



# Independence Day

## DB2 V8 Partitioning

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**

# Agenda



- › **The Benefits of Partitioning**
- › **How the Terminology Changes in V8**
- › **Table Controlled Partitioning**
- › **Data Partitioned Secondary Indexes**
- › **Online Schema Evolution**

# Why Partition?



- › **Cater for large objects**
  - Maximum of 64GB without partitioning and EA datasets
  - 16TB available with EA datasets and partitioning
    - 128TB in v8 and a PAGESIZE of 32K!
- › **Group Data**
- › **Recoverability**
- › **Encourage Parallelism**
- › **Partition Independence**
  - More on how V8 enhances this later

# What Changes in V8?



› **A shift in thinking....**

› **Partitioned, Partitioning and Clustering are No Longer Intertwined**



# Terminology

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**

# V8 Terminology Changes



## › Partitioning Index

- Left most columns aligned to partitioning sequence
- V7
  - Had to exist, be partitioned, and clustering and only one allowed
- V8
  - Can be non-partitioned
  - Can be non-clustering
  - Not required or can have multiple

## › Partitioned Index

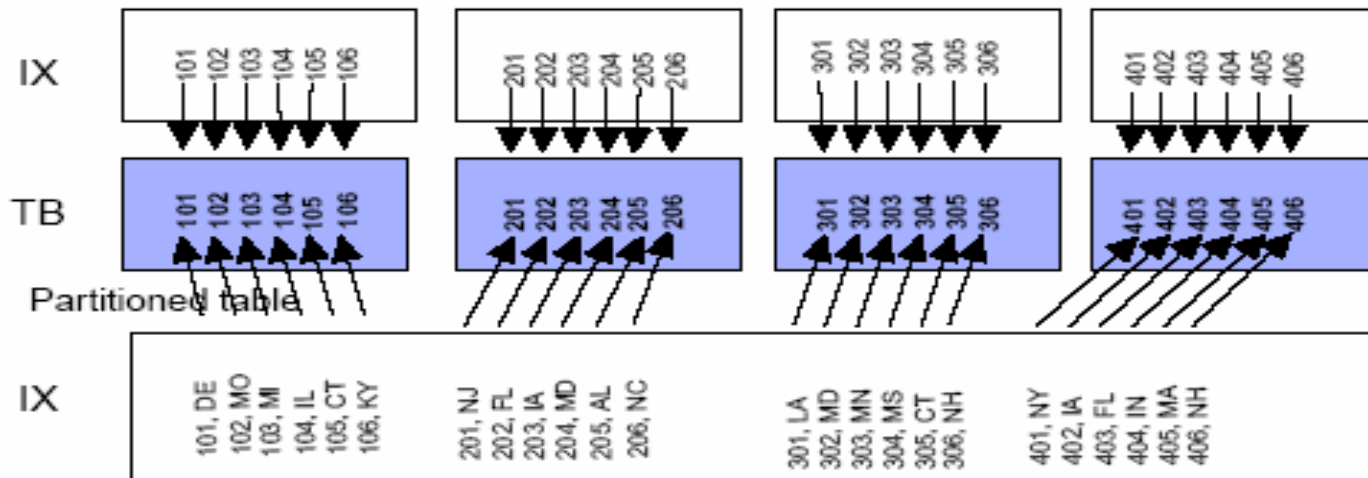
- A physically partitioned index
- V7
  - Only one can be defined
- V8
  - Multiple partitioned indexes can be defined

# A Partitioning Index



A **partitioning** index has the same left-most columns, in the same collating sequence, as the columns which partition the table

Partitioning index `part_IX_1 (ACCOUNT_NUM)`



Partitioning index `part_IX_2 (ACCOUNT_NUM, STATE)`

```
CREATE TABLE CUSTOMER (  
  ACCOUNT_NUM INTEGER,  
  CUST_LAST_NM CHAR(30),  
  ...  
)  
PARTITION BY (ACCOUNT_NUM ASC)  
...
```

# V8 Terminology Changes 2



## › Clustering Index

### – V7

- Was also the partitioned and partitioning index

### – V8

- Can be an index unrelated to the partitioning index
- Can be non-partitioned
- Can be undefined
  - DB2 chooses for you

## › Non-Partitioned Secondary Index (NPSI)

### – V7

- Known as an NPI

### – V8

- Can be clustering



# V8 Terminology Changes 3



## › Data Partitioned Secondary Index (DPSI)

### – V7

- Did not exist

### – V8

- Index keys in partition n of the DPSI only reference data in partition n of the tablespace
- Can be clustering
- Cannot be unique



# Table Controlled Partitioning

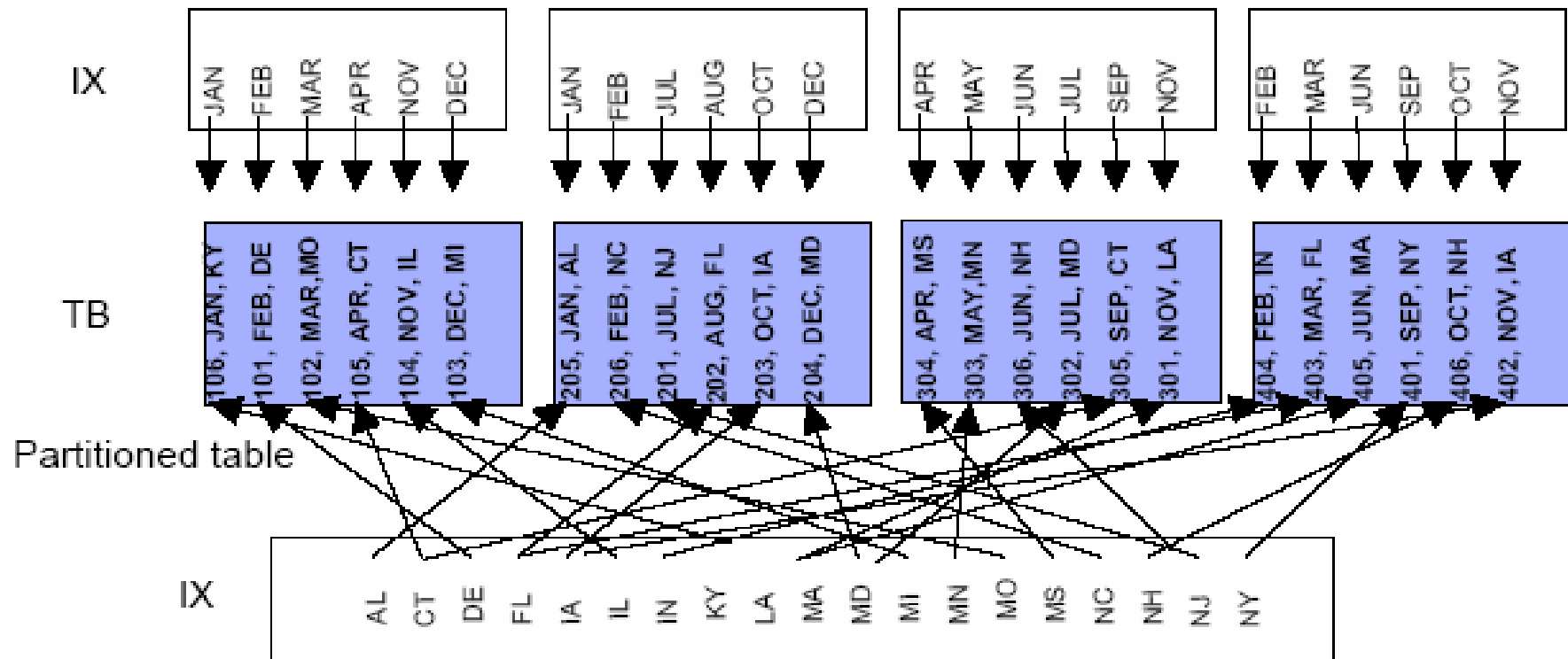
**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**

# Data Partitioned Secondary Indexes



## Data Partitioned Secondary Index -- data\_part\_si\_1



## Non-Partitioned Secondary Index -- non\_part\_si\_2

# The Old and the New



## › Index Controlled Partitioning

- The only method available prior to V8
- Limit Keys specified on the index create statement
- Still available in V8

## › Table Controlled Partitioning

- New in V8
- Likely to become the standard overtime

# Implementing Table Controlled Partitioning



› Executing any of the following DDL statements:

```
DROP partitioning index
```

```
ALTER INDEX NOT CLUSTER (for the partitioning  
index)
```

```
ALTER TABLE ADD PARTITION
```

```
ALTER TABLE ALTER PARTITION ROTATE
```

```
ALTER TABLE ALTER PARTITION part
```

```
CREATE INDEX PARTITIONED
```

```
CREATE INDEX ENDING AT (omitting CLUSTER  
keyword)
```

## But the Easiest Way...



### › Create a DPSI

- DEFER YES
- Automatically converts all the necessary objects
- No outage
- Drop the newly created index

# Table Controlled Partitioning Syntax



- › **The partition boundaries and partitioning columns are now specified in the CREATE TABLE statement**

```
CREATE TABLE tbname (col_1, col_2, col_3...) ...  
PARTITION BY (col_1, col_2) ...  
(PARTITION 1 ENDING AT  
  (col_1_boundary_1, col_2_boundary_1) ,  
PARTITION n ENDING AT  
  (col_1_boundary_n, col_2_boundary_n) ) ;
```

- › **Although unlikely in the real world no index is required to accomplish partitioning**

# Clustering and V8



- › **Any index can now be defined as the clustering index**
  - It is important to define a clustering index under V8
    - If one is not defined
      - The first index in the DBD is used
      - This could change overtime...



# Fixing Problems in the Real World



## › High performance required

- Partitioned object, no free space, ever ascending key
- Very efficient inserts
  - Deferred write
  - Avoids GETPAGES, synchronous I/O, and locks

## › But, is the partitioning key sympathetic to common data access?

- Probably not, therefore NPI's are required
  - Loss of physical partition independence and impact on utility performance
- V8 allows partitioning on one key and clustering on another
  - The clustering index can be a DPSI delivering partition independence



## Data Partitioned Secondary Indexes

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**

# Data Partitioned Secondary Indexes



- › As many physical partitions as the tablespace
- › Keys in partition n only reference data in partition n
- › Cannot be a partitioning index
- › Cannot be **UNIQUE**
  - To enforce uniqueness all of the partitions would need to be scanned
  - A new V8 data object (SEQUENCES) will help here

# Syntax to Create a Clustering DPSI



```
CREATE INDEX TESTIX ON TESTTAB  
(COL_X, COL_Y)  
CLUSTER  
PARTITIONED  
(optional USING block per partition)  
BUFFERPOOL BP3
```

# No NPSI Strategy Benefits



- › **Improved availability**
- › **Utilities can execute with true partition isolation**
  - No NPSI contention
  - No BUILD2 phase
  - Increased parallelism
- › **Data Sharing**
  - Affinity routing
  - No P-Lock contention
- › **Partition Pruning**
  - Strongly consider a Partitioning Index

# A New Way to Enforce Uniqueness



## › Sequences

- Generate unique keys
- Similar to identity columns but,
  - They are stand alone
  - Can be used by multiple objects
  - No retained locks
    - A failing thread will not cause problems later
- Uniqueness can be guaranteed
  - Without the need for an NPSI
- Partition independence can be maintained
- Indexes required solely for uniqueness can be dropped

# An Increase in Partitions



- › **A maximum of 4096 partitions**
  - One partition per day > 11 years worth of data
- › **Beware of current DSSIZE settings**
  - 4K page and 64G DSSIZE only allows 256 partitions to be defined in V8
- › **Consider the number of open datasets**
  - One 4096 partition object with two DPSI's = 12,288 physical datasets
- › **A new dataset naming convention**

# The New Dataset Naming Convention



```
catname.DSNDBx.dbname.psname.p0001.xnnn
```

Where

p is I or J,

xnnn is A001-A999 for partitions 1 through 999,

xnnn is B000-B999 for partitions 1000 through  
1999,

xnnn is C000-C999 for partitions 2000 through  
2999,

xnnn is D000-D999 for partitions 3000 through  
3999, and

xnnn is E000-E096 for partitions 4000 through 4096



# Useful Performance Information

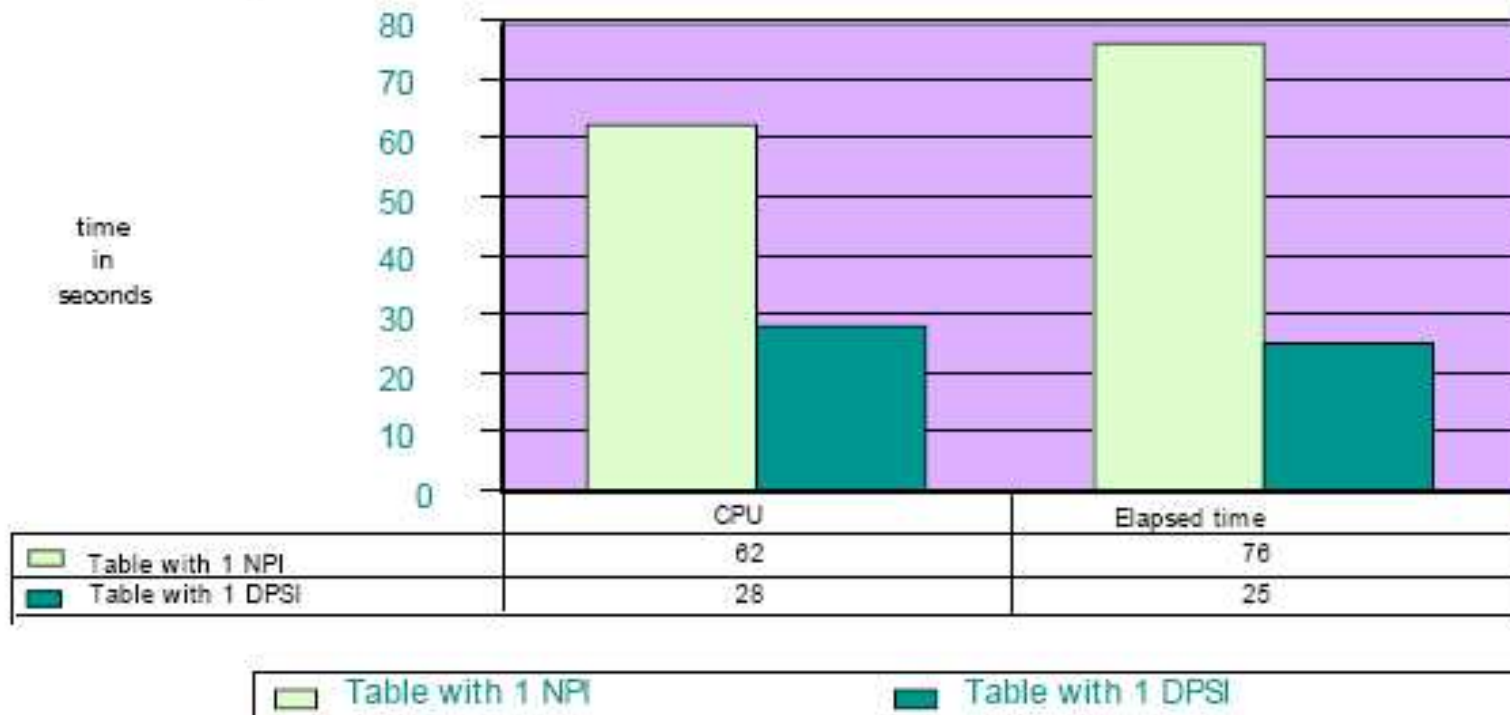


- › **Data for the following slides was taken from IBM Redbook ‘DB2 UDB for z/OS Version 8 Performance Topics’ – SG24-6465-00**
  - This is an excellent source of DPSI performance information
- › **If testing v8 and DPSI’s ensure the following PTF is applied to improve SQL performance when using DPSI’s**
  - UQ93972

# The Good



Load a partition of table of 20M rows 10 partitions with 1 NPI and table with 1 DPSI



# And the bad...



	NPI	DPSI	% difference
Access path	index only	table space scan + sort	
CPU (sec.)	1.21221	8.30175	+ 585
Elapsed (sec.)	1.317186	24.67209	+ 1773
Getpages	2k index getpages	139k data getpages + 29k work file getpages	

- › Query is **SELECTing** a **COUNT** of **DISTINCT** indexed column values
- › Note the differences in the index path
- › This is an extreme example
  - However, having to touch each DPSI partition still adds to overhead even when the index is used



# Online Schema Evolution

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.



# ALTERing Objects



## › Add new columns to an index

- If the UOW also contains the addition of the table column the index is immediately available.
- It is placed into a new V8 state Advisory Reorg Pending (AREO)

## › Partitions

- Manipulated via the ALTER TABLE syntax
- A newly added partition is immediately available
- Rotated
- Rebalanced (not online)

# Partition Rotation



- › The last twelve months of data must always be available
- › Define thirteen partitions
- › Issue the following command:
  - › **ALTER TABLE ALTER PARTITION FIRST TO LAST**
    - If REUSE is specified a logical reset of the partition is executed
    - Bear in mind the rolled partitions data will be deleted
- › Alter the partition boundary keys to receive the new months data
  - Partition enters Reorg Pending state

# Using REORG to Rebalance Partitions



- › **Not an online schema change but pertinent to today's topic**
- › **Automatically evenly distributes data across partitions**
- › **If clustering key does not match partitioning key**
  - Reorg needs to be executed twice
  - Once to move the rows – with REBALANCE
  - Once to order rows – without REBALANCE
- › **Be aware of logical and physical partition numbering**
  - Partition rotation may make a rebalance impossible



And Finally...

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**



# Can You Teach An Old App. New Tricks?



- › **V8 introduces some subtle and radical changes to the partitioning methodology**
  - True physical partition independence via DPSI's
    - Performance benefits
    - Increased parallelism
    - P-lock free affinity routing
  
- › **New applications can easily take advantage of DPSI's**
  
- › **Can the inherent design of old applications make do without NPSI's?**
  - Only time will tell...



Questions?

[Andrew\\_ward@bmc.com](mailto:Andrew_ward@bmc.com)

**ACTIVATE BUSINESS** WITH THE POWER OF I.T.

 **bmcsoftware**