

SZOLEK ANDRÁS

dBASE III PLUS



**LAPOZGATÓ
SOROZAT**

Josef Havel
Max - Stromeyer - Str. 9
7750 Konstanz

Szolek András

dBASE III PLUS

Szolek András

dBASE III PLUS

Műszaki Könyvkiadó ■ Novotrade Rt.

Budapest, 1988

LAPOZGATÓ SOROZAT

Lektorálta:

ZOLTÁN LÁSZLÓ
okl. villamosmérnök

BORS GYÖRGY
okl. matematikus

Szerkesztő:

GYÖRKE TIBORNÉ
okl. villamosmérnök

© Szolek András, Budapest, 1988

ETO: 681.3.0.6:519.683

ISBN: 963 10 7747 0

Tartalom

A dBASE III PLUS technikai jellemzői 8

A dBASE III PLUS parancsok funkció szerinti csoportosítása 10

dBASE III PLUS parancsok 16

A dBASE III PLUS függvények funkció szerinti csoportosítása 66

dBASE III PLUS függvények 69

Ez a referenciatétel feltételezi, hogy az olvasó tisztában van az Ashton-Tate által forgalmazott, dBASE III relációs adatbázis kezelő rendszer alapfogalmaival.

Jelölések:

- [] a kapcsos zárójelek közötti rész opcionális, használata nem kötelező.
- | kizáró VAGY. Vagy az egyik, vagy a másik.
- n numerikus típusú kifejezés.
- str string (karakterlánc) típusú kifejezés.
- log logikai típusú kifejezés.
- d dátum típusú kifejezés.
- kif tetszőleges típusú kifejezés.
- xxx-lista az xxx típusból tetszőleges számú egymás utáni elem.
- hatáskör azon rekordok, amelyekre az adott utasítás hatása kiterjed.
 - all az adatbázis összes rekordjára.
 - rest az aktuális rekordtól kezdve az összes rekordra.
 - next n az aktuális rekordtól kezdve n rekordra.
 - record n az n -edik rekordra.
- a nyilat követő szöveg a dBASE III PLUS képernyőre írt válasza.

A dBASE III PLUS technikai jellemzői

Maximális rekordszám adatbázisonként	1 milliárd rekord.
Maximális file-méret adatbázisonként	2 milliárd byte.
Maximális rekordméret	4000 byte egy adatbázisban.
Maximális mezőszám rekordonként	128 mező.

Egy mező méretére vonatkozó megkötések:

numerikus típusú mező	max. 19 byte,
karakter típusú mező	max. 254 byte,
logikai típusú mező	1 byte,
dátum típusú mező	8 byte,
memo típusú mező	max. 5000 byte, vagy a használt szövegszerkesztő által még meg- engedett méret.

Az egyidejűleg nyitvatartható file-ok száma	max. 15.
Az egyidejűleg nyitvatartható adatbázisok száma	max. 10.
Az egyidejűleg egy adatbázishoz rendelt indexfile-ok száma	max. 7.
Az egyidejűleg egy adatbázishoz rendelt formátumfile-ok száma	max. 1.

A számok pontossága: 15 egész és 9 tizedesjegy.
Számok összehasonlításakor a
rendszer 13 számjegy pontosság-
gal dolgozik.

Memóriaváltozók:

numerikus típus	9 byte,
karakteres típus	(a karakterlánc hossza +2) byte,
dátum típus	9 byte,
logikai típus	2 byte.

Egyszerre max. 256 memóriaváltozót lehet használni.
Az erre fenntartott memóriaterület alapértelmezésben
6000 byte.

A parancssor maximális hossza 254 byte.

A dBASE III PLUS file-típusai:

bináris file	.bin
katalógusfile	.cat
parancsfile	.prg
adatbázisfile	.dbf
előző adatbázis- és programfile	.bak
adatbázis-memofile	.dbt
előző adatbázis-memofile	.tbk
formátumfile	.fmt
indexfile	.ndx
címke- (etikett-) file	.lbl
memóriafile	.mem
kérelemfile	.qry
riportfile	.frm
képernyőfile	.scr
szöveges file	.txt
emlékeztetőfile	.vue

A dBASE III PLUS parancsok funkció szerinti csoportosítása

A felhasználót támogató parancsok

ASSIST	18
DIR	26
DISPLAY MEMORY	27
DISPLAY STATUS	27
DISPLAY STRUCTURE	27
HELP	31

File-okat létrehozó parancsok

COPY	22
COPY FILE	22
CREATE	23
CREATE LABEL	24
CREATE QUERY	24
CREATE REPORT	24
CREATE SCREEN	25
CREATE VIEW	25
EXPORT	30
IMPORT	32
INDEX	32
JOIN	33
MODIFY COMMAND	36
MODIFY STRUCTURE	36
SAVE	44
SET CATALOG TO	46
SORT	61
TOTAL	62

Adatbeviteli parancsok

APPEND	18
BROWSE	19
INSERT	32

Adatmódosító parancsok

BROWSE	19
CHANGE	20
EDIT	29
READ	40
REPLACE	41
UPDATE	63

Adattörölő parancsok

DELETE	26
PACK	37
RECALL	40
ZAP	65

Adatmegjelenítő parancsok

@[... SAY ... GET][... TO ...]	16
?	16
??	16
AVERAGE	19
BROWSE	19
COUNT	23
DISPLAY	27
LABEL	33
LIST	34
REPORT	42
SUM	61

Az aktuális rekord pozícióját változtató parancsok

CONTINUE	22
FIND	30
GO GOTO	31
LOCATE	35
SEEK	44
SKIP	60

Adatbázisfile-okra vonatkozó parancsok

APPEND FROM	18
CLOSE	21
COPY	22
ERASE DELETE FILE	30, 26
MODIFY STRUCTURE	36
RENAME	41
SELECT	45
SORT	61
USE	64

Más típusú file-okra vonatkozó parancsok

MODIFY COMMAND	36
MODIFY LABEL	24
MODIFY QUERY	24
MODIFY REPORT	24
MODIFY SCREEN	25
MODIFY VIEW	25
REINDEX	40

Memóriaváltozókra vonatkozó parancsok

ACCEPT	18
AVERAGE	19
CLEAR ALL	21
CLEAR MEMORY	21
COUNT	23
DISPLAY MEMORY	27
INPUT	32
PARAMETERS	38
PRIVATE	38
PUBLIC	39
READ	40
RELEASE	41
RESTORE	42
SAVE	44
STORE	61
SUM	61
WAIT	64

Programozást támogató parancsok

ACCEPT	18
CANCEL	20
DO	28
DO WHILE	28
EXIT	28
LOOP	28
ENDDO	28
DO CASE	28
OTHERWISE	28
ENDCASE	28
IF	31
ELSE	31
ENDIF	31
INPUT	32
MODIFY COMMAND	36
PROCEDURE	39
QUIT	39
RETRY	43
RETURN	43
TEXT	62
ENDTEXT	62
WAIT	64

Rendszerkörnyezetre vonatkozó parancsok

CLEAR	21
EJECT	29
SET	45

Paraméterbeállító parancsok

SET ALTERNATE on OFF	45
SET ALTERNATE TO	45
SET BELL ON off	46
SET CARRY on OFF	46
SET CATALOG on OFF	46
SET CATALOG TO	46
SET CENTURY on OFF	47
SET COLOR on OFF	47
SET COLOR TO	47
SET CONFIRM on OFF	48
SET CONSOLE ON off	48
SET DATE	48
SET DEBUG on OFF	49
SET DECIMALS TO	49
SET DEFAULT TO	49
SET DELETED on OFF	49
SET DELIMITERS on OFF	50
SET DELIMITERS TO	50
SET DEVICE TO SCREEN print	50
SET DOHISTORY on OFF	50
SET ECHO on OFF	51
SET ESCAPE ON off	51
SET EXACT on OFF	51
SET FIELDS on OFF	52
SET FIELDS TO	52
SET FILTER TO	52
SET FIXED on OFF	52
SET FORMAT TO	53
SET FUNCTION TO	53
SET HEADING ON off	53
SET HELP ON off	53
SET HISTORY ON off	54
SET HISTORY TO	54
SET INDEX TO	54
SET INTENSITY ON off	54
SET MARGIN TO	55
SET MEMOWIDTH TO	55
SET MENUS ON off	55
SET MESSAGE TO	55
SET ORDER TO	56

SET PATH TO	56
SET PRINT on OFF	56
SET PRINTER TO	57
SET PROCEDURE TO	57
SET RELATION TO	57
SET SAFETY ON off	58
SET SCOREBOARD ON off	58
SET STATUS ON off	58
SET STEP on OFF	58
SET TALK ON off	59
SET TITLE ON off	59
SET TYPEAHEAD TO	59
SET UNIQUE on OFF	60
SET VIEW TO	60

A külső programokhoz való csatlakozás parancsai

CALL	20
LOAD	35
RUN	44

Hibakeresést támogató parancsok

DISPLAY HISTORY	27
LIST HISTORY	34
RESUME	43
SET DEBUG on OFF	49
SET DOHISTORY on OFF	50
SET ECHO on OFF	51
SET HISTORY TO	54
SET STEP on OFF	58
SUSPEND	62

Feltételtől függő vezérlésátadás

ON ERROR	37
ON ESCAPE	37
ON KEY	37

dBASE III PLUS parancsok

?

1. ? kifejezéslista
2. ?? kifejezéslista

Hatása

1. a következő sorban jeleníti meg a kifejezéslistát.
2. az aktuális sorban jeleníti meg a kifejezéslistát.

Példák

? 'abcd', 8*(1024 + 16), anyagszám
?? 'abcd', 8, anyagszám

@

1. @ sor,oszlop [[**SAY** kifejezés [**PICTURE** minta]]
[**GET** változó [**PICTURE** minta]
[**RANGE** alsó határ, felső határ]]]
2. @ sor,oszlop **CLEAR**
3. @ sor1,oszlop1 [**CLEAR**] **TO** sor2,oszlop2
[**DOUBLE**]

Hatása

1. adatok előre megtervezett forma szerinti megjelenítése és bekérése.
2. a képernyő meghatározott részének letörése.
3. a képernyő meghatározott részének [letörése és] bekeretezése [dupla vonallal].

Megjegyzés minta=[**PICTURE** függvény + ' ' +]maszk

PICTURE függvények:

- @C pozitív szám után CR-t ír ki (*credit*),
- @X negatív szám után DB-t ír ki (*debit*),
- @(negatív számot zárójelben jelenít meg,
- @B balra igazítja a számot,
- @Z a vezető nullákat szóközként jeleníti meg,
- @D amerikai dátumformátum (hónap/nap/év),
- @E európai dátumformátum (nap/hónap/év),
- @A csak alfabetikus karaktert enged meg,
- @! minden betűt nagybetűvé alakít,
- @R a maszkba írt betűket megjeleníti, de a betűk nem kerülnek be a mezőbe,
- @Sn a mezőszélességet n karakternyire állítja be, és az ennél hosszabb karaktersorozatot vízszintesen „legördíti”.

maszkszimbólumok:

- 9 csak számjegyeket és előjelet fogad el,
- # csak számjegyeket, előjelet és szóközt fogad el,
- A csak betűket fogad el,
- L csak logikai értéket fogad el,
- Y csak az Y, y, N, n értékeket fogadja el nagybetűre átalakítva,
- N csak számjegyeket és betűket fogad el,
- X bármilyen karaktert elfogad,
- ! a kisbetűket nagybetűkre alakítja át,
- \$ a vezető nullák helyén a \$ jelet jeleníti meg,
- * a vezető nullák helyén a * jelet jeleníti meg,
- . a tizedespontot jelöli ki,
- , ezres tagolást tesz lehetővé.

Példák

- @ 10,20 say 'Kérem a személyi számát: ' get
szemszám picture '#-#####-####'
- @ 10,20 clear
- @ 10,20 say szám1 picture '@X \$\$\$,\$\$\$.\$\$'

ACCEPT

ACCEPT [kiírás] **TO** memóriaváltozó

Hatása karaktersorozatot olvas be a memóriaváltozóba.

Példa **accept** 'Kérem az ön azonosítóját: ' **to** azonosító

APPEND

1. **APPEND** [BLANK]

2. **APPEND FROM** file-név [FOR feltétel] [TYPE]
[file-típus]

Hatása 1. az adatbázis végéhez hozzátesz egy [üres] rekordot.
2. egy [másfajta] file-ból rekordo(ka)t másol át.

Megjegyzés file-típus:

DELIMITED [WITH BLANK | <karakter>]
mezőelválasztó karakter,

SDF szövegfile (ASCII file),

DIF VisiCalc file,

SYLK Multiplan file,

WKS Lotus 1–2–3 file (1.A verzió).

Példák **append blank**

append from raktár for anyagszám<10000

append from adó86.wks

ASSIST

ASSIST

Hatása a dBASE III PLUS leggyakrabban használt parancsainak menüvezérelt formával történő elérése.

Példa **assist**

AVERAGE

AVERAGE [n-lista][hatáskör][**WHILE** feltétel]
[**FOR** feltétel] **TO** [memóriaváltozó-lista]

Hatása számtani átlagot számít [bizonyos feltételt teljesítő adatrekordok adott mezőire].

Példa **average** **egységár,mennyiség to átlagegységár,átlagmennyiség**

BROWSE

BROWSE [**FIELDS** mezőlista] [**LOCK** n]
[**FREEZE** mező] [**NOFOLLOW**] [**NOMENU**]
[**WIDTH** n] [**NOAPPEND**]

Hatása az adatbázis rekordjainak módosítása teljes képernyős szerkesztéssel.

Megjegyzés **FIELDS** milyen mezőket akarunk használni,
LOCK hány mezőt akarunk állandóan a képernyőn tartani,
FREEZE csak a megadott mező tartalmát lehet fölülírni,
NOFOLLOW csak indexelt állomány esetén használható; a kulcsmező megváltoztatásakor *nem* a megváltoztatott kulcsértékű rekord után logikailag következő rekord jön, hanem a régi kulcsérték szerint következő,
NOMENU a menüt nem írja ki,
WIDTH a mezőszélesség korlátozása (4–99-ig),
NOAPPEND a file végére érve nem engedi újabb rekordok hozzáillesztését az adatbázishoz.

Példa **browse** **fields** **anyagszám,raktárszám** **nofollow** **nomenu** **width** **30**

CALL

CALL modulnév [**WITH** str | memóriaváltozó]

Hatása a már betöltött gépi kódú program végrehajtása (részletesebben l. a **LOAD** parancsnál).

Példa `call cursor with chr(18)`

CANCEL

CANCEL

Hatása a program végrehajtásának befejezése, az összes nyitott programfile lezárása, visszatérés a **dBASE III PLUS** parancs üzemmódjába.

Megjegyzés az eljárásfile nyitva marad!

Példa `if error() = 16`
`cancel`
`endif`

CHANGE

CHANGE [hatáskör] [**FIELDS** mezőlista]
[**WHILE** feltétel],[**FOR** feltétel]

Hatása az adatbázis adott mezőinek teljes képernyős módosítása.

Megjegyzés a memo típusú mezők módosítása a következőképpen történik: ráállunk a memo föl-iratra, és lenyomjuk a **Ctrl+PgDn** billentyűket. A módosítást a **Ctrl+End** billentyűvel fejezzük be.

Példa `change next 20 fields anyagszám while`
`kelt<ctod('86.12.31')`

CLEAR

1. CLEAR
2. CLEAR ALL
3. CLEAR FIELDS
4. CLEAR GETS
5. CLEAR MEMORY
6. CLEAR TYPEAHEAD

Hatása

1. törli a képernyőt, és érvényteleníti a @... GET parancsokat.
2. lezárja az összes nyitott adatbázist, és törli az összes memóriaváltozót.
3. feloldja a SET FIELDS TO parancs mező-kijelöléseit.
4. érvényteleníti a korábban kiadott @... GET parancsokat.
5. törli az összes memóriaváltozót.
6. törli az összes előre begépelte billentyű-ütést.

Példa `clear`

CLOSE

1. CLOSE file-típus
2. CLOSE ALL

Hatása

1. lezárja az összes file típusú nyitott file-t.
2. mindenféle file-t lezár.

Megjegyzés file-típus: ALTERNATE,
DATABASES,
FORMAT,
INDEX,
PROCEDURE.

Példa `close all`

CONTINUE

CONTINUE

Hatása megkeresi az adatbázisban azt a következő rekordot, amely a már előzőleg kiadott LOCATE parancs feltételeinek megfelel.

Példa `continue`

COPY

1. **COPY TO** új file-név [hatáskör] [**FIELDS** mezőlista] [**FOR** feltétel] [**WHILE** feltétel] [**TYPE**] [file-típus]
2. **COPY FILE** file-név1 **TO** file-név2
3. **COPY STRUCTURE TO** file-név [**FIELDS** mezőlista]
4. **COPY TO** új file **STRUCTURE EXTENDED**

Hatása

1. az adatbázis [bizonyos feltételeknek eleget tevő] rekordjait egy új adatbázisba másolja át.
2. bármely nem nyitott file-ról másolatot készít.
3. az új file-ba csak az aktív adatbázis struktúráját másolja át.
4. létrehoz egy új adatbázist a következő 4 mezővel: **FIELD_NAME**, **FIELD_TYPE**, **FIELD_LEN**, **FIELD_DEC**. Az új adatbázis az aktív adatbázis mezőneveit, mezőtípusait, mezőhosszait, numerikus mezők esetén a tizedesrészt és hosszát tartalmazza.

Megjegyzés a lehetséges file-típusok:

DELIMITED [WITH BLANK | <karakter>]
mezőelválasztó karakter,
SDF szövegfájl (ASCII fájl),
DIF VisiCalc fájl,
SYLK Multiplan fájl,
WKS Lotus 1–2–3 fájl (1.A verzió).

Példák **copy to mentés fields anyagszám,új_ár**
for kelt<ctod('86.12.31')
copy file read.me to read2.me
copy structure to mentés
copy to leírás structure extended

COUNT

COUNT [hatáskör] [**WHILE** feltétel] [**FOR** feltétel]
[**TO** memóriaváltozó]

Hatása a hatáskörnek és a feltételeknek eleget tevő
rekordok számát adja meg [a memóriaváltozó-
ban].

Példa **count for megye='Hajdu'.or.megye='Szabolcs'to a1**

CREATE

1. **CREATE** file-név
2. **CREATE** új file-név **FROM** leírófile

Hatása 1. új adatbázist hoz létre.
2. egy **COPY STRUCTURE EXTENDED** mó-
don létrehozott leírófile-ból kiolvasott
mezőnevekkel, mezőtípusokkal stb. hoz
létre új adatbázist.

Példák **create raktár**
create raktár from mentés

CREATE | MODIFY LABEL

CREATE | MODIFY LABEL file-név | ?

Hatása teljes képernyős parancs, amellyel létrehozhatók vagy módosíthatók az .lbl típusú file-ok. A ? hatására az aktuális katalógusfile-ból választhatjuk ki az .lbl típusú file-t.

Példa `create label rendelés`

CREATE | MODIFY QUERY

CREATE | MODIFY QUERY file-név | ?

Hatása létrehoz, ill. módosít egy .qry típusú file-t, amellyel csak bizonyos feltételeknek eleget tevő rekordok érhetőek el. A ? hatására az aktuális katalógusfile-ból választhatjuk ki a .qry típusú file-t.

Példa `modify query november`

CREATE | MODIFY REPORT

CREATE | MODIFY REPORT file-név | ?

Hatása teljes képernyős parancs, amellyel létrehozhatók vagy módosíthatók az .frm típusú file-ok. Ezekkel a file-okkal különböző táblázatokat tudunk kialakítani. A ? hatására az éppen aktuális katalógusfile-ból választhatjuk ki a megfelelő .frm típusú file-t.

Példa `create report jelentés`

CREATE | MODIFY SCREEN

CREATE | MODIFY SCREEN file-név | ?

Hatása teljes képernyős parancs, amellyel létrehozhatók vagy módosíthatók az .scr típusú file-ok. Ezekkel a file-okkal különböző képernyőterveket tudunk kialakítani. A ? hatására az éppen aktuális katalógusfile-ból választhatjuk ki a megfelelő .scr típusú file-t.

Példa `create screen bevitel1`

CREATE | MODIFY VIEW

CREATE | MODIFY VIEW file-név | ?

Hatása teljes képernyős parancs, amellyel létrehozhatók vagy módosíthatók a .vue típusú file-ok. Ezekkel a file-okkal relációk segítségével különböző adatbázisokat lehet összekapcsolni, és látszólag egyetlen file-ként kezelni. A ? hatására az aktuális katalógusfile-ból választhatjuk ki a megfelelő .vue típusú file-t.

Példa `create view értékesít`

CREATE VIEW FROM ENVIRONMENT

CREATE VIEW file-név FROM ENVIRONMENT

Hatása létrehoz egy .vue típusú file-t, amely a nyitott adatbázisokat, a formátum- és indexfile-okat, a relációkat, a SET FIELDS-eket és a szűrőket tartalmazza.

Példa `create view test from environment`

DELETE

DELETE [hatáskör] [**WHILE** feltétel] [**FOR** feltétel]

Hatása logikailag törli a hatáskörben megjelölt és a feltételeknek eleget tevő rekordokat. (Önmagában az aktuális rekordot törli.)

Példa `delete all for kelt<ctod('86.12.30')`

DIR

DIR [meghajtó:] [útvonal] [file-név (csoport)]

Hatása a meghajtóban levő lemez útvonallal kijelölt alkönyvtárának file-jairól ad tájékoztatást.

Példa `dir a:* .mem`

DISPLAY

1. **DISPLAY** [hatáskör] [[**FIELDS**] kifejezéslista]
[**WHILE** feltétel] [**FOR** feltétel] [**OFF**]
[**TO PRINT**]
2. **DISPLAY HISTORY** [**LAST** *n*] [**TO PRINT**]
3. **DISPLAY MEMORY** [**TO PRINT**]
4. **DISPLAY STATUS** [**TO PRINT**]
5. **DISPLAY STRUCTURE** [**TO PRINT**]

- Hatása
1. megjeleníti az adatbázis azon rekordjainak tartalmát, amelyek a feltételeknek eleget tesznek. (Önmagában az aktuális rekord teljes tartalmát írja ki.)
Ha a képernyő megtelik, egy billentyű-ütésre vár.
 2. a parancspufferben levő [utolsó *n*] már végrehajtott parancso(ka)t jeleníti meg.
 3. kiírja a memóriaváltozók nevét, típusát, értékét és státusát [nyomtatóra is].
 4. megjeleníti többek között a nyitott adatbázisok nevét,
az adatbázisok indexfile-jainak nevét,
az adatbázisok között éppen fennálló relációkat,
a nyitott memofile-ok nevét,
a beállított szűrőfeltételeket,
a nyitott formátumfile-okat,
a SET kapcsolók állapotát,
a funkcióbillentyűk jelentését.
 5. megjeleníti az adatbázis nevét, a benne levő rekordok számát, utolsó módosításának idejét, mezőinek nevét, típusát, méretét, valamint egy rekordjának nagyságát byteban kifejezve.

Megjegyzés **OFF** a rekordsorszámot ne jelenítse meg.

TO PRINT nyomtatóra is írja ki.

Példa `display all for anyagszám>100 off to print`

1. **DO** file-név [**WITH** paraméterlista]

2. **DO CASE**

CASE feltétel_1
parancsok_1

CASE feltétel_2
parancsok_2

.

.

.

CASE feltétel_n
parancsok_n

[**OTHERWISE**
parancsok]

ENDCASE

3. **DO WHILE** feltétel

parancsok

[**EXIT**]

parancsok

[**LOOP**]

parancsok

ENDDO

- Hatása
1. programvégrehajtás [paraméterátadással].
 2. többirányú elágazást enged meg.
 3. ciklikus tevékenységsorozatot valósít meg.

Megjegyzés **EXIT** kilép a **DO WHILE** ciklusból.
LOOP a mögötte álló utasításokat nem hajtja végre, hanem a ciklus elejére ugrik.

Példák

```

do      fejléc with anyagszám ,date()
do case
  case  a=1
        ? 'Az illető iskolás'
  case  a=2
        ? 'Az illető egyetemista'
  otherwise
        ? 'Az illető valószínűleg nem dolgozik'
endcase

do while a<10
  ? a
  a=a+1
enddo

```

EDIT

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

Hatása teljes képernyős parancs, amellyel módosíthatjuk az adatbázis kijelölt és bizonyos feltételeknek eleget tevő rekordjait. (Önmagában az aktuális rekorddal indít.)

Példa `edit all while anyagszám < 100`

EJECT

EJECT

Hatása a nyomtatónak lapdobás parancsot ad.

Példa `eject`

ERASE

ERASE file-név | ?

Hatása a file-név nevű file-t törli a lemeztől. A ? hatására a katalógusfile-ból választhatunk.

Példa **erase color.mem**

EXIT

L. DO WHILE

EXPORT

EXPORT TO file-név [TYPE] PFS

Hatása az adatbázisból, valamint a hozzá kapcsolt index- és formátumfile-okból egy PFS-file-t készít.

Példa **export to pfsraktár type pfs**

FIND

FIND str | n

Hatása az indexelt adatbázisban megkeresi azt az első rekordot, amelynek kulcsa megegyezik a megadott str-rel vagy n -nel.

Példa **find Kovács**

GO | GOTO

1. [GO | GOTO] *n*
2. GO | GOTO TOP | BOTTOM

Hatása 1. a rekordmutató az *n*-edik rekordra áll.
 2. a rekordmutató az adatbázis első | utolsó rekordjára áll.

Példa **go top**

HELP

HELP [utasítás]

Hatása segítség a dBASE III PLUS használatához.

Példa **help find**

IF

IF feltétel
 parancsok
[ELSE
 parancsok]
ENDIF

Hatása a feltételes elágazás megvalósítása a programban.

Példa **if a<10**
 ? 'Túl kicsi'
 else
 ? 'Megfelelő'
 endif

IMPORT

IMPORT FROM file-név [TYPE] PFS

Hatása a file-név nevű PFS-file-ból adatbázist hoz létre.

Példa `import from pfsanyag type pfs`

INDEX

INDEX [ON kulcsképző kifejezés TO file-név [UNIQUE]]

Hatása indexfile-t hoz létre az adatbázishoz.

Példa `index on str (anyagszám,10)+str(raktárszám,3) + dtoc(kelt) to raktár`

INPUT

INPUT [szöveg] TO memóriaváltozó

Hatása adatbekérés a billentyűzetről a memóriaváltozóba.

Példa `input 'Kérem a vezetéknevét: 'to vezetéknev`

INSERT

INSERT [BLANK] [BEFORE]

Hatása új rekordot szúr be az aktuális rekord után.

Megjegyzés BLANK üres rekordot szúr be.

BEFORE az aktuális rekord elé szúrja be az új rekordot.

Példa `insert blank before`

JOIN

JOIN WITH adatbázisnév **TO** file-név **FOR** feltétel
[**FIELDS** mezőlista]

Hatása az aktuális és a megadott adatbázisból új adatbázist hoz létre úgy, hogy annak rekordjai bizonyos feltételeknek eleget tegyenek.

Példa `join with készlet to kivonat for kelt<ctod('86.12.31')`
`fields anyagszám,raktárszám,készlet ->meny-`
`nyiség, készlet -> mozgásnem`

LABEL FORM

LABEL FORM címkefile-név | ? [hatáskör] [**SAMPLE**]
[**WHILE** feltétel] [**FOR** feltétel]
[**TO PRINT** | **TO FILE** file-név]

Hatása a címkefile-név segítségével megjeleníti, ill. ki-nyomtatja a **MODIFY LABEL** paranccsal létrehozott címkéket (etiketteket).

Megjegyzés **SAMPLE** ellenőrizni akarjuk, hogy a printer stb. helyesen van-e beállítva.

Példa `label form levelek next 100 to print`
`ADRESSE FOR TO PRINTER`

`.label form c:\dbase\adresses for`
`NAME=" " TO PRINT`

1. **LIST** [OFF] [hatáskör] [kifejezéslista]
[WHILE feltétel] [FOR feltétel] [TO PRINT]
2. **LIST HISTORY** [LAST *n*] [TO PRINT]
3. **LIST MEMORY** [TO PRINT]
4. **LIST STATUS** [TO PRINT]
5. **LIST STRUCTURE** [TO PRINT]

Hatása

1. megjeleníti az adatbázis azon rekordjainak tartalmát, amelyek a feltételeknek eleget tesznek. (Önmagában az aktuális rekord teljes tartalmát írja ki.)
Megjelenítés közben – ellentétben a DISPLAY paranccsal – nem vár billentyűleütésre.
2. a parancspufferben levő [utolsó *n*] már végrehajtott parancso(ka)t jeleníti meg.
3. kiírja a memóriaváltozók nevét, típusát, értékét és státusát [nyomtatóra is].
4. megjeleníti többek között: a nyitott adatbázisok nevét,
az adatbázisok indexfile-jainak nevét,
az adatbázisok között éppen fennálló relációkat,
a nyitott memofile-ok nevét,
a beállított szűrőfeltételeket,
a nyitott formátumfile-okat,
a SET kapcsolók állapotát,
a funkcióbillentyűk jelentését.
5. megjeleníti az adatbázis nevét, a benne levő rekordok számát, utolsó módosításának idejét, mezőinek nevét, típusát, méretét, valamint egy rekordjának nagyságát byte-ban kifejezve.

Megjegyzés OFF a rekordsorszámot ne jelenítse meg.
TO PRINT nyomtatóra is írja ki.

Példa

`list all for anyagszám>100 off to print`

*list off for ZEITUNG='DOS 1/90'
TO PRINT*

LOAD

LOAD bináris file-név

Hatása a dBASE III PLUS a .bin típusú file gépi kódú programját beolvassa, és a memóriában tartja. A programot CALL paranccsal lehet futtatni. Egyszerre max. 16 különböző modult lehet a memóriában tartani, összesen 32000 byte méretben.

Megjegyzés Fontos! Az assembly nyelven megírt programban az assemblernek ki kell adnunk az ORG 0 utasítást! (Paraméterátadás DS:BX segítségével.)

Példa **load gyorsító**

LOCATE

LOCATE [hatáskör] [WHILE feltétel] [FOR feltétel]

Hatása az adatbázisban szekvenciálisan megkeresi az első olyan rekordot, amely a feltételeknek eleget tesz. (A keresés folytatása CONTINUE-val.)

Példa **locate for kód=34 .and. név='Kovács'**

LOOP

L. DO WHILE

MODIFY COMMAND

MODIFY COMMAND file-név

Hatása aktivizálja a dBASE III PLUS szövegszerkesztőt.

Példa **modi comm bevitel**

MODIFY file-típus

L. CREATE | MODIFY LABEL

L. CREATE | MODIFY QUERY

L. CREATE | MODIFY REPORT

L. CREATE | MODIFY SCREEN

L. CREATE | MODIFY VIEW

MODIFY STRUCTURE

MODIFY STRUCTURE

Hatása a már meglévő adatbázis struktúráját módosítja.

Példa **modify structure**

NOTE | * | &&

NOTE | * [szöveg]

vagy

parancs && szöveg

Hatása az utasításokhoz megjegyzés fűzhető.

Példa **mozgásnem = 99 && ez a hibakód**

ON

ON ERROR | ESCAPE | KEY parancs

Hatása elágazást tesz lehetővé a programban, ha a három eset egyike előfordul.

Megjegyzés ERROR hiba esetén.
ESCAPE az <ESC> billentyű lenyomásakor.
KEY bármely billentyű lenyomásakor.

Példák **on error do hibajavítás**
on escape do kilépés
on key do leállítás

PACK

PACK

Hatása az adatbázisban logikailag törölt rekordokat fizikailag is törli. Az összes nyitott indexfile-t automatikusan újraindexeli.

Példa **pack**

PARAMETERS

PARAMETERS paraméterlista

Hatása a hívott programban kell szerepelnie, ha átadott paraméterekkel dolgozunk. A paraméterlistában szereplő változóknak szám és típus szerint meg kell egyezniük a DO file-név WITH paraméterlista utasításban szereplő paraméterekkel.

Példa

```
* hívott.prg
parameters név,fizetés
.
.
.
* eof hívott.prg
```

PRIVATE

PRIVATE memóriaváltozó-lista |
[ALL[LIKE | EXCEPT (globális karakterek)]]

Hatása a hívott programban úgy hoz létre memóriaváltozókat, hogy azok értéke független a hívó program ugyanolyan nevű változóitól.

Példa `private i,j,k,összeg`

PROCEDURE

PROCEDURE file-név

Hatása az eljárásfile-ban levő összes eljárás így kell, hogy kezdődjék (és RETURN utasítással kell, hogy végződjék).

Példa `procedure lapdobás`
`set print on`
`? chr(12)`
`set print off`
`return`

PUBLIC

PUBLIC memóriaváltozó-lista

Hatása a felsorolt memóriaváltozókat globálissá teszi, vagyis minden program és eljárás hozzáférhet az értékükhöz, és megváltoztathatja őket.

Példa `public sor,oszlop,nap,kulcsszó`

QUIT

QUIT

Hatása minden megnyitott file-t lezár, és kilépve a dBASE III PLUS-ból az operációs rendszernek adja át a vezérlést.

Példa `quit`

READ

READ [SAVE]

Hatása aktivizálja a már előzőleg kiadott @... GET parancsokat. A SAVE hatására a @... GET parancsok nem törölődnek, így újabb READ paranccsal ismét aktivizálhatók.

Példa `read`

RECALL

RECALL [hatáskör] [WHILE feltétel] [FOR feltétel]

Hatása a feltételeknek eleget tevő, logikailag törölt rekordokat visszaállítja nem törölteké.

Példa `recall all for anyagszám>1000`

REINDEX

REINDEX

Hatása újraindexeli az aktuális adatbázist, valamennyi nyitott indexfile szerint.

Példa `reindex`

RELEASE

RELEASE memóriaváltozó-lista |
[**ALL**[**LIKE** | **EXCEPT** (globális karakterek)]] |
[**MODULE** modulnév]

Hatása törli a feltételeknek eleget tevő memóriaváltozókat vagy a LOAD-dal betöltött gépi kódú rutinokat.

Példa `release all like mx*`

RENAME

RENAME régi név **TO** új név

Hatása a lemezen levő file átnevezése.

Példa `rename bevitel.prg to bevitel.old`

REPLACE

REPLACE [hatáskör] mező **WITH** kifejezés
[,mező **WITH** kifejezés ...] [**WHILE** feltétel]
[**FOR** feltétel]

Hatása az adatbázis adott feltételeknek megfelelő rekordjaiban a megadott mezőket a kifejezések értékével tölti fel.

Példa `replace all anyagszám with recno(),kelt with date()
for recno() < 1000`

REPORT

REPORT FORM file-név | ? [hatáskör] [**WHILE** feltétel]
[**FOR** feltétel] [**PLAIN**] [**HEADING** str]
[**NOEJECT**]
[**TO PRINT** | **TO FILE** file-név]
[**SUMMARY**]

Hatása az adatbázisból a **CREATE** | **MODIFY** **REPORT** paranccsal előzőleg már létrehozott .frm file segítségével táblázatot nyomtat.

Megjegyzés **PLAIN** lapszámozás és rendszerdátum nélkül, fejléc csak az első lapon.
HEADING minden lap tetejére kiírja az str-et is.
NOEJECT a kezdeti lapdobást elhagyja.
TO PRINT nyomtatóra küldi a táblázatot.
TO FILE file-ba nyomtat.
SUMMARY a tételsorokat nem, hanem csak a részösszegeket és a végösszeget nyomtatja ki.

Példa **report form** jelentés

RESTORE

RESTORE FROM memóriafile-név [**ADDITIVE**]

Hatása a memóriafile-ban levő memóriaváltozókat beolvassa a dBASE III PLUS memóriájába [úgy, hogy a már meglévő memóriaváltozókat nem törli].

Példa **restore from color additive**

RESUME

RESUME

Hatása a felfüggesztett programot továbbfuttatja (l. SUSPEND).

Példa `resume`

RETRY

RETRY

Hatása általában akkor használjuk, ha valami hiba adódott. A vezérlést a hívó programnak adja vissza, és újra próbálkozik a megszakított parancs végrehajtásával.

Példa

```
* hibakezelő.prg
parameters error
if error = 3
    close databases
endif
retry
return
* eof hibakezelő.prg
```

RETURN

RETURN [TO MASTER]

Hatása a hívó programnak adja vissza a vezérlést.

Megjegyzés TO MASTER a legfelső szintű hívó programhoz kerül a vezérlés.

Példa `return to master`

RUN | !

RUN | ! külső parancs

Hatása végrehajtja a külső .COM vagy .EXE parancsot, és annak befejeződése után a következő utasításra lép.

Példa ! chkdsk

SAVE

SAVE TO file-név [ALL|LIKE | EXCEPT (globális karakterek)]]

Hatása az adott memóriaváltozókat a file-név nevű .mem file-ba menti.

Példa save to color all like szín*

SEEK

SEEK kifejezés

Hatása az indexelt adatbázisban megkeresi azt az első rekordot, amelynek kulcsa megegyezik az adott kifejezéssel.

Példa név = 'Kovács'
seek név

SELECT

SELECT munkaterületszám | adatbázisnév

Hatása a megadott munkaterületen levő vagy a megadott nevű adatbázist aktívvá (kurrenszé) teszi.

Példa `select raktár`

SET

SET

Hatása teljes képernyős, menüvel vezérelt parancs, amellyel beállíthatók a SET ... parancsok értékei.

Példa `set`

SET ALTERNATE

1. **SET ALTERNATE TO** [file-név]
2. **SET ALTERNATE on | OFF**

Hatása

1. minden nem teljes képernyős megjelenítést az adott nevű file-ba irányít. Ha nem adtunk meg file-nevet, lezárja a nyitott ALTERNATE file-t.
2. a file-ba irányítást elindítja vagy leállítja.

Példák `set alternate to jegyzet.txt`
`set alternate on`

SET BELL

SET BELL ON | off

Hatása a figyelmeztető sípolás be-, ill. kikapcsolása.

Példa `set bell off`

SET CARRY

SET CARRY on | **OFF**

Hatása APPEND és INSERT parancsok használatánál az új rekordba másolja az előző rekord adatait.

Példa `set carry on`

SET CATALOG

1. **SET CATALOG TO** [file-név]
2. **SET CATALOG on** | **OFF**

Hatása

1. megnyit vagy létrehoz egy katalógusfile-t. Ha nem adtunk meg file-nevet, lezárja a nyitott katalógusfile-t. A megnyitott vagy létrehozott katalógusfile-ba beírja az újonnan használt file-ok nevét.
2. elindítja vagy leállítja a bejegyzést.

Példák `set catalog főkönyv`
`set catalog on`

SET CENTURY

SET CENTURY on | OFF

Hatása	dátum típusú adatoknál az évszázadot is kiírja.
Példa	<code>set century on</code> <code>? date() (02/28/1987)</code>

SET COLOR

1. SET COLOR on | OFF

2. **SET COLOR TO** [[kiírás] [,inverz] [,keret] [,háttér]]
SET COLOR TO [[kiírás előtér[/háttér]]
[,inverz előtér[/háttér]] [,keret]]

Hatása	1. színes vagy monokrom átkapcsolás. 2. beállítja a megjelenítés színét; paraméte- rek nélkül visszaállítja az alapértelmezés szerinti értékeket.
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------

Megjegyzés	n fekete, b kék, g zöld, bg cián, r piros, rb lila, gr barna, gr+ sárga, w fehér, n+ szürke, x üres, * villogás, + intenzív.
------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------

Példa	<code>set color to gr+/r , w/r , rb/b , gr+</code>
-------	----------------------------------------------------

SET CONFIRM

SET CONFIRM on | OFF

Hatása adatok begépelése után szükséges-e vagy nem szükséges az <ENTER> billentyű lenyomása.

Példa `set confirm on`

SET CONSOLE

SET CONSOLE ON | off

Hatása programon belül az adatok esetleges kettőzött megjelenítését tiltja le.

Példa `set console off`

SET DATE

SET DATE típus

Hatása adott típusú dátummegjelenítést állít be.

Megjegyzés

american (amerikai)	hónap/nap/év,
ansi	év.hónap.nap,
british (brit)	nap/hónap/év,
french (francia)	nap/hónap/év,
german (német)	nap.hónap.év,
italian (olasz)	nap-hónap-év.

Példa `set date ansi`

SET DEBUG

SET DEBUG on | OFF

Hatása programhiba-keresés támogatása; a SET ECHO kimenetét nyomtatóra irányítja át.

Példa `set debug on`

SET DECIMALS

SET DECIMALS TO n

Hatása számolás végeredményének megjelenítése n -tizedesjegy pontossággal.

Példa `set decimals to 4`

SET DEFAULT TO

SET DEFAULT TO meghajtó

Hatása az aktuális meghajtó kijelölése.

Példa `set default to c`

SET DELETED

SET DELETED on | OFF

Hatása a logikailag törölt rekordok nem vesznek részt, ill. részt vesznek a dBASE III PLUS műveleteiben.

Példa `set deleted off`

SET DELIMITERS

1. **SET DELIMITERS TO** [karakterlánc] | **DEFAULT**
2. **SET DELIMITERS on** | **OFF**

Hatása 1. megjelenítéskor milyen jellel határolja, ill.
 2. behatárolja-e egyáltalán a mezőket?

Megjegyzés alapértelmezés = ':',
a karakterlánc max. 2 karakterből állhat.

Példák **set delimiters to '*'**
 set delimiters on

SET DEVICE

SET DEVICE TO SCREEN | print

Hatása a @... SAY parancsok képernyőre vagy nyom-
 tatóra irányítása.

Példa **set device to print**

SET DOHISTORY

SET DOHISTORY on | **OFF**

Hatása a programból végrehajtott utasítások bekerül-
 jenek-e a parancspufferbe vagy sem?

Példa **set dohistory on**

SET ECHO

SET ECHO on | OFF

Hatása a program utasításait végrehajtásukkor kiírja-e a képernyőre vagy sem?

Példa `set echo on`

SET ESCAPE

SET ESCAPE ON | off

Hatása az <ESC> billentyű lenyomásával megszakítható legyen-e a művelet vagy sem?

Példa `set escape off`

SET EXACT

SET EXACT on | OFF

Hatása karakterláncok hasonlításának módját szabja meg. OFF esetén csak a jobb oldali karakterlánc hosszán hasonlít, egyébként a másutt megszokott módon, a hosszabb lánc méretén.

Példa `set exact on`

SET FIELDS

1. **SET FIELDS TO** [mezőlista | **ALL**]

2. **SET FIELDS on** | **OFF**

Hatása 1. beállítja, hogy mely mezőket lehessen elérni a nyitott adatbázisokban, ill.
2. be- és kikapcsolja a korlátozás érvényességét.

Példák **set fields to anyagszám,fkvi_szla,mozgásnem**
set fields on

SET FILTER

SET FILTER TO [**FILE** file-név | ?] [feltétel]

Hatása csak azokat a rekordokat lehet elérni, amelyek az adott feltételnek eleget tesznek, vagy az adott .qry file-ban szerepelnek.

Példa **set filter to file hátralék**

SET FIXED

SET FIXED on | **OFF**

Hatása számok megjelenítésekor azonos számú tizedesjegyet használjon-e?

Példa **set fixed on**

SET FORMAT

SET FORMAT TO [file-név | ?]

Hatása a formátumfile (.fmt) kiválasztása.

Példa **set format to bekér**

SET FUNCTION

SET FUNCTION n TO str

Hatása funkcióbillentyűk beállítása valamely gyakran használatos utasításra.

Megjegyzés $2 \leq n \leq 10$

Példa **set function 3 to disp stru;**

SET HEADING

SET HEADING ON | off

Hatása fejléc kiírásának kérése, ill. letiltása a DISPLAY, LIST, SUM és AVERAGE parancsokhoz.

Példa **set heading off**

SET HELP

SET HELP ON | off

Hatása hibás parancs esetén kérdezze-e meg a dBASE III PLUS, hogy kérünk-e segítséget?

Példa **set help off**

SET HISTORY

1. SET HISTORY TO n
2. SET HISTORY ON | off

Hatása 1. a történetpufferben hány végrehajtott parancsra emlékezzen vissza, ill.
2. tárolja-e a bevitt parancsokat vagy sem?

Megjegyzés $0 \leq n \leq 16\,000$

Példák `set history to 100`
 `set history on`

SET INDEX

SET INDEX TO [file-névlista]

Hatása megnyitja a megadott indexfile-okat.

Példa `set index to anyszám,raktszám,itj`

SET INTENSITY

SET INTENSITY ON | off

Hatása a SET COLOR parancs második operandusának („inverz”) hatását kapcsolja be, ill. ki. OFF esetén az „inverz” attribútumok helyett is a „kiírás” attribútumokat használja.

Példa `set intensity off`

SET MARGIN

SET MARGIN TO *n*

Hatása *n* karakterrel beljebb kezdje a nyomtatást (a képernyőre nincs hatással)!

Példa `set margin to 10`

SET MEMOWIDTH

SET MEMOWIDTH TO *n*

Hatása a memo típusú mezők megjelenítésének szélességét állítja be.

Megjegyzés alapértelmezés = 50

Példa `set memowidth to 70`

SET MENUS

SET MENUS ON | off

Hatása megjelenjen-e a kurzormozgató billentyűk menüje a teljes képernyős parancsok végrehajtása közben, vagy sem?

Példa `set menus off`

SET MESSAGE

SET MESSAGE TO [str]

Hatása ha a SET STATUS ON van érvényben, akkor a legelső sorban az str tartalma íródik ki.

Példa `set message to 'Ne lazálj!'`

SET ORDER

SET ORDER TO [n]

Hatása ha több indexfile-t nyitottunk az aktív adatbázishoz, akkor az *n*-edik indexfile lesz az ún. vezér indexfile.

Példa `use törzs index anyszám,raktszám,ijtj`
`set order to 3`

SET PATH

SET PATH TO [útvonallista]

Hatása nem azonos a DOS PATH parancsával! A dBASE III PLUS külön keresési útvonalakat állít fel az aktuális könyvtárban nem talált file-ok kereséséhez.

Példa `set path to c:\dbase\főkönyv;c:\dbase\bér`

SET PRINT

SET PRINT on | OFF

Hatása a teljes képernyős kiírások (pl. @... SAY) kivételével mindent kiír a nyomtatóra is.

Példa `set print on`

SET PRINTER

SET PRINTER TO [DOS eszköz]

Hatása kijelöli azt a DOS eszközt, amelyet ezután nyomtató helyett használ. Minden nyomtató-kimenet ide kerül.

Példa `set printer to com1`

SET PROCEDURE

SET PROCEDURE TO [file-név]

Hatása lezárja, ill. [megnyitja] a megadott eljárásfile-t, így a benne levő eljárások végrehajthatók lesznek.

Példa `set procedure to rutinok`

SET RELATION

SET RELATION TO

[kulcskifejezés | *n* INTO adatbázis-megnevezés]

Hatása összekapcsol két adatbázist egy kulcs vagy egy rekordsorszám szerint.

Példa

```
select 1
use törzs index t_anyag
select 2
use raktár
set relation to anyagszám into törzs
.
.
set relation to recno() into törzs
```


SET SAFETY

SET SAFETY ON | off

Hatása állományok fölüírásához kér engedélyt.

Példa `set safety off`

SET SCOREBOARD

SET SCOREBOARD ON | off

Hatása megszabja, hogy a dBASE III PLUS bizonyos üzenetei megjelenjenek-e a képernyőn (STATUS ON esetén a 22., OFF esetén a 0. sorban).

Példa `set scoreboard off`

SET STATUS

SET STATUS ON | off

Hatása a státussor megjelenjen-e, vagy sem?

Példa `set status off`

SET STEP

SET STEP on | OFF

Hatása a program soronkénti végrehajtását teszi lehetővé.

Példa `set step on`

SET TALK

SET TALK ON | off

Hatása a dBASE III PLUS válaszoljon-e a parancsokra?

Példa **set talk off**

SET TITLE

SET TITLE ON | off

Hatása be-, ill. kikapcsolja a katalógusfile címbekérését.

Példa **set title off**

SET TYPEAHEAD

SET TYPEAHEAD TO n

Hatása hány karaktert lehessen előre begépelni?

Megjegyzés $0 \leq n \leq 32\ 000$,
alapértelmezés = 20

Példa **set typeahead to 50.**

SET UNIQUE

SET UNIQUE on | OFF

Hatása több, azonos kulccsal rendelkező rekord esetén csak az elsőt vegye fel az indexfile-okba.

Példa `set unique on`

SET VIEW

SET VIEW TO file-név | ?

Hatása megnyitja a megadott .vue file-t és az abban megadott adatbázisokat, valamint beállítja a SET RELATION relációkat.

Példa `set view to kezdés`

SKIP

SKIP [n]

Hatása az adatbázis rekordpointerét (rekordmutatóját) elmozdítja n -nel.

Példa `skip -1`

SORT

SORT hatáskör **TO** új file-név **ON** mező [/A] [/C] [/D]
[,mező2 [/A][/C][/D]...] [**WHILE** feltétel]
[**FOR** feltétel]

Hatása olyan új adatbázist hoz létre, amelyben a rekordok az adott mezők sorrendjében rendezettek.

Megjegyzés /A növekvő sorrendben.
/C nem tesz különbséget kis- és nagybetű között.
/D csökkenő sorrendben.

Példa `sort all to rendes on anyagszám /ac`

STORE

STORE kifejezés **TO** memóriaváltozó-lista

Hatása a memóriaváltozóknak ugyanazt az értéket adja.

Példa `store 0 to n1,n2,n3,totgyűjtő1,totgyűjtő2`

SUM

SUM [hatáskör] [kifejezéslista]
[**TO** memóriaváltozó-lista] [**WHILE** feltétel]
[**FOR** feltétel]

Hatása a feltételeket teljesítő adatbázisrekordokból összegzi a kifejezéseket.

Példa `sum 2*mennyiség,4*érték+20 to m_menny,m_érték
for kelt<ctod('86.12.31')`

SUSPEND

SUSPEND

Hatása az éppen futó programot felfüggeszti, majd a RESUME parancs kiadása után folytatja a félbeszakított tevékenységet.

Példa `suspend`

TEXT

TEXT

szöveg

ENDTEXT

Hatása a szöveg megjelenítése úgy, ahogy a két kulcsszó közé beírtuk.

Példa `text`
`---- most fejeztem be az összegzést ----`
`endtext`

TOTAL

TOTAL ON kulcsmező **TO** adatbázisnév [hatáskör]
[**FIELDS** mezőlista] [**WHILE** feltétel]
[**FOR** feltétel]

Hatása az aktuális adatbázis azonos kulcsú numerikus mezőit összegzi, és ezeket az összegeket beírja a második adatbázisba.

Megjegyzés Fontos! Az összegzendő adatbázisnak a kulcsmező szerint indexeltnek vagy rendezettnek kell lennie!

Példa `use költség index személy`
`total on személy to sz_költség`

TYPE

TYPE filenév [TO PRINT]

Hatása azonos a DOS TYPE parancsával. Az adott file tartalmát kiírja [a nyomtatóra].

Példa `type lista.prg`

UPDATE

UPDATE ON kulcsmező **FROM** adatbázisnév
REPLACE mező **WITH** kifejezés
[,mező2 **WITH** kifejezés2 ...] [**RANDOM**]

Hatása az adatbázis rekordját módosítja a FROM adatbázis azonos kulcsú rekordjának adataival. (Több azonos kulcsú közül csak az elsőt.) A kulcsmezőnek mindkét adatbázisban szerepelnie kell, és mindkét adatbázisnak ezen mező szerint rendezettnek vagy indexeltnek kell lennie.

Ha a REPLACE kifejezésében a FROM adatbázisból kell venni az értéket, akkor az adatbázisnév →mező jelölést kell használni.

Megjegyzés **RANDOM** akkor kell megadni, ha a FROM adatbázis nem rendezett.

A módosítandó adatbázisnak ekkor is indexeltnek kell lennie!

Példa

```
select 2
use munkások
select 1
use létszám index törzsszám
update on törzsszám from munkások replace felvehető with felvehető-1 random
```


USE

USE [file-név | ?] [**INDEX** indexfile-lista]
[**ALIAS** adatbázisnév]

Hatása megnyit egy létező adatbázist a hozzá tartozó max. 7 indexfile-lal együtt. Ezek közül sorrendben az első a vezérindex. Amennyiben az adatbázisnak van memo típusú mezője, a hozzá tartozó .dbt file-okat is megnyitja.

Megjegyzés **ALIAS** az eredeti file-név helyett ezen a néven hivatkozhatunk az adatbázisra.

A **USE** parancs paraméter nélkül lezárja az aktuális munkaterületen levő adatbázist.

Példa `use raktár index r_anyag,r_itj`

WAIT

WAIT [szöveg] [**TO** memóriaváltozó]

Hatása a program csak egy billentyűleütés után folytatódik.

Megjegyzés **szöveg** a várakozás előtt kiírandó szöveg.

TO memóriaváltozó a lenyomott billentyű értékét tárolja.

Példa `wait 'Folytassam?' to tovább
if upper (tovább) <> 'I'.or.'Y'
quit
endif`

ZAP

ZAP

Hatása fizikailag törli az adatbázis összes rekordját, és a megnyitott indexfile-okat újraindexeli.

Példa `zap`

A dBASE III PLUS függvények funkció szerinti csoportosítása

Adatbázisfile-okra vonatkozó függvények

BOF ()	70
DBF ()	72
DELETED ()	72
EOF ()	73
FIELD (n)	74
FOUND ()	75
LUPDATE ()	79
NDX (n)	80
RECCOUNT ()	82
RECNO ()	82
RECSIZE ()	82

Dátum típusú kifejezésekre vonatkozó függvények

CDOW (dátum)	70
CMONTH (dátum)	71
CTOD (str)	71
DATE ()	71
DAY (dátum)	72
DOW (dátum)	73
DTOC (dátum)	73
MONTH (dátum)	80
YEAR (dátum)	87

Rendszerkörnyezetre vonatkozó függvények

COL ()	71
DISKSPACE ()	72
ERROR ()	73
FILE (str)	74
FKLABEL (n)	74
FKMAX ()	75

GETENV (str)	75
INKEY ()	76
ISCOLOR ()	77
MESSAGE ()	79
OS ()	80
PCOL ()	81
PROW ()	81
READKEY ()	81
ROW ()	83
TIME ()	85
TYPE (str)	86
VERSION ()	87

Numerikus típusú kifejezésekre vonatkozó függvények

ABS (n)	69
EXP (n)	74
IIF (log,kifejezés1,kifejezés2)	76
INT (n)	76
LOG (n)	78
MAX (n1,n2)	79
MIN (n1,n2)	79
MOD (n1,n2)	80
ROUND (n1,n2)	83
SQRT (n)	84
STR (n1,n2[,n3])	84
TRANSFORM (n str1,str2)	85
VAL (str)	86

Karakter típusú kifejezésekre vonatkozó függvények

&	69
ASC (str)	69
AT (str1,str2)	70
CHR (n)	70
ISALPHA (str)	76
ISLOWER (str)	77
ISUPPER (str)	77
LEFT (str,n)	77
LEN (str)	78

LOWER (str)	78
LTRIM (str)	78
REPLICATE (str, n)	82
RIGHT (str, n)	83
RTRIM (str)	83
SPACE (n)	84
STUFF (str1, n1, n2, str2)	84
SUBSTR (str, n1 [, n2])	85
TRIM (str)	85
UPPER (str)	86

dBASE III PLUS függvények

&

&str

Hatása az str memóriaváltozóban levő karaktersorozatot parancsként értelmezi, és végre is hajtja (makró).

Példa str='disp memo'
&str

ABS

ABS (n)

Értéke n abszolút értéke.

Példa $n = -5$
? abs (n) → 5

ASC

ASC (str)

Értéke az str karakterlánc első karakterének ASCII értéke.

Példa ? asc ('Munkakör:') → 77

AT

AT (str1, str2)

Értéke az str1 karakterláncnak az str2 karakterláncban elfoglalt pozíciója.
Ha az str1 nem szerepel részláncként az str2-ben, akkor az érték 0.

Példa ? at ('d', 'abcdefgh') → 4

BOF

BOF ()

Értéke igaz, ha az aktuális rekord mutatója a logikailag első rekord előtt áll.

Példa ? bof () → pl. .F.

CDOW

CDOW (d)

Értéke az adott d dátum napjának angol neve.

Példa ? cdown (date()) → pl. Friday

CHR

CHR (n)

Értéke az n ASCII karaktere.

Példa ? chr (65) → A

CMONTH

CMONTH (d)

Értéke az adott d dátum hónapjának angol neve.
Példa ? cmonth (date()) → pl. February

COL

COL ()

Értéke a kurzor aktuális pozíciójának oszlopsorszáma.
Példa ? col () → pl. 45

CTOD

CTOD (str)

Hatása a karakteres formában megadott dátumot dátum típusúra alakítja át.
Példa d = ctod ('02/27/87')

DATE

DATE ()

Értéke a számítógép rendszerdátuma a beállított dátumformátumnak megfelelően.
Példa ? date () → 02/27/87

DAY

DAY (d)

Értéke megadja, hogy a d dátum a hónap hányadik napja.

Példa ? day (date()) → pl. 27

DBF

DBF ()

Értéke az aktuális (kurrens) adatbázis neve.

Példa ? dbf () → pl. c:test.dbf

DELETED

DELETED ()

Értéke igaz, ha az adatbázis aktuális rekordja logikailag törölve van.

Példa ? deleted () → pl. .T.

DISKSPACE

DISKSPACE ()

Értéke az aktuális meghajtó lemezén levő szabad terület nagysága byte-ban kifejezve.

Példa ? diskpace () → pl. 3086336

DOW

DOW (d)

Értéke az adott d dátum hányadik nap az adott héten belül (vasárnap = 1).

Példa ? **dow(date())** → pl. 6

DTOC

DTOC (d)

Hatása az adott d dátumot karaktersorozattá alakítja át a beállított dátumformátum szerint.

Példa **mai_dátum = dtoc (date())**

EOF

EOF ()

Értéke igaz, ha az aktuális rekord mutatója a logikailag utolsó rekord mögött áll (file vége).

Példa ? **eof ()** → pl. .T.

ERROR

ERROR ()

Értéke az ON ERROR feltételt beállító hiba kódja.

Példa ? **error ()** → pl. 16

EXP

EXP (n)

Hatása természetes alapú hatványozás.

Példa $a = \text{exp}(1)$ → pl. $a = 2,71828183$

FIELD

FIELD (n)

Értéke az aktuális adatbázis n -edik mezőjének neve.

Példa ? **field(1)** → pl. kódszám

FILE

FILE (str)

Értéke igaz, ha az aktuális könyvtárban létezik az str változóban megadott nevű file.

Példa ? **file('test.dbf')** → pl. .T.

FKLABEL

FKLABEL (n)

Értéke az n -edik funkcióbillentyű neve.

Példa ? **fklabel(4)** → F5

FKMAX

FKMAX ()

Értéke a programozható funkcióbillentyűk száma az adott számítógépen.

Példa ? `fkmax ()` → 9

FOUND

FOUND ()

Értéke igaz, ha az előzőleg kiadott keresési parancs (FIND, SEEK, LOCATE, CONTINUE) sikeres volt.

Példa `str = 'Kovács'`
`seek str`
? `found ()` → pl. .T.

GETENV

GETENV (str)

Értéke a rendszer környezetleíró táblájából kiveszi az `str` karakterlánccal kért információt.

Példa `str = 'path'`
? `getenv (str)` → pl. `c:\dos;c:\dbase`

IIF

IIF (log,kifejezés1,kifejezés2)

Értéke a log logikai kifejezés értékétől függően a kifejezés1 (ha a log logikailag igaz), ill. a kifejezés2.

Megjegyzés a két kifejezés azonos típusú kell, hogy legyen!

Példa ? iif (.t.,'aaa','bbb') → aaa

INKEY

INKEY ()

Értéke az utoljára lenyomott billentyű ASCII kódja.

Példa i = inkey () → i = 65

INT

INT (n)

Értéke n egészrésze.

Példa ? int(3.14) → 3

ISALPHA

ISALPHA (str)

Értéke igaz, ha az str karakterlánc első karaktere betű.

Példa ? isalpha ('auto') → .T.

ISCOLOR

ISCOLOR ()

Értéke igaz, ha a számítógépnek van színes adaptere és a SET COLOR ON érvényes.

Példa ? iscolor () → pl. .T.

ISLOWER

ISLOWER (str)

Értéke igaz, ha az str karakterlánc első karaktere kisbetű.

Példa ? islower ('Kovács') → .F.

ISUPPER

ISUPPER (str)

Értéke igaz, ha az str karakterlánc első karaktere nagybetű.

Példa ? isupper ('Kovács') → .T.

LEFT

LEFT (str,n)

Értéke az str karakterlánc első n karaktere.

Példa ? left ('abcdefghijkl',5) → abcde

LEN

LEN (str)

Értéke az str karakterlánc hossza.

Példa ? len ('abcdefghij') → 10

LOG

LOG (n)

Értéke n természetes alapú logaritmusa.

Példa ? log (2.72) → 1.00

LOWER

LOWER (str)

Hatása az str karakterláncban levő nagybetűket kisbetűkké alakítja át.

Példa ? lower ('123abcABC') → 123abcabc

LTRIM

LTRIM (str)

Hatása elhagyja az str karakterlánc vezető szóközeit.

Példa ? ltrim (' abcd') → abcd

LUPDATE

LUPDATE ()

Értéke az aktuális adatbázis utolsó módosításának dátuma.

Példa ? lupdate() → pl. 02/27/87

MAX

MAX (n1,n2)

Értéke *n1* és *n2* közül a nagyobb szám.

Példa ? max (23,111) → 111

MESSAGE

MESSAGE ()

Értéke a legutolsó észlelt hiba angol nyelvű leírása.

Példa ? message() → pl. File does not exist.

MIN

MIN (n1,n2)

Értéke *n1* és *n2* közül a kisebb szám.

Példa ? min (23,111) → 23

MOD

MOD (n1,n2)

Értéke $n1$ modulo $n2$ ($n1$ -nek $n2$ -vel való osztásakor keletkező maradék).

Példa ? mod (20,3) → 2

MONTH

MONTH (d)

Értéke az adott d dátum hónapjának sorszáma.

Példa ? month (date()) → pl. 2

NDX

NDX (n)

Értéke az aktuális adatbázis n -edik indexfile-jának neve.

Megjegyzés $1 \leq n \leq 7$.

Példa ? ndx (3) → pl. c:vétel.ndx

OS

OS ()

Értéke az operációs rendszer verziószáma.

Példa ? os() → pl. DOS 3.20

PCOL

PCOL ()

Értéke a nyomtatófej aktuális pozíciójának oszlopsorszám.

Példa

```
if pcol () > 75
    ? chr (13)
endif
```

PROW

PROW ()

Értéke az utolsó lapdobás óta kinyomtatott sorok száma.

Példa

```
if prow () > 60
    ? chr (12)
    akt_sor = 1
endif
```

READKEY

READKEY ()

Értéke a teljes képernyős parancsot befejező billentyű kódja. Jelzi azt is, hogy módosult-e valamely adat.

Példa

```
? readkey ()
```

 → pl. 271

RECCOUNT

RECCOUNT ()

Értéke az aktuális adatbázis rekordjainak száma.

Példa ? `reccount ()` → pl. 934

RECNO

RECNO ()

Értéke az aktuális rekord sorszáma.

Példa ? `recno ()` → pl. 3

RECSIZE

RECSIZE ()

Értéke a rekordméret byte-ban kifejezve.

Példa ? `recsize ()` → pl. 212

REPLICATE

REPLICATE (str,n)

Értéke az str karakterlánc *n*-szer összefűzve.

Példa ? `replicate ('a',8)` → a-a-a-a-a-a-a-a

RIGHT

RIGHT (str, n)

Értéke az str karakterlánc utolsó n karaktere.

Példa ? right ('abcdefghijkl',4) → ijkl

ROUND

ROUND (n1, n2)

Értéke $n1$, amit $n2$ tizedesre kerekítünk.

Példa ? round (3.14159,2) → 3.14000

ROW

ROW ()

Értéke a kurzor aktuális pozíciójának sorszáma.

Példa ? row () → pl. 18

RTRIM

RTRIM (str)

Hatása az str karakterlánc végéről elhagyja a szókö-
zöket.

Példa str2=rtrim ('abcd ') → str2='abcd'

SPACE

SPACE (n)

Értéke n számú szóköz.

Példa ? space (20) → 20 szóközt ír ki

SQRT

SQRT (n)

Értéke n négyzetgyöke.

Példa ? sqrt (81) → 9

STR

STR (n1,n2[,n3])

Hatása az $n1$ számot $n2$ hosszúságú [$n3$ tizedesjeggyel] karakterlánccá alakítja át.

Példa számsor = str (3.14159,10,2)
→ számsor = ' 3.14'

STUFF

STUFF (str1,n1,n2,str2)

Hatása az str1 karakterlánccba az str2-t az $n1$ -edik pozícióval kezdődően úgy szúrja be, hogy az str1-ből $n2$ karaktert fölülír.

Példa a=stuff ('abcdefghijkl',4,1,'XYZ')
→ a='abcXYZefghijkl'

SUBSTR

SUBSTR (str, n1 [,n2])

Hatása az str karakterláncból *n1*-edik pozícióval kezdődő [*n2* hosszúságú] részláncot állít elő.

Példa ? substr ('abcdefghijkl',4,5) → 'defgh'

TIME

TIME ()

Értéke a számítógép rendszerideje (karakteresen).

Példa ? time() → pl. 18:32:23

TRANSFORM

TRANSFORM (n | str1, str2)

Hatása a @ ... SAY utasítás PICTURE formátuma szerinti formátumkialakítás.

Példa a=transform(1234.56,'###,###.##')
→ a=' 1,234.56'

TRIM

TRIM (str)

Hatása az str karakterlánc végéről elhagyja a szóközőket.

Példa str2=trim('abcd ') → str2='abcd'

TYPE

TYPE (str)

Értéke az str karakterlánc típusa.

Megjegyzés N numerikus,
C karakter,
L logikai,
M memo,
U nem létező változó, vagy érvénytelen kifejezés.

Példák **anyagszám = 34**
mező = 'anyagszám'
TYPE ('mező') → C
TYPE (mező) → N !!!!!
TYPE ('anyagszám/100') → N
TYPE ('mező/100') → U

UPPER

UPPER (str)

Hatása az str-ben levő kisbetűket nagybetűkké alakítja át.

Példa ? upper ('123abcABC') → 123ABCABC

VAL

VAL (str)

Hatása karaktersorozatot számmá alakít át (ha lehetséges).

Példa n = val ('3.14215')

VERSION

VERSION ()

Értéke a dBASE III PLUS verziószáma.

Példa ? version () → dBASE III PLUS Version 1.1

YEAR

YEAR (d)

Értéke az adott d dátum teljes évszáma (évszázad is).

Példa ? year (date()) → pl. 1987

Irodalom

dBASE III PLUS Manual (1.10)

Kiadja a Műszaki Könyvkiadó
Felelős kiadó: Szűcs Péter igazgató
Felelős szerkesztő: Györke Tiborné

A szedés a Műszaki Könyvkiadóban készült

AGROÉPSZER

Műszaki vezető: Kőrizs Károly
Műszaki szerkesztő: Molnár Zoltán
A borítót tervezte: Székely Edith
A könyv formátuma: AK/40
Ívterjedelme: 4,35 (A5)
Azonossági szám: 80035
MŰ: 4130—i—8890
Készült az MSZ 5601 és 5602 szerint
A kézirat lezárva: 1987. október

150 Ft

MŰSZAKI KÖNYVKIADÓ

NOVOTRADE RT