Más szóval: jó lenne, ha a rekordok az ATLAG mező tartalma szerint csökkenő sorrendben helyezkednének el az adatbázisfájlban.

Azt, hogy a rekordokat az igényeinknek megfelelő sorrendben kezelhessük, kétféle módszerrel érhetjük el:

- *rendezéssel*
- *indexeléssel.*

73. fejezet

Rendezés

A rendezés módszere az, hogy a rekordok kezelési sorrendjének megváltoztatása érdekében az adatbázisfájlról olyan másolat készül, amely a rekordokat már az igényünknek megfelelő sorrendben tartalmazza. Az így létrejövő másolatot rendezett adatbázisfájlnak nevezzük.

A nyitott adatbázisfájl rendezését a SORT paranccsal kell végrehajtani. A parancs formája:

SORT [hatáskör] ON igénylista [FOR feltétel] TO teljes_fájlazonosító

A TO szó után a rendezett adatbázisfájl teljes azonosítóját kell megadni.

A rendezett adatbázisfájlba csak a hatáskör és a feltétel által meghatározott rekordok kerülnek át.

A hatáskör alapértelmezése: ALL.

A sorrendre vonatkozó igényünket az ON szót követő igénylistával kell kifejezni.

Az igénylista egy vagy több mezőnevet tartalmaz, egymástól vesszővel elválasztva.

Az igénylistában felsorolt mezőknek a tartalma alapján alakul ki a rekordok sorrendje az új adatbázisfájlban.

A mezőnevet (a vessző megadása előtt) — egy / jel mögött — úgynevezett kapcsolók követhetik .

A használható kapcsolók:

A

D

C

Ha megadjuk az **A** kapcsolót, akkor a mező tartalma szerint növekvő sorrend alakul ki.

Ha megadjuk a **D** kapcsolót, akkor a mező tartalma szerint csökkenő sorrend alakul ki.

A **C** kapcsolót csak karakteres mezőnél adhatjuk meg. A **C** kapcsolóval azt fejezzük ki, hogy a mezőben szereplő kis- és nagybetűk között nem akarunk különbséget tenni a sorrend kialakításakor. (Ez a kívánságunk azonban csak az angol ABC-be tartozó betűknél teljesül. Tehát például az *a* és az *A* akkor is különbözőnek minősül, ha megadjuk a **C** kapcsolót.)

Ha sem az **A**, sem a **D** kapcsolót nem adjuk meg, akkor a mező tartalma szerint növekvő sorrend alakul ki.

A sorrendet elsősorban az igénylistában elsőként megadott mező tartalma határozza meg (a kapcsolótól függően növekvően vagy csökkenően). Ha ennek a mezőnek a tartalma bizonyos rekordoknál megegyezik, akkor ezeknek a rekordoknak a sorrendjét a másodikként megadott mező tartalma határozza meg (a kapcsolótól függően növekvően vagy csökkenően). És így tovább, mindaddig, míg el nem fogy az igénylista.

Az igénylistában nem szerepelhet logikai mező.

A rendezést követően az eredeti adatbázisfájl változatlanul megmarad.

Példák

1. A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
ATLAG	N	4	2	Tanulmányi átlaga
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

Az adatbázisfájl tartalma:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
2.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
3.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y
4.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
5.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y

a) Ha a rekordokat a tanuló neve szerint növekvően akarjuk rendezni, és azt akarjuk, hogy a rendezett adatbázisfájl teljes azonosítója A:\KM\NEVSOR.DBF legyen, akkor ezt a parancsot adjuk ki:

```
SORT ALL ON NEV TO A:\KMNEVSOR.DBF
```

vagy rövidebben:

```
SORT ON NEV TO A:\KMNEVSOR
```


A parancs végrehajtása után a NEVSOR.DBF tartalma ez lesz:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
2.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
3.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y
4.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
5.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y

Ha ezek után olyan listát akarunk kapni, amelyben a tanulók névsorban követik egymást, akkor a NEVSOR.DBF tartalmát kell kiíratni.

b) Ha a rekordokat a tanulmányi átlag szerint csökkenően akarjuk rendezni, és azt akarjuk, hogy a rendezett adatbázisfájl teljes azonosítója A:\KM\JEGYEK.DBF legyen, akkor ezt a parancsot adjuk ki:

```
SORT ON ATLAG /D TO A:\KM\JEGYEK.DBF
```

A parancs végrehajtása után a JEGYEK.DBF tartalma ez lesz:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
2.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
3.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
4.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
5.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y

Ha ezek után olyan listát akarunk kapni, amelyben a tanulók a tanulmányi átlag szerint csökkenően követik egymást, akkor a JEGYEK.DBF tartalmát kell kiíratni.

c) Ha kiadjuk a

```
SORT ON SZUL_HELY, TESTVER, SZUL_DATUM /D TO A:\KMHELY.DBF
```

parancsot, akkor a rekordok elsősorban a születési hely szerint növekvően rendeződnek.

Azoknál a rekordoknál, ahol a születési hely megegyezik, ott a sorrend a testvérek száma szerint növekvően alakul ki.

Azoknál a rekordoknál pedig, ahol a születési hely és a testvérek száma is megegyezik, ott a sorrend a születési dátum szerint csökkenően alakul ki.

A rendezésben valamennyi rekord részt vesz, mert a hatáskör alapértelmezése ALL, feltételt pedig nem adtunk meg.

A rendezett adatbázisfájl a gyakorlólemez KM katalógusába kerül, azonosítója HELY.DBF, tartalma pedig ez lesz:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y
2.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
3.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
4.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
5.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
6.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y

d) Ha kiadjuk a

```
SORT ON ATLAG /D, NEV FOR TESTVER = 3 TO A:\KMHAROM.DBF
```

parancsot, akkor a rendezett adatbázisfájlba csak azok a rekordok kerülnek bele, ahol a testvérek száma pontosan 3.

A rekordok elsősorban a tanulmányi átlag szerint csökkenően rendeződnek.

Azoknál a rekordoknál, ahol a tanulmányi átlag megegyezik, a sorrend a név szerint növekvően alakul ki.

A rendezett adatbázisfájl a gyakorlólemez KM katalógusába kerül, azonosítója HAROM.DBF, tartalma pedig ez lesz:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
2.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
3.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N

2. A gyakorlólemez KM katalógusában levő KONYV.DBF adatbázisfájl szerkezetét és a mezők jelentését a következő táblázat szemlélteti:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
SZERZO	C	25	-	A könyv szerzője
CIM	C	32	-	Címe
KIADAS	N	4	0	Kiadásának éve
KEPEK	L	1	-	Vannak-e benne képek?
OLVAS	L	1	-	Olvastad-e?
AR	N	3	0	Ára

a) Ha ki akarjuk íratni az összes könyv szerzőjét és címét a szerző szerint csökkenő sorrendben, de a kis- és nagybetűk között nem akarunk különbséget tenni, akkor az alábbi parancsokat adjuk ki:

1) Az eredeti adatbázisfájl megnyitása:

```
USE A:\KMKONYV.DBF
```

2) Rendezés a szerző neve szerint:

```
SORT ON SZERZO/DC TO A:\KM\SZER.DBF
```

3) A rendezett adatbázisfájl megnyitása:

```
USE A:\KM\SZER.DBF
```

4) A szerző és a cím kiírása:

```
DISPLAY ALL SZERZO, CIM
```

b) Ha ki akarjuk íratni az összes könyv címét, árát és a kiadásának évét a kiadás éve szerint növekvő sorrendben, akkor az alábbi parancsokat adjuk ki:

1) Az eredeti adatbázisfájl megnyitása:

```
USE A:\KMKONYV.DBF
```

2) Rendezés a kiadás éve szerint:

```
SORT ON KIADAS TO A:\KM\KIAD.DBF
```

3) A rendezett adatbázisfájl megnyitása:

```
USE A:\KM\KIAD.DBF
```

4) A cím, az ár és a kiadás évének kiírása:

```
DISPLAY ALL CIM, AR, KIADAS
```

c) Ha csak a 40 Ft-nál drágább könyvek címét és árát akarjuk kiírni az ár csökkenő sorrendjében, akkor az alábbi parancsokat adjuk ki:

1) Az eredeti adatbázisfájl megnyitása:

```
USE A:\KMKONYV.DBF
```

2) Rendezés az ár szerint csökkenő sorrendben:

```
SORT ON AR /D FOR AR > 40 TO A:\KM\ARREN
```


3) A rendezett adatbázisfájl megnyitása:

USE A:\KMMARREN.DBF

4) A cím és az ár kiírása:

DISPLAY ALL CIM, AR

Feladatok

1. Írasd ki a gyakorlólemez KM katalógusában található TANULO.DBF adatbázisfájl tartalmát

a) név szerint növekvő sorrendben

b) tanulmányi átlag szerint csökkenő sorrendben

c) a testvérek száma szerint csökkenő sorrendben, azon belül pedig a születési hely szerint növekvő sorrendben!

2. Írasd ki a gyakorlólemez KM katalógusában található KONYV.DBF adatbázisfájlból a kiadás éve szerint növekvően rendezve azoknak a könyveknek a címét és szerzőjét, amelyek 1954 után jelentek meg, és 30 Ft-nál többbe kerültek!

Megjegyzések

1. A SORT parancsban az igénylista maximum tíz mezőnevet tartalmazhat.

2. Ha a rendezést követően módosítjuk az eredeti adatbázisfájl tartalmát, akkor a rendezett adatbázisfájl már hamis képet mutat. Ennek elkerülése érdekében a módosítás után törölni kell a rendezett adatbázisfájlt, majd a SORT paranccsal újra létre kell hozni.

3. A rendezett adatbázisfájlt érdemes kitörölni, ha már nincs rá szükségünk, mert különben hamar betelik a lemezünk.

74. fejezet

Az indexelés fogalma

Az indexelés módszere az, hogy a rekordok kezelési sorrendjét anélkül módosítjuk, hogy a rekordok tényleges sorrendjét megváltoztatnánk.

A kezelési sorrend módosítását egy általunk megadott kifejezéssel, az úgynevezett kulcskifejezéssel valósítjuk meg.

A kulcskifejezés értéke minden rekordnál kiszámításra kerül, s a rekordok kezelési sorrendje ennek a kifejezésnek az értéke szerint növekvően alakul ki.

Az így kialakuló kezelési sorrend egy külön fájlban, az úgynevezett indexfájlban tárolódik.

Az indexfájl annyi sorból áll, mint ahány rekordja van az adatbázisfájlnak (beleértve a törlésre kijelölt rekordokat is). Minden sor egy-egy rekordnak felel meg, és két dolgot tartalmaz:

- a kulcskifejezés értékét
- a rekord eredeti sorszámát.

Az indexfájlban a sorok a kulcskifejezés növekvő sorrendjében követik egymást.

A dBASE az indexfájlból következtet arra, hogy a rekordokat milyen sorrendben akarjuk kezelni.

Azt az adatbázisfájlt, amelyhez indexfájlt készítettünk, indexelt adatbázisfájlnak nevezzük.

Példa

A nyitott adatbázisfájl szerkezete és tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>
NEV	C	25	-
SZUL_HELY	C	20	-
SZUL_DATUM	D	8	-
TESTVER	N	2	0
ATLAG	N	4	2
KOLLEGIUM	L	1	-

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
2.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
3.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y
4.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
5.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y

Ha kulcskifejezésként a
TESTVER*10+ATLAG

kifejezést használjuk, akkor a kulcskifejezésre a következő értékek adódnak:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K	A kulcskifejezés értéke
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N	35.00
2.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N	34.25
3.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y	13.00
4.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N	55.00
5.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N	34.75
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y	2.25

Az indexfájl sorai:

A kulcskifejezés értéke	A rekord sorszáma
2.25	6.
13.00	3.
34.25	2.
34.75	5.
35.00	1.
55.00	4.

Az indexfájl tehát ezt a kezelési sorrendet határozza meg:

Első a 6. rekord,
második a 3. rekord,
harmadik a 2. rekord,
negyedik az 5. rekord,
ötödik az 1. rekord,
hatodik a 4. rekord.

75. fejezet

Az indexfájlok azonosítása

Az indexfájlokat (mint általában a fájlakat) azonosítóval kell ellátni.

Az indexfájl nevét annak kell meghatározni, aki az indexfájlt létrehozza.

Az indexfájl kiterjesztése azonban nem lehet tetszőleges, mert a dBASE megköveteli, hogy NDX legyen. Ezért a .NDX a fájlazonosítóból elhagyható.

Példa

Indexfájl azonosítója lehet például TANULO.NDX vagy UZLET.NDX, de nem lehet INDEX.DAT vagy UZLET.TXT.

Ha az indexelés módszerét részletesen meg akarjuk ismerni, akkor az alábbi dolgokat kell megtanulni:

- *indexfájl létrehozása*
- *indexfájlok megnyitása és lezárása*
- *főindexfájl kijelölése*
- *a főindexfájl módosító szerepe*
- *újraindexelés.*

76. fejezet

Indexfájl létrehozása

A nyitott adatbázisfájlhoz kapcsolódó indexfájlt az INDEX paranccsal kell létrehozni.

A parancs formája:

INDEX ON kulcskifejezés TO teljes_fájlazonosító

A TO szót követően a létrehozandó indexfájl teljes azonosítóját kell megadni.

Mivel az indexfájl kiterjesztése kötelezően NDX, ezért a teljes fájlazonosító megadásakor elég a nevét megadni.

A rekordok sorrendjére vonatkozó igényünket az ON szót követő kulcskifejezéssel kell megadni. A kulcskifejezés típusa lehet numerikus, karakteres vagy dátum jellegű, de nem lehet logikai.

A parancs végrehajtása során a kulcskifejezés értéke minden rekordnál kiszámításra kerül, s az indexfájlba ezek az értékek növekvő sorrendben kerülnek bele. Minden érték mellé odaíródik a rekord sorszáma is.

Az indexfájl létrejötte után az indexfájl által meghatározott kezelési sorrend lép érvénybe.

Példa

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
ATLAG	N	4	2	Tanulmányi átlaga
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

Az adatbázisfájl tartalma:

	A tanuló neve	Születési helye	Születési dátuma	T. sz	Tan. átl.	K
1.	Koncz Gabriella	Kecskemét	1980.01.23	3	5.00	N
2.	Agócs Erika	Kecskemét	1979.09.07	3	4.25	N
3.	Lovas Erzsébet	Szabadka	1980.04.03	1	3.00	Y
4.	Szabó Katalin	Kecskemét	1980.05.08	5	5.00	N
5.	Nagy Anett	Kecskemét	1979.11.12	3	4.75	N
6.	Varga Zsuzsanna	Budapest	1979.12.23	0	2.25	Y

a) Ha a rekordokat név szerint növekvő sorrendben akarjuk kezelni, akkor így kell eljárni:

Kulcskifejezésként a

NEV

kifejezést kell megadni.

A létrejövő indexfájl teljes azonosítója A:\KM\NEVSOR.NDX legyen.

Az indexfájlt ezzel a paranccsal hozzuk létre:

```
INDEX ON NEV TO A:\KM\NEVSOR.NDX
```

A parancs végrehajtása után azonnal a NEVSOR.NDX indexfájl hatása érvényesül. Ez azt jelenti, hogy mostantól kezdve a rekordokat név szerint növekvően kezeljük.

Ha tehát ezután például a DISPLAY parancsot alkalmazzuk, akkor a kiírás névsorban történik.

b) Ha a rekordokat a tanulmányi átlag szerint csökkenő sorrendben akarjuk kezelni, akkor kulcskifejezésként az

5-ATLAG

kifejezést kell megadni. Ezzel tudjuk biztosítani a tanulmányi átlag szerint csökkenő sorrendet. Akinek ugyanis a tanulmányi átlaga 5.00, annál a kulcskifejezés értéke 0, akinek pedig 1.00 a tanulmányi átlaga, annál a kulcskifejezés értéke 4. Tehát minél nagyobb valakinek a tanulmányi átlaga, annál kisebb nála a kulcskifejezés értéke.

A létrejövő indexfájl teljes azonosítója A:\KM\JEGYEK.NDX legyen.

Az indexfájlt ezzel a paranccsal hozzuk létre:

```
INDEX ON 5-ATLAG TO A:\KM\JEGYEK
```


A parancs végrehajtása után azonnal a JEGYEK.NDX indexfájl hatása érvényesül. Ez azt jelenti, hogy mostantól kezdve a rekordokat a tanulmányi átlag szerint csökkenő sorrendben kezeljük.

Ha tehát ezután például a DISPLAY parancsot alkalmazzuk, akkor a kiírás a tanulmányi átlag szerint csökkenően történik.

c) Ha kiadjuk az

```
INDEX ON SZUL_HELY+NEV TO A:\KMHELY
```

parancsot, akkor a kulcskifejezést a SZUL_HELY és a NEV mezők összefűzése szolgáltatja, tehát

— elsősorban a születési hely szerint növekvő kezelési sorrend alakul ki, és

— azoknál a rekordoknál, ahol a születési hely megegyezik, ott a név szerint növekvő kezelési sorrend lesz érvényben.

Az indexfájl a gyakorlólemez KM katalógusába kerül, azonosítója pedig HELY.NDX lesz.

d) Ha kiadjuk az

```
INDEX ON TESTVER*10+ATLAG TO A:\KMTTEST
```

parancsot, akkor a kulcskifejezés azt biztosítja, hogy

— elsősorban a testvérek száma szerint növekvő kezelési sorrend alakul ki, és

— azoknál a rekordoknál, ahol a testvérek száma megegyezik, ott a tanulmányi átlag szerint növekvő kezelési sorrend lesz érvényben.

Az indexfájl a gyakorlólemez KM katalógusába kerül TEST.NDX azonosítóval.

77. fejezet

Indexfájlok megnyitása és lezárása

Kívánatos, hogy az adatbázisfájlon végrehajtott módosítások szükség esetén azonnal megjelenjenek az indexfájlokban is.

Ha például az adatbázisfájlba új rekordot toldunk be, akkor elvárjuk, hogy ez a betoldás az indexfájlokban is kifejeződjön.

Az adatbázisfájlban végrehajtott módosítások csak azokban az indexfájlokban kerülnek átvezetésre, amelyek a módosítás pillanatában meg vannak nyitva.

Indexfájlokat a SET INDEX paranccsal nyithatunk meg. A parancs formája:

```
SET INDEX TO [teljes_fajlazonosítók_listája]
```

A TO szó után a megnyitandó indexfájlok teljes azonosítóit kell megadni, egymástól vesszővel elválasztva.

Csak olyan indexfájlok megnyitásának van értelme, amelyek a nyitott adatbázisfájllal kapcsolódnak.

Ha a TO szó után nem adunk meg semmit, akkor az összes nyitott indexfájl lezáródik.

Megjegyzés

Ha az adatbázisfájlt lezárjuk, akkor a hozzá kapcsolódó indexfájlok automatikusan lezáródnak.

Példa

A nyitott adatbázisfájllal kapcsolatban a gyakorlólemez KM katalógusában az alábbi indexfájlok léteznek:

NEVSOR.NDX, JEGYEK.NDX, HELY.NDX, TEST.NDX

Ha ezek közül az első hármat akarjuk megnyitni, akkor ezt a parancsot adjuk ki:

```
SET INDEX TO A:\KM\NEVSOR, A:\KM\JEGYEK, A:\KM\HELY
```

Ha az összes nyitott indexfájlt le akarjuk zárni, akkor ezt a parancsot adjuk ki:

```
SET INDEX TO
```

Megjegyzések

1. Ha a SET INDEX paranccsal megnyitunk egy vagy több indexfájlt, akkor az addig nyitott indexfájlok lezáródnak.
2. A SET INDEX paranccsal egyszerre maximum 7 indexfájlt nyithatunk meg.
3. Ha az INDEX paranccsal létrehozunk egy indexfájlt, akkor az addig nyitott indexfájlok lezáródnak, az újonnan létrehozott indexfájl pedig automatikusan megnyílik.

78. fejezet

Főindexfájl kijelölése

A nyitott indexfájlok közül egyet ki lehet tüntetni.

A kitüntetettség abban nyilvánul meg, hogy a rekordok kezelési sorrendjét ez az indexfájl határozza meg.

A kitüntetett indexfájlt főindexfájlnak nevezzük.

A SET INDEX parancsban legelső helyen álló indexfájl automatikusan főindexfájllá válik.

Példa

Ha kiadjuk a

```
SET INDEX TO A:\KM\NEVSOR, A:\KM\JEGYEK, A:\KM\HELY
```

parancsot, akkor a NEVSOR.NDX indexfájl válik főindexfájllá.

Ha később a megnyitott indexfájlok közül egy másikat akarunk főindexfájllá tenni, akkor a SET ORDER parancsot kell alkalmazni.

A parancs formája:

```
SET ORDER TO [numerikus_konstans]
```

A numerikus konstans határozza meg, hogy a legutolsó SET INDEX parancsban szereplő indexfájlok közül hányadik váljon főindexfájllá.

Példa

Ha a

```
SET INDEX TO A:\KM\NEVSOR, A:\KM\JEGYEK, A:\KM\HELY
```

parancs után kiadjuk a

```
SET ORDER TO 2
```

parancsot, akkor a második helyen álló indexfájl, vagyis a JEGYEK.NDX lesz a főindexfájl.

Megjegyzések

1. Ha kiadjuk a

```
SET ORDER TO 0
```

parancsot, akkor a nyitott indexfájlok nyitva maradnak, de a továbbiakban egyikük sem lesz főindexfájl. Ez azt jelenti, hogy a rekordokat az eredeti sorrendben (vagyis a rekordsorszám szerint) kezeljük.

2. Ha létrehozunk egy indexfájlt, akkor mindaddig az lesz a főindexfájl, amíg egy másikat ki nem jelölünk főindexfájlnak.

79. fejezet

A főindexfájl módosító szerepe

Ha van kijelölve főindexfájl, akkor módosul a hatáskör szerepe, és módosul számos parancs működése is.

Vegyük sorra ezeket a módosulásokat.

a) A GO és a SKIP parancs módosulása

Ha a GO parancsban a TOP vagy a BOTTOM szavakat használjuk, akkor a főindexfájl szerint első, illetve utolsó rekord válik aktuálissá.

Ha viszont a GO parancsban numerikus konstanst adunk meg, akkor ez a szám továbbra is az eredeti rekordsorszámra utal.

A SKIP parancs a főindexfájl szerinti sorrendben kezeli a rekordokat.

Példa

A főindexfájl ezt a kezelési sorrendet határozza meg:

Első a 7. rekord,
második a 6. rekord,
harmadik a 10. rekord,
negyedik az 1. rekord,
ötödik a 9. rekord,
hatodik a 3. rekord,
hetedik a 2. rekord,
nyolcadik a 8. rekord,
kilencedik a 4. rekord,
tizedik az 5. rekord.

a) A

GO TOP

parancs a 7. rekordot teszi aktuálissá, mert a főindexfájl szerint ez az első rekord.

b) A

GO BOTTOM

parancs az 5. rekordot teszi aktuálissá, mert a főindexfájl szerint ez az utolsó rekord.

c) A

GO 10

parancs a 10. rekordot teszi aktuálissá. (Ez a rekord a főindexfájl szerint a harmadik helyen áll.)

d) Ha jelenleg a 10. rekord az aktuális, akkor a

SKIP 5

parancs hatására a 8. rekord válik aktuálissá, mert a főindexfájl szerint ő a 10. rekordot követő ötödik rekord.

b) A hatáskör módosulása

A hatáskör szerepe így módosul:

A

RECORD p

most is az eredeti rekordsorszám szerinti p-edik rekordot jelenti.

A

NEXT p

az aktuális rekordot, meg az azt követő p-1 darab rekordot jelenti. Most azonban a főindexfájl határozza meg a rekordok sorrendjét.

A

REST

az aktuális rekordot, meg az azt követő rekordokat jelenti. Most azonban a főindexfájl határozza meg a rekordok sorrendjét.

Az

ALL

most a főindexfájl szerinti sorrendben jelenti az összes rekordot.

Ha egy parancsban a NEXT p vagy a REST vagy az ALL hatáskör valamelyike van megadva, akkor a parancs a rekordokat a főindexfájl által meghatározott sorrend szerint kezeli:

— a DISPLAY és a LIST abban a sorrendben írja ki a tartalmukat

— az EDIT és a CHANGE abban a sorrendben teszi lehetővé a tartalmuk megváltoztatását

— a DELETE és a RECALL abban a sorrendben végzi el a törlésre való kijelölésüket (illetve a törlésre való kijelöltség megszüntetését).

— a LOCATE és a CONTINUE abban a sorrendben végzi a keresést.

Példa

A főindexfájl ezt a kezelési sorrendet határozza meg:

Első a 7. rekord,
második a 6. rekord,
harmadik a 10. rekord,
negyedik az 1. rekord,
ötödik a 9. rekord,
hatodik a 3. rekord,
hetedik a 2. rekord,
nyolcadik a 8. rekord,
kilencedik a 4. rekord,
tizedik az 5. rekord.

a) Ha a 8. rekord az aktuális, akkor a
DELETE REST
parancs a 8., a 4. és az 5. rekordot jelöli ki törlésre.

b) Ha a 6. rekord az aktuális, akkor a
LIST NEXT 3
parancs a 6., a 10. és az 1. rekord tartalmát írja ki.

c) A
DISPLAY ALL
parancs az összes rekord tartalmát kiírja, de a főindexfájl által meghatározott sorrendben.

c) A BROWSE parancs módosulása

A BROWSE parancs működése is módosul: a főindexfájl által meghatározott sorrendben kezeli a rekordokat.

80. fejezet Újraindexelés

Előfordulhat, hogy az adatbázisfájl tartalmát olyankor módosítottuk, amikor nem volt nyitva egy vagy több indexfájl.

Ilyenkor szükségessé válik, hogy ezeket az indexfájlokat kijavítsuk. Ez abból áll, hogy az indexfájlt újra létrehozzuk az eredeti kulcskifejezéssel.

Az indexfájloknak az eredeti kulcskifejezéssel történő újbóli létrehozását újraindexelésnek nevezzük.

Az újraindexelést a REINDEX paranccsal lehet elvégezni. A parancs formája:

REINDEX

A parancs a nyitott indexfájlokat hozza létre újból. A parancs kiadásakor nyitottnak kell lenni magának az adatbázisfájlnak is.

Példa

A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájlhoz az alábbi indexfájlok kapcsolódnak:

NEVSOR.NDX, JEGYEK.NDX, HELY.NDX, TEST.NDX

Az indexfájlok is a KM katalógusban vannak.

Ha a JEGYEK.NDX és a TEST.NDX nem volt nyitva, amikor módosítottuk a TANULO.DBF adatbázisfájlt, akkor újraindexelésük érdekében a következő parancsokat adjuk ki:

a) Az adatbázisfájl megnyitása:

```
USE A:\KMTANULO.DBF
```

b) Az indexfájlok megnyitása:

```
SET INDEX TO A:\KM\JEGYEK, A:\KM\TEST
```

c) Újraindexelés:

```
REINDEX
```

81. fejezet

A rendezés és az indexelés összehasonlítása

A rendezésnek két hátránya van az indexeléshez képest:

— A rendezett adatbázisfájl általában több helyet foglal el a lemezen, mint az indexfájl.

— Az eredeti adatbázisfájl tartalmának módosítása után mindig újra létre kell hozni a rendezett adatbázisfájlt.

Példa

Tegyük fel, hogy a C: meghajtó DBASE katalógusának van egy ADAT nevű alkatalógusa, és abban van a DOLGOZO.DBF adatbázisfájl, amelynek szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
ALAPBER	N	6	0	Havi alapbére
POTLEK	N	6	0	Havi bérpótléka
LEVONAS	N	6	0	Havi levonása
EGYETEM	L	1	-	Végzett-e egyetemet?
VEZETO	L	1	-	Vezető beosztásban dolgozik-e?

Ki akarjuk írni előbb névsor szerint, majd az alapbérek növekvő sorrendjében az egyetemet végzett vezető beosztású dolgozók nevét és havi járandóságát.

a) A kiíratás rendezéssel így valósítható meg:

1) Az eredeti adatbázisfájl megnyitása:

```
USE C:\DBASEVADAT\DOLGOZO.DBF
```

2) A névsor szerint rendezett adatbázisfájl létrehozása:

```
SORT ON NEV FOR EGYETEM .AND. VEZETO TO C:\DBASEVADAT\NEVR.DBF
```

3) A rendezett adatbázisfájl megnyitása:

```
USE C:\DBASEVADAT\NEVR.DBF
```

4) A kiíratás:

```
DISPLAY ALL NEV, ALAPBER+POTLEK-LEVONAS
```

5) Az alapbér szerint rendezett adatbázisfájl létrehozása:

```
SORT ON ALAPBER FOR EGYETEM .AND. VEZETO TO C:\DBASEVADAT\ALAPR
```

6) A rendezett adatbázisfájl megnyitása:

```
USE C:\DBASEVADAT\ALAPR.DBF
```

7) A kiíratás:

```
DISPLAY ALL NEV, ALAPBER+POTLEK-LEVONAS
```

8) Lezárás és a rendezett adatbázisfájlok törlése:

```
USE
```

```
ERASE C:\DBASEVADAT\NEVR.DBF
```

```
ERASE C:\DBASEVADAT\ALAPR.DBF
```

b) A kiíratás indexeléssel így valósítható meg:

1) Az eredeti adatbázisfájl megnyitása:

```
USE C:\DBASEVADAT\DOLGOZO.DBF
```

2) A név szerint növekvő kezelési sorrendet biztosító indexfájl létrehozása:

```
INDEX ON NEV TO C:\DBASEVADAT\NEVSOR.NDX
```

3) Az alapbér szerint növekvő kezelési sorrendet biztosító indexfájl létrehozása:

```
INDEX ON ALAPBER TO C:\DBASEVADAT\FIZETES
```

4) Az indexfájlok megnyitása:

```
SET INDEX TO C:\DBASEVADAT\NEVSOR, C:\DBASEVADAT\FIZETES.NDX
```

(Ennek hatására a NEVSOR.NDX lesz a főindexfájl, tehát a név szerint növekvő kezelési sorrend lesz érvényes.)

5) Kiíratás név szerint növekvő sorrendben:

```
DISPLAY ALL NEV, ALAPBER+POTLEK-LEVONAS FOR EGYETEM. AND. VEZETO
```


6) A főindexfájl megváltoztatása:

SET ORDER TO 2

(Ennek hatására a FIZETES.NDX lesz a főindexfájl, tehát az alapbér szerint növekvő kezelési sorrend lesz érvényes.)

7) Kiíratás alapbér szerint növekvő sorrendben:

DISPLAY ALL NEV, ALAPBER+POTLEK-LEVONAS FOR EGYETEM. AND. VEZETO

82. fejezet

Gyorskeresés az adatbázisfájlban

Indexelt adatbázisfájlban a FIND parancs igen gyorsan megtalálja a főindexfájl kulcskifejezése alapján keresett rekordot.

A FIND parancs azonban csak akkor tud keresni, ha olyan főindexfájl van kijelölve, amely karakteres vagy numerikus kulcskifejezéssel lett létrehozva.

A parancs formája:

FIND karakteres_konstans|numerikus_konstans

(Attól függően kell karakteres vagy numerikus konstans megadni, hogy a főindexfájl karakteres vagy numerikus kulcskifejezéssel lett-e létrehozva.)

A parancs a legelső olyan rekordot teszi aktuálissá, amelynél a kulcskifejezés értéke a megadott konstanssal (numerikus esetben pontosan) egyenlő.

Ha több ilyen rekord is van, akkor — a főindexfájl által meghatározott kezelési sorrend miatt — azok egymás után helyezkednek el.

Példák

A nyitott adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjsz.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
ALAPBER	N	6	0	Havi alapbére
POTLEK	N	6	0	Havi bérpótléka
LEVONAS	N	6	0	Havi levonása
EGYETEM	L	1	-	Végzett-e egyetemet?
VEZETO	L	1	-	Vezető beosztásban dolgozik-e?

1. Ha meg akarjuk keresni azokat a dolgozókat, akiknek 20000 Ft a havi alapbére, akkor a következőképpen kell eljárni:

a) Olyan indexfájlt kell létrehozni, amelyben az ALAPBER a kulcskifejezés:

```
INDEX ON ALAPBER TO A:\KMMALAPR.NDX
```

(Ennek hatására automatikusan az ALAPR.NDX válik főindexfájllá.)

b) A FIND parancsban a 20000-et kell megadni:

```
FIND 20000
```

(Ha több olyan dolgozó is van, akinek 20000 Ft a havi alapbére, akkor ezek sorban egymás után helyezkednek el — a jelenleg érvényes kezelési sorrend miatt —, ezért könnyű őket áttekinteni.)

2. Ha meg akarjuk keresni azokat a dolgozókat, akiket Kovács Bélának hívnak, akkor a következőképpen kell eljárni:

a) Olyan indexfájlt kell létrehozni, amelyben a NEV a kulcskifejezés:

```
INDEX ON NEV TO A:\KMNEVREN.NDX
```

(Ennek hatására automatikusan a NEVREN.NDX válik főindexfájllá.)

b) A FIND parancsban a „Kovács Béla” konstanst kell megadni:

```
FIND "Kovács Béla"
```

(Nem kell kiírni a nevet pontosan, elég az elejéből annyi is, ami a megtaláláshoz kell. Így pl. elég a "Kovács Bél" karaktersorozat megadása is.)

83. fejezet

A munkaterület fogalma

Az adatbázis-kezeléshez tíz munkaterület áll rendelkezésünkre. Az 1-es, a 2-es, és így tovább. Mi eddig mindig az 1-es munkaterületet használtuk.

Bármely munkaterületen megtehetjük mindazt, amit eddig az 1-es munkaterületen végrehajtottunk. (Ugyanaz az adatbázisfájl azonban egyidejűleg csak egyetlen munkaterületen lehet nyitva.)

84. fejezet

A munkaterület azonosítója

A munkaterületek azonosítása a megkülönböztetésüket szolgálja. Az azonosítás háromféleképpen is elvégezhető:

- az 1-től 10-ig terjedő egész számokkal
- az A-tól J-ig terjedő betűkkel
- az illető munkaterületen megnyitott adatbázisfájl nevével.

Példa

Ha az 1-es munkaterületen a TANULO.DBF adatbázisfájl van megnyitva, akkor az 1-es munkaterület azonosításának lehetőségei:

```
1
A
TANULO
```

85. fejezet

Az aktuális munkaterület fogalma

Azt a munkaterületet, amelyet jelenleg használunk, aktuális munkaterületnek nevezük.

Példa

Eddig mindig az 1-es volt az aktuális munkaterület.

86. fejezet

Munkaterület váltása

Ha ezután másik munkaterületet akarunk használni, akkor azt a munkaterületet kell aktuálissá tenni (úgy is fogalmazhatunk, hogy munkaterületet kell váltani).

A váltást a SELECT paranccsal kell elvégezni.

A parancs formája:

SELECT munkaterület-azonosító

Ha tehát a SELECT szó után begépeljük a kívánt munkaterület azonosítóját, akkor máris az a munkaterület válik aktuálissá.

A képernyőn külön jel nem mutatja, hogy melyik az aktuális munkaterület.

Példa

Jelenleg az 1-es munkaterületet használjuk, és ott meg van nyitva a TANULO.DBF adatbázisfájl.

Ha ezután a 2-es munkaterületet akarjuk használni, akkor ezt a parancsot adjuk ki:

```
SELECT 2
```

vagy

```
SELECT B
```

Ha ezután kiadjuk az
USE A:\KMKKONYV.DBF

parancsot, akkor a KONY.DBF adatbázisfájl a 2-es munkaterületen nyílik meg.

Ha újra az 1-es munkaterületet akarjuk használni, akkor az alábbi parancsok valamelyikét adjuk ki:

```
SELECT 1  
SELECT A  
SELECT TANULO
```

Ha ezután ismét a 2-es munkaterületet akarjuk használni, akkor azt most már háromféleképpen is aktuálissá tehetjük:

```
SELECT 2  
    vagy  
SELECT B  
    vagy  
SELECT KONYV
```

Feladat

Gondoskodj róla, hogy a 3-as munkaterületen az ISMEROS.DBF, a 4-esen pedig a POSTA.DBF adatbázisfájl legyen nyitva! (Mindkét adatbázisfájl a gyakorlólemez KM katalógusában található!)

87. fejezet

Az aktív adatbázisfájl fogalma

Az aktuális munkaterületen megnyitott adatbázisfájlt aktív adatbázisfájlnak nevezük.

Megjegyzés

Amikor még csak az 1-es munkaterületet használtuk, akkor mindig az ott megnyitott adatbázisfájl volt az aktív adatbázisfájl.

Eddig azt mondtuk, hogy a DISPLAY STRUCTURE, a LIST STRUCTURE, a GO, a SKIP, az APPEND, az INSERT, a DISPLAY, a LIST, az EDIT, a CHANGE, a BROWSE, a DELETE, a RECALL, a PACK, a ZAP, a ?, a REPLACE, a SUM, az AVERAGE, a COUNT, a MODIFY STRUCTURE, a LOCATE, a CONTINUE, a SORT, az INDEX, a SET INDEX, a SET ORDER és a REINDEX parancsok a nyitott adatbázisfájltra vonatkoznak. Most már elmondhatjuk: ezek a parancsok mindig az aktív adatbázisfájltra vonatkoznak.

Példa

Jelenleg ezek az adatbázisfájlok vannak nyitva:

A munkaterület sorszáma	A megnyitott adatbázisfájl neve
1	TANULO
2	KONYV

Ha kiadjuk a

```
SELECT 2
```

parancsot, akkor a KONYV.DBF lesz az aktív adatbázisfájl.

Ha a TANULO.DBF-et akarjuk aktívvá tenni, akkor a

```
SELECT 1
```

parancsot adjuk ki.

Feladatok

1. Tedd aktívvá a 3-as munkaterületen megnyitott ISMEROS.DBF adatbázisfájlt!
2. Tedd aktívvá a 4-es munkaterületen megnyitott POSTA.DBF adatbázisfájlt!

88. fejezet

Egyszerű és összetett mezőnév

a) Egyszerű mezőnév

Ha az aktív adatbázisfájl mezőjére akarunk hivatkozni, akkor csupán a mező nevét kell leírni.

Példa

Amikor még csak az 1-es munkaterületet használtuk, akkor az ott megnyitott adatbázisfájl volt az aktív adatbázisfájl, tehát a mezőkre egyszerűen a mezőnévvel hivatkoztunk.

b) Összetett mezőnév

Ha nem az aktív adatbázisfájl mezőjére, hanem egy másik munkaterületen megnyitott adatbázisfájl mezőjére akarunk hivatkozni, akkor előbb meg kell adni az adatbázisfájl nevét, majd a -> jeleket, végül a mező nevét.

Példa

Jelenleg ezek az adatbázisfájlok vannak megnyitva:

A munkaterület sorszáma	A megnyitott adatbázisfájl neve
1	TANULO
2	KONYV

A 2-es az aktuális munkaterület.

a) A TANULO.DBF adatbázisfájl ATLAG mezőjére összetett mezőnévvel hivatkozhatunk:

TANULO->ATLAG

b) A KONYV.DBF adatbázisfájl CIM mezőjére egyszerű mezőnévvel hivatkozhatunk:

CIM

c) Ha ki akarjuk írni a TANULO.DBF adatbázisfájl 7. rekordjából a NEV mező tartalmát, és a KONYV.DBF adatbázisfájl 9. rekordjából a CIM mező tartalmát, akkor így járunk el:

1) Aktuálissá tesszük az 1-es munkaterületet:

SELECT 1

2) Aktuálissá tesszük a 7. rekordját:

GO 7

3) A 2-es munkaterületet tesszük aktuálissá:

SELECT 2

4) Aktuálissá tesszük a 9. rekordját:

GO 9

5) Kiíratjuk a mezők tartalmát:

DISPLAY TANULO -> ATLAG, CIM

Feladat

Írased ki az ISMEROS.DBF adatbázisfájl 4. rekordjából az IRSZ mező tartalmát, a POSTA.DBF adatbázisfájl 2. rekordjából pedig a TELEPULES mező tartalmát!

89. fejezet

Adatbázisfájlok kapcsolata

Az adatbázisfájlok közötti kapcsolatot egy feladaton keresztül mutatjuk be.

Feladat

Az ISMEROS.DBF és a POSTA.DBF adatbázisfájlok a gyakorlólemez KM katalógusában helyezkednek el.

Az ISMEROS.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tíjst.</i>	<i>Jelentés</i>
NEV	C	20	-	Az ismerős neve
SZUL_DATUM	D	8	-	Születési dátuma
IRSZ	N	4	0	Irányítószám
UTCA	C	12	-	Melyik utcában lakik?
HAZSZAM	C	6	-	Hány szám alatt?

A POSTA.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tíjst.</i>	<i>Jelentés</i>
SZAM	N	4	0	Irányítószám
TELEPULES	C	20	-	A település neve

a) Gondoskodj róla, hogy ezek az adatbázisfájlok az alábbi munkaterületeken legyenek nyitva:

A munkaterület sorszáma	A megnyitott adatbázisfájl neve
3	ISMEROS
4	POSTA

b) Tedd aktuálissá a 3-as munkaterületet, majd ott

— tedd aktuálissá az 5. rekordot

— írasd ki az 5. rekordból az IRSZ mező tartalmát!

c) Most tedd aktuálissá a 4-es munkaterületet, majd ott

— hozd létre azt az indexfájt, amely biztosítja a POSTA.DBF adatbázisfájl irányítószámok szerint növekvő kezelési sorrendjét

— keresd meg (tedd aktuálissá) azt a rekordot, amelyben ugyanaz az irányítószám szerepel, mint az ISMEROS.DBF adatbázisfájl 5. rekordjában!

d) Végül tedd aktuálissá ismét a 3-as munkaterületet, és

— írasd ki az 5. rekordban szereplő ismerősöd nevét, és azt is, hogy melyik településen lakik!

A fenti feladatban az irányítószám segítségével kapcsolatot teremtettünk két adatbázisfájl között.

Az irányítószám mindkét adatbázisfájlban szerepel. Az egyik adatbázisfájlban az IRSZ nevű mezőben szerepel, a másikban pedig a SZAM nevű mezőben.

Az ISMEROS.DBF adatbázisfájlban azért van benne az irányítószám, hogy megtudjuk az ismerősünk irányítószámát.

A POSTA.DBF adatbázisfájlban azért van benne az irányítószám, hogy megtudjuk a neki megfelelő település nevét.

Az irányítószám tehát alkalmas volt arra, hogy a két adatbázisfájl között kapcsolatot teremtsünk.

A kapcsolatot úgy hoztuk létre, hogy az egyik adatbázisfájl aktuális rekordjából kiértük az irányítószámot, s a másik adatbázisfájlban megkerestük azt a rekordot, amelyben az irányítószám ugyanaz.

Azt a kifejezést, amely alkalmas arra, hogy két adatbázisfájl között kapcsolatot hozunk létre, kapcsoló-kifejezésnek nevezzük.

Példa

Az előző feladatban az ISMEROS.DBF adatbázisfájlban a kapcsoló-kifejezés ez volt:

IRSZ

a POSTA.DBF adatbázisfájlban pedig ez volt a kapcsoló-kifejezés:

SZAM

Megjegyzés

Az előző feladat megoldása során nem jött létre tartós kapcsolat a két adatbázisfájl között.

Ezt úgy kell érteni, hogy ha később a 3-as munkaterületen egy másik rekordot tettünk volna aktuálissá, akkor a 4-es munkaterületen nem vált volna automatikusan aktuálissá az a rekord, amelyben az irányítószám ugyanaz, mint a 3-as munkaterület aktuális rekordjában.

90. fejezet

Két adatbázisfájl összekapcsolása

Két adatbázisfájl összekapcsolása azt jelenti, hogy a kapcsoló-kifejezések segítségével tartós kapcsolat alakul ki az adatbázisfájlok között.

Az összekapcsolt adatbázisfájlok nem egyenrangúak. Az egyiket vezérnek, a másikat pedig szolgának nevezzük.

A kapcsolat tartóssága azt jelenti, hogy a vezérnek bármelyik rekordját tesszük is aktuálissá, a szolgánál automatikusan mindig az a rekord válik aktuálissá, amely biztosítja, hogy a kapcsoló-kifejezések értéke a két adatbázisfájlnál megegyezik.

Adatbázisfájlokat a SET RELATION paranccsal kapcsolunk össze.

Mielőtt a SET RELATION parancsot kiadnánk, az alábbi teendőket el kell végezni:

- A vezért is meg a szolgát is meg kell nyitni egy-egy munkaterületen.
- A szolgálal kapcsolatban létre kell hozni egy olyan indexfájlt, amely a kapcsoló-kifejezés szerint növekvő kezelési sorrendet biztosítja. Ezt az indexfájlt főindexfájllá kell tenni.

— A vezért kell megtenni aktív adatbázisfájlnak.

A SET RELATION parancs formája:

SET RELATION TO kapcsoló-kifejezés INTO fájlnev

A TO szót követően a vezér kapcsoló-kifejezését, az INTO szót követően pedig a szolga nevét kell megadni.

Példa

Az ISMEROS.DBF és a POSTA.DBF adatbázisfájlok a gyakorlólemez KM katalógusában helyezkednek el.

Az ISMEROS.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjcs.</i>	<i>Jelentés</i>
NEV	C	20	-	Az ismerős neve
SZUL_DATUM	D	8	-	Születési dátuma
IRSZ	N	4	0	Hánytűszám
UTCA	C	12	-	Melyik utcában lakik?
HAZSZAM	C	6	-	Hány szám alatt?

A POSTA.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tízs.</i>	<i>Jelentés</i>
SZAM	N	4	0	Irányítószám
TELEPULES	C	20	-	A település neve

Úgy akarjuk összekapcsolni a két adatbázisfájlt, hogy az ISMEROS.DBF legyen a vezér, a POSTA.DBF pedig a szolgál.

A kapcsoló-kifejezés a vezérnél IRSZ, a szolgánál pedig SZAM lesz.

Az összekapcsolást a következőképpen valósítjuk meg:

a) Aktuálissá tesszük a 3-as munkaterületet:

```
SELECT 3
```

b) A 3-as munkaterületen megnyitjuk a vezért:

```
USE A:\KM\ISMEROS.DBF
```

c) Aktuálissá tesszük a 4-es munkaterületet:

```
SELECT 4
```

d) A 4-es munkaterületen megnyitjuk a szolgát:

```
USE A:\KM\POSTA.DBF
```

e) A szolgálal kapcsolatban létrehozuk azt az indexfájlt, amely a kapcsoló-kifejezés szerint növekvő kezelési sorrendet biztosítja:

```
INDEX ON SZAM TO A:\KM\IRANYR.NDX
```

(A most létrejött IRANYR.NDX automatikusan főindexfájl lett.)

f) A vezért tesszük meg aktív adatbázisfájlnak:

```
SELECT 3
```

g) Elvégezzük az összekapcsolást:

```
SET RELATION TO IRSZ INTO POSTA
```

Ha ezután a vezérnek más és más rekordja válik aktuálissá, akkor a szolgál ehhez alkalmazkodik: automatikusan az a rekordja válik aktuálissá, amely biztosítja, hogy a vezér aktuális rekordjában szereplő irányítószám és a szolgál aktuális rekordjában szereplő irányítószám megegyezik.

Így elértük azt, hogy a szolgál mindig a vezér aktuális rekordjában levő irányítószámnak megfelelő településnevet szolgáltatja.

Az összekapcsolás lehetőséget ad olyan kiíratásra, amelynél az ismerőseink adatai között a településnév is megjelenik.

Ha például az ismerőseink nevét és a településük nevét akarjuk kiíratni, akkor ezt a parancsot adjuk ki:

```
LIST NEV, POSTA->TELEPULES
```


Feladat

A gyakorlólemez KM katalógusában levő ISMEROS.DBF és POSTA.DBF adatbázisfájlok felhasználásával írasd ki a képernyőre az összes ismerősöd nevét, születési dátumát és a lakcímét úgy, hogy ne csak az irányítószám, hanem a településnév is szerepeljen benne!

91. fejezet

Adatbázisfájlok együttes tervezése

Ha az adatbázisfájl tervezésekor kiderül, hogy valamely (terjedelmes) mező értékei gyakran ismétlődnek, akkor ezt a mezőt érdemes külön adatbázisfájlban elhelyezni. Ott minden értéket csak egyszer kell szerepeltetni, s (rövid) kóddal kell ellátni.

A kódot mindkét adatbázisfájlban el kell helyezni, és vele mint kapcsoló-kifejezéssel össze kell kapcsolni a két adatbázisfájlt.

Példa

Az ISMEROS.DBF adatbázisfájl tervezésekor nem vettük fel az adatbázisfájlba a településnevet, mert terjedelmes (20 karakternyi), és az értékei várhatóan gyakran ismétlődnek.

A településneveket külön, a POSTA.DBF adatbázisfájlban szerepeltettük. Ott mind-egyiket csak egyszer adtuk meg, s kóddal láttuk el (az irányítószám volt a kód).

A kód szerepelt az ISMEROS.DBF és a POSTA.DBF adatbázisfájlban is, és ez lehetővé tette az adatbázisfájlok összekapcsolását.

Megjegyzés

Gondoljuk meg, hogy a fenti módszerrel mennyi gépelési munkát és a lemezen mennyi helyet megtakaríthatnánk, ha nem csak az ismerőseinket, hanem mondjuk egy megye népességét tartanánk nyilván.

92. fejezet

Több adatbázisfájl összekapcsolása

Egy vezérhez egyidejűleg csak egy szolgát kapcsolhatunk. A szolga azonban lehet vezére egy harmadik (az ő számára szolgálai szerepet betöltő) adatbázisfájlnak. (Ezt a harmadik adatbázisfájlt természetesen egy harmadik munkaterületen kell megnyitni.)

Ily módon több (maximum 8) adatbázisfájlt kapcsolhatunk össze, vagyis egy láncolatot alakíthatunk ki. Csak arra kell vigyázni, hogy a lánc ne záruljon be, vagyis a láncban ne szerepeljen kétszer ugyanaz az adatbázisfájl.

Példa

A FORGALOM.DBF adatbázisfájl egy áruház napi forgalmi adatait tartalmazza. Szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
F_CIKK	C	4	-	Cikkszám
MENNYISEG	N	9	2	Eladott mennyiség

A cikkszám az áru azonosítására szolgál.

Az ARU.DBF adatbázisfájlból lehet megtudni, hogy a cikkszám melyik árut jelöli, ennek mennyi az egységára, és mi a mértékegysége. Ennek szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
CIKK	C	4	-	Cikkszám
ARU_NEV	C	20	-	Az áru neve
EGYSEGAR	N	8	2	Egységára
A_MEKOD	C	2	-	Mértékegységkód

Mint látható, az ARU.DBF adatbázisfájlban a mértékegység kóddal van megadva.

A MERTEK.DBF adatbázisfájlból lehet megtudni, hogy a mértékegységkód melyik mértékegységet jelöli. Ennek szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
MEKOD	C	2	-	Mértékegységkód
ME_NEV	C	10	-	A mértékegység neve

A FORGALOM.DBF és az ARU.DBF adatbázisfájlok a cikkszám segítségével kapcsolhatók össze (kettejük viszonyában a FORGALOM.DBF lesz a vezér, az ARU.DBF pedig a szolga).

Az ARU.DBF és a MERTEK.DBF adatbázisfájlok a mértékegységkód segítségével kapcsolhatók össze (kettejük viszonyában az ARU.DBF lesz a vezér, a MERTEK.DBF pedig a szolga).

Ha mindhárom adatbázisfájl a C: meghajtó DBASE katalógusának ADAT alkatalógusában lenne elhelyezve, akkor a három adatbázisfájlt így kapcsolnánk össze:

a) Az adatbázisfájlok megnyitása:

```
SELECT 1
USE C:\DBASEVADAT\FORGALOM.DBF
SELECT 2
USE C:\DBASEVADAT\ARU.DBF
SELECT 3
USE C:\DBASEVADAT\MERTEK.DBF
```

b) A FORGALOM.DBF és az ARU.DBF adatbázisfájlok összekapcsolása:

```
SELECT 2
INDEX ON CIKK TO C:\DBASEVADAT\R1.NDX
SELECT 1
SET RELATION TO F_CIKK INTO ARU
```

c) Az ARU.DBF és a MERTEK.DBF adatbázisfájlok összekapcsolása:

```
SELECT 3
INDEX ON MEKOD TO C:\DBASEVADAT\R2.NDX
SELECT 2
SET RELATION TO A_MEKOD INTO MERTEK
```

A három adatbázisfájl most már össze van kapcsolva.

Ha ezután ki akarnánk írni a napi forgalmi adatokat úgy, hogy az áru neve, értéke és a mértékegység neve is ki legyen írva, akkor ezeket a parancsokat adnánk ki:

```
SELECT 1
LIST ARU->ARU_NEV, MENNYISEG*ARU->EGYSEGAR, MERTEK->ME_NEV
```

Ha pedig ki akarnánk számolni a napi forgalom összértékét, akkor ezeket a parancsokat adnánk ki:

```
SELECT 1
SUM MENNYISEG*ARU->EGYSEGAR
```

93. fejezet

Az összekapcsoltság megszüntetése

Ha két adatbázisfájl összekapcsoltságát meg akarjuk szüntetni, akkor a vezéret kell aktívvá tenni, majd ezt a parancsot kell kiadni:

```
SET RELATION TO
```

Példa

Ha olyan összekapcsoltságot akarunk megszüntetni, amelyben a vezér a 2-es munkaterületen van megnyitva, akkor ezeket a parancsokat adjuk ki:

```
SELECT 2
```

```
SET RELATION TO
```

94. fejezet A függvény fogalma

Az adatbázis-kezelés lehetőségeinek szélesítése érdekében ismerkedjünk meg a függvény fogalmával!

A függvény olyan eszköz, amelynek segítségével bizonyos kifejezések (az úgynevezett argumentumok) értékéből meghatározott szabályok szerint egy új értéket nyerünk.

Az így nyert értéket a függvény értékének nevezzük.

A függvényeket a nevük alapján különböztetjük meg egymástól.

Minden függvénynél elő van írva, hogy használatakor hány argumentumot kell (vagy lehet) megadni, és hogy az egyes argumentumoknak milyen típusúaknak kell lenniük. Ha több argumentumot adunk meg, akkor azokat vesszővel kell egymástól elválasztani. Az argumentumot (vagy az argumentumokat) a függvény neve után kitett egyetlen gömbölyű zárójelpárban kell megadni.

Van olyan függvény, amelynek nincs argumentuma. A gömbölyű zárójelpárt ilyenkor is ki kell tenni a függvény neve után, noha semmit nem tartalmaz.

Példák

1. A SUBSTR nevű függvénynek három argumentuma lehet. Ha az első argumentuma NEV, a második 4, a harmadik pedig 12, akkor a függvényt így kell megadni:

```
SUBSTR(NEV,4,12)
```

(Majd tanuljuk, hogy a függvény értéke a NEV mezőben szereplő karaktersorozat 4. karakterétől számított 12 karakter.)

2. Egy másik függvény neve SQRT. Ennek csak egyetlen argumentuma lehet. Ha az argumentum 16, akkor a függvényt így kell megadni:

```
SQRT(16)
```

(Majd tanuljuk, hogy a függvény értéke a 16 négyzetgyöke.)

3. Egy harmadik függvény neve DATE. Ennek nem lehet argumentuma. Ezt a függvényt így kell megadni:

```
DATE()
```

(Majd tanuljuk, hogy a függvény értéke a mai dátum.)

95. fejezet

A függvény típusa

A függvény típusát az értéke határozza meg.

Ilyen alapon különböztetünk meg

- numerikus
- dátum jellegű
- karakteres
- logikai

függvényeket.

Példák

1. A SUBSTR karakteres függvény, mert az értéke mindig egy karaktersorozat.
2. Az SQRT numerikus függvény, mert az értéke mindig egy szám.
3. A DATE dátum jellegű függvény, mert az értéke mindig egy dátum.

96. fejezet

Függvény a kifejezésben

Eddig azt mondtuk, hogy a műveletek operandusai konstansok vagy mezőnevek lehetnek. Most már elmondhatjuk: a függvények értékével is lehet műveleteket végezni.

Ebből következik, hogy a kifejezésben függvények is szerepelhetnek.

Fogadjuk el, hogy egy önmagában álló függvény is kifejezés.

Példa

Függvényt is tartalmazó kifejezések:

- a) SQRT(16)
- b) (SQRT(16)+7)/2
- c) SQRT(16)+SQRT(25)

Megjegyzés

Mivel a függvény argumentumai kifejezések, kifejezésben pedig függvény is szerepelhet, ezért elmondhatjuk, hogy egyik függvény a másikba behelyezhető.

Példák

Egymásba helyezett függvények:

- a) SUBSTR(NEV,SQRT(9)+1,12)
- b) SQRT(SQRT(16)+(SQRT(25)))

97. fejezet

A függvény formája

A függvény formája egy olyan képlet, amely szemléletesen mutatja a függvény nevét, valamint argumentumainak számát és típusát.

Ebben a képletben a függvény nevét nagybetűkkel, argumentumait pedig kisbetűkkel jelöljük.

Az argumentumokra olyan jelölést alkalmazunk, amelyből azok típusa is kiderül:

- a numerikus argumentumot **n**-nel
- a dátum jellegű argumentumot **d**-vel
- a karakteres argumentumot **c**-vel
- a logikai argumentumot **l**-lel

jelöljük.

Ha egy argumentum többféle típusú is lehet, akkor azt **t**-vel jelöljük.

Ha ugyanabban a függvényben két azonos típusú argumentum is szerepel, akkor azokat a jel mögé írt számokkal fogjuk egymástól megkülönböztetni.

Ha például ugyanabban a függvényben három argumentum szerepel, és közülük az elsőnek karakteres kifejezésnek, a másodiknak és a harmadiknak pedig numerikus kifejezésnek kell lenni, akkor az argumentumokat így jelöljük majd: **c,n1,n2**.

Ha valamely argumentum szerepelhet a függvényben, de nem kötelező szerepelnie, azt szögletes zárójelbe fogjuk tenni.

Példák

1. A SUBSTR függvénynek három argumentuma lehet: az első egy karakteres kifejezés, a második és a harmadik pedig egy-egy numerikus kifejezés. A harmadik argumentum elhagyható.

Tehát a SUBSTR függvény formája:

SUBSTR(c,n1[,n2])

(Azért tettük szögletes zárójelbe az **n2** előtt álló vesszőt is, mert ha **n2**-t nem adjuk meg, akkor ezt a vesszőt nem kell megadni.)

2. Az SQRT függvénynek egy argumentuma van, és ezt mindig meg kell adni. Az argumentumnak numerikus kifejezésnek kell lenni.

Tehát az SQRT függvény formája:

SQRT(n)

3. Mivel a DATE függvénynek nincs argumentuma, ezért a formája:

DATE()

A függvények ismertetésekor megadjuk majd a függvény

— nevét

— formáját

— típusát,

és elmondjuk majd, hogy az értéke milyen szabályok szerint alakul ki.

98. fejezet

A függvény értékének kiírása

Mivel a függvény maga is egy kifejezés, ezért értékét (éppúgy, mint bármely kifejezés értékét) a ? parancs segítségével írathatjuk ki. Elég, ha a ? után begépeljük a függvényt, s leütjük az Enter billentyűt.

Példák

1. A

SUBSTR(NEV,4,12)

értéke így íratható ki:

? SUBSTR(NEV,4,12)

2. Az

SQRT(16)

értéke így íratható ki:

? SQRT(16)

Ezek után áttekintjük a legfontosabb függvényeket.

99. fejezet

Matematikai függvények

a) Az abszolútérték-függvény

Neve: ABS

Formája: ABS(n)

Típusa: Numerikus

Értéke: Az **n** szám abszolút értéke.

Példák

1) ABS(-41) értéke 41

2) ABS(41) értéke 41

3) Ha meg akarjuk kapni a
254-67*54

kifejezés abszolút értékét, akkor ezt kell leírunk:

ABS(254-67*54)

Ha az eredményt ki akarjuk írni, akkor ezt a parancsot adjuk ki:

? ABS(254-67*54)

b) Az egészrész-függvény

Neve: INT

Formája: INT(n)

Típusa: Numerikus

Értéke: Az **n** szám egészrésze, vagyis a tizedesjegyek levágásával kapott szám.
(Nem egyezik a matematikában használt egészrész-függvénnyel.)

Példák

1) INT(3.45) értéke 3

2) INT(-4.56) értéke -4

3) Ha ki akarjuk írni a
3/7-65/3

kifejezés egészrészét, akkor ezt a parancsot adjuk ki:

? INT(3/7-65/3)

c) A négyzetgyök-függvény

Neve: SQRT

Formája: **SQRT(n)**

Típusa: Numerikus

Értéke: Az **n** szám négyzetgyöke.

Példák

1) SQRT(49) értéke 7

2) Ha ki akarjuk írni a 72 négyzetgyökének egészrészét, akkor előbb az SQRT függvényt, majd az INT függvényt kell alkalmazni:

? INT(SQRT(72))

d) A kerekítő függvény

Neve: ROUND

Formája: **ROUND(n1,n2)**

Típusa: Numerikus

Értéke: Az **n1** szám **n2** számú tizedesjegyre kerekítve.

Példák

1) ROUND(12.78654,2) értéke 12.79

2) ROUND(12.78654,4) értéke 12.7865

3) ROUND(5/3,1) értéke 1.7

4) A 72 négyzetgyökének és a 95 négyzetgyökének összegét 1 tizedesjegy pontossággal így írjuk ki:

? ROUND(SQRT(72)+SQRT(95),1)

e) A maximum-függvény

Neve: MAX

Formája: **MAX(n1,n2)**

Típusa: Numerikus

Értéke: Az **n1** és a **n2** szám közül a nagyobbik.

Példák

- 1) MAX(23,34) értéke 34
- 2) MAX(2/4,20/41) értéke 0.5

f) A minimum-függvény

Neve: MIN

Formája: MIN(n1,n2)

Típusa: Numerikus

Értéke: Az n1 és a n2 szám közül a kisebbik.

Példák

- 1) MIN(23,34) értéke 23
- 2) MIN(2/4,20/39) értéke 0.5

100. fejezet

A matematikai függvények alkalmazása

Példa

Az aktív adatbázisfájlban egy gyár által készített képcsők adatait tartjuk nyilván.

Az adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
TIPUS	C	8	-	A képcső típusa
HOSSZ	N	5	2	A képcső hossza
SZEL	N	5	2	A képcső szélessége
KOLT	N	5	0	A képcső önköltsége
MENNY	N	6	0	Gyártott darabszám

Az aktuális rekord tartalma:

A képcső típusa	A képcső hossza	A képcső szélessége	A képcső önköltsége	Gyártott darabszám
HCM-421E	26.50	18.70	12000	5200

a) Ha ki akarjuk írni a képcső átmérőjét, akkor alkalmazni kell Pythagoras tételét: a szélesség és a hosszúság négyzetösszegéből kell négyzetgyököt vonni:

? SQRT(HOSSZ^2+SZEL^2)

Ha az eredményt egy tizedesjegy pontossággal akarjuk megkapni, akkor alkalmazni kell a ROUND függvényt is:

? ROUND(SQRT(HOSSZ^2+SZEL^2),1)

Az így kapott eredmény 32.4

b) Ha mindazon képcsővek típusát és átmérőjét ki akarjuk írni, amelyekből legalább 1000 darabot gyártottak, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL TIPUS, SQRT(HOSSZ^2+SZEL^2) FOR MENNY >= 1000
```

- c) Ha ki akarjuk írni az összes képcső
- típusát,
 - átmérőjének egységére eső önköltségét,
 - területének egységére eső önköltségét,

akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL TIPUS, KOLT/SQRT(HOSSZ^2+SZEL^2), KOLT/(HOSSZ*SZEL)
```

- d) Ha az összes rekordból a képcső típusát, valamint az
- átmérőjének egységére eső önköltség

és a

- területének egységére eső önköltség

közül a nagyobbiknak az értékét akarjuk kiírni, akkor ezt a parancsot adjuk ki:

```
LIST TIPUS, MAX(KOLT/SQRT(HOSSZ^2+SZEL^2),KOLT/(HOSSZ*SZEL))
```

- e) Ha az összes rekordból a képcső típusát, valamint az
- átmérőjének egységére eső önköltség

és a

- területének egységére eső önköltség

különbségének abszolút értékét akarjuk kiírni, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL TIPUS, ABS(KOLT/SQRT(HOSSZ^2+SZEL^2)-KOLT/(HOSSZ*SZEL))
```

f) Ha az összes rekordból a képcső típusát, valamint a hossza és szélessége hányadosának egészrészét akarjuk kiírni, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL TIPUS, INT(HOSSZ/SZEL)
```

101. fejezet

Karakterkezelő függvények

a) Balrész-függvény

Neve: LEFT

Formája: LEFT(c,n)

Típusa: **Karakteres**

Értéke: A c karaktorsorozat bal oldali n számú karaktere.

Példák

- 1) LEFT("Csacsinszka",6) értéke „Csacsi”
- 2) LEFT("Magyarország",4) értéke „Magy”

b) Jobbrész-függvény

Neve: RIGHT

Formája: RIGHT(c,n)

Típusa: **Karakteres**

Értéke: A c karaktorsorozat jobb oldali n számú karaktere.

Példák

- 1) RIGHT("Csacsinszka",8) értéke „csinszka”
- 2) RIGHT("Magyarország",6) értéke „ország”

c) Középrész-függvény

Neve: SUBSTR

Formája: SUBSTR(c,n1,[n2])

Típusa: **Karakteres**

Értéke: A c karaktorsorozat n1-edik karakterétől számított n2 számú karakter.

Megjegyzés

Ha n2 nincs megadva, akkor végig megkapjuk a karaktereket.

Példák

- 1) SUBSRT(“Csacsinszka”,4,6) értéke „csinsz”
- 2) SUBSRT(“Magyarország”,7,4) értéke „orsz”
- 3) SUBSRT(“Magyarország”,3) értéke „gyarország”

d) Sokszorozó függvény

Neve: REPLICATE

Formája: **REPLICATE(c,n)**

Típusa: **Karakteres**

Értéke: A **c** karaktersorozat **n**-szer megismételve.

Példák

- 1) REPLICATE(“csa”,3) értéke „csacsacs”
- 2) REPLICATE(“*”,10) értéke „*****”

e) Kisbetű-függvény

Neve: LOWER

Formája: **LOWER(c)**

Típusa: **Karakteres**

Értéke: A **c** karaktersorozat úgy, hogy a nagybetűk kisbetűkre vannak kicserélve.

Megjegyzés

Az ékezetes magyar nagybetűket nem változtatja át.

Példák

- 1) LOWER(“Nemecsek”) értéke „nemecsek”
- 2) LOWER(“BÉTA = 60”) értéke „bÉta = 60”

f) Nagybetű-függvény

Neve: UPPER

Formája: **UPPER(c)**

Típusa: **Karakteres**

Értéke: A **c** karaktersorozat úgy, hogy a kisbetűk nagybetűkre vannak kicserélve.

Megjegyzés

Az ékezetes magyar kisbetűket nem változtatja át.

Példák

- 1) UPPER(“nemecsek”) értéke „NEMECSEK”
- 2) UPPER(“béta = 60”) értéke „BÉTA = 60”

g) Bal oldali szóközöket levágó függvény

Neve: LTRIM

Formája: LTRIM(c)

Típusa: **Karakteres**

Értéke: A c karaktersorozat, de a bal oldalán levő szóközök nélkül.

Példa

LTRIM(“ Bartók Béla”) értéke „Bartók Béla”

h) Jobb oldali szóközöket levágó függvény

Neve: RTRIM vagy egyszerűen csak TRIM

Formája: RTRIM(c) vagy TRIM(c)

Típusa: **Karakteres**

Értéke: A c karaktersorozat, de a jobb oldalán levő szóközök nélkül.

Példák

- 1) RTRIM(“Bartók Béla ”) értéke „Bartók Béla”
- 2) TRIM(“ Kodály Zoltán ”) értéke „ Kodály Zoltán”

i) Hossz-függvény

Neve: LEN

Formája: LEN(c)

Típusa: **Numerikus**

Értéke: A c karaktersorozat hossza.

Példák

- 1) LEN(“Ady”) értéke 3
- 2) LEN(“Ady ”) értéke 4

j) Belső kód-függvény

Neve: ASC

Formája: ASC(c)

Típusa: Numerikus

Értéke: A **c** karaktersorozat első karakterének belső kódja (az ASCII táblázat szerint).

Példák

1) ASC("medve") értéke 109

mert az m betű belső kódja 109.

2) ASC("Σ-tag") értéke 228

mert a Σ karakter belső kódja 228.

k) Karaktert adó függvény

Neve: CHR

Formája: CHR(n)

Típusa: Karakteres

Értéke: Az **n** számnak megfelelő belső kóddal rendelkező karakter.

Példa

CHR(156) értéke „£”

mert a £ az a karakter, amelynek a belső kódja 156.

l) Karakterkereső függvény

Neve: AT

Formája: AT(c1,c2)

Típusa: Numerikus

Értéke: Az a szám, amely megmutatja, hogy a **c1** karaktersorozat hányadik pozíciótól kezdve helyezkedik el a **c2** karaktersorozatban.

Ha a **c1** karaktersorozat nem szerepel a **c2** karaktersorozatban, akkor a függvény értéke 0.

Példák

1) AT(„colett”, „Nicoletta”) értéke 3

mert a „colett” karaktersorozat a „Nicoletta” karaktersorozatban a 3. pozíciótól kezdve helyezkedik el.

2) AT(„colett”, „Nikoletta”) értéke 0

mert a „colett” karaktersorozat nem szerepel a „Nikoletta” karaktersorozatban.

m) Szóközök adó függvény

Neve: SPACE

Formája: SPACE(n)

Típusa: Karakteres

Értéke: n számú szóköz.

Példák

1) SPACE(10) értéke „ ”

2) SPACE(5-LEN(„Ady”)) értéke „ ”

n) Betűvizsgáló függvény

Neve: ISALPHA

Formája: ISALPHA(c)

Típusa: Logikai

Értéke: Akkor 'igaz', ha a c karaktersorozat első karaktere betű, egyébként 'hamis'.

Példák

1) ISALPHA(„Cynthia”) értéke 'igaz'

2) ISALPHA(„1965”) értéke 'hamis'

102. fejezet

A karakterkezelő függvények alkalmazása

Példa

Az aktív adatbázisfájlban a magyar filmek adatai vannak elhelyezve.

Az adatbázisfájl szerkezete, a mezők jelentése és az aktuális rekord tartalma:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Típus</i>	<i>Jelentés</i>
CIM	C	40	-	A film címe
RENDEZO	C	20	-	Rendezője
IRO	C	20	-	Írója
OPERATOR	C	20	-	Operatőre
NEZO	N	8	0	A magyarországi nézők száma
BEVETEL	N	6	0	A külföldi forgalmazás bevétele (£)
LELTAR	C	10	-	Leltári azonosítója

<i>Mezőnév</i>	<i>Tartalom</i>
CIM	Szegénylegények
RENDEZO	Jancsó Miklós
IRO	Hernádi Gyula
OPERATOR	Somló Tamás
NEZO	980000
BEVETEL	2400
LELTAR	BS-102/Σ1

a) A

LEN(CIM)

kifejezés értéke mindig 40, hiszen a CIM mező hossza 40.

Ha tehát ki akarjuk írni, hogy a film címe ténylegesen hány karakterből áll, akkor nem elég a hossz-függvényt használni, hanem a jobb oldali szóközöket levágó függvényt is használni kell:

LEN(TRIM(CIM))

(Az aktuális rekordnál ennek a kifejezésnek az értéke 15.)

b) Ha a rendező nevének csak az első 8 karakterét akarjuk kiírni, akkor a balrész-függvényt használjuk:

LEFT(RENDEZO,8)

(Az aktuális rekordnál ennek a kifejezésnek az értéke „Jancsó M”.)

c) Ha az író nevét nagybetűkkel akarjuk kiírni, akkor a nagybetű-függvényt használjuk:

```
UPPER(IRO)
```

(Az aktuális rekordnál ennek a kifejezésnek az értéke „HERNÁDI GYULA ”.)

d) Ha a külföldi forgalmazás bevétele mellé a £ jelet is oda akarjuk írni, akkor a karaktert adó függvényt használjuk:

```
CHR(156)
```

(Ennek a kifejezésnek az értéke „£”.)

e) Ha csak azoknak a filmeknek az adatait akarjuk kiírni, amelyeknek a leltári azonosítójában a 2. karakter betű, akkor a parancsban feltételt adunk meg, a feltételben pedig a használjuk a középrész-függvényt és a betűvizsgáló függvényt:

```
ISALPHA(SUBSTR(LELTAR,2,1))
```

(Az aktuális rekordnál ennek a feltételnek az értéke ‘igaz’, mert a leltári azonosítóban a 2. karakter betű.)

f) Ha a rendező és az operatőr nevét úgy akarjuk kiírni, hogy a nevük között pontosan három szóköz legyen, akkor a jobb oldali szóközöket levágó függvényt és a szóközöket adó függvényt használjuk:

```
TRIM(RENDEZO)+SPACE(3)+TRIM(OPERATOR)
```

(Az aktuális rekordnál ennek a kifejezésnek az értéke „Jancsó Miklós Somló Tamás”.)

g) Ha csak azoknak a filmeknek az adatait akarjuk kiírni, amelyeknek a leltári azonosítójában a 8. helyen a Σ karakter szerepel, akkor a parancsban feltételt adunk meg, a feltételben pedig használjuk a középrész-függvényt és a belső kód-függvényt:

```
ASC(SUBSRT(LELTAR,8,1)) = 228
```

(Az aktuális rekordnál ennek a feltételnek az értéke ‘igaz’, mert a leltári azonosító 8. karakterének belső kódja 228.)

h) Ha csak azoknak a filmeknek az adatait akarjuk kiírni, amelyeket Jancsó rendezett, akkor a parancsban feltételt adunk meg, a feltételben pedig a karakterkereső függvényt használjuk:

```
AT('Jancsó',RENDEZO) > 0
```

(Az aktuális rekordnál ennek a kifejezésnek az értéke ‘igaz’, mert a RENDEZO mezőben szerepel a „Jancsó” karaktersorozat.)

i) Ha a képernyőre elhatárolás céljából egymás mellé 80 darab *-ot akarunk kiírni, akkor ezt a parancsot adjuk ki:

```
? REPLICATE(" ",80)
```


Feladatok

Az alábbi feladatokat számítógép használata nélkül oldd meg (a megoldásokat csak papírra írd le)!

A feladatok a fenti példához kapcsolódnak.

1. Írnod ki az azoknak a filmeknek a címét és operatőrét, amelyeknek a címe 25 karakternél hosszabb!

2. Írnod ki az összes film címét, rendezőjét és íróját úgy, hogy a címből csak az első 20 karakter legyen kiírva!

3. Írnod ki csupa nagybetűvel azoknak a filmeknek a címét, amelyeket Jancsó Miklós rendezett!

4. Írnod ki azoknak a filmeknek a címét és a külföldi forgalmazásából származó bevételét, amelyeknek a magyarországi nézőszáma meghaladja a 100000-et! (A bevétel mellé írd oda a £ jelet is!)

5. Írnod ki azoknak a filmeknek a címét, amelyeknek a leltári azonosítójában az első karakter nem betű!

6. Írnod ki az összes film rendezőjét és címét úgy, hogy a rendező neve és a film címe között ne legyenek szóközök, csak egy kettőspont.

103. fejezet Dátumkezelő függvények

a) A mai dátumot megadó függvény

Neve: DATE

Formája: DATE()

Típusa: Dátum jellegű

Értéke: A mai dátum.

Példa

Ha kiadjuk a

? DATE()

parancsot, akkor a képernyőn megjelenik a mai dátum.

b) Évszám-függvény

Neve: YEAR

Formája: YEAR(d)

Típusa: Numerikus

Értéke: A d dátumban szereplő évszám.

Példa

Ha az aktuális rekordban a SZUL_DATUM nevű dátum jellegű mező az 1965. február 21-i dátumot tartalmazza, akkor

YEAR(SZUL_DATUM)

értéke 1965.

c) Hónap-függvény

Neve: MONTH

Formája: MONTH(d)

Típusa: Numerikus

Értéke: A d dátumban a hónap sorszáma.

Példa

Ha az aktuális rekordban a SZUL_DATUM nevű dátum jellegű mező az 1965. február 21-i dátumot tartalmazza, akkor

MONTH(SZUL_DATUM)

értéke 2, mert február az év 2. hónapja.

d) Nap-függvények

Két olyan függvény van, amely a nap sorszámát adja meg.

Az egyik a napnak a hónapon belüli sorszámát, a másik pedig a napnak a héten belüli sorszámát adja meg. Ez utóbbi sorszám úgy értendő, hogy a héten belül a vasárnap az 1-es sorszámú nap, a hétfő a 2-es sorszámú nap, stb.

Az egyik függvény neve: DAY

Formája: DAY(d)

Típusa: Numerikus

Értéke: A d dátum hányadik nap a hónapon belül.

A másik függvény neve: DOW

Formája: **DOW(d)**

Típusa: **Numerikus**

Értéke: A **d** dátum hányadik nap a héten belül.

Példa

Ha az aktuális rekordban a SZUL_DATUM nevű dátum jellegű mező az 1965. február 21-i dátumot tartalmazza, akkor

DAY(SZUL_DATUM)

értéke 21

DOW(SZUL_DATUM)

értéke pedig 1, mert 1965. február 21. vasárnapra esett.

104. fejezet

A dátumkezelő függvények alkalmazása

Példa

Az aktív adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjss.</i>	<i>Jelentés</i>
NEV	C	20	-	Az ismerős neve
SZUL_DATUM	D	8	-	Születési dátuma
IRSZ	N	4	0	Irányítószáma
UTCA	C	12	-	Melyik utcában lakik?
HAZSZAM	C	6	-	Hány szám alatt?

a) Ha ki akarjuk íratni az 1965-ben született ismerősök nevét, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV FOR YEAR(SZUL_DATUM) = 1965
```

b) Ha ki akarjuk íratni a 30 évesnél idősebb ismerősök összes adatát, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL FOR DATE()-SZUL_DATUM > 30*365
```


c) Ha ki akarjuk íratni a február 21-én született ismerősök nevét, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV FOR MONTH(SZUL_DATUM) = 2 .AND. DAY(SZUL_DATUM) = 21
```

d) Ha tudni akarjuk, hogy az ismerőseink közül ki született keddi napon, akkor ezt a parancsot adjuk ki:

```
DISPLAY ALL NEV FOR DOW(SZUL_DATUM) = 3
```

Feladat

A gyakorló lemezed KM katalógusában található TANULO.DBF adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_HELY	C	20	-	Születési helye
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
ATLAG	N	4	2	Tanulmányi átlaga
KOLLEGIUM	L	1	-	Kollégiumban lakik-e?

a) Nyisd meg az adatbázisfájlt!

b) Írasd ki azoknak a tanulóknak a nevét, akik a Halak csillagképben születtek! (A Halak csillagkép február 20-tól március 20-ig tart.)

c) Számítsd ki, hogy a 16 évesnél idősebb tanulóknak összesen hány testvérük van!

d) Számítsd ki a decemberi születésű tanulók tanulmányi átlagának átlagát!

e) Számláld meg, hogy hány olyan tanuló van, aki csütörtöki napon született!

f) Zárd le az adatbázisfájlt!

105. fejezet

Választó függvény

Neve: IIF

Formája: **IIF(l,t1,t2)**

A második és a harmadik argumentum típusának meg kell egyeznie!

Típusa: a t1 és t2 típusával megegyező.

Értéke: Ha az 1 értéke 'igaz', akkor t1 értéke, egyébként t2 értéke.

Példa

Az aktív adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tíjz.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
MUNKABER	N	6	0	Havi munkabére
NYELV	L	1	-	Tett-e sikeres nyelvvizsgát?

Azok a dolgozók, akik sikeres nyelvvizsgát tettek, havi 2500 Ft munkabér-kiegészítést kapnak.

Ezt a kiegészítést az

IIF(NYELV,2500,0)

kifejezés tükrözi. Ennek értéke ugyanis a sikeres nyelvvizsgát tett dolgozóknál 2500, a többiekénél pedig 0.

A dolgozók havi járandósága tehát ezzel a paranccsal íratható ki:

```
DISPLAY ALL NEV, MUNKABER+IIF(NYELV, 2500, 0)
```

106. fejezet

Típusváltó függvények

A típusváltó függvények arra szolgálnak, hogy egy kifejezés értékének a típusát megváltoztassuk.

A változtatásra akkor van szükség, ha a kifejezés értékével olyan műveletet akarunk végezni, ami az eredeti típus miatt nem lehetséges.

Példák

1. Ha egy karakteres mező mindig csak számjegyeket tartalmaz, akkor adódhat úgy, hogy a mező értékével számtani műveleteket akarunk végezni. Ennek csak az az akadálya, hogy a mező értéke nem szám, hanem karaktorsorozat. Át kell tehát alakítani számmá, s azután már lehet vele számtani műveleteket végezni.

2. Ha egy karakteres mező értékéhez hozzá akarjuk fűzni egy numerikus mező értékét, akkor ennek az akadálya az, hogy a második mező értéke nem karaktorsorozat, hanem szám. Át kell tehát alakítani karaktorsorozattá, s azután már el lehet végezni az összefűzést.

Az alábbi táblázatban összefoglaljuk a típusváltató függvények nevét, s azt is odairjuk mindegyik mellé, hogy milyen típusú adatot milyen típusú adatra változtat.

Neve	Mit...	Mire...
STR	Számot	karaktorsorozatra.
VAL	Karaktorsorozatot	számra.
CTOD	Karaktorsorozatot	dátumra.
DTOC	Dátumot	karaktorsorozatra.

Nézzük mindezt részletesen!

a) A STR függvény

Formája: **STR(n1[,n2][,n3])**

A függvény az **n1** számot karaktorsorozattá alakítja.

A létrejövő karaktorsorozat teljes hosszát az **n2** határozza meg, a benne szereplő tizedesjegyek számát pedig az **n3**. Szükség esetén kerekítés történik.

Példa

STR(16.78,8,4) értéke „ 16.7800”

Megjegyzések

1. Ha az **n2** hiányzik, akkor az olyan, mintha 10 lenne.
2. Ha az **n3** hiányzik, akkor az olyan, mintha 0 lenne.
3. Ha az **n2** hiányzik, de az **n3** meg van adva, akkor az **n3** elé két vesszőt kell tenni, így jelezve az **n2** hiányát.

Példák

- 1) STR(16.78,5) értéke „ 17”
- 2) STR(16.78,,4) értéke „ 16.7800”
- 3) STR(16.78) értéke „ 17”

b) A VAL függvény

Formája: **VAL(c)**

A függvény a **c** karaktorsorozatot számmá alakítja.

Példa

VAL("15.78") értéke 15.78

Megjegyzés

Csak akkor érdemes használni ezt a függvényt, ha a **c** karaktersorozatban csupán az alábbi karakterek szerepelnek:

- számjegyek
- tizedespont
- előjel.

c) A CTOD függvény

Formája: **CTOD(c)**

A függvény a **c** karaktersorozatnak megfelelő dátumot adja.

Példa

CTOD('1965.02.21') értéke az 1965. február 21-i dátum.

Megjegyzés

Most már világos, miért nem probléma az, hogy a dBASE nem ismer dátum jellegű konstanst.

Ugyanis a parancsban egy dátumot úgy adhatunk meg konkrétan, hogy aposztrófok közé zárjuk, majd a CTOD függvény argumentumaként szerepeltetjük.

d) A DTOC függvény

Formája: **DTOC(d)**

A függvény a **d** dátumot karaktersorozatként szolgáltatja.

Példa

Ha az aktuális rekordban a SZUL_DATUM nevű dátum jellegű mező az 1965. február 21-i dátumot tartalmazza, akkor

DTOC(SZUL_DATUM) értéke „1965.02.21”

107. fejezet

A típusváltó függvények alkalmazása

Példák

Az aktív adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tjst.</i>	<i>Jelentés</i>
NEV	C	20	-	A dolgozó neve
SZ_SZ	N	11	0	Személyi száma
MUNKA	D	8	-	Munkaviszonyának kezdete
ALAP	C	6	-	Havi alapbére
POT	C	6	-	Havi bérpótléka
LEV	C	6	-	Havi levonása

1. Ki akarjuk írni külön a férfi és külön a női dolgozók havi járandóságának átlagát. Ezzel kapcsolatban két probléma merül fel:

a) Az adatbázisfájlban a személyi számot tartalmazó SZ_SZ nevű mező numerikus. A dolgozó nemét a személyi szám első karakteréből tudhatnánk meg a LEFT függvény segítségével, de a LEFT függvényt csak karaktersorozatra alkalmazhatjuk. Ezért a személyi számot előbb át kell alakítani karaktersorozattá:

`STR(SZ_SZ,11)`

Az így kapott karaktersorozatra már alkalmazható a LEFT függvény:

`LEFT(STR(SZ_SZ,11),1)`

(Ennek a kifejezésnek az értéke éppen a személyi szám első karaktere.)

Tehát a

`LEFT(STR(SZ_SZ,11),1) = '1'`

feltétel csak a férfi dolgozónál vesz fel 'igaz' értéket.

b) A dolgozó havi járandósága az ALAP, a POT és a LEV mezők értékéből számolható ki, ezek a mezők azonban az adatbázisfájlban karakteresek. Számolást csak számokkal lehet végezni, ezért az ALAP, a POT és a LEV mezők tartalmát a VAL függvénnyel számmá kell átalakítani. A számolás ezzel a kifejezéssel végezhető el:

`VAL(ALAP)+VAL(POT)-VAL(LEV)`

Ezek után elmondhatjuk, hogy a férfi dolgozók átlagos havi járandósága így íratható ki:

```
AVERAGE VAL(ALAP)+VAL(POT)-VAL(LEV) FOR LEFT(STR(SZ_SZ,11),1) = '1'
```

a női dolgozóké pedig így:

```
AVERAGE VAL(ALAP)+VAL(POT)-VAL(LEV) FOR LEFT(STR(SZ_SZ,11),1) = '2'
```

2. Ki akarjuk írni azoknak a dolgozóknak a nevét, akik 1974. szeptember 1. előtt kezdték meg a munkaviszonyukat.

A kiírás kétféleképpen is elvégezhető:

a) A MUNKA nevű mező tartalmát alakítjuk át karaktersorozattá a DTOC függvénnyel:

```
DISPLAY ALL NEV FOR DTOC(MUNKA) < '1974.09.01'
```

b) Az 1974. szeptember 1-i dátumot a CTOD függvény segítségével adjuk meg:

```
DISPLAY ALL NEV FOR MUNKA < CTOD('1974.09.01')
```

108. fejezet

DOS-parancs végrehajtása

A dBASE használata során tetszőleges DOS-parancsot végrehajthatunk anélkül, hogy kilépnénk a dBASE-ből.

Ezt a RUN parancs teszi lehetővé.

A parancs formája:

RUN DOS-parancs

A parancs hatására végrehajtódik a RUN szót követő DOS-parancs, majd újra megjelenik a dBASE készenléti jele.

Példa

Ha az adatbázis-kezelés során létre akarnánk hozni a C: meghajtó DBASE katalógusában az ADATX nevű alkatalógust, akkor ezt a parancsot adnánk ki:

```
RUN MD C:\DBASE\ADATX
```

Megjegyzés

A RUN szó helyett ! is megadható.

109. fejezet

A dBASE működésének szabályozása

a) A képernyő színeinek beállítása

Színes monitornál a képernyőre történő írás színeit a SET COLOR paranccsal állítjuk be.

A parancs formája:

SET COLOR TO [normál_írás színe] [,kiemelt_írás színe] [,keretszín]

A színeket ezekkel a kódokkal kell megadni:

A szín neve	Kódja
Piros	R
Fehér	W
Zöld	G
Kék	B
Fekete	N
Cián	BG
Biboryörös	RB
Barna	GR

A szín kódja után írt + jel magas fényerőt, a * pedig villogást jelent.

A normál írás színét két, egymástól / jellel elválasztott színkóddal lehet megadni. Az első kód a karakter színét, a második pedig a karakter háttérének a színét határozza meg.

A kiemelt írás színét is két, egymástól / jellel elválasztott színkóddal lehet megadni. Az első kód a karakter színét, a második pedig a karakter háttérének a színét határozza meg.

A keretszín (a képernyő keretének színe) egyetlen színkóddal adható meg.

Példa

Ha színes monitorunk van, és azt akarjuk, hogy

— a normál írás fehér karakterekkel történjen, a háttére pedig magas fényerejű kék legyen

— a kiemelt írás barna karakterekkel történjen, a háttere pedig magas fényerejű piros legyen

akkor ezt a parancsot adjuk ki:

SET COLOR TO W/B+,GR/R+

Megjegyzések

1. A sárga szín magas fényerejű barna színként hozható létre.

2. A szürke szín magas fényerejű fekete színként hozható létre.

3. Ha nem alkalmazzuk a SET COLOR parancsot, akkor a színes monitor is fekete-fehéren működik. Ez azt jelenti, hogy a normál írás fekete háttérrel és fehér karakterekkel történik, a kiemelt írás pedig fordítva.

Feladat

Ha színes monitorod van, akkor állítsd be a képernyő színeit úgy, hogy

— a normál írás magas fényerejű piros karakterekkel és szürke háttérrel történjen

— a kiemelt írás barna karakterekkel és magas fényerejű kék háttérrel történjen

— a keret színe cián legyen!

(Ha a kapott színek nem tetszenek, akkor állítsd be a képernyő színeit ízlésed szerint!)

b) A dátumok formájának beállítása

Az eddigiek során a dátumokat a Magyarországon megszokott formában kezeltük.

A dátumok formája a SET DATE paranccsal állítható be.

A parancs formája:

SET DATE AMERICAN|ANSI|BRITISH|ITALIAN|FRENCH|GERMAN

Ez azt jelenti, hogy a SET DATE szavak után az AMERICAN vagy az ANSI vagy a BRITISH vagy az ITALIAN vagy a FRENCH vagy a GERMAN szót kell megadni. Mindegyik szó egy dátumformát határoz meg.

A dátumformák két dologban különböznek egymástól:

— az év, hónap, nap sorrendjében

— az évet a hónaptól, valamint a hónapot a naptól elválasztó határolójelben.

Ezeket a különbségeket az alábbi táblázat szemlélteti.

A táblázat utal arra is, hogy melyik dátumforma melyik országban használatos.

Példaképpen megadjuk, hogy az 1965. február 21-i dátum hogyan néz ki a különböző formák szerint.

Dátumforma	Sorrend	Határolójel	Ország	Példa
AMERICAN	Hónap, nap, év	/	U.S.A.	02/21/1965
ANSI	Év, hónap, nap	.	Magyarország	1965.02.21
BRITISH	Nap, hónap, év	/	Anglia	21/02/1965
ITALIAN	Nap, hónap, év	-	Olaszország	21-02-1965
FRENCH	Nap, hónap, év	/	Franciaország	21/02/1965
GERMAN	Nap, hónap, év	.	Németország	21.02.1965

Ha nem használjuk a SET DATE parancsot, akkor az AMERICAN forma érvényesül.

Példák

1. Ha a német dátumformát akarjuk alkalmazni, akkor ezt a parancsot adjuk ki:

```
SET DATE GERMAN
```

2. Ha a magyar dátumformát akarjuk alkalmazni, akkor ezt a parancsot adjuk ki:

```
SET DATE ANSI
```

Feladat

A gyakorlólemez KM katalógusában levő TANULO.DBF adatbázisfájl felhasználásával írasd ki a tanulók nevét és születési dátumát! Előbb olasz, majd magyar dátumformát alkalmazz!

c) A kiírt tizedesjegyek számának beállítása

Arra, hogy a numerikus kifejezés értéke hány tizedesjeggyel íródik ki, az alábbi szabályok vonatkoznak.

1. szabály

Az összeadás és a kivonás eredményének kiírásakor a tizedesjegyek számát a több tizedesjeggyel rendelkező operandus határozza meg.

Példa

Az

5.213+13.0000101

műveletben az egyik operandus 3, a másik pedig 7 tizedesjegyet tartalmaz, ezért a művelet eredménye 7 tizedesjeggyel íródik ki:

18.2130101

2. szabály

A szorzás eredménye annyi tizedesjeggyel íródik ki, mint amennyi az operandusok tizedesjegyeinek összege.

Példa

Az

5.02*6.134

műveletben az egyik operandus 2, a másik pedig 3 tizedesjegyet tartalmaz, ezért a művelet eredménye 5 tizedesjeggyel íródik ki:

30.79268

3. szabály

Azt, hogy az osztás, a hatványozás és a gyökvonás eredménye hány tizedesjeggyel íródjon ki, a SET DECIMALS paranccsal kell beállítani.

A parancs formája:

SET DECIMALS TO numerikus_konstans

A parancs hatására a továbbiakban az osztás, a hatványozás és a gyökvonás eredménye annyi tizedesjeggyel íródik ki, mint amennyi a numerikus konstans értéke.

Ha nem használjuk a SET DECIMALS parancsot, akkor az osztás, a hatványozás és a gyökvonás eredménye 2 tizedesjeggyel íródik ki.

Példa

Ha kiadjuk a

SET DECIMALS TO 4

parancsot, akkor ezt követően az

5/3

művelet eredménye 4 tizedesjeggyel íródik ki:

1.6667

4. szabály

Egy numerikus kifejezés értékének kiírásakor a tizedesjegyek száma a fenti szabályok (többszöri) alkalmazásával alakul ki.

Példa

Ha a

```
SET DECIMALS TO 4
```

parancs van érvényben, akkor a

```
(4.32/3.1)*5.11035
```

kifejezés értéke 9 tizedesjeggyel íródik ki. Ugyanis az osztás eredménye 4 tizedesjeggyel íródna ki, de a szorzásnál a tizedesjegyek száma (a 4 és az 5) összeadódik, így alakul ki a 9 tizedesjegy.

d) A funkcióbillentyűk beállítása

A funkcióbillentyűk jelentését a SET FUNCTION paranccsal állítjuk be.

A parancs formája:

```
SET FUNCTION numerikus_konstans TO karakteres_konstans
```

A numerikus konstans határozza meg, hogy hányadik funkcióbillentyű jelentését akarjuk beállítani.

A karakteres konstans határozza meg a funkcióbillentyű jelentését.

A karakteres konstansban szereplő pontosvessző az Enter billentyűt jelenti.

Ha a parancs kiadását követően lenyomjuk az illető funkcióbillentyűt, akkor az úgy minősül, mintha a karakteres konstansban szereplő karaktersorozatot gépeltük volna be.

Példák

1. Ha kiadjuk a

```
SET FUNCTION 4 TO "INDEX ON"
```

parancsot, akkor ezt követően az F4 funkcióbillentyű leütésekor a parancssorban ez jelenik meg:

```
INDEX ON
```

s nekünk már csak a parancs további részét kell begépelnünk, majd a végrehajtás érdekében le kell ütnünk az Enter-t.

2. Ha kiadjuk a

```
SET FUNCTION 3 TO 'DISPLAY STRUCTURE;'
```


parancsot, akkor ezt követően az F3 funkcióbillentyű leütésekor a parancssorban ez jelenik meg:

```
DISPLAY STRUCTURE
```

majd az Enter-t jelentő pontosvessző miatt a parancs azonnal végre is hajtódik.

3. Ha kiadjuk a

```
SET FUNCTION 5 TO "USE A:\KMITANULO;DISPAY STRUCTURE;DISPLAY ALL;"
```

parancsot, akkor ezt követően az F5 funkcióbillentyű leütésekor azonnal végrehajtódnak az alábbi parancsok:

```
USE A:\KMITANULO
```

```
DISPLAY STRUCTURE
```

```
DISPLAY ALL
```

Megjegyzések

1. Ha egy funkcióbillentyű jelentését nem állítjuk be a SET FUNCTION paranccsal, akkor az úgynevezett alapértelmezés érvényes.

Az alábbi táblázat feltünteti néhány funkcióbillentyű alapértelmezését:

A funkcióbillentyű	Alapértelmezés
F1	HELP;
F3	LIST;
F4	DIR;
F5	DISPLAY STRUCTURE;
F8	DISPLAY;
F9	APPEND;
F10	EDIT;

Az F1 funkcióbillentyű jelentését mi nem állíthatjuk be, az F1 esetben tehát mindig az alapértelmezés van érvényben.

2. A HELP parancs arra szolgál, hogy a dBASE használatára vonatkozó angol nyelvű segítséget kérjünk. Ennek feltétele, hogy a C: meghajtó DBASE katalógusában rendelkezésünkre álljon a HELP.DBS fájl.

Feladat

Változtasd meg az F2 és az F3 funkcióbillentyűk jelentését úgy, hogy

a) az F2 leütésekor a 4-es munkaterület váljon aktuálissá

b) az F3 leütésekor a parancssorban az

USE A:\KM\

felirat jelenjen meg.

Próbáld ki, hogy jól oldottad-e meg a feladatot!

e) A fájlkeresés szabályozása

Ha az adatbázisfájl vagy az indexfájl megnyitásakor nem adjuk meg a teljes azonosítóját, hanem csak a nevét, akkor a dBASE az aktuális katalógusban, vagyis a C: meghajtó DBASE katalógusában keresi a fájlt.

Ha a keresést egyéb katalógusokra is ki akarjuk terjeszteni, akkor alkalmazni kell a SET PATH parancsot.

A parancs formája:

SET PATH TO [katalógushivatkozások_listája]

A katalógushivatkozásokat egymástól pontosvesszővel kell elválasztani.

Az aktuális katalóguson kívül ezekben a katalógusokban is keresni fogja a dBASE a megnyitandó fájlt.

Ha a parancsban nem adunk meg egyetlen katalógushivatkozást sem, akkor ezután a keresés csak az aktuális katalógusban, vagyis a C: meghajtó DBASE katalógusban történik.

Példák

1. Ha csak azokat a fájlokat akarjuk megnyitni, amelyek a gyakorló lemezünk KM katalógusában vannak, akkor ezt a parancsot érdemes kiadni:

```
SET PATH TO A:\KM
```

Ettől kezdve a megnyitáskor a teljes azonosító helyett elég a fájlok nevét megadni.

Tehát az

```
USE A:\KMTANULO.DBF
```

parancs így egyszerűsödik:

```
USE TANULO
```

A

```
SET INDEX TO A:\KM\NEVSOR.NDX
```

parancs pedig így egyszerűsödik:

```
SET INDEX TO NEVSOR
```


2. Ha nem csak a gyakorló lemezünk KM katalógusában, hanem még a C: meghajtó DBASE katalógusának ADAT alkatalógusában levő fájlokat is meg akarnánk nyitni, akkor ezt a parancsot lenne érdemes kiadni:

```
SET PATH TO A:\KM;C:\DBASE\ADAT
```

f) Szabályozás kapcsolókkal

A dBASE működését kapcsolókkal is szabályozhatjuk.

Minden kapcsolóval más-más dolgot állítunk be. A kapcsolók névvel rendelkeznek.

A legfontosabb kapcsolók neve:

BELL
CENTURY
CONFIRM
EXACT
FIXED
HEADINGS
HELP
HISTORY
MENUS
SAFETY
STATUS

Minden kapcsolónak kétféle állapota lehet: bekapcsolt és kikapcsolt.

A kapcsolók állapotát a SET paranccsal állítjuk be.

A parancs formája:

```
SET kapcsolónév ON|OFF
```

A SET szót követően kell megadni annak a kapcsolónak a nevét, amelynek az állapotát be akarjuk állítani.

A kapcsolónév után az ON vagy az OFF szót kell megadni.

Az ON szóval bekapcsolt, az OFF szóval pedig kikapcsolt állapotba helyezzük a kapcsolót.

Példák

1. A

```
SET BELL ON
```

parancs bekapcsolja a BELL kapcsolót.

2. A

```
SET CENTURY OFF
```

parancs kikapcsolja a CENTURY kapcsolót.

Vegyük sorra a felsorolt kapcsolókat.

1) A BELL kapcsoló

A BELL kapcsolóval a hangjelzést állítjuk be.

Ha a BELL be van kapcsolva, akkor hangjelzést kapunk olyankor, amikor

— meg nem engedett karaktert gépelünk be

— egy mező tartalmának begépelésekor elérjük a mező végét.

Ha a BELL ki van kapcsolva, akkor hangjelzést soha nem kapunk.

2) A CENTURY kapcsoló

A CENTURY kapcsolóval azt állítjuk be, hogy begépeléskor és kiírásakor szerepeljen-e a dátumban az évszázadra utaló két számjegy.

Ha a CENTURY be van kapcsolva, akkor begépeléskor is meg kell adnunk az évszázadra utaló két számjegyet, és a kiírásnál is megjelenik az évszázadra utaló két számjegy.

Ha a CENTURY ki van kapcsolva, akkor

— csak huszadik századi dátum gépelhető be, de az évszázadra utaló két számjegyet (a 19-et) nem kell begépelni

— kiírásakor az évszázadra utaló két számjegy nem jelenik meg.

Példák

A dátumformák közül az ANSI van érvényben.

1. Ha kiadjuk a

SET CENTURY ON

parancsot, akkor ettől kezdve

a) az 1965. február 21-i dátumot így kell begépelni: 1965.02.21

b) a 2004. augusztus 11-i dátumot így kell begépelni: 2004.08.11

c) a fenti dátumok így fognak kiíródni: 1965.02.21 és 2004.08.11

2. Ha kiadjuk a

SET CENTURY OFF

parancsot, akkor ettől kezdve

a) csak huszadik századi dátumot lehet begépelni

b) az 1965. február 21-i dátumot így kell begépelni: 65.02.21

c) az 1965. február 21-i dátum így fog kiíródni: 65.02.21

d) az előzőleg (még a CENTURY bekapcsolt állapotakor) begépelte 2004. augusztus 11-i dátum így fog kiíródni: 04.08.11

3) A CONFIRM kapcsoló

A CONFIRM kapcsolónak akkor van jelentősége, amikor adatokat gépelünk be a CREATE, az APPEND, az INSERT, az EDIT, a CHANGE vagy a BROWSE parancssal.

Ha a CONFIRM ki van kapcsolva, akkor a mező végének elérésekor a kurzor automatikusan áttér a következő mezőre.

Ha a CONFIRM be van kapcsolva, akkor a következő mezőre csak az Enter leütése árán térhetünk át.

4) Az EXACT kapcsoló

Az EXACT kapcsolónak a karaktersorozatok összehasonlításakor van jelentősége.

Ha az EXACT be van kapcsolva, akkor az összehasonlítás „szigorúan” történik: két karaktersorozat csak akkor számít egyenlőnek, ha a végén levő szóközöktől eltekintve a hosszuk is megegyezik, és minden egyes karakterük megegyezik.

Ha az EXACT ki van kapcsolva, akkor legfeljebb annyi karakter hasonlítódik össze, mint amennyi a jobb oldali operandus hossza a végén levő szóközzel együtt. Ha addig nincs eltérés, akkor a két operandus egyenlőnek minősül.

Példák

1. Ha kiadjuk a

```
SET EXACT OFF
```

parancsot, akkor ezután a

```
"Budapest" = 'Buda'
```

összehasonlítás eredménye **'igaz'**, mert a jobb oldali karaktersorozat négy karakterből áll, márpedig a két karaktersorozat az első négy karakterében megegyezik.

2. Ha kiadjuk a

```
SET EXACT ON
```

parancsot, akkor ezután a

```
'Budapest' = "Buda"
```

összehasonlítás eredménye **'hamis'**.

3. A

```
"Budapest" = "Budai"
```

összehasonlítás eredménye az EXACT kapcsoló állásától függetlenül mindig **'hamis'**.

5) A FIXED kapcsoló

A FIXED kapcsolóval azt állítjuk be, hogy a numerikus kifejezések értékének kiírásakor a tizedesjegyek száma a szabályok szerint alakuljon-e ki?

Ha a FIXED ki van kapcsolva, akkor a tizedesjegyek száma a szabályok szerint alakul ki.

Ha a FIXED be van kapcsolva, akkor a kiírásakor nem érvényesülnek a szabályok, hanem a kiírás minden esetben annyi tizedesjeggyel történik, mint amennyi a SET DECIMALS paranccsal (az osztásra, a hatványozásra és a gyökvonásra vonatkozóan) be van állítva.

Példák

1. Ha kiadjuk a

SET FIXED OFF

parancsot, akkor ezután a

7.16+8.104+0.00001

kifejezés eredménye a szabályok szerint, tehát 5 tizedesjeggyel fog kiíródni:

15.26401

2. Ha kiadjuk a

SET DECIMALS TO 7

és a

SET FIXED ON

parancsot, akkor ezután a

7.16+8.104+0.00001

művelet eredménye nem a szabályok szerint, hanem 7 tizedesjeggyel fog kiíródni:

15.2640100

6) A HEADINGS kapcsoló

A HEADINGS kapcsolóval azt állítjuk be, hogy a LIST, a DISPLAY, a SUM és az AVERAGE parancsok végrehajtásakor jelenjen-e meg fejléc a kiírt adatok fölött.

A fejléc arra szolgál, hogy lássuk azt a kifejezést, aminek az értékeit (esetleg az értékeinek összegét vagy átlagát) kiírtattuk.

Ha a HEADINGS be van kapcsolva, akkor a fejléc megjelenik, egyébként nem.

7) A HELP kapcsoló

A HELP kapcsolóval azt állítjuk be, hogy hibás parancs begépelésekor ajánljon-e segítséget a dBASE.

Ha a HELP be van kapcsolva, és a DBASE katalógusban rendelkezésünkre áll a HELP.DBS fájl, akkor minden általunk elkövetett hiba után a képernyőn ez a kérdés jelenik meg:

Do you want some help? (Y/N)

Magyarul:

Kérsz segítséget? (Y/N)

Amennyiben a kérdésre az Y billentyű leütésével válaszolunk, akkor angol nyelvű segítséget kapunk a képernyőn.

Ha a HELP ki van kapcsolva, akkor nem kérhetünk segítséget.

8) A HISTORY kapcsoló

A HISTORY kapcsolóval azt állítjuk be, hogy az előzőleg kiadott parancsok között mozoghassunk-e a ↑ vagy a ↓ nyílbillentyűvel.

Akkor mozoghatunk, ha a HISTORY be van kapcsolva.

9) A MENUS kapcsoló

A MENUS kapcsolónak akkor van jelentősége, amikor adatokat gépelünk be a CREATE, az APPEND, az INSERT, az EDIT, a CHANGE vagy a BROWSE parancsokkal.

Ha a MENUS be van kapcsolva, akkor a begépelés során a képernyő tetején a billentyűk használatára vonatkozó angol nyelvű segédlet látható.

Ha a MENUS ki van kapcsolva, akkor ez a segédlet nem látható.

Gépelés közben a segédlet az F1 funkcióbillentyűvel eltüntethető és újra megjeleníthető.

10) A SAFETY kapcsoló

A SAFETY kapcsolóval azt állítjuk be, hogy kapjunk-e figyelmeztetést akkor, ha olyan fájlt (akár adatbázisfájlt, akár indexfájlt) akarunk létrehozni, amelyen már létezik.

Olyankor, amikor a SAFETY be van kapcsolva, ezt a figyelmeztetést kapjuk:

... file already exists, overwrite it? (Y/N)

Magyarul:

A ... fájl már létezik. Megsemmisíthető? (Y/N)

Ha a kérdésre Y-nal válaszolunk, akkor az előző fájl megsemmisül; ha pedig N-nel, akkor nem.

Olyankor, amikor a SAFETY ki van kapcsolva, az előző fájl figyelmeztetés nélkül megsemmisül.

11) A STATUS kapcsoló

A STATUS kapcsolóval azt állítjuk be, hogy a státuszsor látszódjon-e?

Ha a STATUS be van kapcsolva, akkor látszik a státuszsor, egyébként nem.

Ha valamely kapcsoló állapotát nem állítjuk be, akkor annak alapértelmezése van érvényben. Az alapértelmezések az alábbi táblázatban láthatók:

A kapcsoló neve	Alapértelmezése
BELL	ON
CENTURY	OFF
CONFIRM	OFF
EXACT	OFF
FIXED	OFF
HEADINGS	ON
HELP	ON
HISTORY	ON
MENUS	ON
SAFETY	ON
STATUS	OFF

Példa

Ha eddig még sem a
SET FIXED ON
sem a
SET FIXED OFF
parancsot nem adtuk ki, akkor a FIXED kapcsoló kikapcsolt állapotban van.

g) A tárolt parancsok számának beállítása

Tudjuk, hogy ha a HISTORY be van kapcsolva, akkor az előzőleg kiadott parancsok tárolódnak, ezáltal lehetővé válik, hogy a ↑ vagy a ↓ nyílbillentyűvel mozogjunk a parancsok között.

A tárolt parancsok számát a SET HISTORY paranccsal határozzuk meg.

A parancs formája:

SET HISTORY TO numerikus_konstans

Az előző parancsok közül annyi tárolódik, mint amennyi a numerikus konstans értéke.

Ha nem használjuk a SET HISTORY parancsot, akkor a legutolsó 20 parancs tárolódik.

Példa

Ha kiadjuk a

```
SET HISTORY TO 200
```

parancsot, akkor a legutolsó 200 parancs tárolódik.

110. fejezet A CONFIG.DB fájl szerepe

A dBASE működésének szabályozását a C: meghajtó DBASE katalógusában elhelyezett CONFIG.DB azonosítójú fájl segítségével egyszerűbbé tehetjük. Ezt a fájlt még a dBASE indítása előtt kell létrehozni.

Létrehozható a COPY CON paranccsal vagy egy szövegszerkesztő programmal.

A CONFIG.DB fájl minden sorával más-más dolgot lehet beállítani.

Minden dolog egy kulcsszó és egy érték megadásával állítható be. A kulcsszó és az érték közé = jelet kell tenni.

Tehát a CONFIG.DB fájl egy sorának a formája:

kulcsszó = érték

Az alábbi táblázatban megadjuk, hogy milyen kulcsszavakkal mi állítható be a CONFIG.DB fájlön keresztül.

Kulcsszó	Mi állítható be vele?
BELL	Legyen-e hangjelzés?
CENTURY	A dátumban a begépeléskor és a kiíráskor szerepeljen-e az évszázad?
COLOR	Milyen színű legyen az írás és a keret a képernyőn?
CONFIRM	Kelljen-e Enter-t ütni a mező végének elérésekor?
DECIMALS	Hány tizedesjeggyel íródjon ki az osztás, a hatványozás és a gyökvonás eredménye?

Kulcsszó	Mi állítható be vele?
EXACT	A karaktersorozatok összehasonlítása „szigorú” legyen-e?
HEADINGS	Legyen-e a kiírásakor fejléc?
HELP	Ajánljon-e segítséget hibás parancs kiadásakor?
HISTORY	Hány parancsot akarunk tárolni?
MENUS	Kérünk-e a billentyűhasználathoz segítséget?
PATH	Mely katalógusokban történjen a fájlok keresése?
SAFETY	Létező fájlazonosító esetében legyen-e figyelmeztetés?
STATUS	Látszódjon-e a státuszsor?

A következő táblázat arról tájékoztat, hogy a CONFIG.DB fájlban a kulcsszavak értékét milyen formában kell megadni:

Kulcsszó	Az érték formája
BELL	ON OFF
CENTURY	ON OFF
COLOR	[normál írás színe] [,kiemelt írás színe] [,keretszín]
CONFIRM	ON OFF
DECIMALS	numerikus_konstans
EXACT	ON OFF
HEADINGS	ON OFF
HELP	ON OFF
HISTORY	numerikus_konstans
MENUS	ON OFF
PATH	[katalógushivatkozások_listája]
SAFETY	ON OFF
STATUS	ON OFF

Ha valamelyik kulcsszó nem szerepel a CONFIG.DB fájlban, akkor az olyan, mint- ha az alábbi táblázat szerinti értékkel szerepelne benne:

Kulcsszó	Érték
BELL	ON
CENTURY	OFF
COLOR	W/N, N/W, N
CONFIRM	OFF
DECIMALS	2
EXACT	OFF
HEADINGS	ON
HELP	ON
HISTORY	20
MENUS	ON
PATH	
SAFETY	ON
STATUS	OFF

Feladat

Mondd el, hogy miként szabályozná a CONFIG.DB fájl a dBASE működését, ha ez lenne a tartalma:

a)

BELL = OFF
CENTURY = ON
COLOR = W/B, N/B+, G
CONFIRM = OFF
DECIMALS = 5
EXACT = ON
HEADINGS = OFF
HELP = OFF
HISTORY = 50
MENUS = OFF
PATH = A:\KM
SAFETY = OFF
STATUS = ON

b)

BELL = OFF
COLOR = W/B, N/B+, G
EXACT = ON
HELP = OFF
MENUS = OFF
SAFETY = OFF

A CONFIG.DB fájlban szerepelhet a COMMAND kulcsszó is. Ennek értékeként egy dBASE-parancsot adhatunk meg. Ez a parancs a dBASE minden indításakor automatikusan végrehajtódik.

A COMMAND kulcsszó a CONFIG.DB fájlban csak egyszer szerepelhet.

Példa

Ha azt akarjuk, hogy a dBASE a dátumokat a Magyarországon szokásos formában kezelje, akkor a CONFIG.DB fájlban ezt a sort kell szerepeltetni:

```
COMMAND = SET DATE ANSI
```

Megjegyzés

A dátumformának a CONFIG.DB fájlban keresztül való beállításakor azért kellett a COMMAND kulcsszóhoz folyamodni, mert a DATE nem szerepelhet kulcsszóként a CONFIG.DB fájlban.

A FUNCTION és a FIXED sem szerepelhet kulcsszóként a CONFIG.DB fájlban.

Feladatok

1. Írasd ki a C: meghajtó DBASE katalógusában levő CONFIG.DB fájl tartalmát, és mondd el, hogy ez miként szabályozta az eddigiekben a dBASE működését!

2. Módosítsd a CONFIG.DB fájl tartalmát igényeid szerint!

Összefoglaló feladatsorok

A lemezmellékleten található adatbázisfájlok azt a célt szolgálják, hogy az olvasó ellenőrizze tudását.

Feladatok

1. Töröld le a képernyőt!
2. Nézd meg, hogy milyen katalógusok vannak a lemezmelléklet főkatalógusában!
3. Vedd számba az egyes katalógusokban található adatbázisfájlokat!

Tanulmányilvántartás

A lemezmelléklet ISKOLA katalógusában található az ALTALAN.DBF adatbázisfájl, amely egy általános iskola felső tagozatos tanulóinak adatait tartalmazza. Az adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
NEV	C	25	-	A tanuló neve
SZUL_DATUM	D	8	-	Születési dátuma
TESTVER	N	2	0	Testvéreinek száma
OSZTALY	N	1	0	Hányadik osztályba jár
IRO	N	1	0	Osztályzata irodalomból
NYE	N	1	0	Osztályzata nyelvtanból
IDE	N	1	0	Osztályzata idegen nyelvből
TOR	N	1	0	Osztályzata történelemből
MAT	N	1	0	Osztályzata matematikából
FIZ	N	1	0	Osztályzata fizikából
KEM	N	1	0	Osztályzata kémiából
BIO	N	1	0	Osztályzata biológiából
FOL	N	1	0	Osztályzata földrajzból
TEC	N	1	0	Osztályzata technikából
ENE	N	1	0	Osztályzata énekből
TES	N	1	0	Osztályzata testnevelésből
RAJ	N	1	0	Osztályzata rajzból
SPORTOL	L	1	-	Sportol-e?
ENEKKAROS	L	1	-	Énekkaros-e?
APA_JOVED	N	6	0	Apja havi jövedelme
ANYA_JOVED	N	6	0	Anyja havi jövedelme

Feladatok

1. Nyisd meg az adatbázisfájlt!
2. Győződj meg róla, hogy a szerkezete valóban olyan, mint amit az előző táblázat mutat! (Ha nem olyan, akkor tedd olyanná!)
3. Írasd ki a 17. rekordban levő tanuló nevét, születési dátumát és matematika osztályzatát!
4. A 22. rekordban a testvérek számát növeld meg 1-gyel!
5. A 47. rekordban az apa jövedelmét növeld meg 1500 Ft-tal!
6. A 34., a 35. és a 48. rekordot jelöld ki törlésre!
7. A 48. rekord kijelöltségét szüntesd meg!
8. A törlésre kijelölt rekordokat távolítsd el végleg az adatbázisfájlból!
9. Írasd ki azoknak a tanulóknak a nevét és testvéreinek számát, akiknek 3 vagy annál több testvérük van!
10. Írasd ki a hetedik osztályos tanulók nevét és matematika osztályzatát!
11. Írasd ki azoknak a hatodik osztályos tanulóknak a nevét, akik jobb osztályzattal rendelkeznek fizikából, mint matematikából!
12. Írasd ki azoknak a tanulóknak a nevét, akik sportolnak!
13. Írasd ki azoknak a tanulóknak a nevét, akik énekkarba járnak!
14. Írasd ki azoknak a tanulóknak a nevét, akik sportolnak is, meg énekkarba is járnak!
15. Írasd ki azoknak a tanulóknak a nevét és testnevelés osztályzatát, akik sportolnak, de testnevelésből nem 5-ös az osztályzatuk!
16. Írasd ki azoknak a tanulóknak a nevét, akiknél a szülők havi összjövedelme nem éri el a 40000 Ft-ot!
17. Írasd ki az összes tanuló nevét és a családjában az egy főre eső havi jövedelmet!
18. Írasd ki az összes tanuló nevét és tanulmányi átlagát!
19. Írasd ki a 6. osztályos sportoló tanulók nevét és tanulmányi átlagát!
20. Írasd ki azoknak a 6. osztályba járó tanulóknak a nevét és tanulmányi átlagát, akik nem sportolnak!
21. Írasd ki azoknak a tanulóknak a nevét, akik történelemből vagy földrajzból 2-es osztályzattal rendelkeznek!
22. Írasd ki az összes Katalin keresztnévű tanuló teljes nevét és születési dátumát!

23. Tedd aktuálissá Orosz Péter rekordját, majd írasd ki a születési dátumát!

24. Írasd ki a tanulók osztályát, nevét és matematika osztályzatát úgy, hogy a kiírásnál a tanulók osztályok szerint növekvően, azon belül pedig névsor szerint növekvően kövessék egymást!

a) Először rendezéssel oldd meg a feladatot ! A rendezett fájlt a lemezmelléklet ISKOLA katalógusába tedd, és az azonosítója REN1.DBF legyen!

b) Oldd meg a feladatot indexeléssel is! Az indexfájlt a lemezmelléklet ISKOLA katalógusába tedd, és az azonosítója IND1.NDX legyen!

25. Módosítsd az adatbázisfájl szerkezetét: a RAJ mezőt követően hozz létre egy új mezőt:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Típsz.</i>	<i>Jelentés</i>
ATLAG	N	4	2	Tanulmányi átlaga

26. Az újonnan létrehozott ATLAG mező most még üres. A tartalmát változtasd meg automatikusan úgy, hogy a tanulmányi átlagot tartalmazza!

27. Számláld meg, hogy hány nyolcadik osztályos tanuló van az iskolában!

28. Számold ki külön az ötödik, a hatodik, a hetedik és a nyolcadik osztály tanulmányi átlagát!

29. Számold ki, hogy a hetedik osztályban mennyi a szülők összjövedelmének átlaga!

30. Minden osztályban tíz tanuló kap tanszersegélyt: azok, akiknél a legkevesebb a szülők összjövedelme. Írasd ki ezeknek a tanulóknak a nevét minden osztályból!

31. Írasd ki a tanulók nevét és tanulmányi átlagát a tanulmányi átlag csökkenő sorrendjében!

a) Először rendezéssel oldd meg a feladatot ! A rendezett fájlt a lemezmelléklet ISKOLA katalógusába tedd, és az azonosítója REN2.DBF legyen!

b) Oldd meg a feladatot indexeléssel is! Az indexfájlt a lemezmelléklet ISKOLA katalógusába tedd, és az azonosítója IND2.NDX legyen!

32. Írasd ki az 5. osztályos tanulók nevét és tanulmányi átlagát a tanulmányi átlag csökkenő sorrendjében!

33. Írasd ki a hatodik osztályos tanulók tantárgyi átlagát matematikából, fizikából és kémiából!

34. Számláld meg, hogy hány olyan tanuló van az iskolában, aki a BAK csillagképben született! (A BAK csillagkép december 23-tól január 20-ig tart.)

35. Zárd le az adatbázisfájlt!

Anyaggazdálkodás

A lemez melléklet EPITO katalógusában található az ANYAG.DBF adatbázisfájl, amely egy építőipari vállalat raktárában tárolt anyagok adatait tartalmazza.

Az adatbázisfájl szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
CIKK	C	5	-	Cikkszám
MEGNEVEZES	C	25	-	Az anyag megnevezése
MENNYISEG	N	9	2	Raktári mennyiség
AR	N	5	0	Egységár
VALTOZAS	D	8	-	Az utolsó változás időpontja
MINIMUM	N	7	0	Minimális készlet
ELADAS	L	1	-	Szabad-e eladni belőle?
A_MEKOD	N	2	0	Mértékegységkód

A cikkszám az anyagok azonosítását szolgálja.

Az utolsó változás időpontja azt fejezi ki, hogy mely napon változott legutoljára a raktári mennyiség.

A minimális készlet az a mennyiség, amelynél ha kevesebb van a raktárban, újra-rendelnek.

A mértékegység kóddal van megadva. Az EPITO katalógusban elhelyezett MERTEK.DBF adatbázisfájlból lehet megtudni, hogy egy mértékegységkód milyen mértékegységet jelöl. Ennek szerkezete és a mezők jelentése:

<i>Mezőnév</i>	<i>Típus</i>	<i>Hossz</i>	<i>Tj.sz.</i>	<i>Jelentés</i>
MEKOD	N	2	0	Mértékegységkód
ME_NEV	C	10	-	A mértékegység neve

Feladatok





1. Nyisd meg egy-egy munkaterületen az adatbázisfájlokat!
2. Győződj meg róla, hogy mindkét adatbázisfájl szerkezete olyan, mint amit az előző táblázatok mutatnak! (Ha valamelyik nem olyan, akkor tedd olyanná!)
3. Írasd ki a mértékegységek nevét és kódját!
4. Írasd ki az összes anyag nevét és értékét!

5. Számold ki a raktáron levő anyagok összértékét!
6. A mértékegységkód alapján kapcsold össze az ANYAGOK.DBF és a MERTEK.DBF adatbázisfájlokat (az előbbi legyen a vezér, az utóbbi pedig a szolga)!
7. Keresd meg azokat az anyagokat, amelyeknél kevesebb a raktáron levő készlet, mint a minimumkészlet (ezekből az anyagokból ugyanis sürgősen rendelni kell). Írasd ki az ilyen anyagoknál az anyag nevét, mértékegységét, valamint a minimumkészlet és a raktárkészlet különbségét!
8. Írasd ki az anyagok nevét és raktári értékét a raktári érték szerint növekvő sorrendben!
9. Keresd meg azokat az anyagokat, amelyeknek a mennyisége 1994. január 1. előtt változott utoljára (ezek elfekvő anyagok)! Írasd ki az ilyen anyagok nevét és értékét! Az érték mellé írasd oda, hogy „Ft”.
10. Számold ki az elfekvő anyagok összértékét!
11. Keresd meg az összes olyan anyagot, amelynek a nevében előfordul a „tégla” szó (ugyanis képet akarunk kapni a téglaelátottságról). Írasd ki az ilyen anyagok nevét, mértékegységét, mennyiségét, valamint az „eladható” illetve „nem adható el” megjegyzést attól függően, hogy lehet-e eladni az illető anyagból, vagy nem.
12. Számold ki a raktárban levő összes téglá összértékét!
13. Keresd meg azokat az anyagokat, amelyeknek az értéke 100000 Ft-nál nagyobb (ezek ugyanis jelentősen növelik a raktári összértéket, tehát érdemes odafigyelni rájuk)! Írasd ki az ilyen anyagok nevét, cikkszámát és értékét!
14. Írasd ki azoknak az anyagoknak a cikkszámát és csupa nagybetűvel a megnevezését, amelyeknek a cikkszámában az első helyen a 3-as számjegy szerepel!
15. Számítsd ki a raktárban levő összes olyan anyag értékét, amelynek a cikkszámában az első számjegy 2-es!
16. Zárd le mindkét adatbázisfájlt!

Fontos megjegyzés

A lemez mellékleten még egyéb katalógusok is találhatóak, és azok mindegyikében vannak adatbázisfájlok és feladatok. A feladatok szövege minden katalógusban a FELADAT.TXT fájlban található. Ennek a fájlnak a tartalma kiíratható a DOS TYPE parancsával vagy bármely szövegszerkesztő programmal.

A parancsok összefoglalása

A képernyő törlése	(17. és 63. fejezet)
CLEAR	
DOS-parancs végrehajtása	(108)
RUN DOS-parancs vagy ! DOS-parancs	
Végrehajtódik a megadott DOS-parancs, majd újra megjelenik a dBASE készletléti jele.	
Kilépés a dBASE-ből	(18, 63)
QUIT	
Adatbázisfájl létrehozása	(19, 63)
CREATE teljes_fájlazonosító	
Lehetővé teszi az adatbázisfájl szerkezetének megadását.	
  A szerkezet megadásának befejezése.	
Ezután, ha kérjük, lehetővé teszi az adatbázisfájl feltöltését adatokkal.	
  A feltöltés befejezése.	
Az adatbázisfájlok számbavétele	(21, 63)
DIR katalógushivatkozás[\\][*.*]	
Kiírja a képernyőre a katalógusban levő valamennyi adatbázisfájl azonosítóját, rekordjainak számát, készítésének ill. utolsó módosításának dátumát és méretét.	
Ha begépeljük a *.*-ot is, akkor a katalógusban levő valamennyi fájlról tájékoztat.	
Adatbázisfájl megnyitása vagy lezárása	(22, 23, 63)
USE [teljes_fájlazonosító]	
Ha meg van adva az adatbázisfájl teljes azonosítója, akkor az aktuális munkaterületen megnyitja az adatbázisfájlt.	
Ha nem adunk meg teljes fájlazonosítót, akkor lezárja az aktív adatbázisfájlt.	

Az adatbázisfájl szerkezetének kiírása**(24, 59, 63)****DISPLAY STRUCTURE [TO PRINT]**

vagy

LIST STRUCTURE [TO PRINT]

Mindkét parancs az aktív adatbázisfájl szerkezetét írja ki.

Ha a TO PRINT szavakat megadjuk, akkor a szerkezet a nyomtatóra is kiíródik.

A DISPLAY STRUCTURE a képernyő megtelésekor a kiírást leállítja, s csak akkor folytatja, ha leütünk egy tetszőleges billentyűt.

Aktuális rekord kijelölése**(26, 63, 79/a)****GO numerikus_konstans|TOP|BOTTOM**

Az aktív adatbázisfájlban kijelöli az aktuális rekordot.

Ha numerikus konstans adunk meg, akkor az annak megfelelő sorszámú rekordot teszi aktuálissá.

A TOP szó a legelső, a BOTTOM pedig a legutolsó rekordot teszi aktuálissá.

A főindexfájl csak akkor befolyásolja a parancs működését, ha a TOP vagy a BOTTOM szót használjuk.

SKIP [numerikus_konstans]

A numerikus konstans határozza meg, hogy az aktív adatbázisfájlban az aktuális rekordhoz képest mennyivel nagyobb vagy kisebb sorszámú rekord váljon aktuálissá.

Ha nem adunk meg numerikus konstans, az olyan, mintha 1-et adtunk volna meg.

A főindexfájl befolyásolja a parancs működését.



Rekordok hozzáfűzése az adatbázisfájl végéhez**(27, 30/a, 31, 63)****APPEND**



Lehetővé teszi az aktív adatbázisfájlban

— új rekordok hozzáfűzését

— bármely rekord módosítását

— bármely rekord kijelölését törlésre.

  Lapozás a rekordok között.

  A rekord kijelölése törlésre (vagy a kijelöltség megszüntetése).

  Kilépés a parancsból.

Rekord betoldása az adatbázisfájlba**(28, 63)****INSERT [BEFORE]**

Lehetővé teszi új rekord betoldását az aktív adatbázisfájlba.

Ha a BEFORE szót nem adjuk meg, akkor az aktuális rekord után toldja be a rekordot.

Ha a BEFORE szót megadjuk, akkor az aktuális rekord elé toldja be a rekordot.

Adatbázisfájl tartalmának módosítása**(30/b-d, 31, 51-55, 61, 63, 79/b, c)****CHANGE [hatáskör] [FIELDS mezőlista] [FOR feltétel]**

vagy


EDIT [hatáskör] [FIELDS mezőlista] [FOR feltétel]


Lehetővé teszi az aktív adatbázisfájlban a hatáskör és a feltétel által meghatározott rekordok

— módosítását

— kijelölését törlésre (vagy a kijelöltség megszüntetését).

Mezőlista megadásakor csak a benne szereplő mezők tartalmát lehet módosítani.

 Lapozás a rekordok között.

 A rekord kijelölése törlésre (vagy a kijelöltség megszüntetése).

 Kilépés a parancsból.

BROWSE [FIELDS mezőlista]


Lehetővé teszi az aktív adatbázisfájlban a kiválasztó sávval megjelölt rekord

— módosítását


— kijelölését törlésre (vagy a kijelöltség megszüntetését).


Megengedi új rekord hozzáfűzését is az adatbázisfájl végéhez.

Mezőlista megadásakor csak a benne szereplő mezők tartalmát lehet módosítani.

 Lapozás a rekordok között.

 A rekord kijelölése törlésre (vagy a kijelöltség megszüntetése).

 A képernyő görgetése egy mezővel jobbra.

 A képernyő görgetése egy mezővel balra.

 Kilépés a parancsból.

A főindexfájl befolyásolja mindhárom parancs működését.

Adatbázisfájl tartalmának kiírása	(29, 51-59, 63, 79/b)
DISPLAY [hatáskör] [kifejezéslista] [FOR feltétel] [OFF] [TO PRINT] vagy LIST [hatáskör] [kifejezéslista] [FOR feltétel] [OFF] [TO PRINT]	
<p>Kiírja az aktív adatbázisfájlnak a hatáskör és a feltétel által meghatározott rekordjaiból a kifejezéslistában szereplő kifejezések értékét.</p> <p>Ha nem adunk meg kifejezéslistát, akkor valamennyi mező tartalmát kiírja.</p> <p>Ha megadjuk az OFF szót, akkor a rekordok sorszámát nem írja ki.</p> <p>Ha megadjuk a TO PRINT szavakat, akkor a kiírás nyomtatóra is megtörténik.</p> <p>A DISPLAY parancs a képernyő megtelésekor a kiírást leállítja, s csak akkor folytatja, ha leütünk egy tetszőleges billentyűt.</p> <p>A főindexfájl befolyásolja a parancsok működését.</p>	
Rekordok kijelölése törlésre	(31/b, 51-55, 63, 79/b)
DELETE [hatáskör] [FOR feltétel]	
<p>Az aktív adatbázisfájlban törlésre jelöli ki a hatáskör és a feltétel által meghatározott rekordokat.</p> <p>A főindexfájl befolyásolja a parancs működését.</p>	
Rekordok törlésre való kijelöltségének megszüntetése	(31/c, 51-55, 63, 79/b)
RECALL [hatáskör] [FOR feltétel]	
<p>Az aktív adatbázisfájlban a hatáskör és a feltétel által meghatározott rekordok törlési kijelöltségét megszünteti.</p> <p>A főindexfájl befolyásolja a parancs működését.</p>	
Törlésre kijelölt rekordok eltávolítása az adatbázisfájlból	(31/d, 63)
PACK	
<p>Végleg eltávolítja az aktív adatbázisfájlból a törlésre kijelölt rekordokat.</p>	
Az adatbázisfájl összes rekordjának törlése	(32, 63)
ZAP	
<p>Visszavonhatatlanul kitörli az aktív adatbázisfájl összes rekordját, de a szerkezetét meghagyja.</p>	

Kifejezések értékének kiírása	(37, 46, 56, 63)
? kifejezéslista	
Az aktív adatbázisfájl aktuális rekordjából kiírja a képernyőre a kifejezéslistában szereplő kifejezések értékét.	
A mezők tartalmának automatikus módosítása	(64)
REPLACE [hatáskör] mezőnév₁ WITH kifejezés₁ [,mezőnév₂ WITH kifejezés₂...] [FOR feltétel]	
Az aktív adatbázisfájlban a hatáskör és a feltétel által meghatározott rekordokban megváltoztatja a WITH szó előtt megadott mező tartalmát. Az új tartalom a WITH szó után megadott kifejezés értéke lesz.	
Egyszerre több mező tartalma is módosítható.	
A főindexfájl befolyásolja a parancs működését.	
Összegzés	(65/a)
SUM [hatáskör] [numerikus_kifejezéslista] [FOR feltétel]	
A megadott numerikus kifejezések értékeit külön-külön összegzi az aktív adatbázisfájlban. Az összegzésben a hatáskör és a feltétel által meghatározott rekordok vesznek részt.	
A numerikus kifejezéslista hiánya esetén az aktív adatbázisfájl valamennyi numerikus mezője összegződik.	
A főindexfájl befolyásolja a parancs működését.	
Átlagolás	(65/a)
AVERAGE [hatáskör] [numerikus_kifejezéslista] [FOR feltétel]	
A numerikus kifejezéslistában szereplő kifejezések értékeit külön-külön átlagolja az aktív adatbázisfájlban. Az átlagolásban a hatáskör és a feltétel által meghatározott rekordok vesznek részt.	
A főindexfájl befolyásolja a parancs működését.	
Számlálás	(65/b)
COUNT [hatáskör] [FOR feltétel]	
Megszámlálja, hogy az aktív adatbázisfájlban mennyi a hatáskör és a feltétel által meghatározott rekordok száma. Az eredményt kiírja a képernyőre.	
A főindexfájl befolyásolja a parancs működését.	

Adatbázisfájl szerkezetének megváltoztatása**(66)****MODIFY STRUCTURE**


Az aktív adatbázisfájlban lehetővé teszi

- mező nevének, típusának, hosszának és a tizedesjegyei számának megváltoztatását
- mező törlését
- új mező betoldását.

  Mező törlése.

  Mező betoldása.

  Kilépés a parancsból.

 A módosítások érvénytelenítése.

Fájl másolása**(67)****COPY FILE teljes_fájlazonosító TO teljes_fájlazonosító**

Másolatot készít a TO szó előtt megadott fájlról. A másolat teljes azonosítóját a TO szó után kell megadni.

Ha a fájlazonosító nem tartalmaz kiterjesztést, akkor a név után a pontot ki kell tenni.

Fájl átnevezése**(68)****RENAME teljes_fájlazonosító TO teljes_fájlazonosító**

Átnevezi a TO szó előtt megadott fájlt. Az új teljes azonosítót a TO szó után kell megadni.

Ha a fájlazonosító nem tartalmaz kiterjesztést, akkor a név után a pontot ki kell tenni.

Fájl törlése**(69)****ERASE teljes_fájlazonosító**

Kitörli a megadott fájlt.

Ha a fájlazonosító nem tartalmaz kiterjesztést, akkor a név után a pontot ki kell tenni.

Adatbázisfájl másolása**(70)****COPY [hatáskör] [FIELDS mezőlista] TO teljes_fájlazonosító [FOR feltétel]**

Másolatot készít az aktív adatbázisfájlról.

A TO szó után a másolat teljes azonosítóját kell megadni.

A másolatba csak a hatáskör és a feltétel által meghatározott rekordokat viszi át.

Mezőlista megadásakor a másolatba csak a benne szereplő mezőket viszi át.

Rendezés	(73)
SORT [hatáskör] ON igénylista [FOR feltétel] TO teljes_fájlazonosító	
<p>A rendezésben az aktív adatbázisfájlnak a hatáskör és a feltétel által meghatározott rekordjai vesznek részt.</p> <p>A rendezés eredményeként létrejövő adatbázisfájl teljes azonosítóját a TO szó után kell megadni.</p> <p>A rendezési szempontokat az igénylistával adjuk meg.</p> <p>Az igénylista több elemből állhat. Az elemeket vessző választja el egymástól.</p> <p>Egy elem formája: mezőnév[/[A D]][C]]</p> <p>Az A növekvő sorrendet, a D csökkenő sorrendet határoz meg. A C hatására a kis- és a nagybetűk között nincs különbség.</p>	
Indexfájl létrehozása	(76)
INDEX ON kulcskifejezés TO teljes_fájlazonosító	
<p>Az aktív adatbázisfájllal kapcsolatban olyan indexfájlt hoz létre, amely a kulcskifejezés szerint növekvő kezelési sorrendet biztosítja. A kulcskifejezés típusa nem lehet logikai.</p>	
Indexfájlok megnyitása és lezárása	(77)
SET INDEX TO [teljes_fájlazonosítók_listája]	
<p>Az aktív adatbázisfájllal kapcsolódó indexfájlok közül azokat nyitja meg, amelyeket a TO szót követően felsorolunk, egymástól vesszővel elválasztva.</p> <p>Ha a TO szó után nem adunk meg semmit, akkor az aktív adatbázisfájllal kapcsolatban megnyitott összes indexfájlt lezárja.</p>	
Főindexfájl kijelölése	(78)
SET ORDER TO [numerikus_konstans]	
<p>A numerikus konstans határozza meg, hogy az aktív adatbázisfájllal kapcsolatban legutoljára kiadott SET INDEX parancsban szereplő indexfájlok közül hányadik váljon főindexfájllá.</p>	
Újraindexelés	(80)
REINDEX	
<p>Az aktív adatbázisfájllal kapcsolatban újra létrehozza a nyitva levő indexfájlokat, változatlan kulcskifejezéssel.</p>	

Keresés az adatbázisfájlban	(71, 79/b)
<p>LOCATE [hatáskör] [FOR feltétel] és CONTINUE</p>	
<p>A keresést az aktív adatbázisfájlban a két parancs egyesült erővel valósítja meg. A LOCATE a hatáskör és a feltétel által meghatározott rekordok közül a legelsőt teszi aktuálissá, a CONTINUE pedig mindig a következőt. A főindexfájl befolyásolja a parancsok működését.</p>	
Gyorskeresés az adatbázisfájlban	(82)
<p>FIND karakteres_konstans numerikus_konstans</p>	
<p>A gyorskeresést az aktív adatbázisfájlban végzi. Feltétele, hogy olyan főindexfájl legyen kijelölve, amely karakteres vagy numerikus kulcskifejezéssel lett létrehozva. A legelső olyan rekordot teszi aktuálissá, amelynél a kulcskifejezés értéke megegyezik a parancsban megadott konstans értékével.</p>	
Munkaterület váltása	(86)
<p>SELECT munkaterület-azonosító</p>	
<p>Aktuálissá teszi a munkaterület-azonosító által meghatározott munkaterületet. A munkaterület-azonosító háromféleképpen adható meg: — az 1-től 10-ig terjedő egész számokkal — az A-tól J-ig terjedő betűkkel — az illető munkaterületen megnyitott adatbázisfájl nevével.</p>	
Adatbázisfájlok összekapcsolása	(90, 92, 93)
<p>SET RELATION TO kapcsoló-kifejezés INTO fájlnev</p>	
<p>Kapcsoló-kifejezések segítségével az aktív adatbázisfájlt összekapcsolja egy másik munkaterületen megnyitott adatbázisfájllal. Ez utóbbi adatbázisfájl nevét a INTO szó után kell megadni. A SET RELATION TO parancs az aktív adatbázisfájl összekapcsoltságát megszünteti.</p>	

A könyv segítséget nyújt azoknak,
akik a munkájuk során felhalmozódott adatokat
a jövőben számítógépen kívánják kezelni.

Az ismeretek elsajátítását a fejezetek végén
példák és feladatok könnyítik meg.

A lemezmelléklet arra szolgál, hogy az olvasó
a könyv áttanulmányozása után
ellenőrizhesse tudását.

Stílusa közérthető.

Általános és középiskolai oktatásra,
valamint magántanulásra és
tanfolyami képzésre egyaránt használható.

A sorozat előző kötete:

Katona Viktor: Első PC-könyvem

(Alapismertetek, DOS, szövegszerkesztés,
Norton Commander)

A lemezmellékleten Adamát Lajos Sch. P.:
Magyar nyelvű HELP a Norton Commanderhez

A sorozat következő kötete:

**Katona Viktor: Harmadik PC-könyvem
(Windows)**