

Different Types of Db2 Tablespace

Frank Rhodes

Solution Architect Administrative Products



Abstract:

This is a beginner level topic to describe the different types of tablespaces, and some of the history of them. We will discuss simple, segmented, and partitioned tablespaces, what differentiates them, and their important characteristics, from the points of view of usage at our customers' sites and their effect on our Db2 products.

This topic assumes knowledge of tablespace types and the DDL syntax to create them.

Agenda

Overview

Index Controlled Partitioning

Table Controlled Partitioning

Partition by Growth

Partition by Range

© Copyright 2019 BMC Software, Inc. - BMC Confidential—Internal Use Only



The information in this presentation can be found in part in the Diagnosis Reference for the appropriate release of DB2. With DB2 12, this information is not in the publicly available Diagnosis Reference, but is in the licensed online documentation at www.ibm.com.

It is worth noting that IBM deprecated support for most all of these Partition types with FL504 March 6, 2019. However, customers will still be able to access and run utilities against these objects, so this presentation will still be of value.

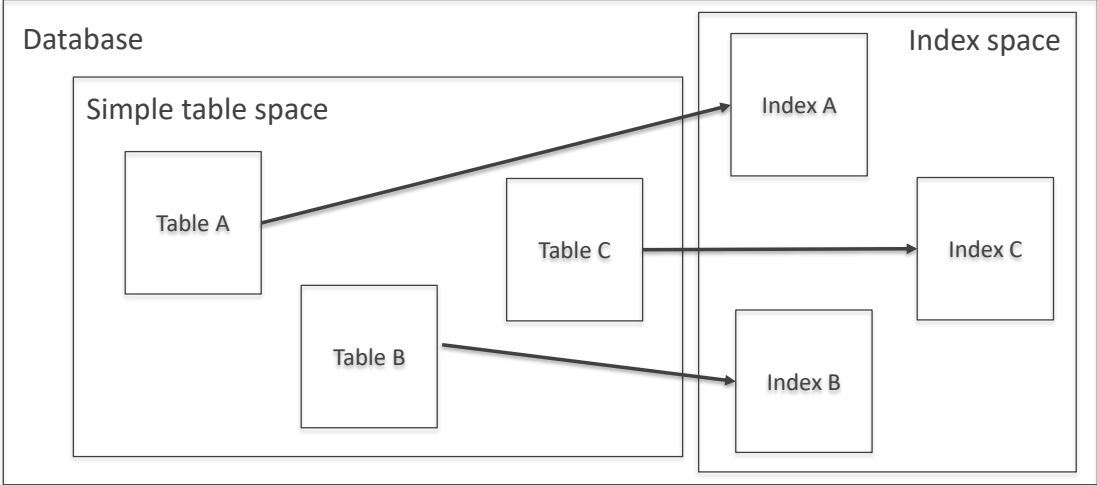
Overview

Types of table spaces

- **Simple**
- **Segmented**
- **Partitioned (non-universal)**
 - Index-controlled partitioned (aka ICP)
 - Table-controlled partitioned (aka TCP)
- **Universal**
 - Partition-by-range (aka range-partitioned, PBR, or PBR2)
 - Partition-by-growth (aka PBG)
- **Large Object (LOB)**
- **XML**

Simple Tablespaces

Simple table spaces



Simple table spaces

- **Characteristics**
 - Space is represented as a page
 - Rows of different tables can be stored in one page
 - Page contains table rows in the order in which they were inserted
- **DB2 catalog table**
 - SYSIBM.SYSTABLESPACE TYPE = blank

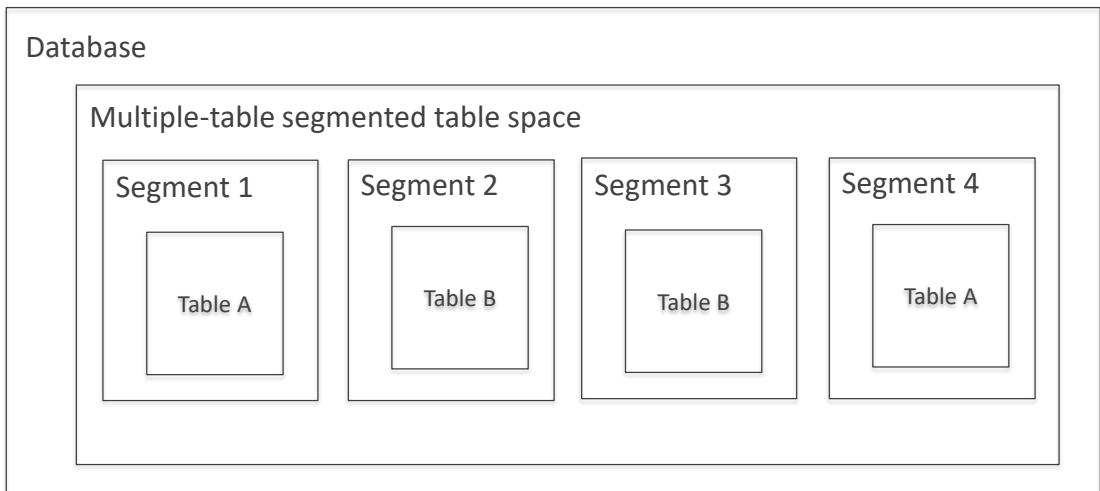
- Page is a unit of storage – it can contain one or more rows of a table

Simple table spaces

- **When to use**
 - Deprecated in DB2 Version 9
 - Cannot create simple table spaces
 - Can alter existing table spaces and update/retrieve data

Segmented Tablespaces

Segmented table spaces



- Diagram is of multi-table table spaces
- Space is represented as segments (equal-sized groups of pages)
- Segments contain rows from only one table
- One table can be stored in more than one segment

Segmented table spaces

- **Characteristics**

- Deprecated, with Db2V12 FL 504, but can still be created
- Space is represented as segments (equal-sized groups of pages)
- Segments contain rows from only one table
- One table can be stored in more than one segment
- Store a maximum of 64 GB of data
- Additional characteristics -
https://www.ibm.com/support/knowledgecenter/en/SSEPEK_12.0.0/intro/src/tpc/db2z_segmentedtablespaces.html

- **DB2 catalog table**

- SYSIBM.SYSTABLESPACE TYPE = blank

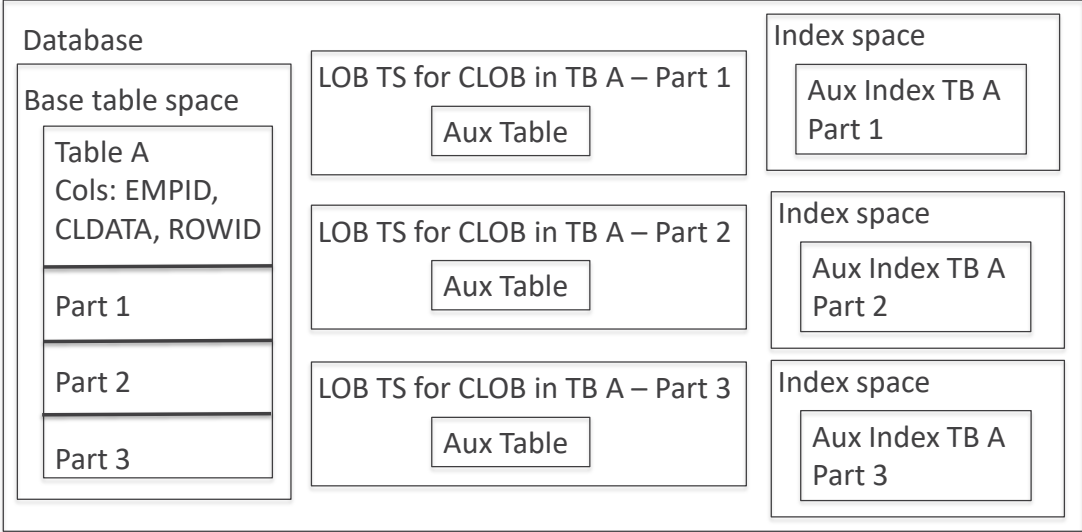
Segmented table spaces

```
CREATE TABLESPACE tableSpaceName IN databaseName
    USING STOGROUP SYSDEFLT
    PRIQTY 96 SECQTY 48
    ERASE YES
    FREEPAGE 0 PCTFREE 15
    BUFFERPOOL BP0
    CLOSE NO
    LOCKSIZE ROW
    SEGSIZE 4 ;
```

- SEGSIZE clause
 - Number of pages in each segment
 - Values range from 4 to 64, in multiples of 4
 - Recommended SEGSIZE depends on number of pages
- LOCKSIZE TABLE clause
 - Specifies the size for locks held on a table space by application processes
- FREEPAGE clause
 - Value should be less than the SEGSIZE value
- If you implicitly create a table space or explicitly create a table space without specifying the SEGSIZE, NUMPARTS, or MAXPARTITIONS options, Db2 creates a **segmented** table space.
- By default, the segmented table space has a SEGSIZE value of 4 and a LOCKSIZE value of ROW.

LOB Tablespaces

Large object (LOB) table spaces



- Also called auxiliary table spaces
- For LOB objects, a *base* table space and one or more *auxiliary* (LOB) table spaces exist
- The *base* table space contains the table with LOB columns
- For each LOB column in the table, you must have an auxiliary table space
- The *auxiliary* table space contains the LOB *data*

LOB table spaces

- **Distinctions**

- For LOB objects, a *base* table space and one or more *auxiliary* (LOB) table spaces exist
- The *base* table space contains the table with LOB columns
- For each LOB column in the table, you must have an auxiliary table space
- The *auxiliary* table space contains the LOB *data*

LOB table spaces

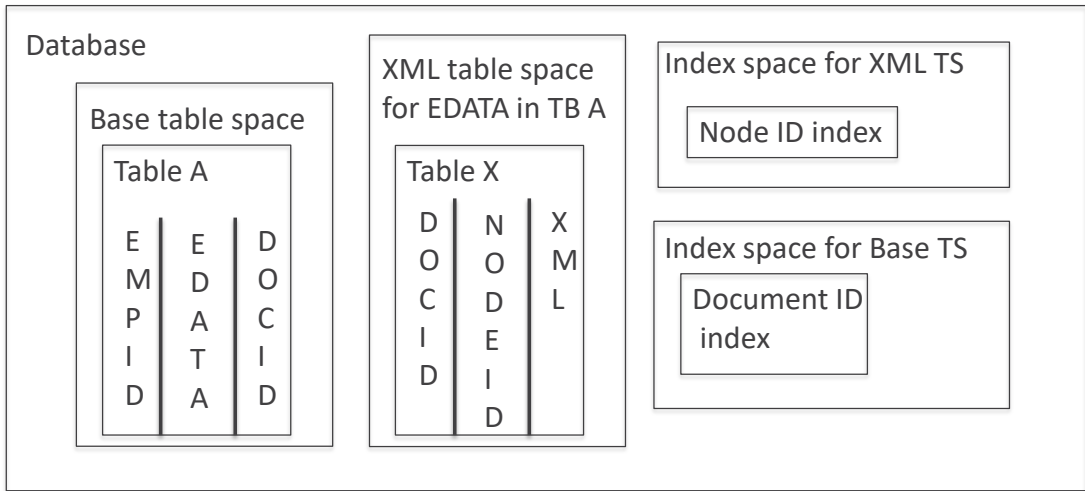
- **Characteristics**
 - Store LOB *data* from a single LOB table – BLOBs, CLOBs, DBCLOBs
 - If the base table space is a partitioned table space, each partition of the base table space must have a LOB table space for each LOB column – thus, you can store more LOB data for each LOB column
- **DB2 catalog table**
 - SYSIBM.SYSTABLESPACE TYPE = O

LOB table spaces

- **When to use**
 - Need to hold large object data, such as graphics, video, or large text strings
- **Notes**
 - SET CURRENT RULES statement can affect creation of auxiliary objects
 - CURRENT RULES DB2 (default) – requires you to create auxiliary objects explicitly
 - CURRENT RULES STD – DB2 creates auxiliary objects implicitly

XML Tablespaces

XML table spaces



XML table spaces

- **Distinctions**

- For XML columns, a *base* table space and one or more *XML* table spaces exist
- The *base* table space contains the table with XML column
- For each XML column in the table, you must have an XML table space
- The *XML* table space contains the XML and a node ID
- *XML table space will be PBG or PBR based upon the TS type of the base table space*

XML table spaces

- **Characteristics**

- Store XML data
- Additional characteristics:
https://www.ibm.com/support/knowledgecenter/en/SSEPEK_12.0.0/intro/src/tpc/db2z_xmlstoragestruct-admin.html

- **DB2 catalog table**

- SYSIBM.SYSTABLESPACE TYPE = P

- **When to use**

- Need to store XML

XML table spaces

- **Creation**

- **Implicitly** created by DB2 when you create an XML column in a base table
- DB2 also **implicitly** creates an XML table to store XML data and a node ID
- Additional objects created – see https://www.ibm.com/support/knowledgecenter/en/SSEPEK_12.0.0/intro/src/tpc/db2z_xmlstoragestruct-admin.html

Auxiliary table spaces

- **Large Object (LOB)**
 - Holds a single LOB column
 - If Base Table Space is Partitioned, holds LOB column data for a single Base TS Partition
- **XML**
 - Holds a single XML column
 - XML Space will inherit from Universal Base TS (PBR or PBG)

Partitioned Tablespaces

© Copyright 2016 BMC Software, Inc.

Internal Use Only

24

Can hold only a single Table

- **Partitioned (non-universal)**
 - Index-controlled partitioned (aka ICP)
 - Table-controlled partitioned (aka TCP)
- **Universal**
 - Partition-by-range (aka range-partitioned, PBR, or PBR2)
 - Partition-by-growth (aka PBG)

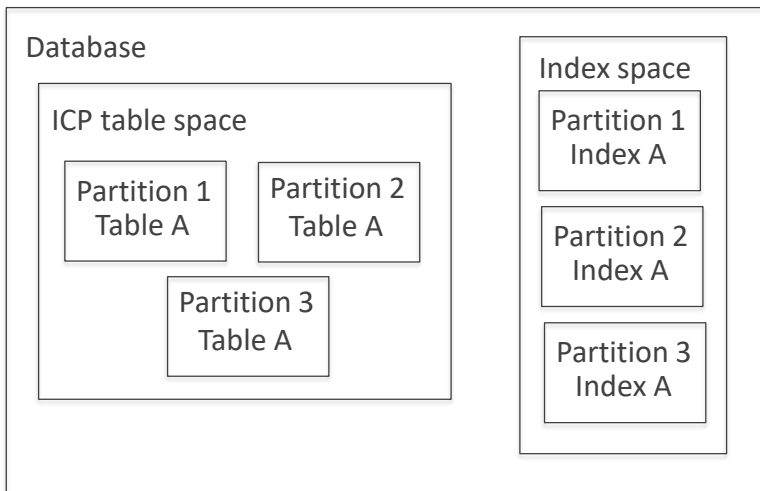
Index Controlled Partitioning

© Copyright 2016 BMC Software, Inc.

Internal Use Only

26

Index-controlled partitioned (ICP) table spaces



- Space is represented as partitions, which are identified by a physical partition number
- Partitioning is controlled by a clustering index (*partitioning index*)
- Can use 1 to 4096 partitions
- Store one table, with up to 128 TB of data

ICP table spaces

- **Characteristics**
 - Deprecated, but can still be created*
 - Space is represented as partitions, which are identified by a physical partition number
 - Partitioning is controlled by a clustering index (*partitioning index*)
 - Can use 1 to 4096 partitions
 - Store one table, with up to 128 TB of data
- **DB2 catalog table**
 - SYSIBM.SYSTABLESPACE TYPE = blank

* If the PREVENT_NEW_IXCTRL_PART subsystem parameter is set to YES, you cannot create tables that use index-controlled partitioning.

Create of the TABLESPACE

```
CREATE TABLESPACE T01ICP                BUFFERPOOL BPO
      IN RDAFJRDB                          CLOSE NO
      NUMPARTS 4                          CCSID EBCDIC
          USING STOGROUP SYSDEFLT          SEGSIZE 0
      PRIQTY 1440
      SECQTY 720
      ERASE NO
      FREEPAGE 0
      PCTFREE 0
      GBPCACHE CHANGED
      TRACKMOD YES
      COMPRESS NO
```

You do not have to specify attributes for each partition. You can specify partition attributes for the TS and these attributes will be the defaults for each partition.

Create of the TABLESPACE explicitly defining each partition

```
CREATE TABLESPACE T01ICP                                ,PART 2
    IN RDAFJRDB                                         USING STOGROUP SYSDEFLT
    NUMPARTS 4                                          PRIQTY      1440
(PART 1                                                 SECQTY      720
    USING STOGROUP SYSDEFLT                             PCTFREE    0   FOR UPDATE 0
    PRIQTY      1440                                    , . . . .
    SECQTY      720                                    )
    PCTFREE    0   FOR UPDATE 0                       BUFFERPOOL BPO
                                                         CLOSE NO
                                                         CCSID EBCDIC
                                                         SEGSIZE 0
```

But if you like you can explicitly define each partition. If you have 4096 part tablespace, you are looking at a very long DDL statement. You can also mix and match. You can define some attributes in the Tablespace section and override the partitions you want with an explicit statement

At this point we do not know if we are dealing with Index Controlled or Table Controlled partitioning. We know we are not dealing with a Universal TS since the SEGSIZE is 0.

Create of the TABLE

```
CREATE TABLE ACML#M.T_T01ICP
  (COLC_1      CHAR(20) NOT NULL WITH DEFAULT
  , COLSI_2    SMALLINT NOT NULL WITH DEFAULT
  , COLIN_3    INTEGER NOT NULL WITH DEFAULT
  , COLSI_4    SMALLINT NOT NULL WITH DEFAULT
  , COLDC_5    DECIMAL(6,2) NOT NULL WITH DEFAULT
  )
CCSID EBCDIC
IN RDAFJRDB.T01ICP
```

Note that the Table Create has no reference to the partitioning keys or limitkeys, these will be defined in the Index definition

Create of the Partitioning Index

```
CREATE UNIQUE INDEX                                (PART 1 VALUES (10)
  RDAFJRDB.I_T01ICP1                               ,PART 2 VALUES (13)
ON RDAFJRDB.T_T01ICP                             ,PART 3 VALUES (16)
  ( COLSI_2 )                                     ,PART 4 VALUES (MAXVALUE)
COMPRESS YES                                     )
CLUSTER                                          BUFFERPOOL BP8K0
  USING STOGROUP SYSDEFLT                        CLOSE        YES
  PRIQTY      1440
  SECQTY      720
  PCTFREE     0
```

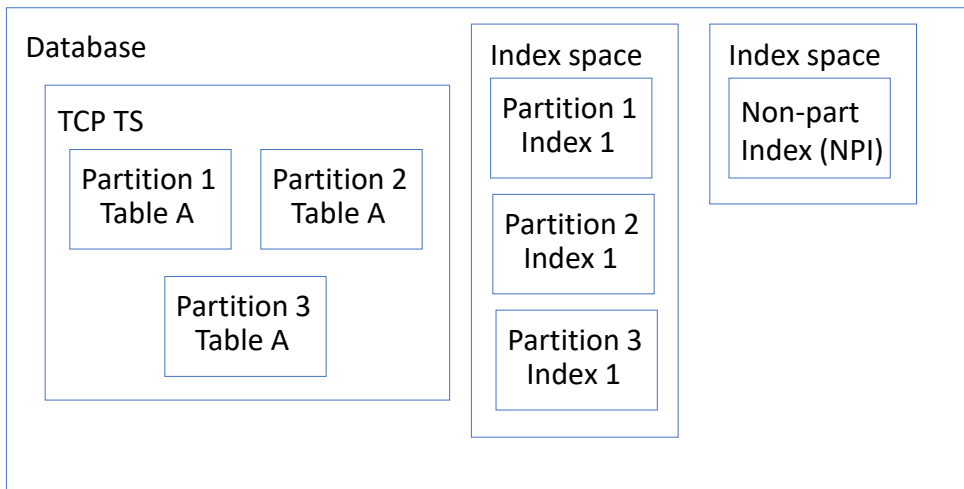
The create of the partitioning index is where the LIMITKEY values are specified controlling how data will be distributed between the 4 partitions.

Note: The Limitkey for the last partition of an ICP is never enforced. We could have specified a value of 30, but rows with a COLCI_2 > 30 would still be stored in the last partition

Table Controlled Partitioning



Table-controlled partitioned (TCP) table spaces



- Looks similar to an ICP table space – biggest difference is how partitions are identified – in TCP, partitions are on the table, in ICP, they're on the index
- Space is represented as partitions
- Partitions are identified by both a physical partition number and a logical partition number
- Store one table
- NPI index has one index space that contains keys for the rows of all partitions of the table space.

TCP table spaces

- Characteristics
 - Space is represented as partitions
 - Partitions are identified by both a physical partition number and a logical partition number
 - Store one table
 - Additional characteristics -
https://www.ibm.com/support/knowledgecenter/en/SSEPEK_12.0.0/intro/src/tpc/db2z_partitionedtablespaces.html
- DB2 catalog table
 - SYSIBM.SYSTABLESPACE TYPE = blank

Create of the TABLESPACE

```
CREATE TABLESPACE T01TCP IN RDAFJRDB
    NUMPARTS 4
    USING STOGROUP SYSDEFLT
    PRIQTY      1440
    SECQTY      736
    PCTFREE     0   FOR UPDATE -1
    BUFFERPOOL BP0
    LOCKSIZE    ANY
    LOCKMAX     10
    CLOSE       NO
    SEGSIZE     0
```

Beside the TS name, there is no difference between the ICP TS and the TCP TS

Create of the TABLE

```
CREATE TABLE RDAFJRDB.T_T04TCP                                PARTITION BY ( COLCH01 )
    (COLCH01 CHAR(50) NOT NULL WITH DEFAULT                    (PARTITION 1 ENDING AT ('EEEEEEE'))
    , COLSI02 SMALLINT NOT NULL WITH DEFAULT                    ,PARTITION 2 ENDING AT ('MMMMMMM')
    , COLIN03 INTEGER NOT NULL WITH DEFAULT                      ,PARTITION 3 ENDING AT ('RRRRRRR')
    , COLVC04 VARCHAR(3900) NOT NULL WITH DEFAULT              ,PARTITION 4 ENDING AT ('ZZZZZZZ')
    , COLDC05 DECIMAL(13,2) NOT NULL WITH DEFAULT              )
)
CCSID EBCDIC
IN RDAFJRDB.T04TCP
```

For Table Controlled partitioning the partitioning key is defined in the TABLE create.
And the limitkey values are also defined in the TABLE definition

Create of the Partitioning Index (NOT REQUIRED)

```
CREATE INDEX RDAFJRDB.I_T04TCP1
  ON RDAFJRDB.T_T04TCP
  ( COLCH01 )
  CLUSTER
  PARTITIONED
  USING STOGROUP SYSDEFLT
  PRIQTY 12
  SECQTY 12
  PCTFREE 10
  BUFFERPOOL BP0
  CLOSE YES
```

The Index Create does not require the number of partitioned to be defined. It will be inherited from the TABLESPACE. We still could have defined the partitions individual if we wanted.

```
CREATE INDEX
  RDAFJRDB.I_T04TCP1
  ON RDAFJRDB.T_T04TCP
  ( COLCH01 )
  PARTITIONED
  CLUSTER
  (PART 1
  USING STOGROUP SYSDEFLT
  PRIQTY 12
  SECQTY 12
  PCTFREE 10
  ,PART 2
  USING STOGROUP SYSDEFLT
  PRIQTY 12
  SECQTY 12
  PCTFREE 10
  ,PART 3
  USING STOGROUP SYSDEFLT
```

```
PRIQTY 12
SECQTY 12
PCTFREE 10
,PART 4
USING STOGROUP SYSDEFLT
PRIQTY 12
SECQTY 12
PCTFREE 10
)
BUFFERPOOL BPO
CLOSE YES
```

Supports Partitioned Indexes as well

For ICP a Partitioning Index is required

- Defines the keys that will be used to control data partitioning
- No other indexes can be partitioned

TCP

- A Partitioning index is allowed but not required
 - The first x number index keys match the partitioning keys defined in the TB create
- Other Partitioned indexes are allowed and do not need to match the partitioning keys

Universal TABLESPACES



The future of partitioned TABLESPACES

Introduced with DB2 V9

Most all Online Schema Changes require a UTS

SIMPLE TABLESPACES have been deprecated since DB2 V9

ZPARM can control if Index Controlled Partitioning objects can be defined

DB2 V12 FL504 deprecated support for SEGMENTED TABLESPACES and ICP

- You can no longer have a single TABLESPACE hold multiple TABLES

Flavors of Universal Tablespaces

Partition by Range

- In simplest terms it is a TCP with a SEGSIZE and a DSSIZE

Partition by Growth

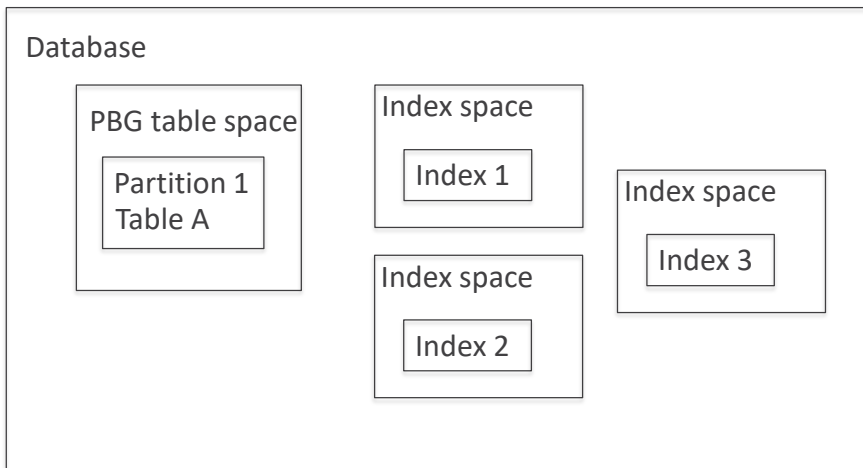
- It is a single table Segmented TS with a DSSIZE
- Instead of growing via Multiple datasets, it grows via new partitions

Partition by Growth

© Copyright 2016 BMC Software, Inc.

43

Partition-by-growth (PBG) table spaces



- DB2 adds new partitions automatically as needed
- Data within each partition is organized according to segmented architecture
- Tables do not include limit keys
- Indexes are nonpartitioned
- Table space stores one table

PBG table spaces

- **Characteristics**
 - Space is represented as a partition, which can grow up to 128 TB
 - DB2 adds new partitions automatically as needed
 - Data within each partition is organized according to segmented architecture
 - Store one table
- **DB2 catalog table**
 - SYSIBM.SYSTABLESPACE TYPE = G

Create of the TABLESPACE

```
CREATE TABLESPACE T06PBG IN RDAFJRDB
MAXPARTITIONS 20
USING STOGROUP SYSDEFLT
PRIQTY 48
SECQTY 32
PCTFREE 0 FOR UPDATE 0
BUFFERPOOL BP16K0
SEGSIZE 32
DSSIZE 1G
```

This statement will initially only create a single partition, as data is added the number of partitions can grow up to MAXPARTITIONS.

You can create the TS with more than 1 part if you also specify Numparts #. This will create the number of parts specified, up to MAX Parts.

Create of the TABLE and INDEXes

TABLE

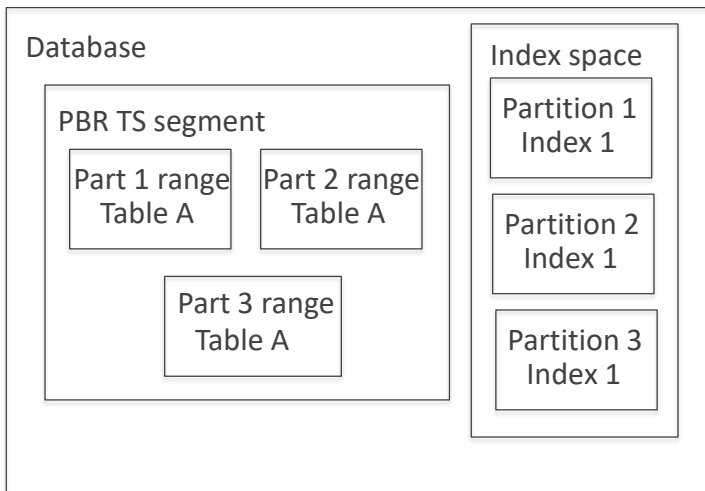
- No impact to the create TABLE DDL
- Only one TABLE is allowed to reside in the PBG

INDEX

- Partitioned Indexes are not allowed

Partition by Range

Partition-by-range (PBR) table spaces



- Also called range-partitioned table spaces
- Tables are partitioned by range, partition definitions include high and low values for limit key ranges
- Indexes are partitioned
- Table space stores one table

PBR table spaces

- **Characteristics**
 - Space is represented as segments (equal-sized groups of pages)
 - Based on partitioning ranges (limit keys) in the table
 - Store one table
- **DB2 catalog table**
 - SYSIBM.SYSTABLESPACE TYPE = R

Create of the TABLESPACE

```
CREATE TABLESPACE T05PBR IN RDAFJRDB
  NUMPARTS 4
  USING STOGROUP SYSDEFLT
  PRIQTY      1440
  SECQTY      736
  PCTFREE     0   FOR UPDATE -1
  BUFFERPOOL BP32K1
  LOCKSIZE    ANY
  LOCKMAX     10
  CLOSE       NO
  SEGSIZE     16
  DSSIZE      4G
```

As mentioned the only real difference between this TABLESPACE and a TCP TS is the SEGSIZE > 0 and the DSSIZE > 0

There are Partition count limits based upon DSSIZE and Bufferpool size

DSSIZE value	4K page size	8K page size	16K page size	32K page size
1G - 4G (1 GB to 4 GB)	4096	4096	4096	4096
8G (8 GB)	2048	4096	4096	4096
16G (16 GB)	1024	2048	4096	4096
32G (32 GB)	512	1024	2048	4096
64G (64 GB)	254	512	1024	2048
128G (128 GB)	128	256	512	1024
256G (256 GB)	64	128	256	512

Creating a PBR using Relative Page numbering

```
CREATE TABLESPACE T05PBR IN RDAFJRDB
  NUMPARTS 4
  USING STOGROUP SYSDEFLT
  PRIQTY      1440
  SECQTY      736
  PCTFREE     0   FOR UPDATE -1
  BUFFERPOOL BP32K1
  LOCKSIZE    ANY
  LOCKMAX     10
  CLOSE       NO
  SEGSIZE     16
  DSSIZE      4G
  PAGENUM     RELATIVE
```

As mentioned the only real difference between this TABLESPACE and a TCP TS is the SEGSIZE > 0 and the DSSIZE > 0

Addressing partition limits

New UTS PBR called UTS PBR Relative Page Number (RPN)

Eliminates relationship between number of parts and DSSIZE

Pages are numbered PER PARTITION

7 byte RID

- Up to 1TB partition size
- Up to 4PB table size (but not finished yet)
- **256** TRILLION rows per table space
- And DSSIZE is now at partition level

Prior to db2 12,,,we had some limitations that were causing availability issues.

DSSIZE is at Table Space Level not Part Level

All Parts inherit the same DSSIZE set at Table Space

No ability to have differing Partition sizes

Altering DSSIZE requires REORG of entire tablespace

Running out of space in a partition is an application outage

- When altering DSSIZE, REORG must run on entire table space

Internally, Db2 always uses the new 7-byte RIDs. As a result, longer log records are created in Db2 12, regardless of whether PBR RPN table spaces are used.

Log

records that are related to table spaces and indexes increase by 20 bytes.

More on PBR RPN

Zparm for system default

PAGENUM ABSOLUTE/RELATIVE for DDL

- Can ALTER – this is a pending change

Increasing DSSIZE is immediate

- No reorg needed

Decreasing DSSIZE is a pending alter

- Still requires entire tablespace reorg

Partitioned IX can use DSSIZE



Thank You

—
Bring IT to Life

