

Componentes Físicos y Lógicos de Oracle

Para obtener más información, visite [www.slinfo.una.ac.cr](http://www.slinfo.una.ac.cr)

## **INTRODUCCIÓN**

Los Sistemas Gestores de Bases de Datos (SGBD), son sistemas de software muy complejo, altamente eficiente y diseñado para ser flexibles (a nivel de hardware) y minimizar los errores causados por viejas tecnologías (archivos indexados, ISAMs, entre demás tecnologías).

La administración de estos SGBD es fundamental para permitir la disponibilidad de la información, asegurar la accesibilidad a los datos a aquellas personas que realmente tienen derecho de verla y mantener intactos los datos y por lo tanto íntegros. La administración es un proceso continuo de verificaciones, controles y monitoreos para garantizar los elementos antes citados.

## **CONCEPTO DE UNA BASE DE DATOS**

Una Base de Datos Oracle es un conjunto de datos almacenado y accesible según el formato de tablas relacionales. Una tabla relacional tiene un nombre y unas columnas. Los datos están almacenados en forma de matrices.

Una Base de Datos Oracle está almacenada físicamente en archivos, y la correspondencia entre los archivos y las tablas es posible gracias a las estructuras internas de la BD, que permiten que diferentes tipos de datos estén almacenados físicamente separados. Esta división lógica se hace gracias a los espacios de tablas: tablespaces.

## **CONCEPTO DE TABLESPACE**

Cuando se crean tablas en un base de datos Oracle, estas tablas se van alojar en el tablespace por defecto, normalmente SYSTEM a menos que se le indique explícitamente la ubicación del tablespace. Sin embargo SYSTEM también contiene tablas del diccionario de datos, por lo que es **altamente recomendado** no almacenar datos en el mismo tablespace de SYSTEM.

Lo aconsejable y razonable es que cada aplicación tenga su propio tablespace, en el caso de sistemas muy grandes se pueden realizar una segmentación de tablespace por áreas de la organización.

A continuación se detallan las razones por las que se aconseja tener tablespace separados:

1. Un espacio de tablas puede quedarse offline por mantenimiento o por fallas en el disco duro, permitiendo que las demás tablespaces sigan funcionando correctamente.
2. Para reducir costos de infraestructura, se pueden montar tablespaces en unidades de solo lectura si afectan la estructura de las operaciones de la base de datos.
3. Permite distribuir de forma lógica y física los distintos objetos de la base de datos.
4. Al ser unidades independientes, se logra un aislamiento de la información sensible.
5. Se pueden realizar respaldos en caliente a nivel de tablas o tablespaces mientras la base de datos sigue funcionando.

Algunas consultas importantes cuando se quiere conocer más sobre los tablespaces de un SGBD:

```
SQL> select * from user_tablespaces;  
SQL> select * from v$tablespace;
```

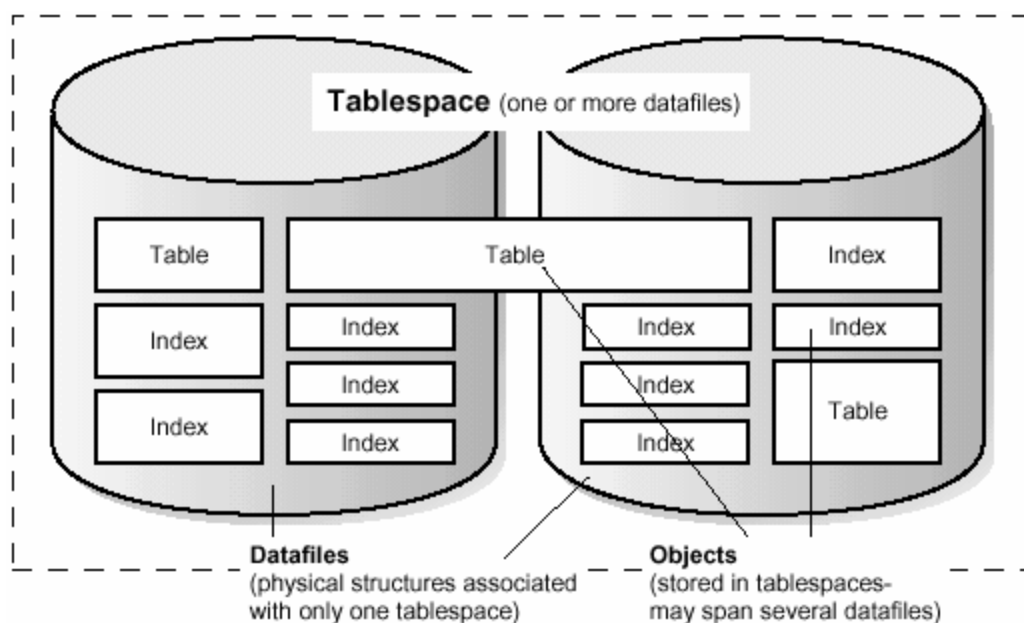
## SOBRE LA ESTRUCTURA DE UNA BASE DE DATOS ORACLE

En el presente curso, vamos a utilizar una base de datos Oracle como herramienta principal para llevar a cabo todas las labores de administración de bases de datos, criterios de seguridad, respaldos y recuperación de la información. En términos generales las bases de datos cuentan con dos tipos de estructuras: las **estructuras lógicas** y las **estructuras físicas**, estas varían según el SGBD.

## Estructura lógica

Es el nivel de abstracción más alto en las bases de datos, en este caso cuando nos referimos a estructura lógica, lo que realmente estamos hablando es sobre el "producto terminado" de una base de datos, es decir **tablespaces** los cuales contienen a la vez objetos tales como: tablas, vistas, procedimientos almacenados, triggers u otro objetos.

Nótese que en la estructura lógica, las tablas podrían estar repartidas en muchos discos duros dentro y fuera del servidor. Y sin embargo, siempre estamos hablando de la misma tabla sin importar como este físicamente distribuida en archivos del Sistema Operativo (OS), es por ello que hablamos de estructura lógica y no de la estructura física.

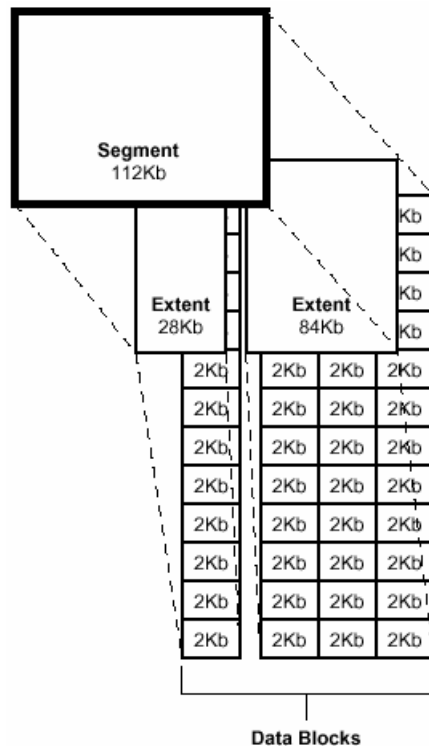


Véase en la imagen anterior, como una misma tabla puede estar ubicada en muchos Datafiles, pero sin embargo la tabla se asume como una entidad lógica única e indivisible.

Cada Base de datos está formada por uno o más tablespaces, como mínimo una base de datos debe contener el **tablespace SYSTEM** llamado normalmente como el diccionario de datos (catálogo del sistema). Cada tablespace se corresponde con uno o más ficheros de datos (**Datafiles**).

Oracle define un **esquema** como una colección de objetos o estructura lógicas relacionadas entre ellas. Cada usuario tiene su propio esquema (relación 1 a 1). Es así como se incrementa la seguridad en las bases de datos, puesto que ciertos usuarios no pueden ver algunas tablas a las que no deberían tener acceso, por ejemplo usuarios del departamento de Legal de una empresa, no necesitan ver los datos del departamento de cómputo.

Elementos del mismo esquema, pueden estar habitando en diferentes tablespaces. De la misma forma, diferentes objetos del tablespaces podrían pertenecer a diferentes esquemas. Por lo tanto se concluye que **no existe relación directa con el concepto de esquema y tablespaces**.



El control de uso de espacio del disco se obtiene mediante las estructuras lógicas de almacenamiento, de la siguiente forma: Tablespace > Segmento > Extensión > Bloques

El nivel más pequeño de almacenamiento es el del **Bloque** el cual corresponde a un número específico de bytes contiguos de espacio físico en el disco.

El siguiente nivel es el de **extensión**, el cual es un número específico de bloques de datos **contiguos** en el disco.

También tenemos el **segmento** el cual es un conjunto de extensiones utilizadas para almacenar alguna estructura lógica.

Existe a la vez, una serie de segmentos listados a continuación:

1. Segmentos de datos: almacenan datos de las tablas
2. Segmentos de índices: cada índice se encuentra en segmentos separados, estos permiten tener un acceso más rápido a las tablas de los datos, se aconseja tener los segmentos de índices separados de los segmentos de datos.
3. Segmentos de rollback: Los segmentos de rollback permiten regresar una transacción a un estado consistente en el pasado, para ello los segmentos de rollback cuentan con los siguientes datos:
  - a. Identificador de transacción
  - b. Dirección del bloque donde se encuentra la tabla
  - c. Numero de fila
  - d. Numero de columna
  - e. Valor del dato antiguo (antes de ser modificado)
4. Segmentos temporales: Son creados por Oracle cuando se requieren realizar operaciones de ordenamiento, estos son destruidos una vez se termina las sentencias. Las sentencias que hacen uso de estos segmentos son:
  - a. INDEX

- b. ORDER BY
- c. GROUP BY
- d. DISTINCT
- e. UNION
- f. INSERT
- g. MINUS

Finalmente la base de datos cuenta con los **tablespaces** los cuales contiene todos los objetos de la base de datos.

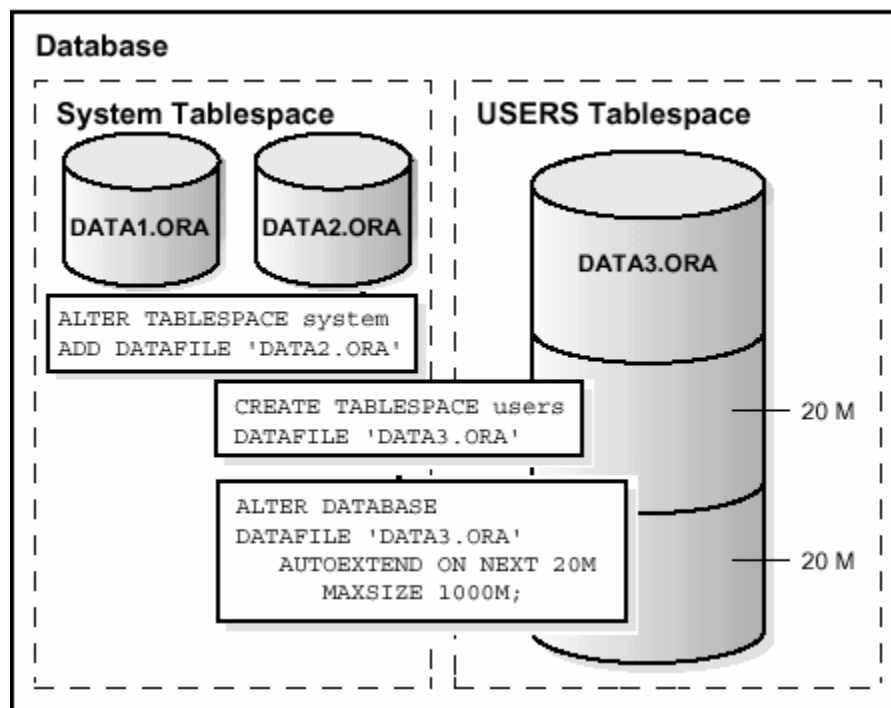
## Estructura física

Una base de datos tiene uno o más archivos de datos, estos archivos son de tamaño fijo y se establecen en el momento en que se crea una base de datos o cuando se crean los tablespaces.

Existen tres elementos físicos importantes:

1. **Datafiles:** los cuales son archivos binarios donde se almacenan propiamente las tablas y demás estructuras
2. **Redo log:** permiten deshacer los cambios de un procedimiento de modificación de la base de datos (Insert, update, delete, o transacción).
3. **Control Files:** Son archivos de control determinan el comportamiento de la base de datos, son por así decirlo archivos de configuración que se utilizan para iniciar el SGA.

Los ficheros de control (**Datafiles**) almacenan información de la estructura física de la base de datos.



# System Global Area (SGA)

Se dice que una base de datos está instanciada, cuando sus procesos se encuentran corriendo. Es decir cuando hay en el sistema operativo una serie de procesos ejecutándose. A este bloque de memoria se le denomina SGA, el cual a su vez está formado por una serie de bloques de memoria cada uno de ellos con una tarea específica.

## Shared Pool

Una forma de optimizar las consultas contra la base de datos, es la denominada búsqueda por caminos estadísticos. Este proceso es básicamente buscar previamente en el SGBD si una consulta fue realizada en el pasado, si fue previamente utilizada Oracle no debe evaluar nuevamente la integridad de la consulta y de esta forma evita aplicar un análisis lexicográfico, semántico y sintáctico de la consulta. Específicamente el Shared Pool contiene:

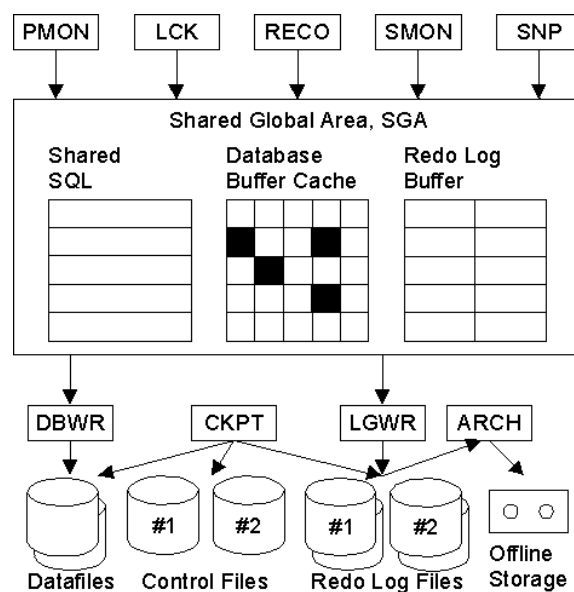
1. Plan de ejecución de la sentencia SQL
2. Texto de la sentencia
3. Lista de Objetos referenciados
4. Cache del diccionario

## Database Buffer Cache

En esta zona de la memoria, se almacenan los datos de las tablas leídos previamente, estos datos son almacenados en memoria para acceder de forma rápida si se vuelve alguna sentencia que utilice las tablas, índices o clúster leídos previamente. Cuando los objetos son modificados, a estos datos se les denomina bloques sucios.

## Buffer Redo Log

Cuando se necesita realizar rollback en las bases de datos, estos datos son almacenados en el cache del buffer Redo Log. Cuando el tamaño en memoria se agota, entonces los datos que están en el cache pasan directamente a los archivos de redo log.



# Procesos del SGA

---

El servidor se vale de una serie de procesos para hacer de puente entre los ficheros y los caches del SGA. Cada proceso es de vital importancia durante el uso de la base de datos.

## SMON

El System MONitor es el encargado de recuperar la base de datos ante alguna caída del sistema, ya sea causada por errores físicos, lógicos u otras caídas no detectadas. El proceso realiza una recuperación de los datos utilizando los archivos de Redo Log, limpia los segmentos temporales no utilizados y compacta los huecos libres contiguos de los ficheros de datos. El SMON se activa periódicamente para analizar el estado del SGBD.

## PMON

El Process MONitor está monitoreando las transacciones no terminada de forma exitosa, de esta forma si una transacción no activa el **End Transaccion** por parte del usuario, el PMON se activa para liberar los bloqueos y recursos del SGA.

## DBWR

El DataBase WRiter es el proceso encargado de hacer posible las modificaciones en la base de datos, siempre esta gestionando el contenido de los diferentes buffers y la cache del diccionario. Este proceso tiene la particularidad, de ser el único encargado de escribir en la base de datos, para garantizar la integridad de la información. Cada modificación no se lleva a la base de datos de forma inmediata sino que los bloques se vuelcan a los ficheros de datos periódicamente o cuando se realiza un *checkpoint*.

Para aumentar la velocidad de escritura, el DBWR sólo escribe a disco duro cuando tiene una cantidad considerable de datos en cache, esto significa que la escritura no se da en orden de transacción sino que podría incluso escribirse en paralelo para aumentar la velocidad. Para mantener el orden de las transacciones se utiliza el *redo log* para llevar un orden de transacción.

## LGWR

El LoG WRiter es el proceso encargado de escribir los *redo log buffer*, a los archivos de redo log. Los ficheros de redo log, siempre contiene los estados más recientes de la base de datos. De forma análoga con el DBWR el LGWR tiene la responsabilidad única de acceder a los redo logs, tanto para escribir en ellos como para leer los datos del redo log.

## CKPT

El Checkpoint es el proceso encargado de escribir en los ficheros de control los checkpoints. Estos puntos son referencias al estado coherente de todos los archivos de la BD en un instante determinado. Este proceso asegura que todos los datos que están en los *redo log* se apliquen de forma permanente a los archivos de datos. El proceso se activa en las siguientes circunstancias:

1. Un tablespace pasa a un estado Offline
2. Se llena un *redo log*
3. Se detiene la base de datos
4. Después de un número determinado de segundos

## **ARCH**

El proceso archivador tiene que ver con los ficheros redo log. Por defecto, estos ficheros se reutilizan de manera cíclica de modo que se van perdiendo los registros redo log que tienen una cierta antigüedad. Cuando la BD se ejecuta en modo **ARCHIVELOG**, antes de reutilizar un fichero redo log realiza una copia del mismo. De esta manera se mantiene una copia de todos los registros redo log por si fueran necesarios para una recuperación.