

Datatype matrikelnummer.

For SIM er der i Ingres defineret:

```
1 datatype:          matrikelnummer
7 funktioner:       char(7) char (matrikelnummer)
                    matrikelnummer matrikelnummer (varchar)
                    char(7) morder (matrikelnummer)
                    integer2 stamnr (matrikelnummer)
                    integer2 matrnr_klasse (matrikelnummer)
                    char(7) vejoplysning (integer, char)
                    char(7) vandoplysning (integer, char)
6 typekonverteringer:
                    matrikelnummer -> char(7)
                    matrikelnummer -> longtext(7)
                    matrikelnummer -> matrikelnummer
                    char(*) -> matrikelnummer
                    varchar(*) -> matrikelnummer
                    longtext(*) -> matrikelnummer
```

1. Datatypen matrikelnummer.

Matrikelnummer består af 7 tegn, de første 4 er stamnummerets cifre, de sidste 3 er litra. Både stamnummer og litra højrestilles og felterne fyldes op med tegnet '/' (skråstreg) mod venstre. Første ciffer i stamnummer må ikke være '0'. Tilladte bogstaver i litra er: 'a' til 'z', 'æ' og 'ø' dog ikke 'j' og 'w'.

Et "blankt" matrikelnummer består af 7 skråstreger ("////////").

Hvis stamnummer = 7000 må litra ikke være blank.

Hvis stamnummer > 7000 skal litra være blank.

Eksempler på matrikelnummer:

```
///5///  
///7///  
///7//a  
///7//b  
///7/ab  
///7/bh  
1234abc  
7000//a  
7000abc  
7000fæø  
7001///  
7010///  
7084///  
9999///
```

(Eksempler på ikke tilladte matrikelnumre: //05//a, ///5///j, //12//å, 7000/// og 7001//a.)

Matrikelnummer definerer ordning efter følgende regler:

- 1) Blankt litra ("////") opfattes som litra "//a".
- 2) Alle tegn sammenlignes ud fra deres 8-bits tegnværdi uden fortegn (0 - 255): '/' < cifre < bogstaver a til z < 'æ' < 'ø', (OBS: 'å' er ikke med i denne liste!).
- 3) To matrikelnummer-felter sammenlignes (under hensyn til 1 og 2 ovenfor) som to tegnfølger.

Pas på: I den nuværende udgave af Ingres (6.3, 04-jul-1990 på md9) vil et SQL-udtryk som:

```
select ...,matrn timer,.... from rel_mat order by matrn;
```

blive sorteret alfabetisk efter den eksterne repræsentation af matrn - altså ikke efter den ovenfor definerede ordning af matrikelnummer. Se endvidere under funktionen **morder**(matrikelnummer).

1.1 Ekstern repræsentation af matrikelnummer.

Matrikelnummer præsenteres "eksternt" som 7 tegn (char(7)), med stamnummeret højrestillet i de første 4 positioner og litra venstrestillet i de sidste 3. Resten af positionerne udfyldes med blanke tegn (mellemlrum). Eksempler:

```
" 5 "
```

```
" 7 "
```

```
" 7a "
```

```
" 7b "
```

```
" 7ab "
```

```
" 7bh "
```

```
"1234abc"
```

```
"7000a "
```

```
"7000abc"
```

```
"7000fæø"
```

```
"7001 "
```

```
"7010 "
```

```
"7084 "
```

```
"9999 "
```

2. Funktioner.

I det følgende er funktionerne beskrevet.

2.1 Funktion: char(7) char (matrikelnummer).

Funktionen **char** konverterer fra typen matrikelnummer til typen char(7) (typen char med fast længde på 7 tegn) i eksternt format.

2.2 Funktion: matrikelnummer matrikelnummer (varchar).

Funktionen **matrikelnummer** konverterer fra typen varchar til typen matrikelnummer. Tekstkonstanter anført i SQL-udtryk (eks: '12a') bliver af Ingres opfattet som varchar. Foran- og efterstillede blanktegn ignoreres. Første ciffer i stamnummer må ikke være '0'.

2.3 Funktion: char(7) morder (matrikelnummer).

Funktionen **morder** (matrikel-order) konverterer fra typen matrikelnummer til typen char(7). Alle 7 tegn af det interne format for matrikelnummer kopieres til funktionsværdien. Morder(matrikelnummer) er indført af hensyn til "order by" i select-udtryk, som ikke benytter den sammenligning, der er defineret for matrikelnummer.

Et SQL-udtryk som:

```
select matrnr,mo=morder(matrnr),.... from rel_mat order by mo;
```

vil sortere resultatet i matrikelnummerorden.

2.4 Funktion: integer2 stamnr (matrikelnummer).

Funktionen **stamnr** omsætter stamnummerdelen af et matrikelnummer til et heltal (2 byte integer).

2.5 Funktion: integer2 matrnr_klasse (matrikelnummer).

Funktionen **matrnr_klasse** klassificerer et matrikelnummer. Funktionsværdien er et tal (2 byte integer) med følgende værdier:

- 1 = "Almindeligt" matrikelnummer (op til 7000) uden litra (stamnr).
- 2 = "Almindeligt" matrikelnummer (op til 7000) med litra.
- 3 = "Vejnummer", matrikelnummer = 7000 (har altid litra).
- 4 = "Registernumre", matrikelnummer over 7000 (har aldrig litra).

2.6 Funktion: char(7) vejoplysning (integer, char)

Funktionen **vejoplysning** har to parametre, et tal og en tekst (på mindst et tegn). Tallet er vejareal og teksten (det første tegn -

normalt en char(1)) er vejkode. Funktionen kombinerer vejareal og vejkode til én oplysning på 7 tegn, char(7):

vejkode = 'u' -> vejoplysning = 'uoplyst'

vejkode = 'e' -> vejoplysning = 'ej ber.'

vejkode = 'b' og $0 \leq \text{vejareal} \leq 9999999$
-> vejoplysning = vejareal omsat til tegn og
højrestillet i en 7-tegns værdi.

andre værdier -> vejoplysning = '*****'

2.7 Funktion: char(7) vandoplysning (integer, char)

Funktionen **vandoplysning** har to parametre, et tal og en tekst (på mindst et tegn). Tallet er vandareal og teksten (det første tegn - normalt en char(1)) er vandkode. Funktionen kombinerer vandareal og vandkode til én oplysning på 7 tegn, char(7):

vandkode = 'i' -> vandoplysning = ' incl. '

vandkode = 'e' -> vandoplysning = ' excl. '

vandkode = 'b' og $0 \leq \text{vandareal} \leq 9999999$
-> vandoplysning = vandareal omsat til tegn og
højrestillet i en 7-tegns værdi.

andre værdier -> vandoplysning = '*****'

3. Typekonverteringer (coersion).

Der er defineret følgende typekonverteringer, som Ingres (automatisk) benytter, når den skal omsætte data fra den ene type til den anden.

```
matrikelnummer -> char(7)
matrikelnummer -> longtext(7)
matrikelnummer -> matrikelnummer
char(*) -> matrikelnummer
varchar(*) -> matrikelnummer
longtext(*) -> matrikelnummer
```

Matrikelnummer konverteres til char(7) eller longtext(7) efter samme definition som til eksternt format. Konverteringen matrikelnummer til matrikelnummer er blot en kopiering. Konvertering fra char(*), varchar(*) og longtext(*) foregår på samme måde som beskrevet under funktionen matrikelnummer. Stjernen i parenteserne angiver en vilkårlig længde (max 64 for char og 62 for varchar og longtext).

4. Fejlmeldinger.

Følgende liste giver en oversigt over fejlkoder og -tekster, som kan forekomme i forbindelse med matrikelnummer. De fleste af den "skulle ikke kunne forekomme", men er for fuldstændighedens skyld taget med.

```
0x200000, "Usmnr_compare: Type/length mismatch"
0x200001, "Usmnr_lenchk: Invalid length for MATRNR datatype"
0x200002, "Usmnr_keybld: Invalid key type"
0x200003, "Usmnr_getempty: type/length mismatch"
0x200022, "usmnr_valchk: Vejnumre (7000) skal have litra"
0x200022, "usmnr_valchk: Reg.numre (>7000) må IKKE have litra"
0x200004, "Usmnr_hashprep: type/length mismatch"
0x200005, "Usmnr_helem: Type for histogram incorrect"
0x200006, "Usmnr_helem: Length for histogram incorrect"
0x200007, "Usmnr_helem: Base type for histogram incorrect"
0x200008, "Usmnr_helem: Base length for histogram incorrect"
0x200008, "Usmnr_hmin: Invalid input parameters"
0x200009, -- do --
0x20000a, -- do --
0x20000b, -- do --
0x20000c, "Usmnr_dhmin: Invalid input parameters"
0x20000d, -- do --
0x20000e, -- do --
0x20000f, -- do --
0x200010, "Usmnr_hmax: Invalid input parameters"
0x200011, -- do --
0x200012, -- do --
0x200013, -- do --
0x200014, "Usmnr_dhmax: Invalid parameters"
0x200015, -- do --
0x200016, -- do --
0x200017, -- do --
0x200018, "Usmnr_hg_dtlm: Invalid parameters"
0x200019, -- do --
0x20001a, "Usmnr_minmaxdv: Invalid parameters"
0x20001b, -- do --
0x20001c, -- do --
0x20001d, -- do --
0x20001e, "Usmnr_convert: Insufficient space for matrnr output"
0x20001e, "Usmnr_convert: Insufficient space for matrnr output"
0x20001F, "Usmnr_convert: Unknown input type"
0x200020, "Unable to convert to matrikelnummer (1)"
0x200020, "Invalid syntax for matrikelnummer"
0x200020, "Unable to convert to matrikelnummer (2)"
0x200021, "usmnr_tmlem: Invalid input data"
0x200021, "Usmnr_dbtoev: Invalid input data"
0x200026, "Usmnr_morder: Invalid input"
0x200027, "Usmnr_stamnr: Invalid input/output"
0x200028, "Usmnr_matrnr_klasse: Invalid input/output"
0x200029, "Usmnr_vejoplysning: Invalid input/output"
0x20002a, "Usmnr_vandoplysning: Invalid input/output"
```

```
sys$input/opt /nodebug/notrace 'p1'
!
! This CLUSTER statement forces the transfer vector to the beginning of
! the shared image. It should not be removed.
!
Cluster = TRANSFER_VECTOR,, udviklings:[sim.objekt]ii_useradt_xfer.obj
!
! Replace the object module below with the object modules
! defining Installation Datatypes
!
udviklings:[sim.objekt]matrn timer. obj, -
sys$library:vaxcrtl/share
psect_attr = _HUGE, NOWRT
!
! End of object modules defining datatypes.
!
NAME = IIUSERADT
!
! Note that the shared image id should not be changed. INGRES expects
this
! level. The shared image ID can be changed ONLY be the product vendor.
!
IDENTIFICATION = "v1-000"
GSMATCH=LEQUAL, 1, 0
-----
```

Bemærk, at MATRNRZZ.EXE skal have "file protection w:re" for at kunne ses af alle (det gælder også katalogfilerne).

For at få Ingres til at benytte denne del af programmet, skal der være defineret en "logical" som refererer til MATRNRZZ.EXE. Den skal være defineret med "executive mode" for at kunne ses af installerede programmer. Hvis den skal kunne ses af en "Ingres-server" i testsystemet, skal den defineres i TINGRES' gruppe-tabel. For at gøre det udføres en kommando under TINGRES:

```
$ define/group/exec ii_useradt udviklings:<sim.objekt>matrn timer.exe
```

Det gøres i den tilrettede kommandoprocedure

```
TESTINGDEV:<INGRES>iistartup1_sim.com
```

som starter testingres (aktiveres af II_STARTUP.COM i samme katalog). Ligeledes er IIJOBDEF_SIM.COM tilrettet med hensyn til II_USERADT. Den aktiveres af RUNSV.COM, som sætter/skifter "omgivelserne" til testinstallation af Ingres.

6. Indkøring.

Indkøring er foretaget under KRX i et underkatalog:

UDVIKLINGS:[KRX.IIOBJ]

Her er MATRNR.C tilrettet, oversat og testet. En "lokal" Ingres-server startes i en subprocess med følgende kommandoprocedure:

```
-----
$! zzstart.com, 90.06.26/kr          starter privat ingres-server
$!
$ on error then exit $status
$ on warning then exit $status
$!
$ priv ! ask for more priviledges
$ set noon !!!!!!!!!!!!!!!
$ deassign/job ii_useradt
$ set on !!!!!!!!!!!!!!!
$ define/job/executive_mode -
      ii_useradt udviklings:[krx.iiobj]matrnrrzz.exe
$ show logical/full ii_useradt
$!
$ if p1 .eqs. "" then p1 = f$trnlm("TT")
$ if p2 .eqs. "" then -
      p2 ="prmbx, sysprv, cmkrnl, world, readall, syslck, " + -
          "share, exquota, sysgbl, shmem"
$!
$ set command TESTINGDEV:[ingres.utility]iirundbms.cld
$ iirundbms -
    /sole_server -
    /fast_commit -
    /write_behind = 4 -
    /quantum -
    /connected_sessions = 32 -
    /active_sessions = 32 -
    /cursors_per_session = 16 -
    /database_count = 33 -
    /stack_size = 65536 -
    /input = 'f$trnlm("TT")' -
    /output = 'p1' -
    /error=udviklings:[krx.iiobj]iibdms.error -
    /priority = 4 -
    /priv=('p2') -
    /page = 40800 -
    /buffer_lim = 66384 -
    /file_limit = 75 -
    /io_direct = 37 -
    /queue_limit = 37 -
    /ast_limit = 74 -
    /io_buffered = 74 -
    /extent = 20300 -
    /nopublic -
    /nonames -
    TESTINGDEV:[ingres.bin]iibdms.exe
$ nopriv ! ask for no priviledges (only std. priv.)
$!
$ show proc/sub/out=zzstart_sub.tmp
$ search/window=1/out=zzstart_serv.tmp zzstart_sub.tmp ii_dbms_zz_
```

```

$ open/read file zzstart_serv.tmp
$ read/end_of_file=slut file server
$slut:
$ close file
$ delete zzstart_sub.tmp;*, zzstart_serv.tmp;*
$ assign 'server' ii_dbms_server
$!
$ exit
-----

```

På den måde kan en server startes uden at genere andres anvendelse af testsystemet. Serveren stoppes igen med følgende kommandoprocedure:

```

$! zzshut.com 90.06.26/kr
$!
$ iimonitor
set server shut
quit
$ deassign ii_dbms_server
$exit

```

NB: Man må selv holde øje med at stoppe en server inden man "logger" ud af systemet. Man skal også lade være med at starte mere en en.

I MATRNR.C er der muligheder for at bestille nogle testudskrifter. Det gøres ved i starten af filen at kommentere nogle C-macro definitioner ud: De her nævnte kan ophæve definitionen af dem, der danner udskrifterne:

```

/*****
#define DEBUG_N(navn)
#define DEBUG_DV(dvp)
#define DEBUG_I(i)
*****/

```

Testudskrifterne skrives normalt til den skærm, som har startet serveren. De kan omdirigeres ved at benytte parameter 1 til ZZSTART.COM (se ovenfor).

Følgende kommandoprocedure sætter "omgivelserne" for indkøring under KRX:

```

$! iiobj_login.com 90.06.14/kr
$ iitest
$ set def udviklings:[krx.iiobj]
$ define/job vaxc$include udviklings:[krx.iiobj]
$ define/job tlog testingdev:[ingres.files]
$ zzst*art := @udviklings:[krx.iiobj]zzstart.com
$ zzsh*ut := @udviklings:[krx.iiobj]zzshut.com
$ cd

```

hvor \$cd blot udskriver en påmindelse om katalog (\$sh def, og lignedne) og symbolet IITEST skifter til Testingres:

```

$ sh sym iitest
IIT*EST == "@TESTINGDEV:[INGRES]RUNSV.COM"

```

Kildetekst og oversættelse svarer til det, som er beskrevet under installation.