



Database Administration TableSpace & Data File Management

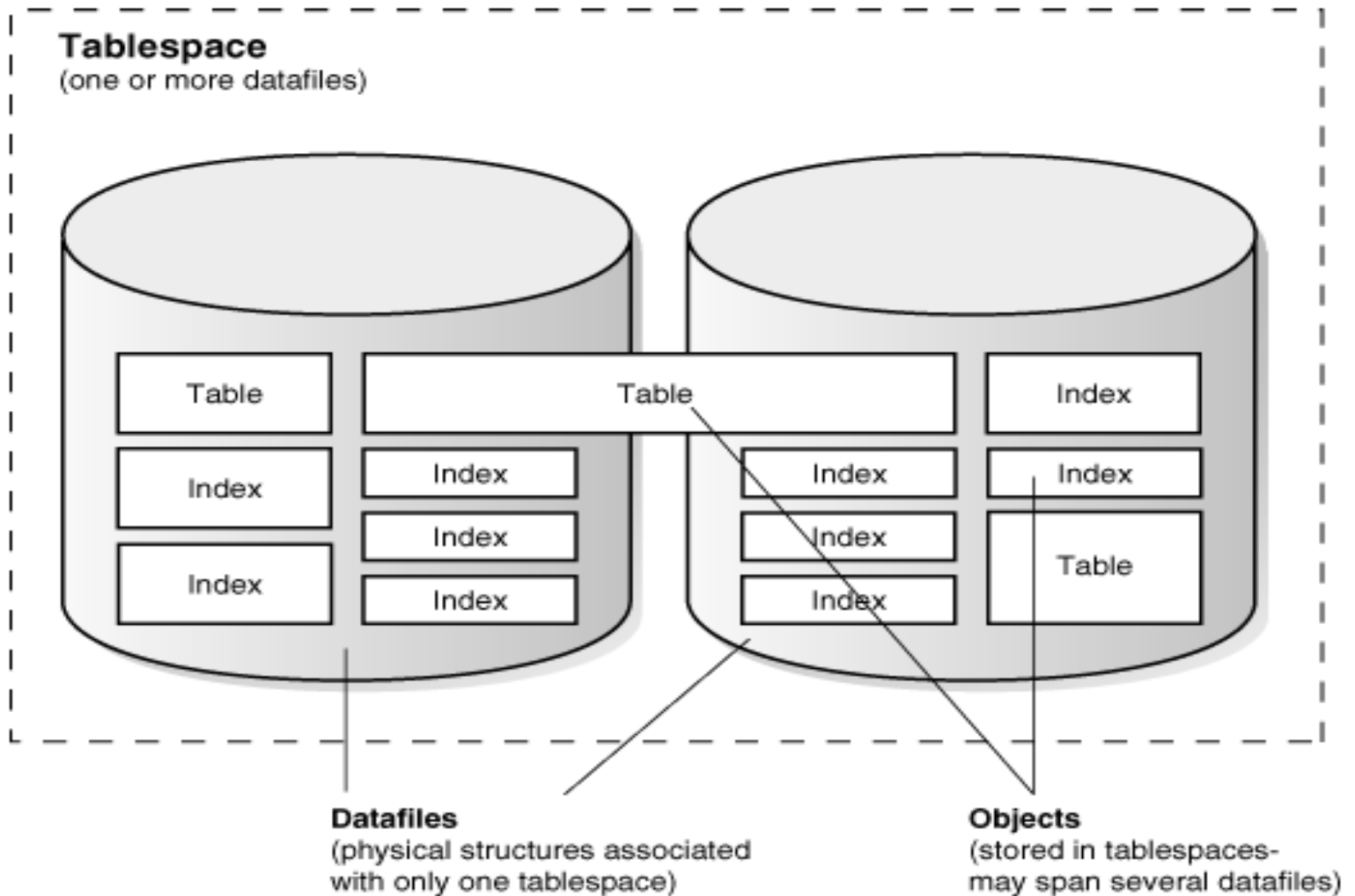
Table spaces

- A tablespace is a *logical storage unit* within an Oracle database. It is logical because a table space is not visible in the file system of the machine on which the database resides. A table space, in turn, consists of at least one data file which, in turn, are physically located in the file system of the server. Btw, a datafile belongs to exactly one tablespace.
- Each table, index and so on that is stored in an Oracle database belongs to a table space. The table space builds the bridge between the Oracle database and the file system in which the table's or index' data is stored.

There are three types of table spaces in Oracle:

- Permanent table spaces
- Undo table spaces
- temporary table spaces

Table spaces



Create Table Space

Permanent tablespace

- create tablespace ts_something datafile '/dbf1/ts_sth.dbf' size 32m autoextend on next 32m maxsize 2048m extent management local;
- create tablespace data datafile '/home/oracle/databases/ora10/data.dbf' size 10M autoextend on maxsize 200M extent management local;

Create Table Space

Temporary table space

- create temporary tablespace temp_mtr **tempfile** '/dbf1/mtr_temp01.dbf' size 32m autoextend on next 32m maxsize 2048m extent management local;
- Note, a temporary tablespace has tempfiles, not data files.

Undo table space

- create undo tablespace ts_undo datafile '/dbf/undo.dbf' size 100M;

Renaming Table spaces

- This is a feature that is available with Oracle 10g and can be useful in transportable table space scenarios.
- `alter tablespace ts_user rename to ts_user_01;`
- The system and sysaux table space cannot be renamed, though. But that will not be much of a problem, will it? However, read only table spaces *can* be renamed.
- After renaming a table space, the controlfiles should immediately be backed up.

The system and sysaux tablespaces

- The system tablespace is always available when a database is open (it cannot be taken offline).
- The system tablespace stores the data dictionary (or their base tables, respectively).

The sysaux tablespace

- The **sysaux** tablespace is new with Oracle 10g. It is used to store database components that were stored in the system tablespace in prior releases of the database. Also, the tablespaces that were needed for RMAN's recovery catalog, for Ultra Search, for Data Mining, for XDP and for OLAP are going to sysaux with 10g. Additionally, it is the place where automatic workload repository stores its information.

Bigfile tablespaces

- This is a feature of Oracle 10g. A bigfile tablespace contains only one datafile (or tempfile) which can be as big as 2^{32} (=4GB) blocks.
- create **bigfile** tablespace beeeeg_ts data file '/o1/dat/beeeeg.dbf' size **2T**
- Bigfile tablespaces are supported only for locally managed tablespaces with automatic segment-space management (which is the default setting since Oracle 9i). There are two exceptions: locally managed undo and temporary tablespaces can be bigfile tablespaces, even though their segments are manually managed. The *system* and *sysaux* tablespace cannot be created as bigfile tablespace.
- Bigfile tablespaces should be used with automatic storage management, or other logical volume managers that support dynamically extensible logical volumes, striping and RAID.

Space Management

- Oracle maintains extents for a tablespace. There are two different methods for Oracle to keep track of free and used (occupied) extents:
- Dictionary managed tablespace
- Locally managed tablespace

Space Management

Dictionary managed tablespaces

Extents are allocated according to the following **storage parameters**

- initial
- next
- pctincrease

The information about used and free extents is stored in the dictionary.

- **Locally managed tablespaces**
- A 'bitmap' is stored within the tablespace. Each bit within this bitmap determines if a corresponding extent in the tablespace is free or used.

Space Management

- The extent sizes are either *uniform* or *autoallocate*.
- Locally managed tablespaces have two advantages: *recursive space management* is avoided and adjacent free space is automatically coalesced.

State of table spaces

- A table space is either *online* (ready to be accessed) or *offline* (not accessible).
- A table space becomes offline either when the DBA explicitly makes the tablespace offline, or
- when an error occurs while Oracle accesses a datafile within the tablespace.
- Offline table spaces cannot be transported to other databases.

Read Only Table spaces

- Read only table spaces have the benefit that they need be backed up only once.

Determine the Value of the DB_FILES Initialization Parameter

- When starting an Oracle instance, the DB_FILES initialization parameter indicates the amount of SGA space to reserve for datafile information and thus, the maximum number of datafiles that can be created for the instance. This limit applies for the life of the instance. You can change the value of DB_FILES (by changing the initialization parameter setting), but the new value does not take effect until you shut down the instance and restart.
- When determining a value for DB_FILES, take the following into consideration:
- If the value of DB_FILES is too low, you cannot add datafiles beyond the DB_FILES limit without first shutting down the database.
- If the value of DB_FILES is too high, memory is unnecessarily consumed.

Enabling and Disabling Automatic Extension for a Datafile

- You can create data files or alter existing data files so that they automatically increase in size when more space is needed in the database. The files increase in specified increments up to a specified maximum.
- Setting your data files to extend automatically provides these advantages:
 - Reduces the need for immediate intervention when a table space runs out of space
 - Ensures applications will not halt because of failures to allocate extents
- To determine whether a data file is auto-extensible, query the `DBA_DATA_FILES` view and examine the `AUTOEXTENSIBLE` column.

Enabling and Disabling Automatic Extension for a Datafile

- You can specify automatic file extension by specifying an **AUTOEXTEND ON** clause when you create datafiles using the following SQL statements:
 - **CREATE DATABASE**
 - **CREATE TABLESPACE**
 - **ALTER TABLESPACE**
- You can enable or disable automatic file extension for existing datafiles, or manually resize a datafile using the **ALTER DATABASE** statement.
- The following example enables automatic extension for a datafile added to the users tablespace:
 - **ALTER TABLESPACE users ADD DATAFILE**
'u02/oracle/rbdb1/users03.dbf' SIZE 10M AUTOEXTEND
ON NEXT 512K MAXSIZE 250M;

Enabling and Disabling Automatic Extension for a Datafile

- The value of NEXT is the minimum size of the increments added to the file when it extends. The value of MAXSIZE is the maximum size to which the file can automatically extend.
- The next example disables the automatic extension for the datafile.
- ALTER DATABASE DATAFILE
'u02/oracle/rbdb1/users03.dbf' AUTOEXTEND OFF;

Manually Resizing a Datafile

- You can manually increase or decrease the size of a datafile using the `ALTER DATABASE` statement.
- Because you can change the sizes of datafiles, you can add more space to your database without adding more datafiles. This is beneficial if you are concerned about reaching the maximum number of datafiles allowed in your database.
- Manually reducing the sizes of datafiles enables you to reclaim unused space in the database. This is useful for correcting errors in estimates of space requirements.

Manually Resizing a Datafile

- In the next example, assume that the datafile `/u02/oracle/rbdb1/stuff01.dbf` has extended up to 250M. However, because its tablespace now stores smaller objects, the datafile can be reduced in size.
- The following statement decreases the size of datafile `/u02/oracle/rbdb1/stuff01.dbf`:
- ```
ALTER DATABASE DATAFILE
'/u02/oracle/rbdb1/stuff01.dbf' RESIZE 100M
```

# Altering the Availability of All Datafiles or Tempfiles in a Tablespace

- Clauses of the ALTER TABLESPACE statement allow you to change the online or offline status of all of the datafiles or tempfiles within a tablespace. Specifically, the statements that affect online/offline status are:
- ALTER TABLESPACE OFFLINE NORMAL;
- You are required only to enter the tablespace name, not the individual datafiles or tempfiles. All of the datafiles or tempfiles are affected, but the online/offline status of the tablespace itself is not changed.
- ALTER DATABASE DATAFILE 'PATH' OFFLINE DROP; **NON ARCHIVELOG MODE**
- ALTER DATABASE DATAFILE 'PATH' OFFLINE; **ARCHIVELOG MODE**

## Altering the Availability of All Datafiles or Tempfiles in a Tablespace

- In most cases the above ALTER TABLESPACE statements can be issued whenever the database is mounted, even if it is not open. However, the database *must not* be open if the tablespace is the system tablespace, an undo tablespace, or the default temporary tablespace. The ALTER DATABASE DATAFILE and ALTER DATABASE TEMPFILE statements also have ONLINE/OFFLINE clauses, however in those statements you must enter all of the filenames for the tablespace.
- The syntax is different from the ALTER TABLESPACE ... ONLINE|OFFLINE statement that alters a tablespace's availability, because that is a different operation. The ALTER TABLESPACE statement takes datafiles offline as well as the tablespace.

## Renaming Datafiles in a Single Tablespace

- To rename datafiles from a single tablespace, complete the following steps:
- Take the non-SYSTEM tablespace that contains the datafiles offline. `ALTER TABLESPACE users OFFLINE NORMAL;`
- Rename the datafiles using the operating system.
- Use the `ALTER TABLESPACE` statement with the `RENAME DATAFILE` clause to change the filenames within the database.
- `ALTER TABLESPACE users RENAME DATAFILE`  
`'/u02/oracle/rbdb1/user1.dbf', '/u02/oracle/rbdb1/user2.dbf'`  
`TO '/u02/oracle/rbdb1/users01.dbf',`  
`'/u02/oracle/rbdb1/users02.dbf';`

## Renaming Datafiles in a Single Tablespace

- The new files must already exist; this statement does not create the files. Also, always provide complete filenames (including their paths) to properly identify the old and new datafiles. In particular, specify the old datafile name exactly as it appears in the `DBA_DATA_FILES` view of the data dictionary.
- Back up the database. After making any structural changes to a database, always perform an immediate and complete backup.



# Dropping Data files

- You can not drop the first datafile of a tablespace.
- alter tablespace tty drop datafile '/u01/oradata/tt3.dbf'

## Dropping TABLESPACES

- DROP TABLESPACE users INCLUDING CONTENTS AND DATAFILES;