

Temporära tabeller i DB2

RUG 23 januari 2007

Peter Backlund
Peter Backlund DB2-Konsult AB

Tack och “disclaimer”

- Tack till
 - Tine Björk, Sandvik
 - Jan-Eirik Bergesen, DnB NOR
- Jag är ansvarig för hela innehållet i den här presentationen – alla påståenden och eventuella felaktigheter är helt mina egna

Vad Tine ville ha...

```
IX_NAME      ! COL_NAMES
-----
PCART        ! PROPID (A) , GRADEID (A)
UCART12     ! PROPID (A) , GRADEID (A) , BRANDCD (A)
XCART1      ! ARTCDPACK (A) , BRANDCD (A)
XCART10     ! PROPID (A) , GRADEID (A) , ARTSTATUSCD (A)
XCART11     ! PROPID (A) , GRADEID (A) , STDCATEG (A)
XCART13     ! COROPAKNO (A)
XCART14     ! PROJNO (A) , STDLISTCD (A)
XCART15     ! PRODNOEAN (A)
XCART16     ! ARTCDPREP (A) , PROPID (A) , GRADEID (A) , BRANDCD (A)
XCART17     ! SUBCOMPCD (A)
XCART19     ! BRANDCD (A) , PROPID (A) , GRADEID (A)
XCART2      ! ARTCDPREP (A) , BRANDCD (A)
XCART3      ! PROPID (A) , GRADEID (A) , ARTCDPACK (A) , BRANDCD (A)
```

Pivotering

- Från SYSKEYS

IxName	ColName	ColSeq	Ordering
KONTOIX	KONTONR	1	A
KONTOIX	DATUM	2	D
KONTOIX	TID	3	A

- Till

KONTOIX KONTONR(A), DATUM(D), TID(A)

eller

KONTOIX +KONTONR-DATUM+TID

ColNames i DB2 LUW

- I katalogtabellen SYSIBM.SYSINDEXES finns en column
COLNAMES VarChar(640)
- För indexet KONTOIX står där

+KONTONR-DATUM+TID

SYSKEYS i DB2 LUW

- SYSIBM.SYSKEYS finns inte i DB2 LUW, däremot SYSIBM.SYSINDEXCOLUSE
- Jag har därför skapat en vy

```
create view SYSKEYS as
select indschema as ixcreator
      , indname   as ixname
      , colname
      , colseq
      , colorder  as ordering
from SYSIBM.SYSINDEXCOLUSE
```

Rekursiv SQL

```
with t(ixcols,n) as
(select cast('' as varchar(1000)), 0
 from sysibm.sysdummy1
 union all
 select ixcols concat case ordering when 'A'
 then '+' else '-' end concat colname, n+1
 from syskeys,t where colseq = n+1
 and ixcreator = 'BACKLUND'
 and ixname = 'KONTOIX')

select ixcols from t
where n = (select max(n) from t);
```

	0
+KONTONR	1
+KONTONR-DATUM	2
+KONTONR-DATUM+TID	3

Driva rekursiv SQL - UDF

```
create function RUG2007_01
(p_ixcreator varchar(128), p_ixname varchar(128))
returns varchar(1000)
return
with t(ixcols,n) as
(select cast('' as varchar(1000)), 0 from sysibm.sysdummy1
 union all
 select ixcols concat case ordering when "A" then "+" else "-" end concat
 colname, n+1
 from syskeys,t where colseq = n+1
 and ixcreator = upper(p_ixcreator)
 and ixname = upper(p_ixname))
select ixcols from t
where n = (select max(n) from t);

values rug2007_01('backlund','kontoix') +KONTONR-DATUM+TID
```

Driva Rekursiv SQL - SP

```
create procedure RUG2007_02(in pi_ixcreator varchar(128)
, in pi_ixname varchar(128), out po_ixcols varchar(1000))
begin
  declare stmt varchar(4000);
  declare x cursor for s;
  set stmt = 'with t(ixcols,n) as
(select cast('' as varchar(1000)), 0 from sysibm.sysdummy1
union all
select ixcols concat case ordering when "A" then "+" else "-" end concat
colname, n+1
from syskeys,t where colseq = n+1
and ixcreator = upper(cast(? as varchar(128)))
and ixname = upper(cast(? as varchar(128))))
select ixcols from t
where n = (select max(n) from t);
prepare s from stmt;
open x using pi_ixcreator, pi_ixname;
fetch x into po_ixcols;
end
```

Tines lösning

```
SELECT INDEXES.NAME
, KEYS1.COLNAME || '(' || KEYS1.ORDERING || ')',
  || COALESCE(KEYS2.COLNAME, ' ') || '('
  || COALESCE(KEYS2.ORDERING, ' ') || ')',
  || COALESCE(KEYS3.COLNAME, ' ') || '('
  || COALESCE(KEYS3.ORDERING, ' ') || ')'

FROM DB2U.SYSINDEXES INDEXES
LEFT JOIN DB2U.SYSKEYS KEYS1
ON KEYS1.IXCREATOR = INDEXES.TBCreator
AND KEYS1.IXNAME = INDEXES.NAME
AND KEYS1.COLSEQ = 1
LEFT JOIN DB2U.SYSKEYS KEYS2
ON KEYS2.IXCREATOR = INDEXES.TBCreator
AND KEYS2.IXNAME = INDEXES.NAME
AND KEYS2.COLSEQ = 2
LEFT JOIN DB2U.SYSKEYS KEYS3
ON KEYS3.IXCREATOR = INDEXES.TBCreator
AND KEYS3.IXNAME = INDEXES.NAME
AND KEYS3.COLSEQ = 3

WHERE INDEXES.TBNAME = 'TCART'
AND INDEXES.TBCreator = 'C20'
AND INDEXES.CREATOR = 'C20'
```

SQL Procedure *IxCols*

- In Pi_IxCreator, Pi_IxName
- Out Po_IxCols

```
declare c2 cursor for select
case ordering when 'A' then '+' else '-' end concat colname
from syskeys where ixcreator = pi_ixcreator
and ixname = pi_ixname order by colseq;
```

```
Set po_ixcols = "";
l2: loop
    fetch c2 into v_n;
    if sqlstate <> '00000' then leave l2; end if;
    set po_ixcols = po_ixcols concat v_n;
end loop l2;
```

Alternativ *IxCols*

```
declare c2 cursor for select rtrim(colname) concat
case ordering when 'A' then '(A)' else '(D)' end
from sysibm.syskeys
where ixcreator = upper(pi_ixcreator)
and ixname = upper(pi_ixname) order by colseq;
```

```
set po_ixcols = po_ixcols concat
case po_ixcols when " then " else ', ' end concat v_n ;
```

SQL Procedure *TblxCols*

- In Pi_TbCreator, Pi_TbName
- Result T(IxCreator, IxName, ColCount, IxCols)

```
declare c1 cursor for select creator, name, colcount
from sysibm.sysindexes where tbcreator = pi_tbcreator
and tbname = pi_tbname order by creator, name;
```

```
l1: loop
  fetch c1 into v_ixcreator, v_ixname, v_colcount;
  if sqlstate <> '00000' then leave l1; end if;
  call IxCols(v_ixcreator, v_ixname, v_ixcols);
  insert into session.t
    values(v_ixcreator, v_ixname, v_colcount, v_ixcols);
end loop l1;
```

Result Set kräver tabell

- Denna lösning kräver en tabell för att kunna returnera ett variabelt antal rader
- I första försöket använde jag en *Declare Global Temporary Table* (DTT)
- Väsentliga problem med DTT
 - hantering (2 problem – minst!)
 - resurskrävande

Problem 1) med DTT

- Vi behöver följande 2 SQL-satser (bland annat)

a: Declare x cursor with return for
select name from session.dtt;

b: Declare global temporary table dtt (name varchar(128));

- b är en exekverbar sats och måste därför placeras efter a, men satsen a refererar tabellen dtt som deklarerar i satsen b
- Vi har ett Moment 22!

Problem 1) med DTT - LUW

- Vi kan använda multipla Begin-End block

```
Begin
    Declare global temporary table dtt
        (name varchar(128));
    ...
    Begin
        Declare x cursor with return for
            select name from session.dtt;
    ...
    End;
End
...
```

Problem 1) med DTT - Generellt

- Vi kan använda dynamisk SQL

```
declare v_s varchar(50)
    default 'select name from session.dtt';
declare x cursor with return for s;
...
declare global temporary table dtt
    (name varchar(128));
...
prepare s from v_s;
open x;
```

Problem 2) med DTT

- Den temporära tabellen finns ibland kvar när vi försöker skapa den ...

Problem 2) med DTT - LUW

```
Begin
  Declare global temporary table dtt
    (name varchar(128)) with replace;
  ...
  Begin
    Declare x cursor with return for
      select name from session.dtt;
  ...
  End;
  ...
End
```

Problem 2) med DTT – z/OS V8

```
declare v_s varchar(50)
  default 'select name from session.dtt';
declare x cursor with return for s;
...
declare global temporary table dtt
  (name varchar(128)) on commit drop table;
...
prepare s from v_s;
open x;
```

Problem 2) – norsk lösning

```
declare v_s varchar(50)
    default 'select name from session.dtt';
declare x cursor with return for s;
...
declare global temporary table dtt
    (name varchar(128));
...
prepare s from v_s;
open x;
drop table session.dtt;
end
```

SQL0910N The SQL statement cannot access an object on which a modification is pending. SQLSTATE=57007

Problem 2) – Drop

```
declare continue handler for
    SQLState '42704' set v_name="";
...
drop table session.dtt;
declare global temporary table dtt
    (name varchar(128));
...
```

Problem 2) – Delete

```
declare continue handler for
  SQLState '42710' set v_name="";
...
declare global temporary table dtt
  (name varchar(128));
delete from session.dtt;
...
```

Jämförelse BTT, CTT och DTT

- Eftersom DTT sägs vara resurskrävande, har jag jämfört följande
 - BTT - "basic table"
 - CTT - "created table" – endast i z/OS
 - DTT - "declared table"

Vad mäter jag?

```
Declare C1 Cursor for  
Select Name From SYSIBM.SYSCOLUMNS
```

```
Delete from xTT;  
L1: loop  
    Fetch C1 Into v_Name;  
    If SQLState <> '00000' Then Leave L1; End If;  
    Insert Into xTT Values(v_Name);  
End Loop L1;
```

Vad mäter jag?

```
Declare C1 Cursor for  
    Select Name From SYSIBM.SYSCOLUMNS;  
-- Starta tidtagningen;
```

```
Delete from xTT;  
L1: loop  
    Fetch C1 Into v_Name;  
    If SQLState <> '00000' Then Leave L1; End If;  
    Insert Into xTT Values(v_Name);  
End Loop L1;  
-- Avsluta tidtagningen;
```

Tillägg för DTT

```
Declare C1 Cursor for
  Select Name From SYSIBM.SYSCOLUMNS;
-- Starta tidtagningen
Declare Global Temporary Table DTT(Name Varchar(128));
Delete from xTT;
L1: loop
  Fetch C1 Into v_Name;
  If SQLState <> '00000' Then Leave L1; End If;
  Insert Into xTT Values(v_Name);
End Loop L1;
-- Avsluta tidtagningen
```

Hur mäter jag?

```
Declare Global Temporary Table TidTab
  (Tid TimeStamp Not Null);
Insert Into Session.TidTab
  Values(Current Timestamp);

-- Kod som skall mätas

Select Current Timestamp – Tid
  Into Po_Tid From Session.TidTab;
Drop Table Session.TidTab;
```

Mätningar i DB2 9 Windows

Tider i millisekunder

- BTT 118
- DTT 81

Mätningar i z/OS

Tider i millisekunder

- BTT 29,4
- CTT 21,8
- DTT 33,5

```
--drop procedure tbixcols
--drop table session.t
Create procedure tbixcols
(in pi_tbcreator varchar(30)
,in pi_tbname    varchar(30))
begin
declare v_creator, v_name varchar(30);
declare v_colcount smallint;
declare v_ixstring, v_n varchar(3000);
declare c_string varchar(80) default 'select * from session.t';
declare sqlstate char(5);
declare c1 cursor for
    select creator, name, colcount
    from sysindexes
    where tbcreator = upper(pi_tbcreator)
    and   tbname    = upper(pi_tbname)
    order by creator, name;
declare c2 cursor for
    select case ordering
        when 'A' then '+'
        else      '-'
    end
    concat
        rtrim(colname)
    from syskeys
    where ixcreator = upper(v_creator)
    and   ixname    = upper(v_name)
    order by colseq;
Declare r cursor with return for stmt;
declare continue handler for SQLState '42710' set v_ixstring='';
declare global temporary table t(t_ixcreator varchar(30)
                                , t_ixname   varchar(30)
                                , t_colcount  smallint
                                , t_ixstring  varchar(3000));

delete from session.t;
open c1;
l1: loop
    fetch c1 into v_creator, v_name, v_colcount;
    if sqlstate <> '00000' then leave l1; end if;
    set v_ixstring = '';
    open c2;
    l2: loop
        fetch c2 into v_n;
        if sqlstate <> '00000' then leave l2; end if;
        set v_ixstring = v_ixstring concat v_n;
    end loop l2;
    close c2;
    insert into session.t(t_ixcreator, t_ixname, t_colcount, t_ixstring)
        values(v_creator, v_name, v_colcount, v_ixstring);
    end loop l1;
prepare stmt from c_string;
open r;
end
```