

Oversigt over databaseprocedurer i simbas

Procedure Navn: fk_ejerlav_brugt

Ejer: sim

Oprettet: 19-jun-1993

```
create procedure fk_ejerlav_brugt (ejerlav integer4) as declare msg
varchar(80) not null; antal integer; begin select count(*) into antal from
sim. omraade where ejerlav = :ejerlav; if 0 < antal then msg = 'Fejl, tabel
ejerlav (1): Ejerlavskode ' + varchar(:ejerlav) + ' bruges i omraade';
raise error 1 :msg; return; endif; end
```

Procedure Navn: fk_kommune_brugt

Ejer: sim

Oprettet: 19-jun-1993

```
create procedure fk_kommune_brugt (kommunenr integer2) as declare msg
varchar(80) not null; antal integer; begin select count(*) into antal from
sim. omraade where kommunenr = :kommunenr; if 0 < antal then msg = 'Fejl,
tabel kommune (1): Kommunenr ' + varchar(:kommunenr) + ' bruges i omraade';
raise error 1 :msg; return; endif; end
```

Procedure Navn: fk_kort_brugt

Ejer: sim

Oprettet: 19-jun-1993

```
create procedure fk_kort_brugt (kortnr integer2) as declare msg
varchar(80) not null; antal integer; begin select count(*) into antal from
sim. matkort where kortnr = :kortnr; if 0 < antal then msg = 'Fejl, tabel
kort (1): Kortnr ' + varchar(:kortnr) + ' bruges i matkort'; raise error 1
:msg; return; endif; end
```

Procedure Navn: fk_omraade_brugt

Ejer: sim

Oprettet: 19-jun-1993

```
create procedure fk_omraade_brugt (tesseractnr integer2) as declare msg
varchar(80) not null; antal integer; begin select count(*) into antal from
sim. grund where tesseractnr = :tesseractnr; if 0 < antal then msg = 'Fejl,
tabel omraade (3): Tesseractnr ' + varchar(:tesseractnr) + ' bruges i grund';
raise error 3 :msg; return; endif; select count(*) into antal from sim.
grundtrans where tesseractnr = :tesseractnr; if 0 < antal then msg = 'Fejl,
tabel omraade (3): Tesseractnr ' + varchar(:tesseractnr) + ' bruges i
grundtrans'; raise error 3 :msg; return; endif; select count(*) into antal
from sim. matkort where tesseractnr = :tesseractnr; if 0 < antal then msg =
'Fejl, tabel omraade (3): Tesseractnr ' + varchar(:tesseractnr) + ' bruges i
matkort'; raise error 3 :msg; return; endif; select count(*) into antal
from sim. flslod where ejertesseractnr = :tesseractnr; if 0 < antal then msg =
'Fejl, tabel omraade (3): Tesseractnr ' + varchar(:tesseractnr) + ' bruges i
flslod (ejer...)'; raise error 3 :msg; return; endif; select count(*) into
antal from sim. flslod where flstesseractnr = :tesseractnr; if 0 < antal then
msg = 'Fejl, tabel omraade (3): Tesseractnr ' + varchar(:tesseractnr) + '
bruges i flslod (fls...)'; raise error3 :msg; return; endif; select
count(*) into antal from sim. frastyk where tesseractnr = :tesseractnr; if 0 <
antal then msg = 'Fejl, tabel omraade (3): Tesseractnr ' +
varchar(:tesseractnr) + ' bruges i frastyk'; raise error 3 :msg; return;
endif; select count(*) into antal from sim. protokol where tesseractnr =
:tesseractnr; if 0 < antal then msg = 'Fejl, tabel omraade (3): Tesseractnr ' +
varchar(:tesseractnr) + ' bruges i protokol'; raise error 3 :msg; return;
endif; select count(*) into antal from sim. lod where tesseractnr =
:tesseractnr; if 0 < antal then msg = 'Fejl, tabel omraade (3): Tesseractnr ' +
varchar(:tesseractnr) + ' bruges i lod'; raise error 3 :msg; return; endif;
end
```

Oversigt over databaseprocedurer i simbas

Procedure Navn: fk_omraade_valid

Ejer: sim

Oprettet: 19-jun-1993

```

create procedure fk_omraade_valid (ejerlav integer4, kommunenr integer2)
as declare msg varchar(80) not null; check_ejl integer; check_kom integer;
begin select count(*) into check_ejl from sim.ejerlav where ejerlav =
:ejerlav; if check_ejl = 0 then msg = 'Fejl, tabel omraade (1):
Ejerlavskode ' + varchar(:ejerlav) + ' ukendt.'; raise error 1 :msg;
return; endif; select count(*) into check_kom from sim.kommune where
kommunenr = :kommunenr; if check_kom = 0 then msg = 'Fejl, tabel omraade
(2): Kommunenr ' + varchar(:kommunenr) + ' ukendt.'; raise error 1 :msg;
return; endif; end

```

Procedure Navn: iiqef_check_table

Ejer: \$ingres

Oprettet: 19-jun-1993

```

create procedure iiQEF_check_table( table_id i4 not null not default,
index_id i4 not null not default ) as BEGIN execute internal; END

```

Procedure Navn: ny_ejl_hist

Ejer: sim

Oprettet: 15-dec-1993

```

create procedure ny_ejl_hist(ejlkod integer not null, glejlbet
varchar(40) ) as begin insert into sim.ejerlavshist (ejerlav, dato,
ejerlavsbet) values( ejlkod, 'now', glejlbet); if (iierrornumber > 0) then
raise error iierrornumber; endif; end

```

Procedure Navn: ret_bemrk

Ejer: sim

Oprettet: 19-jun-1993

```

create procedure ret_bemrk ( sagsnr char(12) not null, ejerlav integer4
not null, matrnr char(7) not null, bemrk char(132) not null ) as declare
orden integer4 not null; bruger char(32) not null; antal integer4 not null;
begin select dbmsinfo('username') into :bruger; if iierrornumber != 0 then
rollback; return iierrornumber; endif; commit; select orden into orden from
sim.sag where sagsnr = :sagsnr; if iierrornumber != 0 then rollback;
return iierrornumber; elseif irowcount = 0 then rollback; message 'Sagen
findes ikke.'; return 1; elseif orden >= 2000000000 then rollback; message
'Sagen er endnu ikke afsluttet.'; return 2; endif; update sim.sb_bemrk set
bemrk = :bemrk, sidst_rettet = date('today'), bruger = :bruger where sagsnr
= :sagsnr and ejerlav = :ejerlav and matrnr = :matrnr; if iierrornumber !=
0 then rollback; return iierrornumber; elseif irowcount = 0 then rollback;
select count(*) into antal from sim.grund g, sim.omraade o, sim.sag s
where g.tesseractnr = o.tesseractnr and o.ejerlav = :ejerlav and g.matrnr =
matrikelnummer(:matrnr) and g.snit = s.snit and s.sagsnr = :sagsnr; if
iierrornumber != 0 then rollback; return iierrornumber; elseif antal = 0
then rollback; message 'Sag, matr.nr. findes ikke.'; return 3; else insert
into sim.sb_bemrk (sagsnr, ejerlav, matrnr, bemrk, sidst_rettet, bruger)
values (:sagsnr, :ejerlav, :matrnr, :bemrk, date('today'), :bruger); if
iierrornumber != 0 then rollback; return iierrornumber; endif; endif;
endif; commit; return 0; end

```

Procedure Navn: risidxdel

Ejer: dikreg

Oprettet: 22-feb-1994

```

create procedure risidxdel (objname varchar(31)) as begin delete from
'dikreg'.ris_indexes where index_name = :objname; delete from 'dikreg'.
ris_index_columns where index_name = :objname; end

```

Oversigt over databaseprocedurer i simbas

Procedure Navn: ristabel

Ejer: dikreg

Oprettet: 22-feb-1994

```
create procedure ristabel (objname varchar(31)) as begin delete from
'dikreg'. ris_privs where table_name = :objname; delete from 'dikreg'.
ris_columns where table_name = :objname; delete from 'dikreg'. ris_tables
where table_name = :objname; delete from 'dikreg'. ris_indexes where
table_name = :objname; delete from 'dikreg'. ris_index_columns where
table_name = :objname; end
```

Procedure Navn: risviewdel

Ejer: dikreg

Oprettet: 22-feb-1994

```
create procedure risviewdel (objname varchar(31)) as begin delete from
'dikreg'. ris_privs where table_name = :objname; delete from 'dikreg'.
ris_columns where table_name = :objname; delete from 'dikreg'. ris_views
where view_name = :objname ; delete from 'dikreg'. ris_tables where
table_name = :objname; end
```

Procedure Navn: sr_sagsplac_insert_proc

Ejer: sim

Oprettet: 19-jun-1993

```
create procedure sr_sagsplac insert_proc (journr char(12) not null, linr
integer not null, intern char (7) not null) as begin insert into sim.
sr_sagsplacering (journr, linr, dato, aktion, intern) values (:journr,
:linr, date ('today'), 'T', :intern); return ierrornumber; end
```

Oversigt over rules i simbas

Rule Navn: `fk_kommune_del`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_kommune_del` after delete of `kommune` execute procedure `fk_kommune_brugt` (`kommunenr = old.kommunenr`)

Rule Navn: `fk_kommune_upd`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_kommune_upd` after update(`kommunenr`) of `kommune` where `old.kommunenr != new.kommunenr` execute procedure `fk_kommune_brugt` (`kommunenr = old.kommunenr`)

Rule Navn: `fk_kort_del`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_kort_del` after delete of `kort` execute procedure `fk_kort_brugt` (`kortnr = old.kortnr`)

Rule Navn: `fk_kort_upd`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_kort_upd` after update(`kortnr`) of `kort` where `old.kortnr != new.kortnr` execute procedure `fk_kort_brugt` (`kortnr = old.kortnr`)

Rule Navn: `fk_omraade_del`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_omraade_del` after delete of `omraade` execute procedure `fk_omraade_brugt` (`tesseractnr = old.tesseractnr`)

Rule Navn: `fk_omraade_ins`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_omraade_ins` after insert of `omraade` execute procedure `fk_omraade_valid` (`ejerlav = new.ejerlav`, `kommunenr = new.kommunenr`)

Rule Navn: `fk_omraade_upd1`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_omraade_upd1` after update(`tesseractnr`) of `omraade` where `old.tesseractnr != new.tesseractnr` execute procedure `fk_omraade_brugt` (`tesseractnr = old.tesseractnr`)

Rule Navn: `fk_omraade_upd2`

Ejer: `sim`

Oprettet: 19-jun-1993

create rule `fk_omraade_upd2` after update of `omraade` execute procedure `fk_omraade_valid` (`ejerlav = new.ejerlav`, `kommunenr = new.kommunenr`)

Oversigt over rules i simbas

Rule Navn: ny_ejl_bet

Ejer: sim

Oprettet: 15-dec-1993

create rule ny_ejl_bet after update (ejerlavsbet) on ejerlav execute
procedure ny_ejl_hisE (ejlkod = old.ejerlav, glejlbet = old.ejerlavsbet)
