



# Parallellisera själv!

Sven Heidorn  
Centrala Studiestödsnämnden  
[sven.heidorn@csn.se](mailto:sven.heidorn@csn.se)

Swedish DB2 User Group 04/01/20

## Bakgrund & Problem



- Vad gör man när bearbetningens natur är sådan att DB2 inte själv kan parallellisera jobbet?
- En lösning är att köra samma program parallellt över olika intervall av datat.
- Hur ska man då dela upp jobbet mellan de parallella programmen?
  
- Ta reda på MAX och MIN, och sedan räkna fram värdet mitt i mellan dessa två.
  - Fördel: Enkelt i vissa fall.....
  - Nackdel: ... men oftast ändå inte så enkelt och grupperna blir nästan säkert inte lika stora.
  
- Vad är det egentligen vi vill ha?
  - Medianvärdet – vilket värde har nyckeln på den rad som representerar hälften av datat .
  - Tekniken är den samma oavsett hur många delar man vill ha.
  
  - Fördel: Exakt, nästan perfekt fördelning.
  - Nackdel: Bearbetningens natur är sådan att det kan kosta en del att köra.

När vi tog fram den här lösningen var det med avsikt att lösa ovanstående problem. Tekniken borde dock även gå att använda till att beräkna partitionerings gränser i mainframe DB2.

## Ur DB2 UDB V7.1 SQL Cookbook



"The median is the point where half the items in the sample are above (in value) and half are below. To calculate the median, one must first get a count of all the items in the sample, then sequence each individual value (from low to high by value). Lastly, one must get that item which is halfway down the series"

"All of the above can be done using a single SQL statement in DB2, but the performance stinks!"  
<Graeme Birchall>

...men låt oss ändå försöka....

DB2 UDB V8.1 SQL Cookbook innehåller bl.a grunden till det vi har gjort i denna presentation.

Det är en utmärkt lärobok i SQL som dessutom går att ladda ner gratis från följande adress:

[http://ourworld.compuserve.com/homepages/graeme\\_birchall/HTM\\_COOK.HTM](http://ourworld.compuserve.com/homepages/graeme_birchall/HTM_COOK.HTM)

## DB2 LUW V8.1 – Förslag 1



```
WITH ROW_NMBR AS  
  (SELECT A.FK_AAB_PERS_CSNNR, ROW_NUMBER () OVER (ORDER BY A.FK_AAB_PERS_CSNNR) AS ROW#  
   FROM E569DBP.TLAAN A WHERE SLUTREGLERAD = 'N')
```

```
SELECT FK_AAB_PERS_CSNNR  
FROM ROW_NMBR  
WHERE ROW# = (SELECT (MAX(ROW#) / 3) + 1 FROM ROW_NMBR)  
OR ROW# = (SELECT ((MAX(ROW#) / 3) * 2) + 1 FROM ROW_NMBR);
```

then sequence each individual value (from low to high by value)

```
FK_AAB_PERS_CSNNR  
-----  
15801087  
29665528
```

get a count of all the items in the sample

get that item which is halfway down the series. Eller som i det här fallet 1/3 respektive 2/3.

```
2 record(s) selected.  
real 0m17.89s
```

Tabellen innehåller ca. 2350000 rader

## DB2 LUW V8.1 – Förslag 1



```
SELECT SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 0          AND 15801087 THEN 1 ELSE 0 END) AS INT1,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 15801088 AND 29665528 THEN 1 ELSE 0 END) AS INT2,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 29665529 AND 99999999 THEN 1 ELSE 0 END) AS INT3  
FROM E569DBP.TLAAN  
WHERE SLUTREGLERAD = 'N'  
WITH UR;
```

INT1	INT2	INT3
599022	599022	599019

ELSE 0 är redundant, det går lika bra med:

```
SUM(CASE WHEN COL1 BETWEEN 0 AND 100 THEN 1 END)
```

## DB2 LUW V8.1 – Förslag 2



```
WITH ROW_NMBR AS
  (SELECT A.FK_AAB_PERS_CSNNR, ROW_NUMBER () OVER (ORDER BY A.FK_AAB_PERS_CSNNR) AS ROW#
   FROM E569DBP.TLAAN A TABLESAMPLE SYSTEM(10) WHERE SLUTREGLERAD = 'N')
SELECT FK_AAB_PERS_CSNNR
FROM ROW_NMBR
WHERE ROW# = (SELECT (MAX(ROW#) / 3) + 1 FROM ROW_NMBR)
OR   ROW# = (SELECT ((MAX(ROW#) / 3) * 2) + 1 FROM ROW_NMBR);

FK_AAB_PERS_CSNNR
-----
      15401490
      29509510

2 record(s) selected.
real    0m3.68s
```

## DB2 LUW V8.1 – Förslag 2

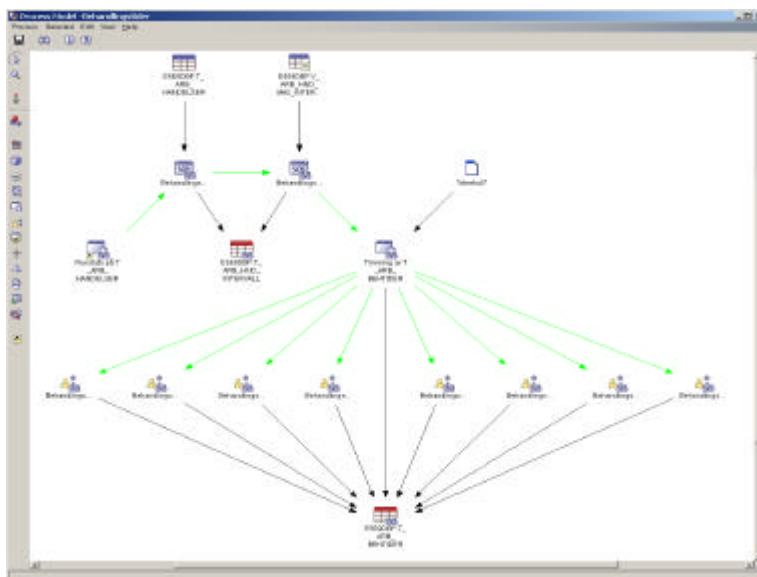


```
SELECT SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 0          AND 15401490 THEN 1 ELSE 0 END) AS INT1,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 15401491 AND 15401490 THEN 1 ELSE 0 END) AS INT2,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 15401491 AND 99999999 THEN 1 ELSE 0 END) AS INT3  
FROM E569DBP.TLAAN  
WHERE SLUTREGLERAD = 'N'
```

INT1	INT2	INT3
560211	628474	608378

Skilnaden mellan största och minsta gruppen är ca 68000.  
Om det är acceptabelt får man själv bedöma.  
Sampling är nog fördelaktigare vid större datamängder och fler grupper.

# Real world example



## Resultat



The screenshot shows the 'Work in Progress' window in SQL Server Enterprise Manager. The table below represents the data visible in the window. A red circle highlights the 'Completed' column, and two green circles highlight the 'Completed' times for the 12570 and 12580 jobs.

Name	Type	Process name	Status	Run ID	Scheduled	Completed	Source database
Botaniser körning - OPU8 - Load	DB1 UDB	Botaniser Data...	Successful	12045	01/19/2004 5:44:05 AM	01/19/2004 5:44:16 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12581	01/17/2004 6:50:39 AM	01/17/2004 7:24:32 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12583	01/17/2004 6:50:39 AM	01/17/2004 7:15:16 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12578	01/17/2004 6:50:39 AM	01/17/2004 7:24:17 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12585	01/17/2004 6:50:39 AM	01/17/2004 7:15:47 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12582	01/17/2004 6:50:39 AM	01/17/2004 7:25:15 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12579	01/17/2004 6:50:39 AM	01/17/2004 7:13:20 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12580	01/17/2004 6:50:39 AM	01/17/2004 7:23:56 AM	-
Behandlingstider - BehandlingstiderSP	DW Stored Procedures	Behandlingstider	Successful	12584	01/17/2004 6:50:39 AM	01/17/2004 7:27:22 AM	-
Behandlingstider - Faktatabell - Export	DB1 UDB	Behandlingstid...	Successful	12484	01/18/2004 10:18:53 AM	01/18/2004 10:22:06 AM	STARPROO
Behandlingstider - Faktatabell - Load	DB1 UDB	Behandlingstid...	Successful	12483	01/18/2004 10:23:06 AM	01/18/2004 10:24:00 AM	-
Behandlingstider - Faktatabell - Runstate	DB1 UDB	Behandlingstid...	Successful	12282	01/12/2004 1:15:09 PM	01/12/2004 1:22:17 PM	-

- Differensen i tid mellan det snabbaste och det långsammaste jobbet är 3 minuter och 24 sekunder, helt acceptabelt för en genomsnittlig körtid på 35 minuter.
- Utan table sampling är differensen mellan jobben mindre än 1 minut.

## Mainframe DB2

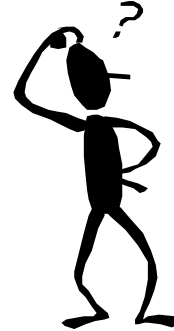


- Recap:

```
- WITH ROW_NMBR AS
  (SELECT A.FK_AAB_PERS_CSNNR, ROW_NUMBER () OVER (ORDER BY A.FK_AAB_PERS_CSNNR) AS ROW#
   FROM E569DBP.TLAAN A WHERE SLUTREGLERAD = 'N')
SELECT FK_AAB_PERS_CSNNR
FROM ROW_NMBR
WHERE ROW# = (SELECT (MAX(ROW#) / 3) + 1 FROM ROW_NMBR)
OR ROW# = (SELECT ((MAX(ROW#) / 3) * 2) + 1 FROM ROW_NMBR);
```

- Problem:

```
- WITH table name AS ( fullselect )
- ROW_NUMBER () OVER ( window partition clause | window order clause )
- Funktionaliteten saknas i mainframe DB2
```



Funktionaliteten att skapa en temp tabell in-line dvs. WITH, kommer att finnas i DB2 for z/Os V8.

Problemet att numrera och sortera datat kvarstår dock och är egentligen den springande punkten.

## Lösningförslag Mainframe DB2 Del 1



```
DECLARE GLOBAL TEMPORARY TABLE SESSION.GTLAAN_INTERVAL
(CSN_NUMMER INT,
 LOEPNR SMALLINT,
 RAD_NUMMER INT GENERATED ALWAYS
 AS IDENTITY (START WITH 1,
 INCREMENT BY 1,
 CACHE 20))
ON COMMIT PRESERVE ROWS;

INSERT INTO SESSION.GTLAAN_INTERVAL OVERRIDING USER VALUE
SELECT FK_AAB_PERS_CSNNR, LOEPNR, 1
FROM E569DBP.TLAAN
WHERE SLUTREGLERAD = 'N'
GROUP BY FK_AAB_PERS_CSNNR, LOEPNR;

SELECT CSN_NUMMER
FROM SESSION.GTLAAN_INTERVAL
WHERE RAD_NUMMER = (SELECT (MAX(RAD_NUMMER) / 3) + 1 FROM SESSION.GTLAAN_INTERVAL)
OR RAD_NUMMER = (SELECT ((MAX(RAD_NUMMER) / 3) * 2) + 1 FROM SESSION.GTLAAN_INTERVAL);
```

Istället för WITH använder vi en deklarerad GTT

Vi använder en IDENTITY kolumn istället för ROW\_NUMBER() funktionen

Nödvändig för att inte konstanten i select listan ska hamna i konflikt med IDENTITY kolumnen

Vi använder en GROUP BY för att tvinga en sortering



Insert steget har ett inneboende problem eftersom en subselect inte får innehålla en order by, därför använder vi en group by för att tvinga fram en sortering. Om nycklarna inte sorteras kommer intervallen att överlappa.

## Lösningförslag Mainframe DB2 Del 2



CSN\_NUMMER

-----  
15801087  
29665510

2 record(s) selected.

real 2m35.41s

```
SELECT SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 0          AND 15801087 THEN 1 ELSE 0 END) AS INT1,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 15801088 AND 29665510 THEN 1 ELSE 0 END) AS INT2,  
       SUM(CASE WHEN FK_AAB_PERS_CSNNR BETWEEN 29665511 AND 99999999 THEN 1 ELSE 0 END) AS INT3  
FROM E569DBP.TLAAN  
WHERE SLUTREGLERAD = 'N'  
WITH UR;
```

INT1	INT2	INT3
599023	599022	599022

INT1	INT2	INT3
599022	599022	599019

( DB2 UDB LUW V8.1)

1 record(s) selected.

real 1m1.73s

Frågor?

