

DB2 Version 8 Partitioning Scary Stories



GUIDE Share France

Une Association Indépendante d'Utilisateurs IBM



Phil Grainger (CA)

*Réunion du Guide DB2 pour z/OS France
Jeudi 22 novembre 2007
Immeuble CA, Paris-La Défense*



Agenda

- > Increased Number of Partitions
- > What is Table Based Partitioning?
- > Data Partitioned Secondary Indexes
- > Adding Partitions
- > Rotating Partitions
- > Questions?

Increased number of partitions

- > Now up to a potential maximum of 4,096 partitions
- > However, actual maximum depends on page size
- > And also on DSSIZE

Increased number of partitions

DSSIZE	4K pages	8K pages	16K pages	32K pages
1-4 Gb	4,096	4,096	4,096	4,096
8Gb	2,048	4,096	4,096	4,096
16Gb	1,024	2,048	4,096	4,096
32Gb	512	1,024	2,048	4,096
64Gb	256	512	1,024	2,048

Increased number of partitions

- > However, how does the VSAM cluster naming convention work?
- > A001-A999 for partitions 1-999
- > B000-B999 for partitions 1000-1999
- > C000-C999 for partitions 2000-2999
- > D000-D999 for partitions 3000-3999
- > E000-E096 for partitions 4000-4096

Increased number of partitions

- > Also be careful of the limit to the number of objects in a database....
 - Still 65,535
- > A 4,096 partition tablespace with LOB columns needs:
 - 4,096 tablespace objects
 - And (4,096 LOB tablespace, aux table and aux index = 12,288) objects per LOB column
 - So only room for FOUR LOB columns
 - NOT the FIVE that the IBM doc states – they seem to have forgotten the base tablespace objects

Table Based Partitioning

- > Before we can look at DPSIs, ROTATING and ADDing partitions
- > Version 8 introduced TABLE BASED PARTITIONING
- > What we had before was INDEX BASED PARTITIONING
- > Version 8 will support BOTH
 - But can almost invisibly switch from index-based to table-based.....

Table Based Partitioning

- > A partitioned table NO LONGER needs a partitioning index
- > The table itself is defined as partitioned with the relevant key ranges
 - Note that the tablespace in which the table will reside must also have the same number of partitions

```
CREATE TABLE table_name (col1, col2, col3),  
    PARTITION BY (col1, col2),  
    ( PART 1 VALUES (aaa,xxx),  
  
    PART n VALUES (yyy,zzz);
```

Case Study V8 Partitioning

PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2003/12/30 07:17
COMMAND ==> SCROLL ==> PAGE
User ID: RASST02

-DIS DB(STEENDB) SPACE(PART9*)

***** TOP OF DATA *****

NAME	TYPE	PART	STATUS	PHYERRLO	PHYERRHI	CATALOG	PIECE
PART9TS	TS	0001	RW				
	-THRU	0009					

***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I !D81A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****

PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2003/12/30 07:52
COMMAND ==> SCROLL ==> PAGE
User ID: RASST02

-sta DB(STEENDB) SPACE(PART9ts) part(4) access(R0)

-DIS DB(STEENDB) SPACE(PART9TS)

***** TOP OF DATA *****

NAME	TYPE	PART	STATUS	PHYERRLO	PHYERRHI	CATALOG	PIECE
PART9TS	TS	0001	RW				
	-THRU	0003					
PART9TS	TS	0004	R0				
PART9TS	TS	0005	RW				
	-THRU	0009					

***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I !D81A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****

DB2 Command output changed too.

Adjacent partitions with same status are grouped together – and tied by "-THRU" keyword

Connected to tpx.ca.com port 23 2/15 13:17:36 IBM-3279-4-E - A55T7501

Connected to tpx.ca.com port 23 2/15 13:53:05 IBM-3279-4-E - A55T7501

Case Study V8 Partitioning - Display

```
TPX
QWS3270 Edit View Options Tools Help
----- DB2 Command Processor ----- 2004/01/16 17:26
COMMAND ==> SCROLL ==> PAGE
User ID: RASST02
-----
-DIS DB(STEENDB) OVERVIEW
-----
***** TOP OF DATA *****
DSNT360I  !D81A *****
DSNT361I  !D81A * DISPLAY DATABASE SUMMARY
          * OVERVIEW
DSNT360I  !D81A *****
DSNT362I  !D81A DATABASE = STEENDB STATUS = RW
          DBD LENGTH = 12104
DSNT397I  !D81A
NAME      TYPE PARTS
-----
PART2TS  TS      0004
PART5    TS      0005
PART9TS  TS      0013
PART1CTD IX      0013
PART1VRY IX      D0013
PART9TSR IX      L0013
***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I  !D81A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****
-----
Connected to tpx.ca.com port 23 3/15 23:21:07 IBM-3279-4-E - A55T5314
```

Since partitions are displayed in LOGICAL PARTITION sequence – this new command can be helpful to get the overview.

Table Based Partitioning

- > Notice also, that because partitioning does not use an index, CLUSTERing is not related to PARTITIONing any more
- > Conversion happens when ANY ALTER is issued that is valid for table based partitioning
 - ALTER INDEX NOT CLUSTER for the partitioning index
 - ALTER TABLE ADD PART
 - CREATE INDEX PARTITIONED
 - Etc.
- > IBM suggest CREATE a PARTITIONED SECONDARY INDEX (DPSI) with the DEFINE NO, DEFER YES keywords
 - This does no work really and does not affect availability
- > And then DROP it

Table Based Partitioning

- > With INDEX BASED partitioning all secondary indexes could contain keys for any partition
 - This severely affects availability and operation of utilities
- > With TABLE BASED partitioning secondary indexes can also be partitioned on the same keys
 - However they can NOT be UNIQUE.....

Table Based Partitioning

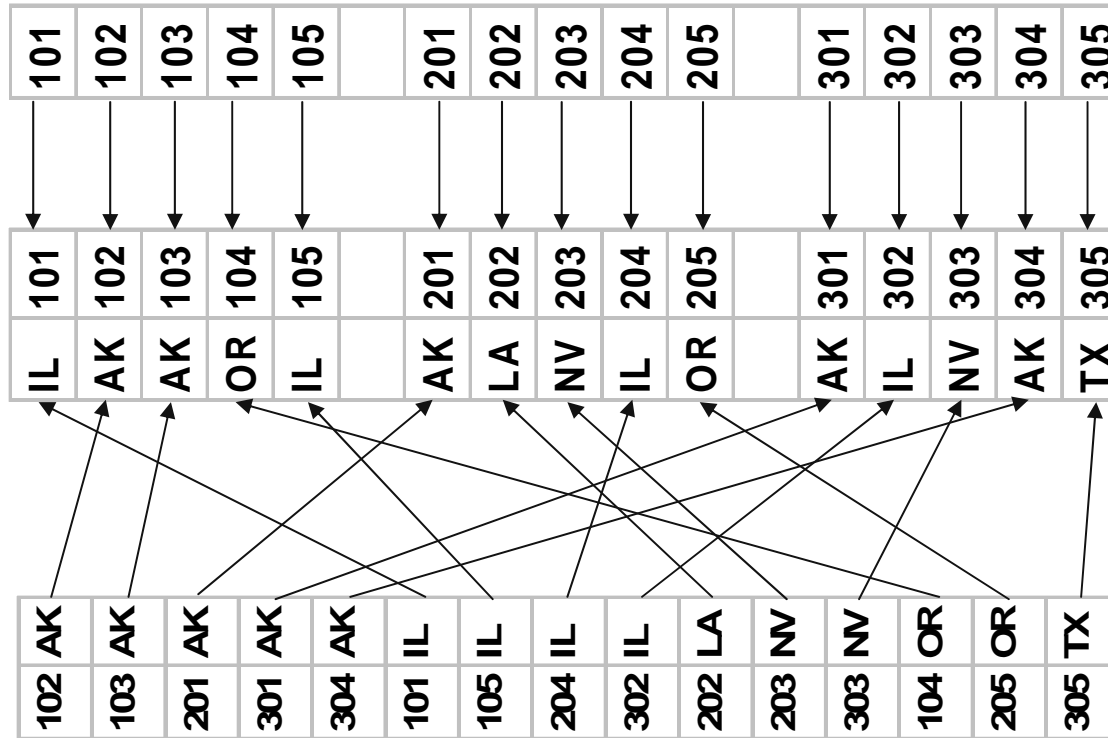
> Time for some pictures.....

Version 7 partitioned table

Partitioned
by customer
number

Therefore
also clustered
by customer
number

Secondary
index on
state



Version 8 partitioned table

NOTE
No partitioning
index

Still partitioned
by customer
number, and
clustered
likewise

IL	101	AK	102	AK	103	OR	104	IL	105		AK	201	LA	202	NV	203	IL	204	OR	205		AK	301	IL	302	NV	303	AK	304	TX	305
----	-----	----	-----	----	-----	----	-----	----	-----	--	----	-----	----	-----	----	-----	----	-----	----	-----	--	----	-----	----	-----	----	-----	----	-----	----	-----

Version 8 partitioned table

NOTE
No partitioning
index

Still partitioned
by customer
number, now
clustered by
state!

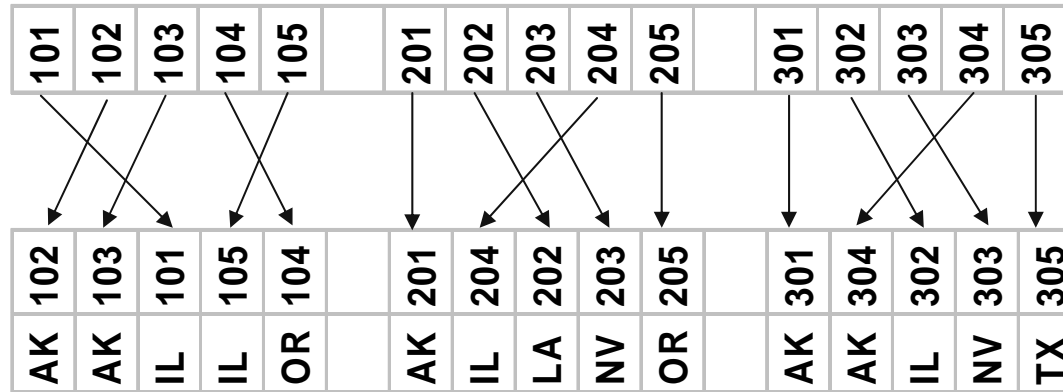
AK	102	AK	103	IL	101	IL	105	OR	104		AK	201	IL	204	LA	202	NV	203	OR	205		AK	301	AK	304	IL	302	NV	303	TX	305
----	-----	----	-----	----	-----	----	-----	----	-----	--	----	-----	----	-----	----	-----	----	-----	----	-----	--	----	-----	----	-----	----	-----	----	-----	----	-----

To be clustered on state there must also be an
Index on state – this is omitted for clarity

Version 8 partitioned table

With a partitioned index on customer

Still partitioned by customer number, now clustered by state!



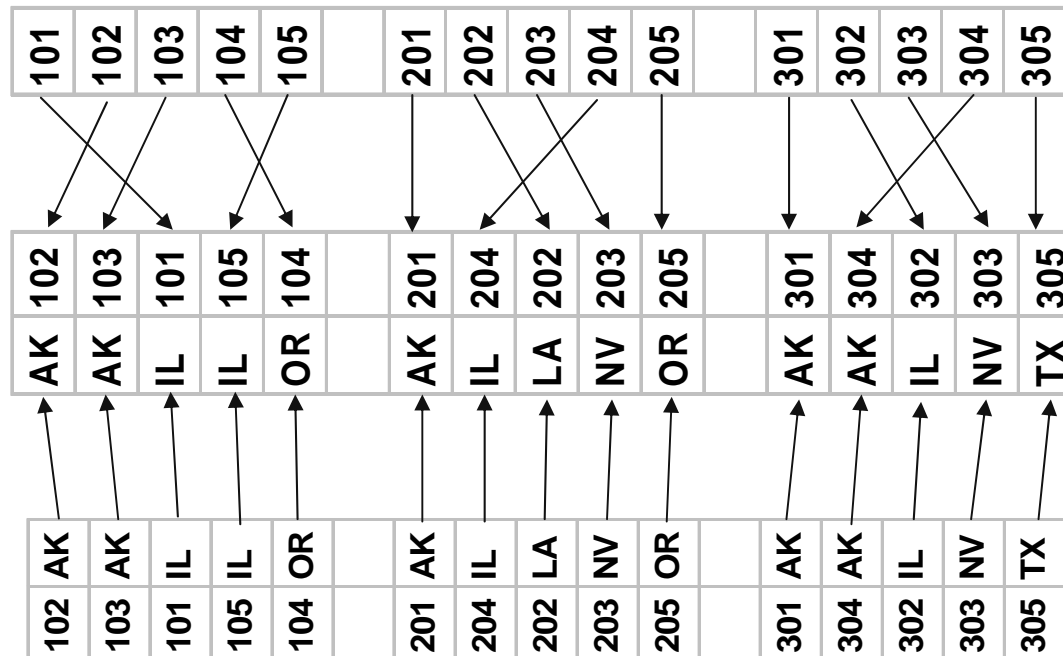
Any index that has the SAME columns in the SAME sequence as the partitioning key is called a PARTITIONING index. Partitioning indexes can be unique or non-unique

Version 8 partitioned table

With a partitioned index on customer

Still partitioned by customer number, now clustered by state!

DPSI on state, customer



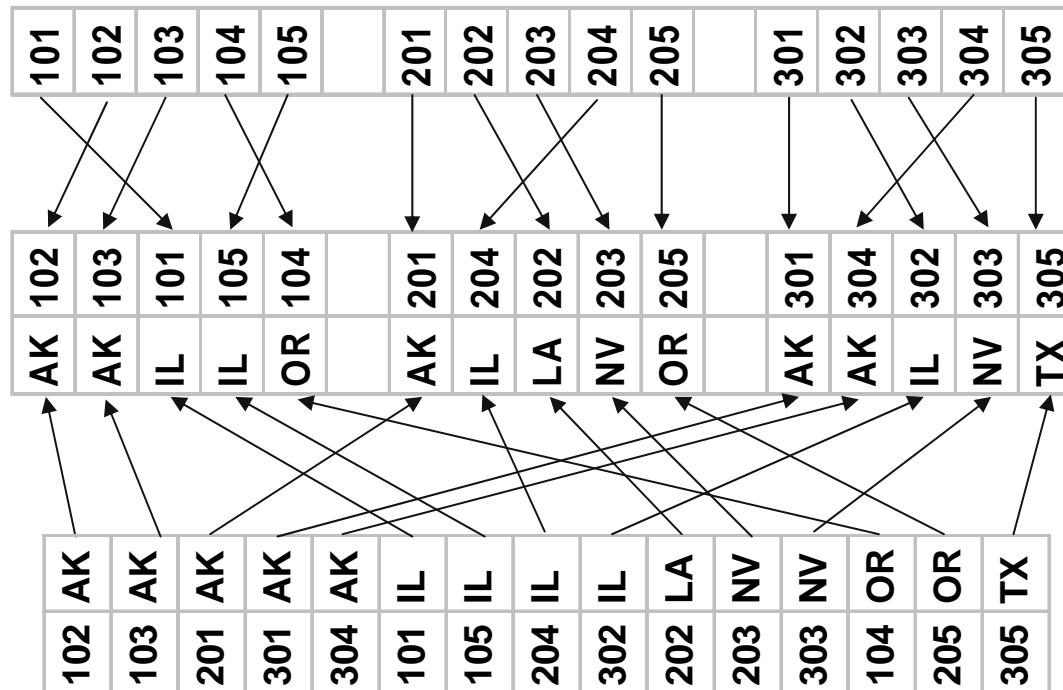
Data Partitioned Secondary Indexes are physically partitioned in the same ranges as the table, but MUST be non-unique

Version 8 partitioned table

With a partitioned index on customer

Still partitioned by customer number, now clustered by state!

Traditional NPI



If your secondary index has to be unique then it CANNOT be a DPSI – it's a traditional NPI

Case Study V8 Partitioning - INDEXes

```

TPX
QW53270 Edit View Options Tools Help
BPPAUDT R11.00 ----- AUDIT Message File ----- 2004/01/13 17:11:42
COMMAND ==>
-----
CREATE INDEX RASST02.PART9TS_NONPARTIX1
ON RASST02.TABLEPART_PART9
(COLUMN003 ASC
)
USING STOGROUP SYSDEFLT
PRIQTY 48
SECQTY 48
ERASE NO
FREEPAGE 0
PCTFREE 10
CLUSTER
BUFFERPOOL BP1
CLOSE YES
PIECESIZE 1M ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CREATE INDEX RASST02.PART9TS_NONPARTDPSI1
ON RASST02.TABLEPART_PART9
(COLUMN005 ASC
)
USING STOGROUP SYSDEFLT
PRIQTY 12
SECQTY 12
ERASE NO
FREEPAGE 0
PCTFREE 10
BUFFERPOOL BP1
CLOSE YES PARTITIONED :
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
COMMIT WORK ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
CONTROL PROCESS CARD:
BATCH PROCESSOR PLAN ==> RBPA720
BATCH PROCESSOR ID ==> (NONE)
STRATEGY NAME ==> (NONE)
DEVICE ALLOCATION UNIT => SYSDA
***** BATCH PROCESSOR COMPLETE *****
***** BOTTOM OF DATA *****
PF7/8: Scroll Up/Down PF3/15: End
Connected to tpx.ca.com port 23 2/15 23:11:46 IBM-3279-4-E - A55T8958
  
```

Create non partitioned index.
Cluster is completely separated.

Create DPSI

```

TPX
QW53270 Edit View Options Tools Help
PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2004/01/13 17:13
COMMAND ==>
-----
--DIS DB(STEENDB)
-----
***** TOP OF DATA *****
DSNT360I !D81A *****
DSNT361I !D81A * DISPLAY DATABASE SUMMARY
* GLOBAL
DSNT360I !D81A *****
DSNT362I !D81A DATABASE = STEENDB STATUS = RW
DBD LENGTH = 8066
DSNT397I !D81A
NAME TYPE PART STATUS PHYERRLO PHYERRHI CATALOG PIECE
-----
PART5 TS 0002 RW
-THRU 0005
PART6 TS 0004 RW
-THRU 0007
PART9TS TS 0003 RW
-THRU 0009
PART9TS TS 0001 RW
PART9TS TS 0010 RW
-THRU 0012
PART9TS TS 0002 RW
PART9TS TS 0013 RW
PART1VRY IX D0001 RW
-THRU 0013
PART9TSR IX L* RW
***** DISPLAY OF DATABASE STEENDB END *****
DSN9022I !D81A DSNTDDIS 'DISPLAY DATABA
***** BOTTOM *****
Connected to tpx.ca.com port 23 5/15 23:13:58 IBM-3279-4-E - A55T8958
  
```

DPSI partitions can
be identified by Dxxx

NPI – maintains the L
to illustrate the piece's
which is not relevant
for partitioned.

Case Study V8 Partitioning

```
TPX
QW53270 Edit View Options Tools Help
-----
BPPAUDT R11.00 ----- AUDIT Message File ----- 2004/01/14 18:41:43
COMMAND ==> | SCROLL ==> CSR
-----
USER ID: RASST02
ALTER INDEX RASST02.PART9TS_NONPARTIX1 NOT CLUSTER ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION

COMMIT;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION

CREATE TYPE 2 INDEX RASST02.PART9TS_PARTIX
ON RASST02.TABLEPART_PART9
(COLUMN001 ASC ,COLUMN004 DESC )
CLUSTER PARTITIONED
(PART 1 USING STOGROUP SYSDEFLT
,PART 2 USING STOGROUP SYSDEFLT
,PART 3 USING STOGROUP SYSDEFLT
,PART 4 USING STOGROUP SYSDEFLT
,PART 5 USING STOGROUP SYSDEFLT
,PART 6 USING STOGROUP SYSDEFLT
,PART 7 USING STOGROUP SYSDEFLT
,PART 8 USING STOGROUP SYSDEFLT
,PART 9 USING STOGROUP SYSDEFLT
,PART 10 USING STOGROUP SYSDEFLT
,PART 11 USING STOGROUP SYSDEFLT
,PART 12 USING STOGROUP SYSDEFLT
,PART 13 USING STOGROUP SYSDEFLT
)
BUFFERPOOL BP1

-----
Connected to tpx.ca.com port 23 3/15 00:41:50 IBM-3279-4-E - A55T0064
```

Before we can make a partitioned cluster index which matches the table partitions, we will have to remove the existing cluster index.

When the columns match the table partitioning statement AND the two keywords CLUSTER and PARTITIONED are used – then you get a partitioned cluster index for the table (almost like we're used to).

Remember that VALUES can no longer be specified on the index create since they are specified on the create TABLE statement.

Case Study V8 Partitioning

```
TPX
QWS3270 Edit View Options Tools Help
PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2004/01/14 18:51
COMMAND ==> SCROLL ==> PAGE
User ID: RASST02
-----
- DIS DB(STEENDB)
-----
***** TOP OF DATA *****
DSNT360I !D81A *****
DSNT361I !D81A * DISPLAY DATABASE SUMMARY
* GLOBAL
DSNT360I !D81A *****
DSNT362I !D81A DATABASE = STEENDB STATUS = RW
DBD LENGTH = 12104
DSNT397I !D81A
NAME TYPE PART STATUS PHYERRLO PHYERRHI CATALOG PIECE
-----
PART2TS TS 0002 RW
PART2TS TS 0001 RW
PART2TS TS 0003 RW
-THRU 0004
PART5 TS 0002 RW
-THRU 0005
PART5 TS 0001 RW
PART9TS TS 0003 RW
-THRU 0009
PART9TS TS 0001 RW
PART9TS TS 0010 RW
-THRU 0012
PART9TS TS 0002 RW
PART9TS TS 0013 RW
PART10TD IX 0004 RW
-THRU 0013
PART1VRY IX D0001 RW
-THRU 0013
PART9TSR IX L* RW
***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I !D81A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****
```

Our object now has FOUR different types in the display output:

- The tablespace itself.
- The cluster index matching the table partitions (no prefix).
- The DPSI index partitions (prefix = D).
- The NPI (prefix L as usual).



Adding Partitions

- > A partition can be added at the end of an existing partitioned table
- > But only up to the maximum number of partitions (see earlier)
- > The partition number itself is not specified
 - It is chosen by DB2
- > But the partition keys must, of course, be specified

Adding Partitions

- > For STOGROUP defined objects, DB2 will automatically define the relevant pagesets
 - For the tablespace and for all partitioned indexes
 - DB2 uses the allocation quantities for the immediately preceding partition

- > As soon as the partition is added, it is immediately available when the pagesets are restarted

Adding Partitions

- > All plans, packages and cached dynamic SQL statements are invalidated
- > Only ONE partition can be added in a single unit of work
- > We now need to be aware of a VERY subtle difference between LOGICAL partition number and PHYSICAL partition number
- > LOGICAL refers to the partition number of the TABLE (effectively)
- > PHYSICAL refers to the Annn node of the pageset name

Adding Partitions

- > For example, if we start with a 10 partition table
- > The table will have 10 partitions(!)
- > The underlying tablespace will also have 10 partitions
 - They will be named A001 – A010
- > So far so good.....
 - But what happens when we MOVE partitions...?

Catalog Changes for V8 Partitioning

```
TPX
QWS3270 Edit View Options Tools Help
RUBROWC R11.00 ----- RC/Browse: Column Mode ----- 2004/01/06 08:44:09
COMMAND ==>
For Table ==> SYSIBM.SYSCOPY Row number==> 1 OF 6
Browse Mode ==> Max Char ==> 070
SSID: D81A ----- FETCH STATUS: COMPLETE ----- RASST02
TSNAME DSNUM ICDATE STYPE OLDEST_VERSION LOGICAL_PART
PART9TS 13 040106 A 0 13
PART9TS 2 040106 R 0 1
PART9TS 12 040106 A 0 12
PART9TS 11 040106 A 0 11
PART9TS 10 031231 A 0 10
PART9TS 1 031231 R 0 1
***** BOTTOM OF DATA *****
```

STYPE :
"R" and "A" for rotated and added partitions.

LOGICAL PART: Describes the logical partition where "STYPE Function" was performed for physical dataset DSNUM.

- 1) Logical PART=1 with DSNUM=1 was rotated
- 2) P10+P11+P12 were added
- 3) Logical PART=1 with DSNUM=2 was rotated (DSNUM=1 rotated earlier)
- 4) P13 was added

Rotating Partitions

- > DB2 Version 8 also allows partition ROTATING
- > This is where the partition that is currently the "first" (the one with the lowest keys) is Rotated to become the "last" (which will now contain the highest keys)
- > All rows will be deleted from the Rotated partition
 - And all SYSCOPY & SYSLGRNX rows for it will be purged
- > Also, the new limit keys for the "new" partition will be specified on the ROTATE statement

Rotating Partitions

- > When we Rotate FIRST to LAST, we effectively remove the old first partition and what was the SECOND one now becomes the first
- > Remember, partition keys are only specified ONCE for each partition
 - We list the HIGHEST key that can reside in any partition
- > So, after ROTATING a row that would have been placed into PART 1 will now be placed into PART 2 instead!

Rotating Partitions

- > Remember that logical vs physical bit a few pages back?
- > Now we start with the same 10 partition table
 - That has physical datasets A001 – A010
- > Now we “ROTATE FIRST TO LAST”
 - Logically we still have partitions 1-10 (it’s just that the old part 2 is now part 1 etc)
 - BUT DB2 doesn’t rename the pagsets
 - So logical partition 1 is now physical pageset A002
 - Logical partition 9 is now physical pageset A010
 - And logical partition 10 is physical pageset A001(!)

Case Study V8 Partitioning

TPX QW53270 Edit View Options Tools Help

BPPAUDT R11.00 ----- AUDIT Message File ----- 2003/12/31 07:33:32
 COMMAND ==> SCROLL ==> PAGE

----- USER ID: RASST02 -----

PARTITION ROTATE DDL:

More than one partition can be Rotated in a single UOW (unit of work) but Only one partition in one statement.

```

ALTER TABLE RASST02.TABLEPART_PART9 ALTER PART
ROTATE FIRST TO LAST VALUES ('X500000000') RESET ;
DSNT400I  SQLCODE = 000, SUCCESSFUL EXECUTION
  
```

PF7/8: Scroll Up/Down PF9/15: End

Connected to tpx.ca.com port 23 2/15 13:33:44 IBM-3279-4-E - A55T7958

TPX QW53270 Edit View Options Tools Help

PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2003/12/31 07:35
 COMMAND ==> SCROLL ==> PAGE

----- User ID: RASST02 -----

```

- -DIS DB(STEENDB) SPACE(PART9*)
- - - - -
***** TOP OF DATA *****
DSNT360I  ?D81A *****
DSNT361I  ?D81A * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I  ?D81A *****
DSNT362I  ?D81A DATABASE = STEENDB STATUS = RW
          DBD LENGTH = 4028
DSNT397I  ?D81A
NAME      TYPE PART STATUS PHYERRLO PHYERRHI CATALOG PIECE
-----
PART9TS  TS      0002 RW
          -THRU  0009
PART9TS  TS      0001 RW
***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I  ?D81A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****
  
```

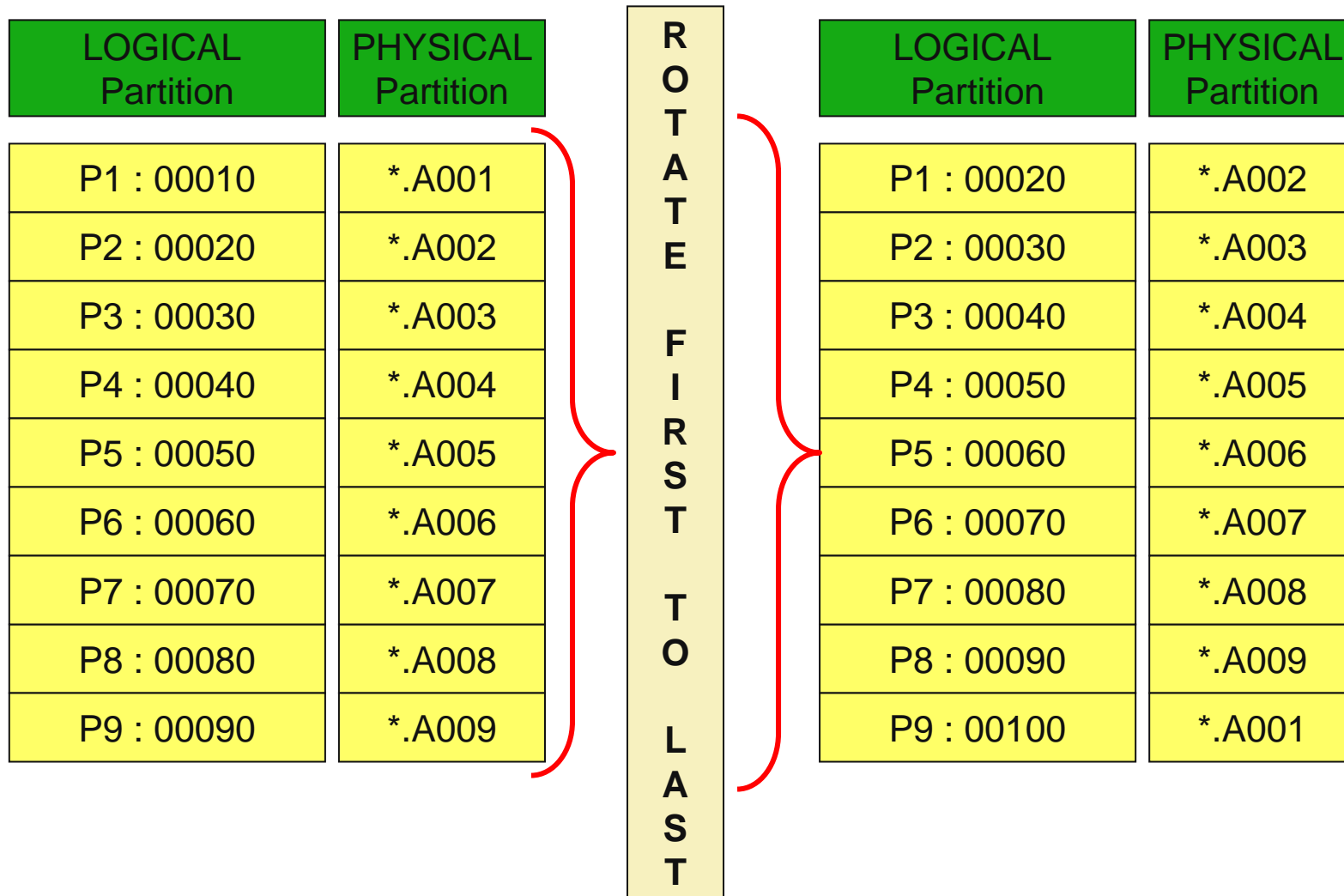
PARTITION ROTATE:

The DISPLAY changes again due to the partition Rotate - always displayed in LOGICAL order (limit key sequence).

Connected to tpx.ca.com port 23 2/15 13:35:49 IBM-3279-4-E - A55T7958

Case Study V8 Partitioning

> Rotating a partition (here limitkey is char(5)):



Rotate One Partition!

```
.CONNECT D81A
BPA0027I: SYNCPOINT TAKEN AND ALL WORK COMMITTED.
RETCODE =      0

ALTER TABLE RASST02.PART1 ALTER PART ROTATE FIRST TO LAST
VALUES ('1000000000') RESET ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
INSERT INTO RASST02.PART1
(COLUMN001,COLUMN002,COLUMN003,COLUMN004,COLUMN005)
VALUES ('0310000000' , 1234567 , 220000000 , '2004-01-01' ,
'steen ras SENIOR 1 ' );
DSNT408I SQLCODE = -910, ERROR: THE SQL STATEMENT CANNOT ACCESS AN
OBJECT ON WHICH A DROP OR ALTER IS PENDING
DSNT418I SQLSTATE = 57007 SQLSTATE RETURN CODE
DSNT415I SQLERRP = DSNXEDP SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD = -303 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'FFFFFFED1' X'000000000' X'000000000'
X'FFFFFFFFF' X'000000000' X'000000000' SQL DIAGNOSTIC
INFORMATION
BPA0012E: DB2 SQL/DDL ERROR HAS OCCURRED - ROTATEBACK ISSUED.
RETCODE =      8
```

ROTATE one PARTITION is legal - data still deleted !!!

Rotate and Rollback

```
ALTER TABLE RASST02.PART1 ALTER PART ROTATE FIRST TO LAST
VALUES ('1400000000') RESET ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
ALTER TABLE RASST02.PART1 ALTER PART ROTATE FIRST TO LAST
VALUES ('1500000000') RESET ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
Rollback ;
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
INSERT INTO RASST02.PART1
(COLUMN001,COLUMN002,COLUMN003,COLUMN004,COLUMN005)
VALUES ('0310000000' , 1234567 , 220000000 , '2004-01-01' ,
'steen ras SENIOR 1 ' );
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
```

ROLLBACK keeps the OLD records (of course).

Rotating AND Adding

- > This is where it will get messy
- > Start
 - with Parts 1-10 (A001-A010)
- > Rotate first to last
 - Now we have A002-A010 followed by A001
- > Add a partition
 - Now there's A002-A010, then A001, finally A011
- > Got all that???
- > LOGICAL and PHYSICAL partition numbers are no longer guaranteed to be the same....

REORG enhancements

- > REORG can now be used to rebalance partitions
- > The rows in the selected (or all) partitions will be redistributed in equal measure across the partitions
 - Eg REORG TABLESPACE dbname.tsname PART 12:16 REBALANCE
 - Will rebalance partitions 12 through 16
 - However, this CANNOT be done as a SHRLEVEL CHANGE (on-line) REORG
 - If the clustering key is NOT the partitioning key, REORG must be run twice
 - Once to rebalance and once to sort into clustering sequence within partition!

REORG Enhancements

- > There is no BUILD2 phase for DPSIs
- > SHRLEVEL CHANGE and DISCARD can now be specified together
 - BUT if any discarded rows are modified during the reorg, then the reorganisation willabend

Case Study V8 Partitioning - REORG

```
TPX
QW53270 Edit View Options Tools Help
PTPDBCDD R11.00 ---- DB2 Command Processor ---- 2004
COMMAND ==>
-----
- DIS DB(STEENDB)
-----
***** TOP OF DATA *****
DSNT360I !D81A *****
DSNT361I !D81A * DISPLAY DATABASE SUMMARY
* GLOBAL
DSNT360I !D81A *****
DSNT362I !D81A DATABASE = STEENDB STATUS = RW
DBD LENGTH = 12104
DSNT397I !D81A
NAME TYPE PART STATUS PHYERRLO PHYERRHI CATALOG PIECE
-----
PART2TS TS 0002 RW
PART2TS TS 0001 RW
PART2TS TS 0003 RW
-THRU 0004
PART5 TS 0002 RW
-THRU 0005
PART5 TS 0001 RW
PART9TS TS 0003 RW,COPY
PART9TS TS 0004 RW
-THRU 0009
PART9TS TS 0001 RW
PART9TS TS 0010 RW
-THRU 0012
PART9TS TS 0002 RW
PART9TS TS 0013 RW
PART1CTD IX 0001 RW
-THRU 0013
PART1VRY IX D0001 RW
-THRU 0013
PART9TSR IX L* RW
***** DISPLAY OF DATABASE STEENDB ENDED *****
DSN9022I !D81A DSNTDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** BOTTOM OF DATA *****
```

```
REORG TABLESPACE STEENDB.PART9TS PART 3
SORTDEVT SYSDA SORTNUM 3 LOG NO
UNLOAD PHASE STATISTICS -
NUMBER OF RECORDS UNLOADED=5
INDEXES WILL BE BUILT IN PARALLEL,
NUMBER OF TASKS = 6
.....
TABLESPACE STEENDB.PART9TS PARTITION 3
IS IN COPY PENDING
```

The partition number specified in the utility statement is the PHYSICAL partition – not the LOGICAL. The logical partition ONE is now The physical partition THREE.

Case Study V8 Partitioning - REORG

```

TPX
QW53270 Edit View Options Tools Help
----- RC/Q Table Space Partition List ----- 2004/01/15 14:0
COMMAND ==>

DB2 Object ==> TS          Option ==> PL
Table Space ==> PART9TS    Creator ==> RASST02  Where => N
Data Base ==> STEENDB      Grantor ==> *
Loc: LOCAL ----- SSID: D81A LUL: 02 -RASST02 - LINE 01 OF 13 <
CMD      NAME      PART      CARD      LIMITKEY
-----
PART9TS  1          1          1  'X500000000'
          2          0          0  'y700000000'
          3          5          5  'HHHHHHHHHH'
          4          0          0  'KKKKKKKKKK'
          5          0          0  'NNNNNNNNNN'
          6          2          2  'RRRRRRRRRR'
          7          0          0  'TTTTTTTTTT'
          8          0          0  'UUUUUUUUUU'
          9          0          0  'X111111111'
         10         0          0  'X999999999'
         11         0          0  'Y200000000'
         12         0          0  'Y400000000'
         13         0          0  'y800000000'
***** BOTTOM OF DATA **
    
```

Some partitions have too much data so we need to change the limitkeys for the table and reorg the partitions involved to distribute the data evenly. (Data for partition 5,6,7,8)

```

REORG TABLESPACE STEENDB.PART9TS PART (005:008)
REBALANCE SORTDEVT SYSDA SORTNUM 3 LOG NO
(SYSCOPY DD CARD REQUIRED when using REBALANCE)
    
```

DSNU1128I !D81A DSNURBAL-FEWER PAGES THAN PARTS-REBALANCE IGNORED
 DSNU395I DSNURPIB - INDEXES WILL BE BUILT IN PARALLEL
 DSNU400I DSNURBID - COPY PROCESSED FOR TABLESPACE
 STEENDB.PART9TS PART 5:8
 NUMBER OF PAGES=13
 AVERAGE PERCENT FREE SPACE PER PAGE = 13.07
 PERCENT OF CHANGED PAGES =100.00
 ELAPSED TIME=00:00:23



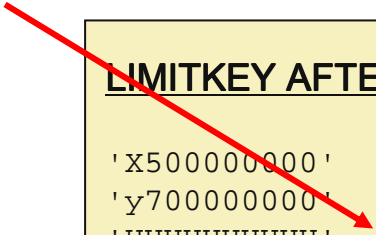
Case Study V8 Partitioning - REBALANCE

- > Once partitions have data – rebalance does work (in some cases.....)
- > Partition 4:6 will be rebalanced
- > Interesting that only FIRST partition key was specified by creation – and now

NAME	DATABASE	PART	CARD-BEFORE	CARD-AFTER
PART9TS	STEENDB	1	1	1
		2	0	0
		3	5	5
		4	199	99
		5	281	100
		6	2	84
		7	0	0
		8	0	0
		9	0	0
		10	0	0
		11	0	0
		12	0	0
		13	0	0

LIMITKEY AFTER REORG:

```
'X5000000000'  
'y7000000000'  
'HHHHHHHHHHH'  
'K1          ', '2004-05-25'  
'K4567898   ', '1997-01-31'  
'RRRRRRRRRR', '0000-00-00'  
'TTTTTTTTTTT'  
'VVVVVVVVVVV'  
'X1111111111'  
'X9999999999'  
'Y2000000000'  
'Y4000000000'  
'y8000000000'
```



REBALANCE and ROTATE

- > Be VERY careful if you rotate partitions and later want to rebalance them
- > Rebalancing can ONLY be carried out on ascending and contiguous partitions
- > What...?
 - I hear you ask

REBALANCE and ROTATE

- > Consider
you have done some adding and rotating and ended up with
physical parts:
2, 3, 4, 5, 6, 1, 7, 8, 9
- > You can rebalance:
2:6
7:9
- > But not
5:8
or even all of them

REORG and Many Partitions

- > Because of limitations internally, REORG can ONLY operate on a maximum of 254 partitions at a time if they are compressed
- > So, a 255 partition, compressed, tablespace CANNOT be reorganised in one go
- > There IS an open APAR to fix this
 - But it is STILL open

Phew

> Any questions.....

Bibliography

- > Keep up to date with the IBM web site
- > The Version 8 Technical Preview (SG24-6871)
- > And look for SG24-6079
"DB2 for z/OS Version 8 Everything You Wanted to Know... and More"