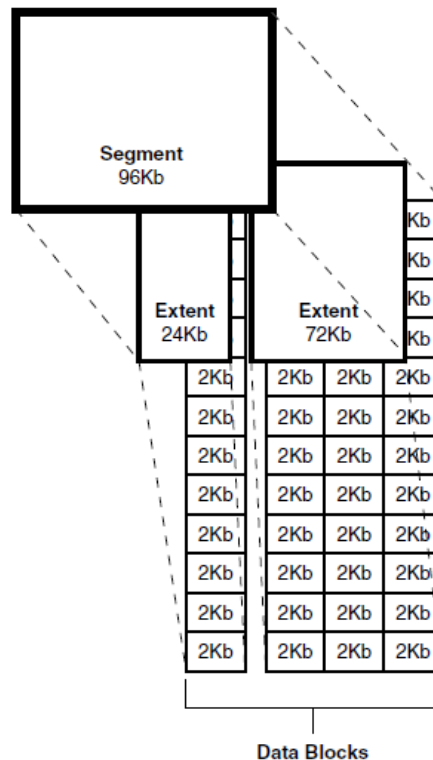


Tema 1

Segmentos,
extensiones y
bloques de datos.

Segmentos, extensiones y bloques

- Oracle almacena la información en unidades lógicas que son los segmentos, las extensiones y los bloques.
- Estas tres unidades están relacionadas entre sí:
 - Un segmento está formado por una o varias extensiones y
 - Cada extensión está formada por varios bloques.



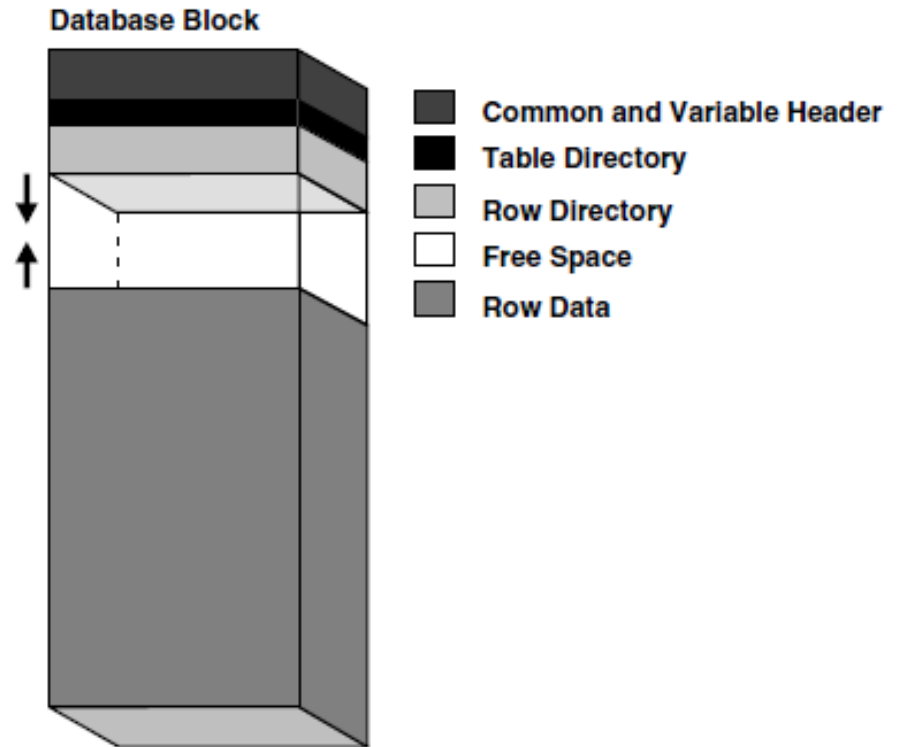
Bloques de datos

- Un bloque es la unidad mínima de almacenamiento de información de Oracle.
 - También se conocen como «bloques lógicos» o «bloques Oracle».
- Así mismo, un bloque de datos es la unidad mínima de I/O en una base de datos Oracle;
 - Oracle no lee y escribe en bloques del SO, sino que lo hace en unidades lógicas que son los bloques de datos.
- Al crear la Base de Datos Oracle se debe establecer el tamaño del Bloque de Datos (parámetro DB_BLOCK_SIZE),
 - procurando que sea un múltiplo de la medida del bloque del SO, dentro de un límite máximo, para evitar I/O innecesarios.

Estructura de un bloque

Los bloques de la base de datos pueden contener información de tablas, índices o segmentos de rollback, pero no importa qué información contengan, siempre tienen la misma estructura:

- Cabecera
- Directorio de tablas
- Directorio de filas
- Espacio libre
- Filas de datos



Estructura de un bloque

Cabecera:

Contiene información general sobre el bloque, como el tipo de segmento al que pertenece (índice, tabla, rollback) o la dirección del bloque.

Directorio de Tablas:

Contiene información acerca de las tablas que tienen datos en el bloque.

Directorio de Filas:

Indica las filas que se encuentran en cada momento en el bloque.

Espacio Libre:

Esta zona está reservada para la inserción de nuevas filas en el bloque, o para la modificación de campos que requieren más espacio que el que tenían con anterioridad.

Filas de datos:

Se almacenan los datos de las tablas o de los índices.

Overhead (cabecera + directorio de tablas + directorio de filas):

Espacio del bloque que realmente no se rellena con datos, sino que está ocupado por la información que necesita Oracle para saber exactamente qué datos tiene en dicho bloque.

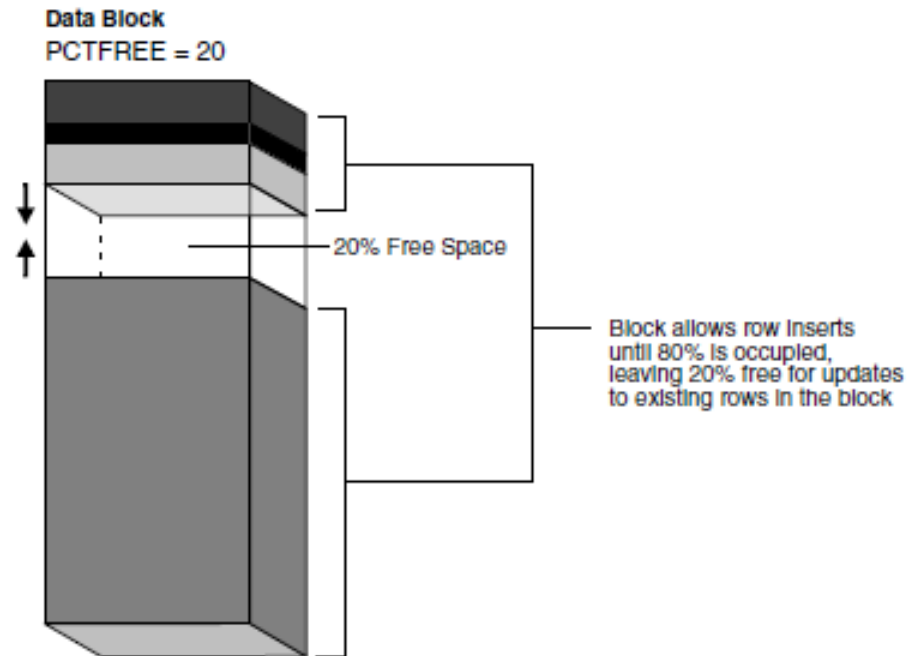
PCTFREE y PCTUSED

Los parámetros PCTFREE Y PCTUSED permiten controlar el espacio libre para inserciones y actualizaciones de filas en los bloques de un segmento.

Estos parámetros se especifican cuando se crea o altera una tabla o clúster; para el caso de los índices sólo tiene efecto PCTFREE.

PCTFREE:

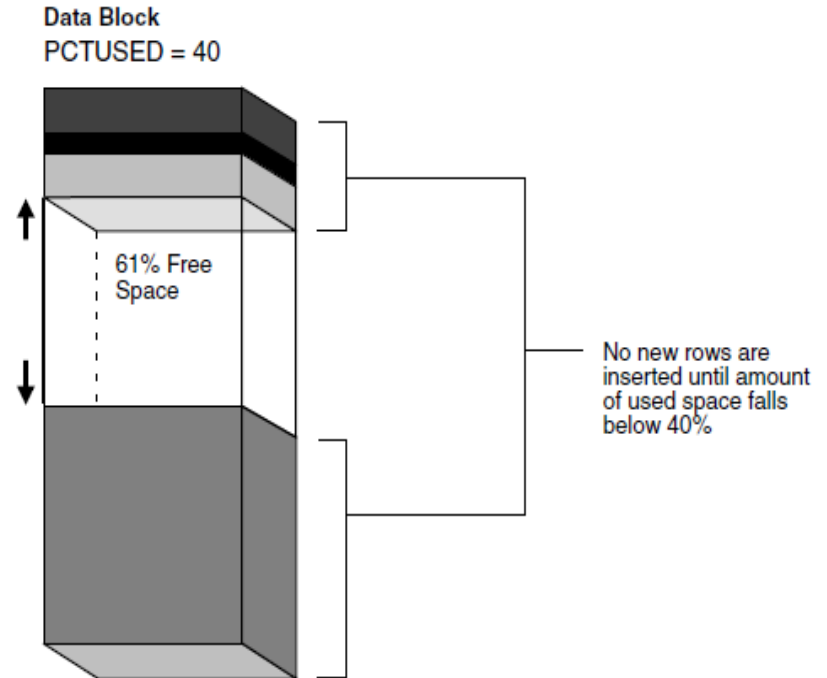
Indica el porcentaje mínimo que se debe dejar libre para posteriores modificaciones de los datos de las filas que ya existen dentro del bloque.



PCTFREE y PCTUSED

PCTUSED:

Indica a partir de qué porcentaje de espacio libre puede volver a emplearse un bloque para insertar nuevos datos de filas.



Un bloque está disponible para inserción mientras tenga libre el porcentaje que indica PCTFREE.

Cuando éste es menor, se marcará como no disponible para inserciones, hasta que el porcentaje de uso descienda por debajo del parámetro PCTUSED.

PCTFREE y PCTUSED

Para consultar el valor tanto del parámetro PCTFREE como del parámetro PCTUSED de cada segmento (de tipo tabla o de tipo índice), podemos leer las vistas dba_tables y dba_indexes.

```
select owner, table_name, pct_free, pct_used from dba_tables;
```

```
select owner, index_name, pct_free from dba_indexes;
```

Para modificar el valor de los parámetros de una tabla o de un índice se pueden utilizar las siguientes sentencias:

```
alter table nombre_de_tabla pctfree nuevo_pct_free;
```

```
alter table nombre_de_tabla pctused nuevo_pct_used;
```

```
alter index nombre_de_indice pctfree nuevo_pct_free;
```


Encadenamiento y Migración de Filas

Existen dos circunstancias diferentes en las cuales ocurre que los datos de una fila no tienen espacio suficiente dentro del bloque.

- En el momento de la inserción de la fila, ya se excede el tamaño del bloque.

- Se modifica la información almacenada y se excede el espacio disponible.

Hay que intentar por todos los medios evitar que estas circunstancias se den para que no disminuya el rendimiento del sistema.

Cuando se produce una de estas circunstancias, los datos de una fila se dispersan por varios bloques, con lo que, para obtener esos datos o para modificarlos, Oracle debe recorrer varios bloques que, posiblemente, no estén contiguos.

Encadenamiento y Migración de Filas

Oracle establece dos estrategias diferentes según la situación:

- Encadenamiento de filas.
- Migración de filas.

Encadenamiento de filas:

Se da cuando los datos de una fila ocupan tanto espacio que no caben físicamente en un solo bloque y Oracle debe guardarlos en dos o más bloques de los reservados para ese segmento.

Encadenamiento y Migración de Filas

Migración de filas:

Se da cuando modificamos los datos de una fila de un bloque, aumentándolos de tamaño.

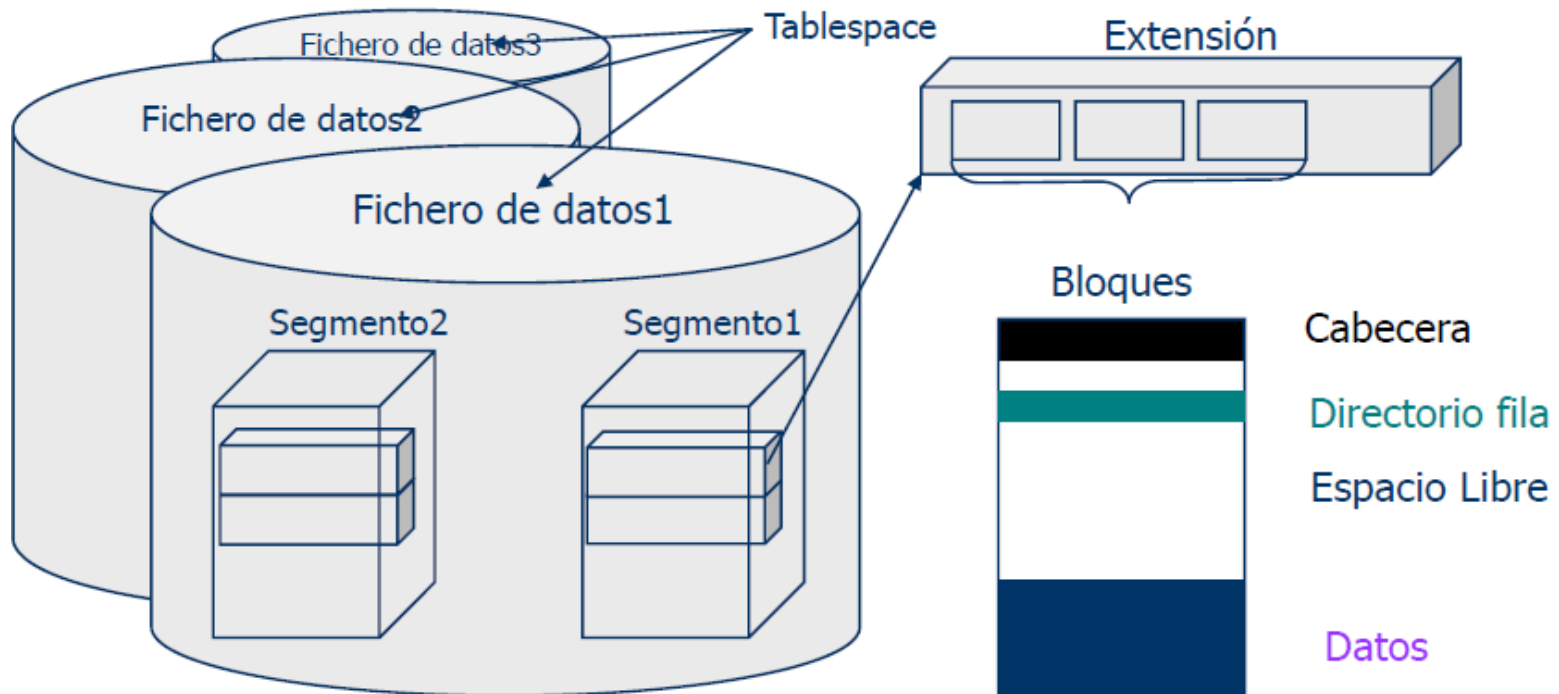
Si en la zona en el bloque no disponemos de espacio suficiente, Oracle migra toda la fila a un nuevo bloque en el que si haya espacio para toda la fila.

Para no tener que cambiar a dicha fila el rowid, lo que se hace es dejar en el bloque inicial una información mínima de la fila, que será simplemente un puntero hacia la dirección del nuevo bloque en el que se ha reubicado toda esta fila.

Extensiones

Una extensión es una unidad lógica de almacenamiento que está formada por un número determinado de bloques de datos contiguos.

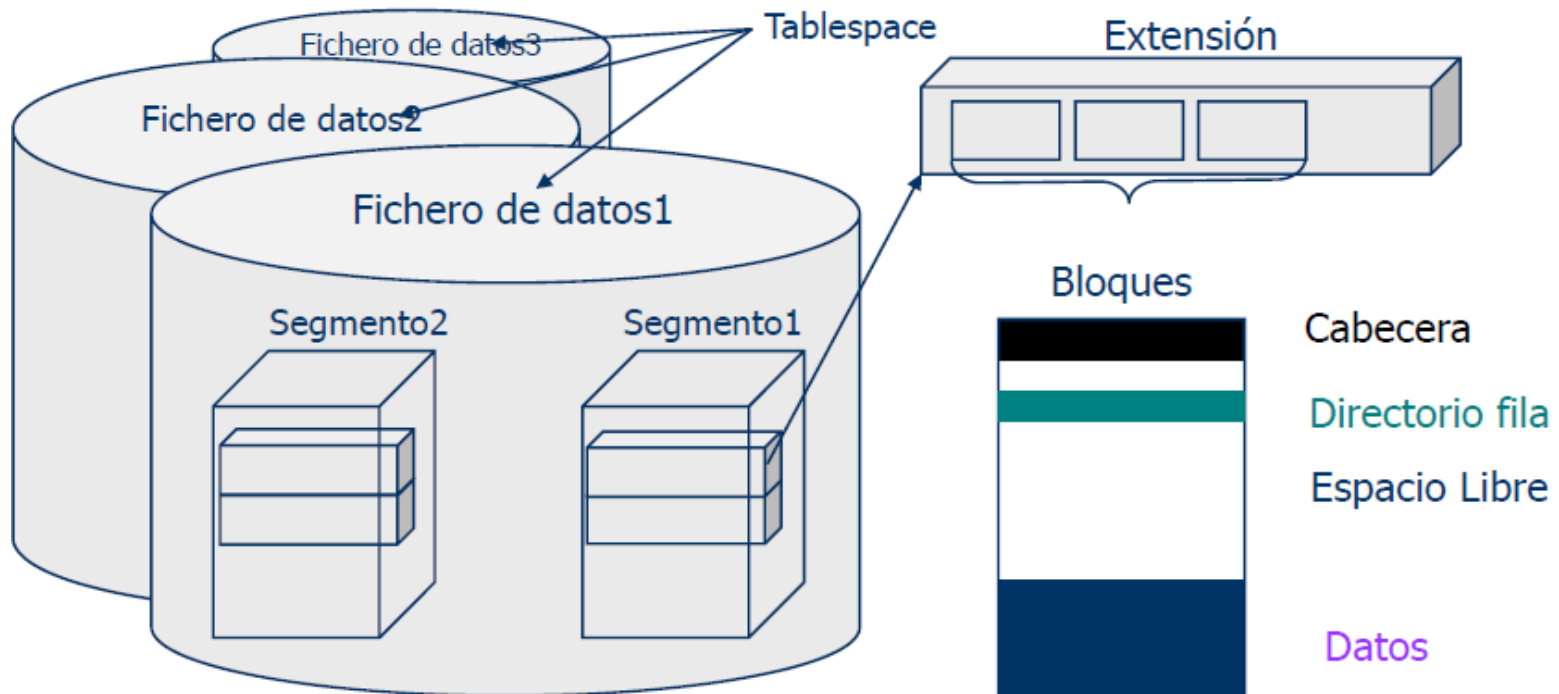
La agrupación de una o varias extensiones forman un segmento que puede ser una tabla, un índice, un segmento de rollback o un segmento temporal.



Extensiones

Cuando se crea un segmento nuevo, se crea obligatoriamente una extensión en dicho segmento (en el caso de los rollback se crean dos).

El tamaño inicial de esta extensión viene dado por el valor del parámetro «initial» que se indica en el momento de crear el segmento.



Asignación de extensiones

Al asignar una nueva extensión al segmento, se reserva espacio en el disco para almacenar los nuevos datos de dicho segmento.

Al crear una nueva extensión está totalmente vacía y todo su espacio está disponible para almacenar los datos del segmento.

Además en el disco debe haber espacio libre para que Oracle reserve todo el tamaño que necesita la extensión, y lo formatee de forma especial para poder utilizarlo.

En la extensión creada solamente se podrán almacenar datos del segmento al que pertenece.

Cuando se llenan todos los bloques de datos de una extensión, el segmento solicita una nueva extensión al sistema para poder seguir almacenando información.

Para establecer el tamaño de las futuras extensiones que irá solicitando un segmento se utilizan varios parámetros a los que se da valor en el momento de creación de la tabla, índice o segmento de rollback.

Liberación de extensiones

Las extensiones que han sido reservadas por un segmento no son devueltas a Oracle, para que se las pueda asignar a otro segmento del mismo tablespace, hasta que se elimina el objeto mediante la instrucción DROP.

Por lo tanto, cuando tenemos una tabla que nos ocupa varias extensiones, a pesar de que borremos todas sus filas, esa tabla seguirá teniendo reservadas las extensiones aunque, eso sí, estarán vacías.

Excepciones:

- Se pueden devolver todas las extensiones de una tabla, excepto las indicadas por el parámetro MIN_EXTENTS, haciendo un truncate de la misma.
- También se pueden liberar las extensiones que no se usan en una tabla mediante la sentencia:

```
alter table nombre_de_tabla deallocate unused;
```

Liberación de extensiones

En cuanto se libera una extensión del segmento al que pertenecía, Oracle ya la puede volver a emplear para que la puedan utilizar otros segmentos del mismo tablespace.

Por lo tanto:

1. cuando un nuevo objeto solicite una nueva extensión,
2. si Oracle no encuentra espacio libre suficiente para crear una nueva, y entre las que ha ido desasignando tampoco encuentra ninguna suficientemente grande como para asignarla,
3. puede unir varias de estas extensiones hasta conseguir una lo suficientemente grande como para poder asignársela al segmento.

Segmentos

- Conjunto de extensiones dedicadas a un objeto de la BD y almacenado en un fichero.
- Hay cuatro tipos de segmentos:
 - de datos
 - de índices
 - temporales
 - de rollback (o Undo)
- La cantidad de espacio que utiliza está determinada por sus parámetros de almacenamiento:
 - al crear el objeto (tabla, índice, clúster, Undo segment)
 - utiliza los parámetros de almacenamiento predeterminados del tablespace en el que se almacena (se permite posterior modificación)
- Oracle asigna a cada tabla una o más extensiones para formar el segmento de datos.

Segmentos

Cuando las extensiones de un segmento ya no pueden contener más datos el segmento se amplía con nuevas extensiones.

Así sucesivamente hasta que:

- no haya más espacio disponible en los ficheros de datos de los Tablespaces (no ampliables automáticamente).
- hasta alcanzar el valor del parámetro MAXEXTENTS por segmento (si está definido).
- no quede más espacio para asignar a ese usuario (cuota) en el tablespace.

Segmentos de datos e índices

- Segmentos de datos:
 - En un segmento de datos se almacenan todos los datos de una tabla que no esté particionada o que no forme parte de un clúster.
- En el caso de los índices:
 - existe un segmento para cada índice no particionado o para cada partición de un índice particionado.
- Se crea el segmento de datos o índice a la hora de ejecutar la sentencia CREATE que crea la tabla, clúster, partición o índice.
- En dicha sentencia se pueden indicar también los valores de la cláusula STORAGE para la tabla, clúster, partición o índice que se está creando.

Segmentos de Rollback (Undo)

- Incluyen información del bloque y los datos como estaban antes de la operación en una transacción.
- Estas transacciones pueden ser definitivas, es decir, se ha realizado ya el commit de ellas, o puede que aún no se haya hecho dicho commit.
- Sólo Oracle puede acceder a los segmentos de Rollback, ni los usuarios ni el DBA pueden acceder a ellos.

Segmentos temporales

- Cuando Oracle procesa las consultas se puede ver en la necesidad de utilizar espacio en disco para poder llevar a cabo la ejecución de las mismas.
- Solamente utilizará este tipo de segmentos cuando no pueda realizar la consulta íntegramente en memoria o cuando no pueda buscarse un método alternativo para realizarla utilizando los índices.
- Hay varios tipos de sentencias en las que Oracle se puede ver en la necesidad de utilizar los segmentos temporales:
 - SELECT ... ORDER BY.
 - CREATE INDEX.
 - SELECT ... GROUP BY.
 - SELECT ... UNION ...
 - SELECT DISTINCT ...
 - SELECT MINUS ...

Tipos de Segmentos

Al crear un segmento de datos:

- Al menos se asigna una extensión.
- Su espacio no se libera hasta que se elimina.
- Se puede utilizar “alter table” para modificar el *STORAGE*. Y el comando “move” para cambiarlo de tablespace.

Al crear un segmento de índice:

- Al menos se asigna una extensión.
- Su espacio no se libera hasta que se elimina, pero éstos se pueden eliminar automáticamente al eliminar las TABLAS o los CLÚSTERS a los que indexan.
- Se aconseja guardar los índices y las tablas en diferentes Tablespaces, para evitar interferencias.
- Se puede utilizar la opción “REBUILD” del comando “alter index” para modificar la configuración del *STORAGE* y del tablespace de un índice.

Tipos de Segmentos

Al crear un segmento de rollback:

- Sus extensiones mantendrán copias temporales de los bloques de datos que cambiaron durante alguna transacción.
- Al menos se asignan DOS extensiones.
- Todas las extensiones tienen el mismo tamaño.
- Se podrá reducir el tamaño del segmento dinámica o manualmente hasta un tamaño específico (cláusula OPTIMAL).

Al crear un segmento temporal:

- Almacenan datos temporales durante las operaciones (p.e.: consultas de gran tamaño, creación de índices, operaciones de unión, etc.)
- Cada usuario tiene su tablespace temporal (se estableció al ejecutar create user).
- Habitualmente se crea un tablespace temporal para cada BD y todos sus usuarios.

Información sobre Segmentos

La tabla que guarda la información de los segmentos de usuario es `user_segments`, y se puede visualizar su contenido con la sentencia SQL siguiente:

```
select * from user_segments;
```


Parámetros de los Segmentos

Cada segmento tiene un conjunto de parámetros de almacenamiento que controla su crecimiento:

- **initial**: tamaño de la extensión inicial.
- **next**: tamaño de la siguiente extensión a asignar.
- **minextents**: número de extensiones asignadas en el momento de la creación del segmento.
 - Por defecto es una, excepto en los segmentos de rollback que son dos.
 - Puede que las extensiones no estén contiguas.
- **maxextents**: número máximo de extensiones.
 - Se puede especificar UNLIMITED, y así pueden crecer indefinidamente.
- **pctincrease**: Porcentaje en el que crecerá la siguiente extensión antes de que se asigne, en relación con la última extensión utilizada.
- **pctfree**: porcentaje de espacio libre para actualizaciones de filas que se reserva dentro de cada bloque asignado al segmento.
- **pctused**: porcentaje de utilización del bloque por debajo del cual Oracle considera que un bloque puede ser utilizado para insertar nuevas filas en él.
- **tablespace**: nombre del tablespace donde se creará el segmento.

Ejemplo creación de Segmento

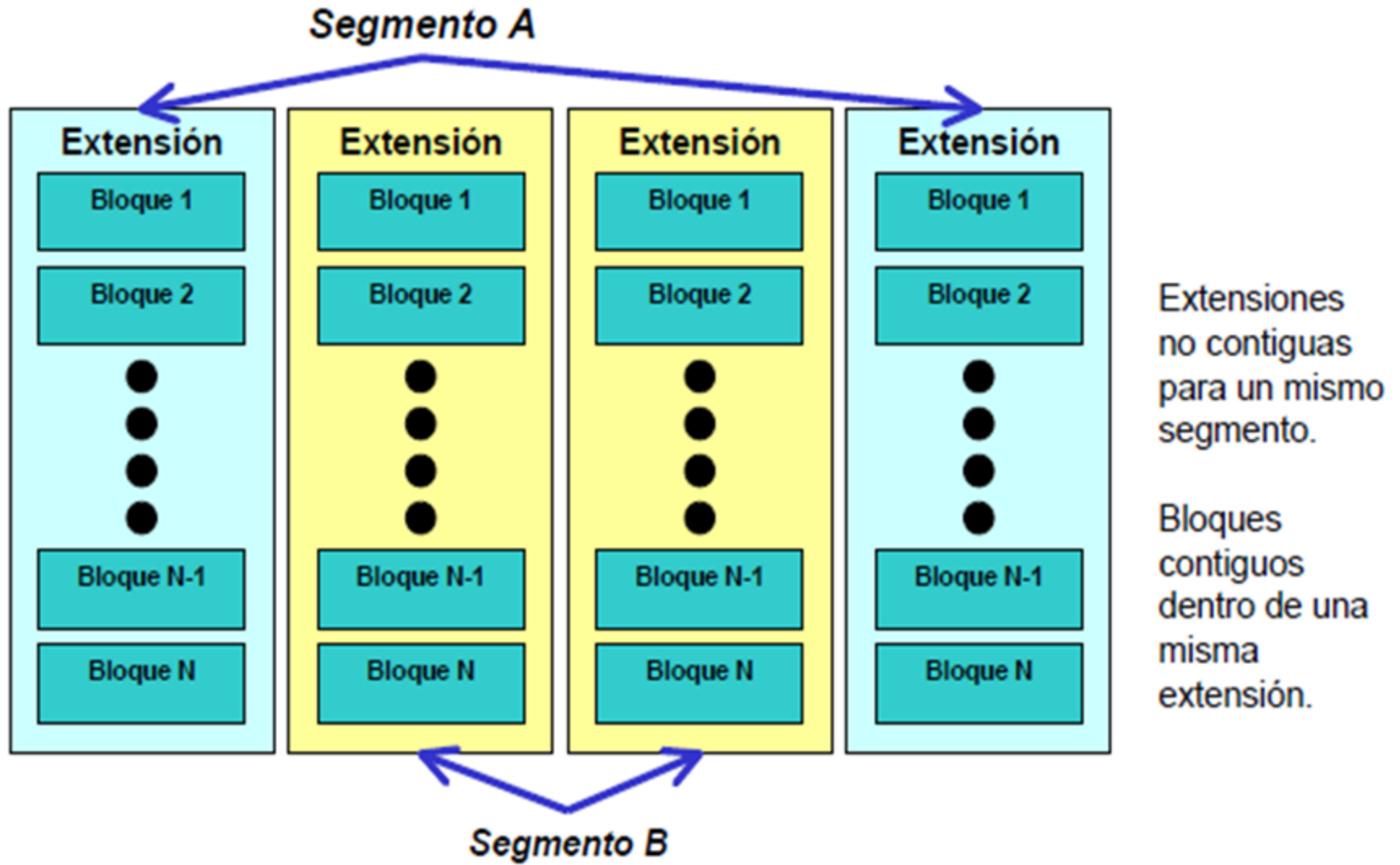
A continuación se ilustra la creación de una tabla en la que se indican valores para los parámetros de la cláusula STORAGE.

Creamos una tabla llamada empleado que contiene:

- Un solo campo: nombre
- Initial: 256 Kilobytes
- Next: 512 Kilobytes
- Pctincrease: 50
- 3 extensiones iniciales y un máximo de 10 extensiones.

```
create table empleado (nombre varchar2(50))  
storage (initial 256K next 512K pctincrease 50 minextents 3  
maxextents 10);
```

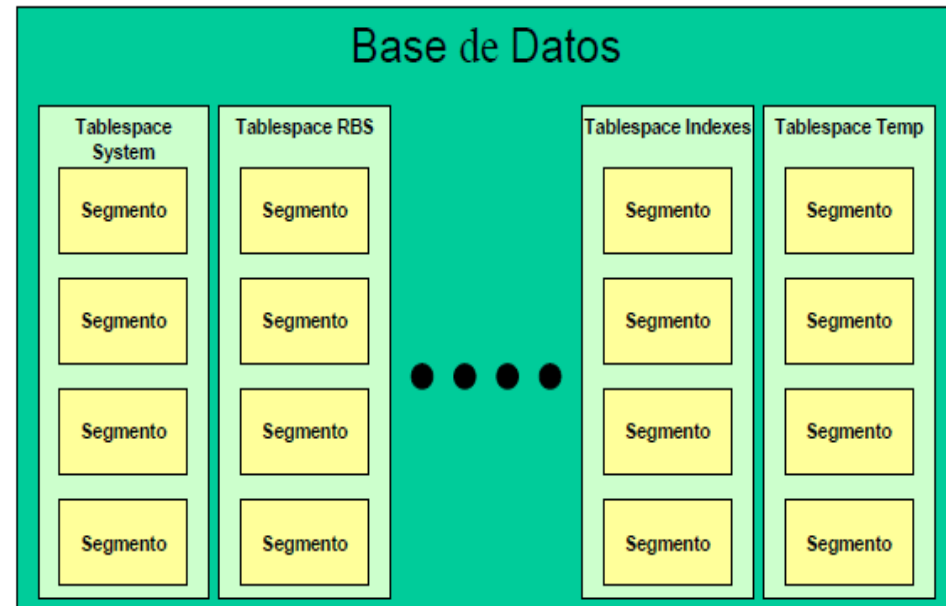
En resumen ...



Tablespaces, ficheros de datos y ficheros de control

Implementación básica de una BD Oracle

- Uno o más ficheros de datos
- Uno o más ficheros de control
- Internamente la base de datos contiene
 - Múltiples esquemas de usuario
 - Dos o más segmentos undo ó de Rollback
 - Varios tablespaces
 - Tablas del Diccionario de datos
 - Objetos de usuario (tablas, índices, etc)



Ficheros controlados por Oracle (OMF)

- Facilitan la labor del DBA automatizando la creación y eliminación de los archivos de datos que forman las estructuras lógicas de la base de datos.
- Sin OMF, un DBA podría eliminar un tablespace y olvidarse de borrar los archivos subyacentes del sistema operativo.
 - Esto provocaría un uso poco eficiente del disco e incrementaría innecesariamente el tiempo de realización de las copias de seguridad.

Asignar más espacio a una base de datos

- El tamaño de un tablespace se corresponde con el tamaño de los ficheros de datos que lo constituyen.
- El tamaño de una base de datos es el tamaño colectivo de los tablespaces que la constituyen.
- Se puede ampliar una base de datos de tres maneras:
 - Añadiendo un fichero de datos a un tablespace.
 - Añadiendo un nuevo tablespace.
 - Aumentando el tamaño de un fichero de datos.

Tablespace

- Permite organizar de forma lógica las tablas dentro de la base de datos.
- Una tabla está contenida dentro de un solo tablespace pero puede residir físicamente en uno o más ficheros del mismo tablespace.
- Si se utilizan varios ficheros de datos, se pueden localizar en discos diferentes, lo que aumentará el rendimiento del sistema.
- Cualquier objeto de una base de datos debe estar almacenado obligatoriamente dentro de un tablespace.

Tablespace

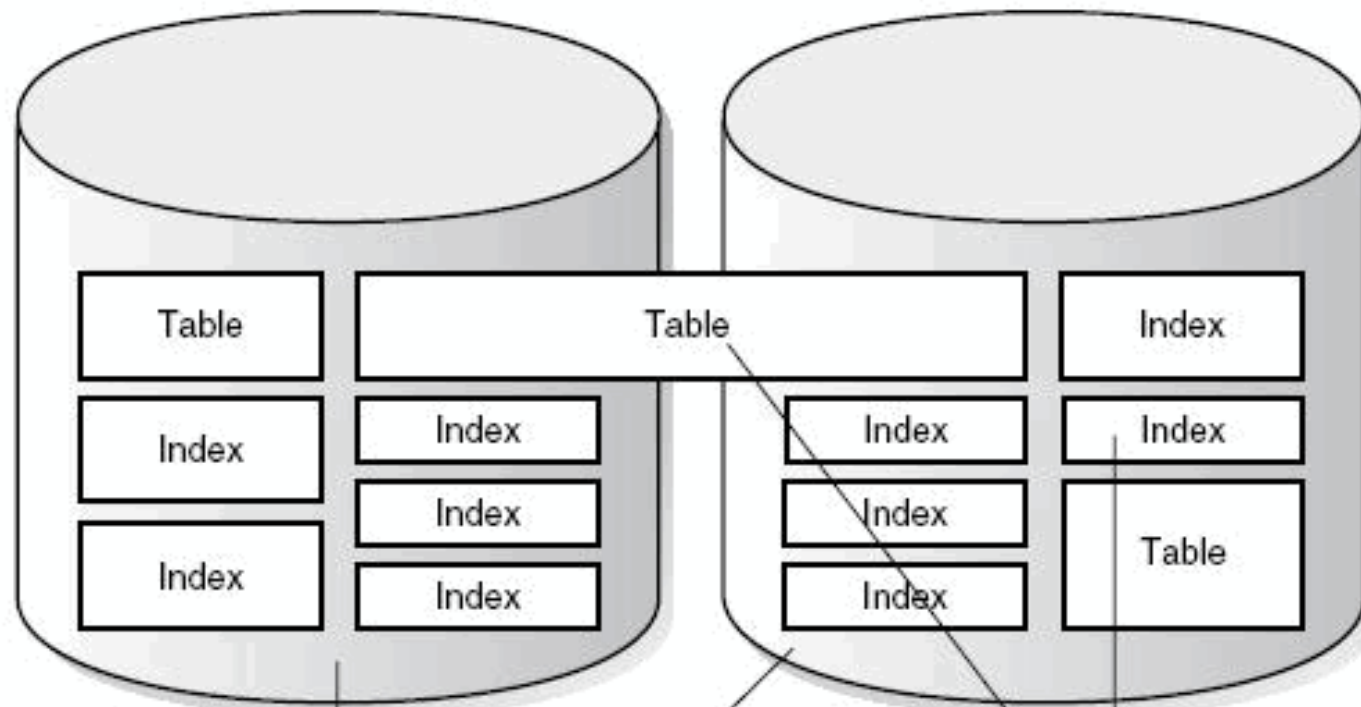
- Se pueden ver los tablespaces definidos en nuestra BD con el siguiente comando:

```
select * from user_tablespace;
```

- Propiedades de un tablespace:
 - Especificación de cuotas máximas de consumo de disco.
 - Localización de los ficheros de datos.
 - Control de la disponibilidad de los datos (online ó offline).
 - Backup de los datos.
- Cuando un objeto se crea en un tablespace, este objeto adquiere todas las propiedades del tablespace utilizado.

Tablespace

(one or more datafiles)



Datafiles

(physical structures associated with only one tablespace)

Objects

(stored in tablespaces- may span several datafiles)

Creación de un tablespace

- Cuando se crea un tablespace se le asigna espacio en disco que Oracle reserva inmediatamente, se utilice o no.
- Si este espacio inicial se queda pequeño Oracle puede gestionar el crecimiento dinámico de los ficheros.
 - Se elimina así la posibilidad de error en las aplicaciones por fallos de dimensionamiento inicial.
- Los parámetros de crecimiento de los tablespaces se especifican en la creación de los mismos. (INITIAL, NEXT).

Manipulación de tablespaces

- Ilustramos la creación de un tablespace llamado “prueba”, formado físicamente por un fichero llamado prueba01.dbf de 100 Mbytes.

```
create tablespace prueba datafile  
    '/users/oradata/orcl/prueba01.dbf' size 100M;
```

Manipulación de tablespaces

- Para aumentar el tamaño de un tablespace:
 - Crear un nuevo fichero de datos y asignarlo al tablespace:
 - `alter tablespace prueba add datafile`
`'/users/oradata/orcl/prueba02.dbf' size 50M;`
 - Ampliar el tamaño de uno de los ficheros existentes:
 - `alter datafile '/users/oradata/orcl/prueba01.dbf' resize 150M;`

Manipulación de tablespaces

- Borrar un tablespace.
`drop tablespace prueba;`
- Tanto en la creación como en el aumento de tamaño se puede observar que un fichero de datos pertenece solamente a un tablespace, ya que en las propias sentencias se crea el fichero físico.

Gestión del espacio

- El primer tablespace creado es el SYSTEM y debe estar disponible siempre durante el funcionamiento de la BD porque contiene el diccionario de datos.
- Se recomienda la creación de otros tablespaces para separar los datos de los usuarios de los del diccionario de datos.
- De igual manera, si se van a ejecutar varias aplicaciones sobre la misma BD es recomendable que sus datos estén separados.

Diferentes clasificaciones de los tablespaces

- Creados por defecto: SYSTEM, SYSAUX
- Ficheros en los que se implementan:
 - Muchos ficheros pequeños (predeterminado)
 - Un fichero muy grande (Maxifichero)
- Tablespaces de solo lectura
- Tablespaces temporales
- Tablespaces gestionados localmente
- Tablespaces offline y online
- Tablespaces Undo

Tablespaces con maxiarchivos

- En ellos se asigna un único fichero de datos, que puede tener un tamaño de hasta 8 ExaBytes.
- Por defecto, el sistema crea Tablespaces haciendo uso de ficheros pequeños.
- Una BD Oracle puede contener ambos tipos de fichero, ya que son totalmente transparentes en términos de la ejecución de instrucciones SQL.

Tablespace SYSTEM

- Se necesita un tablespace inicial para crear los usuarios SYSTEM y SYS.
- En él residen todos los objetos de los usuarios SYS y SYSTEM.
- Los segmentos de usuario no deben residir nunca en el Tablespace SYSTEM, bajo ningún concepto.
- La utilización del espacio es gestionada mediante un segmento de mapa de bits, situado al principio del primer archivo de datos del tablespace.
- Se encarga de almacenar:
 - El diccionario de datos.
 - Códigos PL/SQL fuentes y compilados.

Tablespace SYSAUX

- Es un espacio auxiliar para el tablespace SYSTEM.
- No debe contener ningún segmento de usuario.
- Proporciona información sobre los metadatos.
- No se puede detener o renombrar mientras la base de datos esté en funcionamiento.

Tablespace Undo

- Se utiliza exclusivamente para el almacenamiento de información de Undo.
- Una base de datos puede no tener, o tener muchos, Tablespaces Undo.
- En el modo de gestión automático, a cada instancia de Oracle se le asigna un único tablespace Undo.
- Cada tablespace Undo lleva asociados ficheros Undo, que se gestionan a nivel local.

Creación de un tablespace de undo

- Ventajas:
 - Los segmentos de deshacer son gestionados automáticamente por Oracle en el tablespace undo.
 - Los Tablespaces de undo pueden proporcionar un mecanismo para que el usuario pueda recuperar filas de una tabla.
- Inconvenientes:
 - Si se asigna un tamaño demasiado pequeño, la ventana de disponibilidad para las consultas flashback será muy pequeña.
 - Si se le asigna un espacio demasiado grande, se desperdiciará espacio de disco y las operaciones de copia de seguridad pueden requerir más tiempo del necesario.

Tablespaces temporales

- Son aquellos en los que sólo puede haber objetos temporales.
- Están especialmente preparados para optimizar las operaciones de ordenación.
- Los rendimientos son muy superiores comparados con los Tablespaces normales porque el mecanismo para reservar y liberar el espacio es muy distinto.
- Para crear un tablespace temporal:

```
create tablespace prueba datafile  
'/users/oradata/orcl/prueba01.dbf' size 100M  
temporary;
```

Tablespaces gestionados localmente

- Este tipo de Tablespaces controlan el espacio libre mediante mapa de bits.
- Los mapas de bits son muy eficientes porque son compactos, comparados con las listas de bloques libres.
- Se almacenan dentro del propio tablespace, por lo que se disminuye el uso de tablespace SYSTEM.

Tablespace Online y Offline

- Online: está disponible para operar en él.
- Offline: no se puede utilizar.
- Por defecto, los Tablespaces se crean online.
- Podemos saber qué Tablespaces tenemos en nuestra BD y en qué estado se encuentran con:

```
select tablespace_name, status from dba_tablespace;
```

- Es interesante tener un tablespace offline:
 - Copias de seguridad, con certeza de que nadie esté modificando los datos.
 - Actualizar una aplicación que se basa en los objetos de este tablespace, sin que ningún usuario pueda modificar los datos en medio de la actualización.

Tablespaces de sólo lectura

- Existe la posibilidad de poner un tablespace en un estado en el cual, solamente, se pueden consultar los datos de los objetos, no se puede ni borrar ni insertar nada en ellos.
- Su principal ventaja es que no se necesita hacer un backup del mismo.
- Para poner un tablespace read only:

`alter tablespace nombre_de_tablespace read only;`

- Si necesitamos modificar los datos que se encuentran almacenados en un tablespace read only, simplemente debemos ponerlo en primer lugar, en estado read write y, una vez realizada la modificación, volver a ponerlo en su estado read only.

Tablespaces de sólo lectura

- Solamente se puede consultar los datos de los objetos, no se puede ni borrar ni insertar nada en ellos.
- Para poner un tablespace read only:

`alter tablespace nombre_de_tablespace read only;`

- Como no se pueden modificar los datos que se encuentran en estos espacios, no hace falta hacer backup de los datos.
- Si se necesita modificar los datos almacenados en un tablespace de sólo lectura, deberemos ponerlo en estado lectura/escritura y después devolverlo a su estado original.
- Al igual que en los Tablespaces offline, sí se pueden borrar objetos enteros del tablespace, como por ejemplo un índice o una tabla.

Tablespaces de sólo lectura

- Una curiosidad sobre los tablespaces que están offline
 - No se pueden realizar consultas ni modificaciones ni inserciones en los datos de las tablas que están en ellos
 - Pero sí que se pueden eliminar objetos de dicho tablespace, que no es lo mismo que borrar datos de objetos del tablespace.
- Es habitual que en las bases de datos se creen tablespaces para almacenar los índices de la aplicación y otros distintos para almacenar las tablas o datos.
 - En estos casos, si desactivamos el tablespace en el que se encuentran los índices, podemos seguir accediendo a las tablas y realizar consultas sobre ellas porque su tablespace está accesible.

Manipulación de los tablespaces

- Sentencias:
 - poner un tablespace offline
`ALTER TABLESPACE app_data OFFLINE;`
 - poner un tablespace online
`ALTER TABLESPACE app_data ONLINE;`
 - poner un tablespace en modo read-only
`ALTER TABLESPACE app_data READ ONLY;`
 - poner un tablespace en modo read-write
`ALTER TABLESPACE app_data READ WRITE;`

Borrando tablespaces

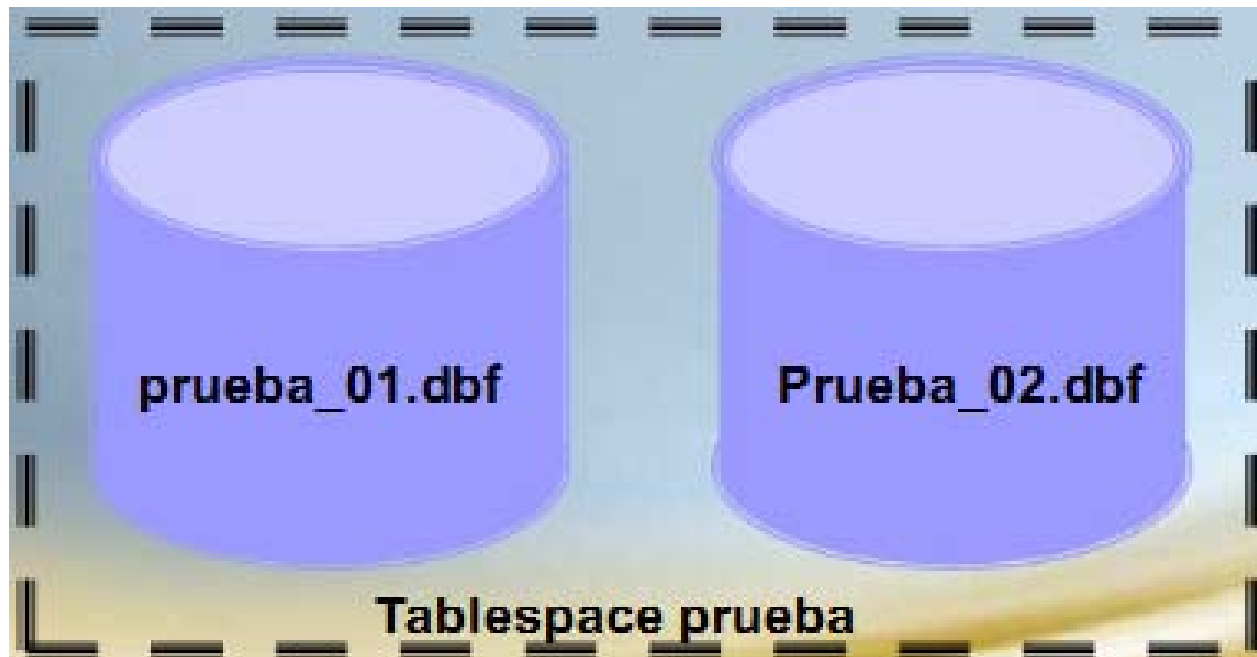
- Sintaxis

```
DROP TABLESPACE name [INCLUDING CONTENTS  
[CASCADE CONSTRAINTS]];
```

- Si contiene datos tiene que ser borrado con INCLUDING...
- Se borran los punteros, pero el fichero existe y hay que borrarlo desde el SO.
- Aunque sean de sólo lectura se pueden borrar.
- Se recomienda ponerlos offline antes de borrarlos.

Tablespaces y ficheros de datos

- Los tablespaces se componen de uno o más archivos físicos.
- Los ficheros de datos pertenecen a un solo tablespace.



Ficheros de datos

- Los ficheros de datos almacenan toda la información de una base de datos.
- Se pueden tener sólo uno o cientos de ellos.
- Los datos lógicos (tablas, índices, etc.) se almacenan físicamente en los ficheros de datos.
- Un fichero de datos puede ser asociado con una única base de datos.
- Al modificar los datos, no se actualizan inmediatamente en los ficheros de datos, esperan confirmación, para así reducir accesos a disco.

Ficheros de datos

- Los ficheros de datos tienen una propiedad llamada `AUTOEXTEND`, que si está activa, permite que el fichero crezca automáticamente.
- Los ficheros de datos pueden estar online ó offline.
- Al crearlos, están totalmente vacíos, simplemente es un espacio reservado y formateado por Oracle para su uso.
- Se puede limitar la expansión de un archivo con el parámetro `MAXSIZE`.

Creación, manipulación y contenido de los ficheros de datos

- Cada vez que se crea un tablespace nuevo, hay que indicar obligatoriamente cual es el fichero de datos que va a pertenecer a dicho tablespace.
- También se pueden añadir ficheros de datos nuevos a un tablespace que ya existe.
- Cuando un archivo de datos es creado por primera vez, el espacio de disco asignado es formateado pero no contiene los datos del usuario.
- Oracle asigna espacio para los datos asociados a un objeto de esquema en uno o más ficheros de datos de un tablespace.
 - Por lo tanto, un objeto de esquema puede abarcar uno o más ficheros de datos.

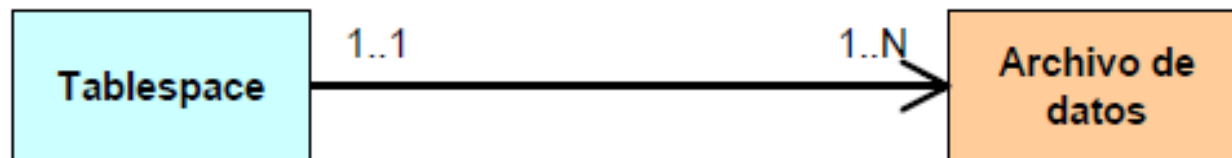
Obteniendo información de los tablespaces y ficheros

- Ejemplo: información sobre los tablespaces

```
SELECT tablespace_name, initial_extent, next_extent, max_extents,  
pct_increase  
FROM dba_tablespaces;
```

- Ejemplo: información sobre los ficheros de datos

```
SELECT file_name, tablespace_name, bytes, autoextensible, maxbytes,  
increment_by  
FROM dba_data_files;
```



Ficheros de control

- Pequeño fichero binario necesario para que la base de datos pueda iniciarse y operar con éxito.
- Contiene la especificación de la estructura física de la base de datos:
 - Nombre de la B.D.
 - Nombres y localización de los fichero de datos y RedoLogs.
 - Fecha y hora de la creación de la base de datos.
- Esta información es utilizada cuando se ejecuta una instancia.
- Deben encontrarse siempre protegidos.

En resumen ...

