

Introducción a la Arquitectura del RDBMS ORACLE

Una primera y sencilla aproximación a la arquitectura general utilizada por el RDBMS ORACLE para el manejo de base de datos (independientemente de la configuración -single, multi-thread, parallel- utilizada) es la mostrada en la Figura 1.1.

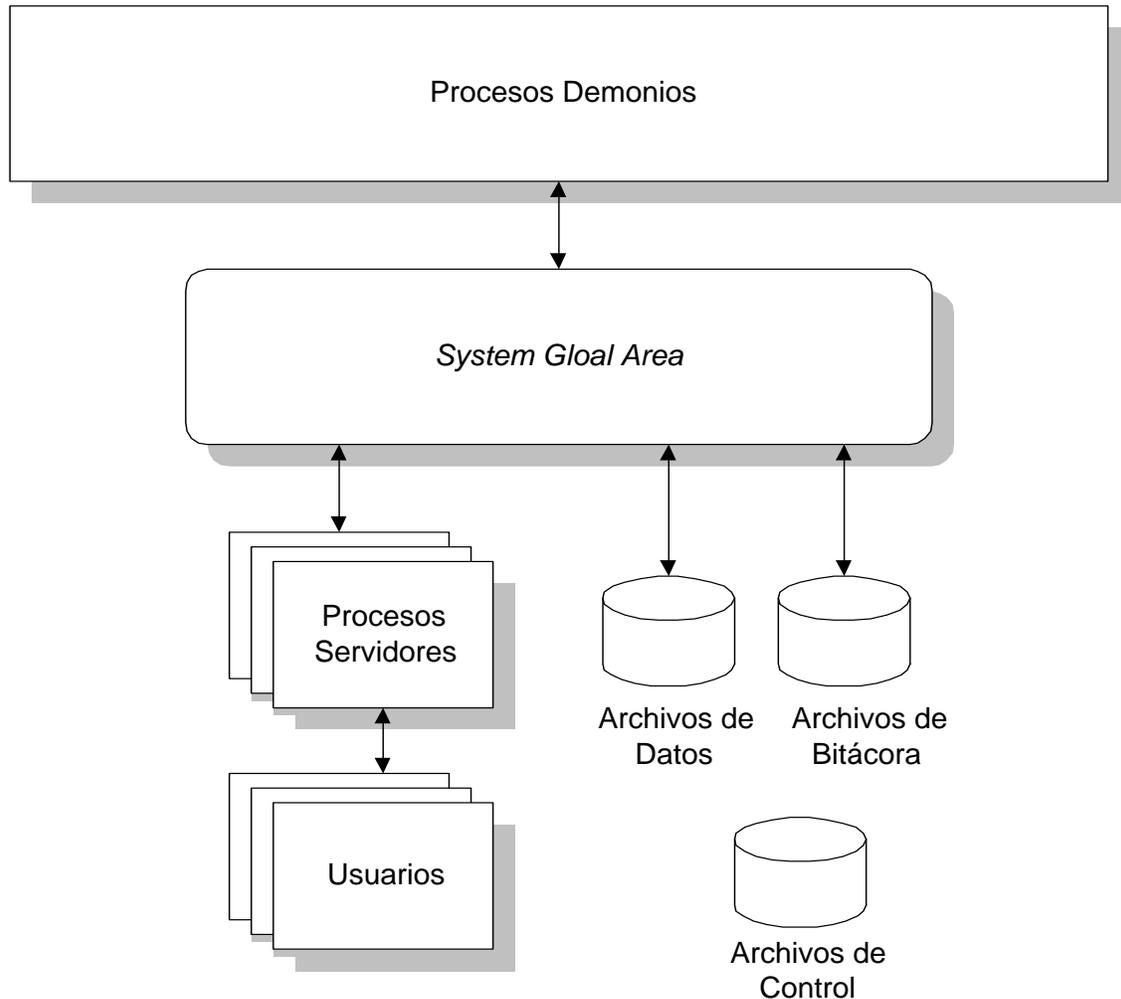


Figura 1.1: Arquitectura del RDBMS ORACLE

Esta arquitectura puede ser dividida en dos porciones lógicas:

- Estructura de procesos y memoria.
- Estructura para el manejo de los datos.

Estructura de Procesos y Memoria

Independientemente de la arquitectura computacional, o de su configuración, cada base de datos dentro del RDBMS ORACLE es asociada a una determinada instancia, y de igual forma una instancia puede abrir y utilizar sólo una base datos ORACLE en cualquier momento de su

ejecución. Es posible poseer múltiples instancias ejecutándose concurrentemente dentro de una misma máquina, cada una accediendo su propio espacio físico de datos (su base de datos ORACLE). En el sistema de operación, la variable de ambiente ORACLE_SID permite identificar el nombre de la instancia ORACLE a la cual se conectarán, por defecto, las aplicaciones de usuario.

Cada vez que el RDBMS ORACLE es inicializado, tanto el *System Global Area* (SGA) como los procesos demonios son levantados. El SGA junto con los procesos demonios es lo que se demonina como una instancia ORACLE.

El *System Global Area* es un conjunto de estructuras de memoria compartida que contienen datos e información de control para una determinada instancia ORACLE. El SGA se mantiene en la memoria virtual del computador en el que reside la instancia ORACLE. Si dentro de la instancia existe la posibilidad de que mas de un usuario se encuentren conectados simultáneamente, los datos dentro del SGA de la instancia son compartidos entre todos los usuarios. Es por esto que algunas veces al SGA también se le suele denominar *Shared Global Area*. La estructura interna de la SGA puede observarse en la Figura 1.2

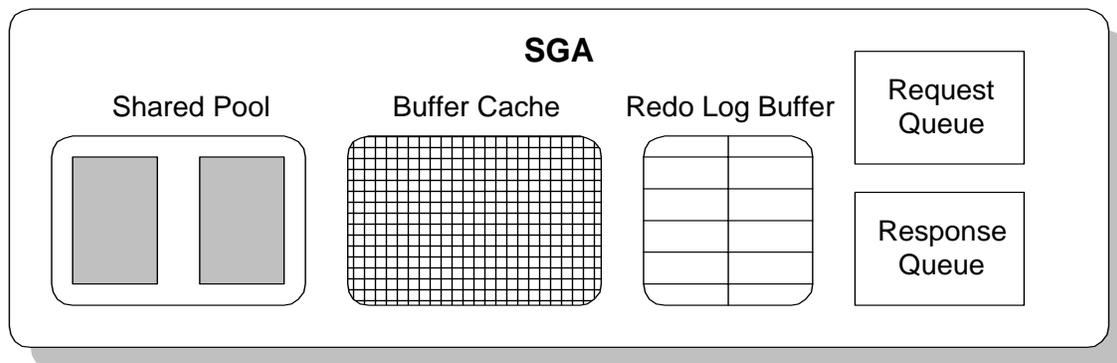


Figura 1.2: Estructura básica del *System Global Area*

- *Database Buffer Cache*: Su función es mantener bloques de datos leídos directamente de los archivos de datos. Cuando se procesa una consulta, el servidor busca los bloques de datos requeridos en esta estructura. Si el bloque no se encuentra en esta estructura, el proceso servidor lee el bloque de la memoria secundaria y coloca una copia en esta estructura. De esta forma, otras peticiones que requieran de este bloque de datos no requerirán de acceso a memoria secundaria (lecturas físicas).

El *Database Buffer Cache* está organizado en dos listas: la lista de sucios y la lista de menos recientemente usados. La lista de sucios mantiene aquellos bloques de datos que han sido modificados y que aún no han sido escritos a disco. La lista de menos recientemente usados mantiene los bloques libres, los bloques a los que se está accediendo actualmente y los bloques sucios que aún no han sido remitidos a la lista de sucios. El número de bloques manejados por el *Buffer Cache* puede ser configurado para mejorar el rendimiento, así como el tamaño del bloque de datos. En cualquier caso el tamaño de bloque de datos utilizado debe ser el mismo que el que se

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

ha configurado para la instancia como tamaño de bloque de datos utilizado por el DBMS.

- *Redo Log Buffer*: Es un buffer circular que mantiene todos los cambios que han sido realizados sobre la base de datos por operaciones de *insert*, *update*, *delete*, *create*, *alter* y *drop*. Las entradas de este buffer contienen toda la información necesaria para reconstruir los cambios realizados a la base de datos por medio de cualquier sentencia del DDL o del DML (el bloque que ha sido cambiado, la posición de cambio y el nuevo valor). El uso del Redo Buffer es estrictamente secuencial, en tal sentido pueden entrelazarse cambios en los bloques de datos producidos por transacciones diferentes. El tamaño de este Buffer también puede ser configurado para mejorar el rendimiento de la instancia y de las aplicaciones que sobre ellas se ejecutan.
- *Shared Pool*: Esta estructura se utiliza durante el procesamiento de comandos. Su tamaño también es configurable. Contiene dos zonas específicas:
 - *Library Cache*, almacena la siguiente información relacionada con una instrucción de SQL:
 - Texto de la instrucción.
 - Arbol de *parsing*, es decir la versión compilada de la instrucción.
 - Plan de Ejecución, es decir la secuencia de pasos a ser realizados para ejecutar la instrucción a bajo nivel de acuerdo con los resultados producidos por el optimizador de consultas.

Basándose en esta información, si una consulta es ejecutada nuevamente, y su información permanece todavía en el *Library Cache*, no será necesario compilar de nuevo la instrucción. En tal sentido este componente de la arquitectura permite mejorar el rendimiento de las aplicaciones que se ejecutan periódicamente.
 - *Data Dictionary Cache*, también conocido como *Dictionary Cache* o *Row Cache*, almacena la información de uso más reciente sobre el diccionario de datos. Esta información incluye definición de columnas, usuarios, *passwords* y privilegios. Durante la fase de compilación, esta información es necesaria para resolver los nombres de los objetos utilizados en un comando SQL y para validar los privilegios de acceso.
- *Request Queue y Response Queues*: Estas estructuras constituyen los elementos básicos para el manejo de conexiones de usuarios al manejador. Por cada instancia de ORACLE existirá una *Request Queue* y varias *Response Queues*. Cuando un proceso usuario desea establecer conexión con el manejador, la solicitud de conexión será encolada en la *Request Queue*. Los procesos servidores tomarán las solicitudes de conexión, efectuarán las acciones necesarias para que la base de datos complete la solicitud y colocarán la petición en la *Response Queue* asociada al proceso que atendió la solicitud.

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

Los Procesos en ORACLE

Los procesos de ORACLE realizan funciones para los usuarios del manejador. En general puede establecer una partición en los procesos de ORACLE:

- o Procesos Servidores o *Shadow Proceses*, los cuales ejecutan funciones de interacción con el servidor de datos basándose en las peticiones de los usuarios.
- o Procesos Demonio o *Background Proceses*, quienes realizan funciones para el DBMS como un todo.
- o Procesos de Red, encargados de proveer la interconexión entre procesos de usuario y el DBMS o entre DBMS que establecen un sistema de bases de datos distribuidas.

Procesos Servidores y el Program Global Area (PGA)

Los procesos servidores (*Snnn*) se comunican con los diferentes procesos de usuario e interactúan con ORACLE para satisfacer las peticiones. Por ejemplo, cuando un proceso de usuario solicita datos que no se encuentran en el SGA (en alguno de los buffer caches, dependiendo del tipo de operación y de la fase de su procesamiento), el proceso servidor que atiende la petición será el encargado de leer los bloques de datos y almacenarlos en el SGA (recuerde que el SGA es un área de memoria compartida). Puede existir una correspondencia de uno a uno entre procesos de usuario y procesos servidor (por ejemplo en la configuración de DBMS dedicado); aunque un proceso servidor puede conectarse a múltiples procesos de usuario (como en la configuración de DBMS multi-threaded). Las conexiones admitidas dependerán en todo momento de la configuración de opciones del DBMS realizadas durante la instalación del mismo.

El PGA constituye una región de memoria asociada a cada proceso servidor, la cual contiene datos e información de control para cada una de las sesiones que los usuarios mantienen con el RDBMS ORACLE a través de éste proceso servidor. Por lo tanto el PGA no es un área de memoria compartida. Una región de memoria para almacenar un PGA es solicitada cuando un proceso usuario establece una sesión de trabajo con el manejador de datos. El tipo de información que se almacena en el PGA depende de las opciones instaladas para el servidor ORACLE. Por ejemplo, cuando se utiliza una configuración de servidor dedicado, el PGA contiene los siguientes componentes:

- o *Sort Area*, que es utilizada para llevar a cabo los posibles ordenamientos de filas requeridos antes de que las filas sean procesadas o devueltas al usuario como resultado de una consulta.
- o *Stack Space*, el cual contiene las variables de sesión de usuario y sus valores.
- o *Cursor State*, el cual almacena el estado de los diferentes cursores que están siendo utilizados en la sesión del usuario.
- o *Session Information*, la cual mantiene información sobre los privilegios que el usuario que ejecuta la sesión.

Ejemplo: Cada vez que se invoca SQL*Plus, se crea un proceso usuario. Este proceso usuario se comunicará (bien sea por los mecanismos de IPC –en caso de que el servidor ORACLE y el proceso usuario estén en la misma máquina- o por mecanismos de software de comunicación

en Red como SQL*Net –en caso de que el servidor ORACLE y el proceso usuario estén en máquinas diferentes) con el proceso servidor que le proveerá del acceso necesario al servidor ORACLE.

Procesos Demonios

Los procesos demonio, también conocidos como *Background Proceses*, constituyen programas que llevan a cabo funciones específicas de soporte y mantenimiento a la ejecución del servidor de bases de datos. Esto no quiere decir que sean opcionales, por el contrario sin ellos no se podría operar correctamente en un entorno basado en ORACLE. Estos procesos pueden ser observados en la Figura 1.3



Figura 1.3: Procesos Demonios Manejados por el RDBMS ORACLE

- *Database Writer (DBWR)*: Encargado de copiar los bloques de datos desde el buffer cache a la memoria secundaria. Cuando una transacción cambia los datos de un bloque de datos, no es necesario que este bloque se escriba inmediatamente en el disco. Debido a esto, el DBWR puede escribir estos datos de forma que se logre mayor eficiencia de la obtenida si se escribiera siempre que una transacción terminara. Usualmente el DBWR escribe sólo cuando se necesitan nuevos bloques datos en el *Database Buffer Cache*. Los datos se escriben utilizando el algoritmo LRU (*Least Recently Used*). Si el entorno de memoria secundaria está basado en entrada salida asíncrona (AIO) debe existir un único proceso DBWR. Si el entorno no posee AIO, el rendimiento del sistema puede ser mejorado agregando más procesos DBWR.
- *Log Writer (LGWR)*: Encargado de escribir las entradas desde el Log Buffer a disco. La escritura de bloques del Redo Log Buffer a disco ocurre secuencialmente y bajo las siguientes reglas:
 - Cuando el Redo Log Buffer está lleno en un 33% o más.
 - Cuando ocurre un *timeout* (cada 3 segundos).
 - Antes de que el DBWR escriba algún bloque modificado a disco.
 - Cuando una transacción se comprometa.
- *Checkpoint (CKPT)*: Encargado de notificar al DBWR, para que se escriban en los archivos de datos todos los bloques contenidos en la lista de sucios. Este proceso es invocado en intervalos de tiempo determinados. El CKPT es opcional. Si este proceso no está presente las funciones son realizadas por el LGWR.
- *System Monitor (SMON)*: Este proceso es el encargado de hacer un porceso de recuperación rápida cada vez que una instancia es inicializada. Esta labor incluye limpieza de las estructuras de datos de soporte a la ejecución de consultas y llevar la base de

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

datos a un estado estable previo a la ejecución de aquellas transacciones que no hayan culminado exitosamente. De igual forma, el proceso SMON se encarga de desfragmentar el espacio físico de almacenamiento uniendo bloques de datos libres en la memoria secundaria.

- *Process Monitor (PMON)*: Es el proceso encargado de llevar la pista de los procesos de la base de datos y efectuar labores de limpieza (liberar los recursos y liberar bloques ocupados en los caches) si alguno de ellos termina prematuramente. En la opción de *multi-threading* también es el encargado de reiniciar cualquier proceso *dispatcher* en caso de fallas.
- *Archiver (ARCH)*: Proceso encargado de copiar las bitácoras activas en el sistema cuando estas se encuentran llenas. Este proceso se encuentra activo sólo cuando el DBMS se encuentra operando en modo ARCHIVELOG, el único modo que admite recuperación de los datos frente a fallas del sistema.
- *Recoverer (RECO)*: Encargado de resolver transacciones distribuídas que se encuentran pendientes debido a la red o a fallas ocurridas en la base de datos distribuída.
- *Dispatcher (Dnnn)*: Un proceso dispatcher es creado por cada sesión de trabajo activa. Cada dispatcher es responsable de enrutar los requerimientos desde el proceso usuario, al cual se encuentra asociado, hacia los procesos servidores y retornar la respuesta al proceso de usuario adecuado. Los procesos *Dnnn* se crearán sólo en entornos donde el DBMS se ejecute con la opción de multi-threading.

Al analizar en detalle toda la arquitectura de procesos de ORACLE, la Figura 1.4 puede servir de marco de referencia para visualizar las diferentes relaciones de uso existentes entre los componentes que componen la arquitectura de procesos del manejador.

Control de Recursos

ORACLE administra el uso de los diferentes recursos que gestiona a través de "*latches*". Un "*latch*" es un semáforo mantenido por la instancia de ORACLE que permite administrar una cierta actividad. Los "*latches*" limitan la cantidad de tiempo y espacio en los que un proceso puede mantener un recurso en un instante dado. De igual forma, los "*latches*" son utilizados para garantizar el acceso exclusivo a un recurso. El monitoreo de "*latches*" permite determinar situaciones en la que se crea contención por el acceso a un recurso. El número de "*latches*" existentes depende de la configuración y la información de los "*latches*" definidos y manejados por la instancia puede ser extraída del diccionario de datos.

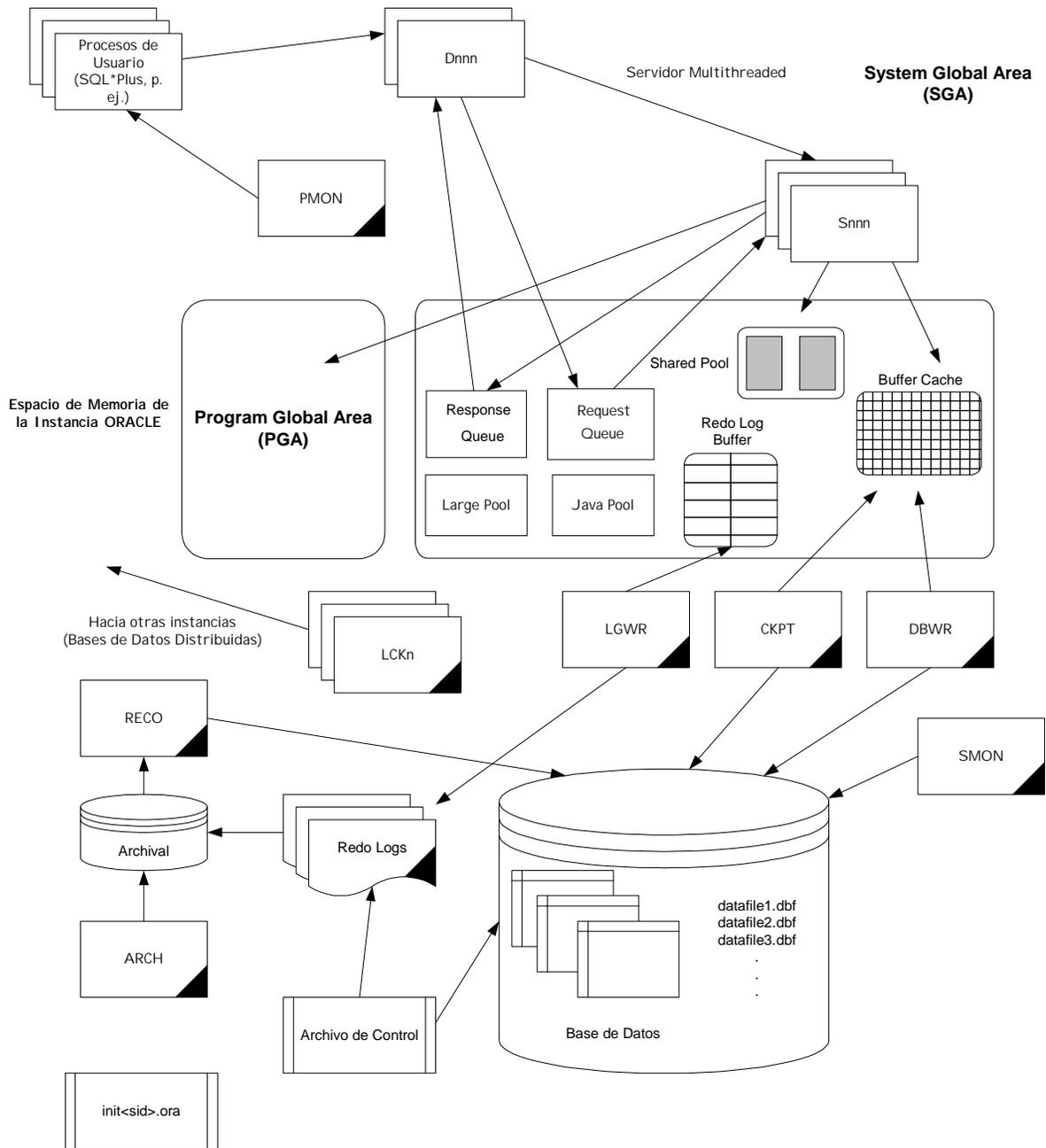


Figura 1.4: Arquitectura detallada del RDBMS ORACLE

Estructura Física para el Manejo de Datos

Una base de datos ORACLE, identificada por un nombre lógico mantenido en la variable de entorno DB_NAME, representa las estructuras físicas y está compuesta de archivos del sistema de operación. Aunque es posible utilizar un nombre de base de datos diferente al del nombre de la instancia, se recomienda utilizar el mismo nombre para hacer más sencilla la administración.

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

Los archivos que conforman la base de datos contienen los datos del usuario e información adicional que es necesaria para garantizar el funcionamiento de la base de datos. Una base de datos ORACLE consiste de los siguientes tipos de archivos:

- o Archivos de datos (*Data Files*): son aquellos que almacenan el diccionario de datos, los objetos del usuario, y los valores de los datos previos a su modificación. Una base de datos ORACLE debe poseer al menos un archivo de datos. La información de una tabla puede expandirse a muchos *Data Files* y un *Data File* puede almacenar múltiples tablas. El número máximo de *Data Files* que soporta una instancia de ORACLE puede ser configurado.
- o Archivos de bitácora (*Redo Log Files* o *Redo Log*): Mantienen información sobre todos los cambios efectuados sobre la base de datos para asegurar su reconstrucción en caso de fallas. Toda base de datos ORACLE requiere de por lo menos dos *Redo Log Files*. Es esencial que los *Redo Log* posea un buen rendimiento y estén protegidos contra fallas de hardware. Si se pierde la información contenida en estos archivos será imposible la recuperación de la base de datos en caso de fallas del sistema. ORACLE provee mecanismos para almacenar múltiples copias de los *Redo Logs*.
- o Archivos de Control (*Control Files*): Contienen la información necesaria para mantener y verificar la integridad de la base de datos, como la ubicación de los datos y los *Redo Log*. ORACLE requiere de esta información toda vez que se "arranque" la instancia del DBMS. Toda base de datos ORACLE requiere de por lo menos un *Control File*. Es esencial que los *Control Files* estén protegidos ya que en caso de pérdida no se podrá reiniciar la instancia de la base de datos con la consecuente pérdida de toda la información. ORACLE provee mecanismos para almacenar múltiples copias de los archivos de control.

Estructura Lógica de Almacenamiento

ORACLE es el encargado de manejar el espacio donde van a ser almacenados todos los objetos de una base de datos. Las unidades lógicas de almacenamiento son: bloques de datos, extents, segmentos y *tablespaces*. La Figura 1.5 muestra la relación existente entre estas estructuras de datos.

Tablespaces

El nivel más alto en la estructura de almacenamiento de datos en ORACLE es la base de datos. Una base de datos (DATABASE) agrupa las piezas lógicas de mayor nivel de almacenamiento en ORACLE: los *tablespaces*.

Un *tablespace* es utilizado para agrupar lógicamente los datos. Por ejemplo, puede tenerse un *tablespace* para almacenar los datos de contabilidad y otro para almacenar los datos de ventas en una empresa de servicios. Al segmentar grupos de datos relacionados en *tablespaces* diferentes se simplifican las tareas de administración de dichos grupos.

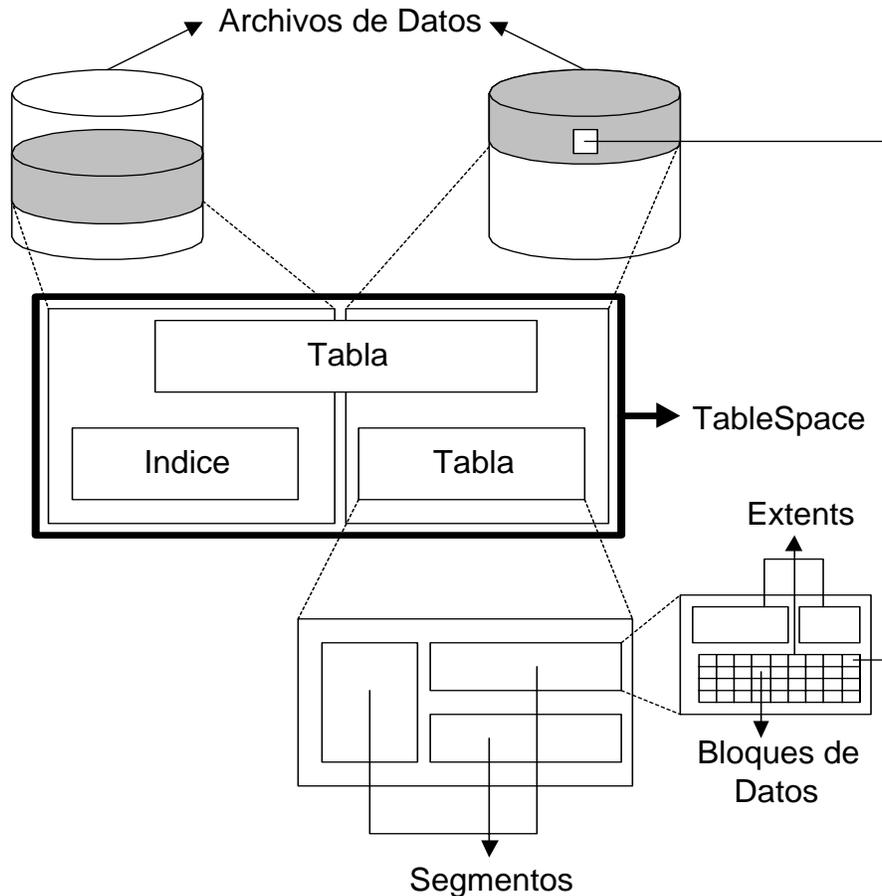


Figura 1.5: Estructura de datos manejadas por el RDBMS ORACLE

Los *tablespaces* pueden estar constituidos de varios *Data Files*. Al utilizar más de un *Data File* por *tablespace* puede distribuirse los datos sobre varios discos y balancear la carga de E/S, mejorando así el rendimiento del sistema.

Como parte del proceso de crear la base de datos, ORACLE automáticamente crear un *tablespace* llamado SYSTEM. Aunque sólo una base de datos pequeña puede ser almacenada en SYSTEM, es recomendable que se cree uno (o varios) *tablespace(s)* para los datos del usuario. El *tablespace* SYSTEM almacena los datos del diccionario.

Los *Data Files* pueden ser archivos del sistema de operación, y en algunos sistemas de operación se permite que un archivo de datos sea un dispositivo de almacenamiento secundario crudo (o parte de él).

Segmentos

Un segmento es un conjunto de *extents* que han sido asignados para el almacenamiento de un tipo de datos específico y todos ellos se encuentran ubicados dentro del mismo *tablespace*.

Existen diferentes tipos de segmentos como lo son:

- o Segmentos de datos: Cada segmento de datos almacena los datos correspondientes a una tabla.
- o Segmentos de índice: Cada segmento de índice mantiene los datos para un índice definido dentro de la base de datos.
- o Segmento de *Rollback*: un segmento de Rollback permite almacenar las acciones de una transacción que deben ser deshechas bajo ciertas circunstancias.
- o Segmentos Temporales: Los segmentos temporales se crean cuando se requiere de un espacio temporal para procesar una instrucción de SQL, y son destruidos una vez que haya culminado el procesamiento de la instrucción.

Extents

Los *extents* son las piezas utilizadas para constituir segmentos. Cada *extent* se compone de una serie de bloques de datos. La razón principal de esta estructura es la de minimizar el espacio desperdiciado (vacío) de un *tablespace*. A medida que se insertan o eliminan filas de una tabla, los *extents* del *tablespace* asociado a la tabla pueden aumentar o disminuir de tamaño. De esta forma el espacio para el almacenamiento de los datos puede ser administrado dependiendo de cómo la tabla va sufriendo modificaciones en el número de filas.

Cuando se crea un *tablespace* se puede especificar el número mínimo de *extents* a ser asignados, así como el número de *extents* a ser agregados cada vez que se agote el espacio disponible para almacenar datos.

Bloques de Datos

Una base de datos se encuentra almacenada en bloques de datos que es el nivel mas fino de unidades de almacenamiento. El tamaño de un bloque de datos siempre corresponde a un múltiplo del tamaño de bloque manejado por el sistema de operación. El tamaño del bloque de datos es un valor configurable en el DBMS.

La Figura 1.6 muestra conceptualmente las relaciones existentes entre las estructuras de almacenamiento lógico y físico de ORACLE.

El Diccionario de Datos de ORACLE

El diccionario de datos es una parte fundamental de toda base de datos ORACLE. El diccionario de datos de ORACLE está constituido por una serie de tablas, vistas y ORACLE *packages* a los que puede accederse para obtener información sobre la base de datos.

Las tablas del diccionario de datos son creadas automáticamente durante el proceso de instalación de ORACLE y permiten al administrador conocer, entre otros:

- o La estructura lógica y física de la base de datos
- o Los usuarios de la base de datos.

- o Las restricciones de integridad definidas sobre las tablas de la base de datos.
- o El espacio asociado a cada objeto en la base de datos y cuánto de este espacio está siendo realmente utilizado.

Los diferentes objetos creados por los usuarios de la base de datos.

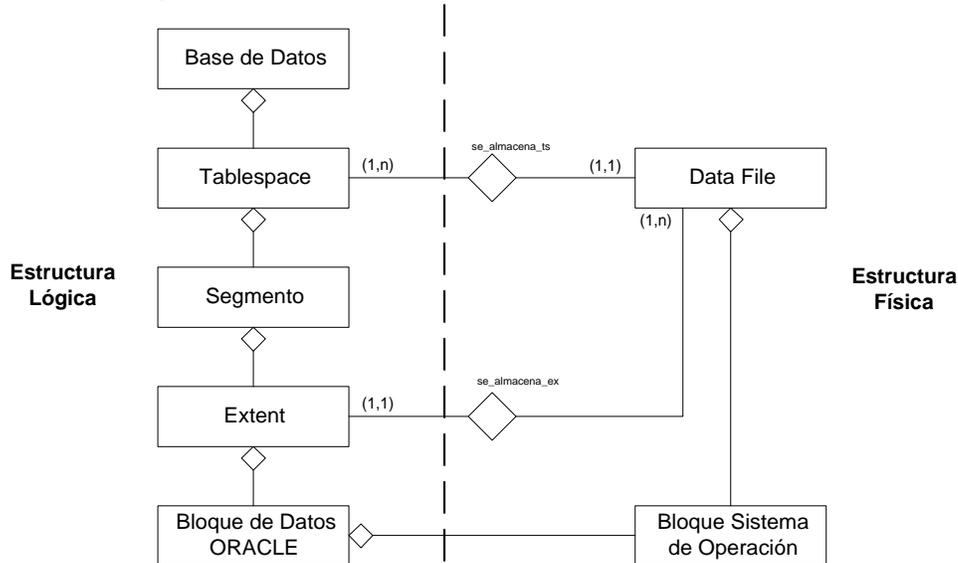


Figura 1.6 Relación entre estructuras de almacenamiento en ORACLE

ORACLE predefine un usuario dueño de todo el diccionario de datos denominado SYS. Este usuario tiene todos los permisos sobre cualquier objeto de la base de datos (incluidos los objetos de cualquier usuario). Debido a que este usuario puede modificar entradas del diccionario de datos es recomendable no utilizarlo ya que cualquier error generado sobre el diccionario de datos puede provocar errores irre recuperables en el DBMS. Sólo se deberá hacer uso cuando se requiera efectuar modificaciones CONTROLADAS sobre el diccionario para reparar errores en el mismo (cirugía de la base de datos).

Tablas Base

La base del diccionario de datos es una serie de tablas a las que el servidor de datos accede cada vez que se procesa una instrucción del DDL de SQL, o en determinados comandos del DML. Estas tablas raramente son accedidas por usuarios ya que la mayor parte de su información está codificada.

Por ejemplo, la tabla IND\$ provee información sobre los índices definidos en la base de datos, y la tabla OBJ\$ provee información sobre los diferentes objetos lógicos (tablas, índices, secuencias, etc.) creados por los usuarios de la base de datos.

Vistas Estáticas

Las vistas estáticas se crean durante la instalación del DBMS y decodifican y resumen la información contenida en las tablas base. Durante la ejecución del script de creación de estas vistas se generan sinónimos públicos para proveer el acceso a los usuarios del DBMS.

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

Estas vistas son las que deben ser utilizadas para efectuar labores de administración rutinarias que requieran de información específica sobre la configuración y estado de la base de datos. A estas vistas se les conoce como estáticas porque no mantienen información relacionada con las sesiones que actualmente existen en el sistema de base de datos.

Estas vistas se dividen en tres categorías de acuerdo con el prefijo de sus nombres:

- o Vistas con prefijo USER: Estas vistas pueden ser utilizadas por cualquier usuario de la base de datos y se refieren a los objetos poseídos por dicho usuario. Por ejemplo si el usuario ci531301 ejecuta el comando SQL

```
SELECT *
FROM
USER_TABLES
```

Se desplegará toda la información de aquellas tablas cuyo dueño sea el usuario ci531301.

- o Vistas con prefijo ALL: Estas vistas pueden ser utilizadas por cualquier usuario e incluyen la columna OWNER adicionalmente a la demás información. Con estas vistas se puede tener acceso a la información de aquellos objetos de los que el usuario es dueño, más todos aquellos objetos públicos y todos aquellos objetos a los que el usuario tiene acceso por pertenecer a cierto grupo de seguridad o por poseer ciertos privilegios sobre la base de datos. Por ejemplo, si el usuario CI 531301 posee el privilegio de ver la tabla EMPLEADO creada por el usuario CI 531302 (dicho privilegio le fue otorgado por un administrador o por el mismo usuario CI 531302), entonces la consulta anterior devolverá la misma información (incluyendo la columna OWNER) mas una fila específica con la información sobre la tabla EMPLEADO del usuario CI 531302.
- o Vistas con prefijo DBA: Estas vistas dan información sobre TODOS los objetos de la base de datos, y usualmente incluyen la columna OWNER. Estas vistas sólo pueden ser utilizadas por el administrador de la base de datos o por cualquier usuario que posea el privilegio SELECT ANY TABLE o pertenezca a un rol que incluya tal privilegio.

Vistas Dinámicas

A diferencia de las vistas estáticas, las vistas dinámicas (también conocidas como *performance views*) incluyen información sobre las condiciones actuales (en un instante dado) de operación del DBMS. La mayor parte de estas vistas son creadas durante la instalación y existen algunas que se crean explícitamente para monitorear cierta actividad (el DBMS provee los archivos necesarios para crear estas vistas).

Todas las vistas dinámicas se identifican por poseer el prefijo V\$. Por ejemplo la vista dinámica V\$_SESSION incluye información sobre las sesiones actuales y la vista V\$SYSSTAT provee información estadística sobre el DBMS.

CI-5313 Arquitectura y Administración de DBMSs I	Prof. Nunzio Savino Septiembre-Diciembre 2001	Tema Nro. 1 Introducción a la Arquitectura del RDBMS ORACLE
--	--	---

Para obtener información general sobre las vistas del diccionario de datos puede usarse la consulta

```
SELECT *
FROM DICTIONARY
WHERE table_name LIKE '%indicador%'
```

Por ejemplo, para ver todas las vistas relacionadas con tablas podríamos ejecutar la instrucción SQL

```
SELECT *
FROM DICTIONARY
WHERE table_name LIKE '%TABLE%'
```

En el manual ORACLE8 Reference se dispone de toda la información detallada de las vistas estáticas y dinámicas que provee el DBMS.

Packages de Administración

ORACLE provee una serie de *packages* que incluyen funciones específicas de ayuda a la administración del sistema de bases de datos. Entre estos *packages* se encuentra:

- o DBMS_UTILITY: Provee diferentes utilitarios
- o DBMS_SPACE: Provee información sobre la disponibilidad de segmentos.
- o DBMS_SHARED_POOL: Provee información sobre el *shared pool*.

Todos y cada uno de los packages tienen el prefijo DBMS (provisto por el DBMS). En caso de requerir alguno de los paquetes pueden seguirse los siguientes pasos:

1. Ejecutar el comando

```
SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
FROM SYS.DBA_OBJECTS
WHERE
OBJECT_NAME LIKE 'DBMS_%' AND
OBJECT_TYPE = 'PACKAGE'
```

Que desplegará los *packages* ofrecidos por el DBMS. Esto no quiere decir que sólo sean esos los *packages* si no que sólo estos son los *packages* instalados.

2. Una vez seleccionado el *package* el comando

```
DESCRIBE <package>
```

Ofrece la descripción general del *package*

Nota: Sólo se podrá utilizar aquellos *packages* con estado VALID (correctamente compilado).