

Nivel Físico

Tema 3

1 - El SGBD de Oracle	4
Estructura básica	4
Estructura lógica	4
Estructura física	4
Mecanismos para almacenar ficheros	5
Instancias de Oracle	5
Definición	5
Partes	6
2 - Espacios de tablas y datafiles	7
Espacio de tablas	7
El espacio de tablas SYSTEM	7
Espacios de tablas read-only	8
Espacios de tablas online / offline	9
Espacios de tablas temporales	9
Datafile	9
Contenido	9
Tamaño	10
Crecimiento de una BD	10
3 - Vistas dinámicas del diccionario	11
4 - Gestión del espacio lógico	12
Almacenamiento lógico vs físico	12
Bloques de datos / página	13
Tamaño	13
Formato	13
Control del espacio libre	13
Reagrupación del espacio libre	14
Encadenamiento y migración de filas	14
Extensiones	15
Segmentos	15
Segmentos de datos	16
Segmentos de índices	16
Segmentos temporales	17
Segmentos de rollback / undo	17
5 - Estructura de la memoria	18
Gestión de memoria	18

Estructuras básicas de memoria	18
Área Global del Sistema (SGA)	19
Caché de los buffers de la BD (Database Buffer Cache)	19
Registro de Rehacer (Redo Log Buffer)	19
El Pool compartido (Shared Pool)	20
Pool grande	21
Pool de Java	22
Streams Pool	22
SGA fijo (fixed SGA)	22
Áreas Globales de Programas (PGA)	23
Área Global de Usuario (UGA)	23
Áreas de Código Software (SCA)	24
6 - Estructura de los procesos	25
Tipos de procesos	25
Procesos de Usuario	25
Procesos de Oracle	25
Procesos de servidor	25
Procesos de segundo plano	25
7 - Objetos del esquema	28
Tablas	28
Tablas temporales	29
Tablas externas	29
Tablas organizadas por índice (IOT)	30
Índices	30
Cuando se crea un índice	31
Árboles binarios	32
Formato del ROWID	32
Índices de mapa de bits	32
Uso de índices	33
Clusters	33
8 - Administración del SGBD de Oracle	34
Funciones del administrador o DBA	34
Utilidades de administración	34
Privilegios para la administración	35
Crear una BD de Oracle	35
Algunas sentencias SQL del DBA	36
9 - Iniciar / Finalizar Oracle	37
Inicialización (startup)	37
Finalización (shutdown)	37
Permisos necesarios	37

Apertura / Cierre con SQL*PLUS	37
Apertura	37
Cierre	38

1 - El SGBD de Oracle

Estructura básica

Estructura lógica

Está compuesta por:

- **Objetos del esquema:** Definición de tablas, vistas, índices, sinónimos, procedimientos...
- **Espacios de tablas:** Es un área lógica de almacenamiento.
 - Informan de cómo debe usarse el espacio físico de la BD.
 - Describen y gestionan el espacio físico de la BD.
 - Existe un espacio de tablas como mínimo para cada BD.
 - Cada espacio de tabla pertenece a una sola BD.
 - *Los espacios de tablas pueden dividirse en varios ficheros.*

Estructura física

Está compuesta por **4 tipos** de ficheros:

- **Datos:** Registran los datos actuales en uno o más ficheros con un formato especial (propiedad de Oracle) para que no puedan leerse por otros programas.
- **Registro de rehacer (redo log):** Registran cambios efectuados, para poder ejecutar una recuperación si fuera necesario. Esto se denomina **rolling forward**.
 - Cada BD tiene un conjunto de 2 o más archivos de este tipo.
 - Si un fallo impide la escritura de datos modificados pueden recuperarse los cambios a través del registro.
 - Oracle permite un registro multiplexado (copias en discos distintos).
 - La información de un archivo del registro solo se usa para recuperar la BD de un fallo del sistema o de medios que impiden que se escriban en los archivos de datos de la BD.
- **Control:** Información general como el nombre de la BD, como el nombre y la localización de los ficheros, fecha de creación, copias de seguridad...
- **Rastreo (trace files) y registro de alarmas (alert log):** Se registran las operaciones por las que han pasado determinados procesos y los eventos importantes ocurridos en la BD.

Ficheros secuenciales:

- **Desventaja:** *el acceso al dato n-ésimo.*
- **Ventaja:** *siempre almacena la última posición y son compactos (están llenos).*

Mecanismos para almacenar ficheros

- **Mediante el SO (más usado)**
 - Los archivos se almacenan en un sistema de archivos.
 - El SO asigna y libera espacio en disco en los archivos.
 - Cada archivo tiene un nombre y se ofrece como un espacio de direcciones contiguo.
 - La BD puede crear, leer, escribir, cambiar el tamaño y eliminar archivos.
- **Oracle ASM (Automatic Storage Management)**
 - Gestor de volumen que proporciona un sistema de archivos diseñado exclusivamente para uso de la BD.
- **Raw device**
 - Son particiones de disco o volúmenes lógicos no formateados.
- **Cluster File System**
 - Varias computadoras comparten el almacenamiento de archivos y mantienen consistente la distribución del espacio y el contenido del archivo.
 - Si un equipo del clúster falla, el sistema de archivos sigue disponible.
- **Combinaciones de las anteriores**

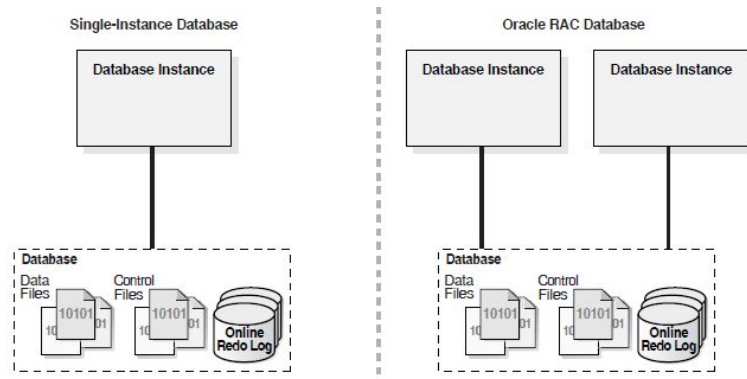
Instancias de Oracle

Definición

Una **instancia** / **servidor de BD** es un conjunto de estructuras de memoria y procesos que acceden a los archivos de una BD.

- Distintas instancias pueden acceder a la misma BD.
- Distintas BDs no pueden ser montadas sobre la misma instancia.
- Se necesita una instancia para acceder a los ficheros.

Desde Oracle 12 una instancia puede ser de solo-lectura, cambiando el modo a READ_ONLY.



Partes

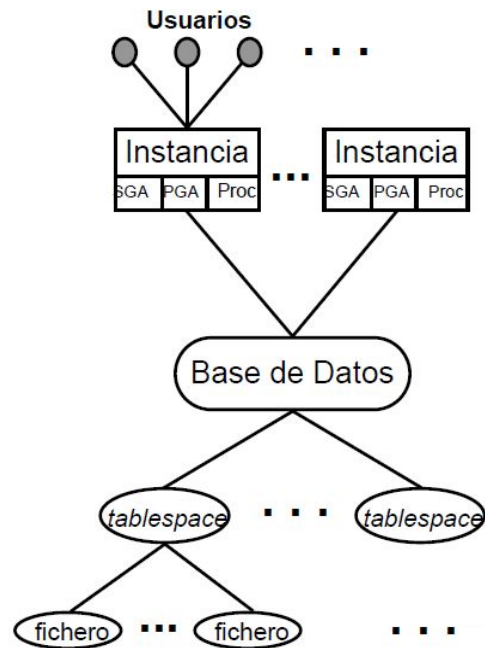
- **SGA (System Global Area)**

Área de memoria con la información de la BD que pueden compartir los usuarios. Se crea cuando se empieza a usar la BD.

Puede usarse el comando `SHOW SGA` para ver algunos datos.

Está compuesta por:

- **Caché de la BD:** Compuesta por los bloques de BD más recientemente accedidos, para reducir los accesos al disco.
 - **Buffer del Registro de Rehacer (Redo Log Buffer).**
- **PGA (Program Global Area)**
Buffer de memoria con información sobre los procesos.
 - **Procesos de Usuario**
Aplicaciones que ejecuta el usuario.
 - **Procesos de Oracle**
Procesos del servidor para atender a los usuarios y procesos de segundo plano para tareas de registro, monitorización...



2 - Espacios de tablas y *datafiles*

Oracle almacena los datos:

- **Lógicamente**, en los **espacios de tabla**.
- **Físicamente**, en los ***datafiles***.

Espacio de tablas

- Una BD está dividida en **uno o más espacios de tablas**.
- El **administrador** usa los espacios de tablas para:
 - Controlar la **creación de espacios en discos** para los datos de la BD.
 - Asignar **cuotas específicas** para los usuarios.
 - Controlar la **accesibilidad** de los datos: *online / offline*, READ_ONLY...
 - Realizar **operaciones parciales**: *backup / restore*.
 - Repartir los datos en **varios discos** para mejorar el **rendimiento**.

Ejemplos:

Crear / cambiar un espacio de tabla temporal / permanente.

CREATE / ALTER TABLESPACE espacio_tabla TEMPORARY / PERMANENT;

Borrar un espacio de tabla.

DROP TABLESPACE espacio_tabla;

El espacio de tablas SYSTEM

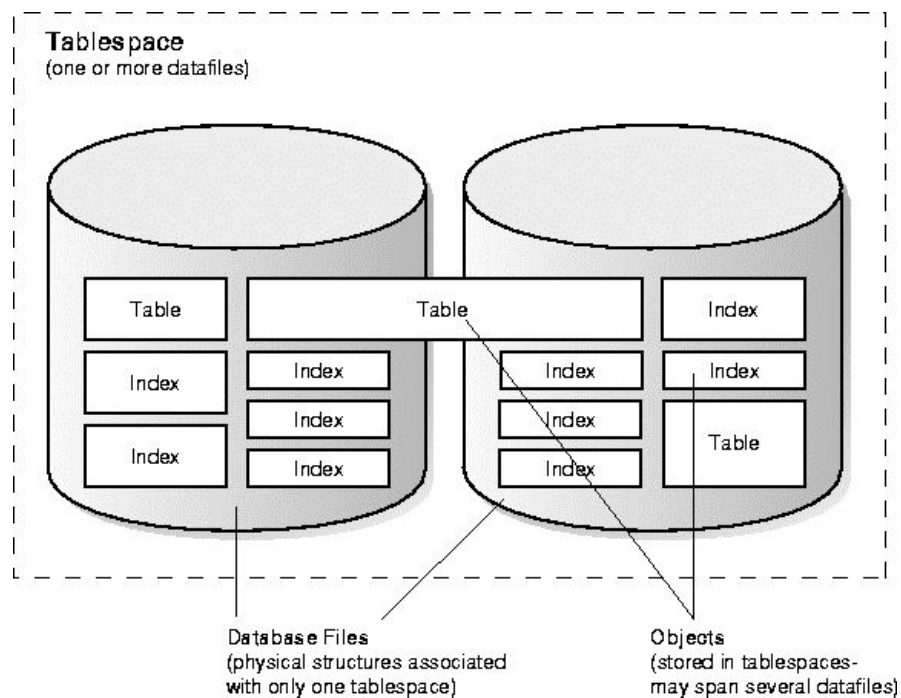
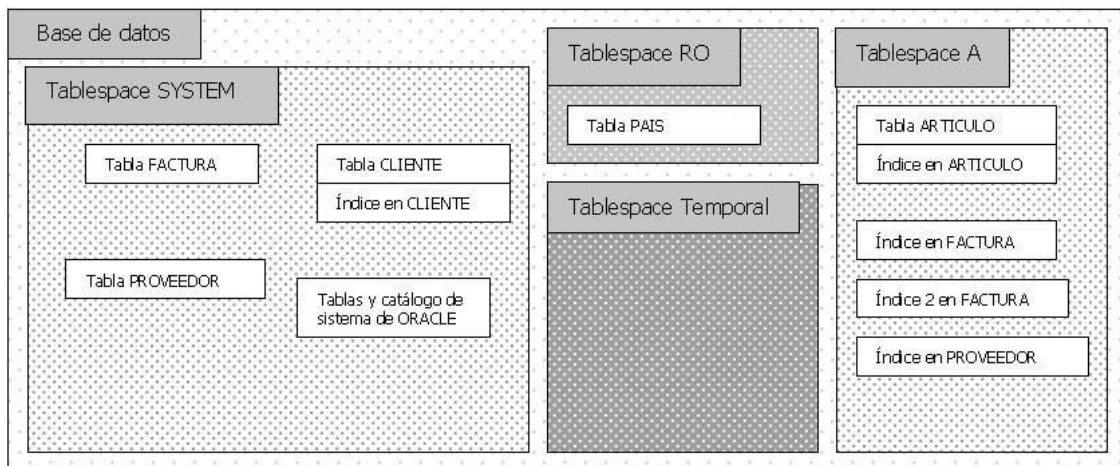
Cada BD de Oracle contiene un espacio de tabla llamado **SYSTEM**, creado automáticamente con la BD.

- SYSTEM contiene las **tablas del Diccionario de Datos** de la BD.
- Una **BD pequeña** puede **bastarse** con solo este espacio de tablas.

Sin embargo, se recomienda crear al menos otro adicional, para separar la información del Diccionario de Datos, esto proporciona:

- Más flexibilidad en las tareas de administración.
- Reducción de problemas de acceso concurrente al espacio de tablas.

- **SYSTEM siempre está online** mientras la BD esté abierta.



Espacios de tablas *read-only*

```
ALTER TABLESPACE espacio_tabla READ ONLY;
```

- **Eliminan** la necesidad de crear **backups** y **recuperaciones** de **porciones de BD**.
- **Oracle nunca actualiza** los ficheros de un espacio de tabla de solo-lectura, por lo que estos espacios de tablas pueden residir en dispositivos solo de lectura.

Como por ejemplo, un CD-ROM.

Espacios de tablas *online* / *offline*

Como se mencionó anteriormente, un administrador de la BD puede hacer o no accesibles los espacios de tablas mientras la BD está abierta.

- **Generalmente, un espacio de tablas está *online*** para que los usuarios accedan a sus datos.
- Un administrador puede hacer las tablas inaccesibles (*offline*) para ...
 - ... dejar inaccesible una **porción** de la BD.
 - ... realizar un ***backup offline*** del espacio de tablas.
 - ... hacer que algunas aplicaciones / tablas queden temporalmente inaccesibles frente a operaciones de **mantenimiento / modificación**.
- Oracle puede poner un espacio de tablas *offline* si se produce un **error grave**.

No pueden ponerse offline:

- *Espacios de tablas que contengan segmentos de rollback en uso.*
- *El espacio de tablas SYSTEM.*

Espacios de tablas temporales

Se usan para realizar operaciones de **ordenación (*sort*) de forma más eficiente**.

- Pueden ser utilizados para contener únicamente segmentos *sort*.
- **Objetos permanentes no pueden residir en ellas.**

Desde Oracle 12, pueden definirse espacios de tablas temporales locales a cada instancia.

Datafile

Un espacio de tablas de una BD está compuesto por uno o más *datafiles*.

- Un *datafile* puede estar asociado a un único espacio de tablas.
- El primer *datafile* siempre es el asociado al espacio de tablas SYSTEM.

Contenido

- **Cuando un *datafile* se crea, su espacio de tablas es formateado** para que pueda contener los datos del usuario.
- Los datos asociados con los objetos del esquema serán almacenados físicamente en *datafiles*, pero no existe una correspondencia directa entre ellos.
 - **Un objeto se corresponde con un espacio de tablas** y no un *datafile*.

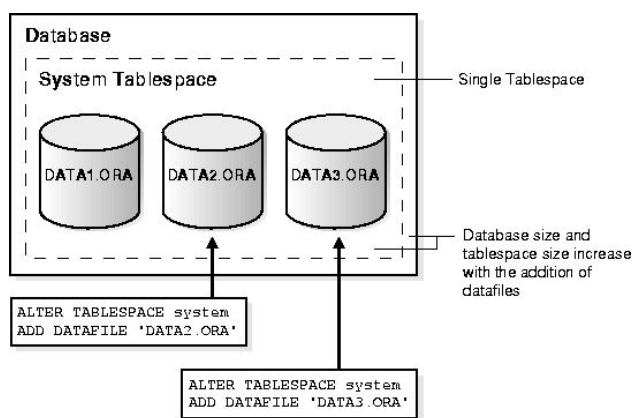
Tamaño

- Puede alterarse el tamaño de un *datafile* **después** de haber sido creado.
- Esto permite facilitar las tareas de administración de la BD.
- La instrucción correspondiente está dentro de ALTER DATABASE (...).

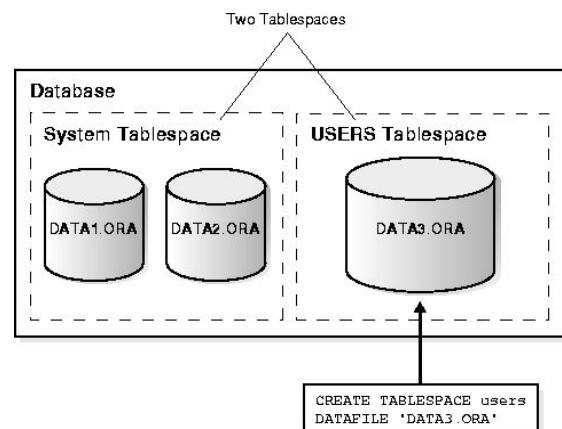
Crecimiento de una BD

La BD puede crecer de 3 formas diferentes:

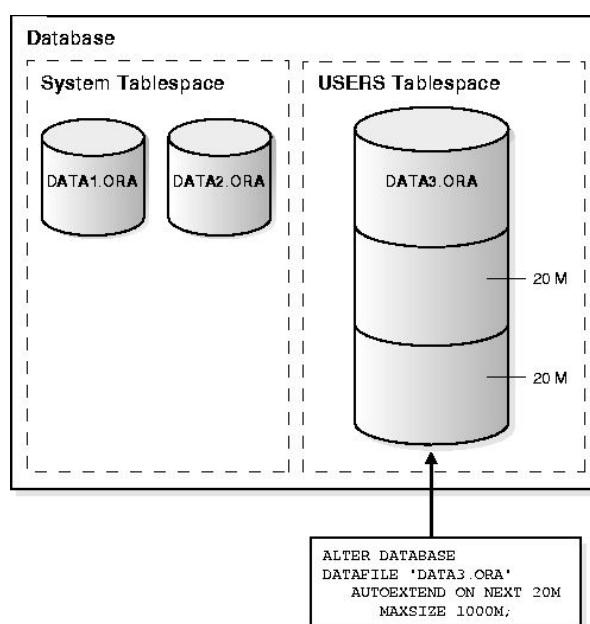
Añadiendo un *datafile* al espacio de tablas



Añadiendo un nuevo **espacio de tablas**



Incrementando el **tamaño de un datafile**



3 - Vistas dinámicas del diccionario

Las **tablas de ejecución dinámica** son tablas virtuales con información sobre la actividad de la BD durante su funcionamiento; no son verdaderas tablas y no son accesibles para la mayoría de los usuarios.

Los administradores de la BD pueden crear / consultar vistas -llamadas ***fixed views***- sobre esas tablas y autorizar el acceso a esas vistas por parte de otros usuarios. *Sin embargo, no puede modificarlas ni eliminarlas.*

El propietario es SYS y los nombres de las mismas comienzan por V_\$; para esas tablas se crean vistas y para ellas se crean sinónimos que comienzan por V\$. Algunos de ellos son:

Sinónimos	Contenido
v\$database	Información sobre la BD
v\$datafile	<i>Datafiles</i> asociados a los espacios de tablas
v\$fixedtable	Información sobre tablas y vistas dinámicas
v\$process	Procesos que se están ejecutando
v\$controlfile	Lista de ficheros de control
v\$session	Información sobre las sesiones activas
v\$logfile	Archivos de <i>redo log</i>
v\$log	Archivos de <i>redo log</i> y su estado

Algunas instrucciones:

<i>Eliminar una sesión (forzar cierre)</i>	<code>ALTER SYSTEM KILL SESSION 'sid,serial#';</code>
<i>Generar las instrucciones para matar las sesiones de un usuario denominado ALUMNOBD1</i>	<code>SELECT 'ALTER SYSTEM KILL SESSION ''' sid ',' serial# ''';' FROM v\$session WHERE username = 'ALUMNOBD1';</code>
<i>Bloquear / desbloquear una sesión</i>	<code>ALTER USER usuario ACCOUNT [lock unlock];</code>

4 - Gestión del espacio lógico

El SGBD gestiona las extensiones dentro de un espacio de tablas **localizando uno libre o liberando uno** que no se usa. Estos pueden gestionarse de 2 formas:

- **Gestionados localmente (por defecto)**

- **ASSM** (*Automatic Segment Space Management*)

Solo hay que configurar una parámetro:

- PCTFREE

- **MSSM** (*Manual Segment Space Management*)

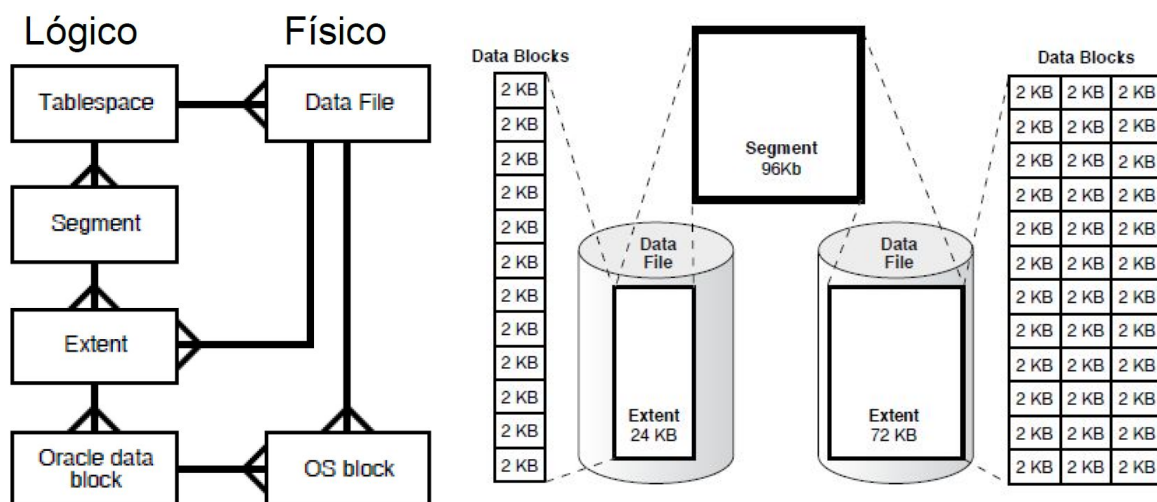
Hay que configurar los parámetros:

- PCTFREE
 - PCTUSED
 - FREELISTS
 - FREELISTS GROUPS

- **Gestionados por diccionario**

Las extensiones libres y utilizadas se guardan en el propio diccionario, pero se produce recursividad, lo que es menos eficiente.

Almacenamiento lógico vs físico



Cuando un fichero debe crecer, lo hace en una extensión.

Bloques de datos / página

Consiste en la **unidad mínima de asignación de espacio** en la BD. Se trata de la menor unidad E/S que puede utilizar la BD, independientemente de que el tamaño de bloque del SO sea menor.

Tamaño

- Especificado por el DBA cuando crea la BD.
- Debe ser múltiplo del tamaño de bloque del SO.
- No puede cambiarse a menos que se cree la BD de nuevo.
- Pueden crearse espacios de tablas con un tamaño de bloque distinto.

Formato

- **Overhead**, formado por 3 zonas:
 - **Block header**: Información general, como la dirección en el disco, tipo de segmento al que pertenece (datos, índice, *rollback...*).
 - **Table directory**: Información sobre las tablas con filas de ese bloque.
 - **Row directory**: Información sobre las filas almacenadas en ese bloque.
 - **Transaction entries**: Información sobre las transacciones (*INSERT*, *UPDATE* y *DELETE*) sobre filas de ese bloque.
- **Free space**, espacio libre para insertar filas nuevas o modificar valores de filas, si requieren más espacio.
- **Row data**, zona con datos de tablas o índices.
Una fila puede estar en varias bloques.

Control del espacio libre

Existen 2 parámetros que se especifican cuando se crea / modifica una tabla / índice:

- **PCTFREE**: Indica el mínimo % de bloque que se reserva como espacio libre para futuras modificaciones de filas existentes.
 - Por defecto, un 10 %; en una tabla algo estática, un 5 % es un buen valor.

```
CREATE TABLE tabla (n NUMBER) PCTFREE 20;
```

Cuando a un bloque le queda un 20 % de PCTFREE, Oracle ya no lo sigue usando. Si se hace un UPDATE de una fila y el valor nuevo ocupa más espacio que el antiguo, el nuevo se coloca en la parte reservada, evitando leer 2 bloques al leer la fila, ya que un bloque siempre se lee entero.

- **PCTUSED:** Gestión de espacios de tablas manual. Después de que el bloque sea considerado como «lleno» (en función del límite especificado en el PCTFREE), Oracle no vuelve a introducir ninguna fila nueva hasta que el % de bloque ocupado sea menor que el especificado con PCTUSED. Hasta entonces, sólo podrán modificarse las filas existentes.
 - Por defecto, 40 %; en una tabla algo estática, un 75% es un buen valor.
 - **Espacio libre para inserciones** = tam.bloque - tam.overhead - PCTFREE.
 - *Cuando ocurre un INSERT, Oracle mira la lista de páginas disponibles (free list) y selecciona la primera de ellas.*
 - Para actualizaciones (UPDATE), cualquier espacio libre puede ser usado.

Reagrupación del espacio libre

El espacio libre puede aumentar de 2 formas:

- Haciendo DELETE, porque libera espacio.
- Haciendo UPDATE, si los datos nuevos ocupan menos espacio que los anteriores.

Todo el espacio libre podrá usarlo una instrucción INSERT si:

- Está en la misma transacción que la instrucción que ha generado el espacio libre.
- Está en otra transacción y la transacción que dejó el espacio libre ya efectuó un *commit*. Generalmente, ambas transacciones son de usuarios distintos.

Encadenamiento y migración de filas

Los datos de una fila pueden ser demasiado grandes para caber en un bloque, de forma que **se fragmenta la información** y se almacenan en **distintos bloques**.

- Si una fila es muy grande, entonces Oracle almacena la fila en una cadena de bloques del mismo segmento.

Con datos grandes, como los de tipo LONG, la fragmentación es inevitable.

- Si una fila es actualizada y sus nuevos valores no caben, Oracle traslada toda la fila a un nuevo bloque (suponiendo que caben).

Oracle conserva la cabecera de la fila en su página inicial, apuntando a la dirección del nuevo bloque. Así el identificador de fila (ROWID) no cambia.

El tratamiento de este tipo de filas implica una **eficiencia menor**, ya que Oracle debe recuperar los datos desde varios bloques.

Extensiones

Conjunto de bloques de datos contiguos, con un tipo de información específico.

Hay 2 formas de gestionar las extensiones:

- **Gestionadas localmente (LOCAL)**

Al crear un espacio de tablas, especificando:

- AUTOALLOCATE: Gestionadas por el sistema, especificando el tamaño de la extensión inicial, el del resto es calculado por Oracle (mínimo 64KB).
- UNIFORM: Especificando el mismo tamaño indicado para todas las extensiones (por defecto 1MB).

- **Gestionadas por el diccionario de datos (DICTIONARY)**

Utilizan como valores por defecto los valores almacenados en el diccionario de datos de la BD. Dichos valores pueden modificarse en cualquier momento, y son.

- INITIAL, tamaño del primero.
- NEXT, tamaño del segundo.
- PCTINCREASE, % de incremento del siguiente respecto al anterior.

En general, las extensiones de un segmento no son liberadas (*deallocated*) a no ser que se borre el objeto almacenado en el segmento con un `DROP TABLE` o `DROP CLUSTER`. Sin embargo, pueden producirse algunas excepciones: por ejemplo, el administrador puede «desasignar» extensiones no utilizadas con `ALTER TABLE tabla DEALLOCATE UNUSED`; y usando el paquete `DBMS_SPACE_ADMIN`.

Segmentos

Conjunto de extensiones que almacenan un determinado tipo de datos.

Si un segmento se llena, Oracle crea una nueva extensión para ese segmento (incremental extent) del mismo tamaño o superior.

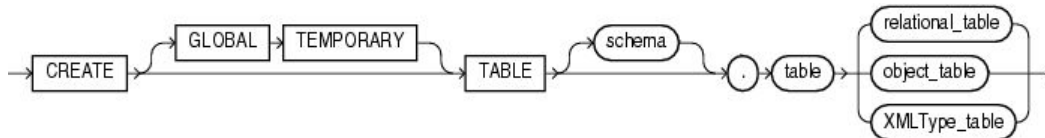
- Al crear una estructura de datos, Oracle le asigna **un segmento con una sola extensión**.
- Cuando se **llena esa extensión** se le asignan **otras extensiones** a ese segmento. Por eso las distintas extensiones no pueden ocupar espacios consecutivos (lo ideal).
- Un **segmento completo** se almacena en **un espacio de tablas**, que puede distribuir sus extensiones en distintos ficheros.
- Cada extensión se almacenará siempre en **un único fichero**.

Existen 4 tipos de segmentos:

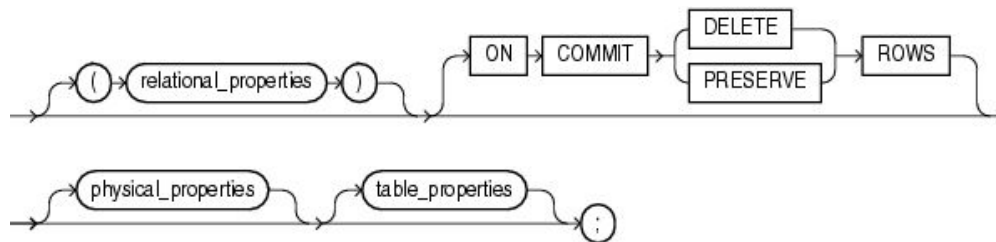
Segmentos de datos

Se generan al usar una sentencia CREATE. Los parámetros de almacenamiento de bloques de datos y extensiones se asignan con CREATE o ALTER y afectan a la eficiencia en el almacenamiento y la recuperación de datos.

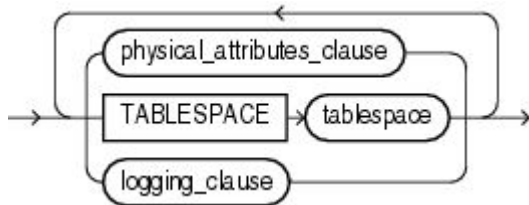
CREATE TABLE



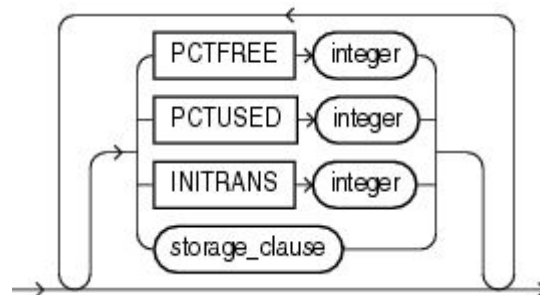
RELATIONAL TABLE



PHYSICAL PROPERTIES



PHYSICAL ATTRIBUTES CLAUSE



Segmentos de índices

Se generan al usar la sentencia CREATE INDEX. Estos índices permiten que puedan fijarse sus parámetros de almacenamiento.

Una tabla y sus índices pueden tener segmentos en un espacio de tablas distinto, lo que permite que el índice y los datos se almacenen en ficheros distintos e incluso, en discos distintos, **aumentando el rendimiento.**

Segmentos temporales

Se generan cuando Oracle quiere procesar una consulta, ya que requiere de un espacio temporal para realizar las operaciones intermedias de la instrucción SQL.

Operaciones que usan segmentos temporales:

- CREATE INDEX;
- SELECT DISTINCT;
- SELECT (...) ORDER BY / GROUP BY;
- SELECT (...) UNION / INTERSECT / MINUS;

Normalmente, solo se requiere este espacio cuando se necesita ordenar un resultado que no cabe en memoria.

Oracle crea los segmentos temporales durante una sesión de usuario en el espacio de tablas temporal del **usuario que realiza la instrucción**.

- Este espacio de tablas es especificado en la instrucción CREATE USER o con ALTER USER (...) TEMPORARY TABLESPACE.
- Si el administrador no ha definido estos parámetros, por defecto, el espacio de tablas temporal será TEMP o SYSTEM.

Una vez que la instrucción se completa, Oracle borra el segmento temporal.

Estos segmentos también pueden usarse para tablas temporales.

Segmentos de *rollback* / *undo*

Almacenan los valores antiguos de datos modificados por transacciones para:

- Lograr consistencia.
- Realizar *rollbacks*.
- Permitir la recuperación (*recovery*) de la BD.

Cada BD contiene al menos un segmento de rollback (*undo tablespace*).

Un segmento de *rollback* contiene numerosos registros (*rollback entries*), como qué datos se modificaron y su contenido.

Oracle enlaza cada entrada de una transacción en caso de tener que deshacerlas. Al grabar las entradas de *rollback*, cambian los bloques de datos del segmento de *rollback* y Oracle guarda todos los cambios en los bloques de datos -incluyendo estas entradas de *rollback*-, en el fichero **redo log**.

- Este segundo almacenamiento es muy importante para transacciones sin finalizar.
- Si el sistema cae, se restaura la información de los segmentos de *rollback*, incluyendo entradas de transacciones activas.
- Al realizar un *commit*, Oracle libera la información de *rollback*.

5 - Estructura de la memoria

Oracle utiliza la memoria para almacenar:

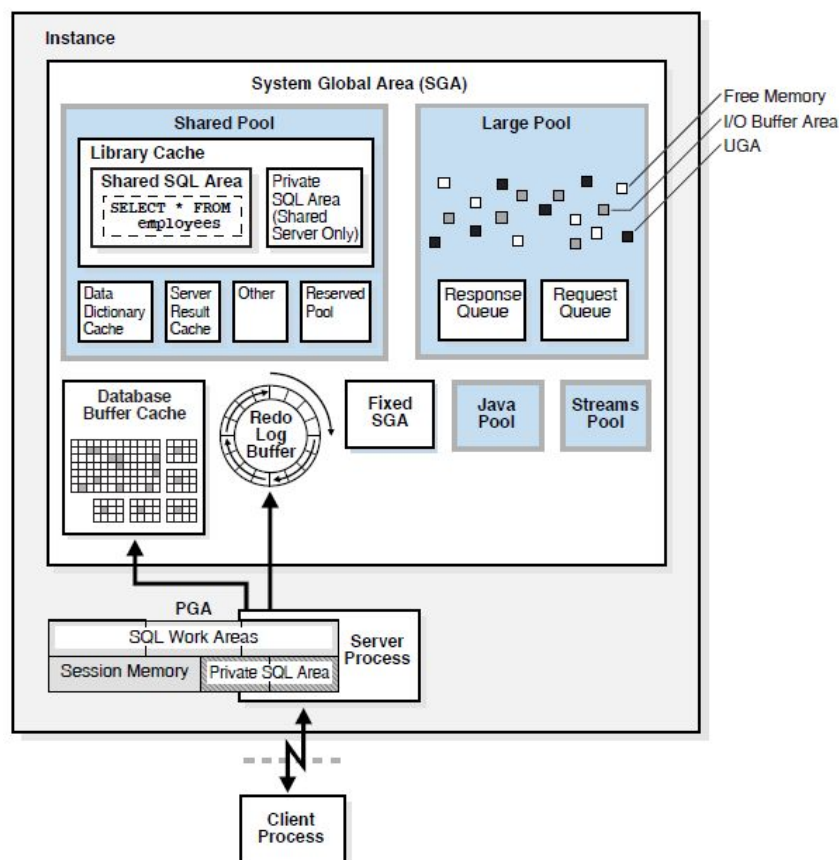
- **Código** del programa que se ejecuta.
- Información sobre las **sesiones conectadas**.
- Información necesaria sobre las **ejecuciones de los programas**.
- Información compartida entre **procesos**.
- **Caché de datos**.

Gestión de memoria

La memoria puede gestionarse de 3 formas:

- **Automática (por defecto)**: se especifica el tamaño objetivo de la instancia y se distribuye automáticamente entre la SGA y la PGA.
- **Compartida**: se especifican los tamaños de la SGA y la PGA, que pueden gestionarse como un todo o individualmente.
- **Manual**: se especifican los tamaños de cada parte de la instancia.

Estructuras básicas de memoria



Área Global del Sistema (SGA)

Un SGA es un **grupo de estructuras de memoria compartida** que contienen datos e información de control de una instancia de BD.

- Si varios usuarios están conectados a una misma instancia concurrentemente, sus datos se comparten en el SGA.
- Un SGA dura lo mismo que una instancia (se crea al inicio y se destruye al final).
- El SGA es de lectura y escritura, y contiene:

Caché de los buffers de la BD (*Database Buffer Cache*)

- Contiene copias de las páginas leídas de los *datafiles*.
- Todos los procesos de los usuarios conectados a la instancia comparten el acceso a esta caché.
- Cuando se hace un *commit* se escriben los *buffers del redo log*, pero no los *bloques de datos* (escritura perezosa en segundo plano).
- Los buffers en la caché están organizados en 2 listas:
 - **Lista en espera (*write list*):** Contiene buffers con datos modificados pero que aún no han sido escritos en disco.
 - **Lista LRU (*Least Recently Used*):** Contiene buffers libres, buffers que están siendo accedidos actualmente y buffers de la lista anterior pero que aún no han sido movidos.
- La primera vez que un proceso requiere un dato, se busca en caché:
 - **Si se encuentra**, se lee directamente de la memoria (*cache hit*), acelerando el proceso de la lectura.
 - **Si no se encuentra (*cache miss*)**, entonces debe obtenerse una copia del bloque de datos en disco y pasarlo a un buffer de la caché antes de leerlo.

Registro de Rehacer (*Redo Log Buffer*)

- Buffer circular en el SGA que contiene información sobre los cambios que se producen en la BD.
- Las entradas contienen información necesaria para reconstruir o rehacer los cambios efectuados en la BD. Estas entradas son copiadas por el servidor de procesos de Oracle desde el espacio de memoria del usuario hasta los buffers del SGA, donde los registros van escribiéndose de forma secuencial.

- Si se produce un fallo en el sistema, los datos que están en la *write list* se perderían, pero todos los cambios se van guardando en los ficheros de *redo log*. Cuando el sistema arranca de nuevo, se rehacen los cambios no grabados (*database recovery*).

- Modificar el archivo sobre el que se vuelcan los cambios:

```
ALTER SYSTEM SWITCH LOGFILE;
```

- Ver el estado del registro:

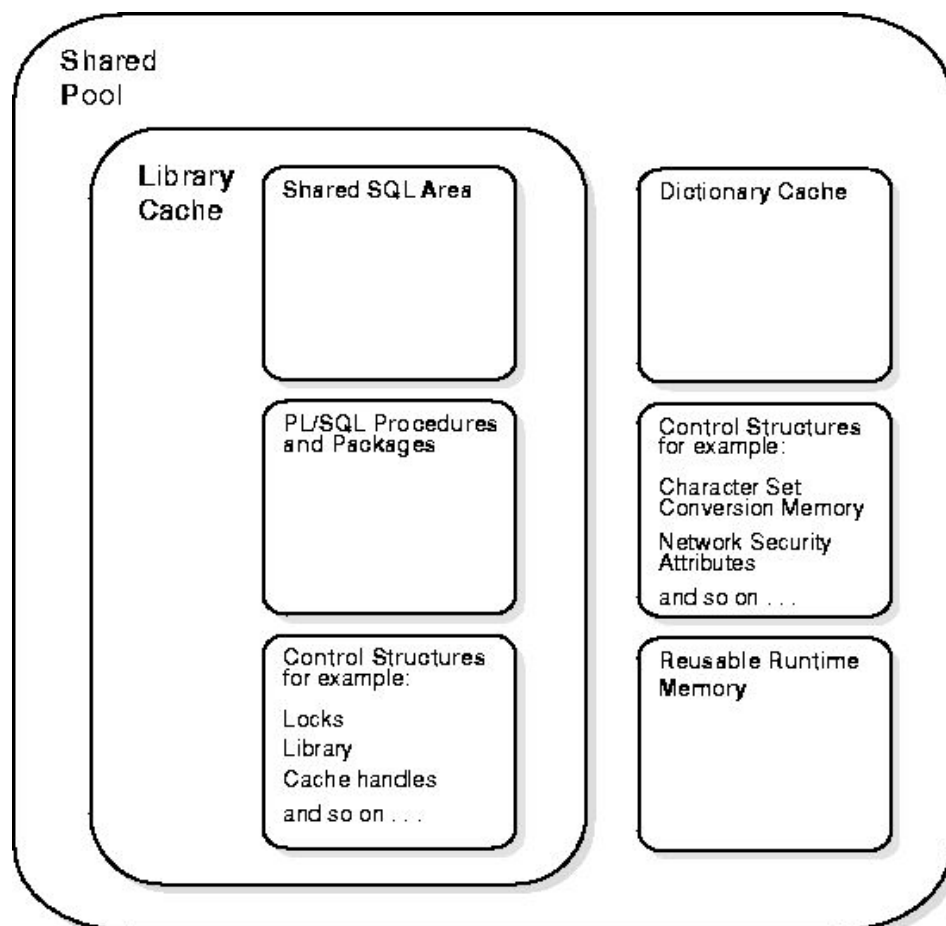
```
SELECT * FROM v$log;
```

El Pool compartido (*Shared Pool*)

Su tamaño total está determinado por el parámetro de inicialización «*SHARED_POOL_SIZE*». Para modificarlo, se usa la instrucción:

```
ALTER SYSTEM SET SHARED_POOL_SIZE = cantidad;
```

Donde «cantidad» es un número en megas (64M, 234M...).



Contiene:

- **Caché de biblioteca** (*library cache*), que incluye:
 - **Áreas compartidas y privadas de SQL**
 - Cada instrucción SQL ejecutada tiene un área compartida y otra privada.
 - Cuando varios usuarios están ejecutando la misma declaración de SQL, se reutiliza el área compartida para dichos usuarios, ahorrándoles tiempo y memoria.
- Sin embargo, cada usuario debe tener una copia separada de la instrucción SQL en su área privada.*
- **Procedimientos y paquetes PS/SQL**
 - Evitan tener varias copias de cada programa PL/SQL, aunque una parte es privada para cada usuario.

Las sentencias SQL contenidas dentro de un programa también utilizan las áreas SQL compartidas y privadas.

- **Estructuras de control**
 - Caché de gestión de bloqueos.
 - Caché de gestión de librerías.
 - (...)
- **Caché de diccionario** (*dictionary cache*)

La continua necesidad de acceso al diccionario hace que Oracle disponga de una caché específica.
- **Estructuras de control**
 - Juegos de caracteres
 - Sistemas de conversión
 - Atributos de seguridad
 - (...)

Pool grande

Estructura de memoria opcional con utilidades de backup y recuperación como RMAN. Su tamaño viene especificado por el parámetro «LARGE_POOL_SIZE».

Pool de Java

Guarda información acerca de la interpretación de las sentencias java que se ejecutan. Su tamaño viene especificado por el parámetro «JAVA_POOL_SIZE».

Solo es necesario si se instala y se usa Java.

Streams Pool

Esta memoria es usada exclusivamente por flujos de Oracle. Esta almacena colas de mensajes y provee memoria para que los flujos de Oracle capturen procesos y los apliquen. A menos que se configure específicamente, el tamaño de esta memoria empieza en cero y crece dinámicamente como sea necesario cuando los flujos son usados.

SGA fijo (fixed SGA)

Parte del SGA con información general sobre el estado de la BD y la instancia, donde los procesos en segundo plano necesitan acceder.

El SGA se puede gestionar de 2 formas:

- **Gestión automática** de la memoria compartida
 - Si se fija el MEMORY_TARGET (dinámico), Oracle modifica dinámicamente el tamaño de los subcomponentes según sea necesario.
 - Si el MEMORY_TARGET tiene un valor mayor que MEMORY_MAX_TARGET (estático) en el *startup*, entonces MEMORY_MAX_TARGET aumenta.

- **Gestión manual / compartida**

El tamaño de la instancia puede crecer hasta el valor marcado por el parámetro de inicialización SGA_MAX_SIZE, pero si los valores iniciales de sus componentes superan ese valor, Oracle ignora ese último.

Existen otros parámetros como:

DB_CACHE_SIZE	Tamaño de la caché de bloques estándar.
LOG_BUFFER	Número de bytes asignados para el <i>redo log buffer</i> .
SHARED_POOL_SIZE	Tamaño en bytes del área de sentencias compartidas.
LARGE_POOL_SIZE	Tamaño del <i>Pool Grande</i> (por defecto, 0M).
JAVA_POOL_SIZE	Tamaño del <i>Pool de Java</i> .

Áreas Globales de Programas (PGA)

Un PGA es un área de memoria...

- ...que contiene datos e información de control para un solo proceso (del servidor o de segundo plano).
- ...no compartida, donde se puede escribir un proceso.

*El PGA es como una zona de un mostrador con un cliente y un dependiente:
el dependiente (proceso) trabaja para el cliente (usuario).*

Para cada proceso del servidor se asigna un PGA.

- Ese PGA es exclusivo para ese proceso.
- Ese PGA se lee y se escribe por Oracle, que actúa en nombre de ese proceso.
- Un PGA es asignado cuando un usuario se conecta a una BD y crea una sesión.
- Un PGA siempre contiene un espacio que contiene las variables de la sesión y alguna otra información. Este espacio es denominado *stack space*.
- El tamaño de un PGA depende del S.O.

Ejemplo:

```
SELECT * FROM empleados e JOIN departamentos d
ON e.departamento_id = d.departamento_id
ORDER BY apellido;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		106	9328	7 (29)	00:00:01
1	SORT ORDER BY		106	9328	7 (29)	00:00:01
*2	HASH JOIN		106	9328	6 (17)	00:00:01
3	TABLE ACCESS FULL	DEPARTMENTS	27	540	2 (0)	00:00:01
4	TABLE ACCESS FULL	EMPLOYEES	107	7276	3 (0)	00:00:01

Run-time: *Número de filas devueltas.*

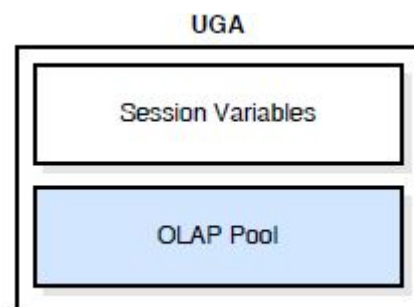
Hash area: *JOIN de las 2 tablas.*

Sort area: *ORDER BY.*

Área Global de Usuario (UGA)

Memoria para las variables de sesión:

- Información de *login*.
- Estado de la sesión.
- Variables de un paquete.



En OLAP (Data Warehouse) se almacenan las OLAP page pools para gestionar mucho mejor los cubos.

La UGA debe estar disponible mientras la sesión esté abierta.

Puede residir en:

- La SGA (servidor dedicado).
- La PGA (servidor compartido).

Áreas de Código Software (SCA)

Zonas de memoria destinadas a almacenar el código de Oracle en ejecución o que puede ejecutarse.

Esta es una zona distinta -y más protegida- de las zonas dedicadas a almacenar los códigos de programas de usuarios.

Son áreas de solo lectura y tamaño estático (depende del S.O).
Solo cambian cuando el software se instala.

Pueden ser compartidas o no compartidas. Las primeras son más eficientes porque el mismo código puede ser usado por distintos usuarios.

6 - Estructura de los procesos

Todo usuario que se conecta a Oracle debe ejecutar 2 módulos de código para acceder a la instancia de la BD. Dichos módulos controlados por el S.O son:

- **Aplicación / Herramienta Oracle:** Una aplicación normal que se conecta a Oracle o una herramienta (como SQL*Plus).
Son programas que permiten ejecutar sentencias SQL.
- **Código del servidor de Oracle:** Es una parte de Oracle que se ejecuta para el usuario y que interpreta y procesa las órdenes SQL.

Los **Sistemas Oracle Multiproceso** son aquellos en los que distintos procesos ejecutan distintas partes de Oracle, además de los procesos de los usuarios.

- La mayoría de los SGBD son multiusuario (una de sus principales ventajas).
- Dividir el trabajo en procesos ofrece un mejor rendimiento para servicios a múltiples usuarios y aplicaciones.

Tipos de procesos

Procesos de Usuario

Generado cuando el usuario ejecuta una aplicación / herramienta de Oracle.

- **Conexión:** Establece el mecanismo de comunicación entre un proceso de usuario y una instancia.
- **Sesión:** Conexión específica de un usuario a una instancia a través de un proceso de usuario. Un mismo usuario puede tener varias sesiones.

Procesos de Oracle

Procesos de servidor

Se crean para cada una de las aplicaciones de usuario.

Procesos de segundo plano

Se crean para maximizar el rendimiento y posibilitar el acceso simultáneo de múltiples usuarios.

- **Database Writer (DBW0 / DBWn)**
Escribe el contenido de los buffers que han sido modificados en los *datafiles*.
- **Log Writer (LGWR)**
Responsable del funcionamiento de los buffers del *red log* mediante la escritura de su contenido al fichero *redo log*.
- **Checkpoint (CKPT)**
Cuando se realiza un *checkpoint*, Oracle actualiza las cabeceras de todos los *datafiles* que almacenan datos a causa del mismo.
- **System Monitor (SMON)**
 - Maneja la recuperación de la BD a partir del fallo de una instancia.
 - Comprueba los espacios de disco periódicamente para determinar si unir pequeños fragmentos de espacio libre.
- **Process Monitor (PMON)**
Actúa cuando falla un proceso de usuario, liberando los recursos que tuviera asignados y deshaciendo los cambios que hubiese realizado desde su último commit.
- **Recoverer (RECO) - Opcional**
Proceso de segundo plano para las BDs distribuidas que gestiona transacciones distribuidas.
- **Archiver (ARCH) - Opcional**
Copia los ficheros del *redo log* a un dispositivo predeterminado cuando este está lleno. *Este proceso solo se presenta cuando el redo log se usa en modo ARCHIVELOG y el archivo automático está activado.*
- **Lock (LCKn) - Opcional**
Gestiona los bloqueos entre distintas instancias.
Solamente para los servidores paralelos.
- **Job Queue (SNPn) - Opcional**
Para BDs distribuidas.
- **Queue Monitor (QMNn) - Opcional**
Monitoriza el orden de salida de los mensajes.
- **Dispatcher (Dnnn) - Opcional**
Permite a los proceso de usuario compartir procesos del servidor, permitiendo que el servidor soporte un mayor número de usuarios.

- **Shared Server (Snnn)**

Cada uno de estos procesos sirve las peticiones de múltiples clientes en configuraciones como la anterior, compartiendo los procesos del servidor.

- **Listener**

- Proceso que escucha peticiones de la conexión del cliente.
- Se configura en el fichero *listener.ora*, con una dirección de protocolo que identifica la BD.
- Conecta al usuario a los *dispatchers* o servidores dedicados.
- Forma parte de Oracle Net Services, no de Oracle.
- Cuando una instancia arranca, establece una ruta de comunicación con Oracle.
- También puede establecer una comunicación entre BDs.
- Sirve para definir servicios y sus instancias en una BD distribuida.

Algunos de estos procesos son creados automáticamente cuando se inicia una instancia. No todos están siempre presentes.

7 - Objetos del esquema

Tablas

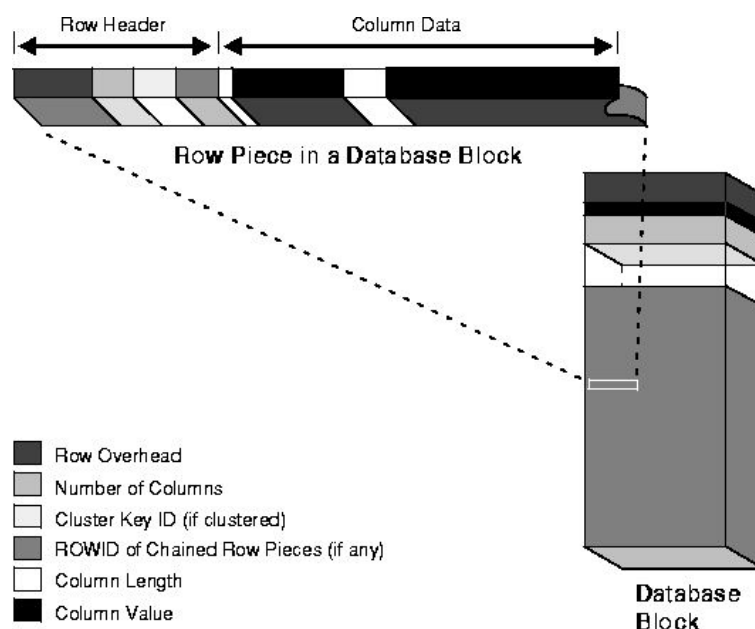
Unidad básica de organización de datos permanentes o temporales.

- Organización de las filas:
 - Heap: se guardan sin orden.
 - IOT (Index-Organized Table): se guardan en el orden de la PK.
 - Tablas externas: sus metadatos se guardan en la BD, pero no sus datos.
- Una tabla puede tener columnas virtuales:

```
CREATE TABLE producto (  
  codigo NUMBER PRIMARY KEY,  
  descripcion VARCHAR2(200),  
  precio_netto NUMBER(9,2),  
  iva NUMBER(2),  
  precio_total  
    AS (precio_netto + precio_netto * iva / 100));
```

- Una tabla puede estar partida (PARTITION) en función de diversos criterios. Cada partición puede almacenarse en un espacio de tabla distinto.

Oracle asigna un segmento de datos de un espacio de tablas para contener los futuros datos de una tabla cuando esta se crea. Puede controlarse la generación de ese espacio mediante los parámetros PCTFREE, PCTUSED... También almacena -habitualmente- cada fila en una única página (un **row piece**).



- Una fila se **fragmenta** en varias páginas **si no puede almacenarse por completo** en una sola, entonces se encadenan las páginas mencionadas.
- Cada fila se almacena en **2 partes**:
 - **Row header**: información de la cabecera, ocupa 3 bytes como mínimo en una fila de un solo bloque.
 - **Column data**: almacena la longitud y los datos de cada columna.
 - NULL se representa como longitud 0 y nada en los datos.
 - Con ALTER TABLE (. . .) SET UNUSED se marca una columna como borrada, posponiendo su borrado.

Tablas temporales

Sus **datos** duran lo mismo que **la transacción o sesión actual**; mientras que la **definición de sus metadatos** son **permanentes**, lo que provoca que puedan **crearse vistas, triggers, índices... sobre tablas temporales**.

Estos **datos son privados** de cada sesión, por lo que **no pueden compartirse**.

Oracle optimiza este tipo de tablas priorizándolas en la caché, haciéndolas rápidas.

```
CREATE GLOBAL TEMPORARY TABLE tabla
ON COMMIT [DELETE|PRESERVE] ROWS;
```

- DELETE: borra el contenido de la tabla tras el *commit*.
- PRESERVE: contrario a DELETE, borra la tabla con la sesión.

Tablas externas

- Accede a datos en fuentes externas usando SQL, PL/SQL o Java.
- Se puede crear una tabla externa, copiar el archivo en la ubicación especificada en la definición de la tabla, y utilizar SQL para consultar los registros en el archivo de texto.
- Se usan mucho para Data Warehouse en operaciones de extracción, transformación y carga.
- Se usan para leer datos, aunque con drivers adecuados también para escribir.

Tablas organizadas por índice (IOT)

```
CREATE TABLE (...) ORGANIZATION INDEX;
```

Tablas en las que los datos están contenidos en el índice asociado (*B-Tree*).

- Las modificaciones en los datos actualizan el índice.
- Realmente, el índice es la tabla:
 - En de estar compuesto por valor de clave y puntero (*ROWID*)...
 - ...está compuesto de valor de clave y el resto de valores de las filas.
 - *No duplica el almacenamiento de claves como en una tabla ordinaria con su índice.*
- Idóneas solo para acceso por claves primaria.
- Pueden crearse índices adicionales sobre ese tipo de tablas para acceder eficientemente por otras columnas.

Principales diferencias entre estas tablas y las ordinarias:

Tabla ordinaria	Tabla organizada por índice
ROWID identifica una fila	La clave primaria identifica una fila
PK opcional	PK obligatoria
Acceso por ROWID	Acceso por PK
Análisis secuencial para recuperar todas las filas	Un análisis completo del índice recupera todas las filas ordenadas por la PK
Almacenables en un clúster	No son almacenables en un clúster
Pueden contener los tipos LONG y LOB	Pueden contener el tipo LOB

Índices

Estructura opcional asociada con una tabla o un clúster.

- Se crean sobre una o varias columnas para acelerar la ejecución de sentencias.
 - Tras cada columna puede especificarse ASC o DESC.
- Una tabla puede contener cualquier cantidad de índices, siempre y cuando la combinación de columnas sea diferente.
- Pueden usarse varios tipos de índices, permitiendo una funcionalidad distinta frente al rendimiento de la BD:
 - Árboles binarios sobre las tablas.
 - Árboles binarios sobre los clusters.
 - índices hash sobre los clusters.
 - índices de clave inversa.
 - índices de mapa de bits.
- Son lógica y físicamente independientes de los datos de las tablas.
- Se mantienen automáticamente de forma dinámica.
 - Esto permite borrar índices sin efectos colaterales.
- Pueden ser únicos (UNIQUE) o no (NONUNIQUE), según lo exijan las columnas.
 - Si esa restricción existe, Oracle crea un índice único automáticamente.
 - Oracle no recomienda crear índices únicos explícitamente.
- Se recomienda:
 - Que sean únicos.
 - Al menos uno por tabla.
 - Uno por columna buscada con frecuencia.
- Pueden crearse índices sobre una función determinista definida por el usuario.

Cuando se crea un índice

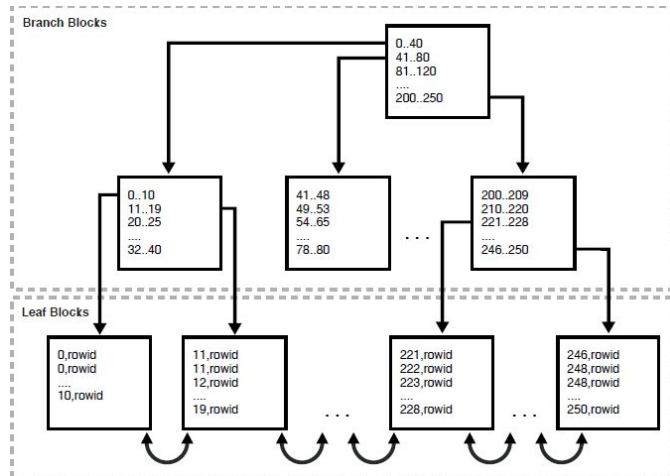
- Se le asigna un segmento de índice para contener sus valores en el espacio de tablas.
 - Se recomienda que estén en un espacio de tablas diferente al de la tabla y en otro disco para que Oracle pueda leerlos en paralelo.
- Oracle ordena las columnas del índice al crearlo y almacena el valor de los índices junto con el ROWID de las filas.

- Los índices pueden crearse en orden ascendente / descendente y comprimidos o no.

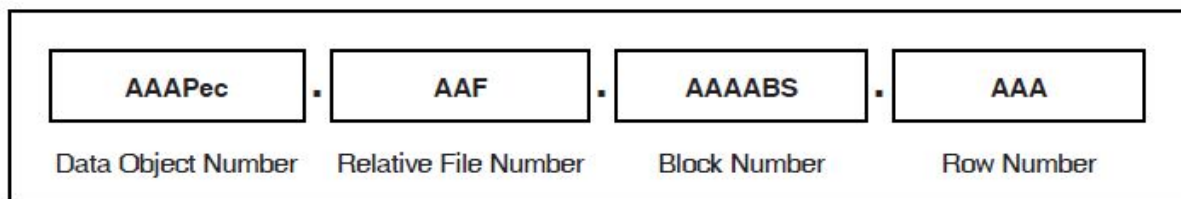
Árboles binarios

Están compuestos de:

- **Branch Blocks** (arriba), para la búsqueda.
- **Leaf Blocks** (abajo), para almacenar los datos.



Formato del ROWID



- **Data Object Number:** Identifica el segmento.
- **Relative File Number:** Identifica el *datafile* dentro del espacio de tablas.
- **Block Number:** Número del bloque de datos dentro del *datafile*.
- **Row Number:** Fila dentro del bloque de datos.

Índices de mapa de bits

CREATE BITMAP INDEX

- Para cada valor de clave se usa un mapa de bits en lugar del ROWID.
- Cada bit del mapa corresponde con un ROWID: 1 si la fila está y 0 si no.
 - Una función de mapeo convierte la posición de un bit en un ROWID.

CODIGO	E CIVIL	...	PROVINCIA
101	Soltero	...	Málaga
102	Casado	...	Madrid
103	Soltero	...	Barcelona
104	Divorciado	...	Barcelona
105	Soltero	...	Madrid
106	Casado	...	Madrid

VALOR	MAPA DE BITS					
Barcelona	0	0	1	1	0	0
Madrid	0	1	0	0	1	1
Málaga	1	0	0	0	0	0

- Si el número de valores que no son clave es pequeño -es decir, la cardinalidad de la columna es baja-, los índices de mapas son muy eficientes.
 - Para columnas con valores repetidos más de 100 veces.
 - Para columnas donde la definición del índice responde a numerosas condiciones de una cláusula WHERE, porque las filas que no satisfacen alguna condición ya son desconsideradas antes de acceder a la tabla.
- Un índice no puede ser BITMAP y UNIQUE a la vez.

Uso de índices

- Oracle usa cada índice automáticamente cuando lo necesita.
 - También lo actualiza tras cada modificación en la tabla.
- Para saber si lo está usando, usar:

<pre>EXPLAIN PLAN FOR SELECT * FROM tabla WHERE UPPER(columna)='NOMBRE' ;</pre>
<pre>SELECT * FROM TABLE(dbms_xplan.display);</pre>

Plan hash value: 2998577752.

Clusters

Grupo de tablas que comparten las **mismas páginas** porque tienen alguna **columna en común** y suelen usarse juntas. Pueden ser:

- **Cluster de índice**, para la columnas de la clave del cluster.
- **Cluster Hash**: acceso a los datos mediante una función hash.
 - Esta función se aplica a la(s) columna(s) de la clave del cluster para obtener la página correspondiente a las filas de dicha clave.
 - Esto ahorra tener que leer el índice para localizar / insertar un dato, porque la función hash no requiere lecturas en disco.

Ejemplo:

*Si se construye un **cluster** con las **tablas** empleados y departamentos utilizando la **columna en común** codigo_departamento, los empleados de un mismo departamento compartirán las mismas páginas junto a los datos del departamento. Esto produce que cada clave del cluster se almacene una sola vez en la BD.*

Si cambia el valor clave del cluster para alguna fila, Oracle la realoja.

8 - Administración del SGBD de Oracle

Funciones del administrador o DBA

- **Instalación y actualización del software** de Oracle:
 - Cambiar las claves de las 2 cuentas DBA de Oracle predefinidas:
 - SYS: esquema donde se encuentran todas las tablas y vistas del diccionario de datos de la BD. No debería modificarse.
 - SYSTEM: esquema con tablas y vistas con información administrativa.
- **Evaluación del hardware.**
- Planificación de los **parámetros de creación** de la BD.
- **Creación de la BD:**
 - Definición de espacios de tablas.
 - Implementación del diseño con objetos y restricciones.
- **Modificar la estructura** de la BD cuando sea necesario.
- **Apertura y cierre** de la BD.
- **Gestión de usuarios** y sistemas de **seguridad** (permisos y roles).
- **Auditoría.**
- **Copias de seguridad y recuperaciones.**
- **Afinamiento** de la BD (optimización).

Utilidades de administración

- **SQL*Loader:** carga datos desde ficheros del SO a tablas de la BD
 - Puede usarse por administradores y usuarios.
 - Permite especificar el formato de entrada de los datos.
- **Export / Import:** permite mover datos entre distintas BDs de Oracle.
 - *Export* guarda los datos en ficheros.
 - *Import* lee datos de los ficheros y los guarda en tablas.
 - Puede usarse para *backups*.

Privilegios para la administración

Se conceden / revocan como cualquier otro permiso:

- **SYSOPER**: puede hacer casi todas las tareas de administración, excepto conceder la administración a otros, crear una BD y poco más.
- **SYSDBA**: contiene todos los privilegios del sistema `WITH ADMIN OPTION` (incluyendo **SYSOPER**) y permite usar `CREATE DATABASE`.

Un usuario puede conectarse usando: `CONNECT usuario AS SYSDBA`;

- Se conecta al esquema por defecto `PUBLIC` y `SYS`, respectivamente, no al esquema asociado a *usuario*, así que este no podrá ver sus propias tablas ni cualificarlas con su nombre de usuario.

Crear una BD de Oracle

1. **Realizar copias de seguridad de otras posibles BDs preexistentes.**
2. **Crear el fichero de parámetros**, que sirve de arranque para la instancia.

- **SPFILE** (*Server Parameter File*).
 - Creado por defecto para que se edite posteriormente.
 - Solo uno por BD y debe residir en el mismo host que la BD.
 - Leído y escrito por el SGBD.
 - Binario, no puede modificarse externamente.
 - Puede usarse esta sentencia para modificar un parámetro:

```
ALTER SYSTEM  
SET parametro=valor SCOPE=SPFILE;
```

- **PFILE** (*Text Parameters File*).
 - Fichero de texto con una lista de parámetros y sus valores.
 - Puede residir en la máquina donde se arranque el cliente de la BD.
 - Puede editarse con un editor de texto (`ALTER SYSTEM` no sirve).

Los parámetros estáticos de la BD deben modificarse en los ficheros.

- **DB_NAME**: nombre local de la BD. No puede ser cambiado.
- **DB_DOMAIN**: localización lógica de la BD en la red que, en combinación con **DB_NAME**, debe identificar un único nombre.
- **CONTROL_FILES**: fichero de control. Si no especifica nada, Oracle creará uno por defecto, aunque se recomienda tener mínimo 2.
- **DB_BLOCK_SIZE**: tamaño de página de la BD. Por defecto es del mismo tamaño que una página del SO.
- **PROCESSES**: máxima cantidad de procesos que pueden conectarse a la BD a la vez. Debe incluir los 5 de *background* y 1 por usuario.
- **ROLLBACK_SEGMENTS**: lista de los segmentos *rollback* que asigna la instancia cuando arranca la BD.

Para crear un PFILE una vez arrancada la BD con el SPFILE:

```
CREATE PFILE FROM SPFILE;
```

3. **Arrancar la instancia:** usando el comando STARTUP desde SQL*Plus.
4. **Crear la BD:** usando la instrucción CREATE DATABASE, provocando la ejecución automática de las siguientes acciones:
 - Creación de
 - *datafiles*: ALTER TABLESPACE.
 - ficheros de control: CREATE CONTROL FILE.
 - *redo log*.
 - espacio de tablas SYSTEM y su segmento de: CREATE TABLESPACE.
 - diccionario de datos.
 - los usuarios SYS y SYSTEM.
 - Especificación del conjunto de caracteres.
 - Montar y abrir la BD.
5. **Copia de seguridad** de la BD.

Algunas sentencias SQL del DBA

CREATE ROLLBACK SEGMENT	Crea un segmento de <i>rollback</i>
ANALYZE	Almacena o borra estadísticas en el diccionario de datos sobre un objeto específico
AUDIT	Empieza un seguimiento sobre las operaciones ejecutadas o sobre objetos accedidos. Los datos se guardan en tablas del diccionario con el texto con los nombres <i>DBA_AUDIT_objeto</i>
NOAUDIT	Finaliza la auditoría

9 - Iniciar / Finalizar Oracle

Inicialización (startup)

Necesaria para que el SGBD pueda utilizarse.

- **Crear una instancia:** crear el SGA y los procesos de *background* (lectura del fichero de parámetros).
- **Montar una BD:** asociar una instancia a una BD concreta.
 - Para montar la BD es necesario leer los ficheros de control.
- **Abrir la BD:** establecer la BD como disponible para sus operaciones.
 - Se produce una recuperación (*recovery*) si la BD no se cerró bien.

Finalización (shutdown)

El proceso inverso al anterior.

- **Cerrar la BD**
 - Escribir los datos de la BD en los *datafiles*.
 - Escribir los datos de recuperación en los ficheros de rehacer.
 - Cerrar los *datafiles* y los ficheros de rehacer.
 - Los ficheros de control permanecen abiertos.
- **Desmontar la BD:** cerrar los ficheros de control.
 - La SGA se mantiene en la memoria.
- **Borrar la instancia:** eliminar la SGA de la memoria.

Permisos necesarios

SYSDBA y **SYSOPER** son privilegios especiales del sistema que permiten el acceso a una instancia, incluso cuando la BD está cerrada.

- El **control** de estos privilegios se encuentra **fuera** de la propia BD.
- Una conexión con:
 - SYSDBA usa el esquema SYS.
 - SYSOPER usa el esquema PUBLIC.
 - Sus privilegios son un subconjunto de los de SYSDBA.

Apertura / Cierre con SQL*PLUS

Apertura

```
STARTUP [PFILE=nombre_archivo] [EXCLUSIVE] [PARALLEL]
[MOUNT [nombre_bd] | OPEN [opciones] [nombre_bd] | NOMOUNT];
```

Esta sentencia arranca la instancia

- **PFILE**: fichero de parámetros.
- **EXCLUSIVE**: la instancia se asociará a la BD sin permitir otras.
- **PARALLEL**: activar si van a usarse varias instancias para la BD.
- **MOUNT**: monta la BD con el nombre *nombre_bd* sin abrirla.
 - Si no se especifica un nombre, se usa el parámetro de **BD_NAME**.
- **OPEN**: monta y abre la BD con las *opciones* especificadas.
 - **READ { ONLY | WRITE [RECOVER] } | RECOVER**.
- **NOMOUNT**: No monta ni abre la BD.

Si se arranca sin montar la BD: **ALTER DATABASE nombre MOUNT;**

Si se monta la BD sin abrirse: **ALTER DATABASE nombre OPEN modo;**

- **READ ONLY**: no puede modificarse.
 - No puede ser **READ WRITE** en la sentencia de apertura.
- **READ WRITE RESETLOG**: borra toda la información del registro de rehacer.
- **READ WRITE NORESETLOG**: no borra la información del registro de rehacer.

Cierre

- **SHUTDOWN [NORMAL]**: espera que termine las conexiones y no admite nuevas.
 - Opción por defecto.
- **SHUTDOWN TRANSACTIONAL**: no permite nuevas transacciones, pero espera a la finalización de las que aún siguen activas.
- **SHUTDOWN IMMEDIATE**: cierra todas las conexiones y transacciones activas (haciendo un *rollback*), se realiza un *checkpoint* y se cierran los ficheros.
- **SHUTDOWN ABORT**: elimina la instancia sin cerrar ni desmontar la BD. Requiere hacer un *recovery* al arrancar de nuevo.
 - Solo para emergencias.