

Arquitectura de ORACLE

La arquitectura de ORACLE tiene tres componentes básicos: las **estructuras de memoria** para almacenar los datos y el código ejecutable, **los procesos** que corren el sistema de bases de datos y las tareas de cada usuario conectado a la base de datos y los **archivos** que sirven para el almacenamiento físico, en disco, de la información de la base de datos.

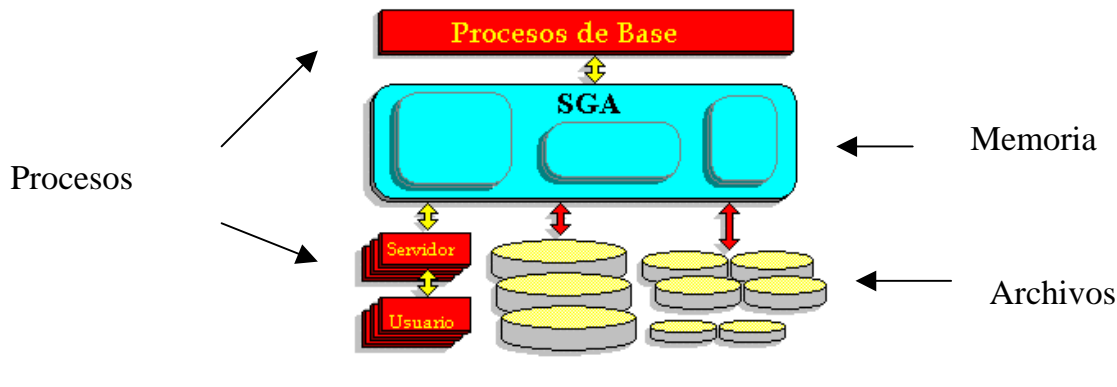


Ilustración 1. Arquitectura de ORACLE

Estructuras de memoria

Hay dos clases de memoria, una de ellas compartida por todos los usuarios conectados y otra dedicada al trabajo de cada uno de ellos.

El área global del sistema o SGA (system global area), es el área compartida por todos los usuarios y se divide en tres partes:

- **Fondo común compartido (Shared pool)**, en ella mantiene el diccionario de datos y las áreas compartidas de las órdenes SQL que se solicitan para su procesamiento.
- **Área de memoria rápida (Database buffer cache)**, donde mantiene los datos traídos por las órdenes SQL de los usuarios conectados a la base de datos.
- **Área de registros de rehacer (Redo log buffer)**, aquí se registran los cambios hechos a la base de datos.

Por cada sesión de usuario se crea también, en memoria, una área específica llamada área global de programa o PGA (program global area); esta área no se comparte con las otras sesiones de usuario.

Los Archivos

Los archivos que maneja ORACLE, se clasifican en cuatro grupos:

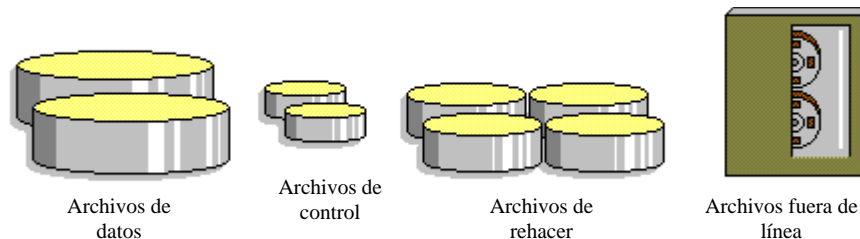
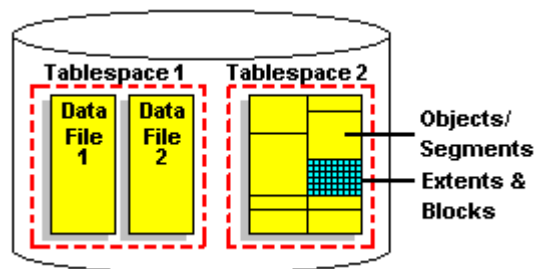


Ilustración 2. Tipos de archivos

- **Los Archivos de Datos (Datafiles):** estos archivos sirven para el almacenamiento físico de las tablas, índices o agrupamientos (clusters) y procedimientos. Estos archivos, son los únicos que contienen los datos de los usuarios de la base de datos.

Las unidades **lógicas** más grandes manejadas por ORACLE, para el almacenamiento de los datos, son llamadas espacios de tablas (tablespaces) que le permiten manejar o controlar el espacio en los discos.



No es necesario que todos los espacios de tablas estén en un mismo disco. Cuando se crean en distintos discos se busca un mejor desempeño y mejor manejo del espacio de almacenamiento.

Una base de datos puede tener un sólo espacio de tablas; pero, por las razones anteriores, se recomiendan varios espacios de tablas. Como mínimo, se debe

tener un espacio de tablas del sistema, un espacio de tablas por cada aplicación, un espacio de tablas para los usuarios y otro espacio de tablas para los índices.

El espacio de tablas SYSTEM se crea automáticamente cuando se crea una base de datos. Allí se guardan los archivos de control y el diccionario de datos y toda la información de los procedimientos almacenados.

El DBA puede crear un espacio de tablas con una orden, como la siguiente:

```
CREATE TABLESPACE indices  
datafile 'discod/bd/datos1.dbf' size 300m;
```

Los archivos de datos (datafiles) almacenan los datos del usuario. Sólo se requiere de uno para una base de datos. Sin embargo, los archivos de datos son fijos en tamaño e inalterables; cuando no haya espacio se deben adicionar más para incrementar el espacio de almacenamiento.

Cuando se agote el espacio, un DBA, tiene dos alternativas:

A) Adicionar un nuevo archivo de datos, con la orden ALTER TABLESPACE:

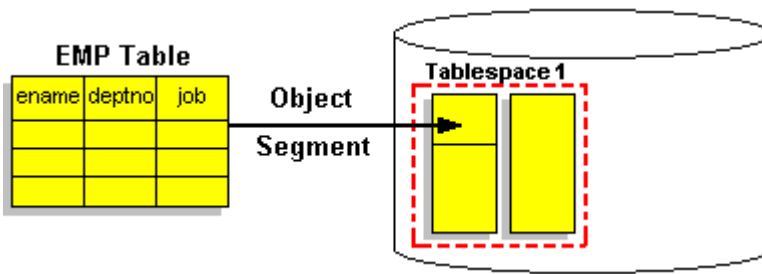
```
ALTER TABLESPACE indices  
add datafile 'discod/bd/datos3.dbf' size 150m;
```

B) Crear un nuevo espacio de tablas, como se mostró previamente.

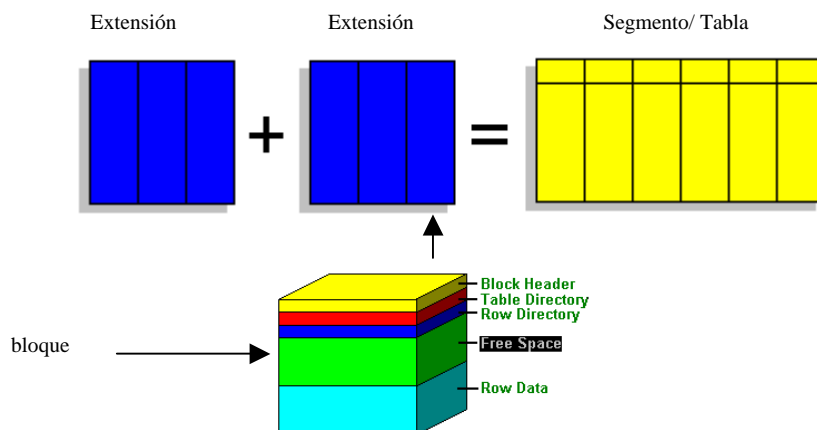
En el momento de la creación de una base de datos, el DBA debe planear o estimar los requerimientos de almacenamiento y también el nombre, tamaño y localización de los archivos de datos, junto con el número máximo de archivos de datos permitido para la base de datos.

El DBA puede crear varios espacios de tablas (tablespaces) en discos separados para planear el crecimiento de la base de datos y hacer una mejor administración de la base de datos.

Un objeto de datos, por su parte, es una estructura lógica que puede ser una tabla, un archivo de índice, un archivo temporal, un archivo de deshacer o un clúster. Estos objetos se almacenan físicamente en segmentos que se componen de extensiones (extents).



A su vez, una extensión está hecha de bloques que, de acuerdo con el sistema operativo subyacente, puede tener un número determinado de bytes y que el DBA especifica, en el momento de la creación de la base de datos. El tamaño del bloque es dependiente del sistema operativo y nunca puede ser menor al que éste maneja.



En una base de datos pueden existir otros objetos que **no** contienen datos como las vistas, los sinónimos y las secuencias. Sin embargo, todo objeto independientemente si contiene datos, o no, debe pertenecer a un esquema. Por eso, una colección de objetos de un usuario se denomina *esquema*.

Un objeto se puede crear en un esquema de tres formas:

1. Si un usuario da una orden de creación de un objeto, por defecto, el sistema lo crea en su propio esquema.
2. Copiando el objeto de otro usuario (al nombre de un objeto siempre se le antepone el nombre del esquema, por ejemplo `juan.empleado`) con una orden como:

Create table empleado as select * from juan.empleado;

3. Otro usuario lo crea para uno, como en la orden:

Create table maria.proyecto (codigo number primay key,

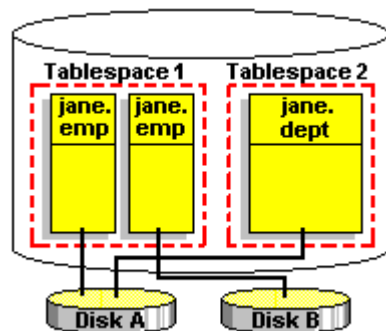
.....
.....)

tablespace planeacion

storage (initial 1000 next 1000

minextents 1 maxextents 6....)

Reglas para el almacenamiento de objetos en la base de datos



1. Un objeto puede almacenarse en uno más archivos de datos (datafiles) pero de un sólo espacio de tablas (tablespace).
 2. Dos objetos diferentes de un esquema, pueden estar en distintos tablespaces.
 3. Los objetos pueden almacenarse en múltiples discos. Por ejemplo, parte de jane.emp es almacenado en el archivo de datos 1 sobre el disco A y parte en el archivo de datos 2 sobre el disco B.
- **Archivos de control** (Control Files): tienen la descripción física y dirección de los archivos de la base de datos y de los archivos de rehacer, para el arranque correcto de la base de datos. En estos archivos se especifican cuáles datafiles conforman la base de datos para poder tener acceso a los datos o para poder recuperar la base de datos, ante una falla.

Los archivos de control se crean automáticamente cuando se da una orden `CREATE DATABASE` y no son editables, pues también se actualizan automáticamente.

- **Archivos de rehacer (redo log files)**, tienen los cambios hechos a la base de datos para la recuperación ante fallas o para el manejo de las transacciones. Poseen los valores antes de una transacción, la orden ejecutada y, opcionalmente, el valor después de la transacción. El principal propósito de estos archivos es servir de respaldo de los datos en la memoria RAM. Este conjunto de archivos debe estar conformado por dos grupos como mínimo y se recomienda que cada grupo esté almacenado en discos separados. El DBMS utiliza la técnica de ir sobreescribiendo sobre la información más vieja, cuando se agota el espacio en estos grupos de archivos.
- **Archivos fuera de línea (archived files)** son archivos opcionales donde se guarda información vieja de los archivos de rehacer, muy convenientes para los respaldos de la base de datos.

Los Procesos

Los procesos son programas que se ejecutan para permitir el acceso a los datos. Los procesos se cargan en memoria y son transparentes para los usuarios. Los procesos se clasifican en tres grupos: procesos de base, de usuario y procesos servidores.

- **Los Procesos de Base o de Soporte**

Los procesos de base (background), son los que se encargan de traer datos desde y hacia la SGA; mejorando el desempeño al consolidar las tareas que son impartidas por todos los usuarios. Cada proceso de base tiene su propia área de memoria. Los procesos de base o soporte son los siguientes.

DBWR : (Database writer) se encarga de manejar los “buffers” de memoria caché para que los procesos del usuario siempre encuentren unos de ellos disponibles. Es un proceso obligatorio que además escribe los bloques de datos modificados por los usuarios, en los archivos de datos que componen la B.D cuando el proceso LGWR le envía el mensaje de hacerlo.

LGWR : (Log writer) este proceso escribe datos desde la SGA a los archivos de rehacer (redo log files) que sirven en caso de fallas en la instancia. Este proceso es obligatorio y es el único encargado de escribir y leer en estos archivos. El proceso de

llenado de estos archivos es circular, por lo tanto antes de empezar a sobrescribir en uno de ellos, se marca un punto de verificación y LGWR envía la orden de escritura en los datafiles al proceso DBWR.

LCKn, Lock: (lock processes) El bloqueo es un proceso opcional. Efectúa los bloqueos entre instancias, en caso de ambientes con servidores paralelos (hasta con 10 servidores).

CKPT : (Check point) El punto de comprobación es un proceso opcional que ocurre cuando los usuarios conectados a la base de datos, hacen solicitudes de exámenes de datos.

SNPn : (Snapshot process) se encarga de refrescar los snapshots o réplicas de tablas que se usan principalmente en ambientes distribuidos.

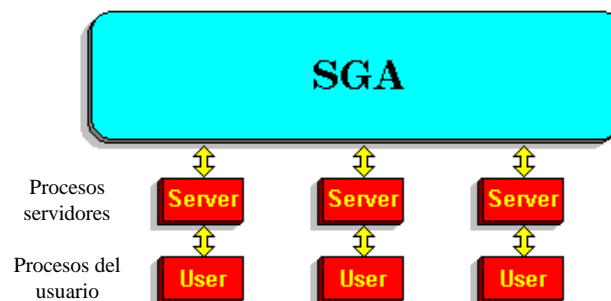
SMON : (System monitor) recupera el sistema ante una falla de la instancia.

RECO : (Recovery) recupera ante las fallas, en una transacción en ambientes distribuidos.

ARCH : (Archive) copia los registros de rehacer de la RAM en archivos de datos que permiten la recuperación cuando se presentan fallas de los medios magnéticos.

PMON : (Process Monitor) recupera ante una falla de un proceso de usuario; libera los recursos del proceso que falló.

- **Procesos del Usuario**



Cuando un usuario se conecta a la base de datos, se crea un proceso de usuario que se encarga de ejecutar el código de aplicación del usuario y manejar el perfil del usuario con sus variables de ambiente. Los procesos de usuario no se pueden

comunicar directamente con la base de datos, únicamente lo hacen a través de procesos servidores.

- **Procesos Servidores**

Ejecutan las órdenes SQL de los usuarios y llevan los datos al “database buffer cache”, para que los procesos del usuario puedan tener acceso a los datos. Se pueden tener distintas arquitecturas para trabajar en ORACLE, según los tipos de servidores: dedicados o multihilos.

Instancia de ORACLE

Se denomina *instancia* al conjunto de estructuras de memoria y procesos de fondo que acceden los archivos de bases de datos. Es posible que una misma base de datos sea accedida por múltiples instancias; cada una de ellas residiendo en una máquina diferente (esta es la opción de servidores paralelos de ORACLE).

El sistema de bases de datos ORACLE cuando inicia, sigue los siguientes pasos que se detallan a continuación.

1. **Iniciar la instancia.** Para hacer este paso, ORACLE lee el archivo de parámetros y configura la instancia, con base en ellos. En ese momento se crea la SGA y se activan los procesos de base; pero aún no se puede hacer nada. Es como encender un auto en neutra, listo para empezar a trabajar.
2. **Montar la base de datos.** Consiste en preparar el sistema para su uso, trayendo a la RAM el diccionario de datos; es como poner el sistema en primera, listo para recibir algunas órdenes del DBA.
3. **Abrir la base de datos.** En este momento se abren los archivos y los usuarios ya pueden tener acceso a los datos.

De acuerdo como se defina la instancia ORACLE, a través de sus parámetros, puede determinarse qué tan poderoso y grande es el motor. Los parámetros se definen en el archivo INIT.ORA, entre ellos se pueden mencionar:

db_block_buffers = número de bloques de bases de datos en la SGA. Existirá un buffer por cada bloque.

db_block_size= tamaño del bloque de la base de datos.

shared_pool_size = tamaño del área compartida “shared pool”, en bytes.

Además, allí se especifica el número de usuarios concurrentes, el número de transacciones concurrentes y los nombres de los archivos de control para la base de datos.

Estos parámetros se pueden ajustar, durante el proceso de afinamiento, porque ellos inciden en el desempeño del sistema. Algunos de los parámetros son específicos a una base de datos y por lo tanto deben ser cambiados antes de crear una base de datos. Se incluyen en éstos:

`database_name` = nombre de la base de datos.

`db_block_size` = tamaño del bloque.