

Administración de Bases de Datos

Configuración y Administración del espacio en disco

Definición de espacio de almacenamiento

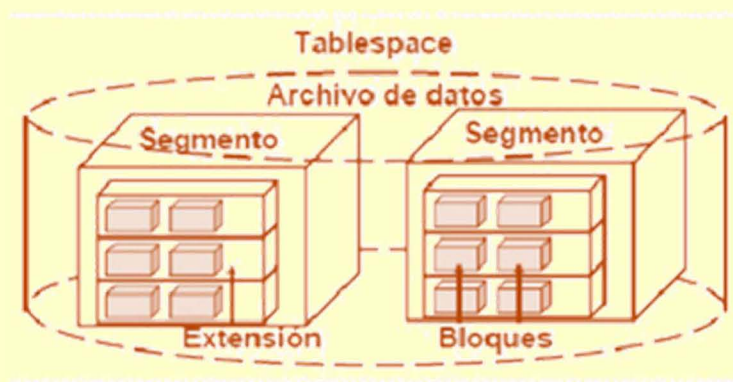
Una Base de Datos se divide en unidades de almacenamiento lógicas llamadas *Tablespaces*.

Una base de datos Oracle está formada por uno o más *Tablespaces*, donde existe al menos el tablespace SYSTEM (que contiene el diccionario de datos).

Cada tablespace contiene distintos objetos relacionados, como las tablas de una misma base de datos y corresponde con uno o más archivos de datos.

Un *Tablespace* consta de uno o más segmentos, cada segmento está formado por extensiones, que son divisiones lógicas.

Una extensión está formada por bloques lógicos, así es como se asigna el espacio, un bloque es una unidad más pequeña para las operaciones de lectura y escritura.



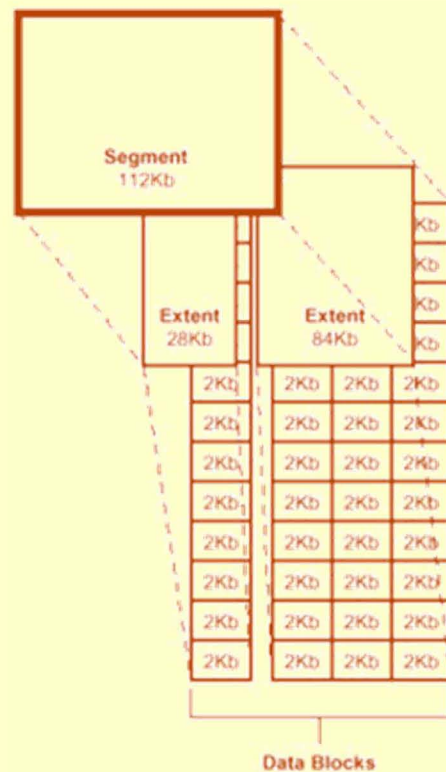
Almacenamiento en una Base de Datos Oracle

Oracle define "esquema" como la colección de estructuras lógicas que corresponden a los datos almacenados, y crea un nuevo esquema por cada usuario que crea objetos en la base de datos.

No existe ninguna relación directa entre tablespace y esquema, los objetos de un mismo esquema pueden estar en diferentes tablespaces y un tablespace puede almacenar varios esquemas distintos.

El control del uso del espacio del disco se obtiene mediante el dimensionamiento de las distintas estructuras lógicas de almacenamiento: bloque de datos, extensión y segmento.

El nivel más bajo de la base de datos es el del bloque de datos.



Tablespace

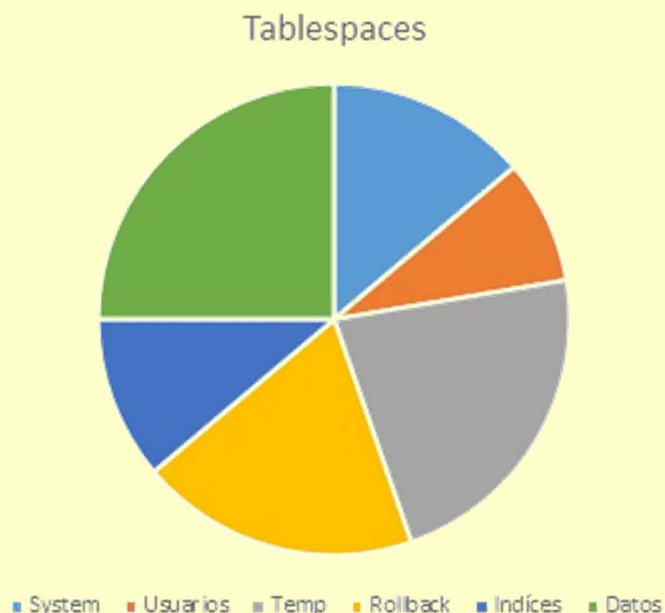
La distribución lógica de los datos está asociada directamente con la eficiencia de la base de datos, por eso es que se manejan varios tipos de datos de las tablespace. Se pueden tener tantos tablespace como sea necesario.

Las propiedades que se asocian a un tablespace cuando un objeto es creado dentro de un tablespace, hereda:

- Localización de los ficheros de datos
- Especificación de consumo de disco.
- Control de disponibilidad de datos
- Backup de datos

A continuación, se muestra la distribución de tablespace que se designa para la base de datos la cual es la siguiente.

- System (Diccionario de datos)
- Usuarios
- Temp
- Rollback
- Indices
- Datos



Si todos los datos están en el mismo espacio de memoria, podrían existir varios problemas, entre ellos problemas de seguridad.

Al tener todos los datos en el mismo disco, si el sistema operativo falla, se cae la base de datos o hay problemas físicos con el disco, todos los datos se perderían, aun teniendo una tabla de respaldo, pues se encuentra en la misma distribución de la memoria.

Una posible solución de este problema es distribuir en diferentes ubicaciones, una tabla podría ir en C: otra en D: y otra en F:, la información sería más fácil respaldar así.

A continuación, se explica con más detalle cada uno de los tablespaces designados, así como su contenido.

System (Diccionario de datos)

El tablespace system será destinado para el diccionario datos el cual contiene toda la información de las tablas como nombres de columnas, cantidad de columnas, atributos, tipos de datos y otros datos importantes para la base de datos.

Así como también serán almacenados catálogos de componentes PL/SQL creados para el control de estos.

Además, es importante tener en cuenta que el espacio de tablas SYSTEM esté al 50% o 75% de su espacio disponible. Finalmente, asegurarse de que los usuarios no tienen privilegios de escritura en el espacio de tablas SYSTEM, solo un usuario de tipo administrador de sistema debe poder modificar estos registros.

Usuarios

Este *tablespace* se utiliza para almacenar la información de todos los usuarios que utilicen las bases de datos, para así poder consultarlos y poder asignales ciertos

permisos, recursos o procedimientos.

Temp

Ciertas operaciones, necesitan de espacio en disco para poder realizarse. El espacio reservado se denomina "segmentos temporales". Se pueden crear segmentos temporales cuando:

- Se crea un índice.
- Se utiliza ORDER BY, DISTINCT o GROUP BY en un SELECT.
- Se utilizan los operadores UNION, INTERSECT o MINUS.
- Se utilizan *joins* entre tablas.
- Se utilizan subconsultas.

El *tablespace* temporal será destinado a todos aquellos datos temporales que ayudaran a mejorar el rendimiento de acceso a registros en la base de datos.

Este espacio es de vital importancia para realizar operaciones en las que se lleven a cabo ordenamientos, la creación y análisis de índices, para el cálculo de estadísticas, etc.

En los casos donde el servidor no disponga de espacio libre suficiente en memoria utilizara el *tablespace temp*.

Es importante mencionar que los rendimientos son muy superiores en comparación con otros tiempos de tablespaces normales, por la manera en que se utiliza este *tablespace*, diseñado para objetos que crecen muy rápido y que rápidamente disminuyen su tamaño y desaparecen.

Rollback

El *tablespace rollback* se destina para almacenar todas las bitácoras correspondientes a las transacciones hechas en la base de datos esto con el fin de mantener una copia de seguridad en caso de fallo en la base de datos.

Índices (index)

Este *tablespace* será destinado a los índices de tablas, debido a que los índices asociados a tablas cuyos datos son modificados frecuentemente crecen en tamaño en mayor proporción que las tablas, es mejor si los índices de tablas grandes se mantienen en espacios de tablas separados.

Datos

Este *tablespace* se utilizará para almacenar los registros de todas las tablas almacenadas en la base de datos. Dado el crecimiento de la cantidad de registros por tabla es recomendable prever un espacio grande para el *datafile* que se encuentre relacionado a este *tablespace*.

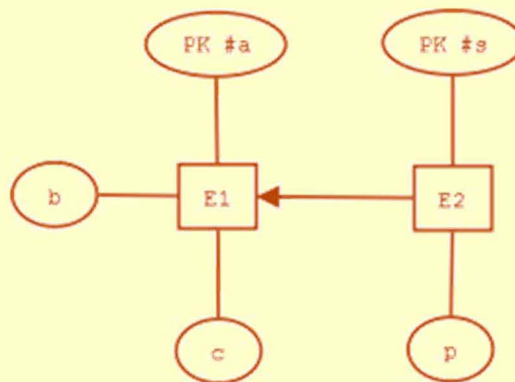
Cálculo de los tamaños de los *tablespaces*

Hay que considerar que los *tablespaces* corresponden a una agrupación lógica, se refiere a cómo se distribuye una base de datos de manera lógica, la memoria que utiliza y cómo se maneja. Es necesario saber cuánto espacio va a ocupar un *tablespace* en la memoria.

Primeramente, hay que tener un diagrama del esquema asociado.

En este ejemplo se tienen dos Entidades, la entidad E1(a, b, c), donde son de tipo (Integer, Char, Char) respectivamente. Y la entidad E2(s, a), donde son de tipo (Integer, Integer) respectivamente.

El tamaño de un atributo tipo Integer es de 4 bytes y el tamaño de un atributo tipo Char es de 1 byte por carácter hasta 255 caracteres.



E1		
Nombre	Tipo	Tamaño
#a	Integer	4 bytes
b	Char	1 byte por carácter
c	Char	1 byte por carácter

E2		
Nombre	Tipo	Tamaño
#s	Integer	4 bytes
a	Integer	4 bytes

Lo primero que se debe hacer es calcular el tamaño de la tupla, eso se hace multiplicando cada dato por su tipo.

Ejemplo

E1 (a = 4, b = 'ABC', c = 'QR') y E2 (s = 3, a = 4), ahora lo que hay que hacer es calcular el tamaño de la tupla.

Para E1: $Er1 = (1 \cdot 4) + (3 \cdot 1) + (2 \cdot 1) = 9 \text{ bytes}$

Para E2: $Er2 = (1 \cdot 4) + (1 \cdot 4) = 8 \text{ bytes}$

Además, hay que considerar la tasa de transacción de las entidades E1 y E2, asumiendo que son 25,000 transacciones por día de E1 y 38.000 transacciones por día de E2.

Ahora se debe multiplicar el tamaño de la tupla por las tasas de transacción de cada entidad, con eso se calcula la cantidad de bytes que se usan por día.

Para E1: $Tr1 \rightarrow 9 * 25,000 = 225000 \text{ bytes/día}$

Para E2: $Tr2 \rightarrow 8 * 38,000 = 304000 \text{ bytes/día}$

Se suman ambas cantidades para saber la cantidad de bytes totales que se consumen por día.

$Tr1 + Tr2 = 529000 \text{ bytes/día}$

Además, se requiere también incluir las llaves primarias y foráneas de cada tabla de la relación. Hay que calcular para cada llave primaria y llave foránea su cantidad en bytes, incluyendo la posición que es una dirección de memoria.

Para E1 se tiene sólo que la llave primaria es de tipo Integer lo cual representa 4 bytes y la posición de memoria que ocupa su apuntador es un long integer que ocupa 16 bytes.

En el caso de E2 se tiene una llave primaria y una foránea, ambas de tipo integer y sus posiciones son de tipo long integer.

Ahora se calcula el tamaño de las llaves con sus posiciones en memoria.

E1 \rightarrow LlavesE1 (PK, posición) $\rightarrow a(\text{Integer}) + \text{long integer}$
 $= 4 \text{ bytes} + 16 \text{ bytes} = 20 \text{ bytes}$

Total de E1 20 bytes.

E2 \rightarrow LlavesE2 (PK, posición) $\rightarrow s(\text{int}) + \text{long integer}$
 $4 \text{ bytes} + 16 \text{ bytes} = 20 \text{ bytes}$
(FK, posición) $\rightarrow a(\text{int}) + \text{long integer}$
 $4 \text{ bytes} + 16 \text{ bytes} = 20 \text{ bytes}$

Una vez calculados los tamaños de las transacciones y sus llaves, se debe hacer un estimado del espacio que puede llegar a ocupar, para este cálculo hay que considerar que las bases de datos relacionales de Oracle consideran que se debe aumentar un 35% el valor calculado como margen de seguridad.

Para E1 $\rightarrow (9 * 25000) + (20 * 25000 * 0.35) = 400000 \text{ bytes/día}$

Para E2 $\rightarrow (8 * 38000) + (40 + 38000 * 0.35) = 317340 \text{ bytes/día}$

Sumando ambos resultados se obtiene el espacio estimado de los tablespace que se consumiría por día.

$400000 + 317340 = 717340 \text{ bytes/día.}$

Datafile

Un datafile son los "archivos de datos" donde se almacena la información físicamente, este puede tener cualquier nombre y extensión, y puede estar

localizado en cualquier directorio del disco duro, además está asociado a un solo tablespace y un tablespace contiene uno o varios datafiles.

Los datafiles tienen una propiedad llamada AUTOEXTEND, que en dado caso de que está activa, esta se encarga de que el datafile crezca automáticamente cada vez que se necesite más espacio.

Aunque el datafile este vacío, tiene el tamaño que ha sido indicado en su creación o el que el sistema le ha dado por defecto. Oracle hace esto para direccionar espacio continuo en disco y evitar así la fragmentación. Cuando se vayan creando objetos en ese tablespace, se va ocupando el espacio a donde se direccionó la memoria.

Segment

Un segment (segmento) es aquel espacio direccionado por la base de datos dentro de un datafile para ser utilizado por un solo objeto. Así una tabla (o cualquier otro objeto) está dentro de su segmento, y nunca podrá salir de él, ya que, si la tabla crece, el segmento también crece. Se podría decir que el segmento es la representación física de un objeto en la base de datos.

Existen cuatro tipos principales de segmentos:

- Segmentos de tipo TABLE: son aquellos que contienen tablas creadas en la base de datos.
- Segmentos de tipo INDEX: son aquellos que contienen índices, que contienen las direcciones de memoria donde se encuentran los datos, inicios de tablas, fin de tablas, etc.
- Segmentos de tipo ROLLBACK: son aquellos se usan para almacenar datos de información de la transacción realizada en la base para poder recuperar información o tener registros.
- Segmentos TEMPORALES: aquellos que se usan para realizar operaciones temporales que no pueden realizarse en memoria, tales como ordenaciones o agrupaciones de conjuntos grandes de datos.

Extend

Extent es un concepto que está relacionado con cualquier objeto que ocupe espacio de disco, es decir cualquier objeto que tenga un segmento relacionado, que se direcciona una sola vez en el tiempo.

El concepto de extent es un concepto físico, unos extent están separados de otros dentro del disco. Una vez aclarado que todo objeto tiene un segmento asociado, pero a su vez se compone de distintas extensiones. Un segmento, puede ser reservado una sola vez, en la creación se asignan 10MB de una vez, o de varias veces 5MB hoy y 5MB mañana.

Existen dos tipos de extends:

- **INITIAL** (extensiones iniciales): estas son las extensiones que se direccionan en el momento de la creación del objeto. Una vez que un objeto está creado, no se puede modificar su extensión inicial.
- **NEXT** (siguientes o subsiguientes extensiones): toda extensión reservada después de la creación del objeto. Si el **INITIAL EXTENT** de una tabla está llena y se está intentando insertar más filas, se intentará crear un **NEXT EXTENT** (siempre y cuando el datafile tenga espacio libre y tengamos cuota de ocupación suficiente).

Sabiendo que las extensiones se crean en momentos distintos de tiempo, es lógico pensar que unas extensiones pueden estar separadas de otras. Un objeto de base de datos no está todo junto dentro del bloque, sino que se secciona en tantos bloques como extensiones tenga. Por eso es crítico definir un buen tamaño de extensión inicial, ya que, si es lo suficientemente grande, el objeto nunca estará fragmentado.

Si el objeto tiene muchas extensiones y éstas están muy separadas en disco, las consultas pueden tardar mucho más.

El tamaño de las extensiones (tanto las **INITIAL** como las **NEXT**), se definen durante la creación del objeto y no puede ser modificado después de la creación. Oracle recomienda que el tamaño del **INITIAL EXTENT** sea igual al tamaño del **NEXT EXTENT**.

La mejor solución es calcular el tamaño que tendrá el objeto ya sea una tabla o un índice, para esto se multiplica el tamaño de cada fila por una estimación del número de filas de la tabla. Este resultado es el que se debe utilizar para el tamaño de la extensión **INITIAL** y **NEXT**, evitando así la fragmentación en el disco.

En caso de detectar que hay más de 10 extensiones en un objeto es recomendable crear un nuevo objeto.

Data block

El concepto de Data block es un concepto físico, pero de una distribución lógica, donde se determina la cantidad de bytes contiguos de espacio físico en el disco, se maneja como bloques de memoria, este proceso es llamado paginación.

El soporte de paginación en las bases de datos Oracle aumenta el desempeño de las aplicaciones de base de datos de mucha memoria, especialmente en los casos en que el caché de buffer tiene varios gigabytes de tamaño, las CPU en el sistema podrán acceder más rápidamente a los buffers en la base de datos.

Ya que los bloques de datos representan la mínima unidad de almacenamiento que es capaz de manejar Oracle. En un disco duro no es posible que un fichero pequeño ocupe menos de lo que indique la unidad de asignación, así si la unidad de asignación es de 4 Kb, ya que así fue asignado por quien creó los ficheros, un fichero que ocupe 1 Kb en realidad ocupa 4 Kb.

Estándar de documentación

A continuación, se dan algunas pautas para realizar la documentación de la arquitectura lógica

1. Iniciar en una introducción que contiene la documentación de la arquitectura lógica.
2. Realizar un gráfico de pastel para representar la división que se hizo de los tablespaces y asignar un color diferente para cada tablespace e indicar el nombre de cada tablespace.
3. Especificar cada tablespace, esta especificación debe contener sus nombres, cuales datos serán almacenados y también indicar a cual datafile se encuentra asociado o en caso de que sean varios indicar cuales.
4. Documentar como fueron calculados los espacios de disco la primera vez cuales fueron los criterios y fórmulas utilizadas para realizar los cálculos.
5. Documentar las fórmulas o estándar que se decida seguir utilizando después de implementado la base de datos para realizar las estimaciones de espacio.
6. Finalmente se recomienda estar actualizando la documentación de la arquitectura lógica cada vez que se cree un tablespace nuevo o se realiza cualquier otro cambio en la arquitectura lógica.