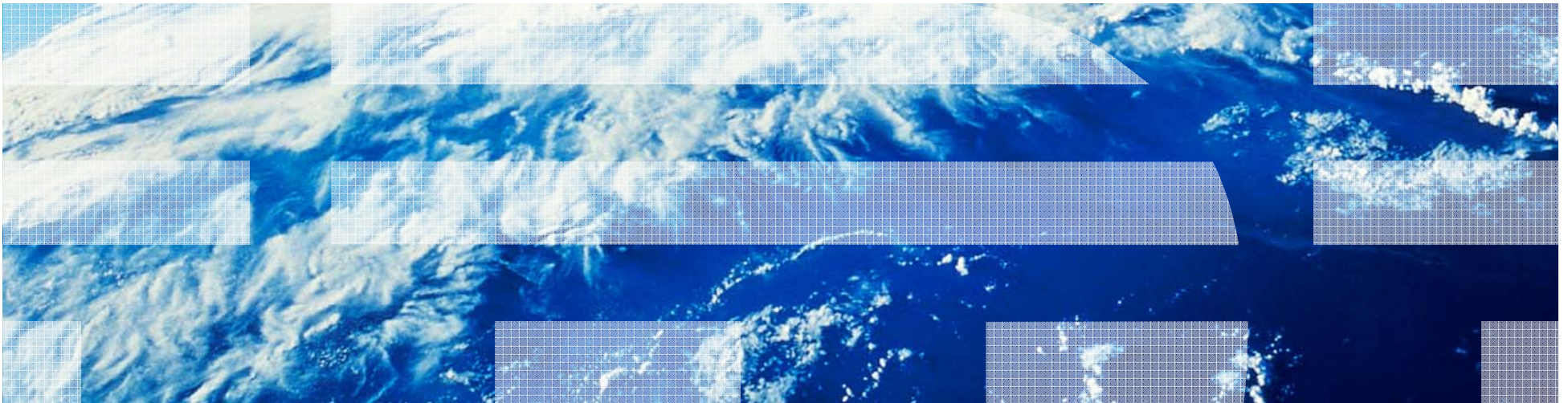


Creating databases and data placement

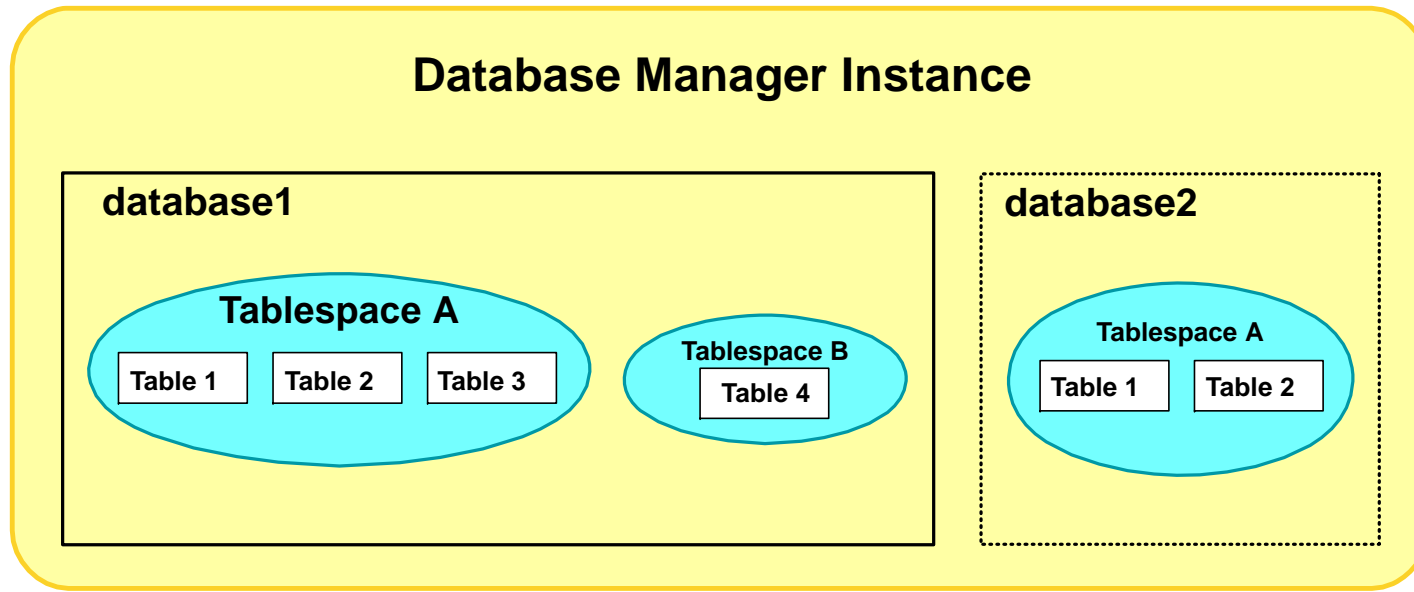


Unit objectives

After completing this unit, you should be able to:

- Review specifics of creating a database
- Explore the System Catalog tables and views
- Check and update Database configuration parameter settings
- Compare DMS, SMS and Automatic Storage managed table spaces
- Describe how to setup and manage a DB2 database with Automatic Storage enabled
- Define Storage Groups to manage databases with different classes of storage available
- Differentiate between table spaces, containers, extents, and pages
- Create and alter table spaces
- Create buffer pools to handle multiple page sizes or improve table access efficiency
- Use DB2 commands and SQL statements to display current table space statistics and status information

Create database overview



- Databases are created within a Database Manager instance
- Table spaces are a logical layer created within a database
- Tables are created within table spaces

Database storage requirements

- Database Path:
 - Database Control Files for each database
 - Includes Database Configuration file, Recovery History, Log Control files, Table space Control file, Bufferpool Control file and others
 - Initial location for database log files
 - Default location is *dftdbpath* in DBM CFG
 - Needs to be a local file system
- Automatic Storage paths:
 - Allows groups of table spaces to be managed rather than managing each table space independently
 - If Automatic storage is enabled there will be at least one path defined
 - Initial Storage Paths can be defined when a database is created
- Default System table spaces:
 - Use Automatic Storage management by default, if enabled, but can be defined to use any supported type of table space
 - SYSCATSPACE: DB2 catalog tables
 - TEMPSPACE1: System Temporary tables, provide work space for sorting and utility processing
 - USERSPACE1: Initial table space for defining user tables and indexes

DB2 Storage Management basics

- DB2 supports three types of storage management for table spaces
- All three types can be used in a single database
- Storage Management type set when a table space is created
- DMS – Database Managed Storage:
 - Table space containers defined using the specified files or raw devices
 - Disk space allocated is reserved for objects in that table space
- SMS – System Managed Storage:
 - Table space containers defined using the specified directories
 - No defined initial size or limit
 - DB2 creates files for each database object
 - Disk space is freed when objects are dropped
 - **Use of SMS management for non-temporary data is deprecated with DB2 10.1**
- Automatic Storage Management:
 - Multiple table spaces share defined disk Storage Paths
 - Multiple Storage groups can be defined (DB2 10.1)
 - Can be enabled when a database is created or added to an existing database
 - DB2 automatically defines the number and names for containers
 - Uses SMS for temporary storage and DMS for other storage types

CREATE DATABASE syntax

```

CREATE DATABASE database-name
    AT DBPARTITIONNUM
    Create Database Options
  
```

Create Database options:

```

    AUTOMATIC STORAGE YES NO ON path drive
    DBPATH ON path drive
  ,
  ALIAS db-alias USING CODESET codeset TERRITORY territory
  COLLATE USING IDENTITY PAGESIZE 4096 n-K DFT_EXTENT_SZ dft-extentsize
  RESTRICTIVE CATALOG TABLESPACE tblspace-defn
  USER TABLESPACE tblspace-defn TEMPORARY TABLESPACE tblspace-defn
  WITH "comment-string" autoconfigure-settings
  
```

tblspace-defn:

```

    MANAGED BY SYSTEM USING ('container-string')
    DATABASE USING (FILE 'container-string' num-pages)
    DEVICE
  EXTENTSIZE num-pages PREFETCHSIZE num-pages
  OVERHEAD number-of-milliseconds TRANSFERRATE number-of-milliseconds
  
```

CREATE DATABASE examples

```
create database sales1 on /dbsales1
```

- Database Path: /dbsales1
- Automatic Storage Path: /dbsales1

```
create database sales2 automatic storage no on /dbsales2
```

- Database Path: /dbsales2
- Automatic Storage not enabled

```
create database sales3 on /dbauto3 dbpath on /dbsales3
```

- Database Path: /dbsales3
- Automatic Storage Path: /dbauto3

```
create database sales4 automatic storage yes  
on /dbauto41,/dbauto42,/dbauto43  
dbpath on /dbsales4
```

- Database Path: /dbsales4
- Automatic Storage Paths: /dbauto41, /dbauto42 and /dbauto43

Completed by DB2 during database creation (1 of 2)

1. Creates database in the specified subdirectory
2. If Automatic storage is enabled a default storage group named IBMSTOGROUP is created
3. Creates SYSCATSPACE, TEMPSPACE1, and USERSPACE1 table spaces
4. Creates the system catalog tables and recovery logs
5. Catalogs database in local database directory and system database directory
6. Stores the specified code set, territory, and collating sequence
7. Creates the schemas SYSCAT, SYSFUN, SYSIBM, and SYSSTAT
8. Binds database manager bind files to the database (db2ubind.lst)
 - DB2 CLI packages are automatically bound to databases when the databases are created or migrated. If a user has intentionally dropped a package, then you must rebind db2cli.lst.

Completed by

DB2 during database creation (2 of 2)

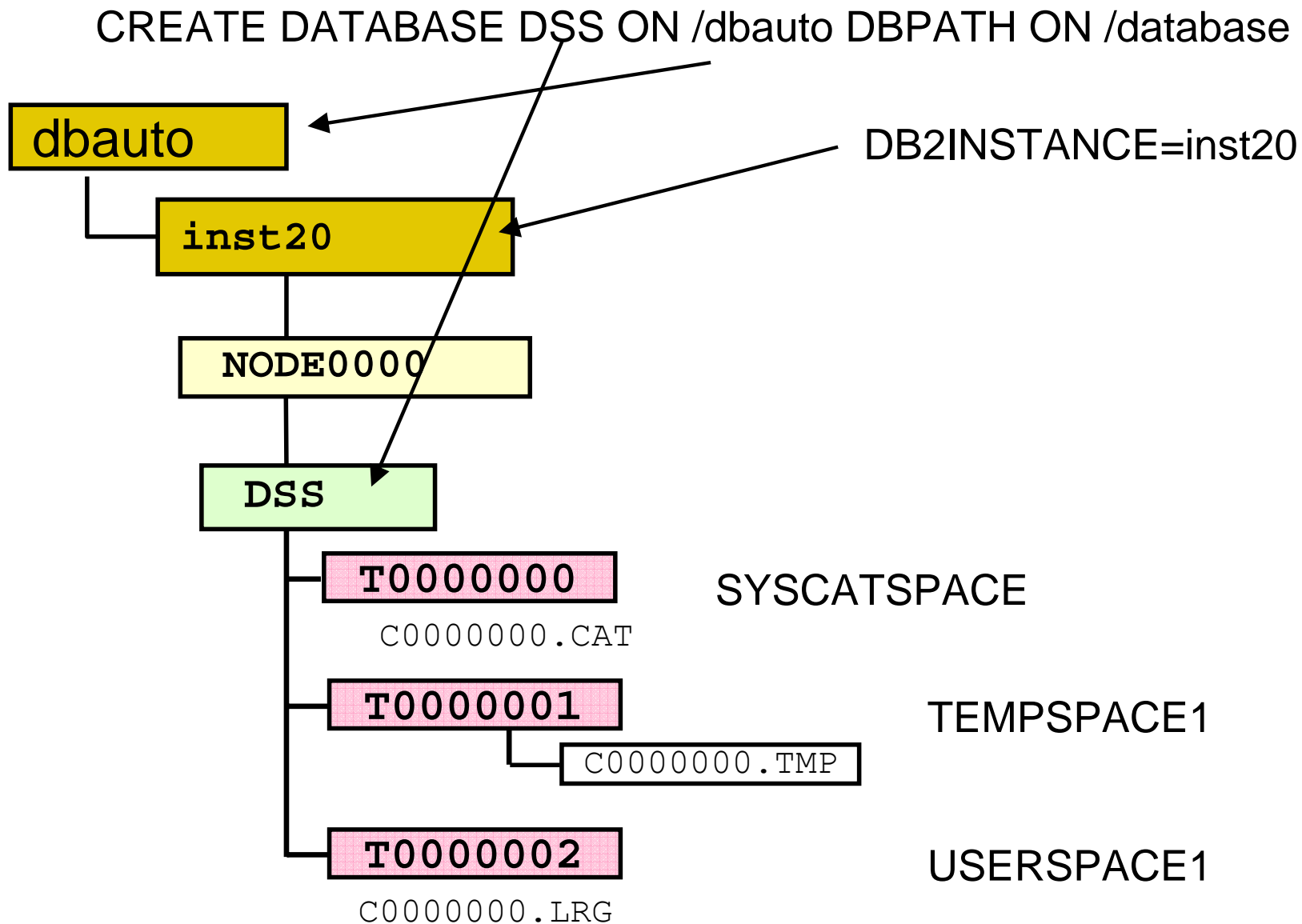
9. Grants the following privileges:

- ACCESSCTRL , DATAACCESS , DBADM and SECADM privileges to database creator
 - SELECT privilege on system catalog tables and views to PUBLIC
 - UPDATE access to the SYSSTAT catalog views
 - BIND and EXECUTE privilege to PUBLIC for each successfully bound utility
 - CREATETAB, BINDADD, IMPLICIT_SCHEMA, and CONNECT authorities to PUBLIC
 - USE privilege on USERSPACE1 table space to PUBLIC
 - Usage of the WLM workload for user class SYSDEFAULTUSERCLASS to PUBLIC
- When the RESTRICTIVE option is used, no privileges are automatically granted to PUBLIC

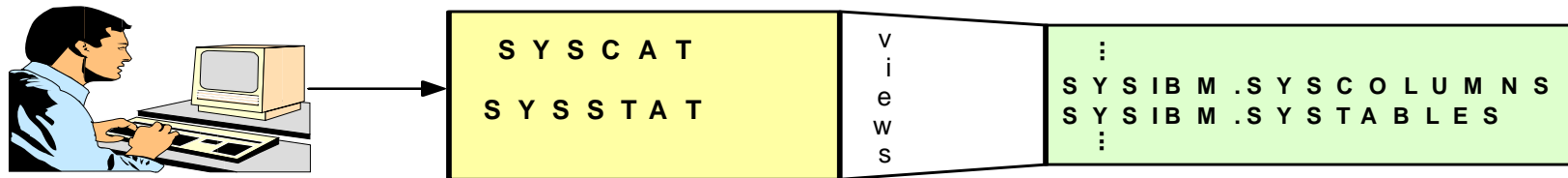
Database Path storage

- **Partition-global directory**
- The partition-global directory has the path: *your_instance/NODExxxx/SQLxxxxx*
- */database/inst481/NODE0000/SQL00001:*
 - db2rhist.asc
 - db2rhist.bak
 - db2rhist.lock
 - HADR
 - load
 - LOGSTREAM0000 (Default database logs)
 - MEMBER0000
 - SQLDBCONF
 - SQLOGAB
 - SQLOGCTL.GLFH.1
 - SQLOGCTL.GLFH.2
 - SQLOGCTL.GLFH.LCK
 - SQLSGF.1
 - SQLSGF.2
 - SQLSPCS.1
 - SQLSPCS.2
- */database/inst481/NODE0000/SQL00001/LOGSTREAM0000:*
 - S0000000.LOG
 - S0000001.LOG
 - S0000002.LOG
- **Member-specific directory**
- The member-specific directory has the path: *your_instance/NODExxxx/SQLxxxx/MEMBERxxxx*
- */database/inst481/NODE0000/SQL00001/MEMBER0000:*
 - db2event
 - DB2TSCHG.HIS
 - HADR
 - SQLBP.1
 - SQLBP.2
 - SQLDBCONF
 - SQLINSLK
 - SQLOGCTL.LFH.1
 - SQLOGCTL.LFH.2
 - SQLOGMIR.LFH
 - SQLTMPLK
- *Local database Directory*
- */database/inst481/NODE0000/sqlldbdir:*
 - sqlddbak
 - sqldbdir
 - sqldbins

Default table space containers with Automatic Storage

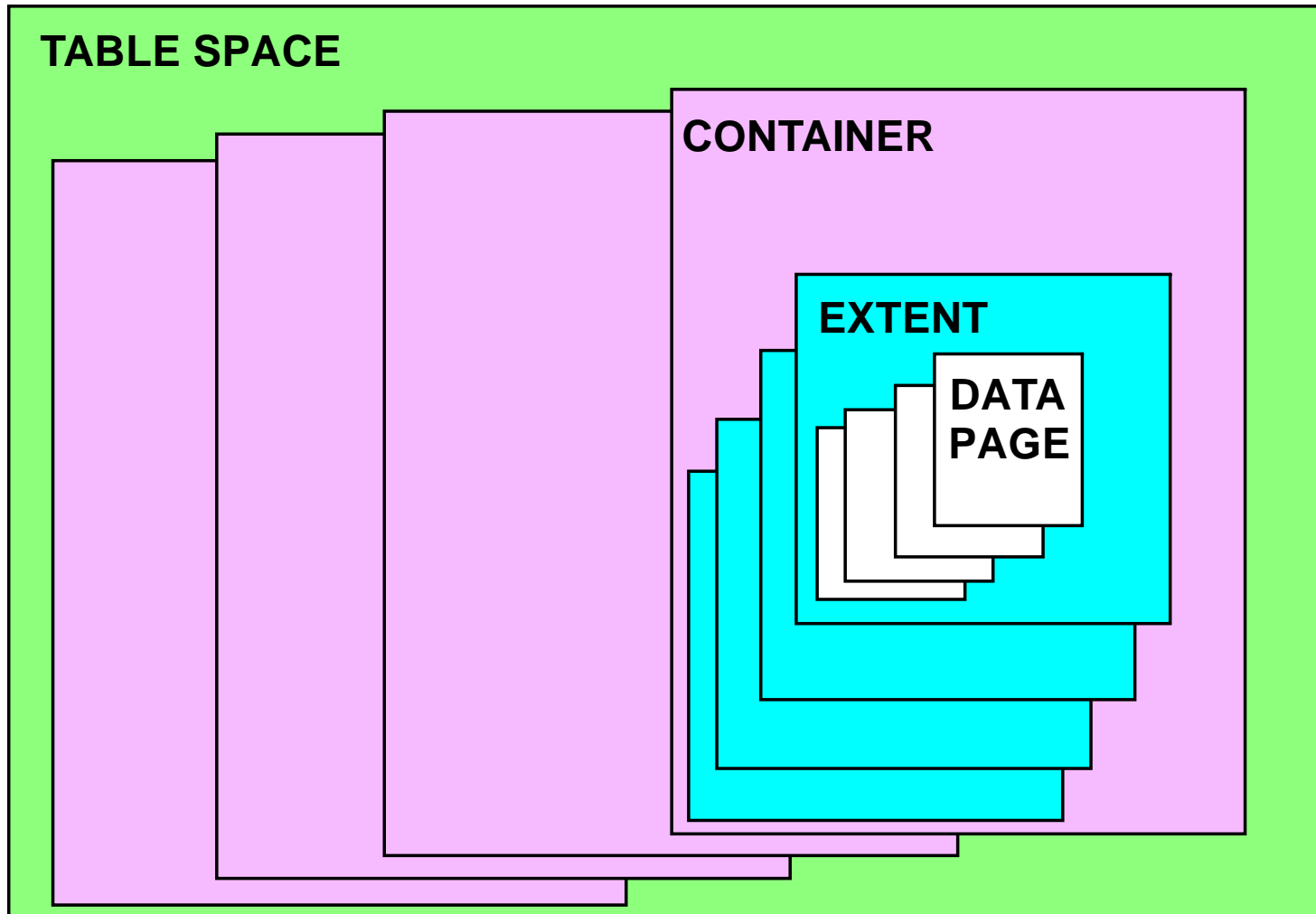


System Catalog tables and views



See notes for actual table and view names.

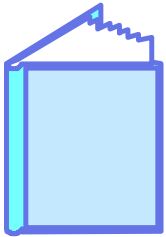
Table space, container, extent, page



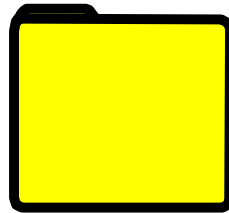
Containers and table spaces

- Container is an Allocation of Physical Space

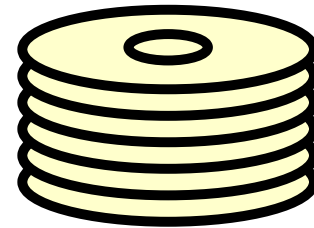
What Does a Container Look Like?



Directory



File



Device

<i>Directory</i>	<i>File</i>	<i>Device</i>
SMS	DMS	DMS

Writing to containers

- DFT_EXTENT_SZ defined at database level
- EXTENTSIZE defined at table space level
- Data written in round-robin manner

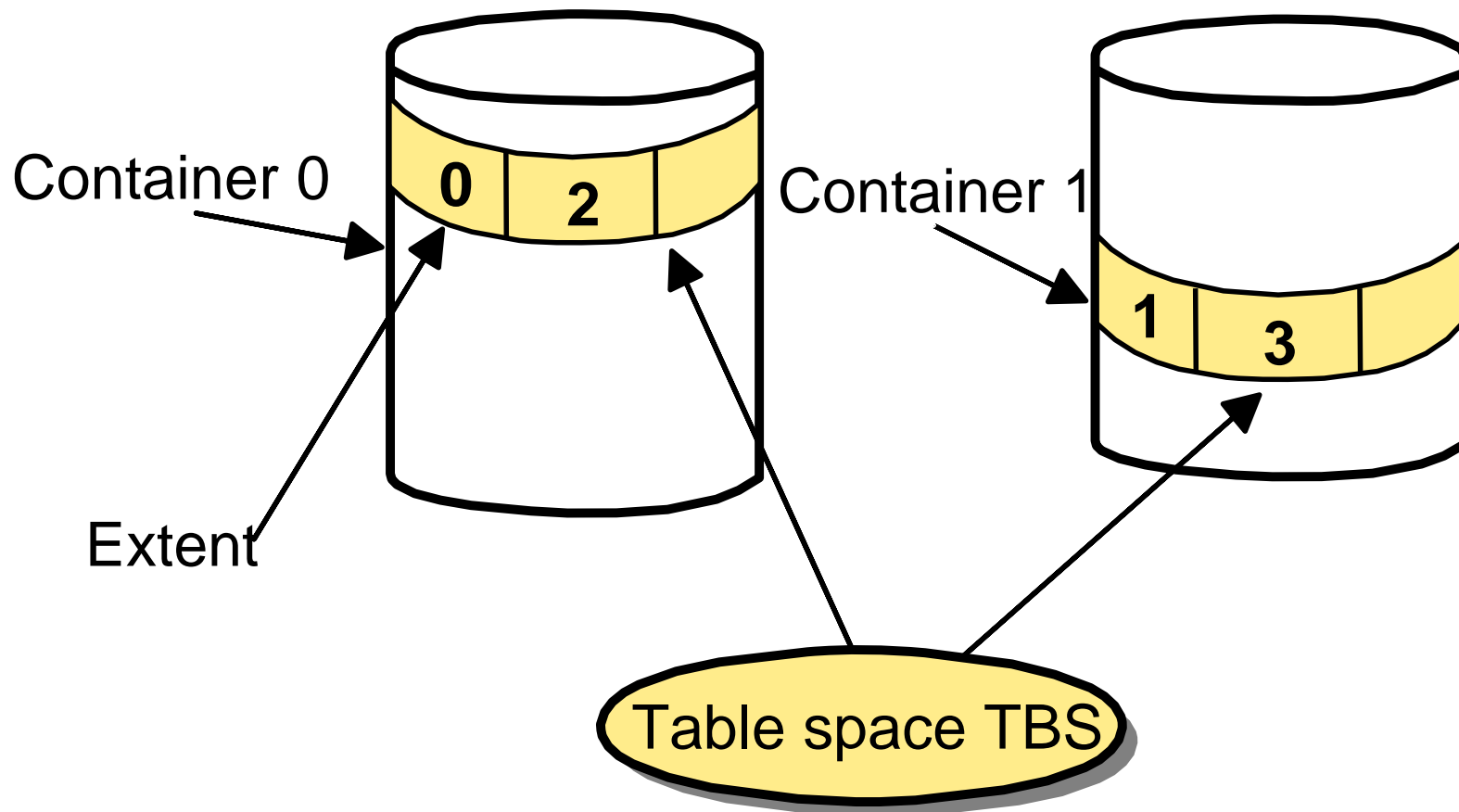
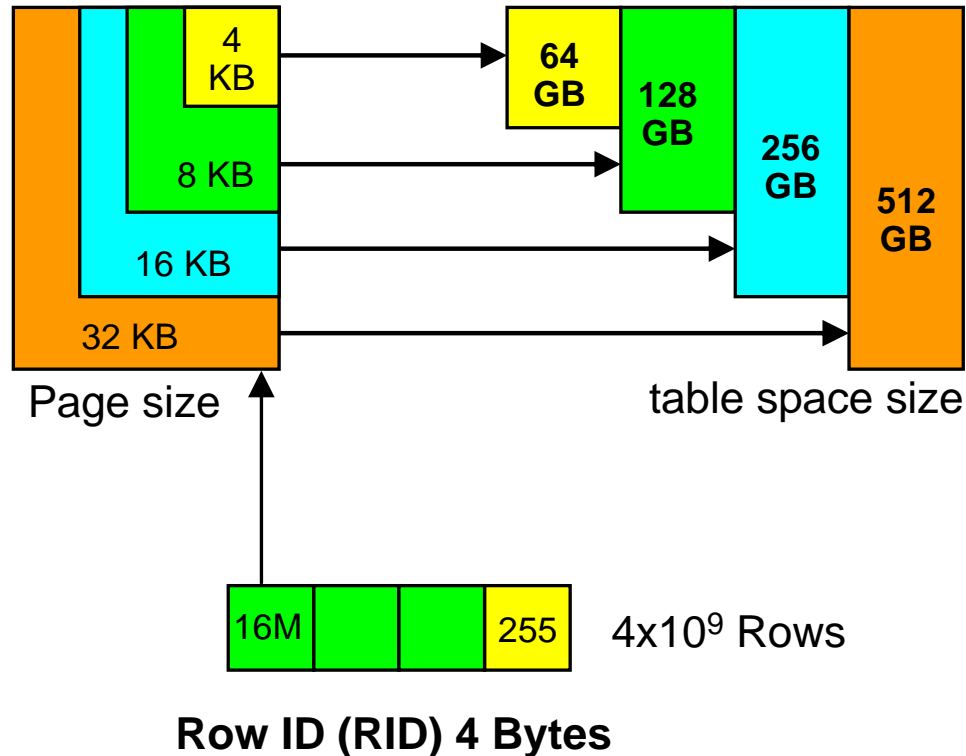


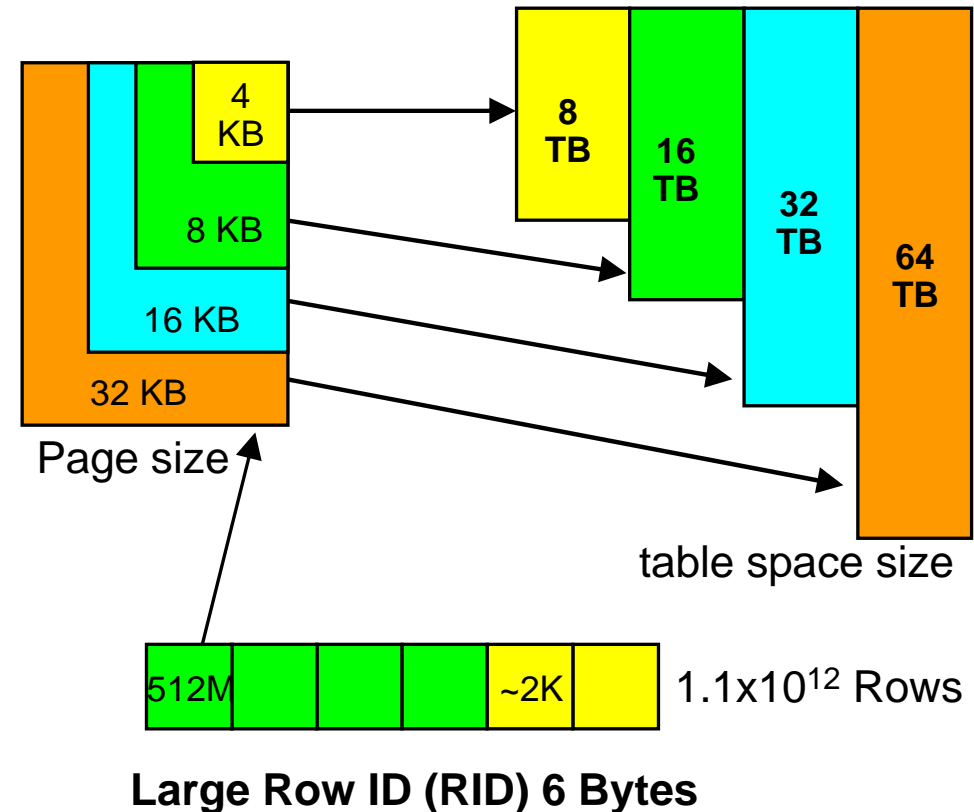
Table space design limits: Row Identifiers

- Standard RIDs



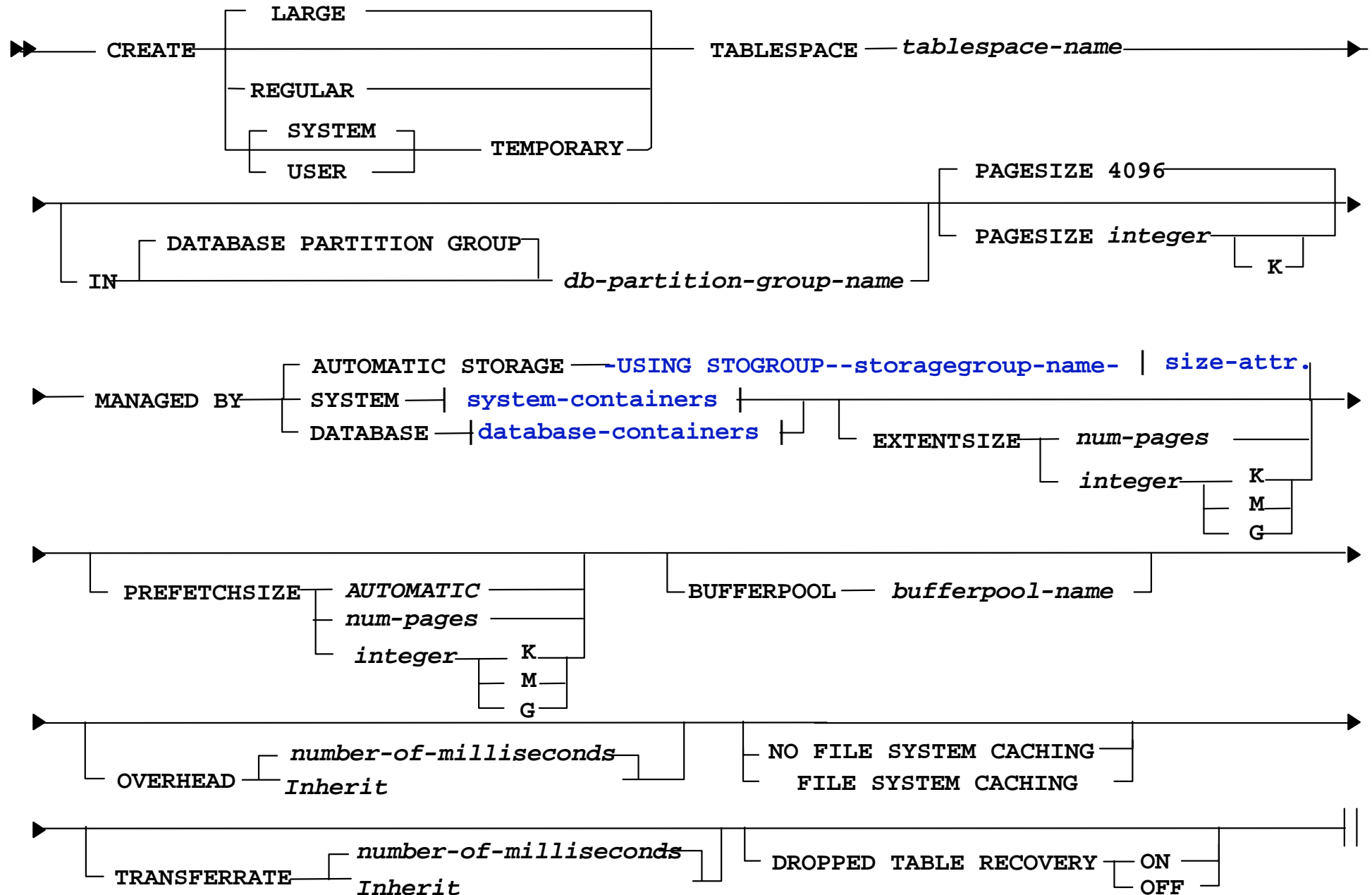
- For tables in Regular table spaces

- Large RIDs



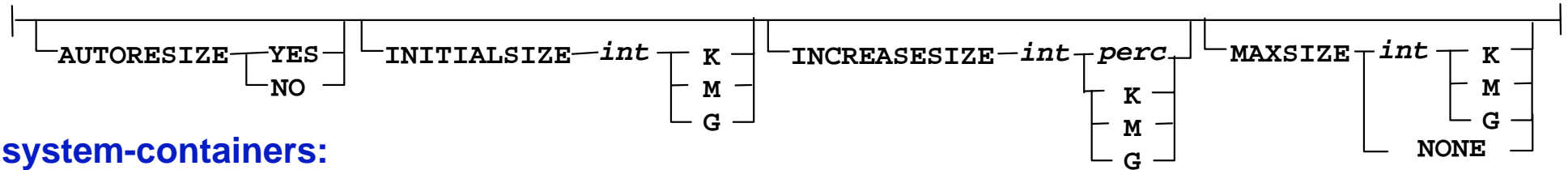
- For tables in LARGE table spaces (DMS or Automatic Storage)
- Also all SYSTEM and USER temporary table spaces

CREATE TABLESPACE syntax (1 of 2)

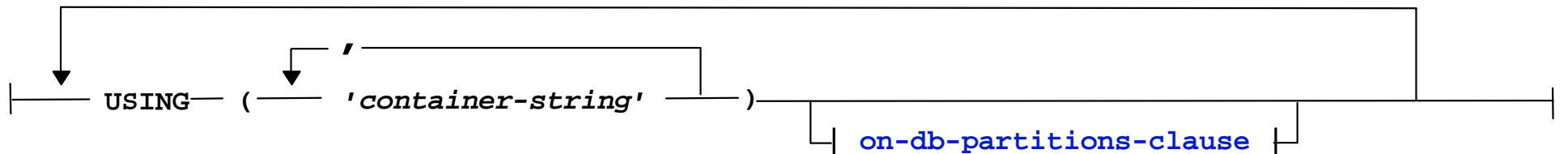


CREATE TABLESPACE syntax (2 of 2)

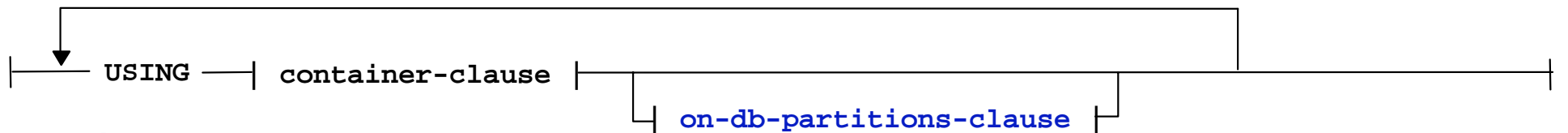
size-attributes:



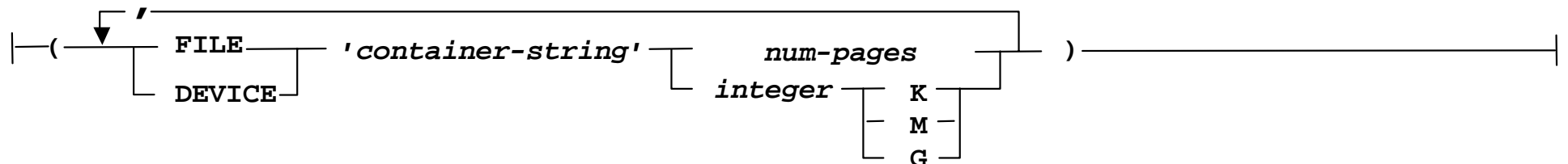
system-containers:



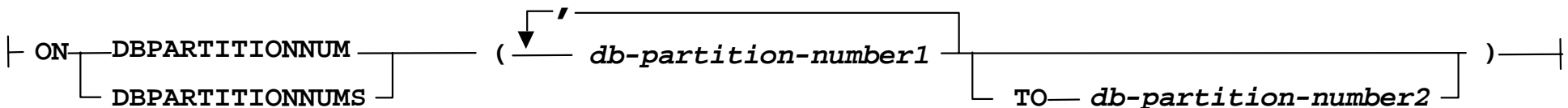
database-containers:



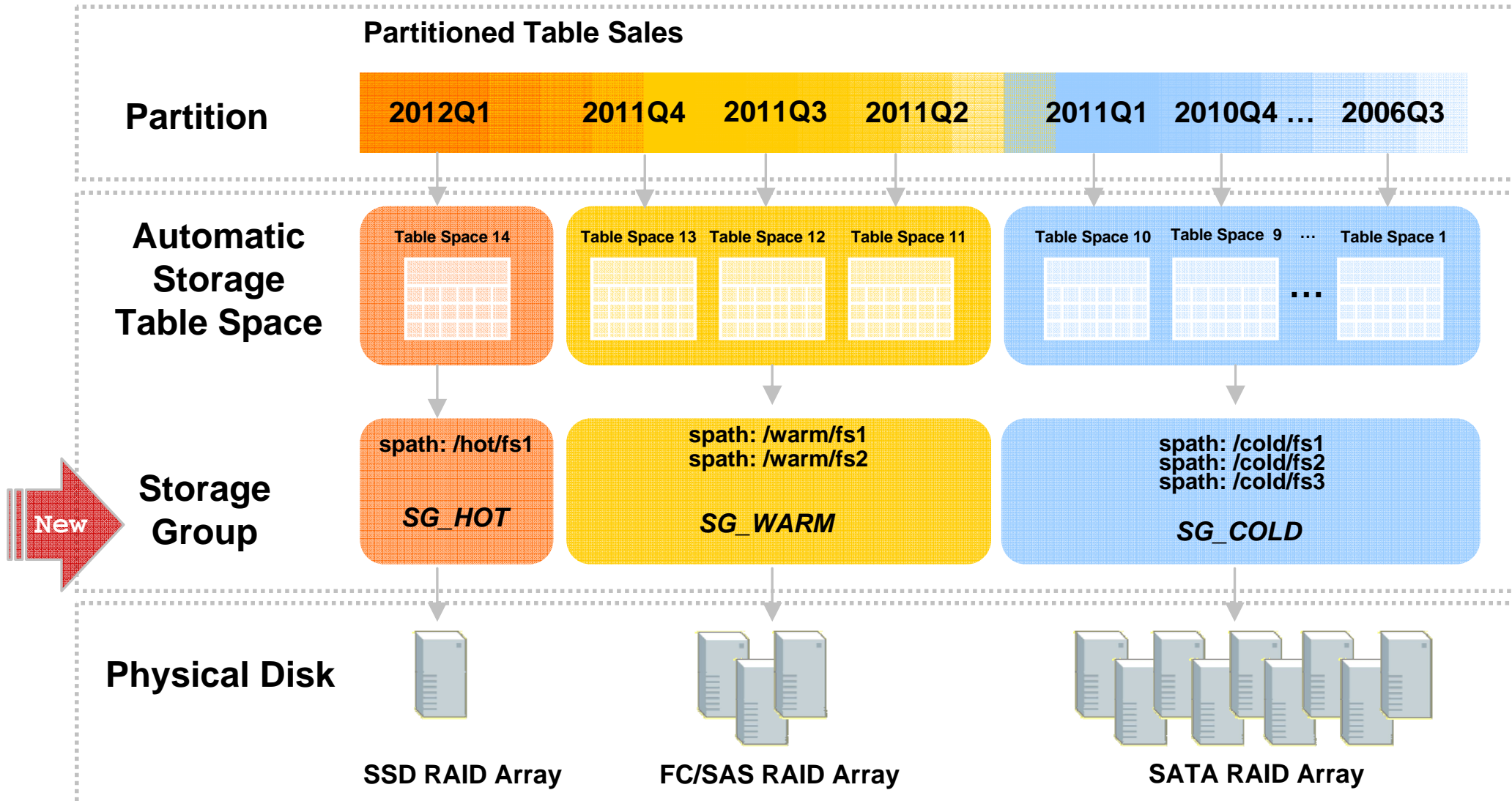
container-clause:



on-db-partitions-clause:



Storage Groups (multi-temperature storage)



Creating a storage group

- Use the `CREATE STOGROUP` statement to create a new storage group

```
>>-CREATE--STOGROUP--storagegroup-name----->
      .-,------.
      V           |
>--ON---'storage-path'+--●----->

>--+-----+--●----->
    '-OVERHEAD--number-of-milliseconds-'
>--+-----+--●----->
    '-DEVICE READ RATE--number-megabytes-per-second-'

>--+-----+--●----->
    '-DATA TAG--+-integer-constant+-'
        '-NONE-----'
>--+-----+--●-----><
    '-SET AS DEFAULT-'
```

```
CREATE STOGROUP HIGHEND
ON '/dbe/filesystem1', '/db2/filesystem2'
OVERHEAD 0.75 DEVICE READ RATE 500
```

```
CREATE STOGROUP MIDRANGE ON 'D:\', 'E:\' SET AS DEFAULT
```

Assigning a table space to a storage group

- A specific storage group can be selected with the USING STOGROUP option of CREATE TABLESPACE

```
create tablespace tsp04
managed by automatic storage using stogroup app_data
initialsize 100 K maxsize none
extentsize 2;
```

- The database storage path defined when the database is created named , *IBMSTOGROUP*
 - This will be the default storage group for automatic storage table spaces when USING STOGROUP is not specified
 - The ALTER STOGROUP statement option SET AS DEFAULT can be used to change which storage group is the default for a database

Managing storage groups using ALTER STOGROUP

- Use the ALTER STOGROUP statement to make changes to a storage group
 - Similar to database storage paths in previous releases
 - New disk storage paths can be added
 - Existing storage paths can be dropped
 - Use ALTER TABLESPACE REBALANCE for each table space to release containers on dropped path

```
>>-ALTER--STOGROUP--storagegroup-name----->
.-----
|          .-,-----
V          V          |          (1) |
>-----+--ADD---'storage-path'+-----+-----><
|          .-,-----
|          V          |          |
+-DROP---'storage-path'+-----+
+-OVERHEAD--number-of-milliseconds-----+
+-DEVICE READ RATE--number-megabytes-per-second-+
+-DATA TAG--+-integer-constant+-----+
|          '-NONE-----'          |
'-SET AS DEFAULT-----'
```


Query storage groups with SQL using the table function ADMIN_GET_STORAGE_PATHS

```
select varchar(storage_group_name,20) as storage_group,  
       storage_group_id,  
       varchar(db_storage_path,20) as storage_path,  
       db_storage_path_state,  
       (fs_total_size / 1000000) as total_path_MB,  
       (sto_path_free_size / 1000000) as path_free_MB  
from table(admin_get_storage_paths('','-1')) as T1
```

STORAGE_GROUP	STORAGE_GROUP_ID	STORAGE_PATH	DB_STORAGE_PATH_STATE
IBMSTOGROUP	0	/dbauto/path1	IN_USE
APP_DATA	1	/dbauto/path2	IN_USE

TOTAL_PATH_MB	PATH_FREE_MB
20940	5649
20940	5649

2 record(s) selected.

Listing storage groups with the db2pd command

- **db2pd -db testdb -storagegroups**

```
Database Member 0 -- Database MUSICDB -- Active -- Up 0 days 00:09:09 -- Date
03/23/2012 09:13:49
```

Storage Group Configuration:

Address	SGID	Default	DataTag	Name
0x8F241740	0	Yes	0	IBMSTOGROUP
0x8F240490	1	No	0	SG_HIGH
0x90C39640	2	No	0	SG_LOW

Storage Group Statistics:

Address	SGID	State	Numpaths	NumDropPen
0x8F241740	0	0x00000000	2	0
0x8F240490	1	0x00000000	2	0
0x90C39640	2	0x00000000	2	0

Storage Group Paths:

Address	SGID	PathID	PathState	PathName
0x8F241850	0	0	InUse	/dbauto/path1
0x8F241BF0	0	1	InUse	/dbauto/path2
0x94F6F210	1	1024	InUse	/dbauto/path1/sg_high
0x94F6F510	1	1025	InUse	/dbauto/path2/sg_high
0x90C39750	2	2048	InUse	/dbauto/path1/sg_low
0x90C39AF0	2	2049	InUse	/dbauto/path2/sg_low

Storage Management alternatives: Automatic

- Automatic Storage Managed:
 - Administration is very easy, no need to define the number or names of the containers
 - Disk space assigned from disk paths for a storage group
 - Monitoring of available space at the storage group level instead of each table space
 - Multiple containers will be created using all available paths for the storage group
 - Automatic Storage can be enabled when the database is created or added to an existing database
 - Default is ON for CREATE DATABASE with DB2 9
 - Storage paths can be added or removed using ALTER STOGROUP
 - Uses standard DMS and SMS under the covers:
 - DMS used for REGULAR and LARGE table spaces
 - SMS used for SYSTEM and USER TEMPORARY table spaces
 - Table space allocation controlled by CREATE/ALTER options:
 - INITIALSIZE: Defaults to 32 MB
 - AUTORESIZE: Can be set to YES or NO
 - INCREASESIZE: Can be set to amount or percent increase
 - MAXSIZE: Can define growth limits

Automatic Storage: Table space examples

- Syntax for CREATE and ALTER TABLESPACE:

```
CREATE TABLESPACE <tsName> [MANAGED BY AUTOMATIC STORAGE]
[USING STOGROUP storagegroup-name ]
[INITIALSIZE integer {K|M|G}]
[AUTORESIZE {NO|YES}] [INCREASESIZE integer {PERCENT|K|M|G}]
[MAXSIZE {NONE | integer {K|M|G}}]
```

- Default initial size is 32 MB
- Default max size is none
- Default increase size is determined by DB2, which might change over the life of the table space
- Examples:

```
CREATE TABLESPACE USER1 USING STOGROUP APP_DATA
CREATE TEMPORARY TABLESPACE TEMPTS
CREATE TABLESPACE MYTS INITIALSIZE 100 M MAXSIZE 1 G
CREATE LARGE TABLESPACE LRGTs INITIALSIZE 5 G AUTORESIZE NO
CREATE REGULAR TABLESPACE USER2 INITIALSIZE 500 M
```

ALTER TABLESPACE

- ALTER TABLESPACE can be used to change table space characteristics:
 - For all types table space management, you can adjust:
 - Bufferpool assigned
 - Prefetch size
 - Overhead and Transfer rate I/O Costs
 - File System Caching option
 - For DMS-managed table spaces, you can:
 - Use the ADD, DROP, RESIZE, EXTEND, REDUCE and BEGIN NEW STRIPE SET to directly adjust container names and sizes.
 - Use MANAGED BY AUTOMATIC STORAGE to convert to Automatic Storage management
 - For Automatic Storage-managed table spaces, you can:
 - Use the REDUCE option to release unused disk space
 - Use REBALANCE to reallocate containers when Automatic Storage paths are added or dropped
 - For DMS and Automatic Storage, you can:
 - Change the MAXSIZE, INCREASESIZE and AUTORESIZE settings
 - For Automatic Storage
 - Change the storage group, USING STOGROUP

Adding or dropping Automatic Storage paths

- New Automatic Storage paths can be added to a storage group using the ALTER STOGROUP

```
alter STOGROUP APP_DATA add storage on  
'/dbauto/path3', '/dbauto/path4'
```

- Automatic Storage can be enabled in an existing database by creating the first storage group, which becomes the default storage group
 - Existing Automatic table spaces will grow using the previously assigned storage paths until remaining space is used
 - Newly created table spaces will begin to use all defined paths
 - Individual table spaces can be altered using REBALANCE to spread data over all storage paths
- Storage paths can also be removed using ALTER STOGROUP

```
alter STOGROUP IBMSTOGROUP drop storage on '/dbauto/path1'
```

- The dropped path will be in *drop pending* state until a ALTER TABLESPACE with the REBALANCE option is used for each table space with containers on the dropped path

Using the SYSCAT.TABLESPACES view

```
SELECT substr(tbspacename,1,18) as tbspacename, substr(definer,1,10) as definer,  
tbspacename, tablespacetype, datatype, sgname  
from syscat.tablespaces
```

TBSPACE	DEFINER	TBSPACEID	TBSPACETYPE	DATATYPE	SGNAME
-----	-----	-----	-----	-----	-----
SYSCATSPACE	SYSIBM	0	D	A	IBMSTOGROUP
TSP06	INST28	9	D	A	IBMSTOGROUP
SYSTOOLSPACE	INST28	3	D	L	IBMSTOGROUP
USERSPACE1	SYSIBM	2	D	L	IBMSTOGROUP
TEMPSPACE1	SYSIBM	1	S	T	IBMSTOGROUP
TSP04	INST28	7	D	L	APP_DATA
TSP05	INST28	8	D	L	APP_DATA
TSP01	INST28	4	D	A	-
TSP02	INST28	5	D	L	-
TSP03	INST28	6	D	L	-
SMS01	INST28	10	S	A	-

11 record(s) selected.

Using the db2pd command to list tablespace status and statistics

db2pd -db musicdb -tablespaces

Database Member 0 -- Database MUSICDB -- Active -- Up 0 days 00:04:04 -- Date 05/03/2012
17:38:55

Tablespace Configuration:

Address	Id	Type	Content	PageSz	ExtentSz	Auto	Prefetch	BufID	BufIDDisk	..Name
0x9396EEC0	0	DMS	Regular	4096	4	Yes	4	1	1	..SYSCATSPACE
0x93977C90	1	SMS	SysTmp	4096	32	Yes	32	1	1	..TEMPSPACE1
0x93982990	2	DMS	Large	4096	32	Yes	32	1	1	..USERSPACE1
0x95727700	3	DMS	Large	4096	4	Yes	4	1	1	..SYSTOOLSPACE
0x957B92B0	4	DMS	Regular	4096	4	Yes	4	1	1	..TSP01
0x957C58B0	5	DMS	Large	4096	2	Yes	2	1	1	..TSP02
0x957D8760	6	DMS	Large	4096	8	Yes	8	1	1	..TSP03
0x957E0EC0	7	DMS	Large	4096	2	Yes	2	1	1	..TSP04
0x957E9620	8	DMS	Large	4096	2	Yes	2	1	1	..TSP05
0x957F1D80	9	DMS	Regular	4096	4	Yes	4	1	1	..TSP06
0x957FA4E0	10	SMS	Regular	4096	4	Yes	4	1	1	..SMS01

Tablespace Statistics:

Address	Id	TotalPgs	UsablePgs	UsedPgs	PndFreePgs	FreePgs	HWM
0x9396EEC0	0	24576	24572	20696	0	3876	20696
0x93977C90	1	1	1	1	0	0	0

....

Using the MON_GET_TABLESPACE function

- The MON_GET_TABLESPACE table function returns monitor metrics for one or more table spaces.

Syntax

```
>>-MON_GET_TABLESPACE-- (--tbsp_name--, --member--) ---><
```

- Example

To list table spaces ordered by number of physical reads from table space containers.

```
SELECT varchar(tbsp_name, 30) as tbsp_name,
        member,
        tbsp_type,
        pool_data_p_reads
FROM TABLE(MON_GET_TABLESPACE(' ', -2)) AS t
ORDER BY pool_data_p_reads DESC
```

TBSP_NAME	MEMBER	TBSP_TYPE	POOL_DATA_P_READS
-----	-----	-----	-----
SYSCATSPACE	0	DMS	79
USERSPACE1	0	DMS	34
TEMPSPACE1	0	SMS	0

Additional DB2 System table spaces

- The SYSTOOLSPACE table space is a user data table space used by the DB2 administration tools and some SQL administrative routines for storing historical data and configuration information:
 - ADMIN_COPY_SCHEMA procedure
 - ADMIN_DROP_SCHEMA procedure
 - ADMIN_MOVE_TABLE procedure
 - ADMIN_MOVE_TABLE_UTIL procedure
 - Administrative task scheduler
 - ALTOBJ procedure
 - Automatic Reorganization (including the db.tb_reorg_req health indicator)
 - Automatic Statistics Collection (including the db.tb_runstats_req health indicator)
 - Configure Automatic Maintenance wizard
 - db2look command
 - GET_DBSIZE_INFO procedure
 - Storage Management tool
 - SYSINSTALLOBJECTS procedure
- The SYSTOOLSTMPSPACE table space is a user temporary table space used by the REORGCHK_TB_STATS, REORGCHK_IX_STATS and the ADMIN_CMD stored procedures for storing temporary data.

Database buffer pools

- Databases have one buffer pool, IBMDEFAULTBP when the database is created
 - The page size is 4K unless CREATE DATABASE includes the PAGESIZE option
- Use the CREATE BUFFERPOOL statement to create a new buffer pool
.-IMMEDIATE-.

```
>>-CREATE BUFFERPOOL--bufferpool-name--+-----+----->
                                     '-DEFERRED--'
.-SIZE--1000--AUTOMATIC-----.
>--+-----+-----+-----+-----+-----+-----+----->
+-SIZE--number-of-pages-----+
|          .-1000-----|
'-SIZE--+-----+-----AUTOMATIC-'
        '-number-of-pages-'
>--●--+-----+-----+-----+-----+-----+-----><
        '-PAGESIZE--integer--+---+-'
                                     '-K-'
```

- Use ALTER BUFFERPOOL to change buffer pool size
- Table spaces must be assigned to a buffer pool with a matching pagesize
 - Use the BUFFERPOOL option of CREATE TABLESPACE to assign a buffer pool
 - ALTER TABLESPACE can be used to change the assigned buffer pool

Designing Buffer Pools

- Advantages of large buffer pools
 - Frequently requested data pages to be kept in the buffer pool, which allows quicker access.
 - Fewer I/O operations can reduce I/O contention, thereby providing better response time
 - They provide the opportunity to achieve higher transaction rates with the same response time.
- Considerations for using more than one buffer pool:
 - Temporary table spaces can be assigned to a separate buffer pool to provide better performance for queries, especially sort-intensive queries
 - If data must be accessed repeatedly and quickly by many short update-transaction applications, consider assigning the table space that contains the data to a separate buffer pool.
 - You can isolate data into separate buffer pools to favor certain applications, data, and indexes..
 - You can use smaller buffer pools for data that is accessed by seldom-used applications, especially applications that require very random access into a very large table.

Maintain or List System Database Directory

- CLP

```
db2 ? CATALOG DATABASE
```

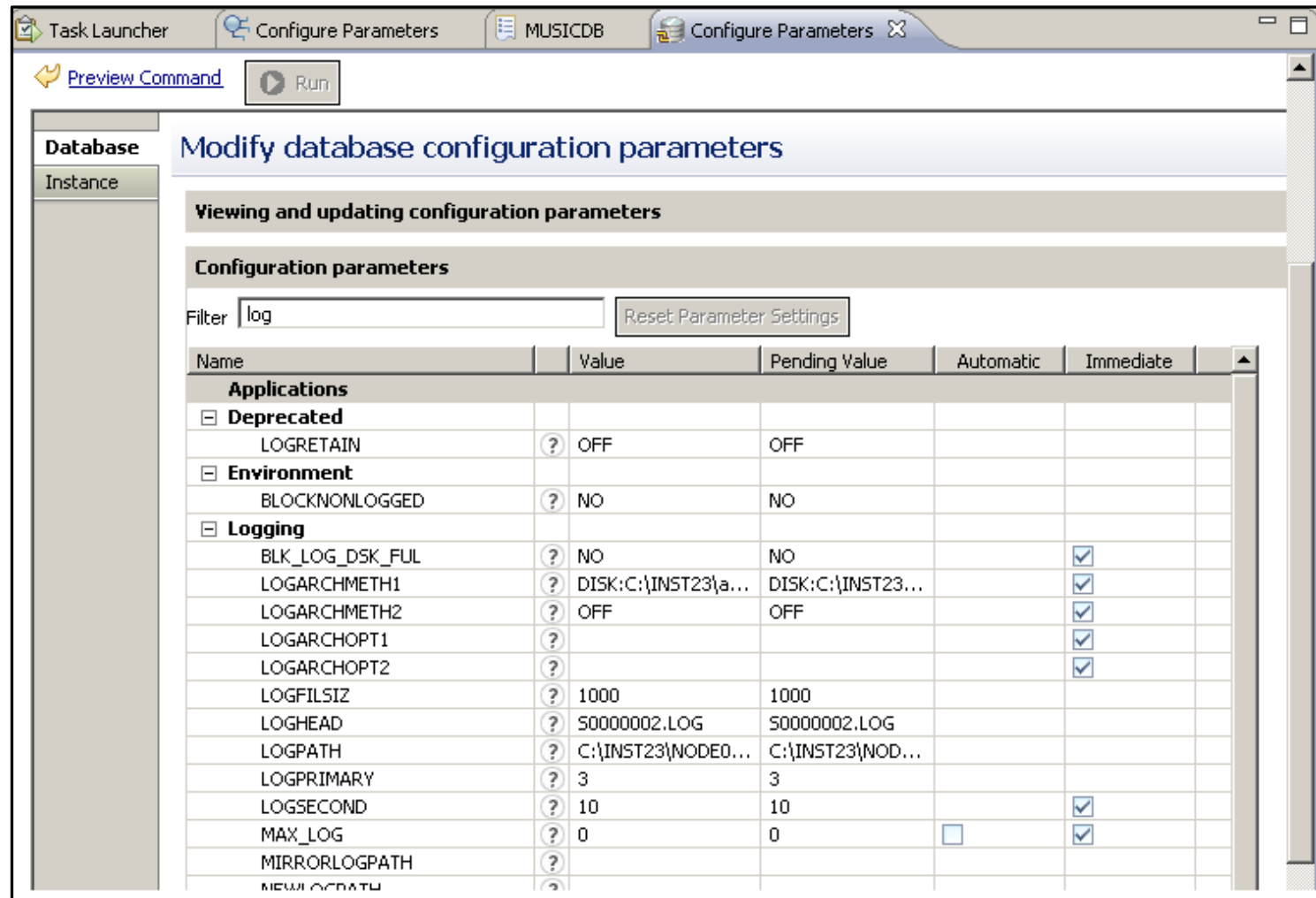
```
CATALOG DATABASE database-name [AS alias] [ON path | AT NODE nodename]  
[AUTHENTICATION {SERVER | CLIENT | ... | SERVER_ENCRYPT}]  
[WITH "comment-string"]
```

```
db2 'CATALOG DATABASE ourdb AS TESTDB ON /database WITH "Test Database"'
```

```
db2 LIST DATABASE DIRECTORY
```

```
⋮  
Database alias           = TESTDB  
Database name            = OURDB  
Local database directory = /database  
Database release level   = a.00  
Comment                  = Test Database  
Directory entry type      = Indirect  
Catalog database partition number = 4
```

Database configuration



db2 get db cfg for musicdb show detail

db2 update db cfg for musicdb using <parm> <value>

ACTIVATE and DEACTIVATE the database

- ACTIVATE DATABASE

- Activates the specified database and starts up all necessary database services, so that the database is available for connection and use by any application.

- Incurs the database startup processing

```
db2 activate db <db_name> user <user> using  
    <password>
```

- DEACTIVATE DATABASE

- Databases initialized by ACTIVATE DATABASE can be shut down using the DEACTIVATE DATABASE command, or using the **db2stop** command.

```
db2 deactivate db <db_name>
```

Unit summary

Having completed this unit, you should be able to:

- Review specifics of creating a database
- Explore the System Catalog tables and views
- Check and update Database configuration parameter settings
- Compare DMS, SMS and Automatic Storage managed table spaces
- Describe how to setup and manage a DB2 database with Automatic Storage enabled
- Define Storage Groups to manage databases with different classes of storage available
- Differentiate between table spaces, containers, extents, and pages
- Create and alter table spaces
- Create buffer pools to handle multiple page sizes or improve table access efficiency
- Use DB2 commands and SQL statements to display current table space statistics and status information

Student exercise

