

Descripción general y directivas

1. Descripción general

Un tablespace es una unidad lógica de almacenamiento, y está compuesto por uno o varios archivos físicos (archivos de datos).

La mayoría de las operaciones de administración relativas al almacenamiento se realiza a nivel del tablespace y no a nivel de los archivos de datos.


Dentro de un tablespace, el almacenamiento se organiza en segmentos, compuestos por una o varias extensiones (*extent*). Estas extensiones se pueden administrar "por el diccionario" o "localmente". En el primer caso (tablespace administrado por el diccionario) la información de las extensiones libres y asignadas se almacena en tablas del diccionario de datos; en el segundo caso (tablespace administrado localmente) la información de las extensiones libres y asignadas se almacena en el encabezado de los archivos de datos del tablespace.

En la versión 10, Oracle ha introducido la noción de tablespace BIGFILE: un tablespace BIGFILE es un tablespace compuesto por un único archivo de datos que puede ser particularmente voluminoso (hasta 2^{32} bloques Oracle, es decir más de 4 billones de bloques). Por el contrario, un tablespace tradicional, normalmente llamado tablespace SMALLFILE, puede contener varios archivos de datos (hasta 1 022 archivos) pero de tamaño más limitado ("solo" 2^{22} bloques Oracle, es decir, más de 4 millones de bloques).

Cuando un tablespace SMALLFILE contiene varios archivos de datos, los archivos generalmente se ubican en discos diferentes, con dos posibles objetivos:

- asignar espacio adicional a un tablespace, para el que el archivo de datos inicial no puede crecer más;
- repartir el almacenamiento del tablespace en varios discos (*striping* a nivel de Oracle).

Una base de datos siempre tiene, al menos, dos tablespaces llamados SYSTEM y SYSAUX (tablespace SYSTEM AUXiliar, que apareció en la versión 10). El tablespace SYSTEM contiene el diccionario de datos. El tablespace SYSAUX contiene los datos de algunos componentes Oracle. Antes de la versión 10 estos datos se almacenaban en varios tablespaces; por tanto, el uso de un tablespace único permite reducir el número de tablespaces utilizados por una base de datos. Normalmente, los tablespaces SYSTEM y SYSAUX no deben contener datos de usuario. Adicionalmente, una base de datos suele contener (es muy aconsejable) dos tablespaces particulares, utilizados de manera interna por Oracle: el tablespace de anulación (consulte el capítulo Gestión de la información de anulación) y el tablespace temporal. Estos tablespaces "técnicos" no pueden contener datos de usuario. Las tablespaces diferentes a los tablespaces de anulación y temporal se denominan "tablespaces permanentes". Un tablespace puede estar ONLINE (accesible) u OFFLINE. Configurar un tablespace OFFLINE es una manera de dejar algunos datos de la aplicación temporalmente no accesibles, o si la base de datos gestiona varias aplicaciones se puede conseguir que una aplicación no esté accesible sin tocar a otra.

 El tablespace SYSTEM siempre debe estar ONLINE.

Un tablespace puede ser READ WRITE (en modo lectura/escritura) o READ ONLY (en modo solo lectura). Configurar un tablespace READ ONLY es una forma sencilla de garantizar que los datos que contiene nunca serán modificados.

2. Directivas

Las principales directivas relativas a la organización de los tablespaces son las siguientes:

- No alojar datos de usuario en los tablespaces `SYSTEM` y `SYSAUX`;
- Además de los tablespaces `SYSTEM` y `SYSAUX`, crear al menos:
 - un tablespace para los segmentos de anulación (tablespace de anulación);
 - un tablespace para los segmentos temporales (tablespace temporal);
 - un tablespace para los índices.
- si es posible, repartir los archivos de datos de estos tablespaces en discos diferentes.

El tablespace es la unidad básica de trabajo para muchas tareas de administración. Por lo tanto, la regla fundamental es utilizar varios tablespaces para separar al máximo los diferentes tipos de elementos y garantizar una mayor flexibilidad en las operaciones de administración. Adicionalmente, si el servidor tiene varios discos, es posible repartir los archivos de datos en tablespaces de discos diferentes para evitar las restricciones de entradas/salidas.

Existen variaciones respecto a la organización de los tablespaces. Si la base de datos gestiona varias aplicaciones, es mejor crear tablespaces de tablas e índices diferentes para cada aplicación.

Puede utilizar el tablespace `SYSAUX` para sus necesidades de administración (creación de tablas particulares); en este caso, use un esquema separado.

Utilizar varios tablespaces permite:

- separar los datos de la aplicación de los datos del diccionario Oracle;
- separar los datos de varias aplicaciones almacenadas 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 y restauraciones parciales;
- controlar la disponibilidad de los datos.

Para las bases de datos voluminosas y los servidores que tienen varios discos, puede utilizar dos tablespaces (o más) para las tablas y dos tablespaces (o más) para los índices y repartir los archivos de datos de los tablespaces en discos diferentes.

En una configuración como ésta, lo ideal es tener también los tablespaces `SYSTEM`, `SYSAUX`, anulación y temporal en discos diferentes y dedicar otros discos a los archivos de traza. El servidor Oracle ideal debe tener, al menos, diez discos. En la práctica, habitualmente se dispone de medios limitados y en este caso intentamos repartir lo mejor posible las entradas/salidas de los diferentes discos.

Las recomendaciones de Oracle respecto al almacenamiento de los archivos de la base de datos se denominan: *SAME* (*Strip And Mirror Everything*). Oracle recomienda utilizar una tecnología de tipo RAID0+1 en cada eje: "Strip" (RAID0) para el rendimiento y "Mirror" (RAID1) para la seguridad.

Cuando la base de datos se almacena en una matriz de almacenamiento (de tipo SAN), el "stripping" y el "mirroring" están asegurados por la matriz y no es necesario hacerlo manualmente. Sin embargo, para asegurar una buena organización del almacenamiento, utilizar varios tablespaces sigue siendo lo más conveniente.

Tablespace permanente

1. Creación de un tablespace permanente

La sentencia SQL CREATE TABLESPACE permite crear un tablespace permanente.

Sintaxis simplificada

```
CREATE [ BIGFILE | SMALLFILE ] TABLESPACE nombre
DATAFILE especificación_archivo [,...]
[ cláusula_gestión_extensión ]
[ cláusula_gestión_segmento ]
[ MINIMUM EXTENT valor [K|M] ]
[ DEFAULT [ cláusula_compresión ] [ cláusula_almacenamiento ] ]
[ BLOCKSIZE valor [K] ]
[ LOGGING | NOLOGGING ]
[ FORCE LOGGING ]
[ FLASHBACK { ON | OFF } ]
[ ONLINE | OFFLINE ];
    - especificación_archivo
'nombre_archivo' [ SIZE valor [K|M|G|T] ] [REUSE]
[ cláusula_auto_extensión ]
    - cláusula_auto_extensión
    AUTOEXTEND OFF
| AUTOEXTEND ON [ NEXT valor [K|M|G|T] ]
    [ MAXSIZE UNLIMITED | valor [K|M|G|T] ]
    - cláusula_gestión_extent
EXTENT MANAGEMENT DICTIONARY
| EXTENT MANAGEMENT LOCAL
    { AUTOALLOCATE | UNIFORM [ SIZE valor [K|M|G|T] ] }
    - cláusula_gestión_segmento
SEGMENT SPACE MANAGEMENT { MANUAL | AUTO }
    - cláusula_almacenamiento
STORAGE ( [ INITIAL valor [K|M] ]
    [ NEXT valor [K|M] ]
    [ MINEXTENTS valor ]
    [ MAXEXTENTS { valor | UNLIMITED } ]
    [ PCTINCREASE valor ] )
- cláusula_compresión
COMPRESS [ { BASIC | FOR OLTP } ]
| NOCOMPRESS
```

Ejemplo:

Tablespace para las tablas, con una gestión local uniforme de las extensiones:

```
CREATE TABLESPACE data
DATAFILE 'e:\oradata\NuestraBase\data01.dbf' SIZE 500M
    AUTOEXTEND ON NEXT 100M MAXSIZE 800M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 10M
SEGMENT SPACE MANAGEMENT AUTO;
```

Tablespace para los índices, con una gestión local automática de las extensiones:

```
CREATE TABLESPACE index
DATAFILE 'e:\oradata\NuestraBase\indx01.dbf' SIZE 500M
        AUTOEXTEND ON NEXT 100M MAXSIZE 800M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```



Todas las operaciones relativas a los tablespaces y a los archivos de datos se guardan en el archivo de alertas de la instancia.

Las opciones de la sentencia SQL CREATE TABLESPACE son:

BIGFILE | SMALLFILE

Esta cláusula indica si el tablespace es un tablespace BIGFILE o SMALLFILE. Si esta cláusula se omite, Oracle utiliza el tipo por defecto definido a nivel de la base de datos (consulte la sección Observaciones de los tablespaces BIGFILE).

nombre

Nombre del tablespace.

DATAFILE especificación_archivo

Esta cláusula permite concretar la ubicación, el nombre y el tamaño de uno (o eventualmente varios) archivo(s) de datos para el tablespace. Es posible especificar un único archivo de datos si el tablespace es de tipo BIGFILE.

Para la especificación de un archivo de datos, la sintaxis es la siguiente:

```
'nombre_archivo' [ SIZE valor [K|M|G|T] ] [REUSE]
[ cláusula_auto_extend ]
```

nombre_archivo	Ruta de acceso completa al archivo de datos, normalmente en un repositorio de datos (oradata), para respetar el estándar OFA.
SIZE	Tamaño inicial del archivo. El tamaño se puede omitir solo si se utiliza la opción REUSE y el archivo ya existe.
REUSE	Si la opción está presente y el archivo ya existe, Oracle lo reutiliza y lo elimina. Si la opción no se informa, en la misma situación se muestra un mensaje de error y se detiene la creación del tablespace. Desde un punto de vista de la seguridad es preferible no seleccionar esta opción para evitar borrar por equivocación un archivo de datos utilizado por otra base de datos.

Sintaxis de la cláusula cláusula_auto_extend:

```
AUTOEXTEND OFF
| AUTOEXTEND ON [ NEXT valor [K|M|G|T] ]
[ MAXSIZE UNLIMITED | valor [K|M|G|T] ]
```

AUTOEXTEND	Indica si el archivo de datos puede (ON) o no (OFF) crecer cuando se utiliza todo el espacio inicialmente asignado.
NEXT	Espacio mínimo asignado al archivo durante la extensión.
MAXSIZE	Tamaño máximo del archivo, eventualmente no limitado (UNLIMITED).

Todos los tamaños se pueden expresar en bytes (sin símbolo), KB (símbolo K), MB (símbolo M), GB (símbolo G) o TB (símbolo T). Los tamaños en TB solo están permitidos para los tablespaces BIGFILE.

Si el archivo de datos no es autoextensible, se producirá un mensaje de error si un segmento almacenado en el tablespace en cuestión no tiene suficiente espacio durante su creación inicial o su extensión. Hacer que el archivo de datos sea autoextensible permite eliminar este tipo de problemas, hasta alcanzar el tamaño límite permitido para el archivo. Si el archivo es autoextensible, el tamaño mínimo de extensión se especifica mediante la opción NEXT. El tamaño real de cada extensión será igual a NEXT si el tamaño solicitado es inferior a NEXT e igual al tamaño solicitado en caso contrario.

Ejemplo con un NEXT de 500K:

- tamaño solicitado = 200 KB: 500 KB asignados al archivo;
- tamaño solicitado = 750 KB: 750 KB asignados al archivo.



Preste atención a la opción UNLIMITED. El disco duro está "ilimitado". En la práctica, es preferible obtener un mensaje de error de Oracle indicando que ha alcanzado el límite especificado que obtener un mensaje de error de Oracle causado por un error del sistema operativo que diga que el disco está lleno.

EXTENT MANAGEMENT ...

Esta cláusula permite definir el modo de gestión de las extensiones dentro del tablespace (por el diccionario o localmente). La organización del almacenamiento en un tablespace se presenta más adelante.

SEGMENT SPACE MANAGEMENT { MANUAL | AUTO }

Esta cláusula permite definir el modo de gestión del espacio libre de los segmentos almacenados en el tablespace (consulte el capítulo Gestión de las tablas e índices). Esta cláusula solo es válida para un tablespace administrado localmente.

MINIMUM EXTENT valor [K|M]

Esta cláusula permite definir el tamaño mínimo de las extensiones en el tablespace. Todas las extensiones asignadas en el tablespace tendrán un tamaño múltiplo del tamaño mínimo. Esta cláusula solo es válida para un tablespace administrado por el diccionario (consulte la sección Organización del almacenamiento dentro de un tablespace).

DEFAULT [COMPRESS | NOCOMPRESS] cláusula_almacenamiento

Esta cláusula permite definir una cláusula de almacenamiento por defecto para los segmentos que se crearán en el tablespace sin cláusula de almacenamiento. La parte `cláusula_almacenamiento` (STORAGE...) solo es válida para un tablespace administrado por el diccionario; la parte `cláusula_compresión` es válida independientemente del modo de gestión del tablespace.

La organización del almacenamiento en un tablespace se presenta en la sección Organización del almacenamiento

dentro de un tablespace. El almacenamiento de los segmentos se detalla en el capítulo Gestión de las tablas e índices.

BLOCKSIZE valor [K]

Esta cláusula define el tamaño de bloque utilizado por el tablespace. Los valores permitidos son 2 KB, 4 KB, 8 KB, 16 KB y 32 KB (algunas plataformas son más restrictivas). El valor por defecto es el tamaño de bloque estándar definido con el argumento `DB_BLOCK_SIZE`. Para utilizar un tamaño de bloque no estándar para un tablespace debe configurar un pool para este tamaño de bloque en el *Buffer Cache* gracias a alguno de los argumentos `DB_nK_CACHE_SIZE` (siendo *n* un valor entre 2, 4, 8, 16 o 32). En caso contrario, obtendrá el siguiente error:

```
ORA-29339: el tamaño de bloque de tablespace nnn no se corresponde con  
los tamaños de bloque configurados
```

LOGGING | NOLOGGING

Esta cláusula define el modo de traza por defecto de los segmentos que se almacenarán en el tablespace y para los no se ha definido ningún modo de traza. `NOLOGGING` elimina la traza de algunas operaciones (inserción por carga directa, creación de tabla a partir de una consulta, creación o reconstrucción de índices). `LOGGING` es la opción por defecto. La opción `NOLOGGING` se ignora si el tablespace o la base de datos están configurados en modo `FORCE LOGGING`.

FORCE LOGGING

Esta cláusula configura el tablespace en modo `FORCE LOGGING`, lo que permite garantizar que todas las modificaciones se guardarán en los archivos de traza, incluso si la operación implicada se realiza en modo `NOLOGGING`.

FLASHBACK { ON | OFF }

Esta cláusula indica si el tablespace participa o no en las operaciones de `FLASHBACK DATABASE` (consulte el capítulo Copia de seguridad y restauración).

ONLINE | OFFLINE

Esta cláusula indica si el tablespace se crea `ONLINE` (defecto) u `OFFLINE`.



Desde la versión 11, un tablespace se puede cifrar (cláusula `ENCRYPTION`). Esta funcionalidad, que necesita la opción Advanced Security, no se presenta en este libro. Para obtener más información consulte la documentación "Oracle® Database Advanced Security Administrator's Guide".

2. Observaciones de los tablespaces BIGFILE

Los tablespaces `BIGFILE` simplifican la administración, ofreciendo transparencia del archivo de datos. Como veremos más adelante, algunas operaciones se pueden realizar directamente en el tablespace `BIGFILE` (mediante una sentencia SQL `ALTER TABLESPACE`) y no con los archivos de datos, como es el caso para un tablespace `SMALLFILE`.

Los tablespaces `BIGFILE` forzosamente se administran localmente (`EXTENT MANAGEMENT LOCAL`), con una

gestión automática del espacio en los segmentos (SEGMENT SPACE MANAGEMENT **AUTO**).

Ejemplo de tablespace BIGFILE

```
CREATE BIGFILE TABLESPACE soy_grande
DATAFILE 'e:\oradata\NuestraBase\soy_grande.dbf' SIZE 10G;
```

Los tablespaces BIGFILE están destinados a utilizarse con un administrador de unidades lógicas, que soporta el *striping* o el RAID. Si no fuera el caso, es posible tener problemas de rendimiento con la ejecución en paralelo de consultas o copias de seguridad RMAN.

Es posible definir un tipo por defecto SMALLFILE o BIGFILE a nivel de la base de datos, ya sea durante la creación de la base de datos (cláusula SET DEFAULT TABLESPACE, de la sentencia SQL CREATE DATABASE - consulte la sección Creación de la base de datos manualmente del capítulo Creación de una nueva base de datos) o más adelante, mediante la sentencia SQL ALTER DATABASE.

Sintaxis:

```
ALTER DATABASE SET DEFAULT { SMALLFILE | BIGFILE } TABLESPACE;
```

El tipo por defecto actual se puede consultar en la vista DATABASE_PROPERTIES, gracias a la propiedad DEFAULT_TBS_TYPE:

```
SQL> SELECT property_value FROM database_properties
      2 WHERE property_name = 'DEFAULT_TBS_TYPE';
PROPERTY_VALUE
-----
SMALLFILE
```

3. Tablespace permanente predeterminado

Cuando un usuario crea un segmento sin concretar el tablespace (consulte la sección Organización del almacenamiento dentro de un tablespace), Oracle almacena el segmento en el tablespace por defecto del usuario.

Este tablespace por defecto se define gracias a la cláusula DEFAULT TABLESPACE de las sentencias SQL CREATE USER y ALTER USER (consulte el capítulo Gestión de usuarios y sus permisos). Si esta cláusula se omite, es el tablespace SYSTEM el que se asigna como tablespace por defecto del usuario. En la práctica, este comportamiento por defecto no supone ningún problema porque los usuarios no DBA no tienen (normalmente - es el caso por defecto) cuotas en el tablespace SYSTEM y, por tanto, no pueden crear segmentos en él (consulte el capítulo Gestión de usuarios y sus permisos).

Desde la versión 10, con el objetivo de simplificar la gestión de los usuarios, es posible definir un tablespace permanente predeterminado. Este tablespace se asigna por defecto a los usuarios durante su creación cuando se omite la cláusula DEFAULT TABLESPACE de la sentencia SQL CREATE USER. Esta técnica no impide utilizar otros tablespaces permanentes, asignados específicamente a los usuarios por necesidades particulares.

El tablespace permanente predeterminado se puede definir durante la creación de la base de datos gracias a la cláusula DEFAULT TABLESPACE de la sentencia SQL CREATE DATABASE.

Sintaxis

```
[ DEFAULT TABLESPACE nombre
  [ DATAFILE especificación_archivo [,...] ]
  [ cláusula_extent_management ] ]
```

Ejemplo:

```
DEFAULT TABLESPACE deftbs
DATAFILE 'e:\oradata\NuestraBase\deflbs01.dbf' SIZE 10M
      AUTOEXTEND ON NEXT 10M MAXSIZE 500M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
```

Observe que el tablespace definido de esta manera obligatoriamente es de tipo `SMALLFILE`.

Para crear y definir un tablespace permanente predeterminado después de la creación de la base de datos debe:

- crear un tablespace permanente mediante la sentencia SQL `CREATE TABLESPACE`, presentada anteriormente;
- definirlo como tablespace permanente predeterminado gracias a la cláusula `DEFAULT TABLESPACE` de la sentencia SQL `ALTER DATABASE`.

Sintaxis

```
ALTER DATABASE DEFAULT TABLESPACE nombre;
```

nombre debe hacer referencia a un tablespace permanente ya existente.

Cuando esta sentencia SQL se ejecuta, a todos los usuarios que tenían asignado el antiguo tablespace permanente predeterminado se les asigna automáticamente el nuevo.

Para encontrar el nombre del tablespace permanente predeterminado puede consultar la vista `DATABASE_PROPERTIES`, utilizando la propiedad `DEFAULT_PERMANENT_ TABLESPACE`:

```
SQL> SELECT property_value FROM database_properties
      2 WHERE property_name = 'DEFAULT_PERMANENT_TABLESPACE';
PROPERTY_VALUE
-----
DEFTBS
```

4. Modificación de un tablespace permanente

a. Descripción general

Es posible modificar un tablespace después de su creación, fundamentalmente para:

- renombrarlo;
- asignarle espacio adicional;
- mover el archivo(s) de datos;

- pasarla de OFFLINE / ONLINE;
- pasar de READ ONLY / READ WRITE;
- modificar el resto de características (LOGGING / NOLOGGING, FORCE LOGGING, FLASHBACK ON / OFF, etc.).

Estas operaciones se realizan, según los casos, con las sentencias SQL ALTER TABLESPACE o ALTER DATABASE.

Es posible asignar el espacio adicional a una base de datos:

- añadiendo un nuevo tablespace (con uno o varios archivos de datos);
- añadiendo un archivo de datos a un tablespace existente;
- aumentando el tamaño de un archivo de datos de un tablespace.

La sintaxis completa de la sentencia SQL ALTER TABLESPACE es "excesivamente larga"; por tanto, no vamos a presentarla completamente, pero sí es necesario indicar la sintaxis que se va a utilizar para realizar diferentes operaciones.

b. Renombrar un tablespace

Renombrar un tablespace se realiza mediante la sentencia SQL ALTER TABLESPACE. Esta posibilidad apareció en la versión 10.

Sintaxis:

```
ALTER TABLESPACE antiguo_nombre RENAME TO nuevo_nombre;
```

Ejemplo:

```
ALTER TABLESPACE deftbs RENAME TO tbsdef;
```

Los tablespaces SYSTEM y SYSAUX, así como los tablespaces OFFLINE, no se pueden renombrar.

Observe que, en el caso del tablespace OFFLINE, el mensaje de error indica que el archivo de datos está "fuera de servicio " (OFFLINE), lo que impide que Oracle pueda modificar el encabezado del archivo de datos para registrar en él el nuevo nombre del tablespace. Un problema similar se podría producir con un tablespace en modo de solo lectura, pero no es el caso; Oracle no modifica el encabezado del archivo de datos, sino que guarda el nuevo nombre en el archivo de control (el encabezado se modificará cuando el tablespace vuelva a pasar a modo lectura/escritura).

c. Añadir un archivo de datos a un tablespace

Es posible añadir un archivo de datos a un tablespace mediante la sentencia SQL ALTER TABLESPACE.

Sintaxis

```
ALTER TABLESPACE nombre ADD DATAFILE especificación_archivo [...];
```

Ejemplo:

```
ALTER TABLESPACE data
  ADD DATAFILE 'f:\oradata\NuestraBase\data02.dbf' SIZE 100M
  AUTOEXTEND ON NEXT 100M MAXSIZE 500M;
```

Añadir un archivo de datos a un tablespace es un primer paso para asignarle espacio adicional; generalmente, este método se utiliza para asignar un nuevo archivo de datos en un disco diferente al disco utilizado actualmente por el tablespace (en caso contrario, es posible modificar el tamaño del archivo de datos existente - lo veremos más adelante). Esta operación está prohibida para un tablespace BIGFILE.

La especificación del archivo de datos (*especificación_archivo*) es la misma que durante la creación del tablespace (sección Creación de un tablespace permanente).

d. Modificar el tamaño de un archivo de datos

Es posible modificar el tamaño de un archivo de datos mediante la sentencia SQL `ALTER DATABASE`, o bien la sentencia SQL `ALTER TABLESPACE` en el caso de un tablespace BIGFILE.

Sintaxis

```
ALTER DATABASE
  DATAFILE 'nombre_completo' | número_archivo [,...] RESIZE valor [K|M|G|T];
ALTER TABLESPACE nombre_tablespace_bigfile RESIZE valor [K|M|G|T];
```

Ejemplo:

- Cualquier tipo de tablespace:

```
ALTER DATABASE
  DATAFILE 'f:\oradata\NuestraBase\data02.dbf' RESIZE 200M;
```

- Tablespace BIGFILE únicamente:

```
ALTER TABLESPACE soy_grande RESIZE 1T;
```

La cláusula `RESIZE` define el nuevo tamaño deseado (por encima o por debajo), para el archivo de datos.

Modificar el tamaño de un archivo de datos permite:

- en el caso de una reducción, recuperar el espacio no utilizado asignado al tablespace;
- en el caso de un aumento, asignar espacio adicional a un tablespace.

En caso de una reducción, el tamaño del archivo de datos no puede disminuir por debajo de la posición de la última extensión ocupada por un segmento en el tablespace (visible en la vista `DBA_EXTENTS`). De lo contrario se muestra un mensaje de error y el tamaño del archivo no se modifica:

```
ORA-03297: el archivo contiene datos utilizados más allá del valor RESIZE necesario
```

e. Modificar la extensión automática de un archivo de datos

Es posible modificar la extensión automática de un archivo de datos mediante la sentencia SQL `ALTER`

DATABASE, o bien la sentencia SQL ALTER TABLESPACE en el caso de un tablespace BIGFILE.

Sintaxis

```
ALTER DATABASE  
DATAFILE 'nombre_completo' | número_archivo[,...] cláusula_auto_extensión;  
ALTER TABLESPACE nombre_tablespace_bigfile cláusula_auto_extensión;
```

La especificación de la cláusula de extensión automática (cláusula_auto_extensión) es la misma que durante la creación del tablespace (sección Creación de un tablespace permanente).

Ejemplo:

- Desactivación de la cláusula AUTOEXTEND:

```
ALTER DATABASE  
DATAFILE 'e:\oradata\NuestraBase\data01.dbf' AUTOEXTEND OFF;
```

- Activación (o modificación) de la cláusula AUTOEXTEND:

```
ALTER DATABASE  
DATAFILE 'e:\oradata\NuestraBase\data01.dbf'  
AUTOEXTEND ON NEXT 200M MAXSIZE 800M;
```

- Ejemplo con un tablespace BIGFILE:

```
ALTER TABLESPACE soy_grande  
AUTOEXTEND ON NEXT 1G MAXSIZE 100G;
```

Activar la extensión automática de un archivo de datos permite a este último crecer automáticamente en caso de necesitar espacio adicional para un segmento (nuevo o ya presente) en el tablespace; es un buen sistema para evitar problemas y no tener que aumentar el tamaño de un archivo de datos (como vimos anteriormente). Es previsible deshabilitar la extensión automática de un archivo de datos (incluso es aconsejable) si no hay espacio disponible en un disco.

f. Poner un tablespace en modo OFFLINE / ONLINE

Es posible poner un tablespace en modo OFFLINE / ONLINE mediante la sentencia SQL ALTER TABLESPACE.

Sintaxis

```
ALTER TABLESPACE nombre ONLINE | OFFLINE;
```

Ejemplo:

- Desactivación:

```
ALTER TABLESPACE data OFFLINE;
```

- Activación:

```
ALTER TABLESPACE data ONLINE;
```

Desactivar un tablespace puede ser necesario para realizar algunas operaciones de administración en el tablespace (por ejemplo, mover alguno de sus archivos de datos) o simplemente para hacer que algunos datos estén temporalmente no accesibles.

El tablespace `SYSTEM` no se puede poner `OFFLINE`; aparece un mensaje de error en caso de intentarlo. El tablespace `SYSAUX` se puede pasar a modo `OFFLINE`, pero algunas funcionalidades corren el riesgo de dejar de funcionar.

El estado de un tablespace (`OFFLINE` / `ONLINE`) se conserva durante la parada; durante el siguiente inicio de la base de datos el tablespace permanece en el estado en el que estuviera durante la parada.

Existen opciones en el modo `OFFLINE` que se deben utilizar si el tablespace que se va a desactivar está dañado (consulte el capítulo Copia de seguridad y restauración).

g. Renombrar o mover un archivo de datos fuera de servicio

Es posible renombrar o mover un archivo de datos fuera de servicio (`OFFLINE`) mediante la sentencia SQL `ALTER TABLESPACE ... RENAME DATAFILE` o `ALTER DATABASE RENAME FILE`.

En caso de usar la sentencia SQL `ALTER TABLESPACE`, la base de datos debe estar abierta, pero el tablespace implicado debe estar `OFFLINE`. En caso de usar la sentencia SQL `ALTER DATABASE`, el tablespace en cuestión debe estar `OFFLINE` o la base de datos en estado `MOUNT`. Es necesario usar la sentencia SQL `ALTER DATABASE`, con la base de datos montada, para mover un archivo de datos del tablespace `SYSTEM`, porque este último no se puede poner `OFFLINE`.



Estas dos sentencias SQL no manipulan físicamente el archivo. Solo actualizan el archivo de control. El archivo de datos se debe renombrar/copiar/mover con ayuda de un comando del sistema operativo antes de ejecutar la sentencia SQL.

Sintaxis

```
- ALTER TABLESPACE
ALTER TABLESPACE nombre
RENAME DATAFILE 'antiguo_nombre_completo'
                TO 'nuevo_nombre_completo';
- ALTER DATABASE
ALTER DATABASE
RENAME FILE 'antiguo_nombre_completo'|numero_archivo
            TO 'nuevo_nombre_completo';
```

"Renombrar" un archivo de datos se utiliza, principalmente, para mover el archivo. Esta posibilidad resulta interesante si el tablespace está lleno y no queda más espacio disponible en el disco en el que está ubicado actualmente; en este caso, es posible mover el archivo de datos del tablespace a un disco que tenga espacio disponible y hacerlo crecer (o darle autorización para que pueda hacerlo).

El modo operativo durante el uso de la sentencia SQL `ALTER TABLESPACE` es el siguiente:

- Conectarse como DBA:

```
SQL> CONNECT system/xxxx
```

- Pasar el tablespace a modo `OFFLINE`:

```
SQL> ALTER TABLESPACE data OFFLINE;
```

- Con un comando del sistema operativo, renombrar, copiar o mover el archivo:

```
SQL> HOST move e:\oradata\NuestraBase\data01.dbf -  
> f:\oradata\NuestraBase\data01.dbf
```

- Ejecutar la sentencia SQL ALTER TABLESPACE:

```
SQL> ALTER TABLESPACE data  
2 RENAME DATAFILE 'e:\oradata\NuestraBase\data01.dbf'  
3 TO 'f:\oradata\NuestraBase\data01.dbf';
```

- Volver a pasar el tablespace a modo ONLINE:

```
SQL> ALTER TABLESPACE data ONLINE;
```

El modo operativo durante el uso de la sentencia SQL ALTER DATABASE es el siguiente:

- Conectarse AS SYSDBA:

```
SQL> CONNECT / AS SYSDBA
```

- Pasar la base de datos al estado MOUNT:

```
SQL> SHUTDOWN IMMEDIATE  
SQL> STARTUP MOUNT
```

- Con un comando del sistema operativo, renombrar, copiar o mover el archivo:

```
SQL> HOST move e:\oradata\NuestraBase\system01.dbf -  
> f:\oradata\NuestraBase\system01.dbf
```

- Ejecutar la sentencia SQL ALTER DATABASE:

```
SQL> ALTER DATABASE  
2 RENAME FILE 'e:\oradata\NuestraBase\system01.dbf'  
3 TO 'f:\oradata\NuestraBase\system01.dbf';
```

- Abrir la base de datos:

```
SQL> ALTER DATABASE OPEN;
```

h. Renombrar o mover un archivo de datos en línea

Desde la versión 12, es posible renombrar o mover un archivo de datos en línea (ONLINE) mediante la sentencia SQL ALTER DATABASE MOVE DATAFILE. Esta funcionalidad solo está disponible en Enterprise Edition.

Sintaxis

```
ALTER DATABASE  
MOVE DATAFILE 'antiguo_nombre_completo' | número_archivo  
TO 'nuevo_nombre_completo' [KEEP];
```

A diferencia de lo que sucede con el uso de la cláusula RENAME, que se ha presentado anteriormente, esta sentencia SQL renombra o mueve el archivo de datos a nivel del sistema operativo y después actualiza el archivo

de control.

Por defecto, si se especifica una nueva ubicación para el archivo de datos, el comando copia y después elimina el archivo de datos de origen. Si lo desea, puede conservar el archivo de datos de origen usando la cláusula `KEEP`.

El modo operativo es el siguiente:

- Conectarse como DBA:

```
SQL> CONNECT system/xxxx
```

- Ejecutar el comando:

```
SQL> ALTER DATABASE  
2      MOVE DATAFILE 'E:\ORADATA\NUESTRABASE\DEFTBS01.DBF'  
3      TO 'F:\ORADATA\NUESTRABASE\DEFTBS01.DBF';
```



En plataformas Windows, el archivo de datos de origen no se puede eliminar, incluso si la cláusula `KEEP` se omite. En ese caso, se escribe un mensaje en el archivo de alertas de la instancia. En ese momento se debe eliminar manualmente el archivo de datos a nivel del sistema operativo.

i. Eliminar un archivo de datos

Es posible eliminar un archivo de datos de un tablespace mediante la sentencia SQL `ALTER TABLESPACE`.

Sintaxis

```
ALTER TABLESPACE nombre DROP DATAFILE 'nombre_completo' | número_archivo;
```

Ejemplo

```
ALTER TABLESPACE data DROP DATAFILE 'E:\ORADATA\NUESTRABASE\DATA02.DBF';
```

Oracle elimina físicamente el archivo de datos. Se aplican las siguientes restricciones:

- El archivo de datos debe estar vacío (no debe contener ninguna extensión);
- El archivo de datos no puede ser el primer archivo creado para el tablespace;
- El archivo de datos no debe pertenecer a un tablespace en modo de solo lectura;
- El archivo de datos debe estar en línea (`ONLINE`);
- El archivo no debe pertenecer al tablespace `SYSTEM`.

j. Otras operaciones

La sentencia SQL `ALTER TABLESPACE` se puede utilizar para modificar las características del tablespace:

```
- READ ONLY / READ WRITE  
ALTER TABLESPACE nombre { READ ONLY | READ WRITE };  
- LOGGING / NOLOGGING  
ALTER TABLESPACE nombre LOGGING | NOLOGGING;
```

```
- FORCE LOGGING
ALTER TABLESPACE nombre [NO] FORCE LOGGING;
- FLASHBACK ON / OFF
ALTER TABLESPACE nombre FLASHBACK ON | OFF;
```

5. Eliminación de un tablespace permanente

La sentencia SQL DROP TABLESPACE permite eliminar un tablespace permanente.

Sintaxis

```
DROP TABLESPACE nombre [ INCLUDING CONTENTS [ AND DATAFILES ]
[ CASCADE CONSTRAINTS ] ];
```

Ejemplo:

```
DROP TABLESPACE data INCLUDING CONTENTS AND DATAFILES;
```

Es una sentencia DDL (*Data Definition Language*): no hay ROLLBACK. La única solución es volver a partir de una copia de seguridad; el archivo físico, incluso si no se elimina, no es recuperable.

El tablespace SYSTEM y el tablespace permanente predeterminado no se pueden eliminar. Se recomienda pasar el tablespace a modo OFFLINE antes de eliminarlo.

Las opciones de la sentencia SQL DROP TABLESPACE son:

INCLUDING CONTENTS

Esta cláusula es necesaria si el tablespace no está vacío, y permite forzar la eliminación anticipada de los segmentos que almacena. Si el tablespace no está vacío y la opción no se utiliza se produce el error ORA-01549:

```
ORA-01549: el tablespace no está vacío. Utilice la opción INCLUDING CONTENTS
```

AND DATAFILES

Además, esta opción de la cláusula anterior permite eliminar los archivos físicos del tablespace. Se obtiene un mensaje en el archivo de alertas de la instancia para cada archivo físico eliminado por Oracle.



En caso contrario, no se eliminan.

En plataformas Windows, es posible que no se eliminen los archivos físicos cuando se utilizara la cláusula AND DATAFILES. Cuando se da este caso, se escribe un mensaje en el archivo de alertas de la instancia. Los archivos afectados se deben eliminar manualmente a nivel del sistema operativo.

CASCADE CONSTRAINTS

Además, esta cláusula permite eliminar las restricciones de integridad referencial definidas en las tablas fuera del tablespace y que referencian a tablas que están dentro del tablespace. Si existen restricciones como esta y no se

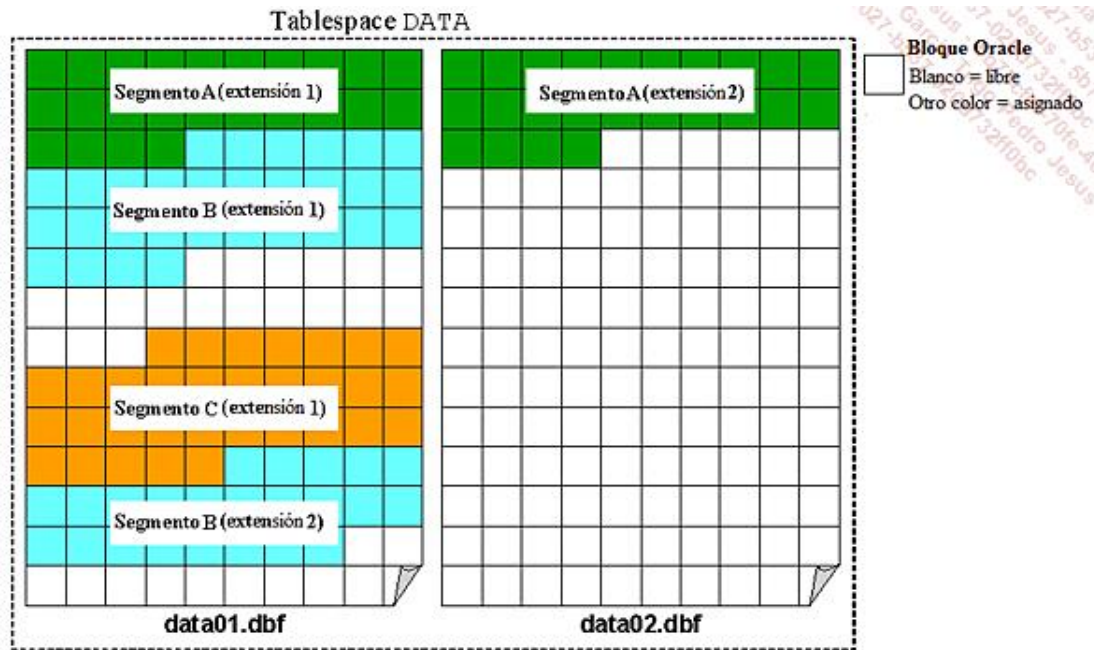
utiliza la opción, se obtiene el error ORA-02449:

ORA-02449: claves únicas/primarias de la tabla, referenciadas por claves extranjeras

Organización del almacenamiento dentro de un tablespace

1. Principios generales

La organización del almacenamiento dentro de un tablespace se puede resumir mediante el siguiente esquema.



Dentro de un tablespace, el almacenamiento se organiza en segmentos que contienen una o varias extensiones (*extents*), siendo una extensión un conjunto de bloques Oracle contiguos.

Cuando se crea un segmento en un tablespace, Oracle le asigna una o varias extensiones en alguno de los archivos de datos del tablespace. Cuando el espacio inicialmente asignado está lleno (después de la inserción de datos, por ejemplo), Oracle asigna una nueva extensión al segmento, y así sucesivamente. Todas las extensiones asignadas a un segmento están en el tablespace de creación del segmento, pero no obligatoriamente juntas, ni forzosamente en el mismo archivo de datos (si el tablespace está compuesto por varios archivos de datos). Cuando se elimina un segmento, las extensiones que ocupa se liberan y quedan disponibles para otros segmentos. La creación/eliminación frecuente de segmentos en un tablespace pueden conducir a una fragmentación del espacio disponible en este tablespace.

Para recordar, existen cuatro tipos principales de segmentos:

- los segmentos de tabla: espacio ocupado por las tablas;
- los segmentos de índice: espacio ocupado por los índices;
- los segmentos de anulación: espacio temporal utilizado para almacenar la información que permite anular una transacción;
- los segmentos temporales: espacio temporal utilizado fundamentalmente durante una operación de ordenación.

La primera extensión de un segmento contiene, al menos, dos bloques. El primero se reserva al encabezado del segmento (no contiene datos útiles, sino la libreta de extensiones asignadas al segmento). Sucede lo mismo para cada archivo de datos del tablespace; el primer bloque es un bloque de encabezado (veremos que el encabezado del archivo puede contener más bloques).

Veremos en el capítulo Gestión de las tablas e índices cómo, bajo ciertas condiciones, es posible liberar extensiones sin eliminar el segmento.

Un tablespace se puede "gestionar por el diccionario" o "gestionar localmente".

En un tablespace "gestionado por el diccionario" la información relativa a la gestión del espacio (extensiones libres/asignadas) se guarda en el diccionario de datos.

En un tablespace "gestionado localmente" la información relativa a la gestión del espacio (extensiones libres/asignadas) se guarda en un *bitmap*, en el encabezado de cada archivo de datos del tablespace. Cada bit del *bitmap* corresponde a una extensión y vale 0 o 1, dependiendo de si la extensión está libre o asignada.

Los tablespaces administrados localmente aparecieron en Oracle8i. Desde Oracle9i, por defecto los tablespaces se administran localmente (salvo el tablespace SYSTEM que, por defecto, se administra por el diccionario - se verá más adelante).



Oracle recomienda utilizar tablespaces administrados localmente. Es el único tipo de tablespace que se estudiará en este libro. Es muy probable que Oracle no soporte los tablespaces administrados por el diccionario en versiones futuras.

Oracle ofrece dos variantes para los tablespaces administrados localmente:

- Una gestión "automática": el tamaño de las extensiones se determina automáticamente por Oracle.
- Una gestión "uniforme": el tamaño de las extensiones es uniforme, igual a un valor definido durante la creación del tablespace.

Por defecto, un tablespace permanente administrado localmente está configurado en modo gestión automática de las extensiones; la gestión uniforme se debe especificar.

Un tablespace temporal administrado localmente obligatoriamente está en modo gestión uniforme de las extensiones (se detalla más adelante, en la sección Tablespace temporal de este capítulo).

2. Especificar el almacenamiento de un segmento

Las cláusulas TABLESPACE y STORAGE se pueden utilizar en las sentencias de creación de los segmentos para especificar el almacenamiento del segmento.

Sintaxis de la cláusula TABLESPACE

```
TABLESPACE nombre_tablespace
```

Sintaxis de la cláusula STORAGE

```
STORAGE ( [ INITIAL valor [K|M] ]  
          [ NEXT valor [K|M] ]  
          [ MINEXTENTS valor ]  
          [ MAXEXTENTS { valor | UNLIMITED } ]  
          [ PCTINCREASE valor ] )
```

Ejemplo para una tabla:

```
CREATE TABLE categoria
(
  identificador  NUMBER(6),
  nombre         VARCHAR2(20)
)
TABLESPACE data
STORAGE (INITIAL 500K);
```

Las opciones de la cláusula STORAGE son:

INITIAL	Tamaño de la primera extensión asignada.
NEXT	Tamaño de la siguiente extensión asignada.
MINEXTENTS	Número inicial de extensiones asignadas
MAXEXTENTS	Número máximo de extensiones asignables.
PCTINCREASE	Porcentaje de aumento (0 a 100) del tamaño de las extensiones, a partir de la tercera, respecto a la anterior.

La manera en que Oracle usa la cláusula STORAGE depende del modo de gestión de las extensiones dentro del tablespace.

La cláusula STORAGE realmente tiene mucha importancia para el almacenamiento de los segmentos en un tablespace administrado por el diccionario, porque permite especificar de manera precisa el almacenamiento del segmento. Si se define una cláusula MINIMUM EXTENT a nivel del tablespace, el tamaño de las extensiones eventualmente se ajusta para que sea un múltiplo de este tamaño mínimo. En caso de ausencia de la cláusula STORAGE, el segmento hereda la cláusula DEFAULT STORAGE, eventualmente definida a nivel del tablespace. Si esta última no está, Oracle utiliza valores por defecto (INITIAL = 5 bloques Oracle, NEXT = 5 bloques Oracle, PCTINCREASE = 50). En caso de un tablespace administrado localmente, la cláusula STORAGE tiene mucha menos importancia porque, por su definición, el tablespace impone restricciones al tamaño de las extensiones (tamaño elegido por Oracle o tamaño uniforme). En la práctica, solo la opción INITIAL tiene realmente importancia, porque indica a Oracle el tamaño inicial deseado para el segmento.

3. Especificar el modo de gestión de un tablespace

La cláusula EXTENT MANAGEMENT de la sentencia SQL CREATE TABLESPACE permite especificar el modo de gestión de un tablespace.

Sintaxis:

```
EXTENT MANAGEMENT
  DICTIONARY
  | LOCAL [ AUTOALLOCATE | UNIFORM [ SIZE valor [K|M] ] ]
```

Ejemplo:

- Tablespace administrado localmente con extensiones uniformes:

```
CREATE TABLESPACE tbs_local_uniform
  DATAFILE 'e:\oradata\NuestraBase\tbs_local_uniform.dbf' SIZE 10M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

- Tablespace administrado localmente con extensiones administradas por Oracle:

```
CREATE TABLESPACE tbs_local_auto
  DATAFILE 'd:\oradata\NuestraBase\tbs_local_auto.dbf' SIZE 10M
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

Las opciones de la cláusula `EXTENT MANAGEMENT` son:

DICTIONARY	Indica que el tablespace se administra por el diccionario. Las cláusulas <code>DEFAULT STORAGE</code> y <code>MINIMUM EXTENT</code> se pueden indicar adicionalmente.
LOCAL	Indica que el tablespace se administra localmente. Las cláusulas <code>DEFAULT STORAGE</code> y <code>MINIMUM EXTENT</code> están prohibidas.
AUTOALLOCATE	Indica que las extensiones se administran automáticamente por Oracle.
UNIFORM	Indica que las extensiones tienen un tamaño uniforme, definido por la cláusula <code>SIZE</code> . Si la cláusula <code>SIZE</code> no se especifica, el tamaño por defecto es 1 MB.
SIZE	Especifica el tamaño de las extensiones para los tablespaces <code>LOCAL</code> <code>UNIFORM</code> . El tamaño se puede definir en bytes (sin símbolo), en KB (símbolo K) o en MB (símbolo M).

Por defecto (cláusula `EXTENT MANAGEMENT` ausente) un tablespace permanente se administra localmente, con una gestión automática de las extensiones (`AUTOALLOCATE`).

Como veremos posteriormente, un tablespace temporal administrado localmente obligatoriamente está configurado en gestión uniforme de las extensiones (`UNIFORM`).

Cuando un tablespace permanente se administra localmente, la gestión automática del espacio libre dentro de los segmentos está, por defecto, activada (cláusula `SEGMENT SPACE MANAGEMENT AUTO` implícita en la sentencia `SQL CREATE TABLESPACE`); estudiaremos este modo de gestión con más detalle en el capítulo Gestión de las tablas e índices. Teniendo en cuenta este modo de gestión, las extensiones deben contener, al menos, cinco bloques; en el caso de un tablespace administrado localmente con una gestión uniforme de las extensiones hay que tenerlo en cuenta en la especificación de la cláusula `SIZE` si no quiere obtener el siguiente mensaje de error:

```
ORA-03249: UNIFORM SIZE para el tablespace administrado por un espacio de
segmento AUTO debe tener al menos 5 bloques
```

El encabezado de cada archivo de datos de un tablespace administrado localmente utiliza, al menos, tres bloques (frente a uno para un tablespace administrado por el diccionario). Por tanto, el tamaño del archivo de datos debe ser, al menos, igual a tres bloques más el tamaño de una extensión (valor explícito o por defecto de la cláusula `SIZE` para un tablespace `UNIFORM`, 64 KB para un tablespace `AUTOALLOCATE`). Si el tamaño inicial del archivo de datos es superior a 64 KB más el tamaño de una extensión, se asigna un encabezado de 64 KB al archivo, lo que permite almacenar una *bitmap* más grande y, por tanto, gestionar un número más grande de extensiones.

El paquete `DBMS_SPACE_ADMIN` ofrece diferentes procedimientos que permiten diagnosticar y reparar los tablespaces administrados localmente (en caso de daño en el *bitmap*, por ejemplo), o realizar una migración del tablespace administrado por el diccionario al tablespace administrado localmente (y recíprocamente). En este último caso el resultado obtenido es mejor, creando un nuevo tablespace administrado de manera local y transfiriendo los segmentos del antiguo tablespace al nuevo; por el contrario, la operación es más larga.

4. Gestión de extensiones dentro de un tablespace gestionado localmente

En caso de una gestión automática de extensiones, Oracle utiliza un pequeño número de tamaños diferentes de

extensión (64 KB, 1 MB, 8 MB, 64 MB) e intenta juntar las extensiones del mismo tamaño, en cantidad suficiente para ocupar el espacio potencialmente utilizable para una extensión de tamaño superior (16 extensiones de 64 KB = una extensión potencial de 1 MB). Esta técnica permite limitar los riesgos de fragmentación del espacio disponible: si se elimina un segmento que contiene muchas extensiones, el espacio liberado se puede reutilizar de diferentes maneras. El tamaño de extensión elegido inicialmente para un segmento depende del tamaño inicial del segmento:

- 64 KB para un segmento de menos de 1 MB;
- 1 MB para un segmento de menos de 64 MB;
- 8 MB para un segmento de menos de 1 024 MB;
- etc.



El algoritmo utilizado por Oracle para calcular el tamaño de las extensiones en un tablespace administrado localmente, con una gestión automática de las extensiones, no está documentado. Los valores indicados aquí son valores comprobados para la creación de una tabla en un tablespace vacío.

En caso de gestión uniforme de las extensiones, todas tienen el mismo tamaño, definido por la opción `SIZE` de la cláusula `EXTENT MANAGEMENT` (1 MB por defecto).

El tamaño inicial del segmento se calcula con los valores `INITIAL`, `NEXT`, `MINEXTENTS` y `PCTINCREASE` de la cláusula `STORAGE`:

- si `MINEXTENTS` = 1 entonces `INITIAL` (es el caso el más frecuente);
- si `MINEXTENTS` = 2 entonces `INITIAL+NEXT`;
- si `MINEXTENTS` = 3 entonces `INITIAL+NEXT+NEXT*(1+PCTINCREASE/100)`;
- etc.

El valor calculado se convierte en el nuevo valor `INITIAL`, tal y como se guarda en el diccionario de datos.

A continuación, Oracle asigna una o varias extensiones (sin tener en cuenta el `MINEXTENTS` inicial) de tamaño uniforme (caso `UNIFORM`) o de tamaño determinado de manera interna (caso `AUTOALLOCATE`) para obtener un tamaño inicial igual o superior al tamaño calculado anteriormente.

Como ejemplo, supongamos que la siguiente tabla se crea en un tablespace administrado localmente con extensiones uniformes de 128 KB:

```
CREATE TABLE miembro (...)  
  TABLESPACE tbs_local_uniform  
  STORAGE (INITIAL 400K NEXT 100K PCTINCREASE 0 MINEXTENTS 2);
```

Oracle asigna $(400+100)/128 = 3,5$ redondeado al entero superior, es decir, 4 extensiones de 128 KB para la tabla.

Si la misma tabla se crea en un tablespace administrado localmente, con una gestión automática de extensiones, Oracle asigna $(400+100)/64 = 7,8$ redondeado al entero superior, es decir, 8 extensiones de 64 KB para la tabla (Oracle elige un tamaño de extensión de 64 KB porque el tamaño inicial del segmento es inferior a 1 MB).

La información que se guarda en el diccionario de datos (vista `DBA_TABLES` y similares) es la siguiente:

- Los valores `MINEXTENTS` y `MAXEXTENTS` se ignoran y se fuerzan respectivamente a 1 (incluso si se han asignado varias extensiones al segmento) y `UNLIMITED`.

- El valor calculado para el tamaño inicial se convierte en el nuevo valor INITIAL (incluso si el espacio realmente asignado es superior, teniendo en cuenta el redondeo del número de extensiones).
- Para un tablespace AUTOALLOCATE se ignoran los valores NEXT y PCTINCREASE y se ponen a NULL (es Oracle quién decide).
- Para un tablespace UNIFORM, NEXT se iguala al tamaño de las extensiones del tablespace (opción SIZE de la cláusula UNIFORM del tablespace) y PCTINCREASE se iguala a 0 (todas las extensiones tienen el mismo tamaño).

Conclusión: las opciones NEXT, PCTINCREASE, MINEXTENTS y MAXEXTENTS no son de gran utilidad para un tablespace administrado localmente.

Cuando se llena el espacio asignado inicialmente al segmento, se asignan extensiones adicionales al segmento.

Si el segmento se almacena en un tablespace UNIFORM, todas las extensiones adicionales asignadas al segmento tienen el mismo tamaño.

Si el segmento se almacena en un tablespace AUTOALLOCATE, es Oracle el que determina el tamaño de las nuevas extensiones asignadas al segmento, según un algoritmo no documentado. A simple vista, cuando se usa un tamaño inicial de 64 KB, el funcionamiento es el siguiente: mientras el segmento tenga menos de 16 extensiones, Oracle asigna extensiones de 64 KB, y después pasa a extensiones de 1 MB, hasta 64 MB, y después a extensiones de 8 MB. Por otra parte, Oracle intenta asignar extensiones del mismo tamaño consecutivamente, hasta obtener un número de extensiones consecutivas que ocupen un espacio igual al tamaño de extensión superior.

Ejemplo:


```
SQL> -- creación de un tablespace administrado localmente con
SQL> -- una gestión automática de extensiones
SQL> CREATE TABLESPACE test
  2 DATAFILE 'e:\oradata\NuestraBase\test01.dbf' SIZE 10M
  3 EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
Tablespace creado.
SQL> -- creación de tres tablas: dos "pequeñas" y una "grande"
SQL> CREATE TABLE table200k(c NUMBER)
  2 TABLESPACE test
  3 STORAGE(INITIAL 200K);
Tabla creada.

SQL> CREATE TABLE tablexk(c NUMBER) sin cláusula STORAGE
  2 TABLESPACE test;
Tabla creada.

SQL> CREATE TABLE table2M(c NUMBER)
  2 TABLESPACE test
  3 STORAGE(INITIAL 2M);
Tabla creada.
SQL> -- supervisión del almacenamiento en el tablespace
SQL> @info_almacenamiento_tablespace
BLOCK_ID  EXTENT_ID SEGMENT_NAME      BLOCKS  TAMAÑO_KB
-----
          9          0 TABLE200K          8        64
         17          1 TABLE200K          8        64
         25          2 TABLE200K          8        64
         33          3 TABLE200K          8        64
         41          0 TABLEXK          8        64
```

49	*** LIBRE ***	88	704
137	0 TABLE2M	128	1024
265	1 TABLE2M	128	1024
393	*** LIBRE ***	888	7104

Este ejemplo ilustra los siguientes puntos:

- Oracle ha elegido extensiones de 64 KB para las tablas "pequeñas" y extensiones de 1 MB para la "grande".
 - Oracle ha dejado espacio libre entre las tablas pequeñas y la grande: 704 KB, es decir, 11 extensiones de 64 KB. Este espacio está reservado a extensiones de 64 KB, lo que le permite obtener un total de 16 extensiones de 64 KB consecutivas (es decir, potencialmente una extensión de 1 MB, en caso de liberación de estas extensiones).
-  Por defecto, desde la versión 11.2, durante la creación de una tabla en un tablespace administrado localmente Oracle retrasa la creación del segmento asociado hasta el momento en que se inserta el primer registro en la tabla. En este caso, la tabla aparece correctamente en el diccionario de datos (vista `*_TABLES`), pero no el segmento ni las extensiones (vistas `*_SEGMENTS` y `*_EXTENTS`). Esta funcionalidad permite ahorrar espacio y acelerar la creación de las tablas para una aplicación que tiene varios centenares de tablas de las cuales un número importante podrían no ser alimentadas nunca. Consulte el capítulo Gestión de las tablas e índices.

5. Caso de los tablespaces SYSTEM y SYSAUX

Desde Oracle9i *release* 2 (versión 9.2) el tablespace SYSTEM se puede administrar localmente en cuyo caso se realiza, necesariamente, una gestión automática de las extensiones (`EXTENT MANAGEMENT LOCAL AUTOALLOCATE`) y una gestión manual del espacio en los segmentos (`SEGMENT SPACE MANAGEMENT MANUAL`). Por defecto, se administra por el diccionario. Crear un tablespace SYSTEM administrado localmente tiene las siguientes consecuencias (positivas):

- Todos los tablespaces se deben administrar localmente (recomendado por Oracle);
- Se debe crear un tablespace temporal predeterminado durante la creación de la base (recomendado por Oracle);
- Si la gestión automática de los segmentos de anulación está activada (recomendado por Oracle), se debe crear un tablespace de anulación durante la creación de la base de datos (recomendado por Oracle).

En la sentencia `SQL CREATE DATABASE`, la cláusula `EXTENT MANAGEMENT LOCAL` permite especificar que el tablespace SYSTEM se administre localmente:

```
CREATE DATABASE NuestraBase
...
DATAFILE 'e:\oradata\NuestraBase\system01.dbf' SIZE 700M
  AUTOEXTEND ON NEXT 10M
  EXTENT MANAGEMENT LOCAL
...
```

El tablespace SYSAUX se administra localmente de manera obligatoria, con una gestión automática de las extensiones (`EXTENT MANAGEMENT LOCAL AUTOALLOCATE`) y una gestión automática del espacio en los segmentos (`SEGMENT SPACE MANAGEMENT AUTO`); no hay nada que especificar durante la creación de la base de datos.

En el caso de actualización funcional de una base de datos, el tablespace SYSAUX se crea mediante una sentencia `SQL CREATE TABLESPACE`. Se debe utilizar la siguiente sintaxis:

```
CREATE TABLESPACE sysaux  
DATAFILE especificación_archivo  
EXTENT MANAGEMENT LOCAL  
SEGMENT SPACE MANAGEMENT AUTO;
```


Tablespace temporal

1. Rol del tablespace temporal

Cuando una consulta necesita una operación de ordenación (cláusula `ORDER BY`, por ejemplo), Oracle intenta hacer la ordenación en memoria, en la PGA del proceso de servidor que ejecuta la consulta.

Si la ordenación no se puede realizar en memoria, Oracle desacopla el conjunto de información en fracciones y ordena cada fracción individualmente, almacenando los resultados intermedios en disco en los segmentos temporales.

Un segmento temporal se puede crear en cualquier tablespace, pero no es deseable a efectos de rendimiento.

Por tanto, Oracle recomienda crear un tablespace dedicado, de tipo `TEMPORARY`, para almacenar los segmentos temporales y, preferiblemente, un tablespace temporal administrado localmente. Es posible crear un tablespace temporal administrado por el diccionario, pero se limita el rendimiento, por lo que Oracle abandona esta opción; por este motivo no se ha comentado inicialmente.

Las consultas que pueden solicitar una operación de ordenación son las siguientes:

- `SELECT ... ORDER BY;`
- `SELECT ... GROUP BY;`
- `SELECT DISTINCT ...;`
- consultas de unión (`UNION`, `INTERSECT`, `MINUS`);
- `CREATE INDEX;`
- cálculo de estadísticas;
- joins por ordenación-mezcla (*sort merge join*).

Es posible utilizar un tablespace permanente como tablespace temporal (esto es lo que pasa por defecto con el tablespace `SYSTEM`), pero no es aconsejable, fundamentalmente desde el punto de vista del rendimiento. De hecho, en un tablespace permanente, los segmentos temporales se asignan y se liberan con cada ordenación; penaliza el rendimiento y se corre el riesgo de fragmentar el espacio disponible del tablespace. En caso de usar un tablespace temporal, se crea un único segmento de ordenación para la primera ordenación y se reutiliza para las siguientes ordenaciones.

El segmento temporal se puede compartir entre varias ordenaciones (pero no las extensiones) y solo se libera durante la parada de la instancia; de esta manera, se produce una menor asignación dinámica de extensiones y el rendimiento se optimiza.



Un tablespace permanente administrado localmente no se puede utilizar como tablespace temporal; este no es el caso de un tablespace permanente, administrado por el diccionario.

Los tablespaces temporales también se utilizan para el almacenamiento de las tablas temporales, creadas por la sentencia `SQL CREATE GLOBAL TEMPORARY TABLE`.

2. Grupo de tablespaces temporales

Antes de la versión 10, una consulta solo podía utilizar un único tablespace temporal, lo que presentaba problemas

de rendimiento si la consulta se ejecutaba en paralelo. En este caso, varios procesos de servidor trataban la consulta en paralelo y cada proceso podía solicitar acceso al tablespace temporal, lo que algunas veces provocaba problemas de restricciones a nivel de entradas/salidas.

Desde la versión 10, es posible definir grupos de tablespaces temporales. En caso de ejecución de una consulta en paralelo se pueden utilizar diferentes tablespaces temporales del grupo por parte de los diferentes procesos de servidor que tratan la consulta. Realmente, esto solo tiene interés desde el punto de vista del rendimiento si los archivos de datos de los diferentes tablespaces temporales se almacenan en discos diferentes.

Se puede utilizar el nombre de un grupo de tablespaces temporales en cualquier sitio donde se use un nombre de tablespace temporal. Además, el espacio de nombres de los grupos de tablespaces temporales es el de los tablespaces; un grupo de tablespaces temporales no puede tener el mismo nombre que un tablespace.

Un grupo de tablespaces temporales no se crea o elimina explícitamente. Se crea implícitamente cuando se asigna un primer tablespace temporal al grupo y se elimina implícitamente cuando el último tablespace temporal se retira del grupo.

3. Creación de un tablespace temporal administrado localmente

La sentencia SQL **CREATE TEMPORARY TABLESPACE** permite crear un tablespace temporal administrado localmente.

Sintaxis

```
CREATE [ BIGFILE | SMALLFILE ] TEMPORARY TABLESPACE nombre
TEMPFILE especificación_archivo [,...]
[ EXTENT MANAGEMENT LOCAL ] [ UNIFORM [ SIZE valor [K|M] ] ]
[ TABLESPACE GROUP nombre_grupo ];
    - especificación_archivo
'nombre_archivo' [ SIZE valor [K|M|G|T] ] [REUSE]
[ cláusula_auto_extend ]
    - cláusula_auto_extend
AUTOEXTEND { OFF | ON [ NEXT valor [K|M|G|T] ]
[ MAXSIZE { UNLIMITED | valor [K|M|G|T] } ] }
```

Ejemplo:

```
CREATE TEMPORARY TABLESPACE tempo
TEMPFILE 'e:\oradata\NuestraBase\tempo01.dbf'
SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 1G;
```

Las opciones de la sentencia SQL **CREATE TEMPORARY TABLESPACE** tienen el mismo significado que las opciones del mismo nombre de la sentencia SQL **CREATE TABLESPACE** (consulte la sección **Tablespace permanente**).

Sin embargo, se puede observar lo siguiente:

- Un tablespace temporal administrado localmente puede ser un tablespace **BIGFILE**; en este caso se puede especificar un único archivo de datos.
- Los archivos de datos de un tablespace temporal administrado localmente se especifican mediante la palabra clave **TEMPFILE** y no **DATAFILE**.

- Un tablespace temporal administrado localmente tiene, forzosamente, una gestión uniforme de sus extensiones. Las cláusulas `EXTENT MANAGEMENT LOCAL` y `UNIFORM` son, por tanto, opcionales. Por defecto, el tamaño de las extensiones es de 1 MB; esto es suficiente en la gran mayoría de los casos. La cláusula `SIZE` se puede utilizar para especificar otro tamaño; en este caso se debe incluir la palabra clave `UNIFORM`.
- Un tablespace temporal administrado localmente utiliza, obligatoriamente, el tamaño de bloque estándar; no es posible emplear otro tamaño de bloque.
- Un tablespace temporal administrado localmente está obligatoriamente `ONLINE`.
- Las cláusulas `LOGGING`, `NOLOGGING`, `FORCE LOGGING` y `FLASHBACK` están prohibidas para un tablespace temporal administrado localmente.

La cláusula `TABLESPACE GROUP` permite asignar el tablespace temporal a un grupo; si el grupo no existe, se crea implícitamente. Por defecto, el tablespace temporal no pertenece a ningún grupo.

4. Tablespace temporal predeterminado

Un tablespace temporal solo se utiliza realmente cuando se "asigna" a los usuarios, gracias a la cláusula `TEMPORARY TABLESPACE` de las sentencias SQL `CREATE USER` y `ALTER USER` (consulte el capítulo Gestión de usuarios y sus permisos). Si esta cláusula se omite, asigna al usuario el tablespace `SYSTEM` como tablespace temporal, lo cual no es conveniente para el rendimiento.

Para resolver este problema y facilitar la gestión de los usuarios es posible definir un tablespace temporal predeterminado durante la creación de la base de datos o posteriormente. Esta técnica no impide utilizar otros tablespaces temporales asignados específicamente a los usuarios para necesidades particulares.

Un tablespace `SYSTEM` administrado localmente no se puede utilizar como tablespace temporal predeterminado, de ahí la necesidad, en este caso, de crear y definir un tablespace temporal predeterminado durante la creación de la base.

Si el tablespace `SYSTEM` se administra por el diccionario y se utiliza como tablespace temporal predeterminado, se escribe un mensaje en el archivo de alertas de la instancia.

Para crear y definir un tablespace temporal predeterminado durante la creación de la base de datos debe utilizar la cláusula `DEFAULT TEMPORARY TABLESPACE` de la sentencia SQL `CREATE DATABASE`.

Sintaxis

```
[ BIGFILE | SMALLFILE ] DEFAULT TEMPORARY TABLESPACE nombre
TEMPFILE especificación_archivo [,...]
[ EXTENT MANAGEMENT LOCAL ] [ UNIFORM [ SIZE valor [K|M] ] ]
```

Ejemplo:

```
DEFAULT TEMPORARY TABLESPACE temp
TEMPFILE 'e:\oradata\NuestraBase\temp01.dbf' SIZE 100M
AUTOEXTEND ON NEXT 10M MAXSIZE 1024M
```



Esta cláusula debe estar presente si el tablespace `SYSTEM` se administra localmente.

La sintaxis es la misma que la de la sentencia SQL `CREATE TEMPORARY TABLESPACE`. Se crea un tablespace

temporal administrado localmente según la especificación y definido como tablespace temporal predeterminado. Observe que el tablespace temporal predeterminado creado de esta manera forzosamente se administra localmente (Oracle lo aconseja).

El tablespace temporal creado de esta manera se tiene en cuenta durante la creación de la base de datos y, por tanto, se asigna como tablespace temporal a los usuarios creados durante esta operación (fundamentalmente SYS y SYSTEM).

Para crear y definir un tablespace temporal predeterminado después de la creación de la base de datos debe:

- crear un tablespace temporal mediante la sentencia SQL `CREATE TEMPORARY TABLESPACE` presentada anteriormente;
- definirlo como tablespace temporal predeterminado utilizando la cláusula `DEFAULT TEMPORARY TABLESPACE` de la sentencia SQL `ALTER DATABASE`.

Sintaxis

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE nombre;
```

nombre debe hacer referenciar a un tablespace temporal o un grupo de tablespaces temporales que existan previamente.

Cuando se ejecuta esta sentencia SQL, a todos los usuarios que tenían el antiguo tablespace temporal predeterminado como tablespace temporal se les asigna automáticamente el nuevo.

Para conocer el nombre del tablespace temporal predeterminado puede consultar la vista `DATABASE_PROPERTIES`, en particular la propiedad `DEFAULT_TEMP_TABLESPACE`:

```
SQL> SELECT property_value FROM database_properties
      2 WHERE property_name = 'DEFAULT_TEMP_TABLESPACE';
PROPERTY_VALUE
-----
TEMP
```

5. Administración de los tablespaces temporales administrados localmente

La administración de un tablespace temporal administrado localmente se realiza mediante las sentencias SQL presentadas para los tablespaces permanentes, con algunas restricciones: `ALTER TABLESPACE`, `ALTER DATABASE` para la gestión de los archivos de datos y `DROP TABLESPACE`.

Los archivos de datos de un tablespace temporal administrado localmente son particulares; Oracle los llama "archivos de datos temporales".

Las diferencias con un archivo de datos normal son las siguientes:

- Siempre están en modo `NOLOGGING`;
- No se pueden desactivar;
- No pueden pasar a modo de solo lectura.

Para los archivos de datos temporales de los tablespaces temporales, administrados localmente en modo

NOLOGGING, las modificaciones no se guardan en los archivos de traza (lo cual resulta interesante en términos de rendimiento). Por el contrario, en caso de pérdida o daño de alguno de estos archivos, no es posible la restauración (RECOVER). Esta situación no es muy grave, porque un tablespace temporal no contiene datos permanentes; en caso de producirse algún problema basta con eliminar el tablespace o simplemente el archivo de datos, y después volver a crearlo.

Los archivos de datos temporales se administran mediante las sentencias SQL ALTER TABLESPACE y ALTER DATABASE, sustituyendo la palabra clave DATAFILE por la palabra clave TEMPFILE.

Se permiten las siguientes operaciones:

- Añadir un archivo de datos temporal a un tablespace temporal, administrado localmente:

```
ALTER TABLESPACE nombre_tablespace
ADD TEMPFILE especificación_archivo;
```

- Modificación del tamaño de un archivo de datos temporal:

- Cualquier tipo de tablespace:

```
ALTER DATABASE TEMPFILE 'nombre_completo' [... ]
RESIZE valor [K|M|G|T];
```

- Tablespace BIGFILE solo:

```
ALTER TABLESPACE nombre_tablespace_bigfile
RESIZE valor [K|M|G|T];
```

- Modificación de la cláusula AUTOEXTEND de un archivo de datos temporal:

- Cualquier tipo de tablespace:

```
ALTER DATABASE TEMPFILE 'nombre_completo' [... ]
cláusula_auto_extensión;
```

- Solo para tablespaces BIGFILE:

```
ALTER TABLESPACE nombre_tablespace_bigfile cláusula_auto_extensión;
```

Por el contrario, un tablespace temporal administrado localmente no puede pasar a modo OFFLINE.

De la misma manera, un archivo de datos temporal no se puede renombrar utilizando la sentencia SQL ALTER TABLESPACE ... RENAME DATAFILE (porque no se puede pasar a modo OFFLINE). Ya no es posible moverlo o renombrarlo por línea de comandos mediante la sentencia SQL ALTER DATABASE MOVE. Por el contrario, es posible renombrar un archivo de datos temporal con un ALTER DATABASE RENAME FILE (siempre y cuando la base de datos esté montada). Para renombrar un archivo de datos temporal de una base de datos abierta y, por tanto, también para moverlo, hay que eliminarlo y volver a crearlo.

Ejemplo:

```
-- eliminar el archivo de datos temporal
SQL> ALTER DATABASE
2     TEMPFILE 'e:\oradata\NuestraBase\temp01.dbf' DROP
3     INCLUDING DATAFILES;
Base de datos modificada.
```

```
-- volver a crearlo
SQL> ALTER TABLESPACE temp ADD
  2     TEMPFILE 'f:\oradata\NuestraBase\temp01.dbf' SIZE 100M
  3     AUTOEXTEND ON NEXT 10M MAXSIZE 1G;
Tablespace modificado.
```

Observe el uso de la opción INCLUDING DATAFILES, que permite eliminar físicamente el archivo.

También se puede eliminar un archivo de datos temporal con una sentencia SQL ALTER TABLESPACE ... DROP TEMPFILE.

Por otra parte, desde la versión 11 es posible reducir un tablespace temporal administrado localmente.

Sintaxis

```
ALTER TABLESPACE nombre SHRINK SPACE [ KEEP tamaño [K|M|G] ];
ALTER TABLESPACE nombre SHRINK TEMPFILE 'nombre_completo' | número_archivo
[ KEEP tamaño [K|M|G] ];
```

Este comando es interesante para liberar el espacio utilizado, por ejemplo, por una operación de ordenación voluminosa que acaba de terminar. La primera sintaxis permite reducir todos los archivos de datos temporales del tablespace, mientras que la segunda trabaja sobre un archivo específico. La cláusula KEEP define un tamaño mínimo que se debe conservar en el tablespace o en el archivo; si esta cláusula no está, Oracle intenta liberar el máximo espacio posible. Si la cláusula KEEP es demasiado baja, se obtiene un error:

```
ORA-03214: El tamaño indicado del archivo es inferior al mínimo necesario
```

Curiosamente, este error también se obtiene si la cláusula KEEP no está y Oracle intenta reducir el archivo a un tamaño inferior al mínimo necesario.

Para terminar, el tablespace temporal predeterminado no se puede eliminar. Sucede lo mismo para todo tablespace temporal que pertenezca a un grupo de tablespaces temporales utilizado como tablespace temporal predeterminado. Para ubicar un tablespace temporal en un grupo de tablespaces temporales, cambiarlo de grupo o eliminarlo, se puede utilizar la cláusula TABLESPACE GROUP de la sentencia SQL ALTER TABLESPACE.

Sintaxis:

```
ALTER TABLESPACE nombre_tablespace TABLESPACE GROUP nombre_grupo | '';
```

Puede utilizar una cadena vacía para no asignar el tablespace a ningún grupo. Durante la asignación de un tablespace temporal a un grupo, el grupo se crea implícitamente si no existe. Cuando un tablespace temporal se elimina de un grupo, el grupo se elimina implícitamente si no contiene más tablespaces temporales.

No puede retirar el último tablespace temporal de un grupo si este grupo se utiliza como tablespace temporal predeterminado.

Conclusiones

1. Ventajas de los tablespaces administrados localmente

Los tablespaces administrados localmente presentan numerosas ventajas:

- menos SQL recursivo, incluso de gestión recursiva del espacio, relacionado con la actualización del diccionario de datos;
- extensiones adyacentes libres, identificadas automáticamente, lo que elimina las operaciones de fusión (*coalesce*) de las extensiones adyacentes libres;
- limitación, incluso desaparición de los problemas de fragmentación del espacio disponible.

Con un tablespace administrado por el diccionario, cuando la instancia asigna o libera una extensión, ésta debe leer y después actualizar el diccionario de datos usando sentencias SQL `SELECT`, `INSERT`, `UPDATE` o `DELETE`; estas diferentes sentencias se llaman "SQL recursivo" y son susceptibles de utilizar espacio de anulación en el segmento de anulación `SYSTEM`. Durante la actualización del diccionario, Oracle puede sufrir falta de espacio en la tabla del diccionario o en el segmento de anulación: el resultado es una asignación recursiva de espacio, lo que penaliza el rendimiento. Estos problemas desaparecen, en gran medida, con los tablespaces administrados localmente.

En un tablespace administrado por el diccionario, cuando una extensión se libera, Oracle no comprueba inmediatamente si es adyacente a una extensión ya libre. Más tarde, como tarea en segundo plano o si se busca una extensión grande, el proceso SMON fusiona extensiones adyacentes libres del tablespace: es la operación de *coalesce*. Esta operación puede consumir mucho tiempo si hay un gran número de extensiones libres en el tablespace. En un tablespace administrado localmente, esta operación no es necesaria, porque las extensiones adyacentes libres se identifican automáticamente en el *bitmap* (ceros consecutivos).

Uno de los objetivos de los tablespaces administrados localmente es racionalizar el uso del espacio en los tablespaces y evitar el fenómeno de fragmentación del espacio disponible. Esta fragmentación del espacio disponible puede aparecer después de una fuerte actividad de asignación/liberación de extensiones: puede haber mucho espacio disponible en el tablespace, pero en forma de una multitud de pequeñas extensiones no adyacentes. El riesgo de fragmentación desaparece completamente en un tablespace administrado localmente, con una gestión uniforme de extensiones: todas las extensiones asignadas en el tablespace tienen, necesariamente, el mismo tamaño y una extensión liberada se podrá reutilizar obligatoriamente.

Cuando Oracle administra las extensiones, la instancia utiliza un algoritmo que intenta reducir el riesgo de fragmentación usando, por una parte, un pequeño número de tamaños diferentes de extensiones y, por otro lado, asignando consecutivamente extensiones que, agrupadas, pueden formar una extensión de tamaño superior.

2. Recomendaciones

Oracle recomienda utilizar tablespaces administrados localmente:

- para el tablespace `SYSTEM`;
- para el tablespace temporal, creándolo durante la creación de la base de datos para tener más de un tablespace temporal predeterminado;
- para los segmentos de anulación (capítulo Gestión de la información de anulación);
- para los tablespaces de las tablas e índices.

¿Qué modo de gestión conviene elegir para las extensiones de los tablespaces de tablas e índices? Es preferible

configurar una gestión automática de las extensiones si no tiene una buena visión de las necesidades en términos de espacio y no desea controlar la asignación de las extensiones. Elija una gestión uniforme de extensiones si desea controlar la asignación de las extensiones y si tiene una buena visión de las necesidades en términos de espacio.

Los tablespaces administrados localmente con una gestión automática de extensiones resultan interesantes cuando el volumen de los segmentos es completamente desconocido; permiten realizar una mejor gestión del espacio que los tablespaces administrados por el diccionario. Si se conocen las necesidades, utilizar tablespaces administrados localmente con una gestión uniforme de extensiones no es necesariamente algo inmediato, en particular para determinar el tamaño correcto de la extensión; en este caso, sin duda, hay que emplear varios tablespaces para separar los segmentos en grandes categorías.

Ejemplo:

- los "pequeños" (por ejemplo, entre 0 y 2 MB): un tablespace con extensiones de 64 KB;
- los "medianos" (por ejemplo, entre 2 MB y 64 MB): un tablespace con extensiones de 2 MB;
- los "grandes" (por ejemplo, más allá de 64 MB): un tablespace con extensiones de 64 MB (y, sin duda, varios tablespaces).

En el capítulo Gestión de las tablas e índices veremos cómo estimar el tamaño de los segmentos en un periodo determinado.

Encontrar información de los tablespaces y archivos de datos

1. Tablespaces y archivos de datos

Varias vistas del diccionario de datos permiten obtener información de los tablespaces y archivos de datos:

- DBA_TABLESPACES o V\$TABLESPACE: información de los tablespaces;
- DBA_DATA_FILES o V\$DATAFILE: información de los archivos de datos (salvo los de los tablespaces temporales administrados localmente);
- DBA_TEMP_FILES o V\$TEMPFILE: información de los archivos de datos de los tablespaces temporales administrados localmente;
- DBA_TABLESPACE_GROUPS: información de los grupos de tablespaces temporales;
- DATABASE_PROPERTIES: propiedades de la base de datos: el tablespace temporal predeterminado, el tablespace permanente predeterminado y el tipo de tablespace por defecto (BIGFILE o SMALLFILE).

Las columnas interesantes de las diferentes vistas se presentan a continuación.

DBA_TABLESPACES

TABLESPACE_NAME	Nombre del tablespace.
CONTENTS	Tipo de tablespace (PERMANENT o TEMPORARY o UNDO).
EXTENT_MANAGEMENT	DICTIONARY: el tablespace se administra por el diccionario. LOCAL: el tablespace se administra localmente.
ALLOCATION_TYPE	USER: gestión de extensiones por "el usuario" (tablespace administrado por el diccionario). SYSTEM: gestión automática de extensiones (tablespace administrado localmente). UNIFORM: gestión uniforme de extensiones (tablespace administrado localmente).
STATUS	Estado del tablespace (ONLINE, OFFLINE o READ ONLY).
BLOCK_SIZE	Tamaño de bloque del tablespace.
LOGGING	Modo de traza por defecto (LOGGING o NOLOGGING).
FORCE_LOGGING	Indica si el tablespace está en FORCE LOGGING (YES o NO).
SEGMENT_SPACE_MANAGEMENT	Indica si el espacio libre en los segmentos se administra manual (MANUAL) o automáticamente (AUTO).
BIGFILE	Indica si el tablespace es un tablespace BIGFILE (YES o NO).

Ejemplo:

SQL> SELECT tablespace_name,contents,extent_management,						
2 asignation_type,bigfile,block_size,status						
3 FROM dba_tablespaces;						
TABLESPACE_NAME	CONTENTS	EXTENT_MAN	ALLOCATIO	BIG	BLOCK_SIZE	STATUS
-----	-----	-----	-----	---	-----	-----
SYSTEM	PERMANENT	LOCAL	SYSTEM	NO	8192	ONLINE
SYS_AUX	PERMANENT	LOCAL	SYSTEM	NO	8192	ONLINE
UNDOTBS1	UNDO	LOCAL	SYSTEM	NO	8192	ONLINE
TEMP	TEMPORARY	LOCAL	UNIFORM	NO	8192	ONLINE

USERS	PERMANENT	LOCAL	SYSTEM	NO	8192	ONLINE
DATA	PERMANENT	LOCAL	UNIFORM	NO	8192	ONLINE
INDX	PERMANENT	LOCAL	SYSTEM	NO	8192	ONLINE
SOY_GRANDE	PERMANENT	LOCAL	SYSTEM	YES	8192	ONLINE
DEFTBS	PERMANENT	LOCAL	SYSTEM	NO	8192	ONLINE

DBA_DATA_FILES y DBA_TEMP_FILES

FILE_NAME	Nombre del archivo de datos (ruta completa).
FILE_ID	Identificador del archivo de datos.
TABLESPACE_NAME	Nombre del tablespace al que pertenece el archivo de datos.
BYTES	Tamaño del archivo en bytes.
BLOCKS	Tamaño del archivo en bloques Oracle.
STATUS	Estado del archivo (INVALID o AVAILABLE).
RELATIVE_FNO	Número relativo del archivo respecto al tablespace.
AUTOEXTENSIBLE	Indicador de auto-extensibilidad (YES o NO).
MAXBYTES	Tamaño máximo del archivo en bytes.
MAXBLOCKS	Tamaño máximo del archivo en bloques Oracle.
INCREMENT_BY	Tamaño del incremento de la auto-extensión en bloques Oracle.
USER_BYTES	Tamaño útil del archivo en bytes (generalmente el tamaño del archivo menos los bloques de encabezado).
USER_BLOCKS	Tamaño útil del archivo en bloques Oracle (generalmente el tamaño del archivo menos los bloques de encabezado).

Ejemplo:

```
SQL> SELECT tablespace_name,file_name,status,autoextensible,
2         blocks,user_blocks,maxblocks
3 FROM (
4         SELECT * FROM dba_data_files
5         UNION ALL SELECT * FROM dba_temp_files);
```

TABLESPACE_NAME	FILE_NAME	STATUS
SYSTEM	F:\ORADATA\NUESTRABASE\SYSTEM01.DBF	AVAILABLE
YES	89600	89472 4194302
SYSAUX	E:\ORADATA\NUESTRABASE\SYSAUX01.DBF	AVAILABLE
YES	70400	70272 4194302
UNDOTBS1	E:\ORADATA\NUESTRABASE\UNDOTBS01.DBF	AVAILABLE
YES	49920	49792 4194302
USERS	E:\ORADATA\NUESTRABASE\USERS01.DBF	AVAILABLE
YES	640	512 4194302
DATA	F:\ORADATA\NUESTRABASE\DATA02.DBF	AVAILABLE
YES	25600	24320 64000

DATA	F:\ORADATA\NUESTRABASE\DATA01.DBF			AVAILABLE
YES	64000	62720	102400	
INDX	E:\ORADATA\NUESTRABASE\INDX01.DBF			AVAILABLE
YES	64000	63872	102400	
SOY_GRANDE	E:\ORADATA\NUESTRABASE\SOY_GRANDE.DBF			AVAILABLE
NO	131072	122380	0	
DEFTBS	F:\ORADATA\NUESTRABASE\DEFTBS01.DBF			AVAILABLE
YES	1280	1152	64000	
TEMP	F:\ORADATA\NUESTRABASE\TEMP01.DBF			ONLINE
YES	12800	12672	131072	

V\$TABLESPACE

TS#	Número del tablespace.
NAME	Nombre del tablespace.
BIGFILE	Indica si el tablespace es un tablespace BIGFILE (YES o NO).

V\$DATAFILE y V\$TEMPFILE

TS#	Número del tablespace al que pertenece el archivo de datos.
FILE#	Identificador del archivo de datos.
NAME	Nombre del archivo de datos (ruta completa).
STATUS	Estado del archivo de datos (OFFLINE, ONLINE, SYSTEM o RECOVER).
ENABLED	Disponibilidad del archivo de datos (DISABLED, READ ONLY, READ WRITE).
BYTES	Tamaño del archivo en bytes.
CREATE_BYTES	Tamaño en bytes del archivo en el momento de su creación.
BLOCKS	Tamaño del archivo en bloques Oracle.
BLOCK_SIZE	Tamaño de bloque del archivo de datos.
CREATION_TIME	Fecha y hora de creación del archivo.
CHECKPOINT_CHANGE#	Número SCN del último punto de control (no existe en V\$TEMPFILE).
CHECKPOINT_TIME	Fecha y hora del último punto de control (no existe en V\$TEMPFILE).

Ejemplo:

```
SQL> SELECT file#,name,status,enabled,checkpoint_change# ckpt_change#
2 FROM v$datafile;
```

FILE#	NAME	STATUS	ENABLED	CKPT_CHANGE#
1	F:\ORADATA\NUESTRABASE\SYSTEM01.DBF	SYSTEM	READ WRITE	668687
2	E:\ORADATA\NUESTRABASE\SYS_AUX01.DBF	ONLINE	READ WRITE	668687
3	E:\ORADATA\NUESTRABASE\UNDOTBS01.DBF	ONLINE	READ WRITE	668687
4	E:\ORADATA\NUESTRABASE\USERS01.DBF	ONLINE	READ WRITE	668687

5	F:\ORADATA\NUESTRABASE\DATA01.DBF	ONLINE	READ	WRITE	668687
6	E:\ORADATA\NUESTRABASE\INDX01.DBF	ONLINE	READ	WRITE	668687
7	E:\ORADATA\NUESTRABASE\SOY_GRANDE.DBF	ONLINE	READ	WRITE	668687
8	F:\ORADATA\NUESTRABASE\DEFTBS01.DBF	ONLINE	READ	WRITE	668687
9	F:\ORADATA\NUESTRABASE\DATA02.DBF	ONLINE	READ	WRITE	668687

DBA_TABLESPACE_GROUPS

GROUP_NAME	Nombre del grupo.
TABLESPACE_NAME	Nombre del tablespace.

DATABASE_PROPERTIES

PROPERTY_NAME	Nombre de la propiedad: DEFAULT_TBS_TYPE: tipo de tablespace por defecto (SMALLFILE o BIGFILE). DEFAULT_TEMP_TABLESPACE: tablespace temporal predeterminado (puede ser un grupo de tablespaces temporales). DEFAULT_PERMANENT_TABLESPACE: tablespace permanente predeterminado.
PROPERTY_VALUE	Nombre del tablespace.

Ejemplo:

```
SQL> SELECT property_name,property_value
2  FROM database_properties
3  WHERE property_name IN ('DEFAULT_TEMP_TABLESPACE',
4                          'DEFAULT_PERMANENT_TABLESPACE',
5                          'DEFAULT_TBS_TYPE');
PROPERTY_NAME          PROPERTY_VALUE
-----
DEFAULT_TEMP_TABLESPACE    TEMP
DEFAULT_PERMANENT_TABLESPACE  DEFTBS
DEFAULT_TBS_TYPE           SMALLFILE
```

2. Supervisión del almacenamiento en los tablespaces

Varias vistas del diccionario de datos permiten obtener información acerca del almacenamiento dentro de los tablespaces:

- DBA_FREE_SPACE: información del espacio disponible dentro de un tablespace;
- DBA_SEGMENTS: información de los segmentos asignados dentro de un tablespace;
- DBA_EXTENTS: información de las extensiones asignadas dentro de un tablespace;
- V\$SORT_SEGMENT: supervisión del almacenamiento de los segmentos temporales;
- V\$SYSAUX_OCCUPANTS: información de los componentes que utilizan espacio en el tablespace SYSAUX.

Las columnas interesantes de las diferentes vistas se presentan a continuación.

DBA_SEGMENTS

OWNER	Nombre del propietario del segmento.
SEGMENT_NAME	Nombre del segmento.
SEGMENT_TYPE	Tipo del segmento (TABLE, INDEX, ROLLBACK, TYPE2UNDO, etc.).
TABLESPACE_NAME	Nombre del tablespace que contiene el segmento.
BYTES	Tamaño del segmento en bytes.
BLOCKS	Tamaño del segmento en bloques Oracle.
EXTENTS	Número de extensiones asignadas al segmento.
INITIAL_EXTENT	Tamaño inicial del segmento.

Ejemplo:

```
SQL> SELECT segment_name,segment_type,
2         initial_extent/1024 initial_ko,blocks,extents
3 FROM dba_segments WHERE tablespace_name='TEST';
```

SEGMENT_NAME	SEGMENT_TYPE	INITIAL_KB	BLOCKS	EXTENTS
TABLE2M	TABLE	2048	256	2
TABLE200K	TABLE	200	32	4

DBA_FREE_SPACE

TABLESPACE_NAME	Nombre del tablespace que contiene la extensión libre.
FILE_ID	Identificador del archivo de datos que contiene la extensión libre.
BLOCK_ID	Número del primer bloque de la extensión libre.
BYTES	Tamaño de la extensión libre en bytes.
BLOCKS	Tamaño de la extensión libre en bloques Oracle.



Un tablespace que no tiene extensiones libres no tiene línea en DBA_FREE_SPACE.

DBA_EXTENTS

OWNER	Nombre del propietario del segmento al que pertenece la extensión.
SEGMENT_NAME	Nombre del segmento al que pertenece la extensión.
SEGMENT_TYPE	Tipo de segmento (TABLE, INDEX, ROLLBACK, TYPE2UNDO, etc.).
TABLESPACE_NAME	Nombre del tablespace que contiene la extensión.
EXTENT_ID	Número de la extensión (0 para la primera).
FILE_ID	Identificador del archivo de datos que contiene la extensión.
BLOCK_ID	Número del primer bloque de la extensión.
BYTES	Tamaño de la extensión en bytes.
BLOCKS	Tamaño de la extensión en bloques Oracle.

Adicionalmente, varias vistas tienen una columna TABLESPACE que indica el nombre del tablespace de almacenamiento, por ejemplo DBA_INDEXES y DBA_TABLES.

Ejemplo:

```
SQL> SELECT block_id,extent_id,segment_name,
2         blocks,bytes/1024 tamaño_kb
3 FROM dba_extents WHERE tablespace_name='TEST'
4 UNION
5 SELECT block_id,NULL,'*** LIBRE ***',
6         blocks,bytes/1024 size_kb
7 FROM dba_free_space WHERE tablespace_name='TEST';
```

BLOCK_ID	EXTENT_ID	SEGMENT_NAME	BLOCKS	TAMAÑO_KB
9	0	TABLE200K	8	64
17	1	TABLE200K	8	64
25	2	TABLE200K	8	64
33	3	TABLE200K	8	64
41		*** LIBRE ***	96	768
137	0	TABLE2M	128	1024
265	1	TABLE2M	128	1024
393		*** LIBRE ***	888	7104

V\$SORT_SEGMENT

TABLESPACE_NAME	Nombre del tablespace.
EXTENT_SIZE	Tamaño de las extensiones.
TOTAL_EXTENTS	Número total de extensiones en el segmento.
TOTAL_BLOCKS	Número total de bloques Oracle en el segmento.
USED_EXTENTS	Número de extensiones asignadas actualmente a las ordenaciones activas.
USED_BLOCKS	Número total de bloques Oracle asignados actualmente a las ordenaciones activas.
MAX_USED_SIZE	Número máximo de extensiones utilizadas por todas las ordenaciones (al mismo tiempo).
MAX_USED_BLOCKS	Número máximo de bloques utilizados por todas las ordenaciones (al mismo tiempo).
MAX_SORT_SIZE	Número máximo de extensiones utilizadas por una operación de ordenación (la ordenación más grande).
MAX_SORT_BLOCKS	Número máximo de bloques utilizados por una operación de ordenación (la ordenación más grande).

Ejemplo:

```
SQL> SELECT tablespace_name,total_blocks,used_blocks,
2         max_used_blocks,max_sort_blocks
3 FROM v$sort_segment;
```

TABLESPACE_NAME	TOTAL_BLOCKS	USED_BLOCKS	MAX_USED_BLOCKS	MAX_SORT_BLOCKS
TEMP	256	0	768	128

También existe una vista, V\$TEMPSEG_USAGE, que puede resultar útil para identificar las sesiones (y las consultas) que utilizan actualmente espacio temporal.

Ejemplo:

```
SQL> SELECT t.username,t.tablespace,t.segtype,t.extents,t.blocks,
2          s.sql_text
3 FROM v$tempseg_usage t,v$sql s
4 WHERE t.sql_id = s.sql_id;
USERNAME      TABLESPACE SEGTYPE      EXTENTS      BLOCKS
-----
SQL_TEXT
-----
OHEU          TEMP        SORT          2            256
SELECT * FROM t ORDER BY c1,c2,c3
```

V\$SYS_AUX_OCCUPANTS

OCCUPANT_NAME	Nombre del componente.
OCCUPANT_DESC	Descripción del componente.
SCHEMA_NAME	Nombre del esquema en el que se almacena el componente.
MOVE_PROCEDURE	Nombre del procedimiento que permiten mover el componente a otro tablespace.
MOVE_PROCEDURE_DESC	Descripción del procedimiento de desplazamiento.
SPACE_USAGE_KBYTES	Espacio utilizado actualmente por el componente (en KB).

Ejemplo:

```
SQL> SELECT occupant_desc,space_usage_kbytes
2 FROM v$sysaux_occupants WHERE space_usage_kbytes <> 0;
OCCUPANT_DESC                                SPACE_USAGE_KBYTES
-----
LogMiner                                     14080
Logical Standby                             1536
Transaction Layer - SCN to TIME mapping      384
PL/SQL Identifier Collection                 1600
Oracle Streams                             1024
XDB                                          64384
Analytical Workspace Object Table            1984
OLAP API History Tables                     1984
Server Manageability - Automatic Workload Repository 32000
Server Manageability - Advisor Framework      8576
Server Manageability - Optimizer Statistics History 13312
Server Manageability - Other Components      46912
Workspace Manager                           7296
Enterprise Manager Monitoring User            512
SQL Management Base Schema                  2496
Automated Maintenance Tasks                  320
Unified Job Scheduler                         768
```

En una instalación básica, sin opciones particulares, hay dos componentes que utilizan un espacio importante:

- el repositorio de las funcionalidades de gestión automática de la base de datos (componentes `Server Manageability - *`);
- Oracle XML DB (XDB).

Si un componente utiliza mucho espacio en el tablespace `SYSAUX`, es posible pensar en moverlo a otro tablespace. Esta operación se realiza generalmente con ayuda de un procedimiento almacenado de algún paquete, en el que las referencias se obtienen de la columna `MOVE_PROCEDURE` de la vista `V$SYSAUX_OCCUPANTS`. Algunos componentes no se pueden mover (por ejemplo, los componentes relativos a la gestión automática de la base de datos).

Utilizar EM Express

- En EM Express, seleccione el elemento **Tablespaces** del menú **Almacenamiento** para acceder a la página de gestión de los tablespaces y archivos de datos:

Nombre	Tamaño	Espacio Libre	Usado (%)	Ampli...	Tam...	Estado	Tipo	Nombre de Tablespace	Directorio
DATA	400MB	298MB	1,5	✓	Limitado	✓			C:\APP\ORACLE\PRODUCT...
DATA01.DBF	200MB	199MB	1,5	✓	Limitado	✓			C:\APP\ORACLE\PRODUCT...
DATA02.DBF	200MB	199MB	1,5	✓	Limitado	✓			C:\APP\ORACLE\PRODUCT...
EXAMPLE	1GB	42MB	96,7	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
EXAMPLE01.DBF	1GB	42MB	96,7	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
SYSAUX	600MB	25MB	95,8	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
SYSAUX01.DBF	600MB	25MB	95,8	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
SYSTEM	780MB	3MB	99,7	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
SYSTEM01.DBF	780MB	3MB	99,7	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
TEMP	60MB	60MB		✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
TEMP01.DBF	60MB	60MB		✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
UNDOTBS1	649MB	336MB	47,9	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
UNDOTBS01.DBF	649MB	336MB	47,9	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
USERS	5MB	3MB	33,8	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...
USERS01.DBF	5MB	3MB	33,8	✓	Limitado	✓			C:\APP\ORACLE\ORADATA\...

Desde esta página puede realizar diversas acciones:

- sobre los tablespaces:



- crear un tablespace (menú o botón **Crear**);
 - eliminar un tablespace (menú o botón **Borrar**);
 - añadir un archivo de datos al tablespace (menú o botón **Agregar Archivo de Datos**);
 - modificar el estado (ONLINE / OFFLINE, READ ONLY / READ WRITE) del tablespace (menú **Definir Estado**);
 - definir el tablespace como tablespace por defecto (menú **Definir como Valor por Defecto**).
- sobre los archivos de datos:



- eliminar un archivo de datos (menú o botón **Borrar**);
- modificar el estado (ONLINE / OFFLINE) de un archivo de datos (menú **Definir Estado**);
- modificar la extensión automática de un archivo de datos (menú **Editar Ampliación Automática**);
- modificar el tamaño de un archivo de datos (menú **Cambiar Tamaño**).
- sobre los grupos de tablespaces temporales:



- añadir un tablespace temporal a un grupo (menú **Grupos de tablespaces - Definir Grupo**);
- eliminar un tablespace temporal de un grupo (menú **Grupos de Tablespaces - Eliminar del Grupo**).

La creación permanente de un tablespace se realiza en cinco etapas:

1. definición de las características generales del tablespace (nombre, tipo, estado):

Crear Tablespace

General Agregar Archivos de Datos Espacio Preferencias de Log Segmentos

Nombre *

Tipo de Tablespace ☒ Permanente ☐ Temporal ☐ Deshacer

Definir como Valor por Defecto ☐

Archivo Grande ☒ Archivo Pequeño ☐ Archivo Grande

Estado ☒ En Línea ☐ Fuera de Línea

2. definición de los archivos de datos del tablespace (ubicación, tamaño, ampliación automática):

Crear Tablespace

General **Agregar Archivos de Datos** Espacio Preferencias de Log Segmentos

Uso de Archivos Gestionados por Oracle ☐

Archivos de Datos *

Nombre de Archivo

Tamaño de Archivo *

Reutilizar Archivo Existente ☐

Ampliación Automática ☒

Incremento

Tamaño Máximo de Archivo

3. definición del tamaño del bloque y del modo de gestión (automático o uniforme) de las extensiones del tablespace:

Crear Tablespace

General Agregar Archivos de Datos **Espacio** Preferencias de Log Segmentos

Tamaño del Bloque: Base de Datos por Defecto (8KB)

Asignación de Extensión: ☒ Automático ☐ Uniforme

Tamaño de Extensión:

Mostrar SQL Aceptar Cancelar

4. definición de las propiedades de traza del tablespace:

Crear Tablespace

General Agregar Archivos de Datos Espacio **Preferencias de Log** Segmentos

Preferencias de Log: ☒ Registro ☐ Sin Registro

Registro Forzoso: ☐

Mostrar SQL Aceptar Cancelar

5. definición de las propiedades de almacenamiento de los segmentos en el tablespace (gestión automática o manual del espacio en los segmentos y compresión):

Crear Tablespace

General Agregar Archivos de Datos Espacio Preferencias de Log **Segmentos**


Gestión de Espacio de Segmento: ☒ Automático ☐ Manual

Compresión: ☒ Ninguno ☐ Básico ☐ OLTP

Mostrar SQL Aceptar Cancelar

En caso de creación de un tablespace temporal, solo se cubren las tres primeras etapas (**General**, **Agregar Archivos de Datos** y **Espacio**).

Problemas habituales y soluciones

 Esto no es más que el principio. Veremos otros posibles problemas, relativos al almacenamiento en un tablespace, en el capítulo Gestión de las tablas e índices (almacenamiento de las tablas e índices).

ORA-01652: imposible ampliar el segmento temporal de N en el tablespace X

Explicación

El segmento temporal no llega a extenderse (durante una operación de ordenación, por ejemplo).

Causa(s)

El segmento temporal no llega a extenderse porque el tablespace en el que se almacena no tiene suficiente espacio disponible y no se puede ampliar.

Acción(es)

Hay que aumentar el espacio disponible en el tablespace:

- asignándole un nuevo archivo de datos (`ALTER TABLESPACE... ADD TEMPFILE ...`);
- aumentando el tamaño de algún archivo de datos del tablespace (`ALTER DATABASE TEMPFILE ... RESIZE ...`);
- autorizando a un archivo de datos del tablespace a ampliarse automáticamente (`ALTER DATABASE TEMPFILE ... AUTOEXTEND ON ...`).

Si es necesario se puede emplear la vista `V$TEMPSEG_USAGE` para supervisar, en tiempo real, las operaciones que utilizan el espacio temporal.

Este problema se puede producir en cualquier consulta que solicite una ordenación.

ORA-01630: número máximo de conjuntos de bloques contiguo (N) esperado en segmento temp, tablespace X

Explicación

El segmento temporal no llega a ampliarse (durante una operación de ordenación, por ejemplo).

Causa(s)

El segmento temporal no llega a ampliarse porque alcanza su número máximo de extensiones definido por el `MAXEXTENTS` de la cláusula `DEFAULT STORAGE` del tablespace en el que se almacena. Esto solo se puede producir cuando el tablespace utilizado para almacenar los segmentos temporales es permanente (`SYSTEM`, por ejemplo).

Acción(es)

Consultar el valor de la columna `CONTENTS` de la vista `DBA_TABLESPACES` para saber si existe un tablespace temporal en la base de datos. Si la respuesta es afirmativa, asignarlo a los usuarios (capítulo Gestión de usuarios y sus permisos) y/o definirlo como tablespace temporal predeterminado (`ALTER DATABASE DEFAULT TEMPORARY TABLESPACE...`). En caso contrario, crear uno (`CREATE`

TEMPORARY TABLESPACE . . .), asignarlo a los usuarios y/o definirlo como tablespace temporal predeterminado.