

347.057
tanulmányok 92/1979

92
/
79

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

KISGÉPES ADATBÁZIS KEZELŐ RENDSZER

Készítette:

ANDOR LASZLÓ

Tanulmányok 92/1979.

A kiadásért felelős:
DR VAMOS TIBOR

ISBN 963 311 079 3
ISSN 0324-2951

Készült a
KSN Nemzetközi Számítástechnikai Oktató és Tájékoztató Központ
Reprográfiai Üzemében 79/161

MAGYAR
KÖZMŰVELÉSI AKADEMIA
KÖNYVTÁRA

TARTALOMJEGYZÉK

| DML | Oldal |
|--|-------|
| Bevezetés | 7 |
| 1. Az adatbázis leírásához használt fogalmak | 11 |
| Hibajelzés | 19 |
| 2. Az adatkezelő rutinok leírása | 21 |
| | |
| DDL | |
| Bevezetés | 75 |
| Utasítások | 78 |
| | |
| Irodalomjegyzék | 91 |
| 1. Melléklet: A DDL-ben használt speciális jelentésű karakterek | 92 |
| 2. Melléklet: A DDL-ben kiadott hibakódok | 93 |
| 3. Melléklet: Ábrás szintaxisleírás | 95 |

D M L

DATA MANIPULATION LANGUAGE

ADAT KEZELŐ RUTINOK

Bevezetés

Az adatbázis a hétköznapi értelmezés és a különböző formális definíciók szerint egyaránt olyan strukturált rendezett adatgyűjteménynek tekinthető, amely egy fizikailag létező rendszer, szervezet vagy szakterület ismereteit, állapotát tükrözi. Elvileg a struktúra is folytonosan változhat /bővíthet, módosulhat/, de mivel a szervezet és az őt tükröző adatbázis is a struktúra ismeretében "kezelhető", a struktúrát hosszabb időre stabilnak tekinthetjük.

A valós világbeli rendszereket az adatbázis szempontjából úgy szemlélhetjük, hogy abban különböző típusokba sorolható egyedek /szerszámok, alkatrészek, emberek, stb./ és az egyedek között kapcsolatok léteznek. Az egyedek számos tulajdonsággal rendelkeznek /a szerszám geometriai jellemzői, megmunkálás számításhoz, beállításhoz felhasználható adatai/, amelyek a rendszerben végbemenő folyamatokhoz, eseményekhez más-más időben szükségesek. Lényegileg tehát egy adatbázis:

- egyedekre,
- egyedek tulajdonságaira,
- egyedek közötti kapcsolatokra

vonatkozó struktúra-leíró információt tartalmaz, valamint a struktúra információval összhangban az egyed-típusok konkrét egyedeire azok "adatait" tárolja.

A számítógépes adatbázis fenntartás, valamint az adatbázis tartalmát az adatkezelő funkciókon keresztül igénybevevő felhasználói programok vagy terminál-felhasználók az adatbázist a struktúra információk és az egyed-, valamint kapcsolattípusok előfordulásait megtestesítő adatállományok /logikai file-ok, indexek, listák, táblázatok, stb./ együtteseként fogja fel. Az adatbázis kezelő szolgáltatások kisebb része a struktúra leíró táblákra, nagyobb része az adatállományra vonatkozik. Az egyes adattípusok és konkrét előfordulásaik elérése a struktúra leíró táblák, indexek, listák segítségével hívásával történik. Ezeknek a "segéd" információknak a fenntartása, kezelése általában rejtve marad az adatkezelő funkciókat igénybe vevő felhasználó vagy felhasználói programokat író programozó elől, mivel számukra nem szükséges az adatbázis belső, tárolási strukturájának ismerete. Elegendő az adatbázis u.n. logikai strukturájának ismerete, ami az egyedek főbb tulajdonságait, a kapcsolatok fajtáit és ezen jellemzőket hordozó adatok mezejét jelenti. A felhasználó az egyes funkciók végrehajtásának kérésekor megadja ezen jellemzők konkrét értékeit és a kért funkció által a fizikai strukturán automatikusan végbemegegy minden változtatás, behívás, kiolvasás, stb.

Az IGYR-630 tervezői konfigurációjára kialakított és a következőkben leírásra kerülő adatbázis kezelő rendszer az előbbieken értelmezett adatbázisok fenntartására, a rendszerben végbemenő folyamatok, történések adattal való ellátására készült. A teljes adatbázis rendszer blokkvázlatát az 1. ábra mutatja. A jelenlegi leírás azokat az alapvető rendszerelemeket, logikai és a fizikai tárolási strukturákat ismerteti, amelyek szükségesek a többé-kevésbé önálló rendszerrészek kialakításához /struktúra generáló program, alkalmazói programok, aktualizáló program, stb./.

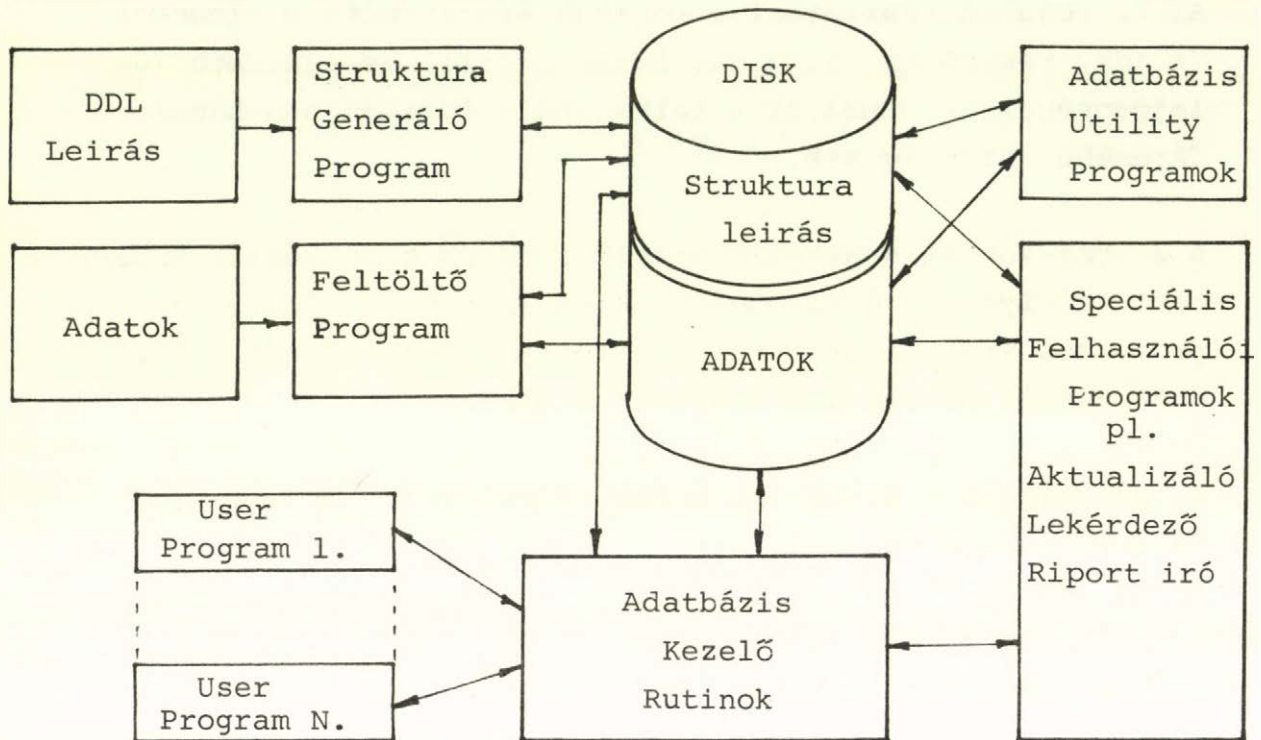
Az 1. fejezet részletezi a logikai adatstruktúra elemeket /mezők, rekordok, halmazok, logikai file/ és alapvető tulajdonságaikat, amelyek a felhasználó logikai adathozzáférésehez szükségesek.

A 2. fejezet az adatbázis kezelő rutinok azon részét ismerteti, amelyek az általánosított és az egyedi alkalmazói programok írói számára rendelkezésre állnak az adatelőfordulásokbeli állományok manipulálásához.

Az adatkezelő rutinok leírásából következik, hogy bázisnyelvre alapozott adatkezelő funkciók állnak rendelkezésre az adatbázis fenntartásához és alkalmazói programokból történő eléréséhez. A bázisnyelv a DOST operációs rendszerben lévő TAL assembler nyelv, vagy egy magasszintű, a TPA 70 számítógépen rendelkezésre álló, rendszerprogramozók és alkalmazói programozók által egyaránt használható nyelv lehet. Az adatkezelő rutinok szintaxisát és paramétereinek jelentését ebben a nyelvben fejezzük ki. Mivel ennek a nyelvnek a fordítóprogramja TAL nyelvre fordít, automatikusan adódik a rutinok TAL-beli szintaxisa is.

A következő leírásokban az adatbázis fizikai strukturájával /rekord előfordulások, egyszeres és többszörös mezők tárolása, indexek és táblázatok strukturája, stb./ csak olyan mértékig foglalkozunk, amennyire a szubrutinok működésének, funkciójának megértéséhez szükségesek.

Hangsúlyozni szeretnénk, hogy a jelen leírás bizonyos programozói gyakorlattal már rendelkező és az adatbázis kezelés kérdéseiben jártas felhasználók számára készült.



1. fejezet

AZ ADATBÁZIS LEÍRÁSÁHOZ HASZNÁLT

FOGALMAK

MEZŐ

A legkisebb címezhető adategység. A következő jellemzői vannak:

- Név
- Típus /jelenleg a következő típusok vannak implementálva: /

| Tipuskód | Név | Méret /byte/ |
|----------|----------|--------------|
| 1 | CHAR | 1 |
| 2 | INTEGER | 2 |
| 3 | LONGINT | 4 |
| 4 | REAL | 4 |
| 5 | LONGREAL | 8 |
| 6 | STRING | * |

/*: Tetszőleges méretű lehet, a DDL-ben van definiálva./

- Ellenőrzés

Minden mezőre előírható ellenőrzés. Jelenleg a következő ellenőrzési kódok vannak implementálva:

| Ellenőrzési kód | Jelentése |
|-----------------|------------------|
| ∅ | Nincs ellenőrzés |
| 1 | X < MAX |
| 2 | X > MIN |

| Ellenőrzési kód | Jelentése | /folyt./ |
|-----------------|-------------------------------------|----------|
| 3 | $\text{MIN} < X < \text{MAX}$ | |
| 4 | $X \leq \text{MAX}$ | |
| 5 | $X \geq \text{MIN}$ | |
| 6 | $\text{MIN} \leq X \leq \text{MAX}$ | |
| 7 | $\text{MIN} \leq X < \text{MAX}$ | |
| 8 | $\text{MIN} < X \leq \text{MAX}$ | |

/Longreal típusu mezőre csak szimpla pontosságú korlát írható elő!/

- Hányszoros a mező

Ez a jellemző azt mutatja meg, hogy maximum hányszor fordulhat elő egy rekordban.

Ha többszörös mező szerepel egy rekordban, úgy az változó méretű rekord lesz.

A többszörös mezőknek a rekord "végén" kell elhelyezkedniük az egyszeres mezők után.

REKORD

Értelmezése azonos a megszokottal. Valamely dologra, fogalomra, illetve logikai egységre vonatkozó adatok csoportja. A rekord a tárolt információ logikai alapegysége. Tetszőleges számú mezőből állhat. Ha a rekordtípus tartalmaz u.n. többszörös mezőt is, úgy a rekord egyes előfordulásaiiban ez a mező különböző számúszor szerepelhet.

Jellemzői:

- Név
 - * Azonosító mező
- Elérési mód

Lehetséges elérési módok:

SQ /még nincs implementálva/
 DIRECT /+ VIA/
 CALC /+ VIA/
 FUZZY /+ VIA/

* Azonosító mező:

Az egyes rekordelőfordulások azonosítására szolgáló mező. Ez a mező az input paramétere a "CIM"-számító rutinnak, ha ez a rekord CALC vagy DIRECT elérésü.

Az elérési módok befolyásolják a tárolási módot. Az utóbbi három elérési módnál az egyes rekordok két részben vannak tárolva. Az egyik részben a strukturális információk /rekord előfordulás tábla/, a másikban pedig az adatok foglalnak helyet. Az ilyen elérésű rekordok lehetnek u.n. halmazon keresztüli elérésűek is. Ez azt jelenti, hogy szerepelhetnek valamely halmazban tulajdonosként vagy tagként.

DIRECT elérés esetén az azonosító mező "indexként" értelmeződik, nem lehet több azonos indexű rekordelőfordulás.

CALC elérés esetén az azonosító mező a HASH-rutin bemenő paramétere. Itt természetesen előfordulhat, hogy egy

cimre több rekord is kerülhet, ilyenkor a legközelebbi üres helyre kerül majd a rekord.

LIST elérés esetén a rendszer keres az új rekordnak üres helyet. Ilyen elérés esetén legalább egy rendezési kritériumot elő kell írni.

RENDEZÉSI KRITÉRIUM

A rendezési kritérium az azonos típusú rekordelőfordulások valamilyen rend szerinti összefűzését jelenti. Ha egy rekordtípusra elő van írva rendezési kritérium, akkor a rekordelőfordulás táblában létesül egy kétutas lista, mely a rekordok kritérium szerinti sorrendjét reprezentálja. Bizonyos manipulációs rutinok segítségével lehet ezeken a láncokon végigmenni.

Jellemzők:

- Név
- Típusok
 1. Új tag beillesztése legutolsónak
 2. Új tag beillesztése legelsőnek
 3. Új tag beillesztése a kurrens mögé
 4. Új tag beillesztése a kurrens elé
 5. Rendezési kulcs szerint:

Ez azt jelenti, hogy a rekord valamely mezője /rendezési kulcs/ vagy mezői szerint növekvő vagy csökkenő sorrendbe vannak rendezve a rekordok. Lehet egy rendezési kritériumhoz több rendezési kulcsot is előírni. Ilyen esetben a

kulcsok hierarchikusak. /Ha az első kulcs egyenlő, akkor a második szerint, ha az is egyenlő, akkor a harmadik szerint, stb. lesz a sorrend eldöntve./ Rendezési kulcs csak egyszeres mező lehet.

Egy rekordtípusra tetszőleges számú rendezési kritérium írható elő. A rendezés a rekordok létesítésekor, automatikusan történik.

HALMAZ (SET)

A logikai egységek /rekordok/ közötti összefüggések ábrázolásának eszköze. Elemei rekordelőfordulások. Kétféle eleme van minden halmaznak:

1. OWNER /vagy tulajdonos vagy fej/
Tulajdonos rekord csak egy és csakis egy van minden halmazelőfordulásban. A halmazelőfordulás azonosítása /megfogása/ ezzel a rekorddal történik.
2. MEMBER /vagy tag vagy elem/
Egy halmazelőfordulásban több tag lehet, de lehet, hogy nincs egy sem.

Halmazok megvalósítási módjai:

Egyutas
 Kétutas
 Pointertömb *

/*:

A pointertömbös megvalósítás még nincs implementálva./

Az egyutas, ill. kétutas halmazmegvalósítási mód azt jelenti, hogy az egyes tag-rekordok egy vagy mindkét irányban össze vannak pointerezve /láncolva/. A tulajdonos rekordban pedig el van helyezve egy pointer az első és az utolsó tagrekordra. Előírható, hogy minden tagrekord a halmaz tulajdonos rekordjára mutató pointert is tartalmazzon ./Tulajdonoshoz kötés./

Halmazokra is előírható a rendezés, ha az egy- vagy kétutas listaként van megvalósítva. Ez az összeláncolás sorrendjét jelenti.

Egy-egy halmaztípusra előírható jellemzők:

- Név
- Megvalósítás módja
- Tulajdonos rekord típusok

Minden halmaztípushoz meg kell határozni, hogy milyen típusu rekordok lehetnek benne tulajdonosok. Értelemszerűen egy halmazelőfordulásban csak egy, de a különböző azonos típusu /nevű/ halmazelőfordulásokban különböző típusu rekordok lehetnek az ownerek.

- Tag rekord típusok

Meg kell határozni, hogy milyen típusu rekordok lehetnek tagok e halmaztípusban.

Minden tagtípusra meg kell határozni, hogy felvételük egy ilyen típusu halmazba automatikus vagy manuális. Ha egy tagot automatikusan kell felfűzni egy halmaztípusba, ez azt jelenti, hogy a rekord létrehozásakor automatikusan az ilyen típusu halmaz kurrens előfordulásához hozzáadódik.

- Rendezés típusa

Halmazokra is előírható rendezés, ha az listaként megvalósított. Ez az összeláncolás sorrendjét jelenti:

1. Új tag befűzése legutolsónak
2. Új tag befűzése legelsőnek
3. Új tag befűzése kurrens mögé
4. Új tag befűzése kurrens elé
5. Új tag befűzése kulcs(ok) szerint

KURRENS MUTATÓK

Egy rekord közvetlen eléréséhez 4 Byte méretű, a rekordra mutató pointerre van szükség. Ezt a pointert a továbbiakban DBK-nak /Data Base Key/ nevezem.

Kurrens Rekord:

Minden rekordtípusra őriz a rendszer egy pointert /DBK/, amely az utoljára referál rekordra mutat.

Kurrens Owner és Member:

A rendszer őriz minden halmaztípusra két pointert, melyek az utoljára referált halmaz Owner, ill. Member rekordjára mutatnak.

A kurrens mutatókat a különböző manipulációs rutinok a leírásukban adott módon állítják be, ill. használják fel az adatbázisban található strukturákon való mozgásra vagy módosításra, vagy a rekordok elérésére.

HIBAJELZÉS

Hibajelzésre általában a rutinok visszatérő paramétere, valamint egy SZINT nevű globális változó szolgál.

Minden manipulációs rutin leírásában megtalálható, hogy az milyen hibakódokat adhat vissza. E kódok felhasználhatók a user-program vezérlésére, vagy az ERROR nevű rendszer-rutin ezen kóddal történő meghívása esetén a megfelelő hibaüzenet kiiratására.

Ez a rutin a SZINT változó értékétől függően vagy visszaadja a vezérlést /SZINT= \emptyset /, vagy terminálja a futást. E rutint a rendszer maga is meghívja, ha fatális hibát talál, ilyenkor a futás mindig terminálódik. A manipulációs rutinok megírásánál egyik fontos szempont volt, hogy a szükséges manipulációt csak akkor kezdje végrehajtani, ha nem talált hibát. Ilymódon ha a rutinok valamelyike hibát észlel és hibajelzéssel tér vissza, a rendszer állapota /ha ez lehetséges volt/, azonos marad a hívás előttivel.

Az egyes hibakódok:

- 1 OVERFLOW
- 2 HIBÁS KÖNYV
- 3 ILLETÉKTELEN IRÁSVÉDELMI KÓD
- 4 ILLETÉKTELEN OLVASÁSVÉDELMI KÓD
- 5 HIBÁS BUFFERCIM
- 6 KURRENS REKORD = \emptyset
- 7 KURRENS OWNER = \emptyset
- 8 KURRENS MEMBER = \emptyset

- 9 ILLEGÁLIS OWNER REKORDTIPUS
- 10 ILLEGÁLIS MEMBER REKORDTIPUS
- 11 ILLEGÁLIS REKORDTIPUS
- 12 HIBÁS ADATBÁZIS KULCS
- 13 EZ A REKORDELŐFORDULÁS NEM TULAJDONOS ILYEN
TIPUSU HALMAZBAN
- 14 EZ A REKORDELŐFORDULÁS NEM TAG ILYEN TIPUSU
HALMAZBAN
- 15 ÜRES A HALMAZ
- 16 VAN MÁR ILYEN AZONOSITÓJU REKORD
- 17 NINCS ILYEN REKORDELŐFORDULÁS
- 18 EZ VOLT AZ ELSŐ TAG A LÁNCON
- 19 EZ VOLT AZ UTOLSÓ TAG A LÁNCON
- 20 FORMÁTUMHIBA
- 21 INDEXHIBA
- 22 SZEKVENCIAÁLIS ELÉRÉSÜ EZ A REKORDTIPUS
- 23 MEZŐ ÉRTÉK HIBA
- 24 A MEZŐ TÖBBSZÖRÖS, MINT MEGENGEDETT
- 25 MUTATÓMEZŐ ÁTIRÁSI KISÉRLET ILLETÉKTELEN
RUTINNAL
- 26 EZ A REKORDTIPUS NEM HALMAZ-ELÉRÉSÜ
- 27 NINCS IMPLEMENTÁLVA
- 28 MEZŐHIBA
- 29 KRITÉRIUMHIBA
- 30 ILLEGÁLIS HALMAZTIPUS
- 31 RENDSZERHIBA
- 32 A HASH-TÁBLA BETELT

2. fejezet" ' /
AZ ADATKEZELO RUTINOK LEIRASA

A rekordok leírását igyekeztünk ABC-rendbe szedni, de ahol a funkcionális összefüggések szükségessé tették, eltekintettünk attól.

A leírásban szereplő rutinok a következők:

| | |
|--------|--------|
| ADDKM | OWNTIP |
| ADDKO | MEMBER |
| ADDKR | OWNER |
| ADDSET | OUTCM |
| CREATE | OUTSET |
| DELCR | PUTFCM |
| ERROR | PUTFCO |
| FNUM | PUTFCR |
| GETCO | RFIRST |
| GETCR | RLAST |
| GETRD | RNEXT |
| GETFCM | RPRED |
| GETFCO | RNUM |
| GETFCR | SFIRST |
| GETFDB | SLAST |
| KMDB | SNEXT |
| KMKO | SPRED |
| KMKM | SNUM |
| KMKR | REKORD |
| KOKM | RKEY |
| KOKO | |
| KOKR | |
| KODB | |
| KRDB | |
| MEMTIP | |

ADDKM

Deklaráció:

```
PROC ADDKM=(INT HT1, HT2) INT HIBA:
```

Paraméterek:

```
HT1      - halmaztípus  
HT2      - halmaztípus  
HIBA     - hibakód
```

Kurrens:

```
rekord: HT2 membre szerint  
owner:  változatlan  
member: HT2 membre szerint
```

Leírás:

HT2 halmaztípus kurrens member rekordját hozzáadja a HT1 halmaztípus kurrens owneréhez tartozó halmazhoz. Ha az adott rekord tagja volt már HT1 típusu halmaznak, akkor onnan leválasztja és utána hozzáadja ehhez a halmazhoz. HT2 halmaztípus kurrens mutatói változatlanok maradnak.

Hivott proc-ok:

```
MEMBER;ADDSET;ERROR
```

Hibakódok:

```
7, 10, 8, 30, 27, 31
```


ADDKO

Deklaráció:

```
PROC ADDKO=(INT HT1, HT2) INT HIBA:
```

Paraméterek:

```
HT1      - halmaztípus  
HT2      - halmaztípus  
HIBA     - hibakód
```

Kurrens:

```
rekord: HT2 ownere szerint  
owner:  változatlan  
member: HT2 ownere szerint
```

Leírás:

HT2 halmaztípus kurrens owner rekordját hozzáadja a HT1 halmaztípus kurrens owneréhez tartozó halmazhoz. Ha az adott rekord tagja volt már HT1 típusu halmaznak, akkor onnan előbb leválasztja, majd utána hozzáadja ehhez a halmazhoz.

HT2 halmaztípus kurrens mutatói változatlanok maradnak.

Hivott proc-ok:

```
OWNER;ADDSET;ERROR
```

Hibakódok:

```
7, 10, 30, 27, 31
```

ADDKR

Deklaráció:

PROC ADDKR=(INT HT, RT) INT HIBA:

Paraméterek:

HT - halmaztípus
RT - rekordtípus
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: kurrens rekord szerint

Leírás:

A RT rekordtípus kurrens rekordját hozzáadja a HT halmaztípus kurrens owneréhez tartozó HT típusu halmazhoz. Ha az adott rekord már tagja volt egy HT típusu halmaznak, akkor abból előbb leválasztja, és utána adja hozzá ehhez a halmazhoz.

Hivott proc-ok:

REKORD;ADDSET;ERROR

Hibakódok:

6, 7, 10, 30, 27, 31

ADDSET

Deklaráció:

PROC ADDSET=(INT HT; REF DBK CIM) INT HIBA:

Paraméterek:

HT - halmaztípus
CIM - adatbázis kulcs
HIBA - hibakód

Kurrens:

rekord: az adott rekord
owner: változatlan
member: az adott rekord

Leírás:

Az adott rekordot hozzáadja a HT halmaztípus kurrens owneréhez tartozó HT típusu halmazhoz. Ha az adott rekord tagja volt már egy HT típusu halmaznak, akkor onnan előbb leválasztja és utána adja hozzá ehhez az halmazhoz.

Hivott proc-ok:

OWNER, READBK, OUTSET, KMDB, SORBA, FELFUZ, MEMBER, PTUL, ERROR

Hibakódok:

7, 8, 10, 11, 12, 27, 30, 31

CREATE

Deklaráció:

PROC CREATE=(INT RT, MOD; REF CHAR B) INT HIBA:

Paraméterek:

RT - rekordtípus
MOD - formátumkód
B - memóriacím
HIBA - hibakód

Kurrens:

rekord: az új rekord

owner: változatlan

member: ahová automatikusan fel kell fűzni, ott az új rekord lesz a kurrens member

Leírás:

Lásd a következő lapon

Hivott proc-ok:

BERTIP;BEMEZO;VALUE;BEHALM;BEKRIT;BERTAG;BEHTAG;HELY;
WRITA;WRITB;MASIR;KRDB;ADDREC;BERTUL;WRDBK;BEHTIP;
REKORD;KIRTIP;FELFUZ;MEMBER;OWNER;PTUL;KMDB;SORBA;ERROR

Hibakódok:

6, 7, 8, 10, 11, 12, 14, 16, 22, 23, 25, 27, 29, 30,
31, 32

CREATE folyt.1.

Leírás:

Létesít egy új RT típusú rekordot.

A B memóriacimről elhelyezve találja az új rekord egyes mezőinek értékeit MOD szerinti formátumban /lásd később/. Először a rutin ellenőrzi a mezők értékeit, a rájuk előírt követelmények szerint. Ha a rekordot automatikusan fel kell venni valamilyen típusú halmazba, akkor a kurrens ownert az ilyen típusú halmaznál a rutin hívása előtt be kell állítani. Ha hibát talál valahol a rutin, úgy hibajelzést ad, de nem változtat meg semmit.

Ha nincs hiba, a rutin létesít egy új rekordot, s feltölti az adott értékekkel. Felfűzi a rekordtípusra előírt rendezési kritériumok szerint a rendezési láncokra.

Ha a rekordra elő van írva az automatikus halmaztagság, valamely halmaztípusban, akkor a rekord felfűzi a halmaz kurrens owneréhez tartozó halmazba.

Ha a rekord valamely halmaztípusban owner lehet, úgy egy ilyen típusú üres halmaz keletkezik. /Nincs tagja./

Formátum:

1. MOD < ϕ /adott, hogy hány-szoros a mező/

A B-ben adott kezdőcímtől kezdve vannak letéve az egyes mezők a rekord definiálása szerinti sorrendben, kivéve a mutató mező/ke/t, melyek a sorból kimaradnak. Ezek értékét a rendszer maga állapítja meg. A többszörös mezőket egy integer szám /2 byte/ előzi meg, mely mutatja, hogy ebben a rekordban ebből a mezőből hány lesz.

A CHAR és STRING típusú mezőket kivéve a mezőknek páros memóriacimen kell kezdődni. Ha ez nem oda kerül, a rendszer kihagy egy byte-ot és a következő

páros címtől várja a mezőt. A közbenső "töltelék" byte értéke közömbös. Pl. a rekord mezői sorban CHAR, INT, CHAR, REAL típusúak. A bufferban az első mező páros címen kezdődik: 1. byte az első mező, majd egy töltelék byte következik, majd két byte a második /INT/ mező, 1 byte a harmadik mező, megint egy töltelék byte következik stb./

2. MOD $\geq \emptyset$ /Terminátoros formátum/

A formátum hasonló mint az előbb, csak a többszörös mezőknél van különbség:

A többszörös mezőknél nincs megadva, hogy hányszoros lesz, hanem egymás után le vannak téve a mezők és a végén van egy u.n. "terminátor" mező. Ilyenkor a rendszer számolja össze a mezőket.

Terminátor mező: Mérete azonos a mezőmérettel, kivéve a STRING típust, melynél 1 byte függetlenül a string típusu mező aktuális hosszától. A terminátor mező első bitje = 1 /a legmagasabb helyiértékű bit/, a többi mint nulla /pl. integer típusnál :8000/.

ERROR

Deklaráció:

PROC ERROR=(INT HIBA):

Paraméterek:

HIBA - hibakód

Kurrens:

rekord: változatlan

owner: változatlan

member: változatlan

Leírás:

Az adott hibakódhoz tartozó hibaüzenet kiírása.- A SZINT nevű globális változó értékétől függően történik a rutinból való visszatérés:

Ø = a vezérlést visszakapja a hívó program

1 = a futás terminálódik

Hivott proc-ok:

-

Hibakódok:

-

DELCR

Deklaráció:

PROC DELCR=(INT RT) INT HIBA:

Paraméterek:

RT - rekordtípus
HIBA - hibakód

Kurrens:

rekord: nullázódik

owner: ahol ez a rekord owner volt, ott a kurrens owner nullázódik

member: ahol ez a rekord szerepelt, ott a kurrens member nullázódik, ha a törlendő volt a kurrens

Leírás:

A RT típusu kurrens rekordot törli az adatbázisból.

Ha ez a rekord owner volt valamely halmazban, akkor a hozzá tartozó member rekordokat leválasztja róla.

Azokból a halmazokból, ahol ez a rekord memberként szerepel, leválasztjuk őt a törlés előtt.

Hivott proc-ok:

BERTIP;REKORD;OUTREC;BEHALM;OUTSET;BERTUL;MEMBER;KODB;
SFIRST;OUTCM;KIHTIP;URIT;MASOL;BEMEZO;CALC;WRITA;READA;
KMDB;KIRTIP

Hibakódok:

22, 6, 27, 30, 31, 29, 11, 12

FNUM

Deklaráció:

PROC FNUM=(INT RT, M) INT N:

Paraméterek:

RT - rekordtípus
M - mező sorszáma
N - a mező aktuális darabszáma

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az RT rekordtípus kurrens rekordja M-edik mezejének aktuális darabszámát adja vissza N-ben.

Hivott proc-ok:

BERTIP, BEMEZO, READA

Hibakódok:

-

GETFDB

Deklaráció:

```
PROC GETFDB=(REF DBK CIM; INT M,X; REF CHAR C) INT HIBA:
```

Paraméterek:

```
CIM      - adatbázis kulcs  
M        - mező sorszáma  
X        - mező index  
C        - memória cím  
HIBA     - hibakód
```

Kurrens:

```
rekord: változatlan  
owner:  változatlan  
member: változatlan
```

LEIRÁS:

A CIM-ben adott rekordból az M-edik mezőt kimásolja a C-ben adott memória címre. Az X index értéke mutatja, hogy többszörös mező esetén hányadikat kérjük. Egyszeres mező esetén X értéke közömbös. Ha X=0, akkor az összes "M" mezőt visszaadja a "C" memóriacimtől kezdve folyamatosan.

Hivott proc-ok:

```
DBKELL, BEMEZO, MASOL, READA
```

Hibakódok:

```
11, 12, 21, 22, 28
```

GETFCR

Deklaráció:

PROC GETFCR=(INT RT,M,X; REF CHAR C) INT HIBA:

Paraméterek:

RT - rekordtípus
M - mező sorszáma
X - mező index
C - memória cím
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az adott rekordtípus kurrens rekordjából az M-edik mezőt kimásolja az adott memóriacímre.

Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat kérik. Egyszeres mező esetén X értéke közömbös. Ha X=0, akkor az összes "M" mezőt visszaadja a "C" memóriacímtől kezdve folyamatosan.

Hivott proc-ok:

REKORD, GETFDB

Hibakódok:

6, 11, 12, 21, 22, 28, 30, 31

GETFCO

Deklaráció:

```
PROC GETFCO=(INT HT,M,X; REF CHAR C) INT HIBA:
```

Paraméterek:

HT - halmaztípus
M - mező sorszáma
X - mező index
C - memória cím
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az adott halmaztípus kurrens owner rekordjából az M-edik mezőt kimásolja az adott memóriacímre. Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat kéri. Egyszeres mező esetén X értéke közömbös. Ha X=0, akkor az összes "M" mezőt visszaadja a "C" memóriacímtől kezdve folyamatosan.

Hivott proc-ok:

OWNER, GETFDB

Hibakódok:

7, 11, 12, 21, 22, 28, 30, 31

GETFCM

Deklaráció:

```
PROC GETFCM=(INT HT,M,X; REF CHAR C) INT HIBA:
```

Paraméterek:

HT - halmaztípus
M - mező sorszáma
X - mező index
C - memória cím
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az adott halmaztípus kurrens member rekordjából az M-edik mezőt kimásolja az adott memóriacímre. Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat kérjük. Egyszeres mező esetén X értéke közömbös. Ha $X = 0$, akkor az összes "M" mezőt visszaadja a "C" memóriacímtől kezdve folyamatosan.

Hivott proc-ok:

GETFDB, MEMBER

Hibakódok:

21, 22, 28, 11, 12, 8, 30, 31

GETRD

Deklaráció:

```
PROC GETRD=(REF DBK CIM: REF CHAF: B; INT MOD) INT HIBA:
```

Paraméterek:

```
CIM      - adatbázis kulcs  
B        - memória cím  
MOD      - formátumkód  
HIBA     - hibakód
```

Kurrens:

```
rekord: változatlan  
owner:  változatlan  
member: változatlan
```

Leírás:

Visszaadja a CIM-ben adott rekord összes mezejét: B memóriacimtől kezdve elhelyezve, MOD szerinti formátumban. /A formátum megegyezik a CREATE rutin bemenő formátumával./

/Ha MOD > 0, akkor terminátoros formátum, különben a többszörös mezőket egy integer mező előzi meg, amely tartalmazza, hogy hányszoros a mező./

Hivott proc-ok:

```
DBKELL;BEMEZO;GETFDB
```

Hibakódok:

```
11, 12, 21, 22, 28
```

GETCR

Deklaráció:

```
PROC GETCR=(INT RT; REF CHAR B; INT MOD) INT HIBA:
```

Paraméterek:

```
RT          - rekordtípus  
B           - memória cím  
MOD         - formátumkód  
HIBA        - hibakód
```

Kurrens:

```
rekord: változatlan  
owner:  változatlan  
member: változatlan
```

Leírás:

Visszaadja RT rekordtípus kurrens rekordjának minden mezejét. B memóriacimről kezdve helyezi el a mezőket MOD szerinti formátumban. /Ha MOD > 0: terminátoros formátumban, különben a többszörös mezőket egy integer mező előzi meg, amely tartalmazza, hogy hányszoros a mező./ A formátum /MOD/ megegyezik a CREATE rutin bemenő formátumával.

Hivott proc-ok:

```
REKORD, GETRD
```

Hibakódok:

```
6, 11, 12, 21, 22, 28, 30, 31
```

GETCO

Deklaráció:

PROC GETCO=(INT HT; REF CHAR B; INT MOD) INT HIBA:

Paraméterek:

HT - halmaztípus
B - memória cím
MOD - formátumkód
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Visszaadja a HT halmaztípus kurrens owner rekordjának minden mezejét. B memóriacimről kezdve helyezi el a mezőket a MOD szerinti formátumban. /Ha MOD > 0, terminátoros formátumban, különben a többszörös mezőket egy integer mező előzi meg, amely tartalmazza, hogy hányszoros a mező./ A formátum megegyezik a CREATE rutin bemenő formátumával.

Hivott proc-ok:

OWNER, GETFD

Hibakódok:

7, 11, 12, 21, 22, 28, 30, 31

GETCM

Deklaráció:

```
PROC GETCM=(INT HT; REF CHAR B; INT MOD) INT HIBA:
```

Paraméterek:

```
HT          - halmaztípus  
B           - memóriacím  
MOD         - formátumkód  
HIBA       - hibakód
```

Kurrens:

```
rekord: változatlan  
owner:  változatlan  
member: változatlan
```

Leírás:

Visszaadja a HT halmaztípus kurrens member rekordjának minden mezejét. B memóriacímtől kezdve helyezi el a mezőket a MOD szerinti formátumban. /Ha MOD > 0, terminátoros formátumban, különben a többszörös mezőket egy integer mező előzi meg, mely tartalmazza, hogy hányszoros a mező./ A formátum megegyezik a CREATE rutin bemenő formátumával.

Hivott proc-ok:

```
MEMBER, GETRD
```

Hibakódok:

```
8, 11, 12, 21, 22, 28, 30, 31
```

KMKM

Deklaráció:

PROC KMKM=(INT HT1,HT2) INT HIBA:

Paraméterek:

HT1 - halmaztípus /beállítandó/
HT2 - halmaztípus
HIBA - hibakód

Kurrens:

rekord: member és owner szerint

owner: az adott memberhez tartozó owner

member: másik halmaztípus kurrens member szerint

Leírás:

A HT2 halmaztípus kurrens membere lesz a HT1 halmaztípus kurrens membere is. A HT1 halmaztípus kurrens ownere az új kurrens memberhez tartozó owner lesz. A HT2 halmaztípus kurrens-mutatói változatlanok maradnak. Ha az új kurrens member és owner azonos típusú rekord, úgy a kurrens rekordmutató a memberre mutat.

Hivott proc-ok:

MEMBER, KMDB, ERROR

Hibakódok:

8, 10, 14, 31, 30

KMKO

Deklaráció:

PROC KMKO=(INT HT1,HT2) INT HIBA:

Paraméterek:

HT1 - halmaztípus /beállítandó/
HIBA - hibakód
HT2 - halmaztípus

Kurrens:

rekord: member és owner szerint
owner: az adott memberhez tartozó owner
member: másik halmaztípus kurrens ownere szerint

Leírás:

A HT2 halmaztípus kurrens ownere lesz a HT1 halmaztípus kurrens membre.

A HT1 halmaztípus kurrens ownere az új kurrens memberhez tartozó owner lesz.

A HT2 kurrensmutató változatlanul maradnak.

Ha az új kurrens member és owner azonos típusu, úgy a kurrens rekord a member lesz.

Hivott proc-ok:

OWNER, KMDB, ERROR

Hibakódok:

7, 10, 14, 31, 30

KMKR

Deklaráció:

PROC KMKR=(INT HT,RT) INT HIBA:

Paraméterek:

HT - halmaztípus
RT - rekordtípus
HIBA - hibakód

Kurrens:

rekord: változatlan /ill. új owner szerint/
owner: az adott member-hez tartozó owner
member: kurrens rekord szerint

Leírás:

Az adott rekordtípus kurrens rekordját teszi az adott halmaztípus kurrens memberévé. Az adott halmaztípus kurrens ownere az adott memberhez tartozó rekord lesz. Ha az új kurrens member és owner azonos típusu, úgy a kurrens rekordmutató a memberre mutat.

Hívott proc-ok:

REKORD, KMDB, ERROR

Hibakódok:

6, 10, 14, 31, 11, 30

KOKO

Deklaráció:

PROC KOKO=(INT HT1,HT2) INT HIBA:

Paraméterek:

HT1 - - halmaztípus /beállítandó/
HT2 - - halmaztípus
HIBA - - hibakód

Kurrens:

rekord: OWNER szerint

owner: HT2 halmaztípus kurrens ownere lesz HT1 halmaztípusban is a kurrens owner

member: nullázódik

Leírás:

A HT2 halmaztípus kurrens ownere lesz a HT1 halmaztípus kurrens ownere is. A HT1 halmaztípus kurrens member-e nullázódik. A HT2 kurrens-mutatói változatlanok maradnak.

Hivott proc-ok:

OWNER, KODB, ERROR

Hibakódok:

7, 30, 31

KOKM

Deklaráció:

PROC KOKM=(INT HT1,HT2) INT HIBA:

Paraméterek:

HT1 - halmaztípus /beállítandó/
HT2 - halmaztípus
HIBA - hibakód

Kurrens:

rekord: owner szerint
owner: HT2 kurrens membre szerint
member: nullázódik /HT1/

Leírás:

A HT2 halmaz kurrens membre lesz a HT1 halmaz-ban kurrens owner. HT1 halmaz kurrens membre nullázódik. HT2 halmaz kurrens-mutatói változatlanok.

Hivott proc-ok:

MEMBER, KODB, ERROR

Hibakódok:

8, 30, 31

KOKR

Deklaráció:

PROC KOKR=(INT HT,RT) INT HIBA:

Paraméterek:

HT - halmaztípus
RT - rekordtípus
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: kurrens rekord szerint
member: nullázódik

Leírás:

Az adott halmaztípusban kurrens ownerré teszi az adott rekordtípus kurrens rekordját.

Hivott proc-ok:

REKORD, KODB, ERROR

Hibakódok:

6, 11, 31

KRDB

Deklaráció:

PROC KRDB=(REF DBK CIM) INT HIBA:

Paraméterek:

CIM - adatbázis kulcs
HIBA - hibakód

Kurrens:

rekord: beállítódik az adatbázis kulcsra
owner: változatlan
member: változatlan

Leírás:

A CIM-ben megadott helyről vesz egy adatbázis kulcsot és azt állítja be kurrens rekordnak.

Hivott proc-ok:

DBKELL;KULCS;KIRTIP

Hibakódok:

11, 12

KODB

Deklaráció:

```
PROC KODB=(INT HT; REF DBK CIM) INT HIBA;
```

Paraméterek:

```
HT      - halmaztípus  
CIM     - adatbázis kulcs  
HIBA    - hibakód
```

Kurrens:

```
rekord: az adatbázis kulcsra állítódik  
owner:  az adatbázis kulcsra állítódik  
member: nullázódik
```

Leírás:

A címben megadott helyről vesz egy adatbázis kulcsot és azt beállítja az adott halmaztípusban kurrens tulajdonosnak.

Hivott proc-ok:

```
DBKELL;KULCS;PTUL;BERTIP;KIRTIP;KIHTIP
```

Hibakódok:

```
9, 11, 12, 30
```

KMDB

Deklaráció:

PROC KMDB=(INT HT; REF DBK CIM) INT HIBA:

Paraméterek:

HT - halmaztípus
CIM - adatbázis kulcs
HIBA - hibakód

Kurrens:

rekord - owner és member szerint
owner - az adott memberhez tartozó owner
member - cim-ben adott rekord

Leírás:

A "CIM"-ben megadott helyről vesz egy adatbázis kulcsot és azt beállítja az adott halmaztípusban kurrens tagnak. Kurrens rekord állítás történik mind az owner, mind a member szerint. Ha ezek azonos típusu rekordok, akkor a member rekord lesz beállítva kurrens rekordnak.

Hivott proc-ok:

DBKELL, ERROR, PTAG, READBK, BEHTIP, BERTIP, KRDB, KIHTIP

Hibakódok:

11, 12, 10, 14, 31, 30

MEMTIP

Deklaráció:

PROC MEMTIP=(INT HT) INT TK;

Paraméterek:

HT - halmaztípus

TK - tipuskód

Kurrens:

rekord: változatlan

owner: változatlan

member: változatlan

Leírás:

Az adott halmaztípus kurrens memberének tipuskódját
/a rekordtípus sorszámát/ adja vissza.

Hivott proc-ok:

BEHTIP

Hibakódok:

-

OWNTIP

Deklaráció:

PROC OWNTIP=(INT HT) INT TK:

Paraméterek:

HT - halmaztípus

TK - a kurrens owner tipuskódja

Kurrens:

rekord: változatlan

owner: változatlan

member: változatlan

Leírás:

Az adott halmaztípus kurrens ownerének tipuskódját
/a rekordtípus sorszámát/ adja vissza .

Hivott proc-k:

BEHTIP

Hibakódok:

-

MEMBER

Deklaráció:

PROC MEMBER=(INT HT; REF DBK CIM) INT HIBA:

Paraméterek:

HT - halmaztípus
CIM - a kurrens tag adatbázis kulcsa

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

A "CIM"-ben megadott helyre betölti a kurrens tag adatbázis kulcsát.

Hivott proc-ok:

DBKELL;BEHTIP

Hibakódok:

8, 30, 31

OWNER

Deklaráció:

PROC OWNER=(INT HT; REF DBK CIM) INT HIBA;

Paraméterek:

HT - halmaztípus

CIM - a kurrens tulajdonos adatbázis kulcsa

Kurrens:

rekord: változatlan

owner: változatlan

member: változatlan

Leírás:

A "CIM"-ben megadott helyre betölti a kurrens tulajdonos adatbázis kulcsát.

Hivott proc-ok:

DBKELL; BEHTIP

Hibakódok:

7, 30, 31

OUTCM

Deklaráció:

PROC OUTCM=(INT HT) INT HIBA:

Paraméterek:

HT - halmaztípus
HIBA - hibakód

Kurrens:

rekord: member szerint
owner: változatlan
member: nullázódik

Leírás:

A HT halmaztípus kurrens memberét leválasztja /törli/
a halmazból. A kurrens member kurrens rekorddá válik.
A kurrens member mutatója nullázódik.

Hivott proc-ok:

MEMBER;OUTSET

Hibakódok:

8, 10, 11, 12, 14, 27, 30, 31

OUTSET

Deklaráció:

PROC OUTSET=(INT HT; REF DBK CIM) INT HIBA:

Paraméterek:

HT - halmaztípus
CIM - adatbázis kulcs
HIBA - hibakód

Kurrens:

rekord: az adott rekord
owner: változatlan
member: nullázódik, ha az adott rekord volt a kurrens

Leírás:

Az adott rekordot - ha az tag egy HT típusu halmazban - leválasztja /törli/ a halmazból. Az adott rekord kurrens rekord lesz. Ha az adott rekord volt a kurrens member a HT típusu halmazban, akkor a kurrens member nullázódik; különben változatlan.

Hivott proc-ok:

DBKELL;PTAG;BEHTIP;KRDB;READBK;PTUL;WRDBK;READA;WRITA;
MEMBER;REKORD;KIHTIP

Hibakódok:

10, 11, 12, 14, 27

PUTFCM

Deklaráció:

PROC PUTFCM=(INT HT,M,X; REF CHAR C) INT HIBA:

Paraméterek:

HT - halmaztípus
M - mező sorszáma
X - mező index
C - memória cím
HIBA - hibakód

Kurrens:

rekord: MEMBER szerint

owner: változatlan

member: változatlan

Leírás:

A HT halmaztípus kurrens memberének "M" mezejének felülírása a "C" memória címen található értékkel. Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat akarjuk felülírni. Egyszeres mező esetén X értéke közömbös.

Hivott proc-ok:

MEMBER, PUTFCR, KRDB

Hibakódok:

8, 11, 21, 23, 25, 27, 28, 30, 31

PUTFCO

Deklaráció:

PROC PUTFCO=(INT HT,M,X; REF CHAR C) INT HIBA:

Paraméterek:

HT - Halmaztípus
M - Mező sorszáma
X - Mező index
C - Memória cím
HIBA - Hibakód

Kurrens:

rekord: OWNER szerint
owner: változatlan
member: változatlan

Leírás:

A HT halmaztípus kurrens ownere "M" mezejének felülírása a "C" memória címen található értékkel. Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat akarjuk felülírni. Egyszeres mező esetén X értéke közömbös.

Hivott proc-ok:

OWNER, PUTFCR, KRDB

Hibakódok:

7, 11, 21, 23, 25, 27, 28, 30, 31

RFIRST

Deklaráció:

PROC RFIRST=(INT RT,KR) INT HIBA:

Paraméterek:

RT - rekordtípus
KR - kritérium száma
HIBA - hibakód

Kurrens:

rekord: a felfüzdött rekordok közül első lesz a kurrens
owner: változatlan
member: változatlan

Leírás:

A RT rekordtípus KR-edik kritériuma szerint felfüzdött rekordok közül az első irja be a kurrensnek a rekordtípus táblába.

Hivott proc-ok:

KRELL;BEKRIT;KRDB;BERTIP;ZAR;ERROR

Hibakódok:

11, 29, 17, 31

PUTFCR

Deklaráció:

PROC PUTFCR=(INT RT,M,X; REF CHAR C) INT HIBA:

Paraméterek:

RT - rekordtípus
M - mező sorszáma
X - mező index
C - memória cím
HIBA - hibakód

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az RT rekordtípus kurrens rekordjának "M"-edik mezőjének felülírása a "C" memória címen található értékkel. Az "X" index értéke mutatja, hogy többszörös mező esetén hányadikat akarjuk felülírni. Egyszeres mező esetén X értéke közömbös.

Hivott proc-ok:

REKORD, BERTIP, BEMEZO, VALUE, READA, MASIR

Hibakódok:

11, 21, 23, 25, 27, 28, 6, 30, 31

RLAST

Deklaráció:

```
PROC RLAST=(INT RT,KR) INT HIBA:
```

Paraméterek:

```
RT          - rekordtípus  
KR          - kritérium száma  
HIBA       - hibakód
```

Kurrens:

```
rekord: a felfüzt rekordok közüli utolsó lesz a kurrens  
owner: változatlan  
member: változatlan
```

Leírás:

A RT rekordtípus KR-edik kritériuma szerint felfüzt rekordok közüli az utolsót írja be a kurrensnek a rekordtípus táblába.

Hivott proc-ok:

```
KRELL;UAR;BEKRIT;KRDB;ERROR
```

Hibakódok:

```
11, 29, 17, 31
```

RNEXT

Deklaráció:

PROC RNEXT=(INT RT,KR) INT HIBA:

Paraméterek:

RT - rekordtípus
KR - kritérium száma
HIBA - hibakód

Kurrens:

rekord: a hívás előtti kurrens tagot követő lesz az új
kurrens

owner: változatlan

member: változatlan

Leírás:

A RT rekordtípus KR-edik kritériuma szerint felfűzött rekordok közül a hívás előtti kurrens tagot követő tag lesz az új kurrens a rekordtípus táblában.

Hivott proc-ok:

KRELL;BEKRIT;REKORD;READA;KRDB;ERROR

Hibakódok:

11, 29, 17, 31, 6, 19

RPRED

Deklaráció:

PROC RPRED=(INT RT,KR) INT HIBA:

Paraméterek:

RT - rekordtípus
KR - kritérium száma
HIBA - hibakód

Kurrens:

rekord: a hívás előtt kurrens tag előtti lesz az új
kurrens

owner: változatlan

member: változatlan

Leírás:

A RT rekordtípus KR-edik kritériuma szerint felfűzött rekordok közül a hívás előtti kurrens tagot megelőző tag lesz az új kurrens a rekordtípus táblában.

Hívott proc-ok:

KRELL;BEKRIT;REKORD;KRDB;READA;ERROR

Hibakódok:

11, 29, 17, 18, 6, 31

RKEY

Deklaráció:

```
PROC RKEY=(INT RT,KR,M; REF CHAR C) INT HIBA:
```

Paraméterek:

```
RT      - rekordtípus
KR      - kritérium száma
M       - mező sorszáma
C       - memória cím, ahol a "minta" található
HIBA    - hibakód
```

Kurrens:

rekord: a keresett rekord, ill. ha nincs megfelelő,
akkor változatlan

owner: változatlan

member: változatlan

Leírás:

Megkeresi azt az RT típusu rekordot, melynek az M-edik mezeje egyenlő a C memóriacímen található minta értékkel.

A keresést a rutin a KR rendezési kritérium szerinti sorrendben végzi az első rekordtól kezdve. A kurrens rekord mutatót beállítja az első olyan rekordra, melynek M-edik mezeje egyenlő a minta értékkel; ha ilyet nem talál, a kurrens mutató változatlan marad, és hibajelzést kapunk. Az M-edik mező csak egyszeres mező lehet.

Abban az esetben, ha M azonosító mező és a rekordtípus DIRECT elérésű, úgy a rutin a rekordot közvetlenül eléri.

Hívott proc-ok:

```
BERTIP; BEMEZO; READA; BEKRIT; MASOL; COMPER; KRDB; ERROR;  
ZAR; DIRECT
```

Hibakódok:

```
11, 17, 28, 29, 31
```


RNUM

Deklaráció:

```
PROC RNUM=(INT RT, KR) INT N:
```

Paraméterek:

```
RT      - rekordtípus  
KR      - kritérium száma  
N       - rekordok száma
```

Kurrens:

```
rekord: változatlan  
owner:  változatlan  
member: változatlan
```

Leírás:

Az adott rekordtípus adott rendezési kritériuma szerint sorbafűzött rekordok számát adja vissza N-ben.

Hivott proc-ok:

```
BEKRIT, BERTIP
```

Hibakódok:

-

SFIRST

Deklaráció:

PROC SFIRST=(INT HT) INT HIBA:

Paraméterek:

HT - halmaztípus
HIBA - hibakód

Kurrens:

rekord: első rekordra fog mutatni
owner: változatlan
member: első rekordra fog mutatni

Leírás:

A HT halmaztípusban felfűzött rekordok közül az elsőt teszi kurrenssé a halmazban a kurrens member mutató átirásával.

Hivott proc-ok:

OWNER, READBK, PTUL, KMDB, ERROR

Hibakódok:

15, 31, 7, 30

SLAST

Deklaráció:

```
PROC SLAST=(INT HT) INT HIBA:
```

Paraméterek:

```
HT      - halmaztípus  
HIBA    - hibakód
```

Kurrens:

```
rekord: az utolsó rekordra fog mutatni  
owner:  változatlan:  
member: az utolsó rekordra fog mutatni
```

Leírás:

A HT halmaztípusban a felfűzött rekordok közül az utolsót teszi kurrenssé a halmazban a kurrens member mutató átirásával.

Hivott proc-ok:

```
OWNER;READBK;PTUL;KMDB;ERROR
```

Hibakódok:

```
15, 31, 7, 30
```

SNEXT

Deklaráció:

PROC SNEXT=(INT HT) INT HIBA:

Paraméterek:

HT - halmaztípus
HIBA - hibakód

Kurrens:

rekord: a hívás előtti aktuális utánira változik
owner: változatlan
member: a hívás előtti aktuális utánira változik

Leírás:

A HT halmaztípusban felfűzött rekordok közül a hívás előtti kurrens utáni lesz a halmazban a kurrens

Hivott proc-ok:

MEMBER;READBK;PTAG;KMDB;ERROR

Hibakódok:

19, 31, 8, 30

OPEN

Deklaráció:

```
PROC OPEN=(INT BASE; REF [ ] CHAR SZOVEG):
```

Paraméterek:

BASE - könyvtár szám (0,1,2, lehet)
SZÖVEG - console-ra kiírandó szövegre mutató pointer

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Adatbázis megnyitása "BASE"-ban megadott könyvtárszámhoz file hozzárendelése. A rendszer kiírja a console-ra a "SZÖVEG"-et és az operátortól kér egy file nevet. Egyszerre maximum 3 adatbázis lehet nyitott állapotban. Az "open" után az éppen megnyitott adatbázis válik kurrensé. (Meghívódik BASE-val a CHANGE rutin.) Ha egy már felhasznált könyvtárszámot újból használunk, úgy a régi hozzárendelés megszűnik, a file lezáródik, és az új rendelődik hozzá.

Hivott proc-ok:

CHANGE

Hibakódok:

-

CHANGE

Deklaráció:

PROC CHANGE=(INT BASE):

Paraméterek:BASE - Könyvtár szám (\emptyset ,1,2 lehet)Kurrens:

rekord: _

owner: _

member: _

Leírás:

Adatbázis váltás. A rutin hatására a "BASE"-ban megadott számú nyitott adatbázis válik kurrensé. Ezután az adatbázis kezelő rutinok ezen az adatbázison fognak dolgozni.

Hivott proc-ok:

MASOL

Hibakódok:

-

FREE

Deklaráció:

PROC FREE=:

Paraméterek:

-

Kurrens:

rekord: _

owner: _

member: _

Leírás:

Könyvtárszám hozzárendelés megszüntetése. Az adott file lezárása a DOST-ban. (Csak akkor szükséges kiadni, ha fel akarunk szabadítani egy file-t, mert nincs több megnyitási lehetőség DOST-ban (max 7 file lehet nyitva egyszerre.)

Hivott proc-ok:

-

Hibakódok:

-

SPRED

Deklaráció:

PROC SPRED=(INT HT) INT HIBA:

Paraméterek:

HT - halmaztípus

HIBA - hibakód

Kurrens:

rekord: a hívás előtt kurrens előttire változik

owner: változatlan

member: a hívás előtt kurrens előttire változik

Leírás:

A HT halmaztípusban felfűzött rekordok közül a hívás előtti kurrens rekord utáni lesz a halmazban a kurrens.

Hivott proc-ok:

MEMBER;READBK;PTAG;KMDB;ERROR

Hibakódok:

18, 31, 8, 30

SNUM

Deklaráció:

PROC SNUM=(INT HT) INT N:

Paraméterek:

HT - halmaztípus
N - a halmaz tagjainak száma

Kurrens:

rekord: változatlan
owner: változatlan
member: változatlan

Leírás:

Az adott halmaztípus kurrens owneréhez tartozó tagok számát adja vissza N-ben.

Hivott proc-ok:

OWNER, READBK, PTUL, PTAG, READA

Hibakódok:

- Ha hiba van, negatív számot ad vissza.

REKORD

Deklaráció:

PROC REKORD=(INT RT; REF DBK CIM) INT HIBA:

Paraméterek:

RT - rekordtípus

CIM - a kurrens rekord adatbázis kulcsa

Kurrens:

rekord: változatlan

owner: változatlan

member: változatlan

Leírás:

A "CIM"-ben megadott helyre betölti a kurrens rekord adatbázis kulcsát

Hivott proc-ok:

BERTIP;DBKELL

Hibakódok:

6,11,31

D D L

DATA DESCRIPTION LANGUAGE -
- ADATBÁZIS DEFINÍCIÓ NYELV

BEVEZETÉS

A DDL nyelv az adatbázis strukturájának definiálására szolgál. Leírható vele az adatbázis logikai és fizikai strukturája. A logikai felépítéshez tartoznak azok az információk, melyek meghatározzák, hogy milyen rekord-típusok szerepelnek az adatbázisban, hogy a rekordok milyen mezőkből állnak, és hogy a rekordok között milyen kapcsolat típusok léteznek. A fizikai felépítéshez tartoznak olyan információk, melyek meghatározzák pl. az egyes rekordok tárolási módját /elérési módját/, vagy pl. az egyes kapcsolat típusok /halmaz típusok/ megvalósítási módját. /Egyutas vagy kétutas lista lehet, vagy tartalmaznak-e a tag-rekordok pointert a tulajdonosra/.

A DDL nyelvű leírásból egy program generálja az üres adatbázist. Ilyen üres adatbázis tartalmaz minden strukturális információt és minden szükséges inicializálást is elvégeztek rajta, így alkalmas a tényleges adatok fogadására. Az adatbázis kezelő rendszer a KALAP diszk-kezelő rendszerre [1] épült. A DDL leírást feldolgozó program a COMPLETER rendszer [2] alatt fut.

A DDL feldolgozó program futtatása előtt diszk területet kell allokálni a KALAP BOOK nevű segédprogramjával. A DOST operációs rendszer [3] számára a teljes adatbázis egy 10-es könyvtárosztályu file. A DDL leírást pedig egy TEXT-file-ban kell elhelyezni /3. könyvtárosztály/.

A COMPLETER először megkérdezi, hogy melyik rendszert kívánjuk futtatni, majd meg kell adni az input file nevét /amely a DDL leírást tartalmazza/, végül megkérdezi az inicializálandó adatbázis nevét /melyet a BOOK programmal hoztunk létre/.

A DDL szabad formátumu command-nyelv. Utasításainak általános felépítése:

Változónév=Főszó/paraméterek ', '-vel elválasztva;

A paraméterek változó nevek, módosítók, különböző konstansok lehetnek. A változónév nem számmal kezdődő, max. 6 karakterből álló jelsorozat. Az utasításban tetszőleges számú "space" vagy "crlf" karakter lehet, ezek ugyanis kiértékeléskor szűrődnek. A változó helyén a paraméterlistában bárhol szerepelhet az ő definíciója is, zárójelek közé zárva. A leírásban minden változónak predefinitnek kell lennie. Az utasítás végét ";" karakter jelzi. A paraméterlistát "/" karakter helyett nyitó zárójel is kezdheti, ilyenkor azonban a lista végére záró zárójelet is kell helyezni.

A változónevek megőrződnek az adatbázisban is. E max. 6 karakterből álló néven kívül adhatunk minden logikai egységnek u.n. "hosszu nev"-et is, amely egy max. 24 karakterből álló tetszőleges jelsorozat. Ez szintén megőrződik az adatbázisban. Ennek akkor vehetjük hasznát, amikor olyan listákat, riportokat készítünk az adatbázis tartalmáról, melyeknek külső emberek számára is érthetőeknek kell lenni. /Általában a 6-karakteres név ehhez túl rövid./

A leírás utasításonként értékelődik ki. Természetesen ha zárójelben egy másik utasítás is van beleskatulyázva, úgy az előbb fog kiértékelődni. Ha a feldolgozó program hibát észlel, error-üzenetet ad. A kiértékelés - ha erre lehetőség van - folytatódik, de a hibás utasítás figyelmen kívül marad. Egymásba skatulyázott utasítások esetén a kiértékelés csak a következő ";" jeltől folytatódik, bárhol is volt a hiba.

UTASÍTÁSOK

Az utasítások leírásánál a "[]" zárójelek közé irt jelek opcionálisak, a "{ }" kapcsos zárójelek közé helyezett elemek közül valamelyiknek szerepelnie kell az adott helyen.

FIELD

Mező definiáló utasítás.

Általános alakja:

```
név=FIELD/'hosszu név', tipus[,KEY][,db][,ellenőrzés];
```

- *név* egy változónév. Előzőleg "FIELD"-nek definiált név újradefiniálható.
- *hosszu név* max. 24 karakteres szöveg
- *tipus*

| | |
|---|--|
| } | <pre>CHAR[,COUNT] INT[,COUNT] LINT REAL LREAL STRING,méret</pre> |
|---|--|

Az adott módosító a mező típusát jelzi, értelmezése azonos a megszokottal. *STRING* típusu mező esetén meg kell adni annak méretét is $\neq \emptyset$. *CHAR* mezőtípus nemcsak betűként értelmeződik: lehet 1 byte hosszú integernek is tekinteni. *COUNT* módosító azt jelenti, hogy a definiálandó mező u.n. "mutató mező" lesz. Ez azt

- jelent, hogy ennek a mezőnek az értékét a rendszer automatikusan tölti, aszerint, hogy egy vele azonos rekordelőfordulásban szereplő egyik többszörös mező éppen hányszor fordul elő. Az ilyen mező automatikusan kiemelt mező lesz. A mutató mező csak egyszeres mező lehet.
- *KEY* ha ez a módosító szerepel az utasításban, úgy a mező kiemelt mező lesz. Csak egyszeres mezőben szerepelhet ez a módosító szó.
 - *db* maximális darabszám / $\emptyset < \text{integer}$ /. Ez az integer konstans azt mutatja meg, hogy ez a mező max. hányszor szerepelhet egy rekordelőfordulásban /hányszoros mező/. Default értéke egy.
 - *ellenőrzés* ha mezőre van előírva ellenőrzés, úgy minden ilyen rekord létesítésénél ill. módosításánál a rendszer megvizsgálja, hogy a mező értéke megfelelő-e. Ha nem, úgy hibajelzést ad.

Előírható ellenőrzések:

| | | |
|-------------------|-----------------------------------|-------------------|
| <i>LT, a</i> | $(x < a)$ | x a mező értéke |
| <i>GT, a</i> | $(x > a)$ | |
| <i>LE, a</i> | $(x \leq a)$ | |
| <i>GE, a</i> | $(x \geq a)$ | |
| <i>GTLT, a, b</i> | $(a < x < b)$ | |
| <i>GTLE, a, b</i> | $(a < x \leq b)$ | |
| <i>GELT, a, b</i> | $(a \leq x < b)$ | |
| <i>GELE, a, b</i> | $(a \leq x \leq b)$ | |
| <i>LTGT, a, b</i> | $(x < a \text{ OR } x > b)$ | |
| <i>LTGE, a, b</i> | $(x < a \text{ OR } x \geq b)$ | |
| <i>LEGT, a, b</i> | $(x \leq a \text{ OR } x > b)$ | |
| <i>LEGE, a, b</i> | $(x \leq a \text{ OR } x \geq b)$ | |

a és b a mező típusával egyező konstansok, melyekre igaz, hogy $a \leq b$

LREAL /longreal/ típusu mezőre csak szimpla pontosságú *REAL* korlát írható elő. Ez 7 jegy pontosságot jelent. *CHAR* típusu mező esetén a konstans lehet egy egy karakterből álló szöveg /' jelek közé zárva adandó meg!/, vagy egy egy byte méretű integer is.

STRING típus esetén a kisebb, nagyobb relációt a rendszer az abc szerinti sorbarendezés szerint értelmezi. A szövegkonstans mérete nem lehet nagyobb, mint a mező mérete, de maximum 4 karakter /még ha a mező mérete nagyobb is/. Szövegkonstansot balra igazítva értelmezzük /space a legkisebb értékü/.

REKORD

Rekordtípus definiáló utasítás.

Általános alakja:

*név=RECORD/'hosszu név', elérési mód, egyszeres mezők
listája, többszörös mezők listája;*

- *név* egy változónév. Csak eddig még nem használt változónév lehet.
- *hosszu név* max. 24 karakteres szöveg /értelmezése azonos, mint FIELD-nél/
- *elérési mód*

$$\left. \begin{array}{l} SQ \\ DIRECT, \text{tábla méret} \\ CALC, \text{tábla méret[, RUTIN, rutinnév]} \\ FUZZY \end{array} \right\}$$

SQ - szekvenciális elérési mód /még nincs implementálva/. *DIRECT* és *CALC* elérési módnál meg kell határozni, hogy hány rekord-előfordulás lesz maximum, mert a rekordelőfordulás-táblát ennek megfelelően kell inicializálni. Mindkét elérési módnál a rekord címét a rekord egyik mezőjéből az u.n. "azonosító mező"ből számítja ki egy rutin. A különbség a két elérési mód között, hogy a *DIRECT* elérés esetén egy-egyszeres a leképezés, vagyis egy címre egy és csakis egy rekord képezhető le. *CALC* /hash/ elérési módnál egy címre több leképezés is eshet,

ilyenkor a rendszer gondoskodik arról, hogy a rekord egy üres helyre kerüljön. *CALC* elérés esetén előírható a user által irt hash-rutin használata is. Ha ez nincs külön feltüntetve, úgy a rendszer a saját rutinját használja. *FUZZY* elérési mód egy ál-szekvenciális, vagy lista-szerinti elérési mód. Itt nem kell előre közölni, hogy hány rekordelőfordulás lesz, a rekordok címét a rendszer önállóan határozza meg. Ilyen elérésű rekordokra legalább egy rendezési kritériumot elő kell írni.

- *egyszeres mezők listája*

Egyszeres mező az, amelynek *db*-ja = 1. A lista ilyen mezők azonosítóiból áll. Az azonosító helyett zárójelbe téve állhat a teljes meződefiníció is. Az "azonosító mező" elé az *IDENT* módosítószót kell írni. Az azonosítómező automatikusan kiemelt mező lesz. Ha a rekord *CALC* vagy *DIRECT* elérésű, kötelező, hogy legyen azonosítómező, ez azonban nem lehet "mutató mező". A listában szerepelhetnek u.n. "mutató mezők" is /definíciójukban szerepel a *COUNT* módosító/, ezek azonosítóját követi egy "!" jel és egy többszörös mező azonosítója. Itt határozzuk meg, hogy a mutatómező melyik többszörös mező számosságát mutatja. A többszörös mezőnek szerepelnie kell a többszörös mezők listájában.

- többszörös mezők listája

Többszörös mező az, melynek *db*-ja > 1 .

Ha a rekordban többszörös mező szerepel, akkor az változó méretű rekord lesz.

/Tehát az egyes rekordelőfordulásokban különböző számúszor fordulhat elő ez a mező/. Többszörös mező nem lehet azonosító mező és nem lehet rendezési kulcs. /Nem lehet érték szerinti rendezést előírni a rekord többszörös mezeje szerint./

ORDER

Rendezési kritérium definiáló utasítás.

Általános alakja:

név=ORDER/'*hosszu név*',*rekord név*,*rendezési mód*;

- *név* egy változónév. Előzőleg ORDER-nek definiált név újra definiálható.
- *hosszu név* max. 24 karakteres szöveg /lásd FIELD/
- *rekord név* a rekord azonosítója, amire a rendezési kritérium vonatkozik /SQ elérésü rekordra nem lehet rendezést előírni/.

- *rendezési mód*

$$\left. \begin{array}{l} \text{FIRST} \\ \text{LAST} \\ \text{BEFORE} \\ \text{AFTER} \\ \text{KEY}(\{ \begin{array}{l} \text{INCR} \\ \text{DECR} \end{array} \}, \text{mező azonosító})_1^k \end{array} \right\}$$

FIRST ill.LAST rendezési mód jelentése, hogy a létesülő új rekordot legelső, ill. legutolsó helyre kell beilleszteni a lista /létesítési sorrend/. A rendszer ezt a két rendezési módot tudja a legegyszerűbben kezelni, tehát ha más szempont nincs, ilyen rendezést írjunk elő. /Pl. minden FUZZY típusú rekordra kötelező előírni legalább egy rendezési kritériumot, de ha nekünk nincs különösebb szükségünk rá, úgy ilyen rendezést írjunk elő./

BEFORE ill. *AFTER* rendezési mód jelentése, hogy a létesülő új rekordot a "kurrens" rekord elé, ill. mögé kell a listába beilleszteni. Ha ilyen rendezés van előírva, úgy az új rekord létesítése előtt a kurrens rekord mutatóját rá kell állítani valamelyik rekordra /kivéve, ha még egy sincs/. Ezt a pointer-állítást elvégezheti a felhasználó valamely speciális kereső ill. rendező rutinja, vagy a felhasználói rendszer előélete folyamán már beállítódott, és így ezt az információt lehet felhasználni a sorrend meghatározására.

KEY rendezési mód azt jelenti, hogy a rekordot valamely mezőjük ill. azok értéke szerinti sorrendbe kell rendezni. A több mező szerinti sorbarendezés hierarchikus besorolást jelent, vagyis ha az első mezők egyformák, akkor a második szerint, ha azok is egyformák akkor a harmadik szerint és így tovább. Definiálni kell, hogy az egyes mezők szerint növekvő vagy csökkenő sorrendet kívánunk-e megvalósítani /*INCR* ill. *DECR*/.

Rendezési kulcs csak egyszeres mező lehet. Természetesen a mezőnek az adott rekordhoz kell tartoznia.

SET

Halmaztípus definiáló utasítás.

Általános alakja:

név=SET/'*hosszu név*', *rendezési mód*, *reprezentáció*,
OWNER, *owner lista*, MEMBER, *member lista*;

- *név* egy változónév, melyet eddig még nem használtunk
- *hosszu név* max. 24 karakteres szöveg
- *rendezési mód*

$$\left. \begin{array}{l} FIRST \\ LAST \\ BEFORE \\ AFTER \\ KEY, (\{ \begin{array}{l} INCR \\ DECR \end{array} \}, \text{mező típus})_1^k \end{array} \right\}$$

FIRST ill. *LAST* jelentése, hogy a halmaz-előfordulásba felveendő új tagot a legelső, ill. legutolsó helyre kell behelyezni.

A rendszer itt is ezt a rendezési módot tudja legegyszerűbben megvalósítani /hasonlóan a rendezési kritériumhoz/. Ha számunkra közömbös a tagok sorrendje, akkor ezek egyikét válasszuk.

BEFORE ill. *AFTER* jelentése, hogy a halmaz-előfordulásba felveendő új tagot a "kurrens" member elé ill. mögé kell felfűzni. Hasonlóan a rendezési kritériumhoz, a kurrens rekordot ill. kurrens membert a halmazba való felvétel előtt be kell állítani.

KEY rendezési mód jelentése, hogy a halmazelőfordulás összes tagrekordját valamely mező vagy mezők értéke szerint kell sorbarendezni. Mivel a halmazelőfordulásban lehetnek különböző típusu tagrekordok /különböző felépítésűek/, ezért a rendezési kulcsot két részben kell meghatározni: először definiálandó a rendezési kulcsok típusa és hogy ezek szerinti növekvő vagy csökkenő sorrendet kívánunk-e fenntartani. Másodszor pedig a member rekordtípusok felsorolásánál meg kell határozni, hogy melyik mező vagy mezők szerint kell rendezni. Az egyes mezők típusának meg kell egyeznie a rendezési kulcsok típusával, amit az első részben definiáltunk, továbbá minden tagtípushoz a rendezési kulcsok számának megfelelő számú mezőt kell felsorolni. Itt is csak egyszeres mező lehet rendezési kulcs. A rendezési kulcsnak kinevezett mezők automatikusan kiemelt mezők lesznek.

- *reprezentáció*

$$\left. \begin{array}{l} \text{ONEWAY} \\ \text{TWOWAY } [, \text{HEADED}] \\ \text{ARRAY} \end{array} \right\}$$

A halmazkapcsolat megvalósítási módjának meghatározása. /ARRAY még nincs implementálva!/

ONEWAY, ill. *TWOWAY* egy-, ill. kétirányu láncolást jelent. Az alkalmazástól függ, hogy mi szorít minket inkább: a hely, vagy az idő. Egyutas listát akkor célszerű hasz-

nálni, ha a halmazelőfordulások elég statikusan viselkednek, vagyis ritka a törlési és beszúrási művelet, különösen ha nem a lánc elejére, vagy végére vonatkozik. /Ilyenkor ugyanis meg kell keresni az előző rekordot és ez elég hosszú időt igényel, ha sok tag van./ A *HEADED* módosítószóval előírható, hogy minden tagrekordban legyen elhelyezve egy pointer a hozzá tartozó tulajdonos rekordra. Ez szintén gyorsítja a futást, ha gyakori az olyan jellegű művelet, melynél erre szükség van. /Valamely kereséssel megfogunk egy rekordot és azt kurrens memberré akarjuk tenni egy halmaztípusban. Ilyenkor a rendszernek meg kell keresnie a hozzá tartozó owner rekordot is. Ezt a műveletet gyorsítja meg nagyon ez a pointer./

- *owner lista* A halmaztípus lehetséges owner rekordtípusainak felsorolása.

- *member lista*

$(\{ \begin{matrix} AUT \\ NOAUT \end{matrix} \} , \text{rekord azonosító} [, (\text{mező azonosító}]_1^k)_1^n$

A halmaztípus lehetséges member rekordtípusainak felsorolása.

Az *AUT* módosító használata azt jelenti, hogy az ilyen típusu rekordelőfordulásokat születésük pillanatában fel kell venni egy ilyen típusu halmazelőfordulásba tagnak is. /Az adott pillanatban kurrens halmazelőfordulásba./ *NOAUT* esetén a programozónak erről külön kell gondoskodnia.

KEY típusu rendezési mód esetén a rekord azonosítót követik a rendezési kulcsok szerepét betöltő mezők azonosítói.

FINISH

Adatbázis leírás végét jelző utasítás.

IRODALOMJEGYZÉK

- [1] Bródy F.: KALAP - Intézeti kiadványok, 1978
- [2] János J.: COMPLETEER rendszer - Belső tanulmány, 1977
- [3] Gallai I.: DOST - Felhasználói kézikönyv, 1977

A DDL-ben használt speciális jelentésű karakterek:

() / , ; = !

A DDL-ben alkalmazott alapszavak:

| | |
|---------------|---------------|
| <i>AFTER</i> | <i>KEY</i> |
| <i>ARRAY</i> | <i>LAST</i> |
| <i>AUT</i> | <i>LE</i> |
| <i>BEFORE</i> | <i>LEGE</i> |
| <i>CALC</i> | <i>LEGT</i> |
| <i>CHAR</i> | <i>LINT</i> |
| <i>COUNT</i> | <i>LREAL</i> |
| <i>DECR</i> | <i>LT</i> |
| <i>DIRECT</i> | <i>LTGE</i> |
| <i>F</i> | <i>LTGT</i> |
| <i>FIELD</i> | <i>MEMBER</i> |
| <i>FINISH</i> | <i>NOAUT</i> |
| <i>FIRST</i> | <i>O</i> |
| <i>FUZZY</i> | <i>ONEWAY</i> |
| <i>GE</i> | <i>ORDER</i> |
| <i>GELE</i> | <i>OWNER</i> |
| <i>GELT</i> | <i>R</i> |
| <i>GT</i> | <i>REAL</i> |
| <i>GTLE</i> | <i>RECORD</i> |
| <i>GTLT</i> | <i>RUTIN</i> |
| <i>HEADED</i> | <i>S</i> |
| <i>IDENT</i> | <i>SET</i> |
| <i>INCR</i> | <i>SQ</i> |
| <i>INT</i> | <i>STRING</i> |
| | <i>TWOWAY</i> |

2. melléklet

A DDL-ben kiadott hibakódok

| | |
|-----|--|
| 100 | Buffer overflow |
| 110 | Kevés paraméter |
| 120 | Rendezési kód hiba |
| 122 | Hiányzik a reprezentáció módja /pl. ONEWAY/ |
| 124 | Hibás reprezentációs mód |
| 126 | "Headed" helyén értelmetlen paraméter |
| 130 | Rendezési kulcs hiba |
| 132 | Hiányzik a rendezési kulcs típus lista |
| 140 | Hibás típus a rendezési kulcs típus listában |
| 150 | "Owner" helyén értelmetlen paraméter |
| 151 | Owner-lista hiányzik |
| 152 | Owner-lista hibás |
| 153 | Owner-lista SQ-elérésű rekordot tartalmaz |
| 160 | "Member" helyén értelmetlen paraméter |
| 161 | "Aut", "Noaut" hiányzik vagy hibás |
| 500 | Felismerhetetlen mezőtípus |
| 501 | String-típusu mező mérete hiányzik |
| 502 | Többszörös mező nem emelhető ki |
| 503 | Számolómező csak int.vagy char. típusu lehet |
| 510 | Hibás integer |
| 517 | Számolómező csak egyszeres lehet |
| 520 | Hibás string |
| 555 | Byte-os integer nagyobb 127-nél |
| 560 | Számolómezőre nem írható elő ellenőrzés |
| 570 | Hibás ellenőrzési előírás |
| 577 | A korlát string hosszabb a definiáltnál |
| 580 | Alsó korlát nagyobb a felsőnél |
| 585 | A korlát string hosszabb 4 karakternél |
| 590 | Hibás ellenőrzési kód |
| 595 | A korlát és a mező típusa eltérő |

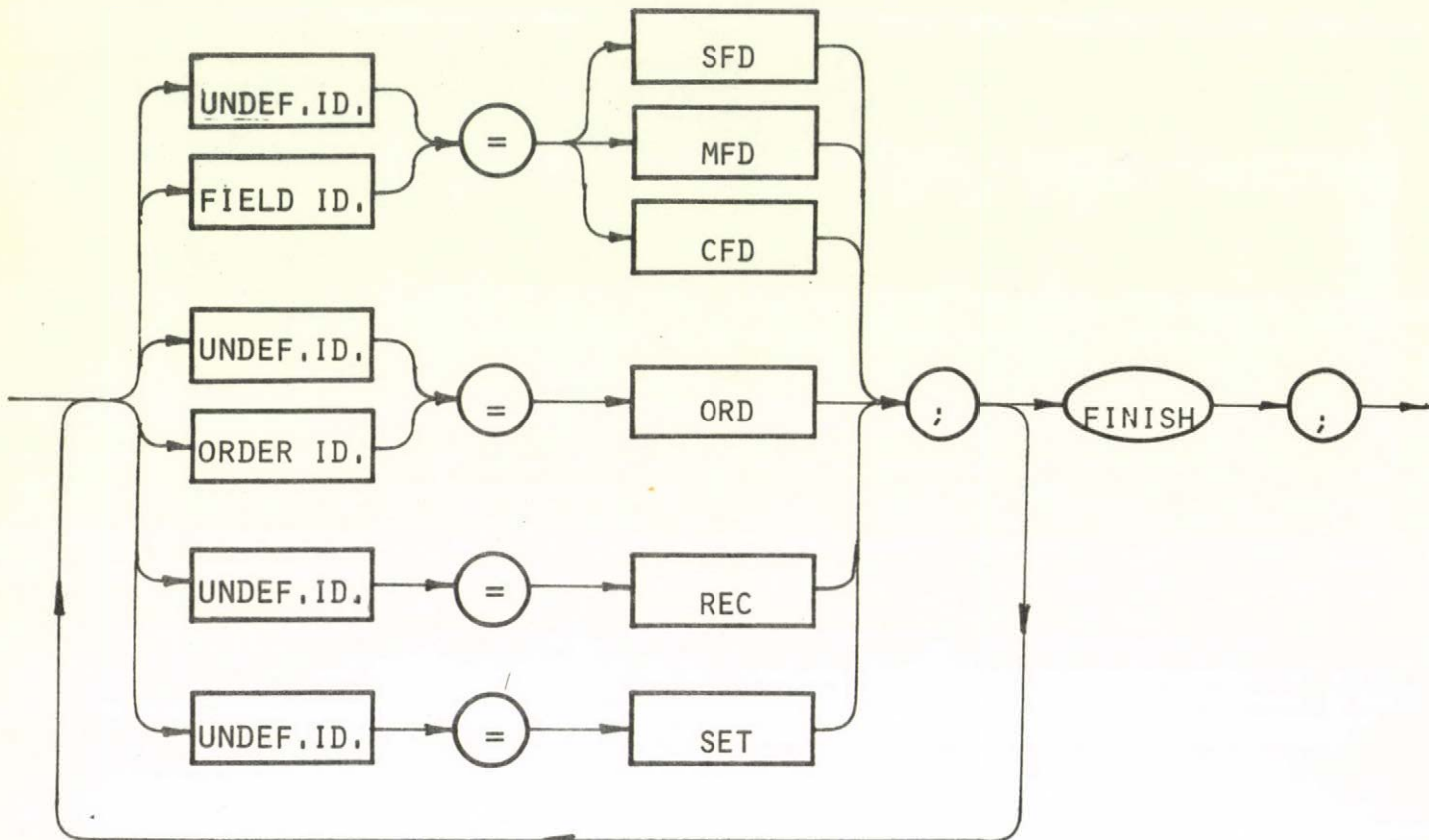
A DDL-ben kiadott hibakódok /folyt./

| | |
|------|---|
| 600 | Hibás elérési mód |
| 601 | Hash rutinnév hibás |
| 602 | Egyszeres mezők listájában idegen módosító |
| 603 | Azonosítómező nem lehet többszörös |
| 604 | Több azonosítómező van |
| 605 | Nincs azonosítómező |
| 606 | Mezők sorrendje hibás /többszöröst a végére!/ Számolómező hiba |
| 607 | |
| 610 | Mezőlista hibás |
| 701 | Rendezési kód után felesleges paraméter |
| 702 | INCR, DECR helyén hibás paraméter |
| 703 | Mező azonosító felismerhetetlen |
| 704 | Hiányzik a rendezési kulcs lista |
| 705 | Rekord azonosító felismerhetetlen |
| 706 | SQ elérésű rekordra nem írható elő rendezés |
| 707 | Számolómező nem lehet rendezési kulcs |
| 708 | Idegen mező a rendezési kulcs listában |
| 770 | Számolómező hiba |
| 771 | Számolómező típus hiba |
| 776 | Jobb operandus nem többszörös mező |
| 800 | Az egyenlőség jobb operandusa hibás |
| 810 | Az egyenlőség baloldala nem azonosító |
| 820 | Foglalt azonosító |
| 1100 | FUZZY elérésű rekordra nincs ellenőrzés előírva |
| 1210 | Memberlista hiba |
| 1215 | Memberlistában SQ elérésű rekord |
| 1230 | Többszörös mező a rendezési kulcsok között |
| 1240 | Nem az előirt típusu a rendezési kulcs |
| 1250 | Az előirtnál több a kulcs |
| 1260 | Az előirtnál kevesebb a kulcs |

3. melléklet

ÁBRÁS SZINTAXISLEIRÁS

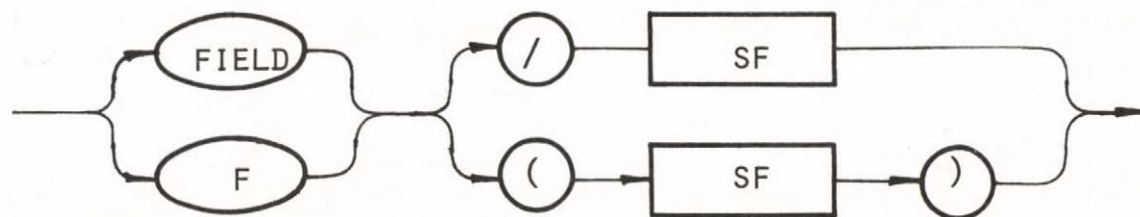
DDL



1. ábra

| | |
|-----------|---|
| FINISH | Záró alapszó |
| UNDEF.ID. | Definiálatlan azonosító |
| FIELD ID. | Mező azonosító |
| ORDER ID. | Rendezési kritérium azonosító |
| SFD | Egyszeres mező definíció /Lásd 2. ábra/ |
| MFD | Többszörös mező definíció /Lásd 3. ábra/ |
| CFD | Számoló mező definíció /Lásd 4. ábra/ |
| ORD | Rendezési kritérium definíció /Lásd 5. ábra/ |
| REC | Rekord definíció /Lásd 6. ábra/ |
| SET | Halmaz definíció /Lásd 7. ábra/ |

SFD



2. ábra

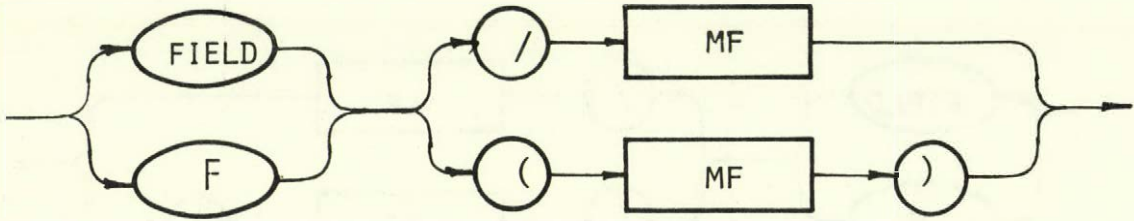
FIELD
F

alapszó
alapszó

SF

egyszeres mező
/Lásd 8. ábra/

MFD



3. ábra

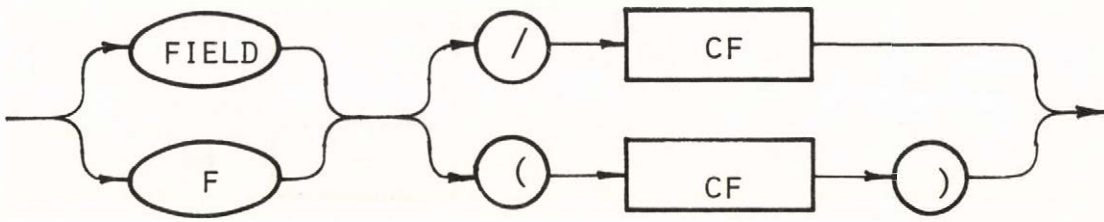
FIELD
F

alapszó
alapszó

MF

többszörös mező
/Lásd 9. ábra/

CFD



4. ábra

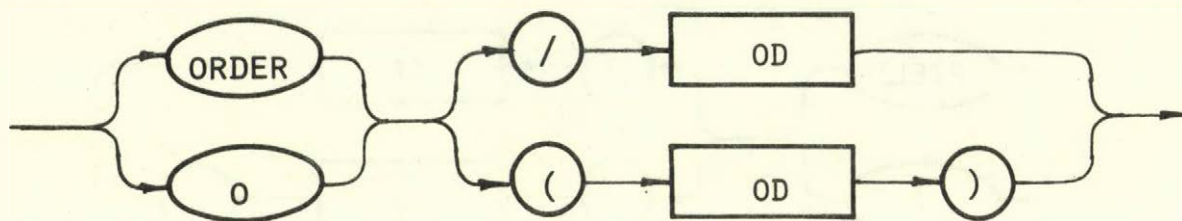
FIELD
F

alapszó
alapszó

CF

számoló mező
/Lásd 10. ábra/

ORD



5. ábra

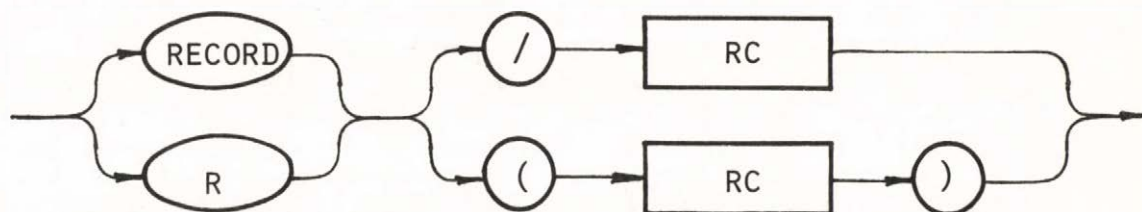
ORDER
0

alapszó
alapszó

OD

rendezési kritérium definíció
/Lásd 11. ábra/

REC



6. ábra

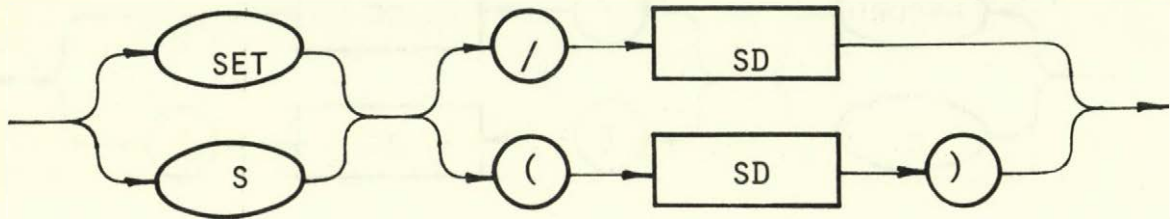
RECORD
R

alapszó
alapszó

RC

rekord definíció
/Lásd 12. ábra/

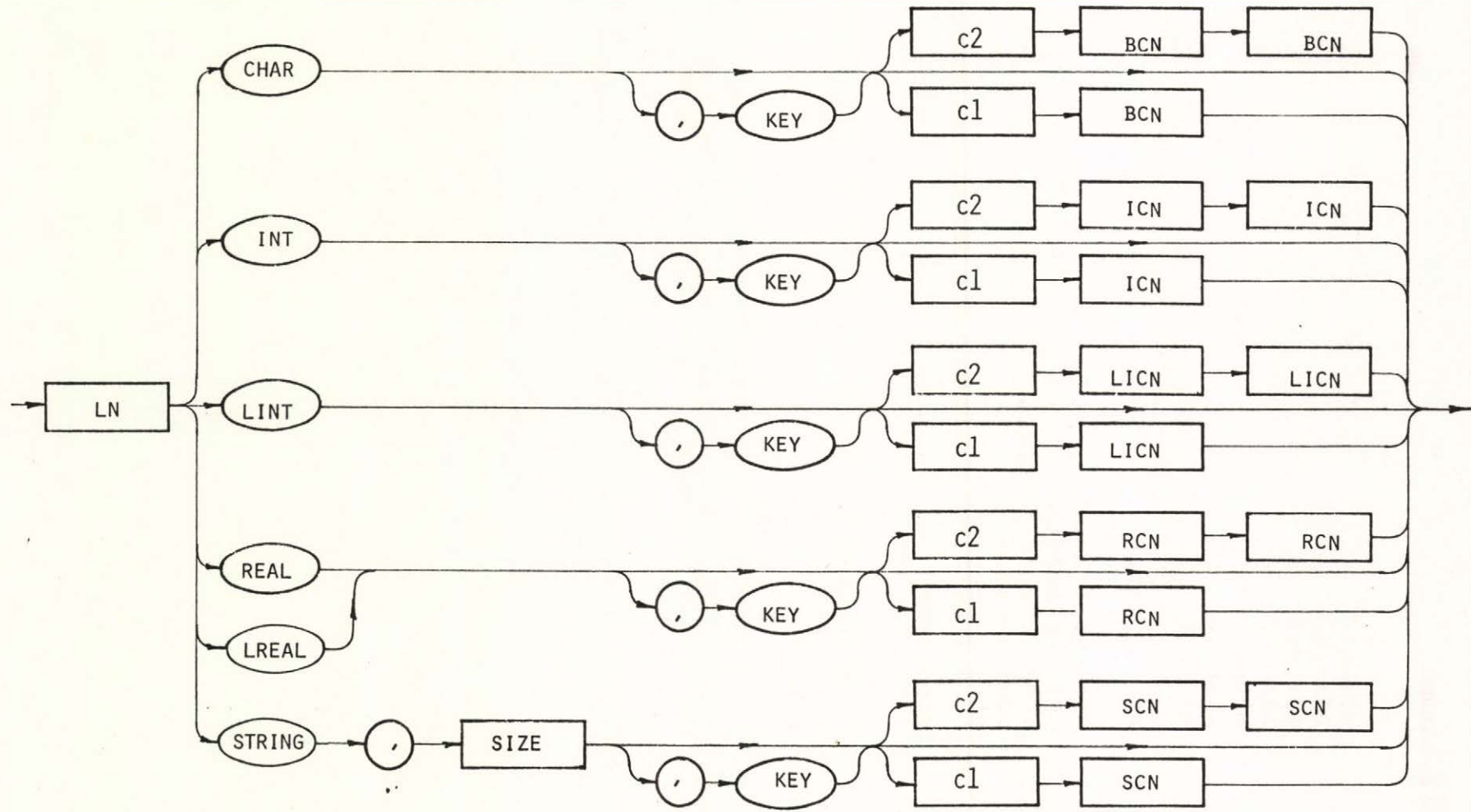
SET



7. ábra

| | |
|-----|-------------------------------------|
| SET | alapszó |
| S | alapszó |
| SD | halmaz definíció /Lásd 13. ábra/ |

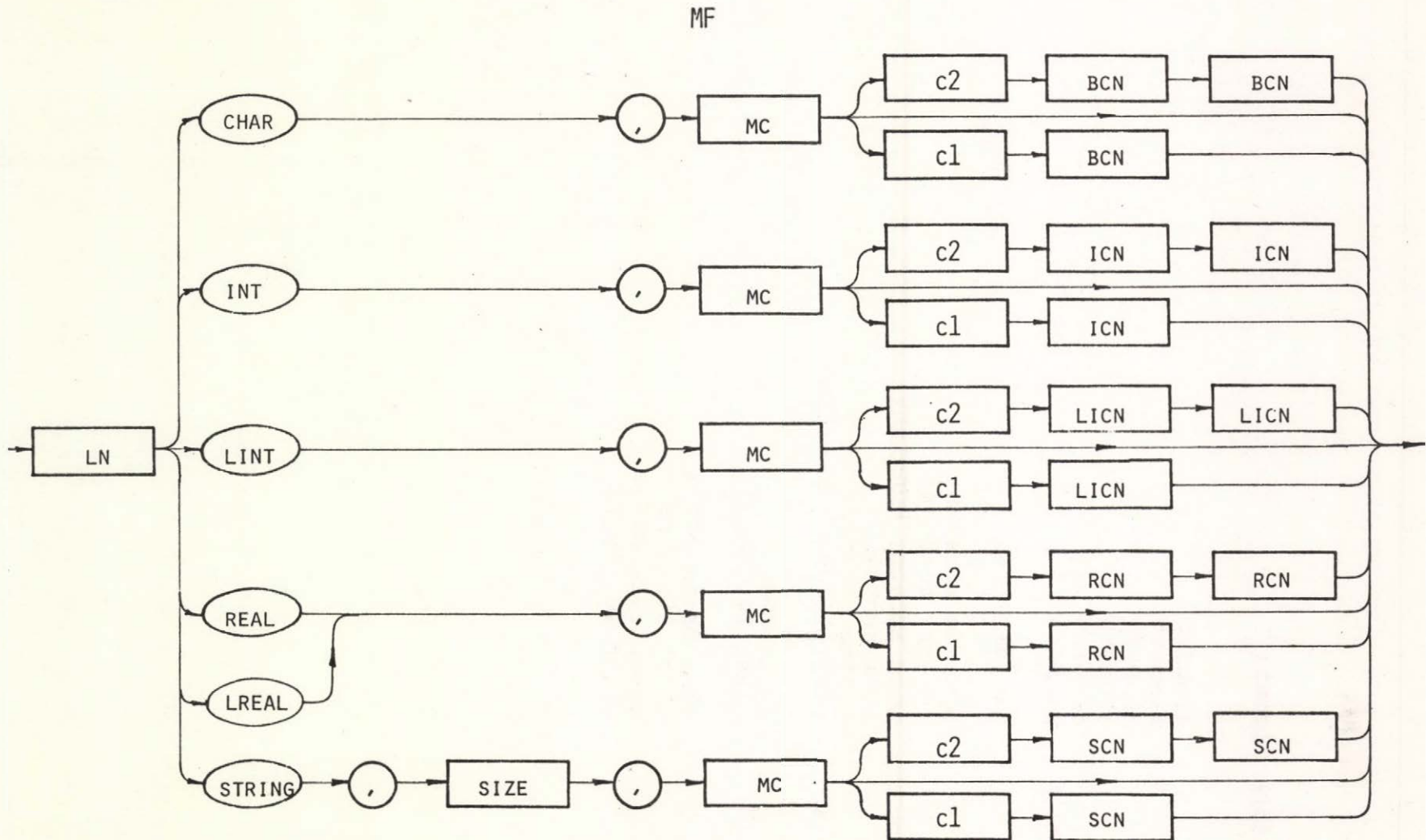
SF



8. ábra

A 8. ábra elemei:

| | |
|--------|-------------------------------------|
| CHAR | alapszó |
| INT | alapszó |
| LINT | alapszó |
| REAL | alapszó |
| LREAL | alapszó |
| STRING | alapszó |
| KEY | alapszó |
| SIZE | string méret |
| BCN | byte-os konstans |
| ICN | integer konstans |
| LICN | long integer konstans |
| RCN | real konstans |
| SCN | string konstans |
| LN | long name /lásd 14. ábra/ |
| C1 | ellenőrzés módja /lásd 15. ábra/ |
| C2 | ellenőrzés módja /lásd 16. ábra/ |

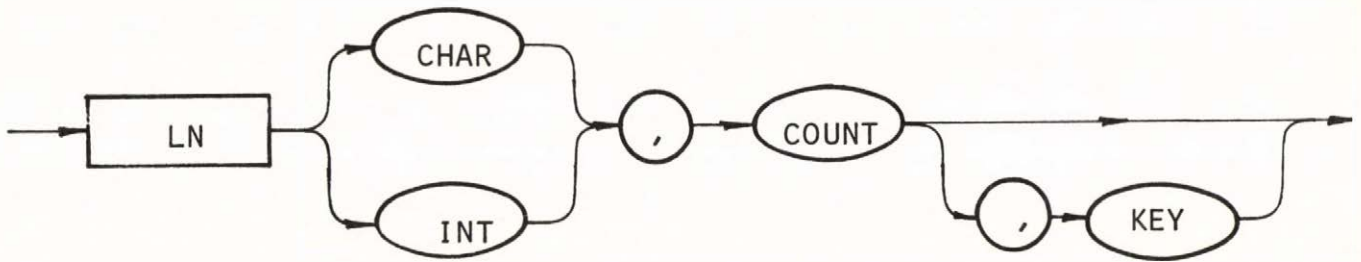


9. ábra

A 9. ábra elemei:

| | |
|--------|-------------------------------------|
| CHAR | alapszó |
| INT | alapszó |
| LINT | alapszó |
| REAL | alapszó |
| LREAL | alapszó |
| STRING | alapszó |
| SIZE | string méret |
| BCN | byte-os konstans |
| ICN | integer konstans |
| LICN | long integer konstans |
| RCN | real konstans |
| SCN | string konstans |
| MC | multiplikációs konstans |
| LN | long name /lásd 14. ábra/ |
| C1 | ellenőrzés módja /lásd 15. ábra/ |
| C2 | ellenőrzés módja /lásd 16. ábra/ |

CF

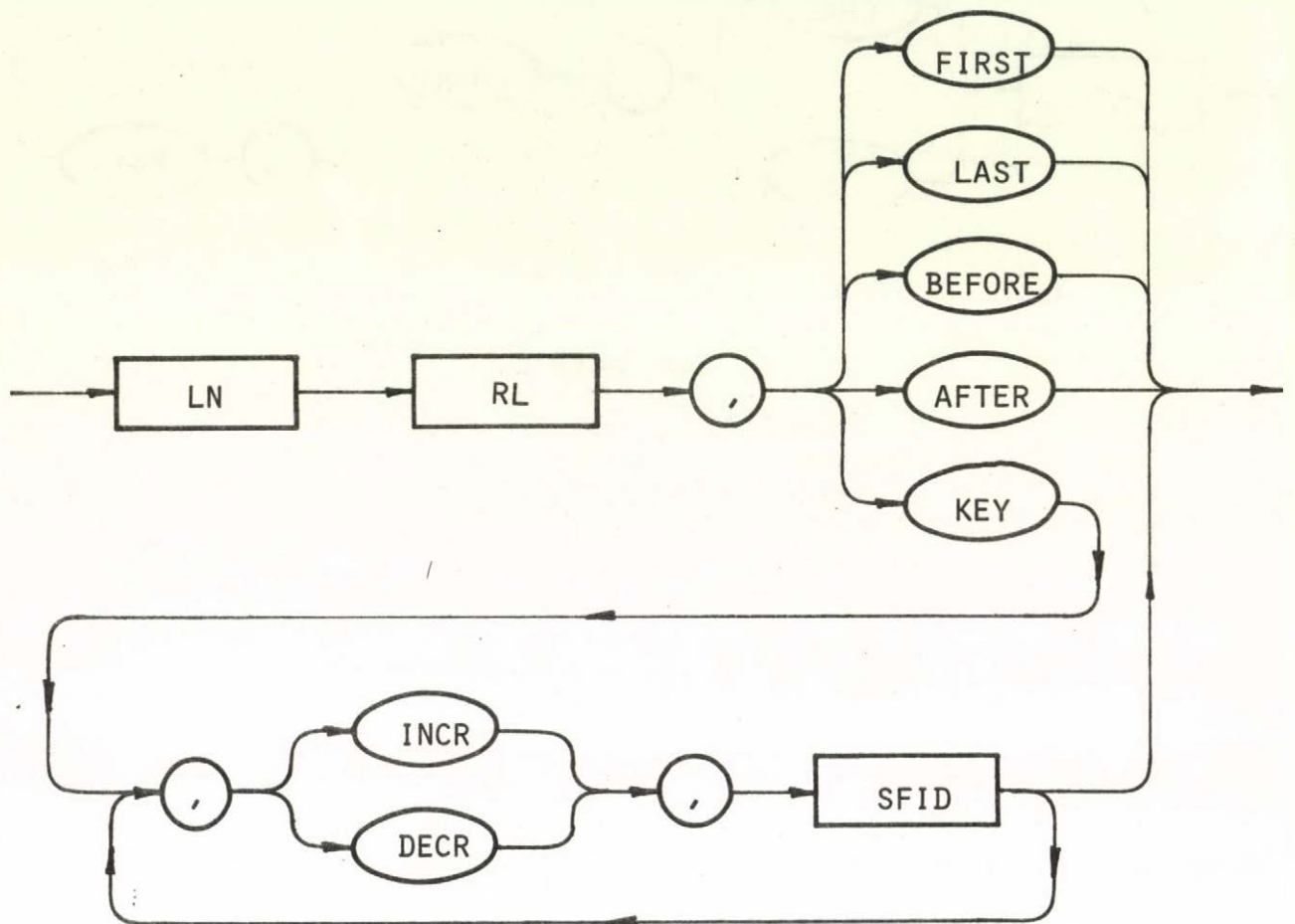


10. ábra

| | |
|-------|---------|
| INT | alapszó |
| CHAR | alapszó |
| KEY | alapszó |
| COUNT | alapszó |

| | |
|----|------------------------------|
| LN | long name /lásd 14. ábra/ |
|----|------------------------------|

OD



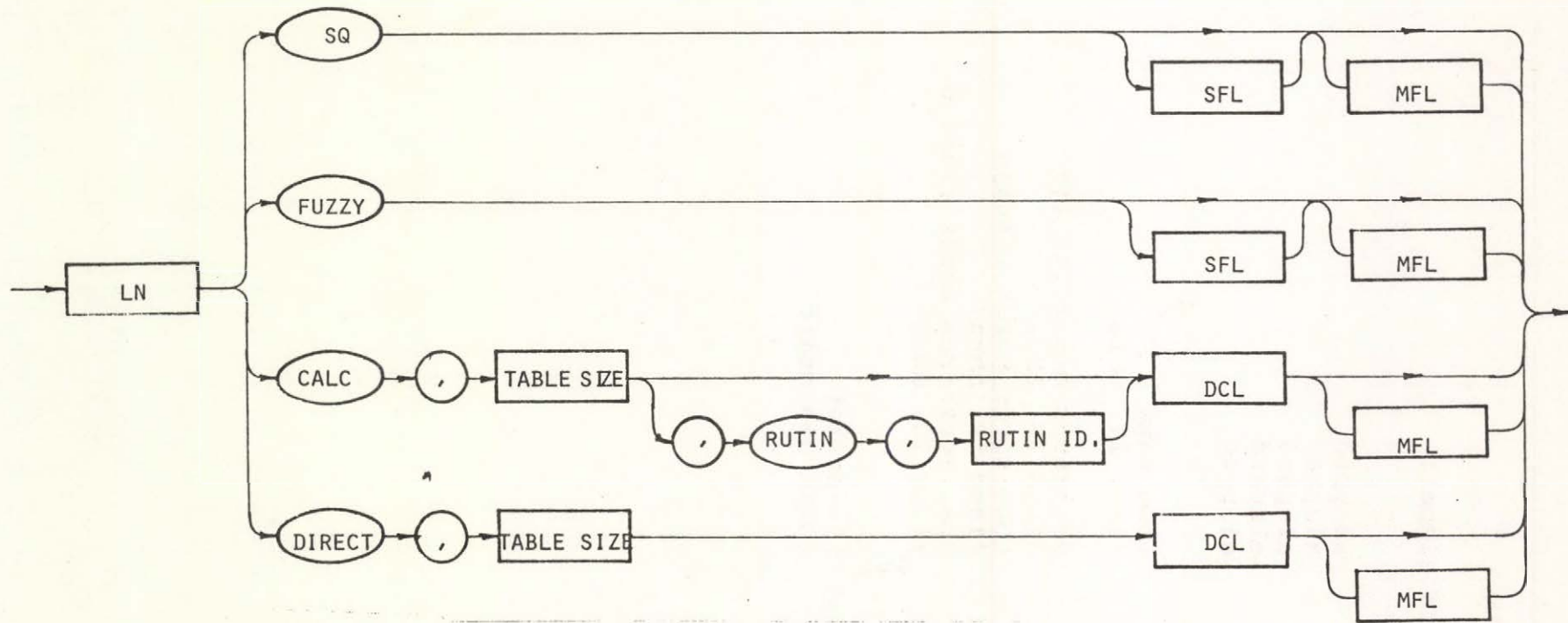
11. ábra

| | |
|--------|---------|
| FIRST | alapszó |
| LAST | alapszó |
| BEFORE | alapszó |
| AFTER | alapszó |
| KEY | alapszó |
| INCR | alapszó |
| DECR | alapszó |

| | |
|------|--------------------------|
| SFID | egyszeres mező azonosító |
|------|--------------------------|

| | | | |
|----|-----------------|----|-------------------|
| LN | long name | RL | rekord hivatkozás |
| | /lásd 14. ábra/ | | /lásd 17. ábra/ |

RC

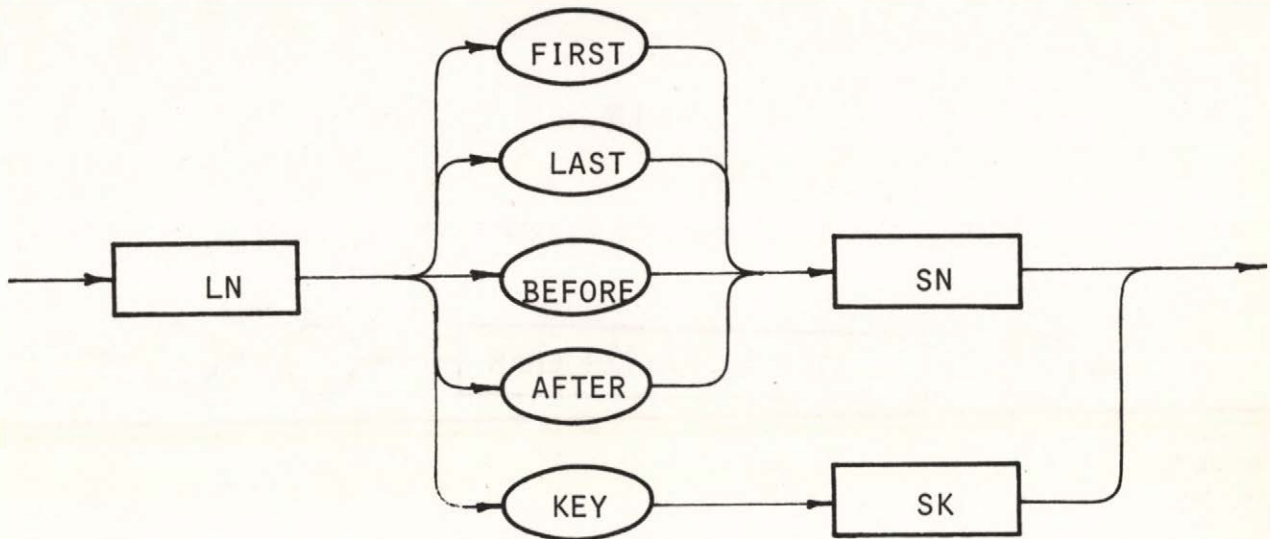


12. ábra

A 12. ábra elemei:

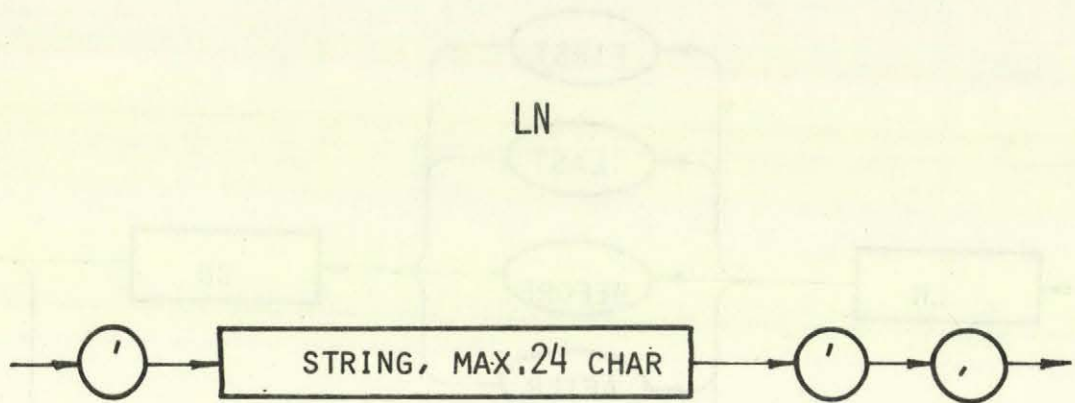
| | |
|------------|---|
| SQ | alapszó |
| FUZZY | alapszó |
| CALC | alapszó |
| DIRECT | alapszó |
| RUTIN | alapszó |
| LN | long name /lásd 14. ábra/ |
| SFL | egyszeres mezők listája /lásd 18. ábra/ |
| MFL | többszörös mezők listája /lásd 19. ábra/ |
| DCL | spec.egyszeres mezők listája /lásd 20. ábra/ |
| TABLE SIZE | táblaméret |
| RUTIN ID. | rutin azonosító |

SD



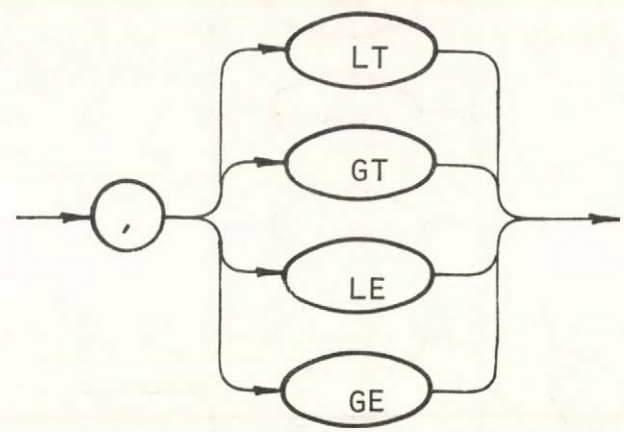
13. ábra

| | |
|--------|--|
| FIRST | alapszó |
| LAST | alapszó |
| BEFORE | alapszó |
| AFTER | alapszó |
| KEY | alapszó |
| LN | long name /lásd 14. ábra/ |
| SN | owner-member lista előirt rendezés esetén /lásd 21. ábra/ |
| SK | speciális owner-member lista kulcs szerinti rendezés esetén /lásd 22. ábra/ |



14. ábra

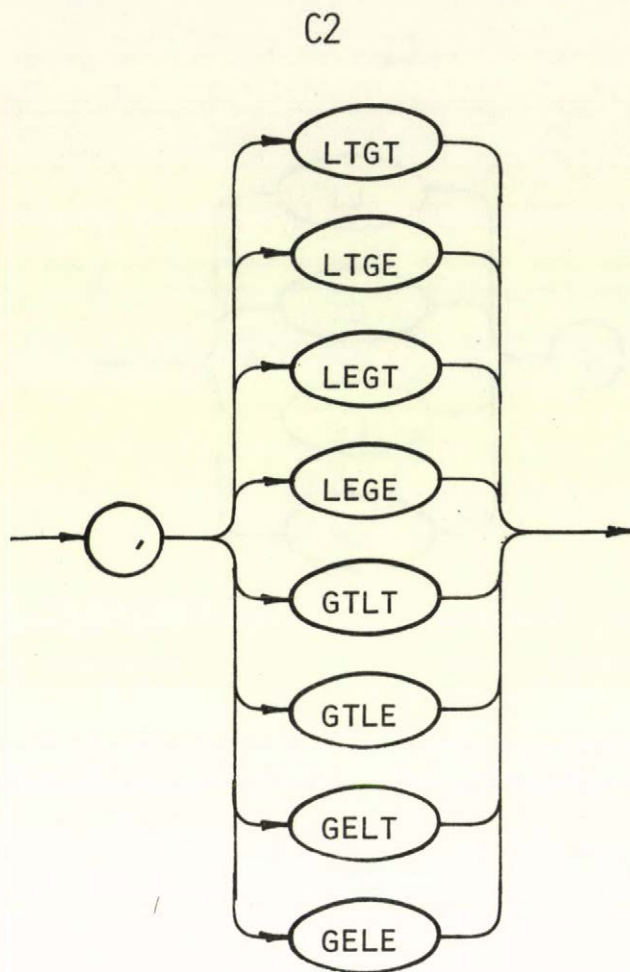
C1



15. ábra

LT
GT
LE
GE

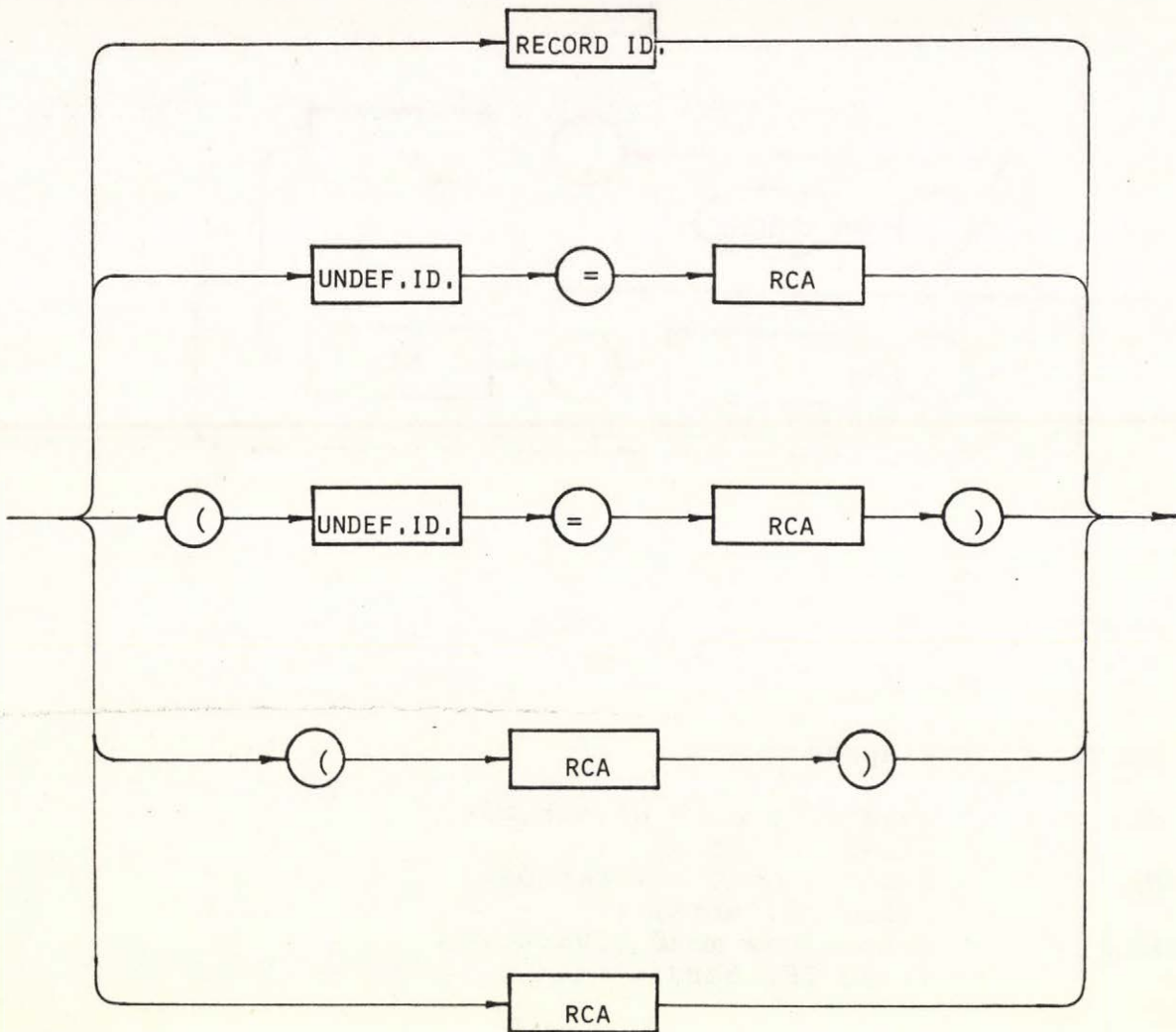
alapszó
alapszó
alapszó
alapszó



16. ábra

| | |
|------|---------|
| LTGT | alapszó |
| LTGE | alapszó |
| LEGT | alapszó |
| LEGE | alapszó |
| GTLT | alapszó |
| GTLE | alapszó |
| GELT | alapszó |
| GELE | alapszó |

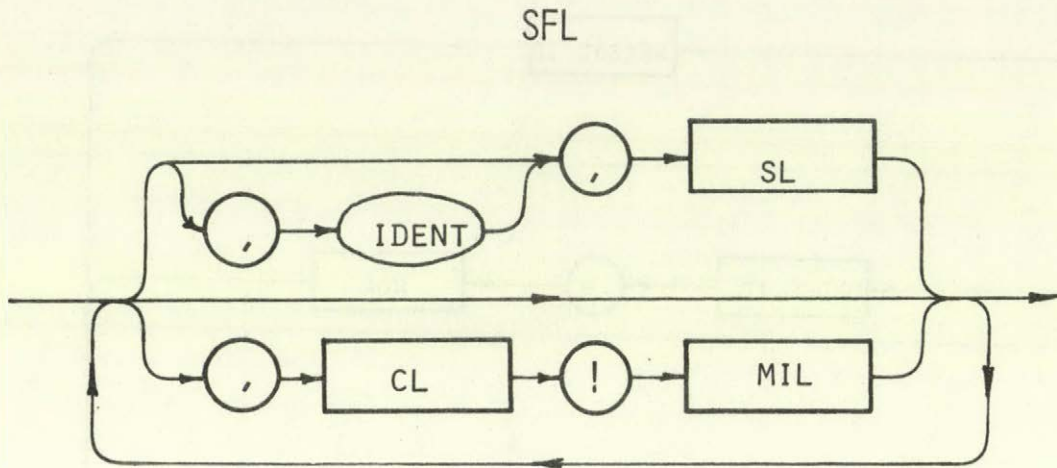
RL



17. ábra

RECORD ID. rekord azonosító
 UNDEF. ID. definiálatlan azonosító

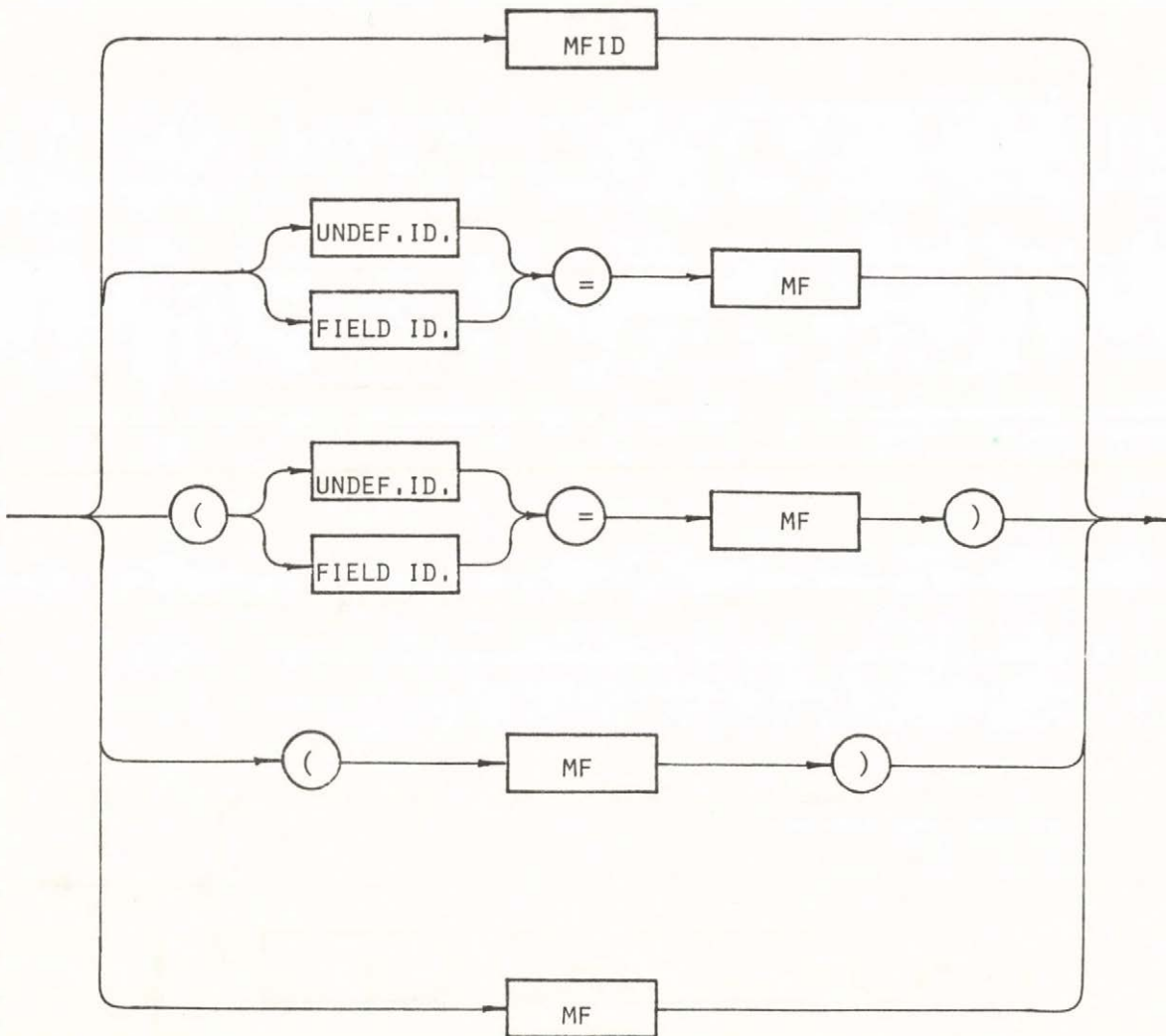
RCA rekord definíció speciális módja
 /lásd 23. ábra/



18. ábra

| | |
|-------|---|
| IDENT | alapszó |
| SL | egyszeres mező hivatkozás /lásd 24. ábra/ |
| CL | számoló mező hivatkozás /lásd 25. ábra/ |
| MIL | többszörös mező hivatkozás /lásd 26. ábra/ |

MFL

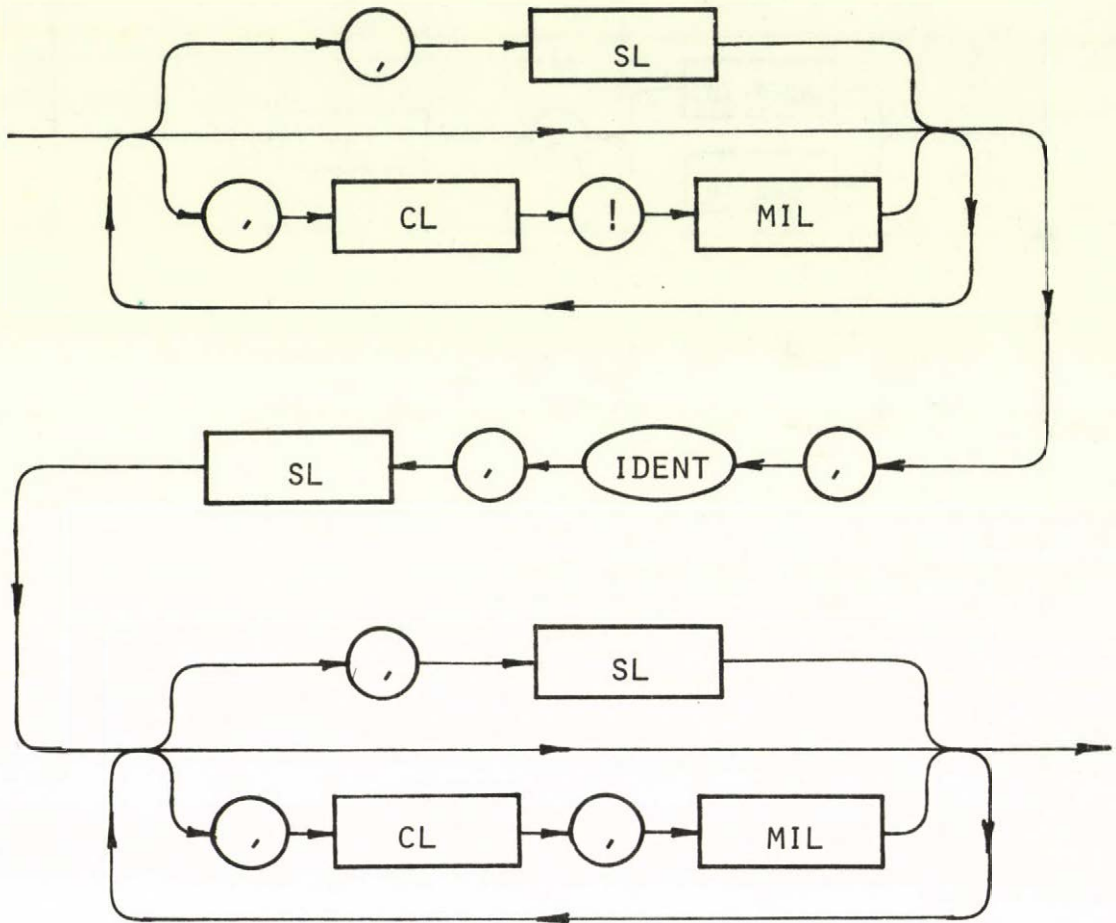


19. ábra

FIELD ID. mező azonosító
 UNDEF.ID. definiálatlan azonosító
 MFID többszörös mező azonosító

MF többszörös mező
 /lásd 9. ábra/

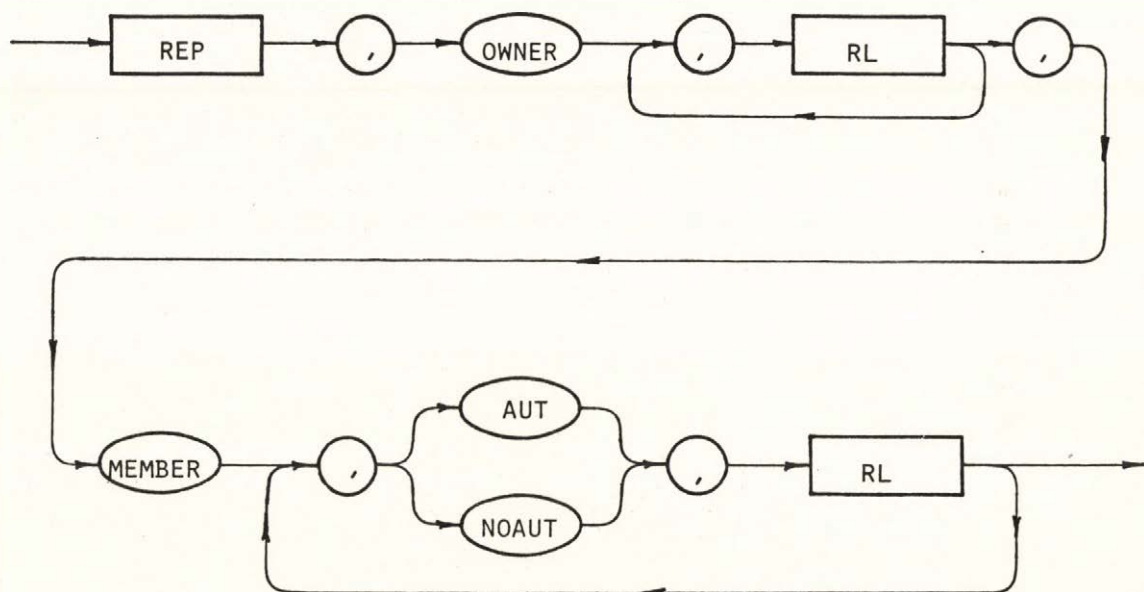
DCL



20. ábra

| | |
|-------|---|
| IDENT | alapszó |
| SL | egyszeres mező hivatkozás /lásd 24. ábra/ |
| CL | számoló mező hivatkozás /lásd 25. ábra/ |
| MIL | többszörös mező hivatkozás /lásd 26. ábra/ |

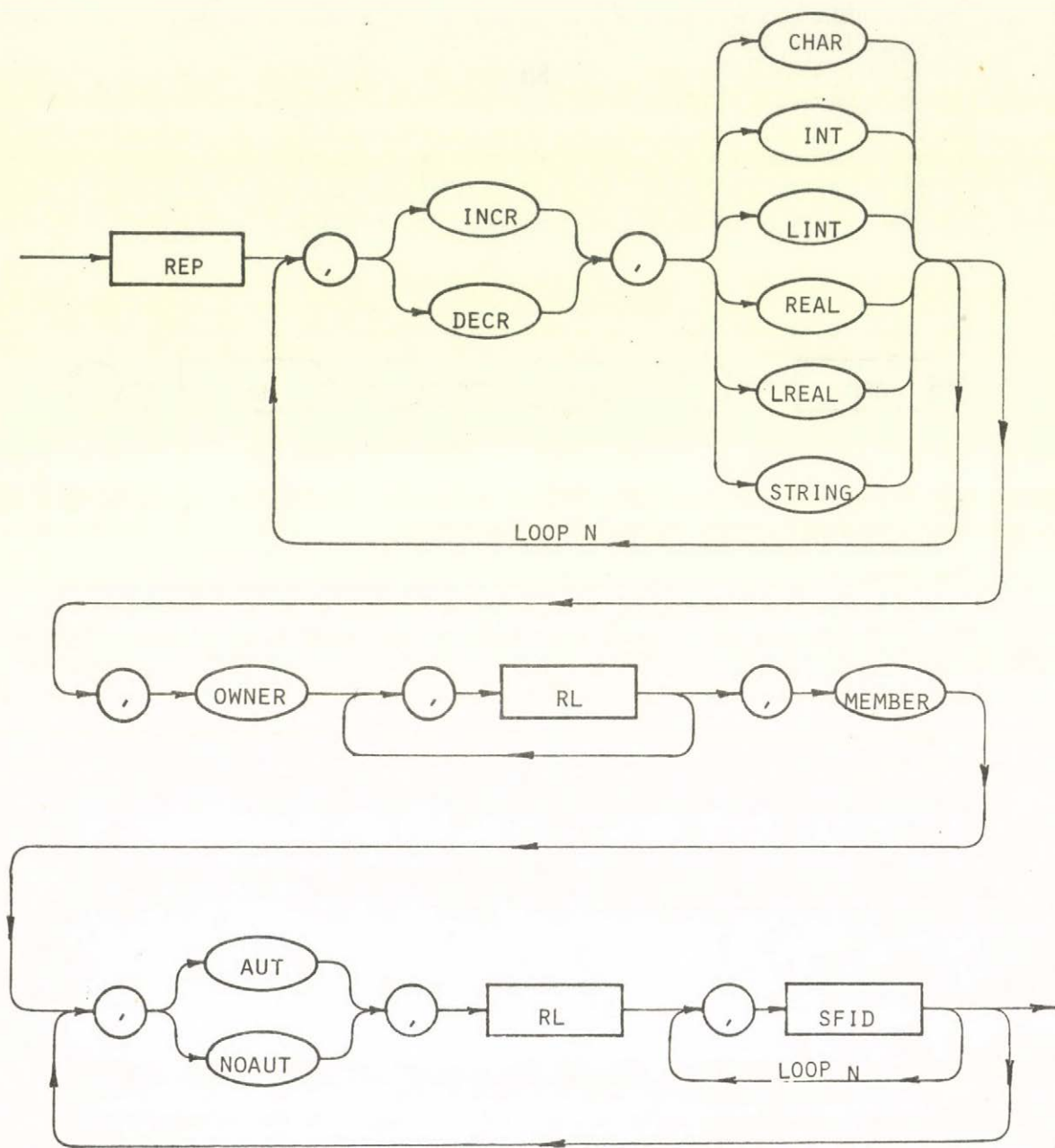
SN



21. ábra

| | |
|--------|--|
| OWNER | alapszó |
| MEMBER | alapszó |
| AUT | alapszó |
| NOAUT | alapszó |
| RL | rekord hivatkozás /lásd 17. ábra/ |
| REP | reprezentáció módja /lásd 27. ábra/ |

SK

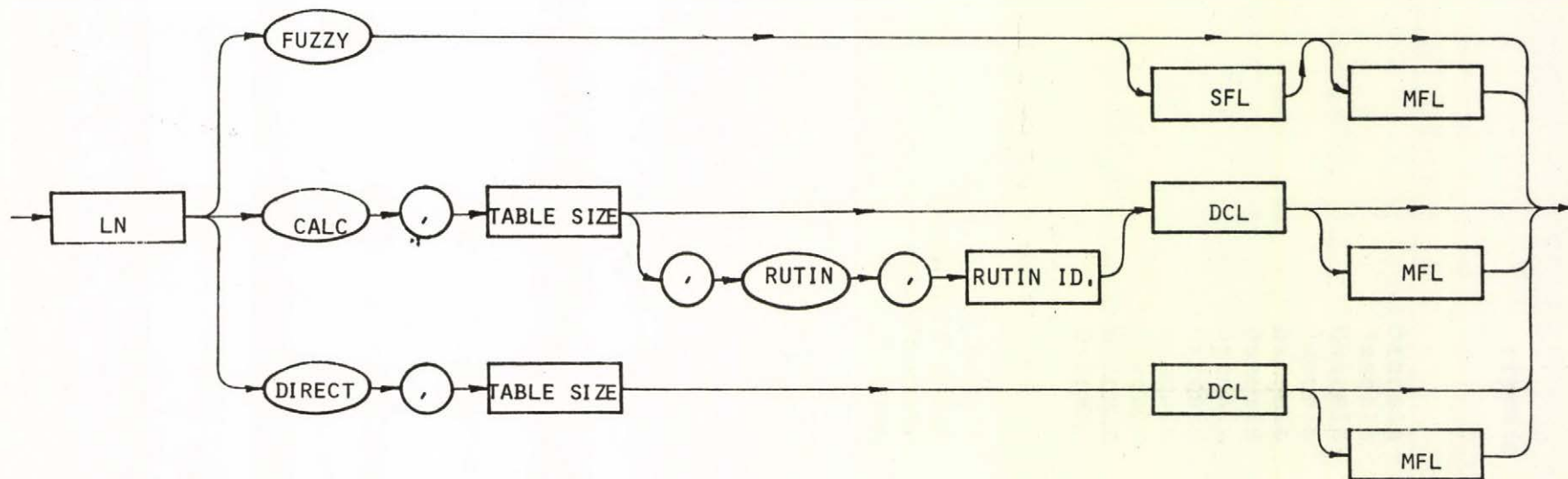


22. ábra

A 22. ábra elemei:

| | |
|--------|--|
| INCR | alapszó |
| DECR | alapszó |
| CHAR | alapszó |
| INT | alapszó |
| LINT | alapszó |
| REAL | alapszó |
| LREAL | alapszó |
| STRING | alapszó |
| OWNER | alapszó |
| MEMBER | alapszó |
| AUT | alapszó |
| NOAUT | alapszó |
| SFID | egyszeres mező azonosító |
| RL | rekord hivatkozás /lásd 17. ábra/ |
| REP | reprezentáció módja /lásd 27. ábra/ |

RCA

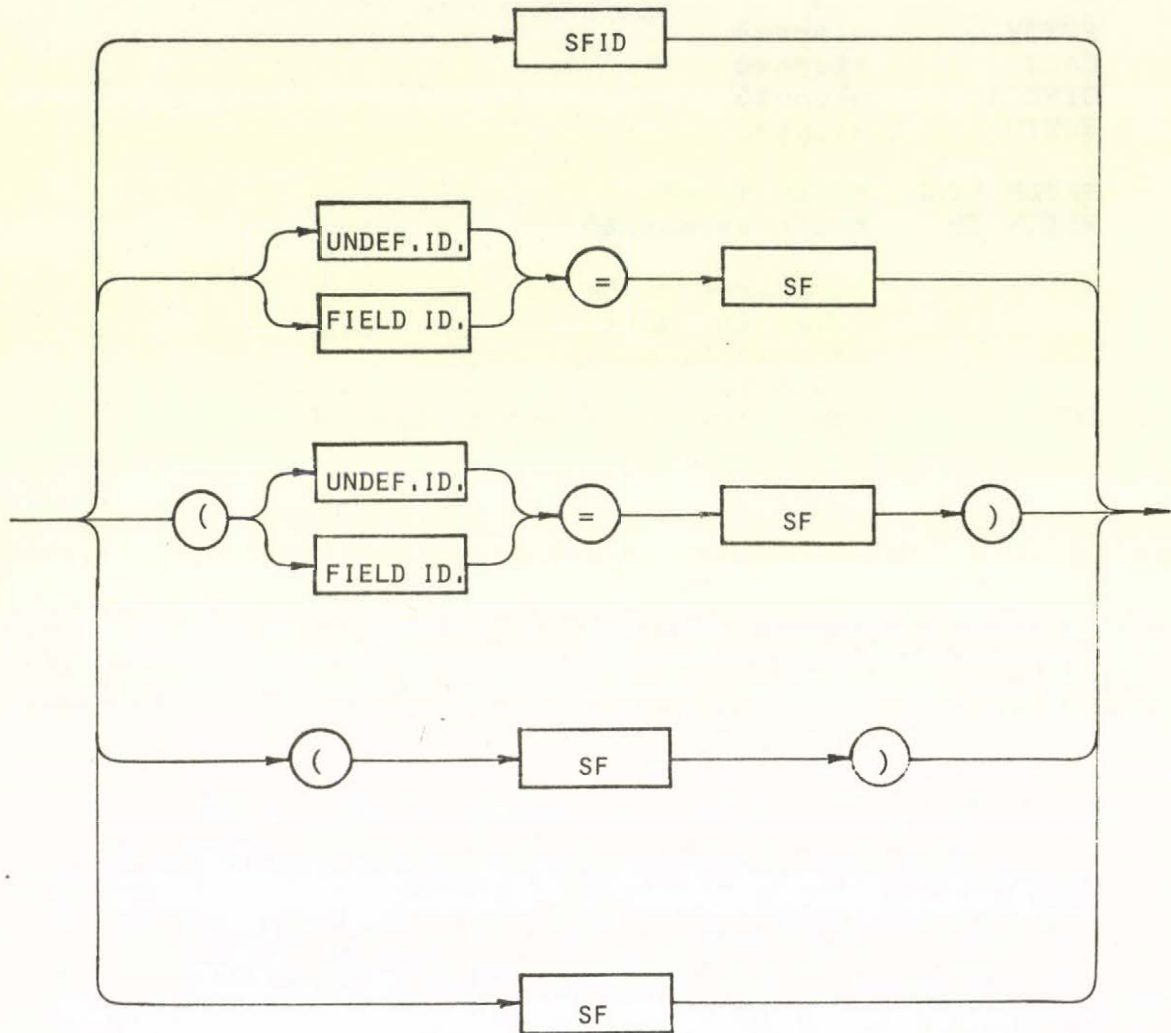


23. ábra

A 23. ábra elemei:

| | |
|------------|--|
| FUZZY | alapszó |
| CALC | alapszó |
| DIRECT | alapszó |
| RUTIN | alapszó |
| TABLE SIZE | tábla méret |
| RUTIN ID. | rutin azonosító |
| SFL | egyszeres mezők listája /lásd 18. ábra/ |
| MFL | többszörös mezők listája /lásd 19. ábra/ |
| DCL | spec. egyszeres mezők listája /lásd 20. ábra/ |

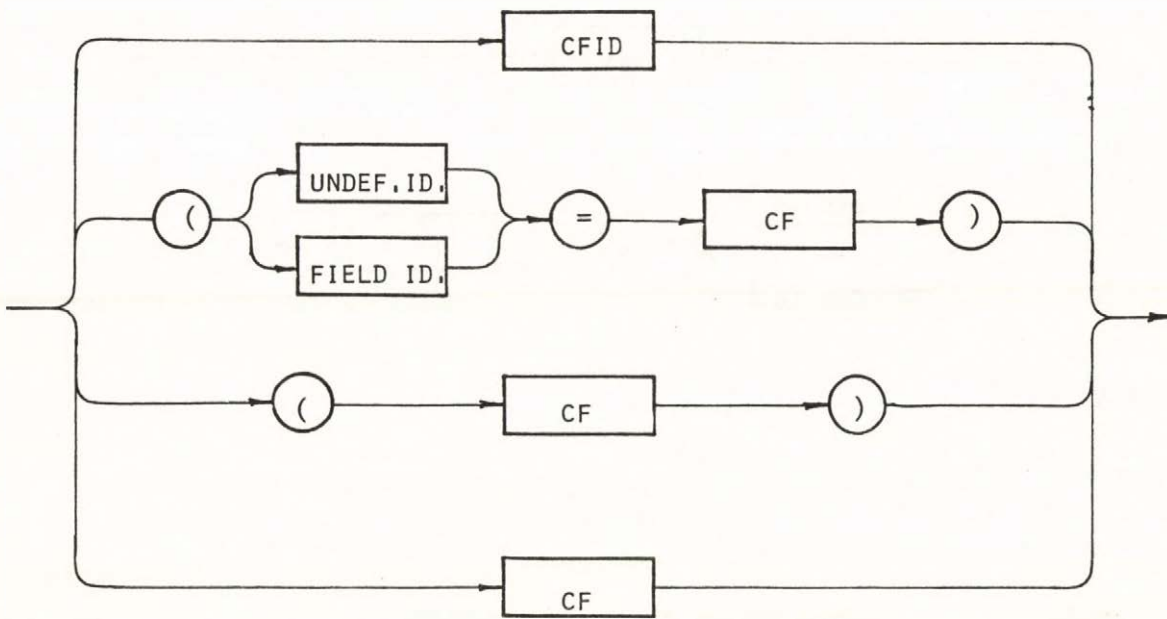
SL



24. ábra

| | |
|-----------|----------------------------------|
| SFID | egyszeres mező azonosító |
| UNDEF.ID. | definiálatlan azonosító |
| FIELD ID. | mező azonosító |
| SF | egyszeres mező /lásd 8. ábra/ |

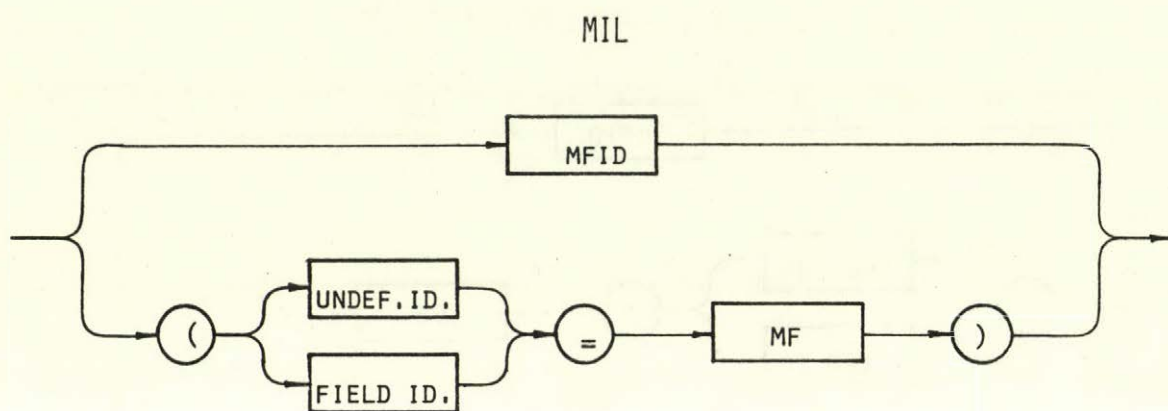
CL



25. ábra

CFID számoló mező azonosító
 UNDEF ID definiálatlan azonosító
 FIELD ID mező azonosító

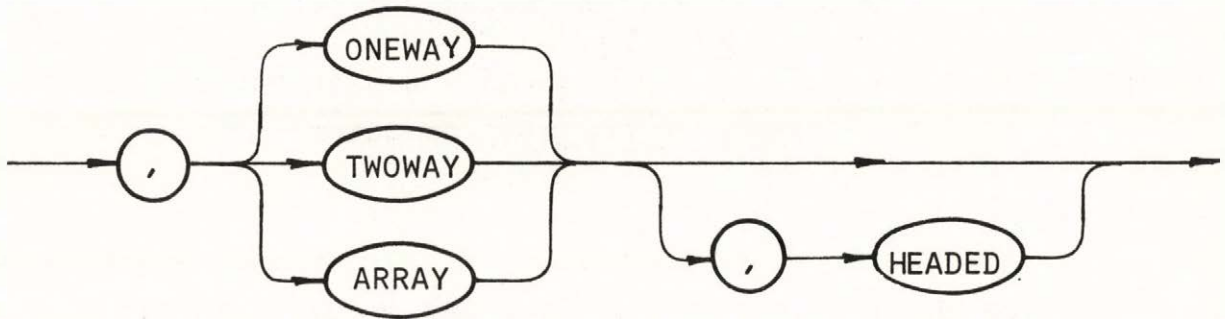
CF számoló mező
 /lásd 10. ábra/



26. ábra

| | |
|-----------|-----------------------------------|
| MFID | többszörös mező azonosító |
| UNDEF.ID. | definiálatlan azonosító |
| FIELD ID. | mező azonosító |
| MF | többszörös mező /lásd 9. ábra/ |

REP



27. ábra

| | |
|--------|---------|
| ONEWAY | alapszó |
| TWOWAY | alapszó |
| ARRAY | alapszó |
| HEADED | alapszó |

A TANULMÁNYOK sorozatban 1978-ban megjelentek:

- 74/1978 Vorträge über das graphische Display GD'71
- 75/1978 Vaskövi István - Gallbavy Márta: Anyagszétválasztási rendszerek tervezésének és optimális ütemeltetésének általános megközelítése
- 76/1978 Somló János - Nagy Judit: Módszer munkadarabok forgácsoló megmunkálási folyamatának optimalizálására.
- 77/1978 Szászné Turchányi Piroska: Optimalizálási feladatok csomagkapcsolt számítógéphálózatok tervezésénél
- 78/1978 Darvas Péter - Gallai István - Hosszu Péter - Kramer Gergely: Papers on Computer Graphics
- 79/1978 Dr. Adolf Kotzauer:
Beschriftung und Bemassung von automatisch erstellten Zeichnungen unter Benutzung des graphischen dialogs
- 80/1978 Studies in Applied Stochastic Programming I.
- 81/1978 Peter Bonitz: Ein Beitrag zur Theorie des Entwurfs doppelt gekrümmter Flächen unter differentialgeometrischen und rechentechnischen Aspekten.
- 82/1978 Tankó József: Szabályos job-folyam párok ütemezésének vizsgálata I.
- 83/1978 Tankó József: Szabályos job-folyam párok ütemezésének vizsgálata II.
- 84/1978 Bányász Csilla - Keviczky László: Discrete Time Identification of Linear Dynamic Process
- 85/1978 Dr. Hoffmann Péter: számítógépes szerszám-gépvezérlés egy alkatrészprogramozási módszere

- 86/1978 Ruda Mihály: A SIS77 statisztikai információs rendszer kialakításának szempontjai, alkalmazásának és továbbfejlesztésének lehetőségei
- 87/1978 Téli iskola - Operációs rendszerek elmélete
- 88/1978 Renner Gábor - Gaál Balázs - Hermann Gyula - Horváth László - Várady Tamás: Szoborszerű felületek tervezése és megmunkálása

A TANULMÁNYOK sorozatban 1979-ben megjelentek:

- 89/1979 Ruda Mihály: A SIS77 statisztikai információs rendszer /a felhasznált számítástechnikai eszközök, a rendszer szerkezete és programjai/
- 90/1979 Bányász Csilla - Keviczky László: Optimum Insensitivity of the Linear-continuous Transformation
- 91/1979 Téli iskola /Szentendre/

MAGYAR
SZEMANTOS AKADEMIA
KÖNYVTÁRA

