



Administración de Bases de Datos

Tema 3. Nivel Físico de una Base de Datos (Parte 1 de 2)

Índice del Tema 3

1. El SGBD ORACLE
2. Tablespaces y Datafiles
3. Vistas Dinámicas del Diccionario
4. Gestión del Espacio Lógico
 - a. Bloques de Datos, Extensiones y Segmentos
5. Estructura de la Memoria
6. Estructura de los Procesos
7. Objetos del Esquema. Tablas, Clusters, Índices
8. Administración del SGBD ORACLE
9. Herramientas
10. Iniciar/finalizar ORACLE

El SGBD ORACLE

► Estructuras básicas:

- E. Lógica: Representación de los datos y sus relaciones (esquema conceptual).
- E. Física: Almacenamiento de datos.

► Estructura Lógica de una BD Oracle:

- Objetos del esquema (*schema objects*): Definición de tablas, vistas, índices, sinónimos, procedimientos almacenados...
- Espacios de Tablas (*tablespaces*): Es un área lógica de almacenamiento.
 - Informan cómo debe ser utilizado el espacio físico de la BD.
 - Describen el almacenamiento físico, gestionando el espacio físico que usa la BD.
 - Cada BD tiene al menos un *tablespace*, aunque puede tener más para mejorar su gestión (uno para usuarios, aplicaciones, *rollback...*).
 - Cada *tablespace* pertenece sólo a una BD y se divide en 1 ó más ficheros de datos.

Bibliografía: La mejor sobre este tema es el **Manual de Oracle** de la última versión.
Concepts:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/cncpt/index.html>
Administrator's Guide:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/admin/index.html>
SQL Reference:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/index.html>

Almacenamiento

- ▶ Cuándo guardamos un dato en la base de datos ¿dónde se almacena realmente?
- ▶ Si se produce un error en la base de datos o un disco falla ¿se pierden las transacciones finalizadas?
- ▶ ¿Cómo sabe el SGBD dónde están los ficheros?
- ▶ Si se produce algún fallo ¿cómo podemos saberlo?



El SGBD ORACLE

► Tipos de Ficheros de una base de datos Oracle (Datos, Rehacer, Control y Rastrear):

- **Datos:** Existen uno o más ficheros que contienen los datos actuales. Todos los datos se almacenan en estos ficheros en un formato propiedad de Oracle de modo que no puedan ser leídos por otros programas.
- **Ficheros del Registro de Rehacer** (*redo log*): Registran los cambios efectuados, para poder efectuar operaciones de recuperación (*recovery*).
 - Cada base de datos Oracle tiene un conjunto de dos o más archivos de registro de rehacer (*redo log files*). El conjunto de archivos de registro de rehacer se conoce colectivamente como el registro de rehacer para la base de datos. La función principal del registro de rehacer es registrar todos los cambios realizados en los datos. Si un fallo impide que los datos modificados se escriban permanentemente en los archivos de datos, los cambios se pueden obtener del registro de rehacer, por lo que el trabajo nunca se pierde.
 - Para protegerse contra un fallo que involucre el registro de rehacer en sí, Oracle permite un registro de rehacer multiplexado para que se puedan mantener dos o más copias del registro de rehacer en diferentes discos. La información en un archivo de registro de rehacer se usa solo para recuperar la base de datos de un fallo de sistema o de medios que impide que los datos de la base de datos se escriban en los archivos de datos.
 - Por ejemplo, si un corte de energía inesperado termina la operación de la base de datos, los datos en la memoria no pueden escribirse en los archivos de datos, y los datos se pierden. Sin embargo, los datos perdidos se pueden recuperar cuando se abre la base de datos, después de que se restablece la energía. Al aplicar la información en los archivos de registro de rehacer más recientes a los archivos de datos de la base de datos, Oracle restaura la base de datos a la hora en que ocurrió el fallo de energía.
 - El proceso de aplicar el registro de rehacer durante una operación de recuperación se denomina **rolling forward**.

El SGBD ORACLE

► Tipos de Ficheros (cont.):

- Ficheros de Control: Información general, como nombre de la BD, nombres de sus ficheros, sus localizaciones, fecha de creación, histórico de *backups*...
- Ficheros para Rastrear (*trace files*) y para Registrar Alarmas (*alert log*): Se registran las operaciones por las que han pasado determinados procesos y los eventos importantes acaecidos a la BD.

El SGBD ORACLE

▶ Mecanismos para almacenar los ficheros:

▶ Mediante el S.O. (el más usado):

- ▶ Los archivos se almacenan en un sistema de archivos
- ▶ El S.O. asigna y libera espacio en disco en los archivos.
- ▶ Cada archivo tiene un nombre y se ofrece como un espacio de direcciones contiguo.
- ▶ La base de datos puede crear, leer, escribir, cambiar el tamaño y eliminar archivos.

▶ Oracle Automatic Storage Management (Oracle ASM):

- ▶ Es un gestor de volumen y proporciona un sistema de archivos diseñado exclusivamente para uso de la base de datos

▶ Raw device:

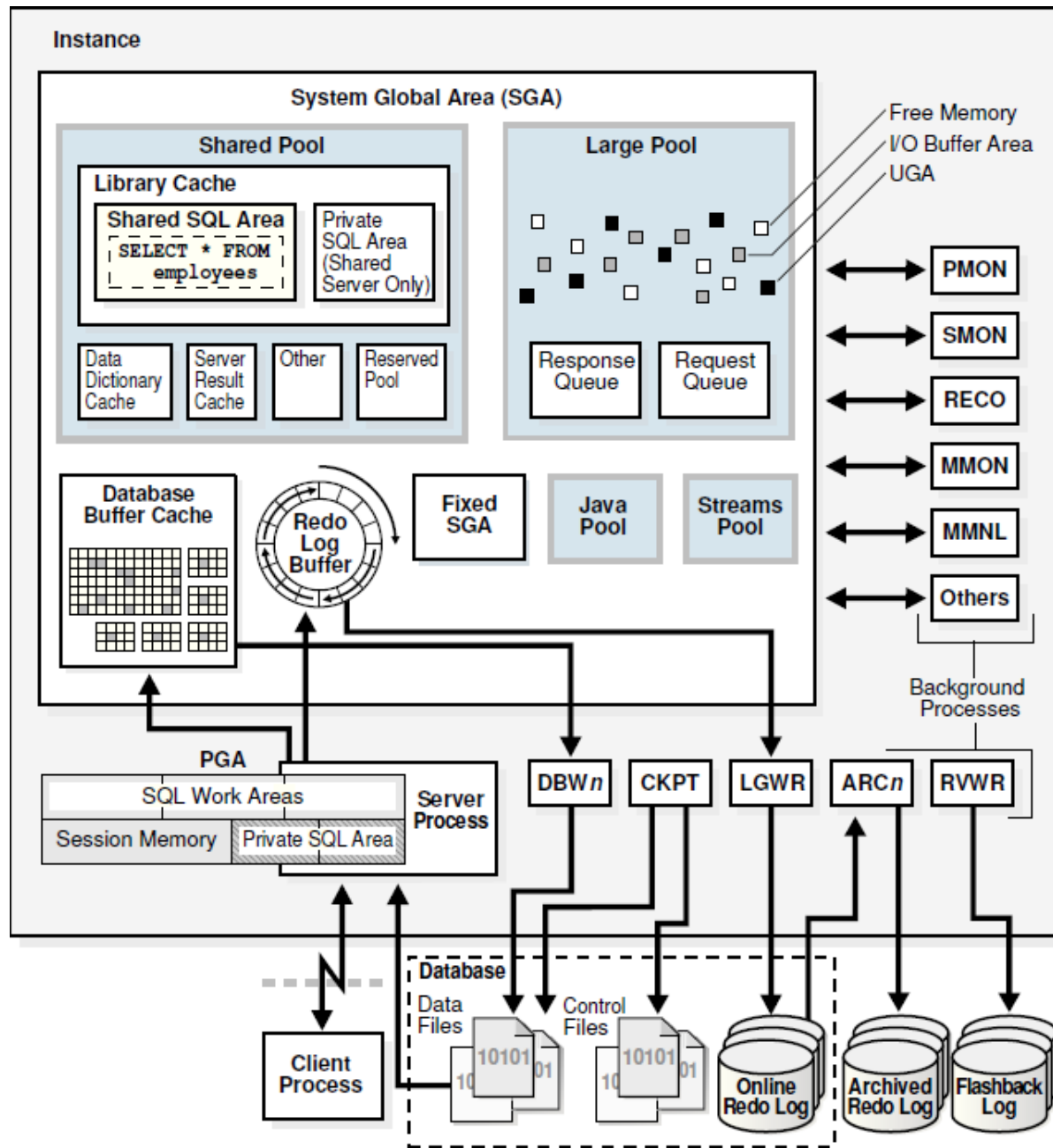
- ▶ Son particiones de disco o volúmenes lógicos no formateados con un sistema de archivos.

▶ Cluster File System:

- ▶ Varias computadoras comparten el almacenamiento de archivos, manteniendo consistente la distribución del espacio y el contenido del archivo.
- ▶ Si un equipo del cluster falla, el sistema de archivos sigue disponible.

▶ Combinaciones de las anteriores

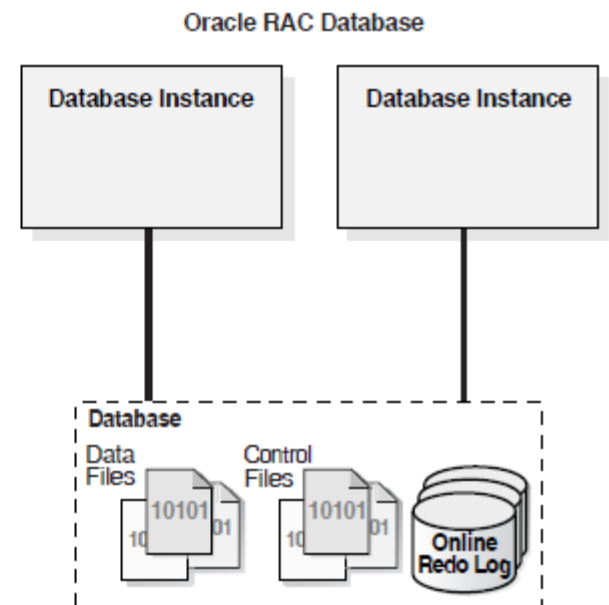
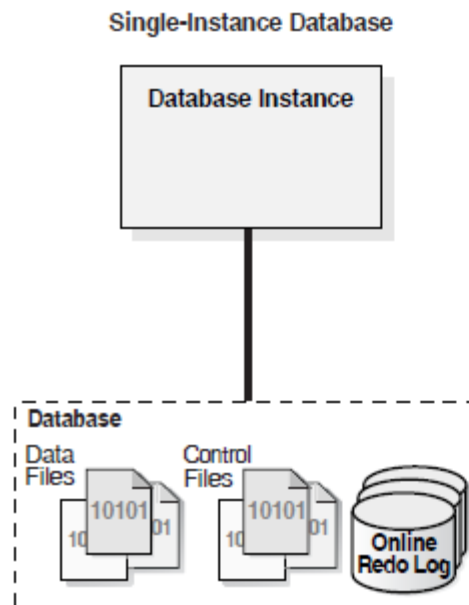
Instancias de ORACLE



► Para acceder a los ficheros se necesita una instancia

Instancias de ORACLE

- ▶ **Instancia o Servidor de BD:** Conjunto de estructuras de memoria y procesos que acceden a los archivos de una BD. Distintas instancias *pueden* acceder a la misma BD. Pero 2 BD no pueden ser montadas sobre una instancia (Relación 1 a N).
- ▶ A Partir de Oracle 12, una instancia puede ser de solo lectura → INSTANCE_MODE → READ_ONLY | READ_WRITE



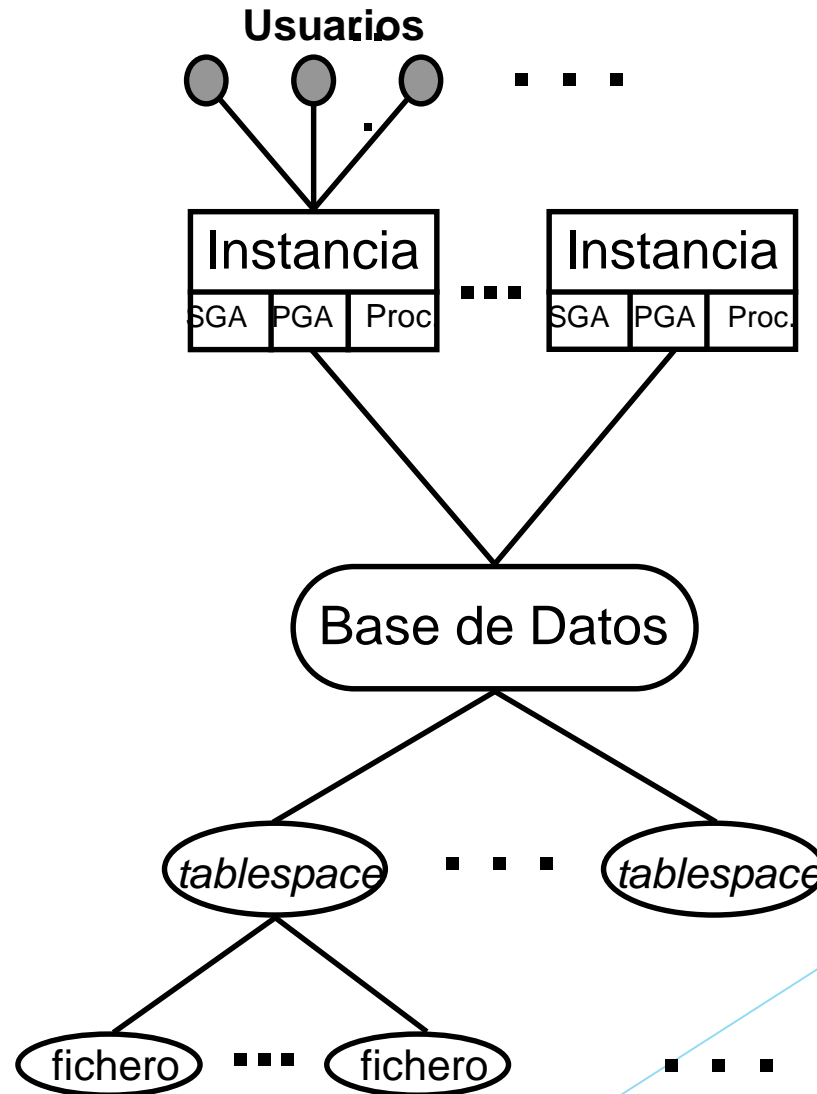
Instancias de ORACLE (Oracle Instance)

► Partes de una instancia:

- System Global Area (SGA): área de memoria con la información de la BD que pueden compartir los usuarios. Se crea cuando se empieza a usar una BD concreta. Puede verse alguna información, usando el comando **SHOW SGA**. Está compuesta por:
 - **Caché de BD**: Con los bloques de BD más recientemente accedidos, para reducir los accesos a disco.
 - **Buffer del Registro de Rehacer (Redo Log Buffer)**, para el fichero de *redo log*.
 - **Memoria compartida**: Para consultas SQL y otros procesos.
- Program Global Area (PGA): *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 (*background*), para tareas de registro, monitorización...

Instancias de ORACLE (Oracle Instance)

- Para conectarse a la base de datos, los usuarios utilizan la instancia
- La instancia se monta sobre una base de datos
- La base de datos consta de Tablespaces que a su vez, se almacenan en ficheros

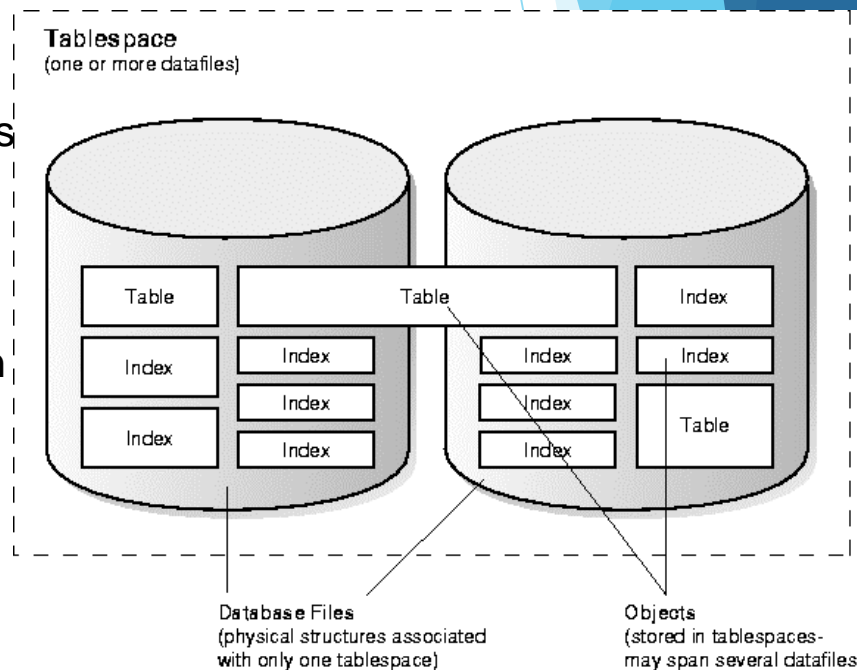
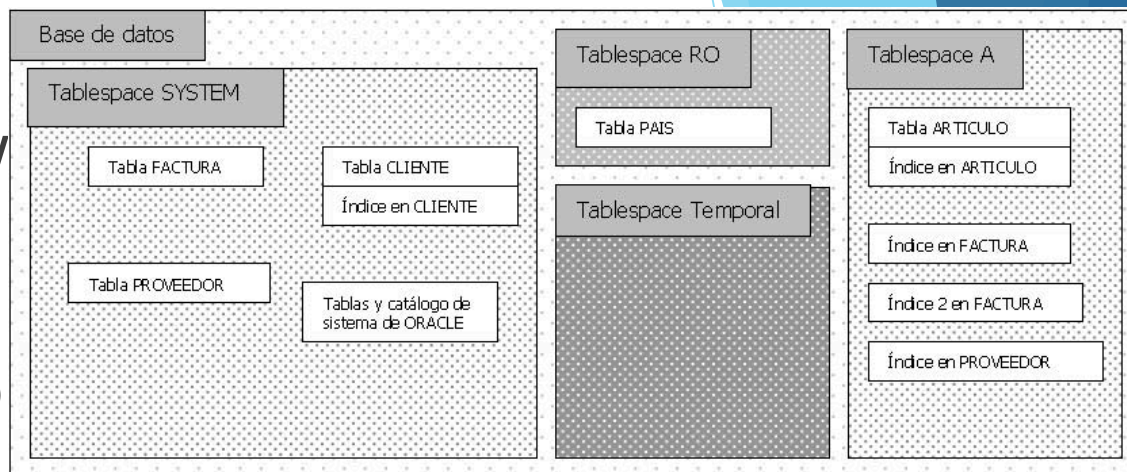




2. Tablespaces y Datafiles

Tablespaces y Datafiles

- ▶ Oracle almacena los datos lógicamente en *tablespaces* y físicamente en *datafiles*.
- ▶ Tablespaces:
 - ▶ Una BD está dividida en uno o más *tablespaces*
 - ▶ El Administrador los usa para:
 - Controlar la **creación de espacios en disco** para los datos de la BD.
 - Asignar **cuotas específicas para los usuarios**.
 - Controlar la **accesibilidad** de los datos (poniendo un *tablespace* en modo *online/offline* o *read-only/read-write*).
 - Realizar operaciones parciales de **backups/restore**.
 - Repartir los datos en **varios discos** para mejorar el **rendimiento**.



Tablespaces y Datafiles

- **Tablespaces**

El **Administrador** puede: crear y borrar *tablespaces*, añadir ficheros a los *tablespaces*, añadir o alterar segmentos del *tablespace*, hacer que un *tablespace* sea temporal o permanente...

Podemos **Crear *Tablespaces*** temporales o no con la instrucción SQL:

```
CREATE TABLESPACE tablespace TEMPORARY / PERMANENT...
```

También podemos **Cambiar** el estado de un ***Tablespace*** con:

```
ALTER TABLESPACE tablespace TEMPORARY;
```

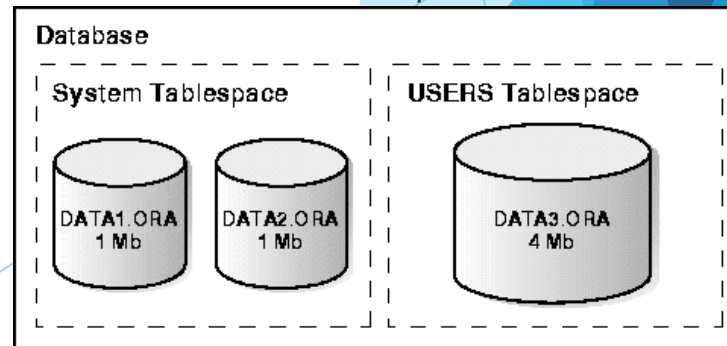
Borrar un *tablespace*:

```
DROP TABLESPACE prueba;
```

Tablespaces y Datafiles

- ▶ El *Tablespace* SYSTEM: Cada BD Oracle contiene un *tablespace* llamado SYSTEM, creado automáticamente al crear la BD.
 - ▶ Contiene las tablas del Diccionario de Datos para toda la BD.
 - ▶ Una BD pequeña puede necesitar sólo el *tablespace* SYSTEM.
 - ▶ Sin embargo, es recomendable crear, al menos, un *tablespace* adicional a fin de separar la información del diccionario de los datos.
 - ▶ Más flexibilidad en las tareas de administración
 - ▶ Reduce los problemas del acceso concurrente al mismo *tablespace*.
 - ▶ Evidentemente, el *tablespace* SYSTEM está siempre *online* mientras la base de datos esté abierta.
- ▶ *Tablespaces Read-Only*: Eliminan la necesidad de realizar *backups* y recuperaciones de porciones de la base de datos.
 - ▶ Oracle nunca actualiza los ficheros de un *tablespace read-only* y, por tanto, estos *tablespaces* pueden residir en dispositivos de sólo lectura, como CD-ROM.

ALTER TABLESPACE nombre READ ONLY



Tablespaces y Datafiles

► Tablespaces Online y Offline:

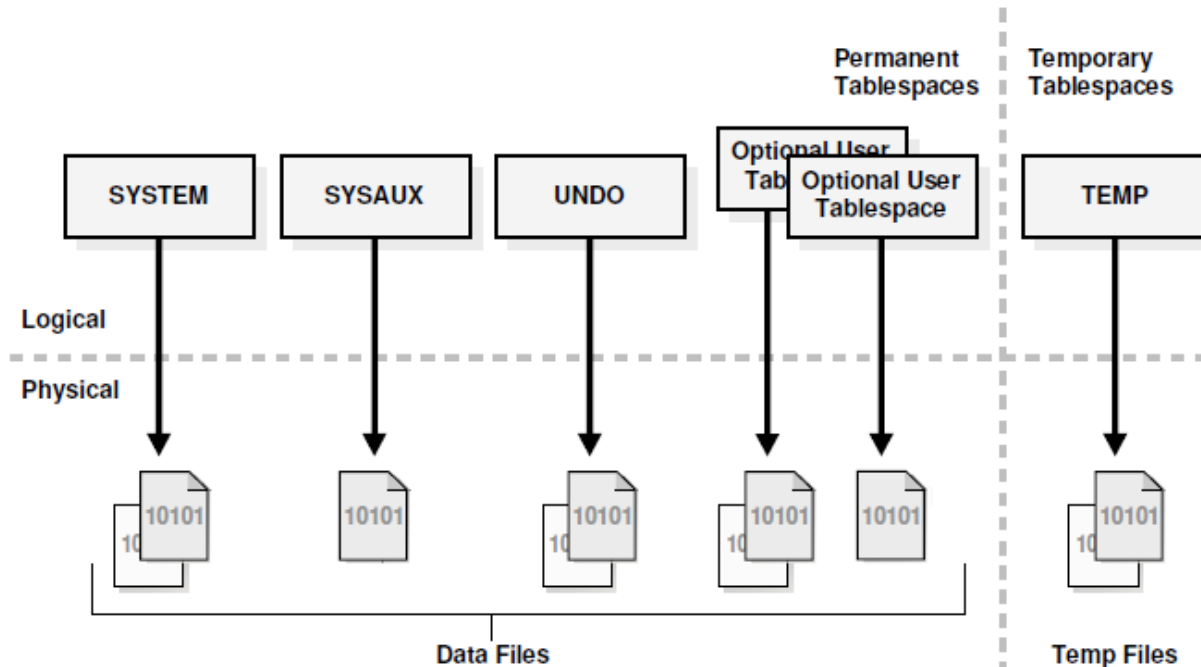
Un Administrador de la BD puede poner a los *tablespaces* accesibles o no, mientras la BD está abierta.

- Normalmente, un *tablespace* está **online** para que los usuarios tengan accesibles los datos.
- Los motivos por los que un Administrador puede poner **offline** a un *tablespace* pueden ser:
 - Para dejar inaccesible una porción de la BD mientras el resto sigue accesible.
 - Para realizar un **backup offline** del *tablespace*.
 - Para hacer que algunas aplicaciones y algunas tablas queden temporalmente inaccesibles al objeto de realizar operaciones de mantenimiento o modificación.
- No se pueden poner **offline** *tablespaces* que contengan segmentos de *rollback* que estén en uso, ni el *tablespace* **SYSTEM**.
- Oracle puede poner un *tablespace* **offline** si se ha producido algún error importante (como un error de disco...).


Tablespaces y Datafiles

► Tablespaces Temporales:

- Se puede utilizar espacio para la realización de operaciones de ordenación (*sort*) de forma más eficiente creando *tablespaces* temporales para este exclusivo uso.
- Un *tablespace* temporal puede ser utilizado únicamente para contener segmentos *sort*.
- Objetos permanentes **no** pueden residir en un *tablespace* temporal.
- A partir de Oracle 12, se pueden definir tablespaces temporales locales a cada instancia



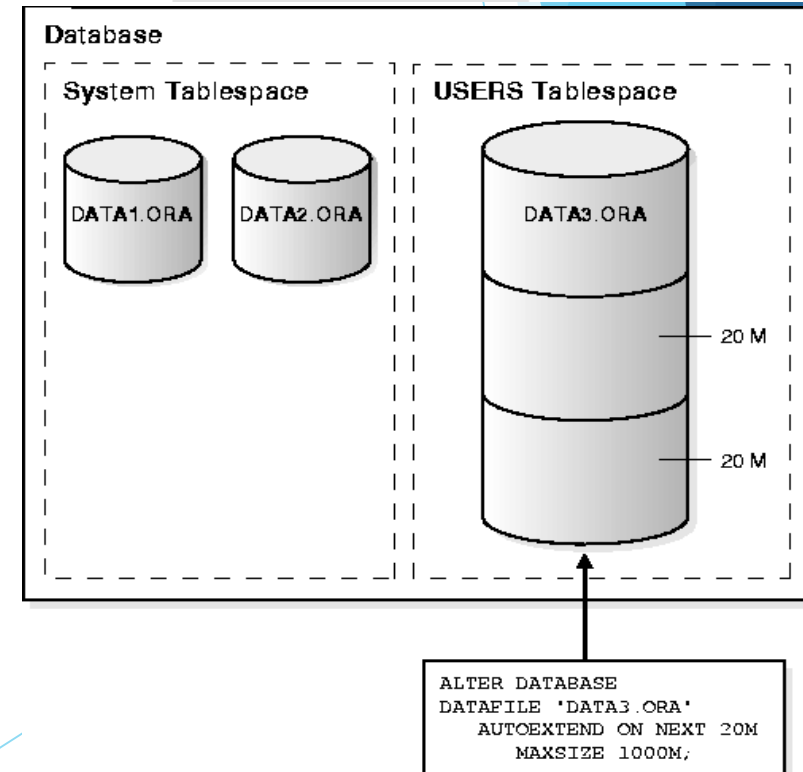
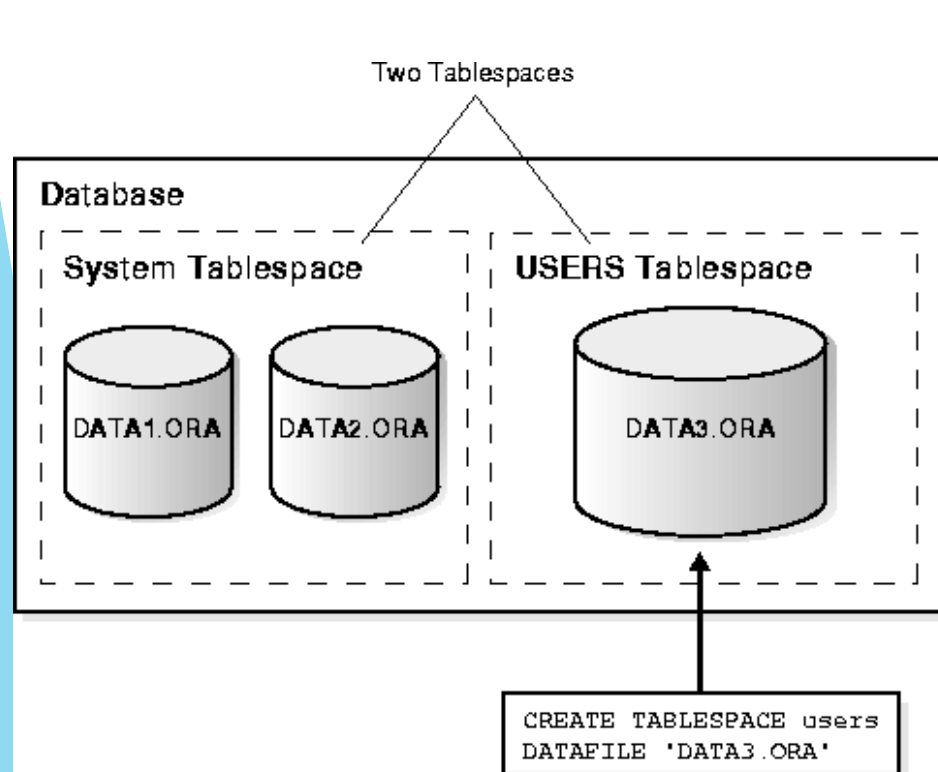
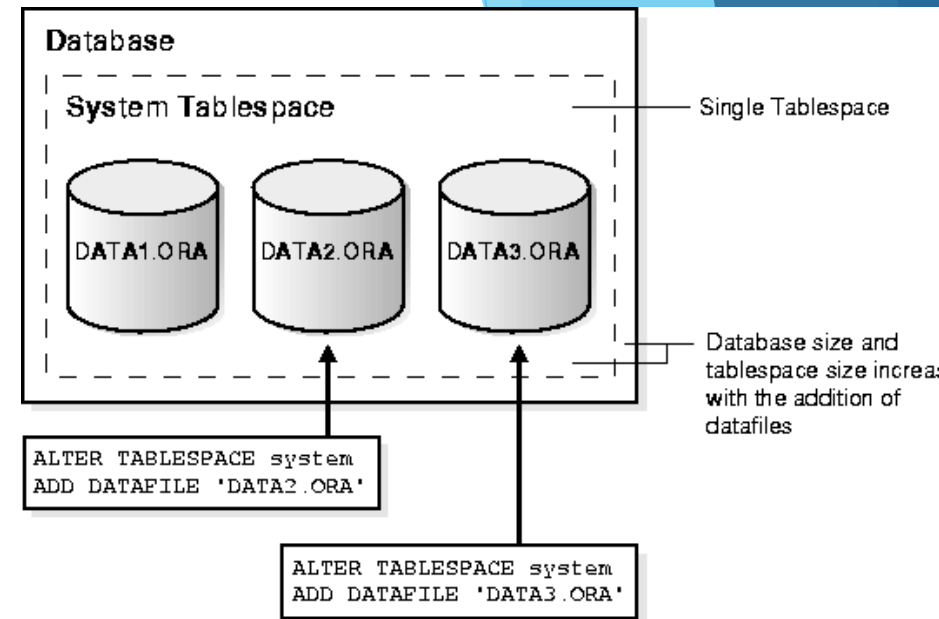
Tablespaces y Datafiles

- ▶ Un tablespace de una BD consiste en uno o más datafiles:
 - ▶ Un *datafile* puede estar asociado con un único *tablespace*.
 - ▶ El primer *tablespace* en cualquier BD Oracle siempre es el *tablespace* SYSTEM y para éste se construye el primer *datafile* cuando creamos la base de datos.
- ▶ Contenido del Datafile:
 - ▶ Cuando un *datafile* se crea, su espacio es formateado para que pueda contener los datos de usuario.
 - ▶ Los datos asociados con los objetos del esquema serán almacenados físicamente en *datafiles*, pero debe tenerse en cuenta que no existe una correspondencia directa entre estos objetos y los *datafiles*.
 -  ▶ Un objeto no se corresponde con un *datafile*, sino con un *tablespace*.
- ▶ Tamaño de los datafiles:
 - ▶ Podemos alterar el tamaño de un *datafile* después de haber sido creado.
 - ▶ Esto permite facilitar las tareas de administración de la base de datos.
 - ▶ La instrucción correspondiente está dentro de **ALTER DATABASE**.

Tablespaces y Datafiles

► Podemos hacer Creacer la BD de tres maneras:

1. Añadiendo un *datafile* al *tablespace*
2. Añadiendo un nuevo *tablespace*
3. Incrementando el tamaño de un *datafile*





3. Vistas Dinámicas del Diccionario



Vistas Dinámicas del Diccionario

► Tablas de Ejecución Dinámica:

- 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 consultar y crear vistas sobre esas tablas (llamadas *fixed views*) y autorizar el acceso a esas vistas por parte de otros usuarios.
- El propietario es **sys** y los nombres de las mismas comienzan por **v_\$**. Para estas tablas se crean vistas y para ellas se crean sinónimos que comienzan todos por **v\$**.

❑ ¿Por qué se llaman *fixed views*?

- ❑ porque un administrador de base de datos no puede modificarlas ni eliminarlas.

Vistas Dinámicas del Diccionario

► Ejemplos:

- **V\$DATABASE** contiene información sobre la BD (nombre...).
- **V\$DATAFILE** contiene información sobre los *datafiles*.
- **V\$FIXED_TABLE** contiene información sobre todas las tablas de ejecución dinámica y vistas de la BD.
- **V\$PROCESS** contiene información sobre los procesos activos.
- **V\$CONTROLFILE** contiene una lista con los ficheros de control.
- **V\$LOGFILE** ficheros de REDO. **V\$LOG** estado de los ficheros
- **V\$SESSION** contiene información sobre las sesiones actuales (usuario, comando actual, programa y terminal de acceso...).

Para matar una sesión:

```
ALTER SYSTEM KILL SESSION 'sid,serial#';
```

- **Generar las instrucciones para matar las sesiones de un usuario denominado ALUMNOBD1**





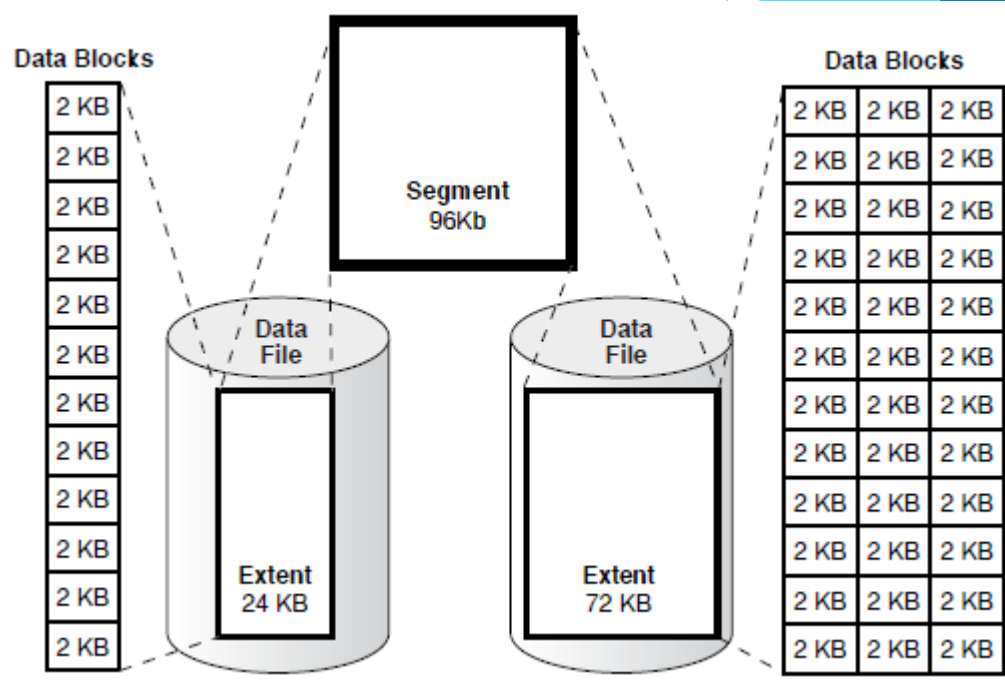
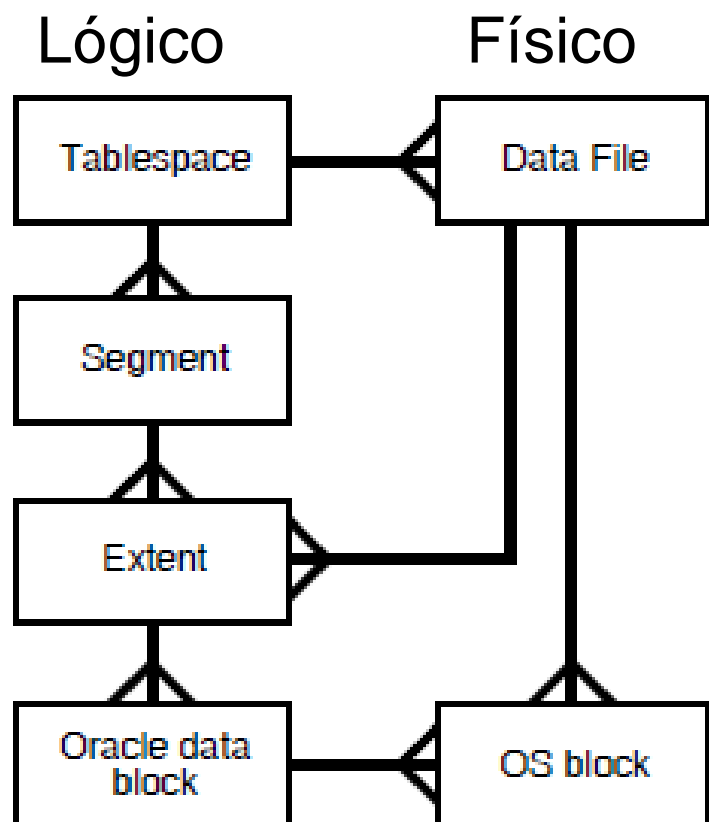
4. Gestión del Espacio Lógico

Bloques de Datos, Extensiones y Segmentos

Gestión del Espacio Lógico

- ▶ El SGBD gestiona las extensiones dentro de un Tablespace (localizar una libre o liberar una que ya no se usa)
 - ▶ Tablespaces Gestionados Localmente (por defecto)
 - ▶ Automático, **Automatic Segment Space Management (ASSM)** → Sólo hay que configurar un parámetro, PCTFREE
 - ▶ Manual (MSSM) → PCTFREE, PCTUSED, FREELISTS, y FREELIST GROUPS
 - ▶ Gestionados por el diccionario. En el propio diccionario se guardan las extensiones libres y utilizadas → Se produce SQL recursivo, lo cual es menos eficiente

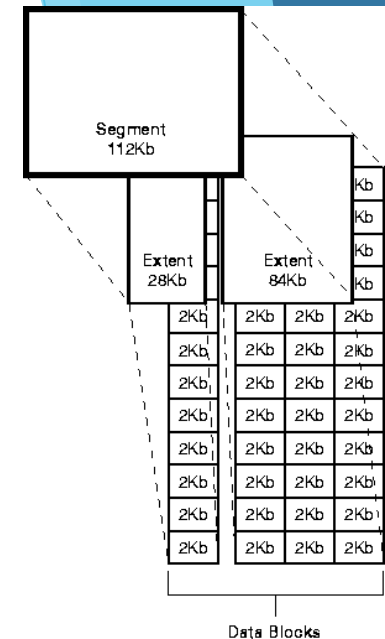
Almacenamiento Lógico vs Físico



- Cuando un fichero tiene que crecer lo hace en una extensión

Bloques de Datos, Extensiones y Segmentos

- ▶ **Bloque de Datos o Página (*data blocks*):** Unidad mínima de asignación de espacio en la Base de Datos.
 - ▶ Es la menor unidad de E/S que puede utilizar la BD (independientemente de que el tamaño de bloque del S.O. sea menor).
- ▶ **Extensión (*extent*):** Conjunto de **data blocks** *contiguos*, con un tipo de información específico.
- ▶ **Segmentos (*segment*):** Conjunto de extensiones que almacenan un determinado tipo de datos.
 - ▶ Al crear una estructura de datos, Oracle le asigna un **segmento** con una única **extensión**.
 - ▶ Cuando se llena esa extensión se le asignan otras **extensiones** a ese **segmento**. Por eso, las distintas extensiones no suelen ocupar espacios consecutivos, como sería deseable.
 - ▶ Un **segmento** completo se almacena en un *tablespace*, que puede distribuir sus **extensiones** en distintos ficheros.
 - ▶ Cada **extensión** se almacenará siempre en un único fichero.



Bloques de Datos

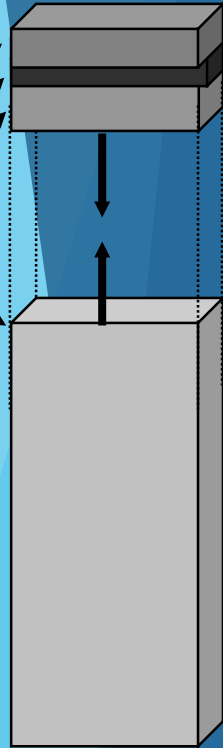
► Tamaño de Datablock:

- Especificado por el DBA cuando crea la BD.
- Debe ser múltiplo del tamaño de bloque del S.O.
- No se puede cambiar a no ser que se cree la BD de nuevo
- Se pueden crear Tablespaces con un tamaño de bloque distinto

► Formato de un Datablock:

- Overhead: Formado por 3 zonas:
 - *Block Header*: Información general, como la dirección en el disco, el tipo de segmento al que pertenece (de datos, de índice, de rollback...).
 - *Transaction entries*: También se guarda aquí información sobre las transacciones (**INSERT**, **UPDATE** y **DELETE**) sobre las filas de esta página
 - *Table Directory*: Información sobre las tablas con filas en este datablock.
 - *Row Directory*: Información sobre las filas almacenadas en este datablock.
- Row Data: Zona con datos de tablas o índices. Una fila puede estar en varias páginas.
- Free Space: Espacio libre para insertar nuevas filas (**INSERT**) o nuevos valores en las filas ya existentes (**UPDATE**), si requieren más espacio.

Common and Variable Header
Table Directory
Row Directory
Free Space
Row Data

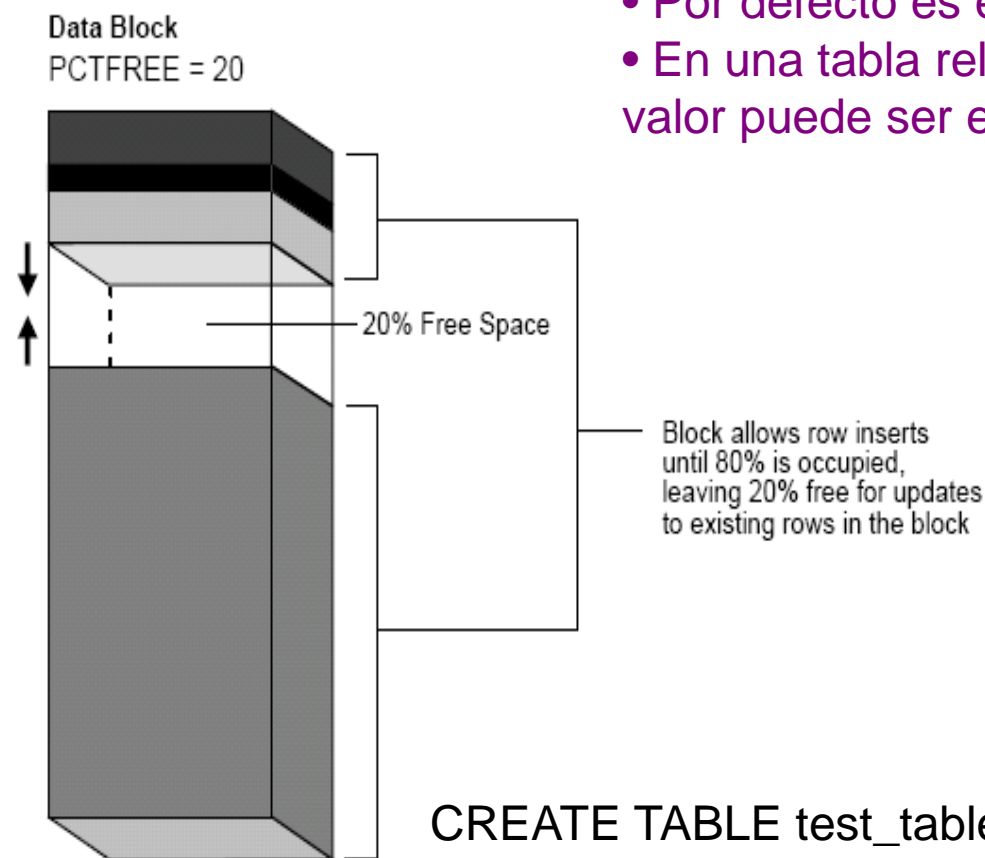


Bloques de Datos

- Control del Espacio Libre para Inserción y Actualización de Filas: Existen dos parámetros que se especifican cuando se crea o altera una tabla o un índice:

PCTFREE: Mínimo porcentaje de página que se reserva como espacio libre para futuras actualizaciones de filas que ya existen en la página.

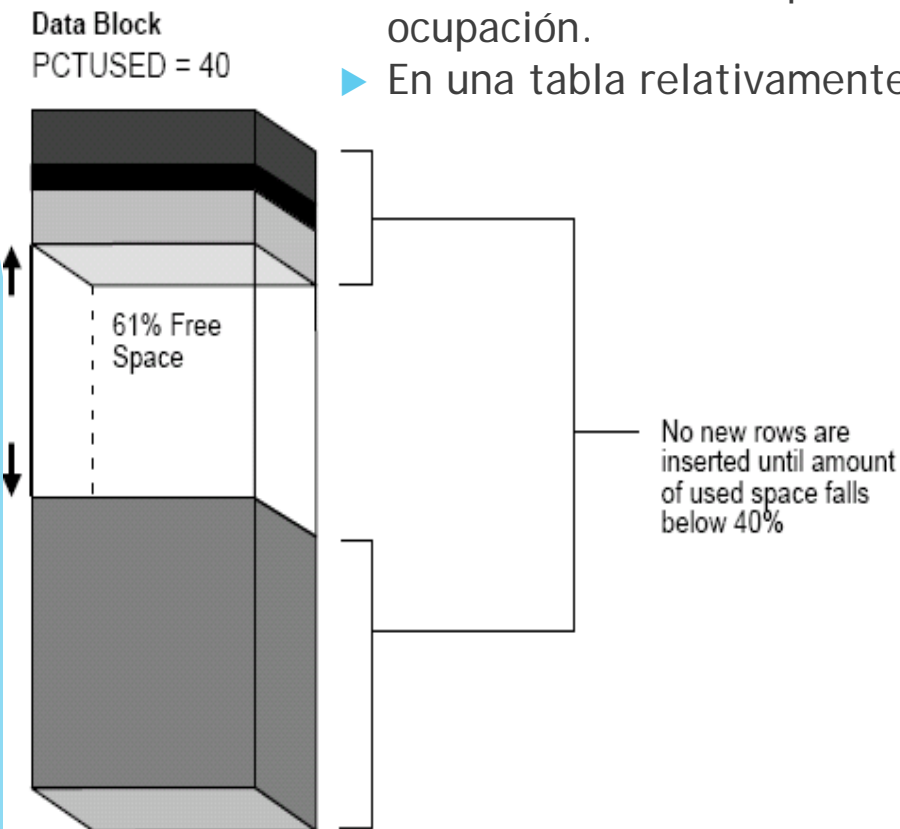
- Por defecto es el 10%.
- En una tabla relativamente estática un buen valor puede ser el 5%.



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

Bloques de Datos

- ▶ **PCTUSED** (gestión de Tablespaces manual): Después de que una página sea considerada llena en función del límite especificado en el **PCTFREE**, Oracle no vuelve a introducir ninguna nueva fila en la misma hasta que el porcentaje de página ocupada sea menor que **PCTUSED**. Hasta entonces, el espacio libre sólo será dedicado a la actualización de las filas ya existentes en la página.
 - ▶ Por defecto deja el 40%, que indica que cuando esa página se llena no volverá a estar libre para inserciones hasta que tenga menos del 40% de ocupación.
 - ▶ En una tabla relativamente estática un buen valor puede ser el 75%.

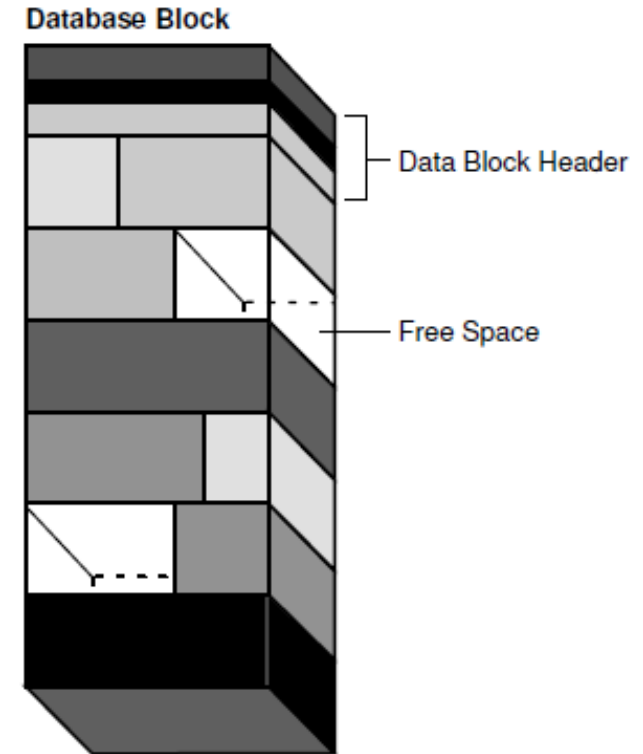


- Espacio libre para inserciones:
Tama_Página - Overhead - PCTFREE
- Cuando se produce un **INSERT**, Oracle mira la lista de páginas que están disponibles (*free list*) y selecciona la primera que encuentra.
- Para actualizaciones (**UPDATE**) cualquier espacio libre puede ser utilizado.

Bloques de Datos

► Reagrupación del Espacio Libre en un Datablock:

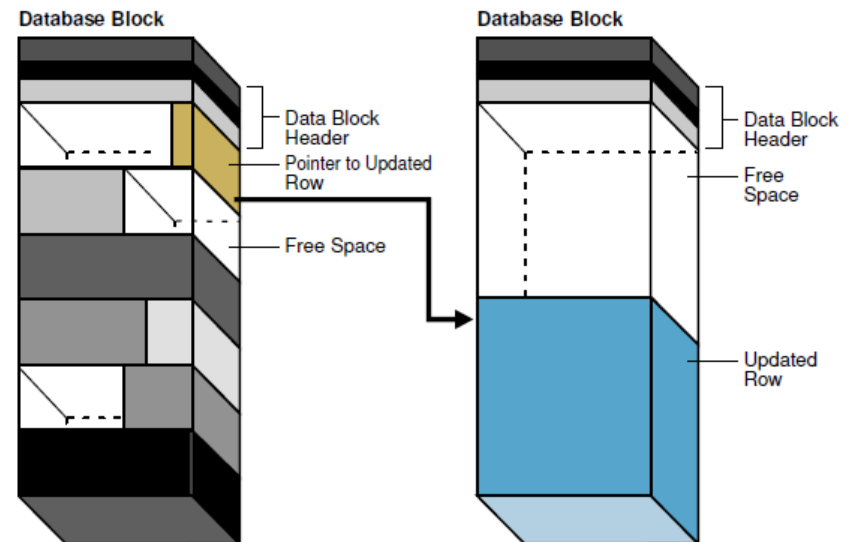
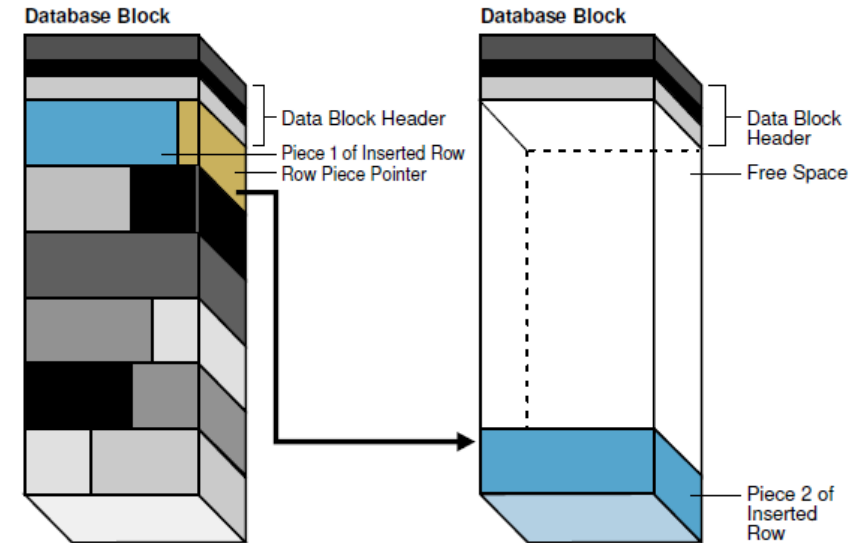
- El espacio libre aumenta por las instrucciones **DELETE** o **UPDATE** (si la actualización establece valores que ocupan menos espacio).
- Todo ese **espacio libre** podrá usarlo una instrucción **INSERT**:
 - Si la instrucción **INSERT** está en la misma transacción que la instrucción que ha generado el espacio libre (y situado después, naturalmente).
 - Si la instrucción **INSERT** está en otra transacción y la transacción que deja el espacio libre ya efectuó su **COMMIT**. Posiblemente ambas transacciones sean de distintos usuarios.



Bloques de Datos

► Encadenamiento y Migración de Filas:

- Los datos de una fila pueden ser demasiado grandes para caber en una página:
 - **La fila es muy grande:** Oracle almacena la fila en una cadena de páginas del mismo segmento.
 - Con datos grandes (como el tipo **LONG**) esta fragmentación es inevitable.
 - **Una fila es actualizada y sus nuevos valores no caben en su página actual:** Oracle traslada toda la fila a una nueva página, suponiendo que caben en una nueva página.
 - Oracle conserva la cabecera de la fila en su página inicial, apuntando a la nueva dirección en la nueva página. Así, el identificador de fila (**ROWID**) no cambia.
- Esto hace que la eficiencia al tratar esta fila sea menor, ya que Oracle debe leer más de una página para recuperar la información de esa fila.



Extensiones

▶ Un Segmento es un Conjunto de Extensiones.

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

▶ Hay Dos Formas de Gestionar las Extensiones:

- ▶ Extensiones Gestionadas Localmente (LOCAL): Al crear un *tablespace* se pueden especificar las siguientes opciones:
 - ▶ **AUTOALLOCATE:** Son gestionadas por el sistema. Se especifica el tamaño de la extensión inicial y el tamaño del resto es calculado por Oracle, con un mínimo de 64KB.
 - ▶ **UNIFORM:** El tamaño especificado es para todas las extensiones (1MB por defecto).
- ▶ Extensiones Gestionadas por el Diccionario de Datos (DICTIONARY): Utilizan como valores por defecto los valores almacenados en el Diccionario de Datos de la Base de Datos. Esos valores por defecto pueden modificarse en cualquier momento.
 - ▶ Estos valores son **INITIAL** (tamaño del primero), **NEXT** (tamaño del segundo) y **PCTINCREASE** (porcentaje de incremento en el tamaño del siguiente respecto al anterior).

Extensiones (extents)

- ▶ Eliminar Extensiones: En general, las extensiones de un segmento **no** son liberadas (*deallocated*) a no ser que borre el objeto almacenado en el segmento (mediante una instrucción **DROP TABLE** o **DROP CLUSTER**).

- ▶ No obstante, se producen algunas excepciones. Por ejemplo, el Administrador puede “desasignar” extensiones no utilizadas mediante la instrucción:

```
ALTER TABLE nombre_de_tabla DEALLOCATE UNUSED;
```

- ▶ También utilizando el paquete DBMS_SPACE_ADMIN

Segmentos

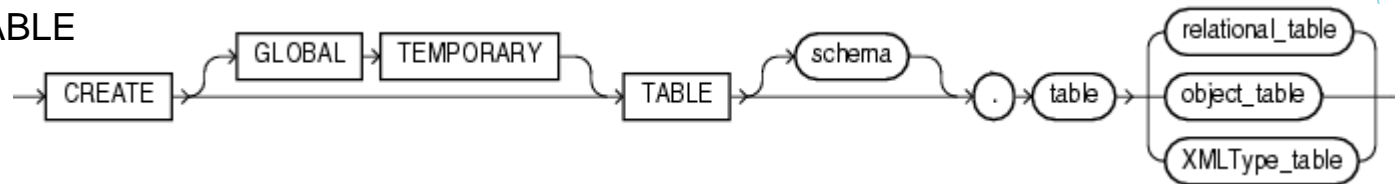
- ▶ Un Segmento es un Conjunto de Extensiones que contienen todos los datos de una estructura lógica específica (una tabla, un índice...) en un *tablespace*.
- ▶ Cuatro Tipos de Segmentos:
 - ▶ Segmentos de Datos (*data segments*)
 - ▶ Segmentos de Índices (*index segments*)
 - ▶ Segmentos Temporales (*temporary segments*).
 - ▶ Segmentos de *Rollback* (*Undo segments*).

Segmentos de datos

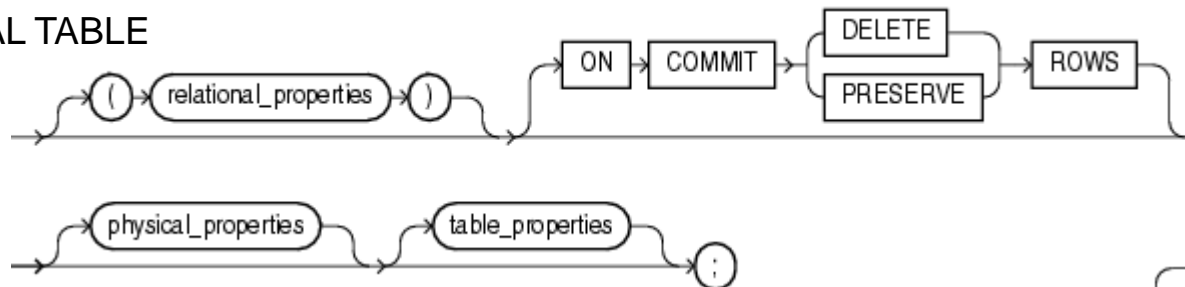
► Data segments:

- Se crean con la sentencia **CREATE** (para tablas, *snapshots*, *clusters*...). Por ejemplo, CREATE TABLE ...
- Los parámetros de almacenamiento de datablocks y extensiones se asignan con **CREATE** o **ALTER**, y afectan a la eficiencia en el almacenamiento y en la recuperación de datos.

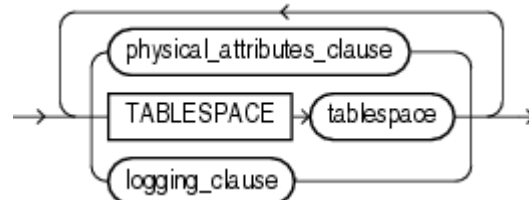
CREATE TABLE



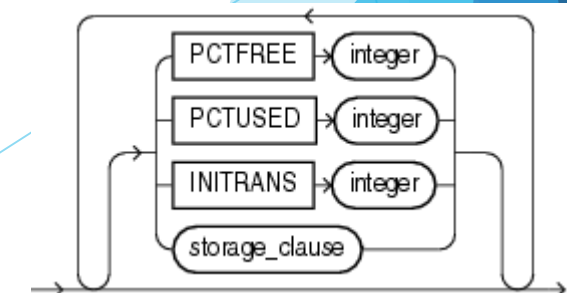
RELATIONAL TABLE



PHYSICAL PROPERTIES



PHYSICAL ATTRIBUTES CLAUSE



Segmentos de índices

▶ *Index segments:*

- ▶ La sentencia **CREATE INDEX** crea un segmento también y pueden fijarse los parámetros de almacenamiento.
- ▶ Una tabla y sus índices pueden tener segmentos en distinto *tablespace*.
- ▶ Esto permite que el índice y los datos se almacenen en distintos ficheros e incluso, en distintos discos, lo cual mejora el rendimiento

Segmentos Temporales

► *Temporary segments:*

- Cuando se procesa una consulta, Oracle requiere espacio temporal para realizar las operaciones intermedias de la instrucción SQL.
- Para ello, automáticamente crea un espacio en disco: **Segmento Temporal**.
- Operaciones que usan segmentos temporales: **CREATE INDEX**, **SELECT...ORDER BY**, **SELECT DISTINCT**, **SELECT...GROUP BY**, **SELECT...UNION**, **SELECT...INTERSECT** y **SELECT...MINUS**.
 - Normalmente solo se requiere este espacio cuando se necesita ordenar un resultado que, además, no cabe en memoria.
- Oracle crea los segmentos temporales que necesita durante una sesión de usuario en el *tablespace* temporal del usuario que realiza la instrucción.
 - Este *tablespace* es especificado en la instrucción **CREATE USER** o con **ALTER USER** y usando la opción **TEMPORARY TABLESPACE**.
 - Si el Administrador no ha definido estos parámetros, por defecto el *tablespace* temporal será el *tablespace* TEMP o el SYSTEM.
- Una vez la instrucción se completa, Oracle borra el segmento temporal asignado.
- Es razonable Crear un *Tablespace* especial para contener los segmentos temporales debido a que la creación y borrado de segmentos temporales ocurre frecuentemente → distribuir las entradas/salidas por distintos discos mejorando los tiempos de respuesta.
- También se pueden utilizar para las *tablas temporales*

Segmentos de Rollback

► *Undo segments:*

- Almacenan los viejos valores de los datos modificados por las transacciones → mantener la consistencia, realizar *rollbacks* y permitir la recuperación (*recovery*) de la BD.
 - Cada BD contiene uno o más segmentos de *rollback* → **undo tablespace**
 - Un segmento de *undo* contiene numerosos registros o *rollback entries*. Por ejemplo, qué dato ha sido modificado (fichero *filenumber* y *block ID*), así como el contenido de éste (si existía antes de realizar la operación).
 - Oracle enlaza cada entrada de una transacción de manera que todas ellas son fácilmente localizables en caso de tener que deshacerla.
 - Al grabar las **entradas de *rollback***, cambian los datablocks del segmento de *undo* y Oracle guarda todos los cambios en los datablocks, incluyendo estas entradas de *rollback*, en el **Redo Log** (registro de rehacer).
 - Este segundo almacenamiento de la información de *rollback* es muy importante para las transacciones activas (sin terminar con **COMMIT** o **ROLLBACK**).
 - Si el sistema cae, se restaura la información de los segmentos de *undo*, incluyendo las entradas para las transacciones activas.
 - Cuando se realiza un **COMMIT**, Oracle libera la información de *rollback*.



Ejercicio

1. Piensa 2 casos en los que sea necesario leer los segmentos de Rollback
2. ¿Es lo mismo un segmento de rollback que un segmento de Undo?
3. ¿Es lo mismo un segmento de rollback que el redo log?



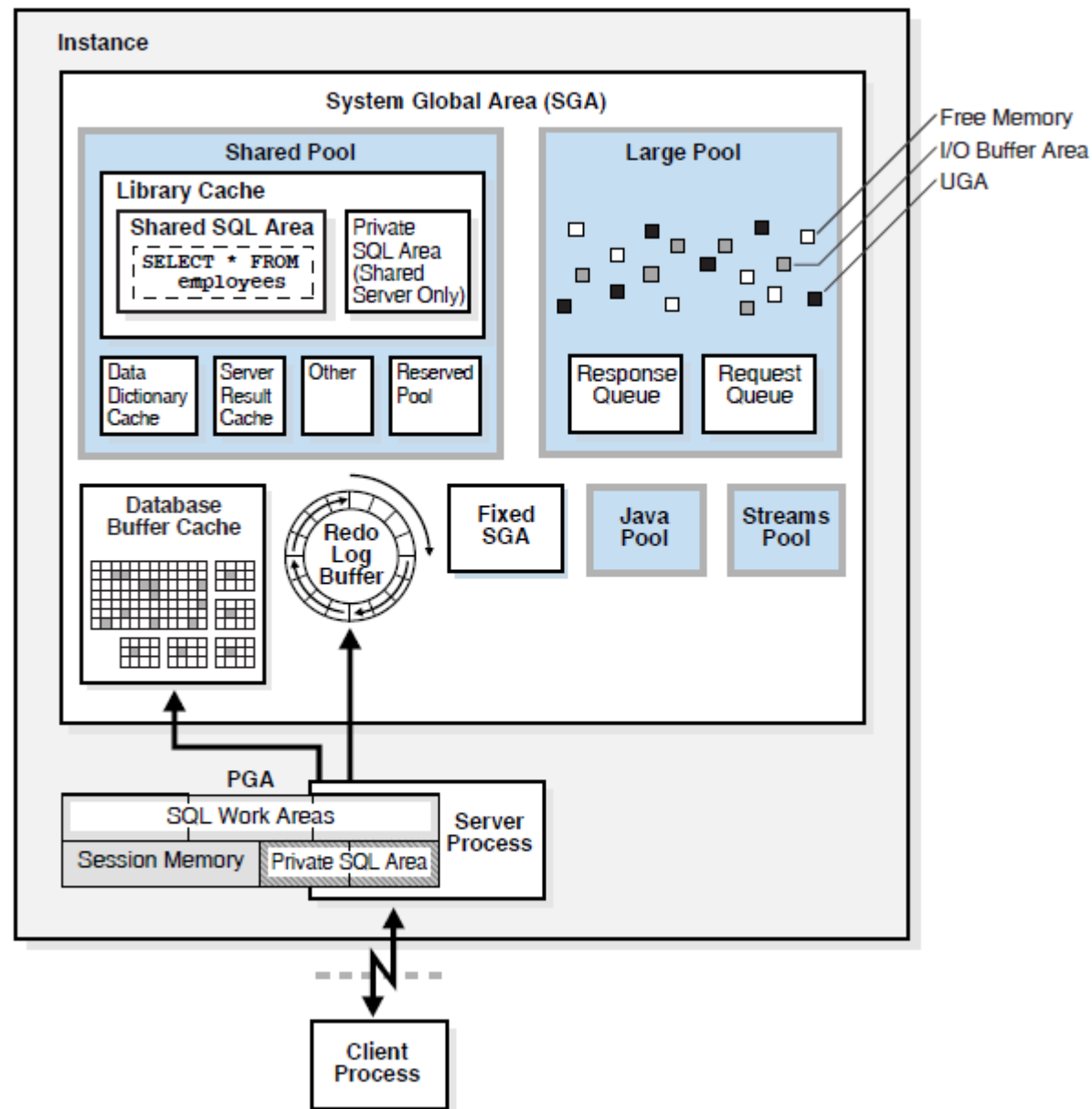
5. Estructura de la Memoria

Estructura de la Memoria en ORACLE

- ▶ 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 (estados de las consultas...).
 - ▶ Información compartida entre procesos (sobre los bloqueos...).
 - ▶ Caché de datos.

Estructura de la Memoria en ORACLE

- ▶ Estructuras Básicas de Memoria de una instancia de Oracle:
 - ▶ Área Global del Sistema (SGA, System Global Area).
 - ▶ Áreas Globales de Programas (PGA, Program Global Areas).
 - ▶ Área Global de Usuario (UGA).
 - ▶ Áreas de Código de Software (SCA, Software Code Areas).

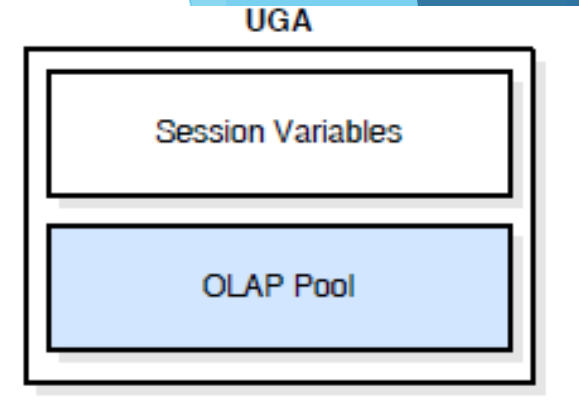


Gestión de la memoria

- ▶ La memoria que ocupa una instancia de Oracle se puede gestionar de varias formas:
 - ▶ Automática (por defecto): Se especifica el tamaño objetivo de la instancia (target size). La memoria se distribuye automáticamente entre la SGA y la PGA
 - ▶ Compartida. Se especifican los tamaños de la SGA y la PGA. Cada una se puede gestionar como un todo o individualmente
 - ▶ Manual. Se especifican los tamaños de cada parte de la instancia

User Global Area

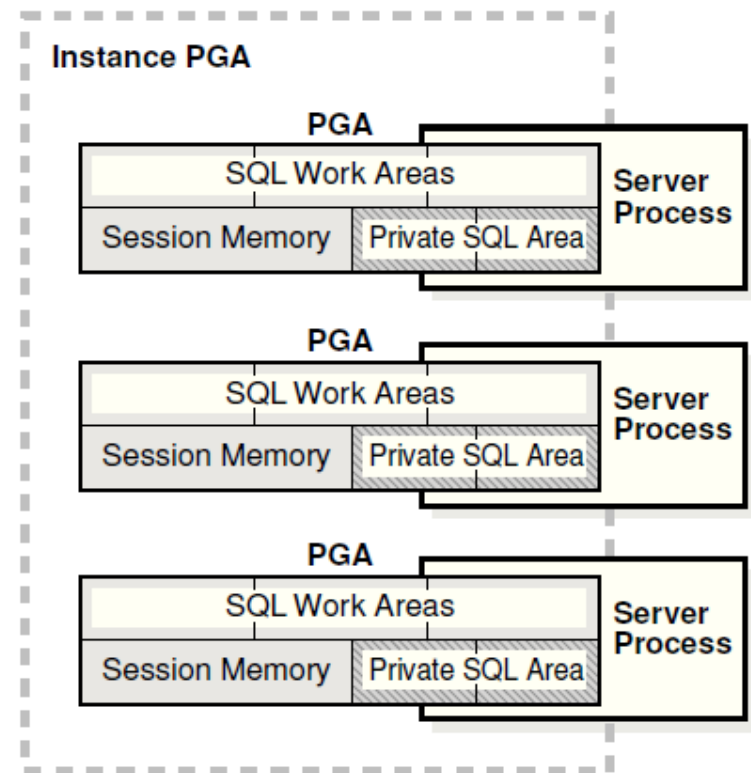
- ▶ Memoria para variables de sesión
 - ▶ Información de logon
 - ▶ Estado de la sesión
 - ▶ Variables de un paquete (package de PL/SQL)
- ▶ En OLAP (Data Warehouse) se almacenan las OLAP page pools para gestionar mejor los cubos
- ▶ La UGA debe estar disponible mientras la sesión esté abierta
- ▶ Puede residir en la SGA (servidor dedicado) o en la PGA (servidor compartido). Cuando se instala Oracle se decide si es dedicado o compartido



PGA (Program Global Areas)

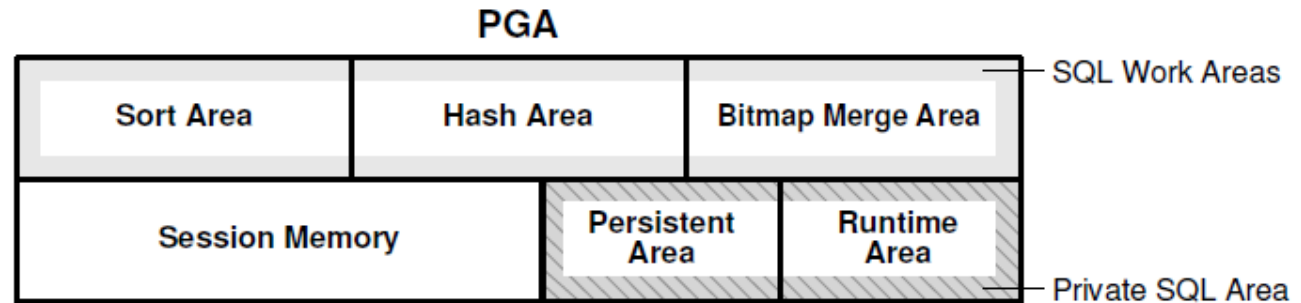
▶ Áreas Globales de Programa (PGA).

- ▶ Un PGA es una región de memoria que contiene datos e información del control para un solo proceso (de servidor o de *background*).
- ▶ Un PGA es un área de memoria no compartida en la que puede escribir un proceso.
- ▶ Analogía: Imaginaros una barra de un bar atendida por un camarero. La PGA es el trozo de la barra usado por el camarero para un cliente. El camarero es el proceso que trabaja para el cliente



PGA (Program Global Areas)

- ▶ 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**, pues, es asignado cuando un usuario se conecta a una base de datos 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 sistema operativo específico.



PGA (Program Global Areas)

► Ejemplo:

```
SELECT * FROM employees e JOIN departments d
ON e.department_id=d.department_id
ORDER BY last_name;
```

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

SGA

- ▶ Área Global del Sistema (SGA, System Global Area).
 - ▶ Un SGA es un grupo de estructuras de memoria compartida que contienen datos e información de control de una Instancia de una BD.
 - ▶ Si a una Instancia están conectados múltiples usuarios concurrentes, los datos de éstos se comparten en el SGA.
 - ▶ El SGA se crea cuando se arranca una Instancia de una BD y se destruye cuando se cierra la Instancia.
 - ▶ El SGA es de lectura y escritura, y contiene las siguientes estructuras de datos:
 - ▶ Caché de los Buffers de la BD (*Database Buffer Cache*).
 - ▶ Registro de Rehacer (*Redo Log Buffer*).
 - ▶ El "Pool" Compartido (*Shared Pool*).
 - ▶ Pool Grande
 - ▶ Pool de Java
 - ▶ Streams Pool
 - ▶ SGA fijo (*fixed SGA*): Parte del SGA con información general sobre el estado de la BD y la Instancia, a la que los procesos de "*background*" necesitan acceder

SGA

- ▶ Caché de los *Buffers* de la BD (*Database Buffer Cache*).
 - ▶ Es la parte del SGA que contiene copias de las **páginas leídas** de los *datafiles*. Todos los procesos de usuarios que están conectados a la Instancia comparten el acceso a esta caché.
 - ▶ Cuando se hace COMMIT se escriben los buffers del redo pero no los datablocks: escritura perezosa en background
 - ▶ Los *buffers* en la caché están organizados en dos listas:
 - ▶ Lista “en espera” (*write list*): Contiene *buffers* en espera (*dirty buffers*) en el sentido de que contienen datos que han sido modificados pero que aún no han sido escritos a disco.
 - ▶ Lista LRU (menos recientemente usada, Least Recently Used): La lista LRU contiene *buffers* libres, *buffers* que están siendo accedidos actualmente (*pinned buffers*) y, por último, los *buffers* “en espera” que aún no han sido movidos a la lista “en espera”.
- ▶ La primera vez que un proceso requiere un dato, se busca en la caché:
 - ▶ Si se encuentra el dato en uno de estos *buffers*, se lee directamente de la memoria (*cache hit*), acelerando el proceso de lectura.
 - ▶ Si no se encuentra (*cache miss*), entonces debe obtenerse una copia de la página en disco y pasarla a un buffer de la caché antes de proceder a leerlo.

Estructura de la Memoria: SGA

▶ Registro de Rehacer (*Redo Log Buffer*):

- ▶ Es un **buffer circular** en el **SGA** que contiene información sobre los cambios que se producen en la BD.
- ▶ Las entradas (*redo entries*) contienen información necesaria para reconstruir o **rehacer los cambios** efectuados a la BD.
- ▶ Estas entradas son copiadas por el servidor de procesos de Oracle desde el espacio de memoria del usuario hasta estos **buffers del SGA**, donde los registros van escribiéndose de forma secuencial.
 - ▶ Ahí esperan a que el proceso de *background LGWR (Log Writer)* escriba el buffer al correspondiente fichero activo (*online*) de "*Redo Log*" en disco.
- ▶ 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 del Registro de Rehacer (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:
 - ▶ `SELECT * FROM V$LOG`

Estructura de la Memoria: SGA

► El “Pool” Compartido (*Shared Pool*):

Su tamaño total está determinado por el parámetro de inicialización

SHARED_POOL_SIZE. Contiene:

► Caché de Biblioteca (*Library Cache*):

Incluye la siguiente información:

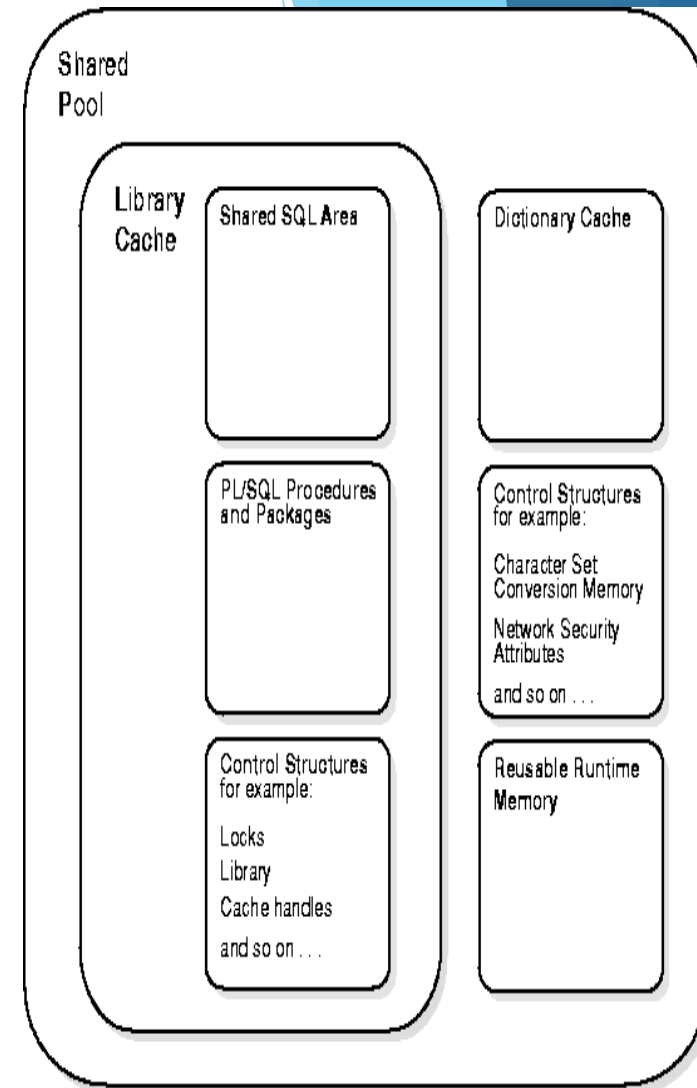
► Áreas compartidas y privadas de SQL.

- Cada instrucción de SQL ejecutada tiene un área de SQL compartida y un área privada.
- Cuando dos usuarios están ejecutando la misma declaración de SQL se reutiliza el área de SQL compartida para esos usuarios, ahorrándose tiempo y memoria (muchos usuarios, misma aplicación):
- Sin embargo, cada usuario debe tener una copia separada de la instrucción en el área de SQL privada.

► Procedimientos y paquetes PL/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 utilizan también las áreas SQL compartidas y privadas.

► Estructuras de control (caché de gestión de bloqueos, de librerías...).



Estructura de la Memoria: SGA

▶ El “Pool” Compartido (*Shared Pool*):

- ▶ Caché del Diccionario (*Dictionary Cache*): La continua necesidad de acceso al diccionario hace que Oracle disponga de una caché específica para el diccionario.
- ▶ Estructuras de Control (*Control Structures*): Juegos de caracteres, sistemas de conversión, atributos de seguridad...
- ▶ ALTER SYSTEM SET SHARED_POOL_SIZE = 64M;

▶ El “Pool” Grande (*LARGE_POOL_SIZE*).

- ▶ Estructura de memoria opcional.
- ▶ Utilidades de backup y recuperación como RMAN.

▶ El “Pool” de JAVA (*JAVA_POOL_SIZE*).

- ▶ Guarda información acerca de la interpretación de las sentencias java que se ejecutan.
- ▶ Sólo necesario si se instala y usa Java dentro de la base de datos.



Estructura de la Memoria: SGA

► Gestión Automática de la Memoria Compartida

- MEMORY_TARGET (dinámico) y MEMORY_MAX_TARGET (estático):
 - Si se fija el MEMORY_TARGET, Oracle modifica dinámicamente el tamaño de los subcomponentes según sea necesario
 - Si MEMORY_TARGET tiene un valor mayor que MEMORY_MAX_TARGET en el startup, entonces MEMORY_MAX_TARGET se aumenta.
 - Después del startup, MEMORY_TARGET puede ser modificado dinámicamente, pero no puede exceder del valor de MEMORY_MAX_TARGET calculado en el startup

► Gestión manual o compartida. Parámetro de Inicialización SGA_MAX_SIZE:

- El tamaño de la instancia puede crecer hasta el valor marcado por SGA_MAX_SIZE. Pero si los valores iniciales de sus componentes superan SGA_MAX_SIZE, Oracle ignora este último.
 - DB_CACHE_SIZE Tamaño de la caché de bloques estándar.
 - LOG_BUFFER Número de bytes asignados para el "redo log buffer".
 - SHARED_POOL_SIZE Tamaño en bytes del área dedicada a sentencias compartidas de SQL y PL/SQL
 - LARGE_POOL_SIZE Tamaño del large pool; por defecto, 0.
 - JAVA_POOL_SIZE Tamaño del Java pool.

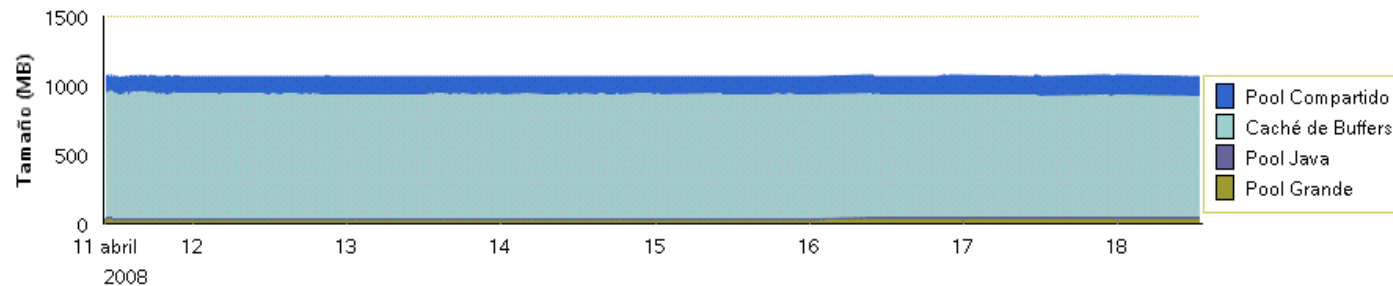
Estructura de la Memoria: SGA

SGA | PGA

El Área Global del Sistema (SGA) es un grupo de estructuras de memoria compartida que contiene datos e información de control para una base de datos Oracle. SGA está en la memoria cuando se inicia la instancia de base de datos Oracle.

Historial de Asignaciones

En este gráfico se muestra el historial de los componentes de SGA.



Asignación Actual

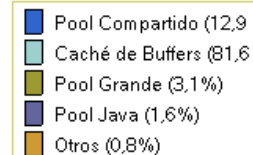
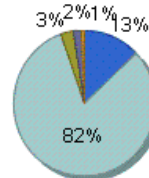
Gestión Automática de Memoria Compartida **Activado**

Desactivar

Tamaño de SGA Total (MB) 1024

Consejo

Componente de SGA	Asignación Actual (MB)
Pool Compartido	132
Caché de Buffers	836
Pool Grande	32
Pool Java	16
Otros	8



Tamaño Máximo de SGA

El tamaño máximo de SGA especifica la memoria máxima que puede asignar la base de datos. Si especifica el tamaño máximo de SGA, más tarde puede cambiar de forma dinámica el tamaño de SGA anterior (siempre que el tamaño de SGA total no exceda el tamaño máximo de SGA).

Tamaño Máximo de SGA* (MB) 1024

Área de Código de *Software* (SCA):

- ▶ Son zonas de memoria destinadas a **almacenar el código de Oracle** en ejecución o que puede ejecutarse. Este código de Oracle se almacena en una zona distinta, y más protegida, que las zonas dedicadas a almacenar los códigos de programas de usuarios.
- ▶ Son **áreas de sólo lectura** de tamaño estático que solamente cambian cuando el *software* se instala y su tamaño depende del S.O.
- ▶ **Pueden ser compartidas o no compartidas:** Las primeras son más eficientes porque el mismo código puede ser usado por distintos usuarios.