



Table Controlled Partitioning

The rules of the game

Kurt Struyf, Competence Partners

Réunion du Guide DB2A pour z/OS France
Jeudi 31 janvier 2008
Hôtel Mercure, Paris-La Défense





Bio Kurt Struyf

Kurt Struyf started his career at a major Belgian bank. Where he was part of the system DBA team. He worked for an outsourcing company where he installed, tuned and migrated DB2 systems for multiple customers. He was also directly involved in the design and tuning of DB2 related applications and structures.

Currently Kurt is working as a consultant/contractor for Competence Partners. He has over ten years experience as a (system) DBA and has been installing, migrating, troubleshooting and tuning DB2 systems throughout Europe.

Besides his consultancy missions he has been teaching a broad spectrum of DB2 courses through IBM education services, both in Europe and the USA.

These courses range from basic courses like DB2 fundamentals, SQL workshops, application programming, database administration workshops, over DB2 application data recovery and application Performance and Tuning, towards more advanced classes like DB2 System Administration, System Performance Analysis, System/ Disaster Recovery Workshop etc

He was a speaker at several IDUG European and North America conferences . He's been a speaker at numerous DB2 regional user groups in Europe.

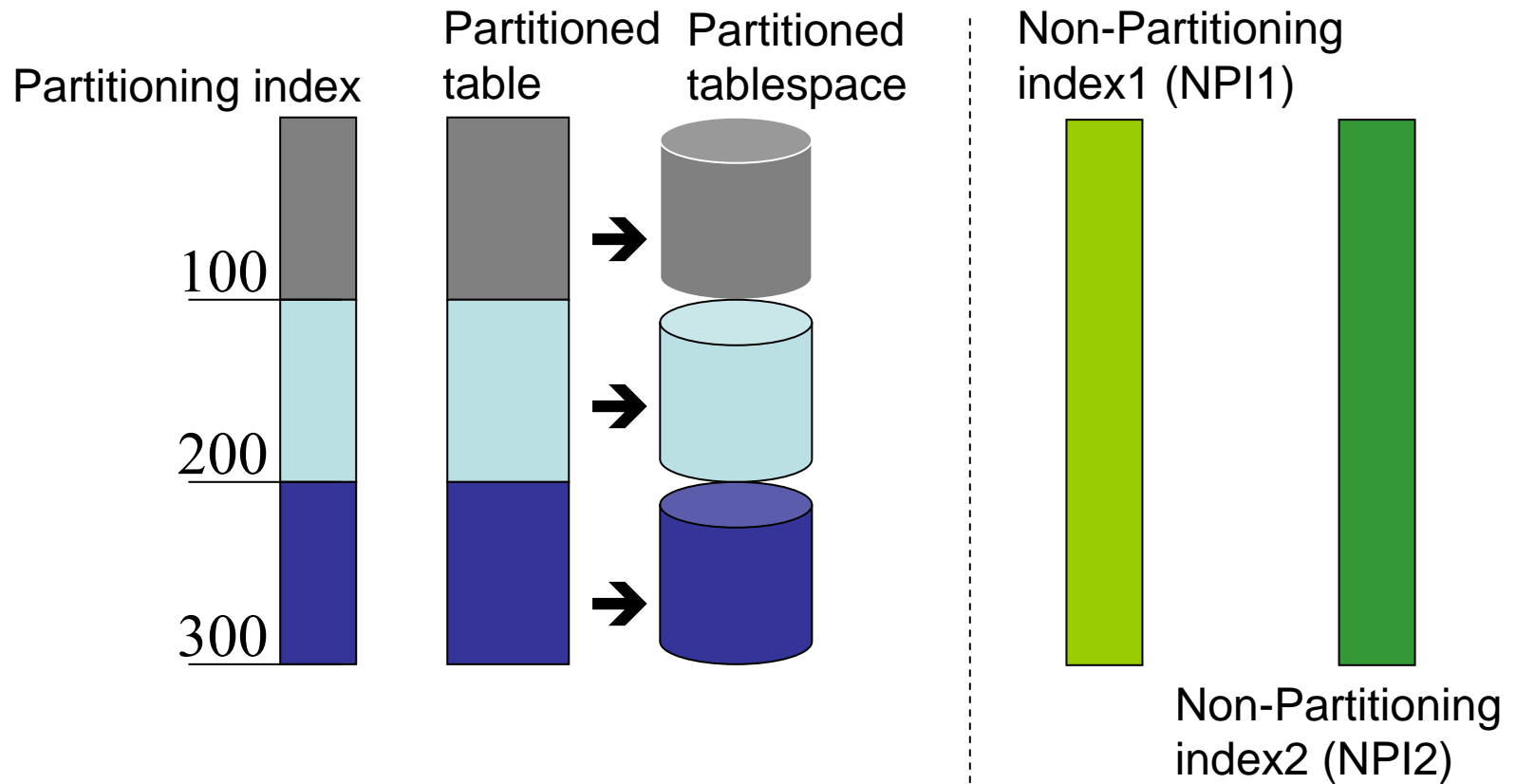


Agenda



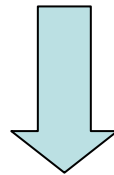
- **Partitioning Pre-V8**
- Partitioning during V8
- Consequences of table controlled partitioning
- Partition Management
 - Add partition
 - Rotate partition
 - Reorg rebalance
- Partitioning during V9

Before Version 8



Syntax Before Version 8

```
CREATE TABLESPACE Z9PARTTS IN ZTRUYKDB NUMPARTS 3 ;
```



```
CREATE TABLE TRUYK.Z9PARTTB  
(COL1 SMALLINT,  
COL2 CHAR(2))  
IN ZTRUYKDB.Z9PARTTS ;
```

INCOMPLETE

```
SYSIBM.SYSTABLES  
Column STATUS='I'
```

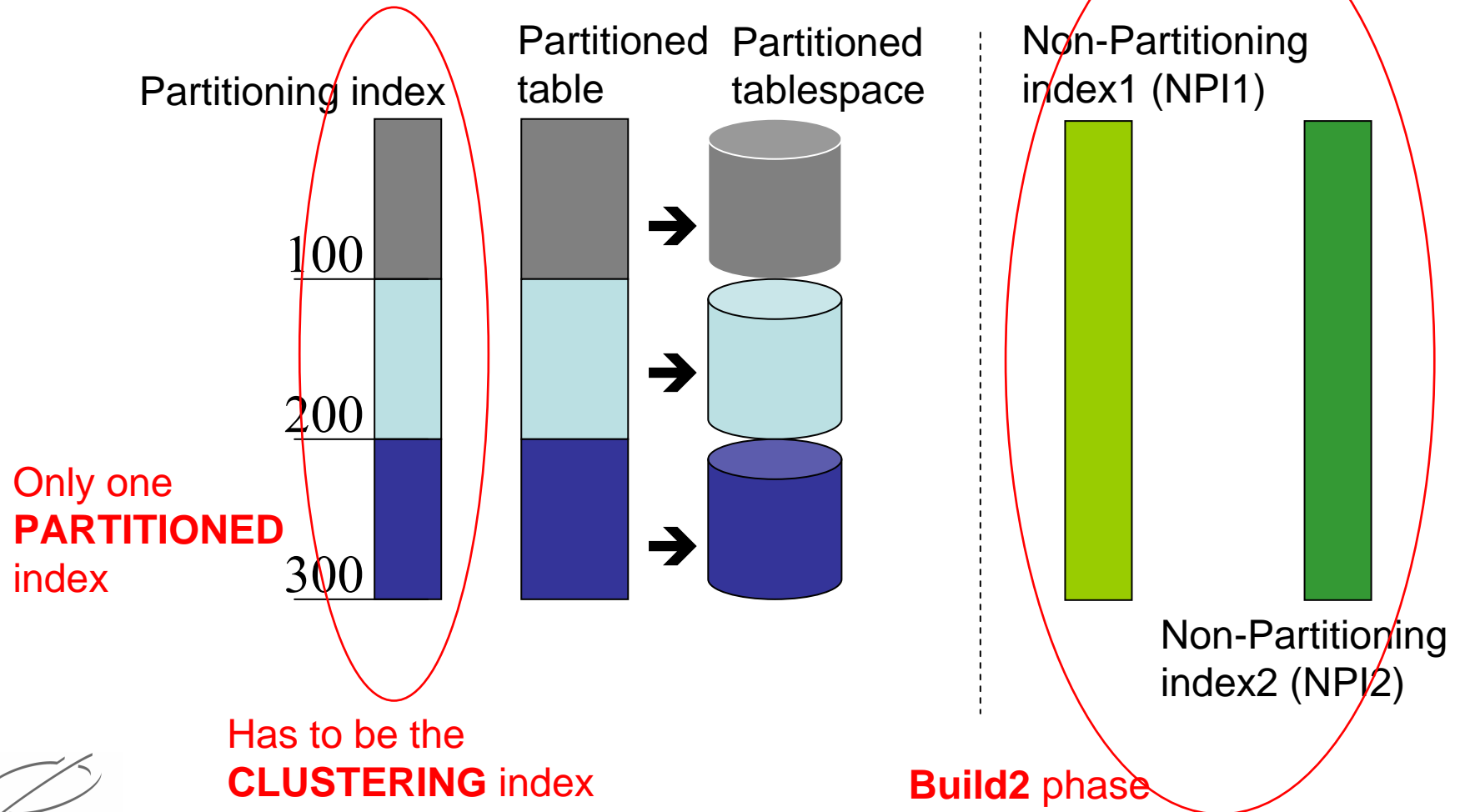
Syntax Before Version 8

```
CREATE INDEX TRUYK.Z9PARTI1
  ON TRUYK.Z9PARTTB ( COL1    ASC )
  CLUSTER
  ( PART    1 VALUES ( 100)
    PART    2 VALUES ( 200)
    PART    3 VALUES ( 300))
```

SYSIBM.SYSINDEXES
Column TYPE = '2'

SYSIBM.SYSINDEXPART
Column LIMITKEY = 100
Column LIMITKEY = 200
Column LIMITKEY = 300

Problem Before Version 8





Agenda



- Partitioning Pre-V8
- **Partitioning during V8**
- Consequences of table controlled partitioning
- Partition Management
 - Add partition
 - Rotate partition
 - Reorg rebalance
- Partitioning during V9

You have a choice

INDEX CONTROLLED

```
CREATE TABLE TRUYK.Z9PARTTB
( EMPNO SMALLINT,
  FNAME CHAR(10)
  LASTNAME CHAR(10)
  ADDRESS CHAR(20)
  CITY CHAR(10)
  STATE CHAR(2))
  IN ZTRUYKDB.Z9PARTTS ;

-----
CREATE INDEX TRUYK.Z9PARTI1
ON TRUYK.Z9PARTTB
( EMPNO ASC )
CLUSTER
PARTITION BY RANGE (
  PARTITION 1 ENDING AT ( 100)
  PARTITION 2 ENDING AT ( 200)
  PARTITION 3 ENDING AT ( 300))
```

TABLE CONTROLLED

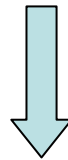
```
CREATE TABLE TRUYK.Z9PARTTB
( EMPNO SMALLINT,
  FNAME CHAR(10)
  LASTNAME CHAR(10)
  ADDRESS CHAR(20)
  CITY CHAR(10)
  STATE CHAR(2))
PARTITION BY (EMPNO ASC)
(PARTITION 1 ENDING AT ( 100),
 PARTITION 2 ENDING AT ( 200),
 PARTITION 3 ENDING AT ( 300))
IN ZTRUYKDB.Z9PARTTS ;
```



COMPLETE

What has changed in Version 8

Table controlled partitioning means :
NO INDEX needed to control partitioned table



- Any index can become the clustering index.
- Any index can be PARTITIONED.
- An index is qualified as partitioning or secondary.

Clustering index

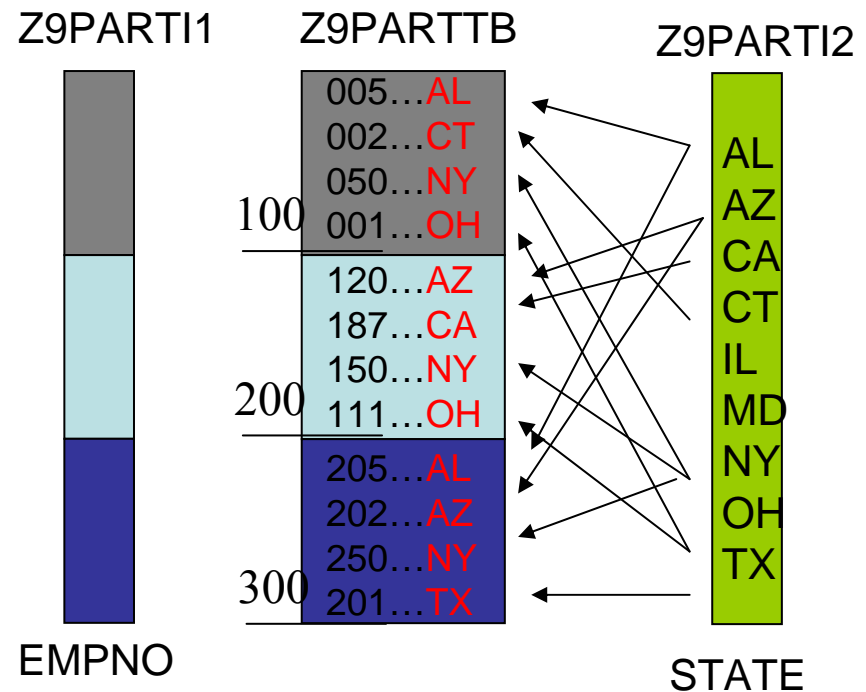
```

CREATE TABLE TRUYK.Z9PARTTB
( EMPNO SMALLINT,
  FNAME CHAR(10)
  LASTNAME CHAR(10)
  ADDRESS CHAR(20)
  CITY CHAR(10)
  STATE CHAR(2))
PARTITION BY (EMPNO ASC)
(PARTITION 1 ENDING AT ( 100),
 PARTITION 2 ENDING AT ( 200),
 PARTITION 3 ENDING AT ( 300))
IN ZTRUYKDB.Z9PARTTS ;

CREATE INDEX TRUYK.Z9PARTI1
ON TRUYK.Z9PARTTB (EMPNO ASC )
PARTITIONED ;

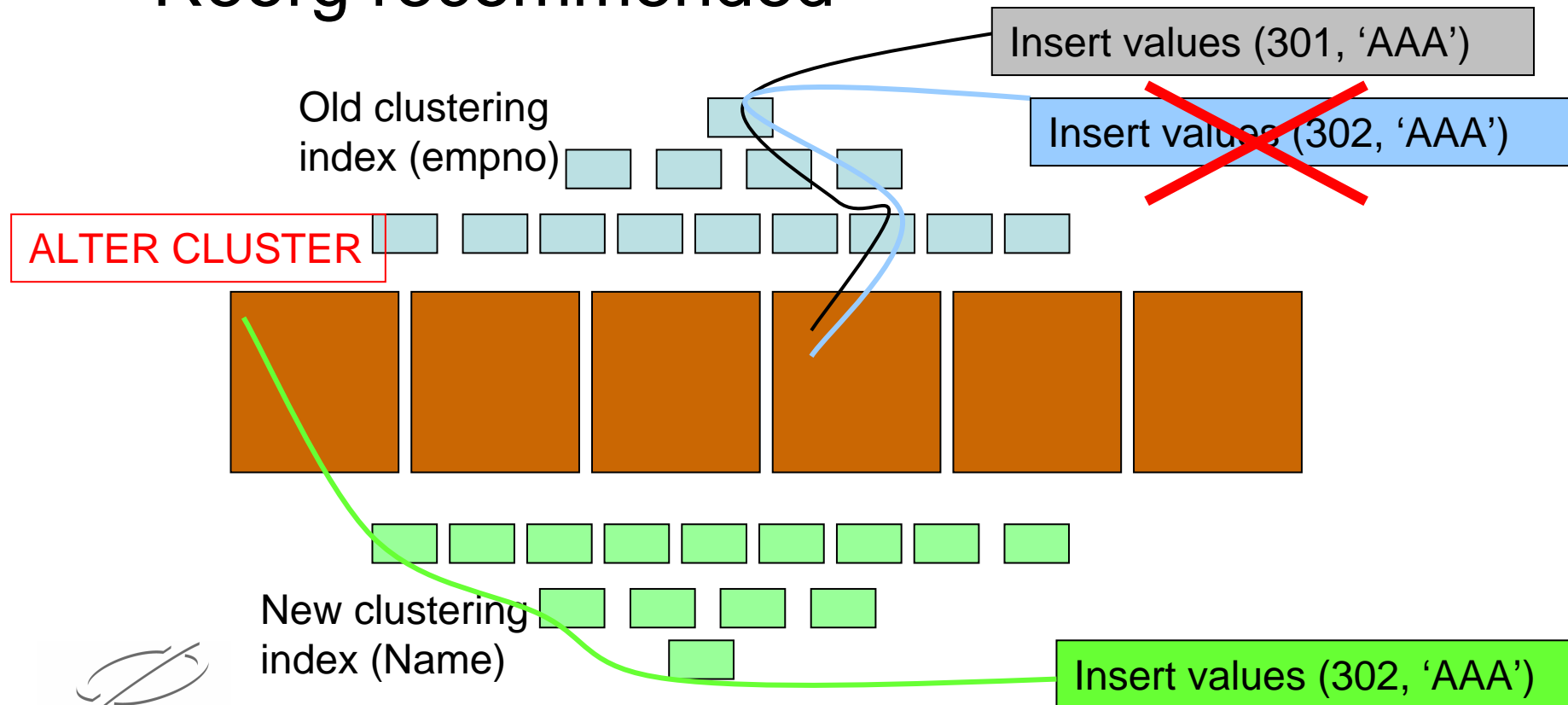
CREATE INDEX TRUYK.Z9PARTI2
ON TRUYK.Z9PARTTB (STATE ASC )
CLUSTER

```



Change Clustering index

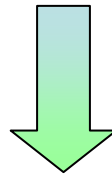
- Active immediately
- Reorg recommended



Partitioning or Secondary

NEW definition of partitioning index

PARTITIONING index
=
FIRST columns of the index,
match the columns of the "PARTITION BY" clause



EVERY other index is a SECONDARY index

Partitioning or Secondary

```
CREATE TABLE TRUYK.Z9PARTTB  
( EMPNO SMALLINT,  
  FNAME CHAR(10)  
  LASTNAME CHAR(10)  
  ADDRESS CHAR(20)  
  CITY CHAR(10)  
  STATE CHAR(2))
```

PARTITION BY (EMPNO ASC)

```
(PARTITION 1 ENDING AT ( 100),  
 PARTITION 2 ENDING AT ( 200),  
 PARTITION 3 ENDING AT ( 300))  
IN ZTRUYKDB.Z9PARTTS ;
```

```
CREATE INDEX TRUYK.Z9PARTI1  
ON TRUYK.Z9PARTTB (EMPNO ASC )  
PARTITIONED ;
```

PARTITIONING

```
CREATE INDEX TRUYK.Z9PARTI2  
ON TRUYK.Z9PARTTB (STATE ASC )  
CLUSTER
```

SECONDARY

```
CREATE INDEX TRUYK.Z9PARTI3  
ON TRUYK.Z9PARTTB  
(EMPNO ASC ,  
 LASTNAME DESC );
```

PARTITIONING

```
CREATE INDEX TRUYK.Z9PARTI4  
ON TRUYK.Z9PARTTB  
(LASTNAME ASC ,  
 CITY ASC )  
PARTITIONED
```

SECONDARY

Partitioned or Non-partitioned

```
CREATE TABLE TRUYK.Z9PARTTB
( EMPNO SMALLINT,
  FNAME CHAR(10)
  LASTNAME CHAR(10)
  ADDRESS CHAR(20)
  CITY CHAR(10)
  STATE CHAR(2))
PARTITION BY (EMPNO ASC)
(PARTITION 1 ENDING AT ( 100),
 PARTITION 2 ENDING AT ( 200),
 PARTITION 3 ENDING AT ( 300))
IN ZTRUYKDB.Z9PARTTS ;
```

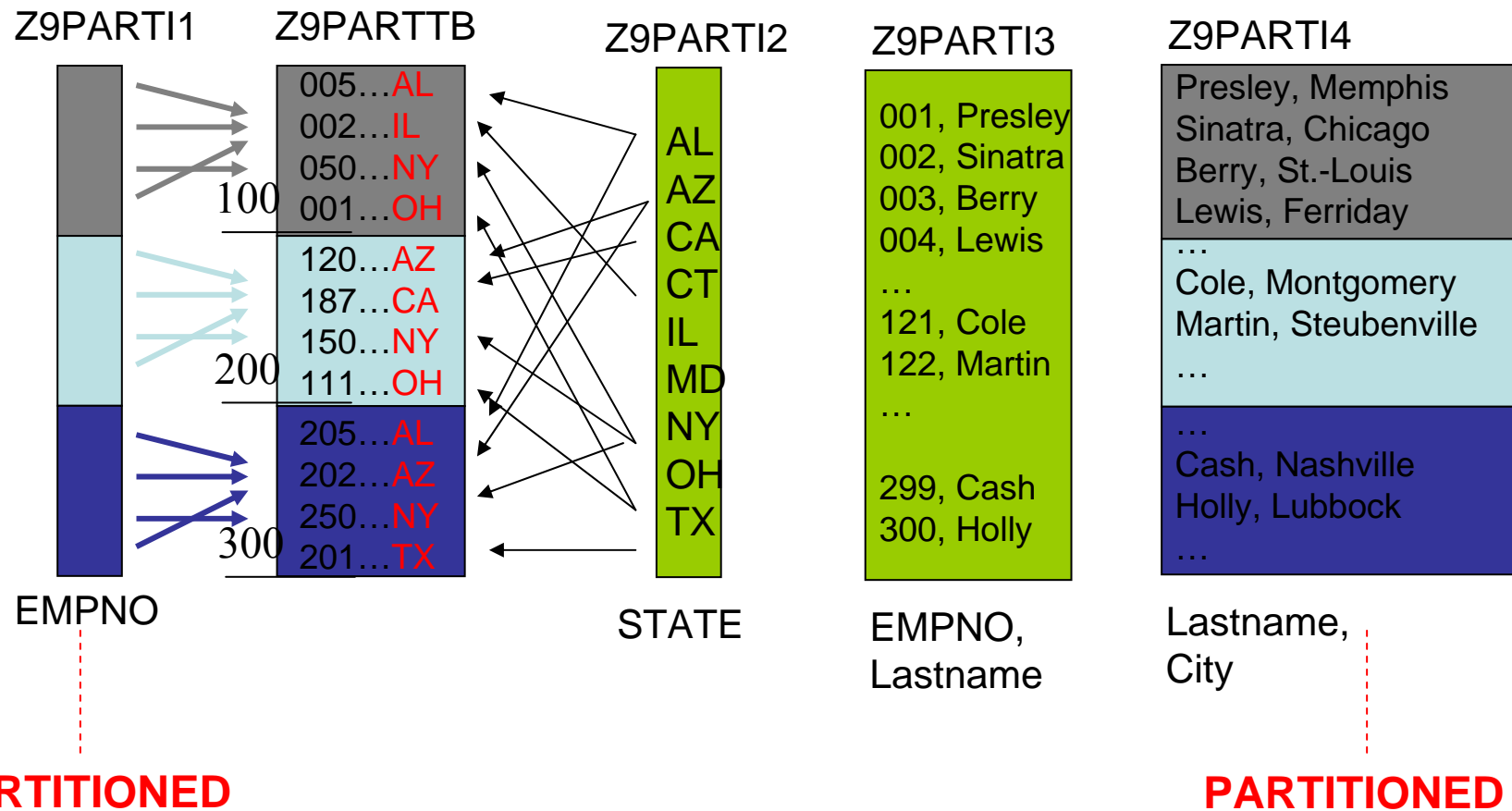
```
CREATE INDEX TRUYK.Z9PARTI1
ON TRUYK.Z9PARTTB (EMPNO ASC )
PARTITIONED ;
```

```
CREATE INDEX TRUYK.Z9PARTI2
ON TRUYK.Z9PARTTB (STATE ASC )
CLUSTER
```

```
CREATE INDEX TRUYK.Z9PARTI3
ON TRUYK.Z9PARTTB
(EMPNO ASC ,
 LASTNAME DESC );
```

```
CREATE INDEX TRUYK.Z9PARTI4
ON TRUYK.Z9PARTTB
(LASTNAME ASC ,
 CITY ASC )
PARTITIONED
```

Partitioned or Non-partitioned



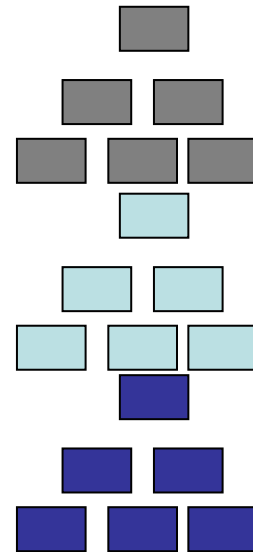


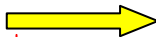
DPSI



Z9PARTI4

Presley, Memphis Sinatra, Chicago Berry, St.-Louis Lewis, Ferriday ...
Cole, Montgomery Martin, Steubenville ...
Cash, Nashville Holly, Lubbock ...



Lastname, 
City

SECONDARY

PARTITIONED

- **Data Partitioned Secondary Index**

- Advantages :

- No Build2 phase
- No contention during LOAD part
- Performance, if partition is known

- Disadvantages :

- Cannot be UNIQUE
- Performance, if partition not known
- Increased DSMAX
- Larger EDMPOOL

Display statement

```
DSNT360I -TD1A *****
DSNT361I -TD1A * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I -TD1A *****
DSNT362I -TD1A DATABASE = ZTRUYKDB STATUS = RW
          DBD LENGTH = 44408
DSNT397I -TD1A
NAME      TYPE PART STATUS PHYERRLO PHYERRHI
-----
Z9PARTTS TS 0001 RW
  -THRU 0003
Z9PARTI1 IX 0001 RW
  -THRU 0003
Z9PARTI2 IX L* RW
Z9PARTI3 IX L* RW
Z9PARTI4 IX D0001 RW
  -THRU 0003
***** DISPLAY OF DATABASE ZTRUYKDB ENDED *****
```



Agenda

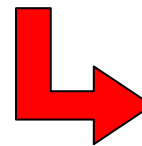


- Partitioning Pre-V8
- Partitioning during V8
- **Consequences of table controlled partitioning**
- Partition Management
 - Add partition
 - Rotate partition
 - Reorg rebalance
- Partitioning during V9

V8 syntax table controlled

```
CREATE TABLE TRUYK.Z9PARTTB
( EMPNO SMALLINT,
  FNAME CHAR(10)
  LASTNAME CHAR(10)
  ADDRESS CHAR(20)
  CITY CHAR(10)
  STATE CHAR(2))
PARTITION BY (EMPNO ASC)
(PARTITION 1 ENDING AT ( 100),
 PARTITION 2 ENDING AT ( 200),
 PARTITION 3 ENDING AT ( 300))
IN ZTRUYKDB.Z9PARTTS ;
```

- **CREATE INDEX PARTITIONED**
- **ALTER INDEX NOT CLUSTER**
on the partitioning index
- **ALTER INDEX CLUSTER**
on the partitioning index
- DROP PARTITIONING INDEX
- ALTER TABLE ADD PARTITION
- ALTER TABLE ROTATE PARTITION
- ALTER TABLE ALTER PARTITION “n”
- CREATE INDEX ENDING AT ...



BE CAREFUL !!

Consequence of alter

PART 1	LIMITKEY	100
PART 2	LIMITKEY	200
PART 3	LIMITKEY	X'FF'

```
INSERT INTO Z9PARTTB VALUES  
(305,'Richie','Vallens', 'some street', 'LosAngeles', 'CA')
```

Index controlled partitioning



Table controlled partitioning



HIGH LIMIT KEY IS ENFORCED



Agenda

- Partitioning Pre-V8
- Partitioning during V8
- Consequences of table controlled partitioning
- **Partition Management**
 - Add partition
 - Rotate partition
 - Reorg rebalance
- Partitioning during V9

Add Partition

- Before V8
 - Unload all partitions
 - Drop tablespace
 - Create tablespace with more parts
 - Load partitions
 - Full Image Copy

- In V8
 - **Alter table**

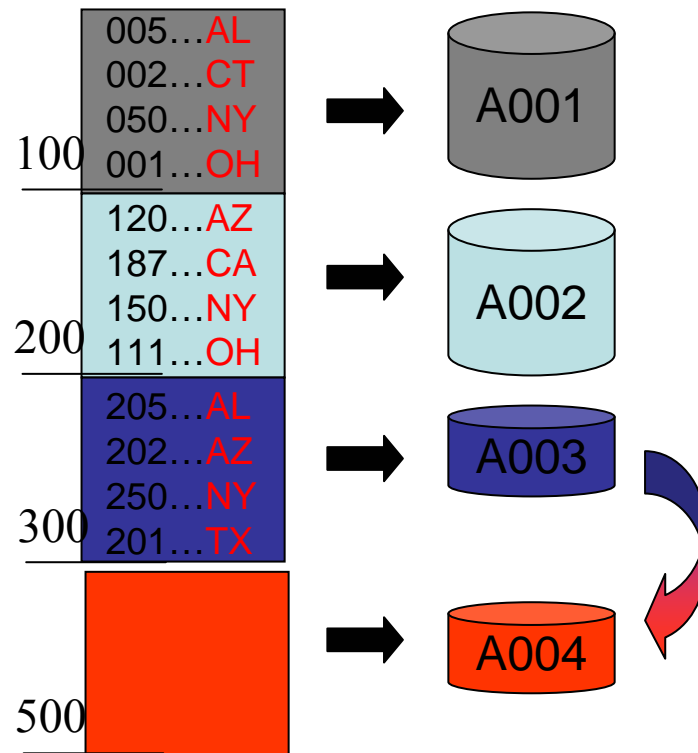
Table Controlled
partitioning needed

Add Partition

```
ALTER TABLE ADDPART_TEST  
ADD PARTITION ENDING AT (500) INCLUSIVE
```

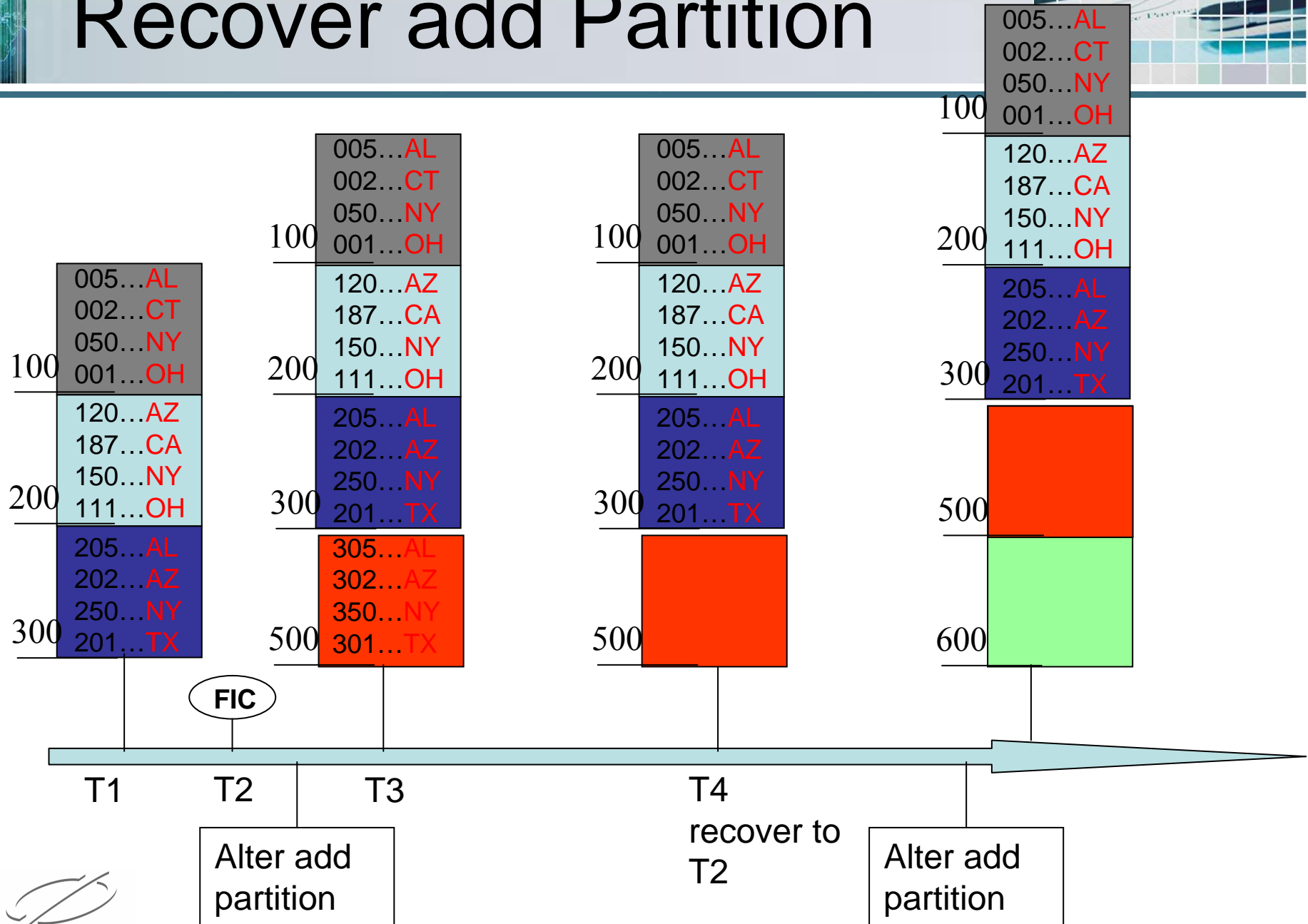
NO partition number

Z9PARTTB

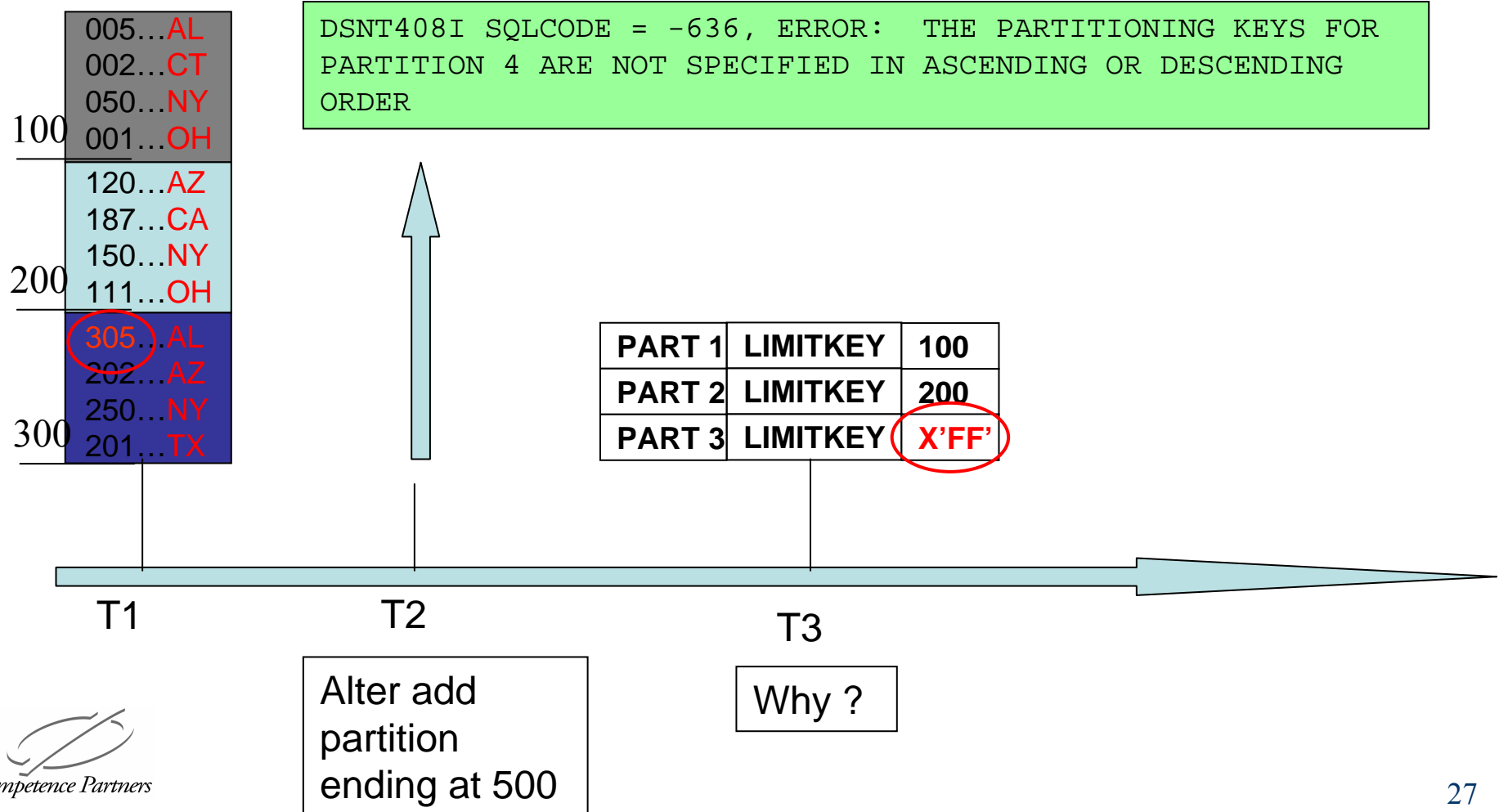


CAN NOT specify size

Recover add Partition



Index controlled partitioning and add partition (1/3)



Index controlled partitioning and add partition (2/3)



```
DSNT404I SQLCODE = 610 , WARNING:  A CREATE/ALTER ON OBJECT  
TRUYK.Z9PARTTB HAS PLACED OBJECT IN REORG PENDING
```

PART 1	LIMITKEY	100
PART 2	LIMITKEY	200
PART 3	LIMITKEY	X'FF'

T3

T4

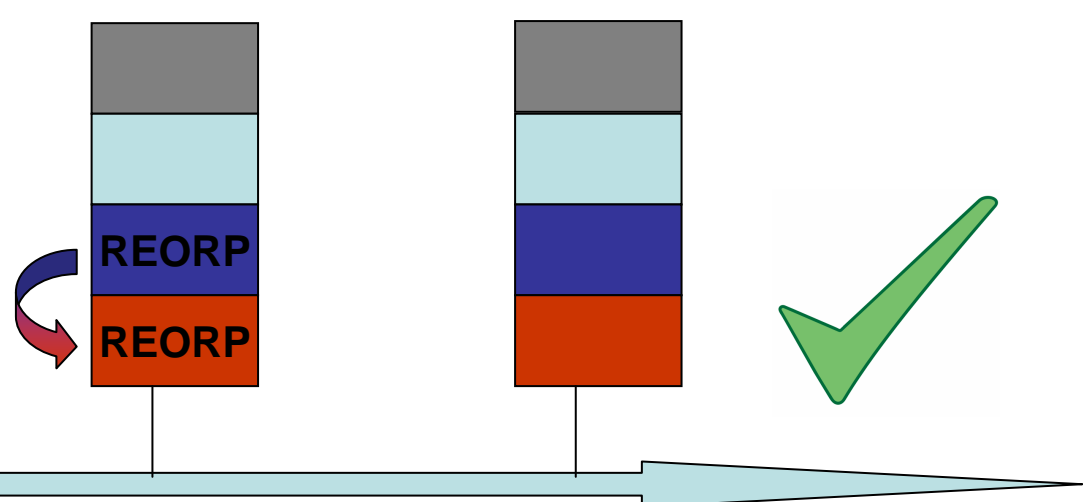
T5

T6

```
ALTER TABLE Z9PARTTB  
ALTER PARTITION 3 ENDING AT ( 300)
```

```
Alter add  
partition  
ending at 500
```

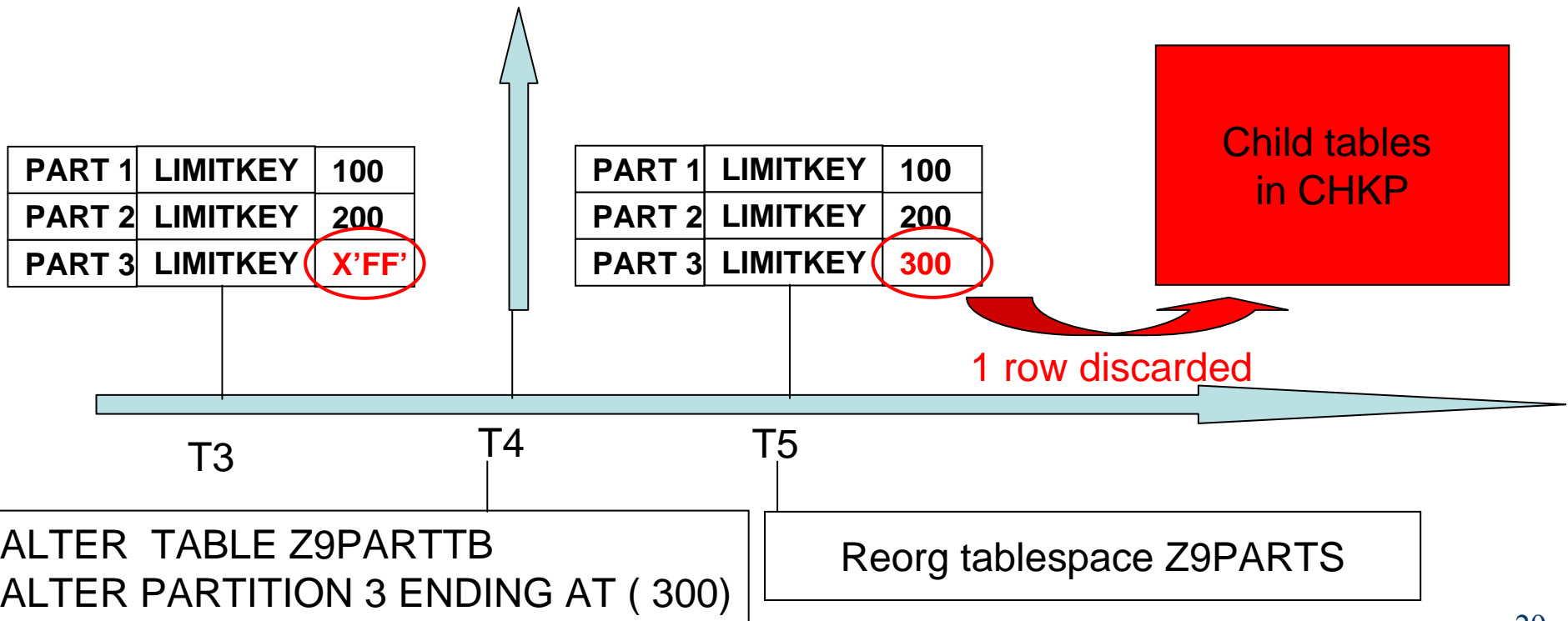
```
Reorg tablespace  
Z9PARTS
```



Index controlled partitioning and add partition (3/3)

BUT !!

```
DSNT404I SQLCODE = 610 , WARNING:  A CREATE/ALTER ON OBJECT  
TRUYK.Z9PARTTB HAS PLACED OBJECT IN REORG PENDING
```



Agenda

- Partitioning Pre-V8
- Partitioning during V8
- Consequences of table controlled partitioning
- **Partition Management**
 - Add partition
 - **Rotate partition**
 - Reorg rebalance
- Partitioning during V9

Rotate Partition

```
CREATE TABLESPACE Z9ROTATS IN ZTRUYKDB  
  NUMPARTS 3 ;
```

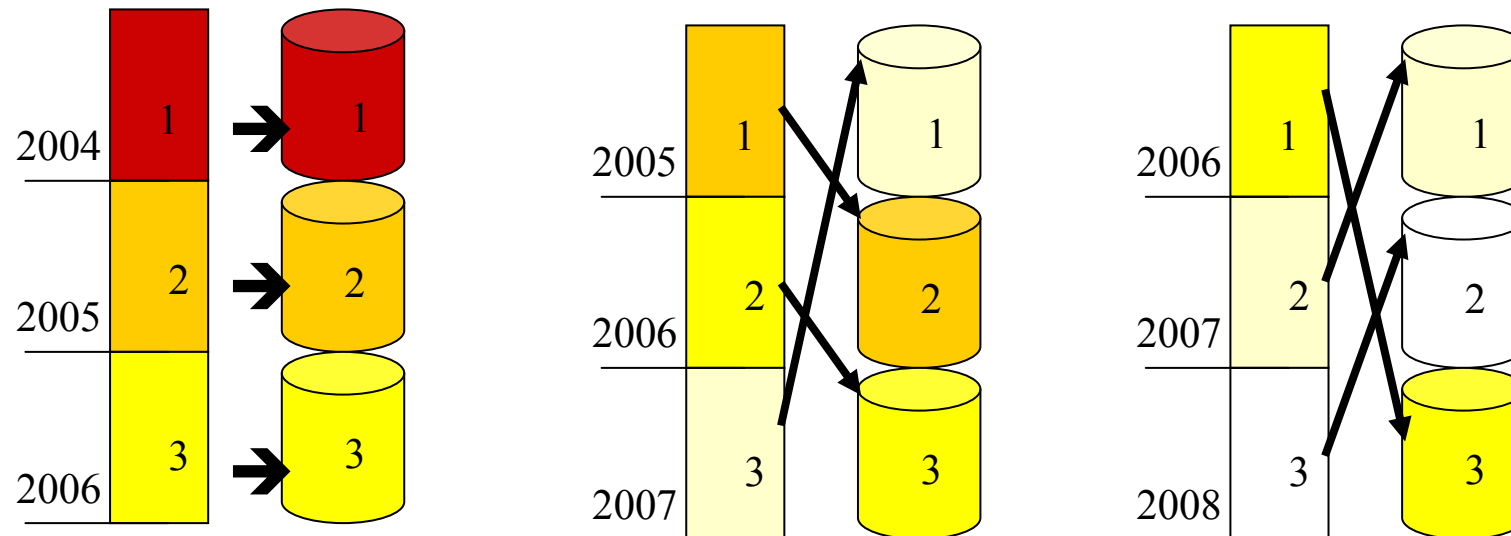
```
CREATE TABLE Z9ROTATB  
  ( YEAR SMALLINT,  
    SALES CHAR(10),  
    REGION CHAR(10))  
  PARTITION BY (YEAR ASC)  
  (PARTITION 1 ENDING AT (2004),  
   PARTITION 2 ENDING AT (2005),  
   PARTITION 3 ENDING AT (2006))  
  IN ZTRUYKDB.Z9ROTATS
```

Rotate Partition

```
DSNT360I  -TD1A *****
DSNT361I  -TD1A *   DISPLAY DATABASE SUMMARY
              *   GLOBAL
DSNT360I  -TD1A *****
DSNT362I  -TD1A   DATABASE = ZTRUYKDB  STATUS = RW
              DBD LENGTH = 48446
DSNT397I  -TD1A
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI
-----
Z9ROTATS TS   0001 RW
          -THRU 0003
***** DISPLAY OF DATABASE ZTRUYKDB ENDED *****
```

Rotate Partition

```
ALTER TABLE ROT_PART_TEST  
ROTATE PARTITION FIRST TO LAST  
ENDING AT (2007) INCLUSIVE RESET
```



```
ALTER TABLE ROT_PART_TEST  
ROTATE PARTITION FIRST TO LAST  
ENDING AT (2008) INCLUSIVE RESET
```

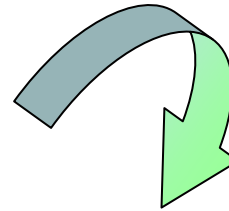
Rotate Partition

```
DSNT360I  -TD1A *****
DSNT361I  -TD1A *   DISPLAY DATABASE SUMMARY
           *   GLOBAL
DSNT360I  -TD1A *****
DSNT362I  -TD1A   DATABASE = ZTRUYKDB  STATUS = RW
           DBD LENGTH = 48446
DSNT397I  -TD1A
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI
-----
Z9ROTATS TS   0003 RW
Z9ROTATS TS   0001 RW
      -THRU 0002
***** DISPLAY OF DATABASE ZTRUYKDB ENDED *****
```

Rotate Partition

```
SELECT DSNUM, LOGICAL_PART, ICTYPE, STYPE
FROM SYSIBM.SYSCOPY
WHERE DBNAME = 'ZTRUYKDB'
AND TSNAME = 'Z9ROTATS' ;
```

DSNUM	LOGICAL_PART	ICTYPE	STYPE
1	1	Q	
2	2	Q	
3	3	Q	
1	1	F	
2	2	F	
3	3	F	



DSNUM	LOGICAL_PART	ICTYPE	STYPE
2		3	A
1		3	A
1		1	Q
2		2	Q
3		3	Q
1		1	F
2		2	F
3		3	F

Rotate Partition

DB2 issues **individual deletes**

- **logging**
- **performance**

DB2 holds **DBD-lock**



DB2 **invalidate plans, packages, statement cache**

GOOD IDEA :
LOAD PART REPLACE

Houston We Have A Problem



**NO Point-in-time RECOVERY
Possible**



Competence Partners



Agenda



- Partitioning Pre-V8
- Partitioning during V8
- Consequences of table controlled partitioning
- **Partition Management**
 - Add partition
 - Rotate partition
 - **Reorg rebalance**
- Partitioning during V9

Reorg Rebalance

```
ALTER TABLE Z9ROTATB  
ADD PARTITION  
ENDING AT (2009) ;
```

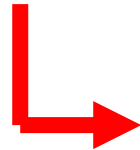
+

```
ALTER TABLE Z9ROTATB  
ROTATE PARTITION FIRST TO LAST  
ENDING AT (2010) INCLUSIVE RESET ;
```

```
DSNT360I  -TD1A *****  
DSNT361I  -TD1A *   DISPLAY DATABASE SUMMARY  
          *   GLOBAL  
DSNT360I  -TD1A *****  
DSNT362I  -TD1A   DATABASE = ZTRUYKDB  STATUS = RW  
          DBD LENGTH = 48446  
DSNT397I  -TD1A  
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI  
-----  
Z9ROTATS TS    0001 RW  
  -THRU    0002  
Z9ROTATS TS    0004 RW  
Z9ROTATS TS    0003 RW  
***** DISPLAY OF DATABASE ZTRUYKDB ENDED *****
```

Reorg Rebalance

```
REORG TABLESPACE ZTRUYKDB.Z9ROTATS PART(2:3)  
SHRLEVEL REFERENCE ;
```



Expected result reorg part 2, 4, 3

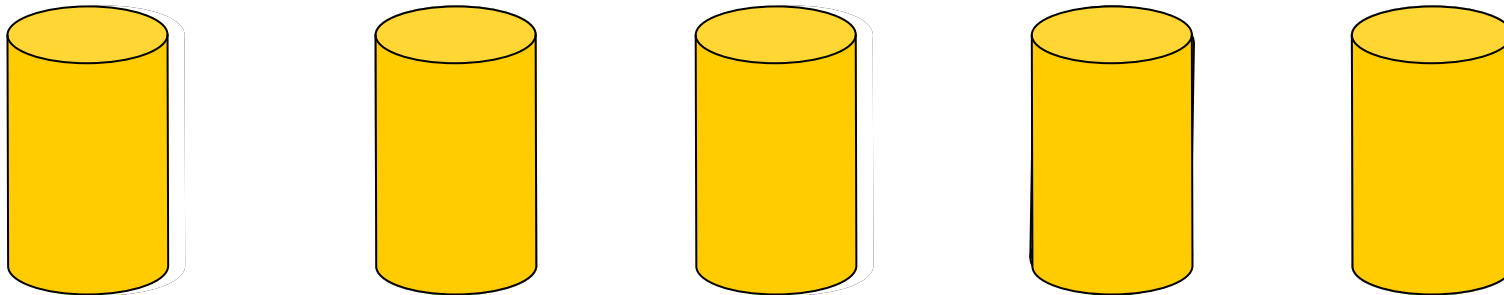
DSNUM	LOGICAL_PART	ICTYPE	SType
3	4	W	
2	2	W	

ACTUAL result : reorg of DSNUM 2 and 3

Add and Rotate Partition Combined



When combined with “add partition” →
keep track of physical dataset



Reorg doesn't solve
logical/physical
problem

ALTER TABLESPACE... PART N
-or-
REORG TABLESPACE... PART N



Reorg Rebalance



- Reorg rebalance allows an automatic “more even” distribution of data over the partitions
- New limit keys are determined at reorg time
- Shrlevel reference is possible

Reorg Rebalance

```
SELECT CARD, DSNUM, LIMITKEY, LOGICAL_PART
FROM SYSIBM.SYSTABLEPART
WHERE DBNAME = 'ZTRUYKDB'
AND TSNAME = 'Z9PARTTS';
```

CARD	DSNUM	LIMITKEY	LOGICAL_PART
25	1	100	1
25	1	200	2
350	1	300	3

```
REORG TABLESPACE ZTRUYKDB.Z9PARTTS REBALANCE
SHRLEVEL REFERENCE
```

Reorg Rebalance

```
SELECT CARD, DSNUM, LIMITKEY, LOGICAL_PART
FROM SYSIBM.SYSTABLEPART
WHERE DBNAME = 'ZTRUYKDB'
AND TSNAME = 'Z9PARTTS';
```

CARD	DSNUM	LIMITKEY	LOGICAL_PART
152	1	210	1
130	1	216	2
118	1	300	3

```
REORG TABLESPACE ZTRUYKDB.Z9PARTTS PART(N:M)
REBALANCE
SHRLEVEL REFERENCE
```

Reorg Rebalance

```
ALTER TABLE Z9ROTATB  
ADD PARTITION  
ENDING AT (2009) ;
```

+

```
ALTER TABLE Z9ROTATB  
ROTATE PARTITION FIRST TO LAST  
ENDING AT (2010) INCLUSIVE RESET ;
```

```
DSNT360I  -TD1A *****  
DSNT361I  -TD1A *   DISPLAY DATABASE SUMMARY  
          *   GLOBAL  
DSNT360I  -TD1A *****  
DSNT362I  -TD1A   DATABASE = ZTRUYKDB  STATUS = RW  
          DBD LENGTH = 48446  
DSNT397I  -TD1A  
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI  
-----  
Z9ROTATS  TS      0001 RW  
  -THRU   0002  
Z9ROTATS  TS      0004 RW  
Z9ROTATS  TS      0003 RW  
***** DISPLAY OF DATABASE ZTRUYKDB ENDED *****
```

Reorg Rebalance

```
REORG TABLESPACE ZTRUYKDB.Z9ROTATS PART(2:3) REBALANCE  
SHRLEVEL REFERENCE ;
```



```
DSNUGUTC - REORG TABLESPACE ZTRUYKDB.Z9ROTATS PART(2:3)  
REBALANCE SHRLEVEL REFERENCE
```

```
DSNURFIT - PARTITION RANGE NOT CONTIGUOUS - REBALANCE IGNORED  
DSNUGBAC - UTILITY EXECUTION TERMINATED, HIGHEST RETURN CODE=8
```



Reorg rebalance



- Be careful : repartition based on amount of keys.
- Varchar rows can still cause uneven VSAM size
- Uneven distribution of keys can still cause uneven VSAM size (e.g. many duplicate keys)



Agenda



- Partitioning Pre-V8
- Partitioning during V8
- Consequences of table controlled partitioning
- Partition Management
 - Add partition
 - Rotate partition
 - Reorg rebalance
- **Partitioning during V9**

Partitioning during V9

Introduction of Universal Table spaces

Advantages :

- partition by growth
- better space management
- improved mass delete performance
- immediate of segments after drop table

Restrictions :

- Cannot be used as workfile
- More space map pages

Partition by growth

```
CREATE TABLESPACE TEST01TS
IN TEST01DB
USING STOGROUP SG1
DSSIZE 2G
MAXPARTITIONS 24
LOCKSIZE ANY
SEGSIZE 4;
COMMIT;
```



Questions ?

Kurt.Struyf@cp.be

www.cp.be