

Administración de Bases de Datos

Tema 2
(parte 4 de 4)

Estructuras y Gestión del
Almacenamiento en un SGBD

2.8. Gestión del Almacenamiento

¿Qué es un *tablespace* (espacio de tablas)?

- ❑ Un espacio de tablas es una unidad lógica de almacenamiento compuesta por uno o varios ficheros del S.O. (ficheros de datos). Un fichero de datos pertenece a un único espacio de tablas
- ❑ Los espacios de tablas pueden ser:
 - **gestionados por el diccionario:** la información de las extensiones libres y asignadas se almacena en tablas del DD
 - **gestionados de modo local:** la información de las extensiones libres y asignadas se almacena en la cabecera de los ficheros de datos del espacio de tablas
- ❑ Oracle 10g permite crear espacios de tablas de gran tamaño (**bigfile tablespaces**), compuesto de un único fichero de datos de hasta 8EB (1 exabyte = 1 millón de terabytes)
- ❑ Exceptuando el espacio de tablas SYSTEM o aquellos que contengan segmentos de rollback activos, un espacio de tablas puede estar **offline** con la BD en funcionamiento.
- ❑ Un espacio de tablas se puede poner en modo sólo lectura (**readonly**) o lectura-escritura (**readwrite**) y viceversa

Tipos de espacios de tablas

❑ Permanentes

- Contienen segmentos que permanecen más allá de la duración de una sesión (datos, aplicaciones, SYSTEM, SYSAUX)

❑ Deshacer o de Anulación

- Pueden contener segmentos que se retengan más allá de la duración de una sesión
- Proporciona coherencia de lectura para las instrucciones SELECT que accedan a tablas que están siendo modificadas
- Los segmentos de deshacer almacenan los valores previos de las columnas que están siendo modificadas o borradas, de modo que si la sesión de usuario falla antes que se ejecute una instrucción COMMIT o ROLLBACK, las operaciones de actualización, inserción y borrados serán canceladas

❑ Temporales

- Contienen datos transitorios que sólo existen mientras dure la sesión
- Principalmente almacenan las operaciones de ordenación

Recomendaciones de Oracle

- ❑ Las principales directivas de Oracle sobre la organización de los espacios de tablas son:
 - No introducir datos de usuario en los espacios de tablas SYSTEM y SYSAUX
 - Crear, como mínimo, los siguientes espacios de tabla:
 - Uno para los segmentos de anulación (espacio de tablas de anulación – UNDO)
 - Uno para los segmentos temporales (especie de tablas temporal)
 - Uno para las tablas
 - Uno para los índices
 - Si es posible, repartir los ficheros de datos de estos espacios de tablas en discos diferentes
- ❑ La utilización de diversos espacios de tablas permite:
 - Separar los datos de las aplicaciones de los datos del diccionario de Oracle
 - Separar los datos de diferentes aplicaciones almacenados en la misma base de datos
 - Separar el almacenamiento de los diferentes tipos de objetos
 - Repartir las entradas/salidas en varios discos
 - Realizar copias de seguridad parciales

Espacios de tablas SYSTEM y SYSAUX

- ❑ Los espacios de tablas SYSTEM y SYSAUX son los únicos que, cómo mínimo, se crean con la BD, es decir, con la sentencia **CREATE DATABASE**
- ❑ El espacio de tablas **SYSTEM** no debe contener datos de aplicaciones ni de usuarios. Almacena los siguientes objetos:
 - Diccionario de Datos (DD)
 - Procedimientos almacenados, funciones, disparadores (*triggers*) y paquetes
 - Segmentos de *rollback system*
- ❑ El espacio de tablas **SYSAUX** (>=10g) libera al espacio de tablas SYSTEM, almacenando las utilidades del sistema (Repositorio OEM, Intermedia, Spatial, OLAP, RMAN, XML DB, etc.). Para saber qué hay en el espacio de tablas SYSAUX y cuánto ocupa se puede consultar la vista **v\$sysaux_occupants**

```
select occupant_name, space_usage_kbytes from v$sysaux_occupants;
```

Creación de un espacio de tablas

```
CREATE [UNDO] [BIGFILE] TABLESPACE nombre
    DATAFILE 'nombre_fichero' [SIZE entero [K|M|G]] [REUSE]
        [AUTOEXTEND [OFF | ON NEXT entero [K|M|G] ] ]
        [MAXSIZE [UNLIMITED | entero [K|M|G] ] ]
    [MINIMUM EXTENT integer[K|M]]
    [BLOCKSIZE integer[K]]
    [LOGGING | NOLOGGING]
    [DEFAULT storage_clause]
    [ONLINE | OFFLINE]
    [PERMANENT | TEMPORARY]
    [extent_management_clause]
```

Descripción de algunos parámetros

❑ Nombre

- Nombre del espacio de tablas

❑ DATAFILE

- Nombres y características de los ficheros asociados al espacio de tablas. Este valor es opcional si el parámetro de inicialización **DB_CREATE_FILE_DEST** tiene un valor asignado. En este caso ORACLE crea un fichero con 100M y un nombre por defecto en el directorio que se hubiera especificado en el parámetro, que será autoextensible y de tamaño ilimitado
- **SIZE:** Tamaño inicial del fichero. Si la opción REUSE está presente y el fichero ya existe, Oracle lo reutiliza y lo borra.
- **AUTOEXTEND:** Indica si el fichero de datos puede (**ON**) o no (**OFF**) crecer una vez que se ha utilizado el espacio inicialmente asignado
- **NEXT:** Espacio mínimo asignado al fichero en el momento de la extensión
- **MAXSIZE:** Tamaño máximo del fichero (**UNLIMITED** indica tamaño ilimitado)

❑ MINIMUM EXTENT

- Indica el tamaño mínimo de las extensiones en el espacio de tablas, obligando a que éstas (estén ocupadas o libres) tengan un tamaño múltiplo del número que se especifique. Esta cláusula sólo es válida para espacios de tablas gestionados por el diccionario

❑ BLOCKSIZE

- Define el tamaño del bloque usado por el espacio de tablas. EL valor por defecto es el tamaño de bloque estándar definido en el parámetro **DB_BLOCK_SIZE**

❑ LOGGIN | NOLOGGING

- Se usa para indicar que las operaciones que van a realizarse sobre los objetos del espacio de tablas deben registrarse en los ficheros de redo log

❑ ONLINE | OFFLINE

- ONLINE es el valor por defecto y permite que, después de la creación, esté disponible para todos los usuarios que tengan autorización. La vista del diccionario **DBA_TABLESPACES** indica la situación de cada espacio de tablas

❑ PERMANENT | TEMPORARY

- Si el espacio de tablas se va a usar para contener objetos permanentes, se debe especificar PERMANENT (valor por defecto)

❑ extent_management_clause

```
[EXTENT MANAGEMENT
```

```
{DICTIONARY | LOCAL {AUTOALLOCATE | UNIFORM [SIZE integer[K|M]] } } ]
```

- Permite especificar cómo se gestionarán las extensiones del espacio de tablas. Si se elige **LOCAL** (valor por defecto), la información necesaria para controlar el uso del espacio estará almacenada dentro del propio espacio de tablas
- Si se indica la opción **AUTOALLOCATE**, el espacio de tablas será gestionado por el sistema y no se permite especificar un tamaño de extensión. Este es el valor por defecto si el parámetro de inicialización **COMPATIBLE** está puesto a 9.0.0 o superior
- El parámetro **UNIFORM** indica que la gestión del espacio de tablas se realizará con extensiones uniformes, que tendrán el tamaño que se indiquen en **SIZE** (por defecto, 1M)
- Por defecto (cláusula EXTENT MANAGEMENT ausente), un espacio de tablas permanente se gestiona localmente con una gestión automática de las extensiones (AUTOLLOCATE)

❑ extent_management_clause

❑ Restricciones:

- 1) no se puede especificar **DICTIONARY** para ningún espacio de tablas si el espacio de tablas **SYSTEM** está gestionado localmente. Oracle recomienda crear espacios de tablas con la opción LOCAL, ya que son más eficientes
- 2) Un espacio de tablas gestionado localmente sólo puede contener objetos permanentes. Si se desea que un espacio de tablas gestionado localmente contenga objetos temporales debe usarse la sentencia **CREATE TEMPORARY TABLESPACE**

❑ Ventajas e inconvenientes de elegir extensiones grandes:

V

- Evitan las extensiones dinámicas
- Pequeñas ventajas de rendimiento: búsqueda de tabla completa
- Lecturas simples contra el mapa de extensiones

I

- Encontrar espacio libre contiguo
- Desperdicio de espacio inicialmente

❑ storage_clause

```
STORAGE (INITIAL n[K|M] [NEXT n[K|M]] [MINEXTENTS n]
[ MAXEXTENTS n | UNLIMITED] [PCTINCREASE n])
```

▪ INITIAL

- Tamaño de la primera extensión del objeto (en Kbytes, Mbytes o, por defecto, en bytes) cuando se crea. Valor predeterminado: 5 bloques de datos

▪ NEXT

- Tamaño de la siguiente extensión que se asignará al objeto. Valor predeterminado: 5 bloques de datos

▪ PCTINCREASE

- Especifica el porcentaje en el que crecen la tercera extensión y siguientes con respecto al tamaño de la última extensión asignada. El valor predeterminado es 50 y el mínimo 0 (cero), que indica que todas las extensiones serán del mismo tamaño

▪ MINEXTENTS

- Indica el número de extensiones que deben asignarse en la creación del objeto. El valor predeterminado es 1

▪ MAXEXTENTS

- Indica el número total de extensiones, incluida la primera, que Oracle puede reservar para el objeto. El valor mínimo es 1 y el predeterminado depende del tamaño de bloque especificado. El parámetro UNLIMITED reserva automáticamente las extensiones en función de las necesidades. Oracle recomienda esta opción para evitar las fragmentaciones

Ejemplos de creación de espacios de tablas

❑ Creación de un espacio de tablas de deshacer

```
CREATE UNDO TABLESPACE ETundo  
DATAFILE 'C:\app\uhu\ORADATA\ONUBA\UNDO.DBF' SIZE 10M  
AUTOEXTEND ON NEXT 5M  
EXTENT MANAGEMENT LOCAL;
```

❑ Creación de un espacio de tablas indicando parámetros de almacenamiento por defecto para los objetos que se creen posteriormente

```
CREATE TABLESPACE ET01  
DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET01.DBF' SIZE 20M  
EXTENT MANAGEMENT DICTIONARY  
DEFAULT STORAGE (INITIAL 128K NEXT 128K MAXEXTENTS 10 PCTINCREASE 0)  
ONLINE;
```

Ejemplos de creación de espacios de tablas

- ❑ Creación de un espacio de tablas con un fichero de datos de 1M de tal forma que cuando se llene, se le asignen extensiones de 500K hasta un máximo de 100M

```
CREATE TABLESPACE ET02  
DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET02.DBF' SIZE 1M REUSE  
AUTOEXTEND ON NEXT 500K MAXSIZE 100M;
```

- ❑ Creación de un espacio de tablas especificando un tamaño mínimo de extensión de 500K

```
CREATE TABLESPACE ET03  
DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET03.DBF' SIZE 20M  
MINIMUM EXTENT 500K;
```

Espacios de tablas de "deshacer" (*undo*)

- ❑ En Oracle 9i se introducen los segmentos de deshacer automáticos, que permiten sustituir la gestión manual de segmentos de *rollback*. Con Oracle 10g todavía se puede optar por la gestión manual, pero se avisa de que ya no estará disponible en futuras versiones
- ❑ Los espacios de tablas "undo" sólo pueden contener segmentos de *rollback* (ningún otro tipo de objeto)
- ❑ Se suelen llamar **espacios de tabla de *rollback***
- ❑ Son del tipo "manejados localmente", es decir, de forma automática

```
CREATE UNDO TABLESPACE UNDO_01  
DATAFILE 'C:\app\uhu\ORADATA\ONUBA\UNDO_01.DBF' SIZE 100M;
```

- ❑ Un espacio de tablas de *rollback* sólo se usa cuando se activa la gestión automática de *rollback* en la BD (parámetros **undo_management = auto** y **undo_tablespace = nombre_espacio_de_tablas** del fichero de inicialización)
- ❑ Sólo puede haber un espacio de tablas de *rollback* activo
- ❑ Para hacer que un espacio de tablas de *rollback* sea el espacio de tablas activo hay que modificar el parámetro **UNDO_TABLESPACE**

```
ALTER SYSTEM SET UNDO_TABLESPACE = UNDO_01
```

Espacios de tablas temporales

- ❑ Los segmentos temporales se crean en espacios de tablas temporales, de forma automática, para mejorar las operaciones de ordenación (*order by*, *joins*, *create index*, etc.) que no caben en memoria
- ❑ Un espacio de tablas temporal no puede contener objetos permanentes
- ❑ Se recomienda que sean "gestionados localmente" (no pueden usar AUTOALLOCATE ni SEGMENT SPACE MANAGEMENT AUTO)

```
CREATE TEMPORARY TABLESPACE temp  
TEMPFILE 'C:\app\uhu\ORADATA\ONUBA\temporal.DBF' SIZE 100M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 2M;
```

- ❑ Para optimizar el rendimiento es recomendable que **UNIFORM SIZE** sea múltiplo de **SORT_AREA_SIZE**
- ❑ No se pueden poner en modo "sólo lectura"
- ❑ Siempre tienen el modo NOLOGGING (no producen entradas de *redo log*)

- ❑ Desde Oracle10g se pueden agrupar espacios de tablas temporales, repartiendo las ordenaciones entre ellos
- ❑ La principal ventaja es que permiten proporcionar a un mismo usuario que tenga múltiples sesiones, la capacidad de utilizar un espacio de tablas temporal distinto para cada sesión
- ❑ Para crear un grupo de espacios de tabla temporales (suponemos que existen los espacios de tablas *temp1*, *temp2* y *temp3*):

```
ALTER TABLESPACE temp1 TABLESPACE GROUP gtemp;  
ALTER TABLESPACE temp2 TABLESPACE GROUP gtemp;  
ALTER TABLESPACE temp3 TABLESPACE GROUP gtemp;
```

- ❑ Para cambiar el espacio de tablas temporal predeterminado a **gtemp**:

```
ALTER DATABASE default temporary tablespace gtemp;
```

- ❑ Para eliminar un grupo de espacios de tablas temporales, primero se deben eliminar todos sus miembros. El grupo se elimina implícitamente cuando se elimina el último espacio de tablas del grupo:

```
ALTER TABLESPACE temp1 TABLESPACE GROUP '';  
ALTER TABLESPACE temp2 TABLESPACE GROUP '';  
ALTER TABLESPACE temp3 TABLESPACE GROUP '';
```

- ❑ Para asignar el grupo de espacio de tablas temporales por defecto a un usuario:

```
ALTER USER nombre_usuario TEMPORARY TABLESPACE gtemp;
```

- ❑ Un espacio de tabla puede estar en dos estados:
 - **ONLINE**, cuando está disponible y pueden usarse todos sus ficheros
 - **OFFLINE**, cuando no lo está

- ❑ Algunos espacios de tablas no se pueden poner **OFFLINE**: SYSTEM, espacios de tablas con segmentos de rollback en uso o temporales activos

- ❑ El cambio del estado se realiza con el comando **ALTER TABLESPACE**

```
ALTER TABLESPACE nombreET ONLINE | OFFLINE;
```

- ❑ Razones para estar OFFLINE:
 - Deshabilitar una parte de la BD y permitir el acceso al resto
 - Realizar un *backup* del espacio de tablas offline
 - Deshabilitar una aplicación temporalmente debido a mantenimiento

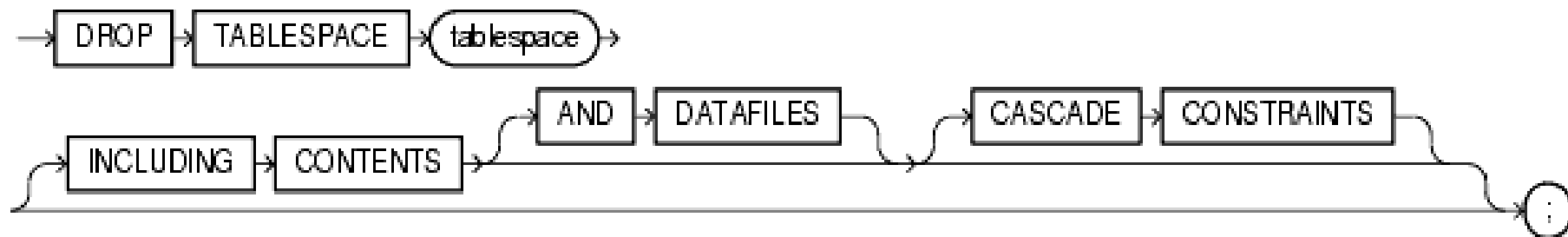
- ❑ Según su estado de escritura, un espacio de tabla puede estar en 2 modos
 - **sólo-lectura** (READ ONLY), de manera que se impidan las modificaciones de datos
 - **lectura-escritura** (READ WRITE), en el que se permite escribir en el espacio de tablas

- ❑ El comando **ALTER TABLESPACE** se utiliza para alternar entre ambos modos:

```
ALTER TABLESPACE nombreET READ [ONLY | WRITE];
```

Borrado de un espacio de tablas

Sintaxis:



❑ **tablespace**

- Nombre del espacio de tablas que se quiere eliminar. Oracle recomienda ponerlo offline para asegurar que no hay ninguna transacción realizando operaciones sobre alguno de los objetos del espacio de tablas. Una vez que ha sido eliminado, los usuarios que lo tuviesen asignado por defecto, no podrán seguir asignando espacio para sus objetos. Habrá que utilizar la sentencia **ALTER USER** para asignar a los usuarios un nuevo espacio de tablas por defecto

Ejemplo: alter user nombreusuario default tablespace XX y si se desea mover tablas de un tablespace a otro: Alter table Esquema.Nombretabla move tablespace XX

❑ **INCLUDING CONTENTS**

- Permite borrar todos los objetos del espacio de tablas. Si se omite y no está vacío, Oracle devuelve un mensaje de error y no lo elimina

❑ AND DATAFILES

- Si se especifican las dos cláusulas, Oracle borra, además de los objetos, los ficheros del sistema operativo asociado a este espacio de tablas

❑ CASCADE CONSTRAINT

- Permite borrar todas las restricciones de integridad referencial de tablas que están almacenadas en otros espacios de tablas y hacen referencia a tablas del espacio de tablas que se desea borrar. Si se omite y existen restricciones, Oracle devuelve un mensaje de error y no lo elimina

Ejemplos:

- ❑ Borrado de un espacio de tablas, eliminando todos los objetos que contiene y todas las restricciones de integridad referencial de objetos almacenados fuera de este espacio de tablas

```
ALTER TABLESPACE ET01 OFFLINE;  
COMMIT;  
DROP TABLESPACE ET01 INCLUDING CONTENTS CASCADE CONSTRAINTS;
```

- ❑ Borrado de un espacio de tablas incluyendo contenidos y ficheros de datos del sistema operativo

```
ALTER TABLESPACE ET02 OFFLINE NORMAL;  
COMMIT;  
DROP TABLESPACE ET02 INCLUDING CONTENTS AND DATAFILES;
```

Redimensionado de un espacio de tablas

- ❑ Existen varias estrategias para cambiar el tamaño de un espacio de tablas:
 - Añadiendo un fichero al espacio de tablas (ALTER TABLESPACE)

```
ALTER TABLESPACE ET02  
    ADD DATAFILE 'ET002.DBF'  
    SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE 100M;
```

- Modificando el tamaño del fichero que lo compone (ALTER DATABASE)

```
ALTER DATABASE DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET03.DBF' RESIZE 20M;
```

- Se puede programar el crecimiento del fichero que compone el espacio de tablas (ALTER DATABASE)

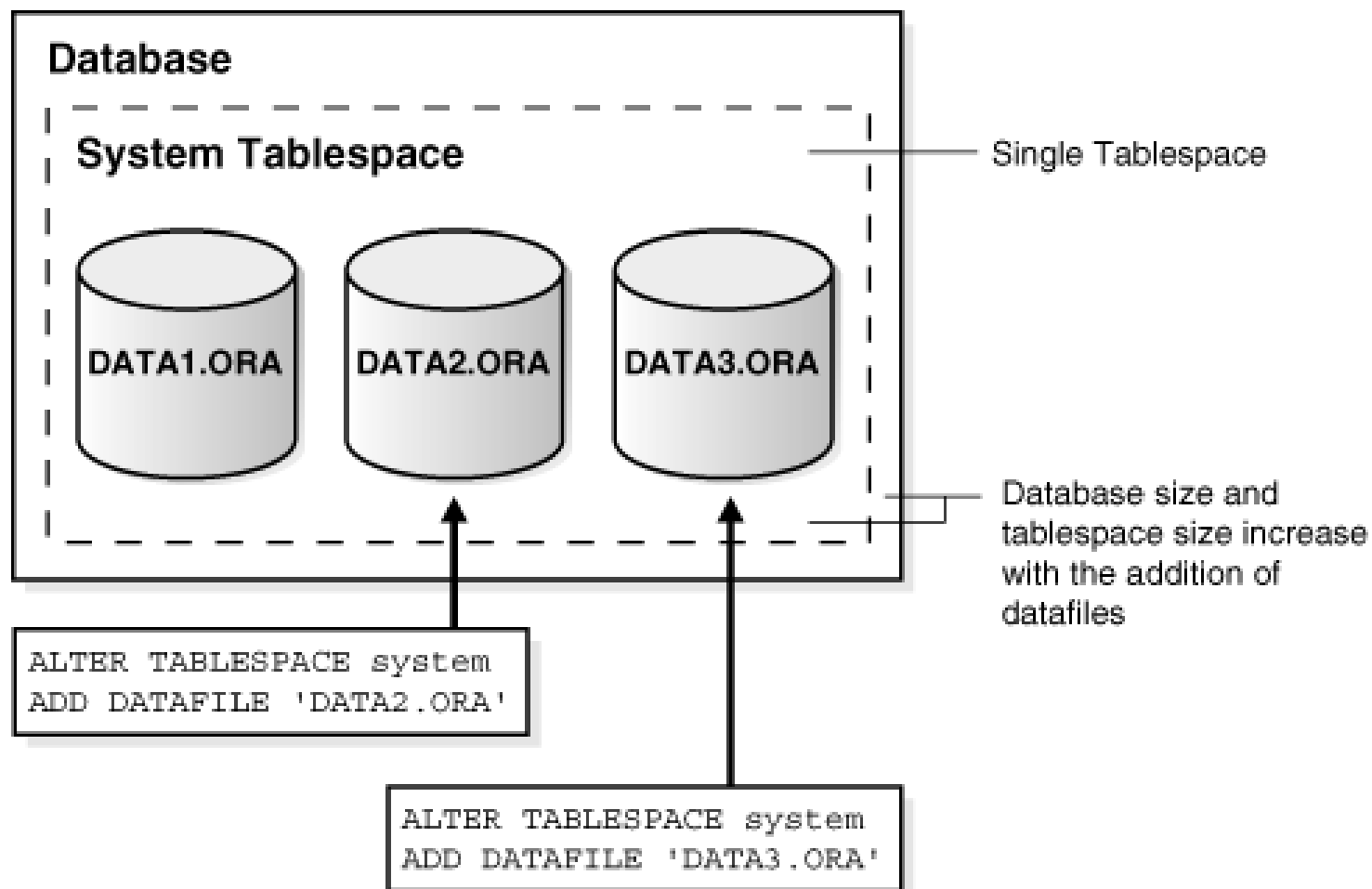
```
ALTER DATABASE DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET03.DBF'  
    AUTOEXTEND ON NEXT 1M MAXSIZE 4M;
```

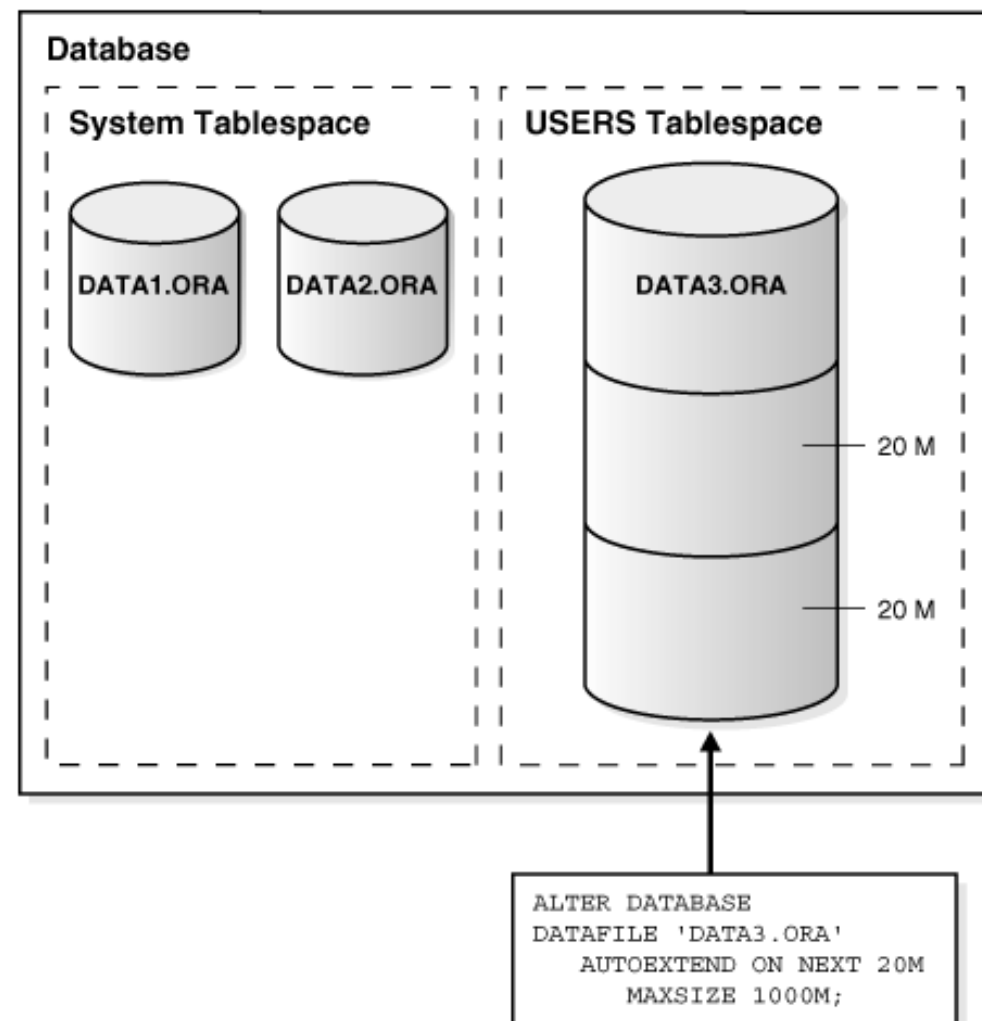
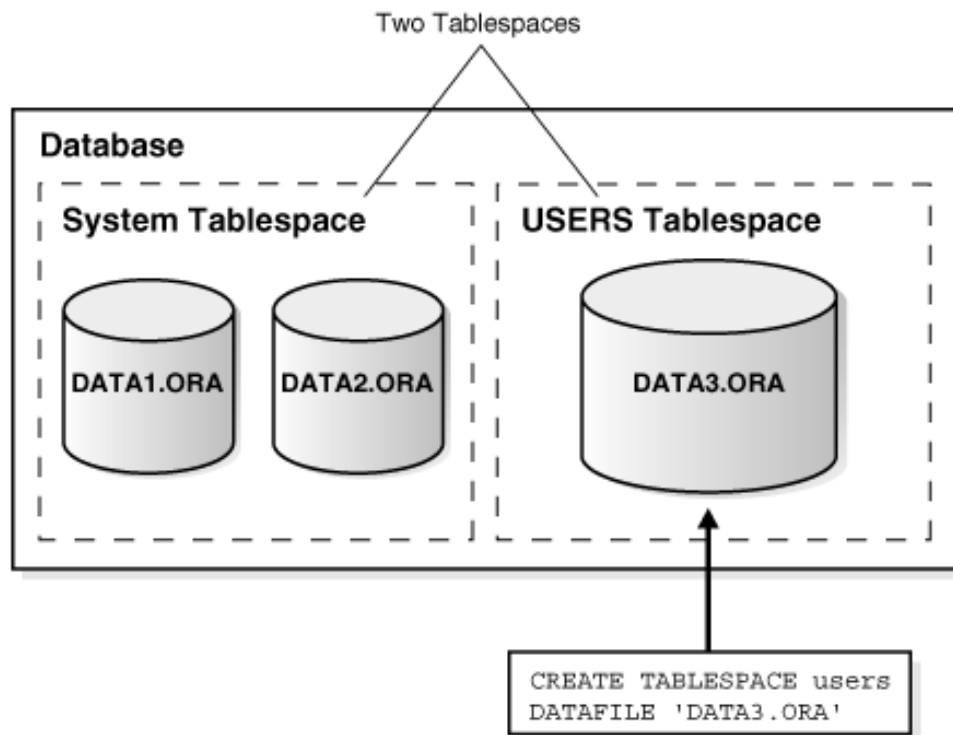
- ❑ Sólo un espacio de tablas BIGFILE se puede redimensionar con **ALTER TABLESPACE**, sin indicar DATAFILE ($\geq 10g$)

```
ALTER TABLESPACE bigtbs RESIZE 60G;
```

- ❑ A partir de la versión 10g se pueden borrar ficheros vacíos del espacio de tablas

```
ALTER TABLESPACE users DROP DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET03.DBF';
```



Mover y renombrar ficheros de datos asociados a un espacio de tablas

- ❑ Pasos para mover y renombrar los ficheros asociados a un espacio de tablas

(1) Poner el espacio de tablas *offline*

```
ALTER TABLESPACE ET01 OFFLINE;
```

(2) Mediante comandos del S.O., mover el fichero a su nueva ubicación y renombrarlo

(3) Hacer que Oracle conozca el nuevo nombre del fichero de datos asociado al espacio de tablas

```
ALTER TABLESPACE ET01  
RENAME DATAFILE 'C:\app\uhu\ORADATA\ONUBA\ET03.DBF' TO  
                'C:\app\uhu\ORADATA\ONUBA\ET0003.DBF';
```

(4) Poner el espacio de tablas online

```
ALTER TABLESPACE ET01 ONLINE;
```

Papelera de reciclaje (*Recycle Bin*)

- ❑ Cuando se borra una tabla en Oracle, no se elimina completamente ni se libera el espacio que ocupaba
- ❑ Se sitúa en la **Papelera de reciclaje** y, si ha sido borrada por error, se puede recuperar posteriormente mediante una operación denominada **Flashback Drop**
- ❑ Mientras que los objetos estén en la papelera, están consumiendo cuota de espacio de tablas. Los usuarios pueden gestionar su propia papelera
- ❑ Para consultar los objetos que se encuentran en la papelera:

```
SELECT * FROM RECYCLEBIN;
```

- ❑ Cuando se elimina un espacio de tablas incluyendo su contenido, los objetos no van a la papelera. Además, Oracle elimina de la papelera todos los objetos que pertenecieran a dicho espacio de tablas
- ❑ Cuando se elimina un usuario, los objetos de éste no se ponen en la papelera de reciclaje. Son eliminados directamente de la base de datos

- ❑ La papelera de reciclaje está activada por defecto. Sin embargo, podemos activarla o desactivarla mediante el parámetro **RECYCLEBIN**

```
ALTER SESSION SET recyclebin = ON | OFF;  
ALTER SYSTEM SET recyclebin = ON | OFF;
```

- ❑ Para eliminar los objetos definitivamente de la BD y liberar el espacio que ocupan se utiliza la sentencia **PURGE**

```
PURGE TABLE nombre_tabla;
```

- ❑ Se pueden eliminar todos los objetos que pertenecen únicamente a un espacio de tablas o los que pertenecen a un usuario en ese espacio de tablas

```
PURGE TABLESPACE ET_01;  
PURGE TABLESPACE ET_01 USER usu1;
```

- ❑ Los usuarios pueden vaciar totalmente sus papeleras de reciclaje

```
PURGE RECYCLEBIN;
```

- ❑ Si se tiene privilegio de administrador, se puede eliminar la papelera del DBA

```
PURGE DBA_RECYCLEBIN;
```

- ❑ Para borrar una tabla sin pasar por la papelera de reciclaje

```
DROP TABLE nombre_tabla PURGE;
```

- ❑ Para recuperar objetos desde la papelera de reciclaje se utiliza la sentencia **FLASHBACK TABLE ... TO BEFORE DROP**. La cláusula opcional **RENAME TO** se utiliza para renombrar la tabla recuperada

```
FLASHBACK TABLE nombre_tabla_01 TO BEFORE DROP RENAME TO nombre_tabla_02;
```

- ❑ Para recuperar objetos se puede utilizar también el nombre que se le asigna automáticamente en la papelera de reciclaje. Esto es de mucha utilidad cuando se ha eliminado una misma tabla varias veces
- ❑ **Ejemplo.** Tenemos 3 versiones de la tabla T1 en la papelera de reciclaje y queremos recuperar la segunda

```
SELECT object_name, original_name, createtime FROM recyclebin;
```

OBJECT_NAME	ORIGINAL_NAME	CREATETIME
-----	-----	-----
BIN\$yrMKlZaLMhfgNAgAIMenRA==\$0 T1		2006-02-05:21:05:52
BIN\$yrMKlZaVMhfgNAgAIMenRA==\$0 T1		2006-02-05:21:25:13
BIN\$yrMKlZaQMhfgNAgAIMenRA==\$0 T1		2006-02-05:22:05:53

```
FLASHBACK TABLE BIN$yrMKlZaVMhfgNAgAIMenRA==$0 TO BEFORE DROP;
```

Principales vistas del diccionario de datos

V\$TABLESPACE	Name and number of all tablespaces from the control file.
DBA_TABLESPACES, USER_TABLESPACES	Descriptions of all (or user accessible) tablespaces.
DBA_TABLESPACE_GROUPS	Displays the tablespace groups and the tablespaces that belong to them.
DBA_SEGMENTS, USER_SEGMENTS	Information about segments within all (or user accessible) tablespaces.
DBA_EXTENTS, USER_EXTENTS	Information about data extents within all (or user accessible) tablespaces.
DBA_FREE_SPACE, USER_FREE_SPACE	Information about free extents within all (or user accessible) tablespaces.
V\$DATAFILE	Information about all datafiles, including tablespace number of owning tablespace.
V\$TEMPFILE	Information about all tempfiles, including tablespace number of owning tablespace.
DBA_DATA_FILES	Shows files (datafiles) belonging to tablespaces.
DBA_TEMP_FILES	Shows files (tempfiles) belonging to temporary tablespaces.
V\$TEMP_EXTENT_MAP	Information for all extents in all locally managed temporary tablespaces.
DBA_TABLESPACE_GROUPS	Information about groups of tablespaces.

Principales vistas del diccionario de datos

V\$TEMP_EXTENT_POOL	For locally managed temporary tablespaces: the state of temporary space cached and used for by each instance.
V\$TEMP_SPACE_HEADER	Shows space used/free for each tempfile.
DBA_USERS	Default and temporary tablespaces for all users.
DBA_TS_QUOTAS	Lists tablespace quotas for all users.
V\$SORT_SEGMENT	Information about every sort segment in a given instance. The view is only updated when the tablespace is of the TEMPORARY type.
V\$TEMPSEG_USAGE	Describes temporary (sort) segment usage by user for temporary or permanent tablespaces.
USER_TABLES	Vista de tablas de usuarios