

Administración de Bases de Datos

Tema 2
(parte 1 de 4)

Estructuras y Gestión del
Almacenamiento en un SGBD

Objetivos

- ❑ Conocer la arquitectura del SGBD Oracle
- ❑ Aprender a consultar el diccionario de datos
- ❑ Aprender los sistemas de arranque y parada de una base de datos
- ❑ Conocer la estructura lógica de almacenamiento de Oracle
- ❑ Aprender a gestionar los espacios de tablas
- ❑ Saber realizar conexiones a bases de datos

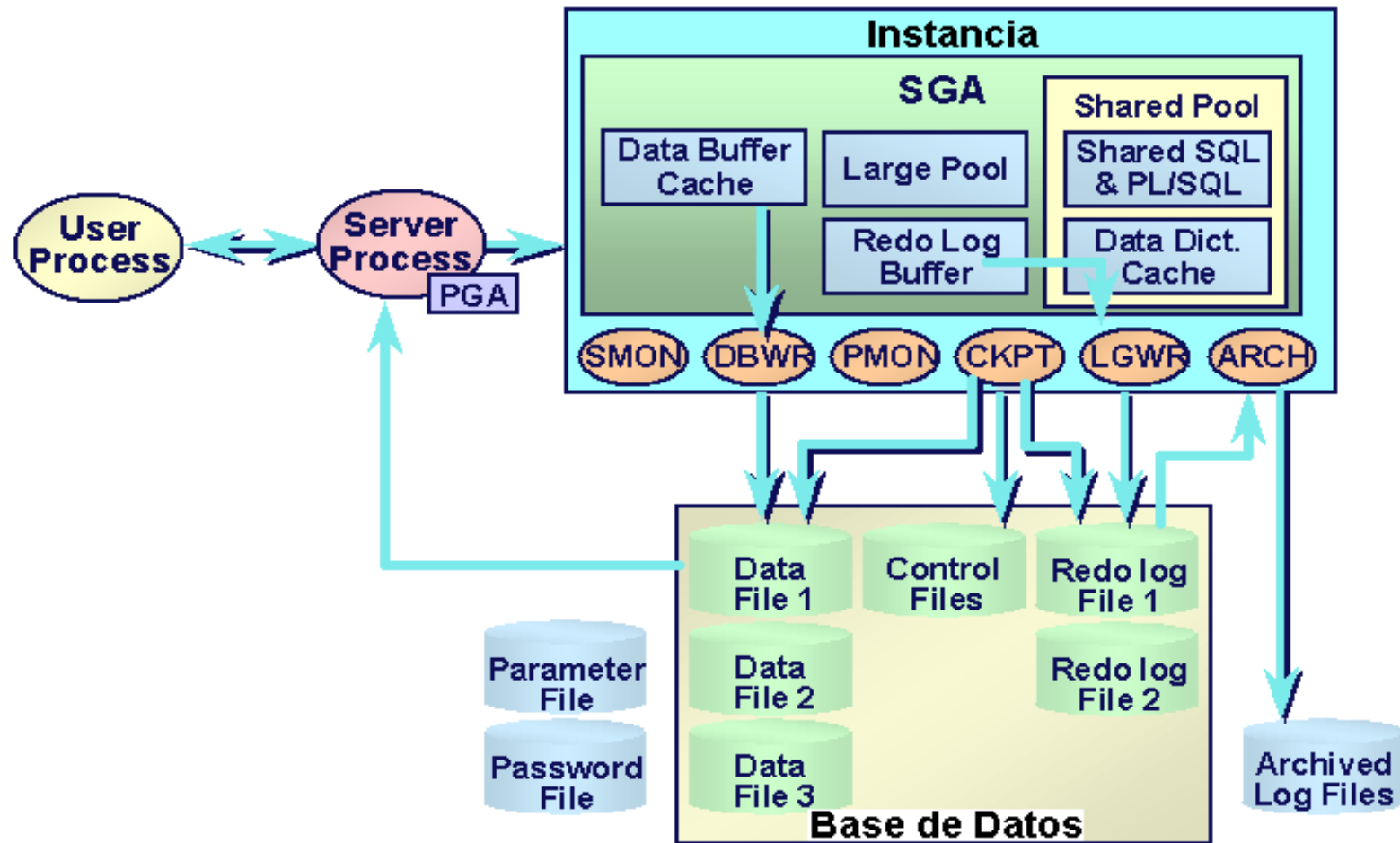
Contenidos

- 2.1. Servidor Oracle e instancia
- 2.2. El Diccionario de Datos de Oracle
- 2.3. Inicio y parada de una base de datos Oracle
- 2.4. Estructura de ficheros en Oracle
- 2.5. Estructura lógica de una BD Oracle
- 2.6. Conexión a una base de datos
- 2.7. Usuarios administradores de una BD Oracle
- 2.8. Gestión del almacenamiento

2.1. Servidor Oracle e instancia

- ❑ El Sistema Gestor de Bases de Datos Oracle está basado en la arquitectura cliente-servidor
- ❑ Una BD Oracle es un conjunto de datos organizados según el modelo relacional. Oracle proporciona la capacidad de almacenarlos y acceder a ellos de forma coherente
- ❑ A partir de la versión Oracle8, también se permite almacenar estructuras orientadas a objeto (tipos abstractos de datos, métodos, etc.)
- ❑ Cada **servidor** Oracle está constituido, al menos, por:
 - una **instancia**, formada por un gran bloque de memoria (**System Global Area, SGA**) y un conjunto de procesos que interactúan entre el SGA y los archivos de la base de datos residentes en disco
 - una **base de datos**, que es el lugar donde se almacenan los datos

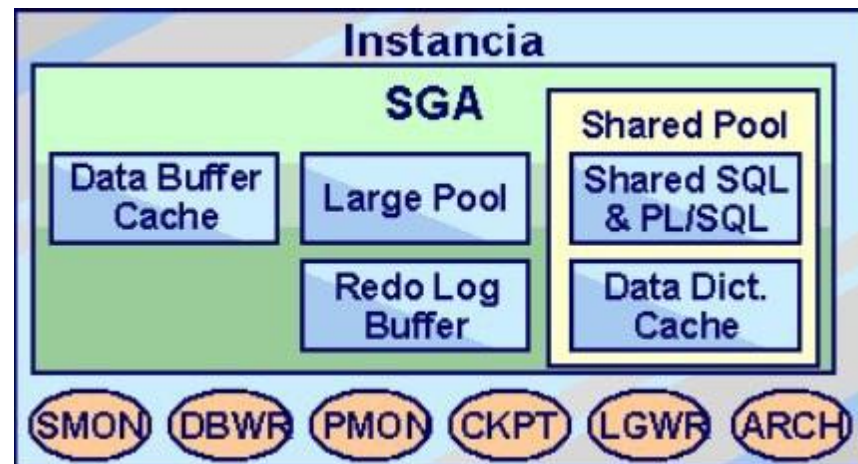
Arquitectura del SGBD Oracle



Consultar <http://www.eygle.com/blog/images/OracleDatabase11gArch.png> para ver una imagen más completa de la arquitectura de Oracle 11g

La Instancia Oracle

- ❑ Formada por **procesos Oracle** (procesos *background*) y la **memoria compartida** (System Global Area, SGA) necesaria para acceder, de forma eficiente, a la información almacenada en la BD
- ❑ Cada instancia Oracle abre una única Base de Datos
 - Con RAC (*Real Application Cluster*), distintas instancias pueden acceder a la misma BD
- ❑ En el equipo donde reside el servidor Oracle, la variable **ORACLE_SID** identifica a la instancia con la que estamos trabajando



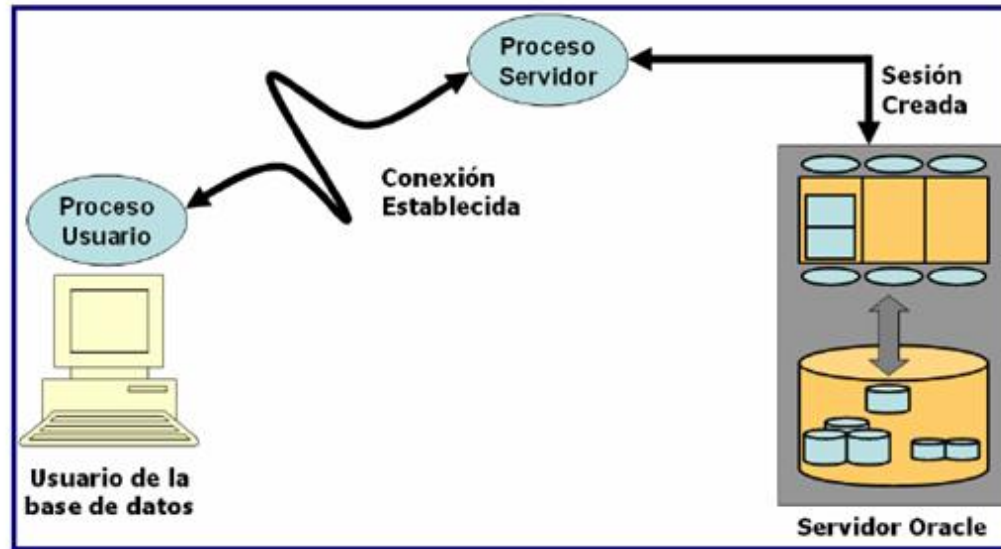
- ❑ Para arrancar la instancia, el servidor Oracle tiene que leer el **fichero de parámetros de inicialización**, cuya ubicación predeterminada es `$ORACLE_HOME/database`

- ❑ El fichero de inicialización puede ser de dos tipos:
 - **Init**: fichero de texto editable
 - **Spfile**: fichero binario no editable (recomendable a partir de la versión 9i)

- ❑ El nombre del archivo de inicialización suele incluir el nombre de la instancia. Si el nombre de la base de datos es ONUBA, el archivo de inicialización binario se llamará **spfileonuba.ora**

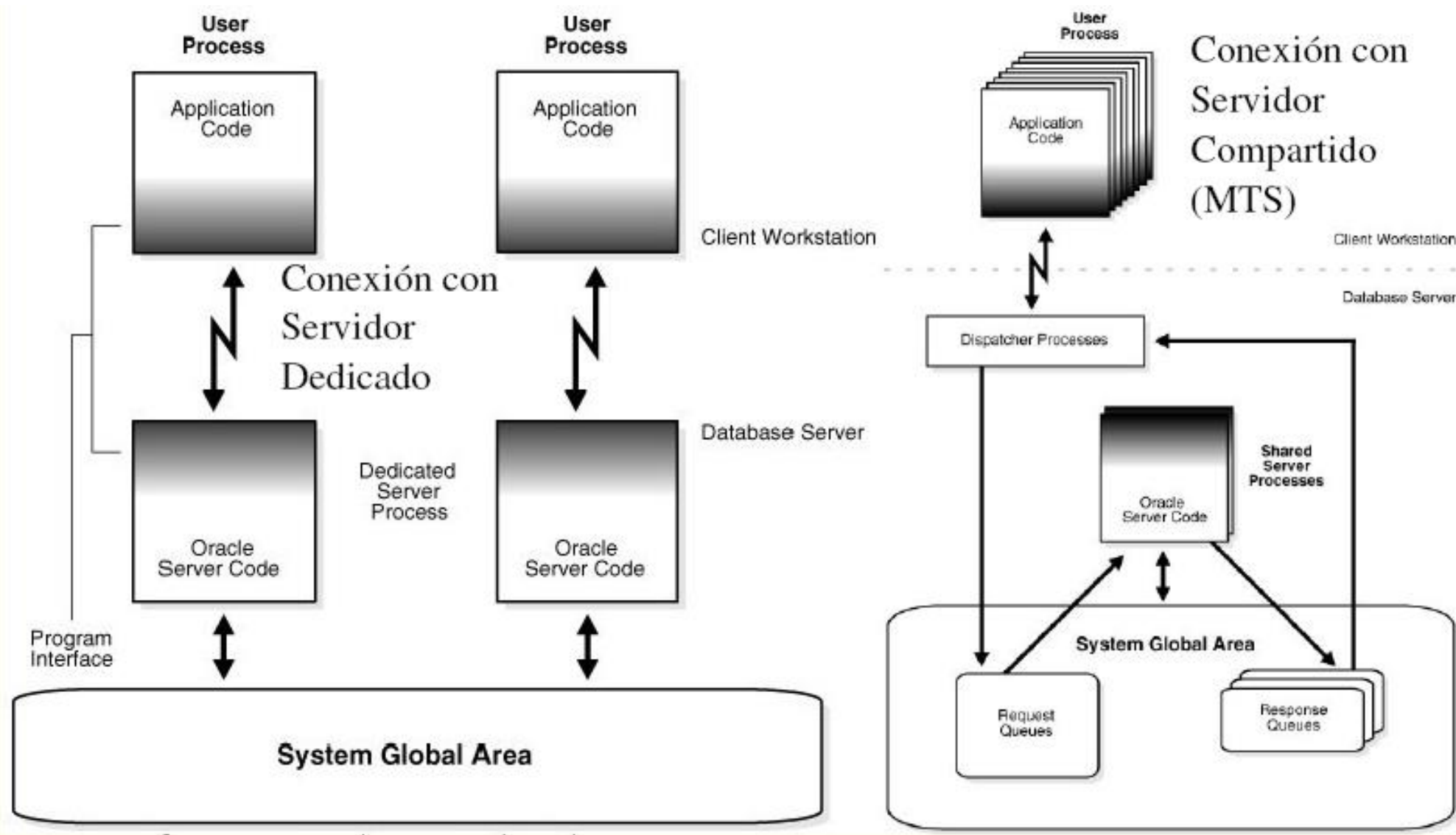
- ❑ El archivo de inicialización se lee durante el arranque de la instancia. Es tarea del administrador su creación y gestión para que la instancia arranque con los parámetros adecuados

Conexión a una Base de Datos Oracle



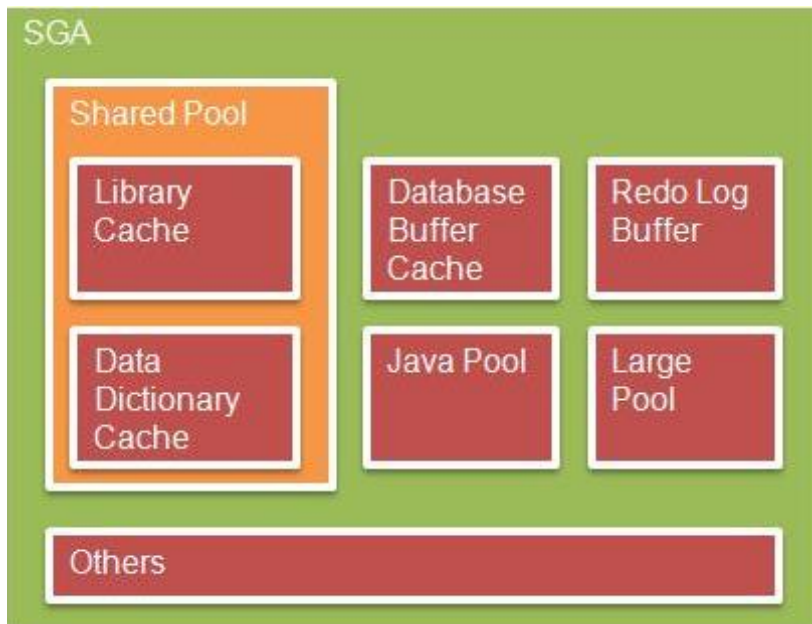
- ❑ **Proceso de usuario:** la ejecución de la aplicación que permite al usuario iniciar la conexión (sql*plus, sql developer o cualquier otro cliente)
- ❑ **Proceso servidor:** se crea en el SGBD cuando el usuario se conecta a la BD, y es el que realmente interactúa con la BD
- ❑ Una conexión de un proceso de usuario al SGBD es una sesión en la BD (puede haber varias del mismo usuario). Se inicia cuando el usuario se valida contra la BD y termina cuando el usuario se desconecta

Conexión a una Base de Datos Oracle



El Área Global del Sistema (*System Global Area, SGA*)

Zona de memoria con la información de la BD que pueden compartir los usuarios. Se crea cuando se empieza a usar una BD (se arranca la instancia) y se destruye al detenerse la instancia



Se dimensiona dinámicamente y su tamaño máximo se establece con el parámetro de inicialización **SGA_MAX_SIZE**

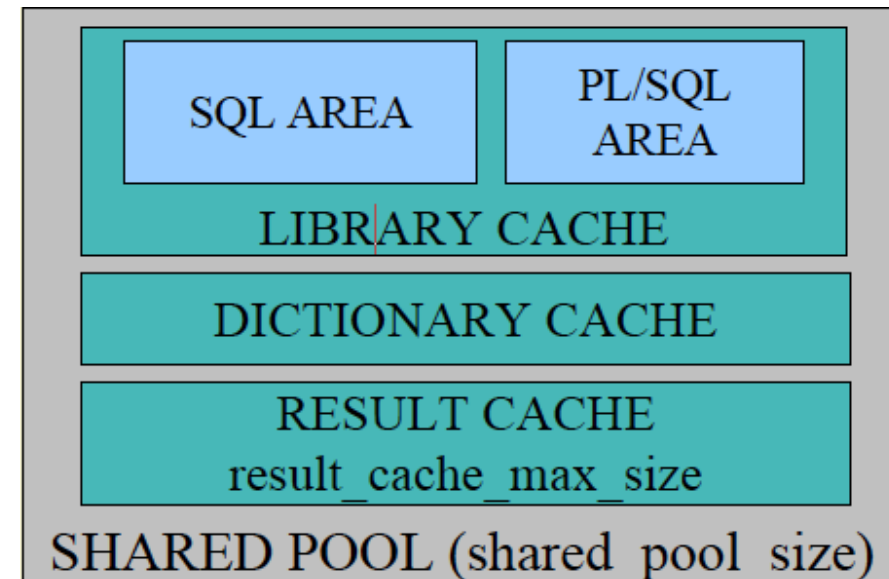
Si se especifica un valor al parámetro dinámico **SGA_TARGET**, Oracle ajusta, de forma automática, el tamaño de los componentes del SGA

La optimización de la SGA es uno de los puntos clave en el ajuste de las Bases de Datos

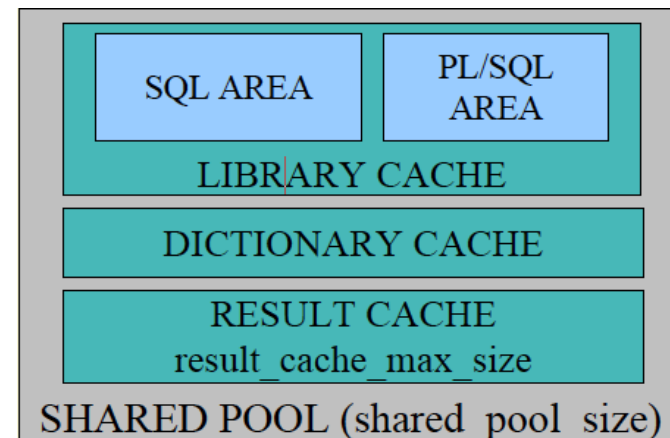
- ❑ El SGA está compuesta, fundamentalmente, por las siguientes estructuras de memoria:
 - **Shared Pool** (memoria compartida para consultas y diccionario de Oracle)
 - **Database Buffer Cache** (caché de búfer de datos)
 - **Redo Log Buffer** (búfer del registro de rehacer)
 - **Large Pool** (zona de memoria utilizada para manejar bloques de gran tamaño)
 - **Streams Pool** (zona de memoria para manejar la funcionalidad de stream – permite la circulación de información entre procesos)
 - **Java Pool** (zona de memoria utilizada por la máquina virtual Java)
 - **Result Cache** (caché para el resultado de consultas SQL, funciones PL/SQL, etc.)

Memoria compartida (*Shared Pool*)

- ❑ Está formada por las estructuras de memoria:
 - **Library Cache** (caché de biblioteca)
 - **Dictionary Cache** (caché de diccionario)
 - **Result Cache** (caché de resultados)
- ❑ Su tamaño viene determinado por el parámetro del fichero de inicialización **SHARED_POOL_SIZE**
- ❑ Desde la versión 10g se recomienda usar **SGA_TARGET**, dejando **SHARED_POOL_SIZE** a cero o indicando un valor mínimo
- ❑ Oracle utiliza un **algoritmo LRU** (*Least Recently Used*) para gestionar la caché

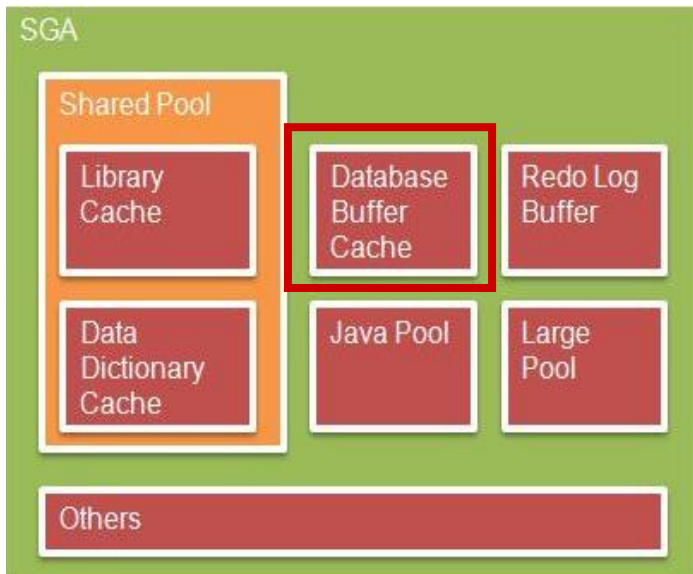


- ❑ En la **Library Cache** se almacena información sobre las sentencias SQL y PL/SQL usadas recientemente. Dado que está compartida por todos los usuarios, diferentes usuarios pueden compartir la misma instrucción SQL



- ❑ Está formada por dos estructuras:
 - **Shared SQL Area.** Se almacenan los planes de ejecución y los árboles sintácticos de las sentencias SQL
 - **Shared PL/SQL Area.** Contiene las unidades de programa compiladas y analizadas sintácticamente: procedimientos, funciones, paquetes y disparadores
- ❑ En la **Dictionary Cache** se almacenan las definiciones de datos usadas más recientemente: tablas, índices, columnas, usuarios, privilegios, etc. Esta información se genera y utiliza en la fase de análisis de las consultas y se obtiene de las tablas del diccionario de datos (DD). Es como una caché de datos para el DD
- ❑ Cuando se ejecuta una consulta SQL o una función, los resultados se buscan primero en la **Result Cache**. Los datos de la Result Cache correspondientes a objetos cuyos datos han sido modificados, quedan invalidados

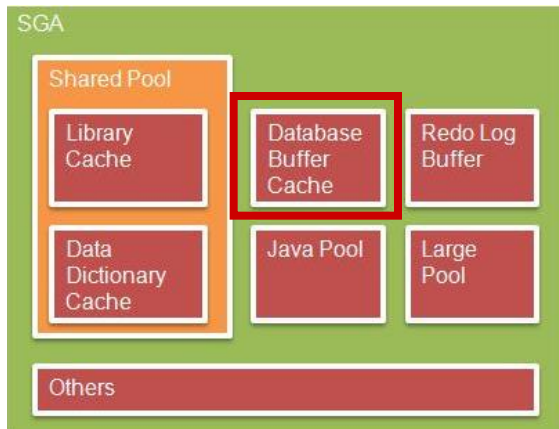
Caché de búfer de la Base de Datos (*Database Buffer Cache*)



- ❑ Almacena los últimos bloques usados: bloques de tablas, de índices y de segmentos de anulación
- ❑ Cuando se procesa una consulta, el proceso servidor busca los bloques de datos en la **Database Buffer Cache**. Si no los encuentra, los lee de los ficheros de datos y guarda una copia en la **Database Buffer Cache**

Principal función → reducir los accesos a disco

- ❑ Oracle utiliza un algoritmo LRU para gestionar la caché. Cuando se llena, elimina los datos utilizados menos recientemente
- ❑ Su tamaño depende, principalmente, del parámetro **DB_CACHE_SIZE** (desde la versión 10g es aconsejable usar SGA_TARGET, dejando DB_CACHE_SIZE a cero)
- ❑ Puede modificarse dinámicamente

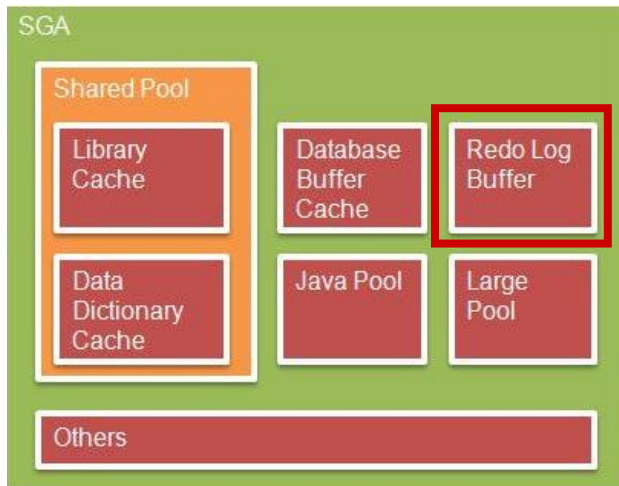


- ❑ Se pueden definir varias cachés de datos independientes:
 - **DB_CACHE_SIZE.** Dimensiona la caché por defecto. Su tamaño no puede valer cero. Siempre existe una caché por defecto
 - **DB_KEEP_CACHE_SIZE.** Dimensiona la caché donde se guardarán los bloques de tablas que se usan frecuentemente
 - **DB_RECYCLE_CACHE_SIZE.** Dimensiona la caché que almacena los bloques de las tablas que se usan con poca frecuencia

- ❑ Cuando se crea una tabla puede especificarse el área de búferes en la que deberán almacenarse los bloques de datos de la tabla. Se lleva a cabo mediante el parámetro **BUFFER_POOL** de la cláusula **STORAGE** de la tabla, utilizando las cláusulas **KEEP**, **RECYCLE** o **DEFAULT**

- ❑ **Ejemplo:** ALTER TABLE tabla1 STORAGE (BUFFER_POOL KEEP);

Búfer del registro de rehacer (*Redo Log Buffer*)

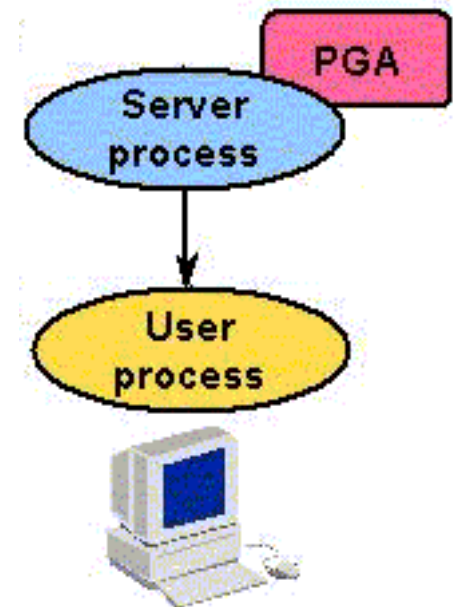


- ❑ Es un buffer que mantiene todos los cambios que han sido realizados sobre la base de datos por operaciones de *insert*, *update*, *delete*, *create*, *alter* y *drop*
 - ❑ Su principal función es la **recuperación** de los datos en caso de que **falle la instancia**
 - ❑ El tamaño viene determinado por el parámetro **LOG_BUFFER** (en bytes)
- ❑ Es un buffer circular. Cuando está lleno comienza de nuevo por el principio después de haber sido escrito en un fichero de actualización

El Área Global de Programa (*Program Global Area, PGA*)

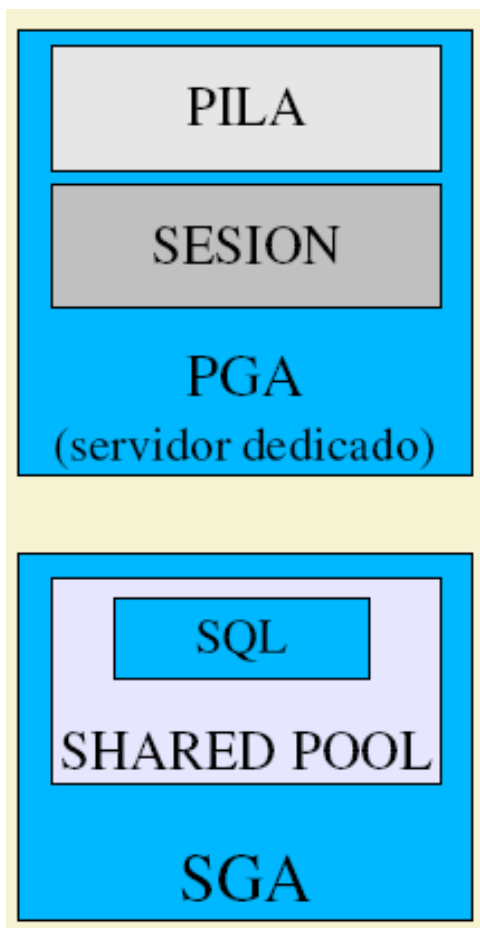
Área de memoria asignada a un proceso y de carácter privado para el mismo

- ❑ El PGA es asignado por Oracle cuando un usuario se conecta a una base de datos y crea una sesión. Se libera cuando el proceso termina
- ❑ El PGA contiene:
 - Una zona de trabajo SQL asignada dinámicamente para ciertas operaciones: ordenación, agrupación, etc.
 - La información de la sesión (privilegios del usuario, estadísticas de la sesión, etc.)
 - Las variables de la sesión
- ❑ La PGA total, asignada a todos los procesos de servidor, se denomina **PGA agregada** (*aggregated PGA*) o **PGA de la instancia** (*instance PGA*)
- ❑ El tamaño de la PGA agregada se define con el parámetro **PGA_AGGREGATE_TARGET**

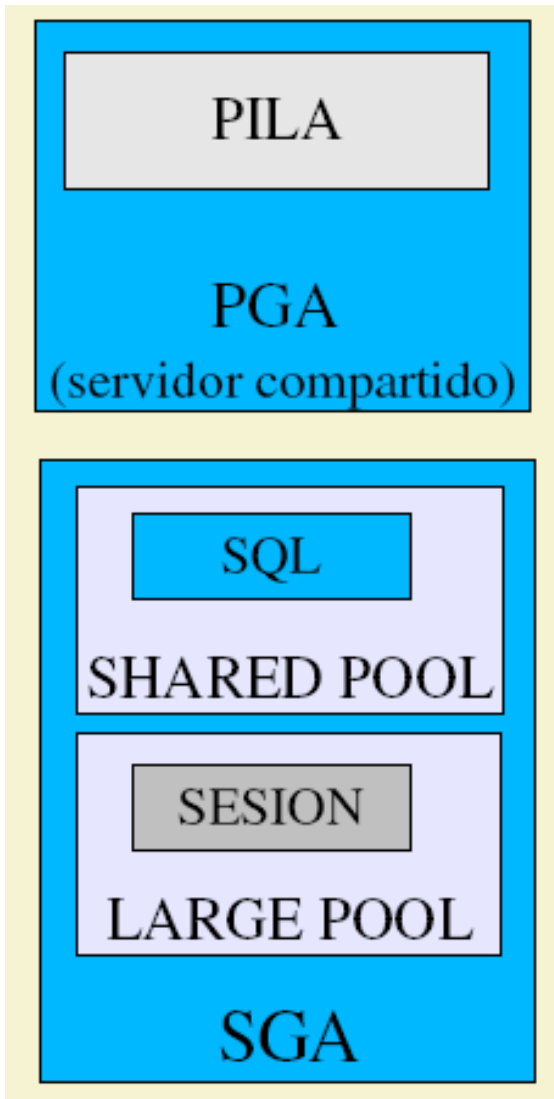


- ❑ La configuración del PGA depende de la configuración de conexión de la Base de Datos:

- **Configuración de servidor dedicado**
- **Configuración de servidor compartido**



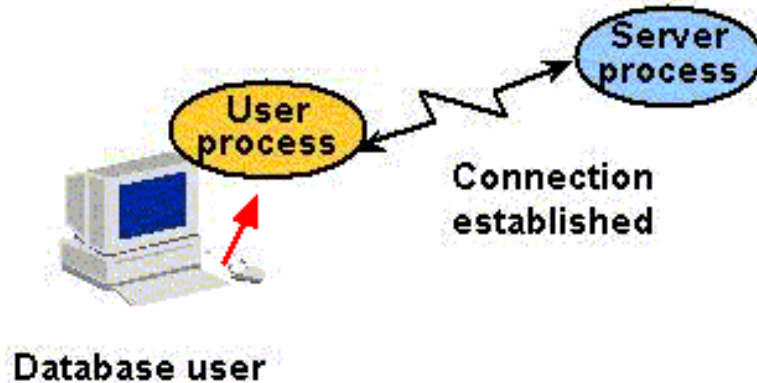
- ❑ En un entorno de servidor dedicado, cada proceso de usuario tiene su propia conexión con la BD
- ❑ El PGA contiene la memoria de sesión
- ❑ El PGA incluye, también, un área de ordenación que se utiliza cada vez que una solicitud de usuario necesita una operación de ordenación, de mezcla o de combinación



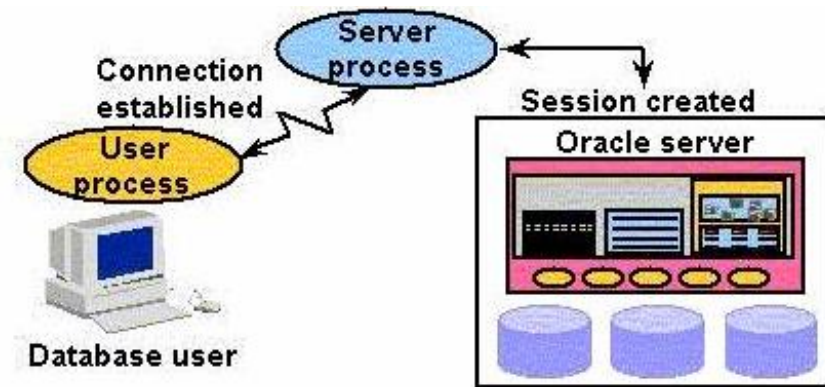
- ❑ En una configuración de servidor compartido, hay múltiples usuarios que comparten una conexión con la BD, minimizándose el uso de la memoria pero perjudicando al tiempo de respuesta a las solicitudes de los usuarios
- ❑ En estos entornos, parte de la información que se guarda en la PGA se almacena en el SGA
- ❑ Si se activa la *Large Pool* se almacenan en ella. En caso contrario, se quedan en la *Shared Pool*

Procesos de Oracle

- ❑ **Procesos de usuario.** Se ejecutan cuando un usuario solicita una conexión a la BD. Establecen la conexión con la BD pero no interactúan directamente con ella



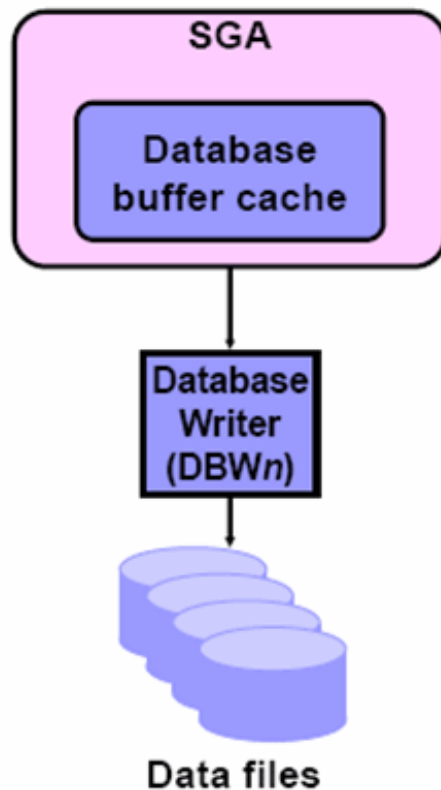
- ❑ **Procesos de servidor.** Realizan funciones para los usuarios, es decir, gestionan las solicitudes procedentes de los procesos de usuario.



- ❑ **Procesos en segundo plano (background).** Se encargan de realizar tareas de gestión de E/S, monitorización, etc. El parámetro **DIAGNOSTIC_DEST** del fichero de inicialización define el directorio donde se guardan los ficheros de traza de los procesos background

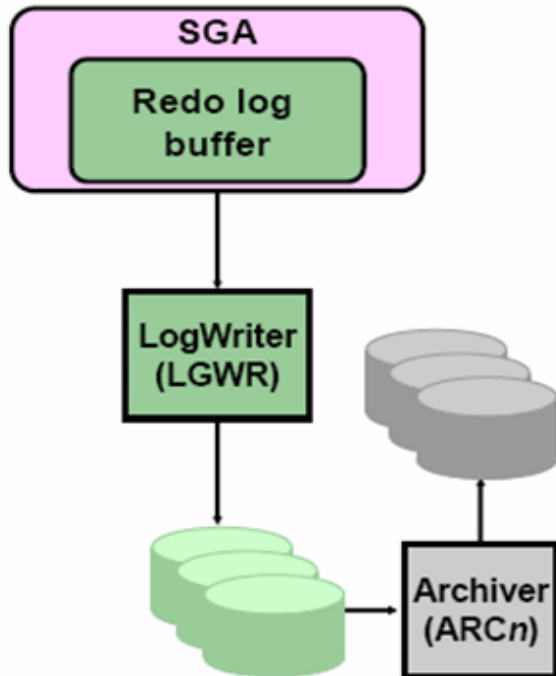
Procesos en segundo plano (background)

Escritor de Base de Datos (*Database Writer, DBWR*)



- ❑ Encargados de escribir los bloques modificados de la Database Buffer Cache en los archivos de datos situados en el disco
 - ❑ Utiliza un algoritmo LRU (el menos recientemente utilizado) para escribir los bloques más antiguos (los menos activos). De esta forma, los bloques más comúnmente solicitados se conservan en memoria
 - ❑ Pueden arrancarse hasta 20 procesos (DBW0..DBW9 y DBWa..DBWj). El número de procesos se controla con el parámetro **DB_WRITER_PROCESSES**
-
- ❑ La escritura se produce de forma asíncrona cuando ocurre un evento (*checkpoint*, número de búferes modificados alcanza un umbral, no quedan búferes libres, *timeout*, *tablespace offline*, *tablespace* en modo *readonly*, borrado de una tabla, etc.)

Escritor de registro (Log Writer, LGWR)



- ❑ Responsable de escribir los datos desde el búfer del registro de rehacer al fichero de rehacer en línea
- ❑ Una transacción no se considera completa hasta que LGWR escribe la información de rehacer en los ficheros de rehacer en línea

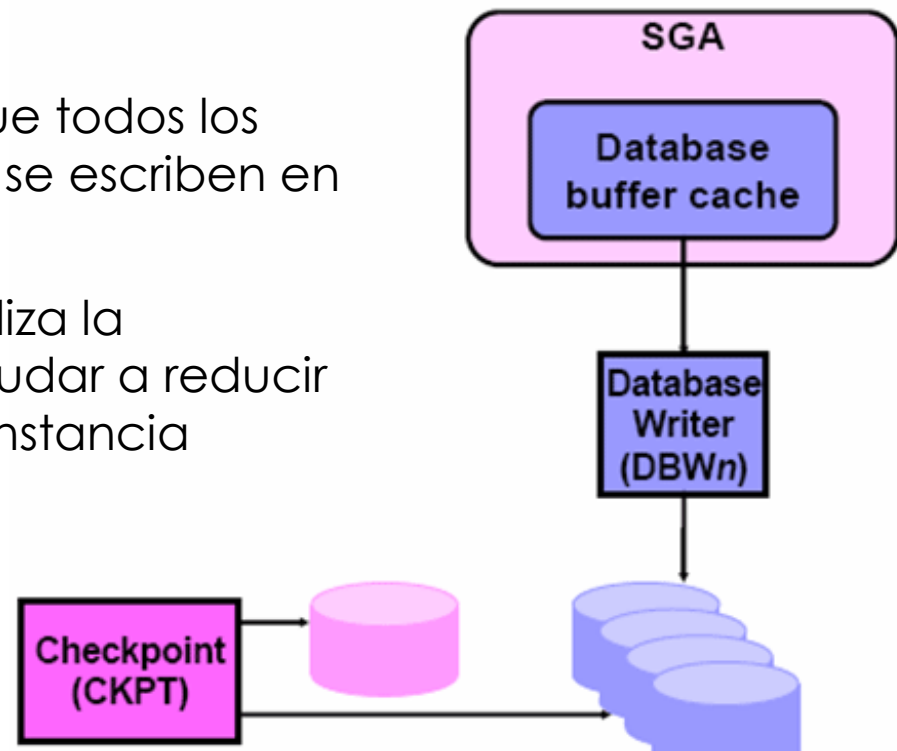
Proceso archivador (Archiver, ARCn)

opcional

- ❑ LGWR escribe en los ficheros de rehacer en línea de forma cíclica. Cuando se llena el primer archivo, se escribe en el siguiente y así sucesivamente. Una vez que el último archivo está lleno, LGWR empieza a sobrescribir los contenidos del primero. Cuando Oracle se ejecuta en modo **ARCHIVELOG**, se realiza una copia de cada uno de los archivos del registro de rehacer en línea cuando se llenan, asegurando que se registran todos los cambios realizados en la base de datos
- ❑ Se pueden arrancar hasta 10 procesos ARCn para cada instancia mediante el parámetro **LOG_ARCHIVE_MAX_PROCESSES**

Punto de control (*Check Point*, CKPT)

- ❑ Un punto de control es un suceso en el que todos los búferes modificados de la base de datos se escriben en los ficheros de datos
- ❑ Durante un punto de control, CKPT actualiza la cabecera del archivo de control para ayudar a reducir el tiempo requerido para recuperar una instancia



Monitor del sistema (*System Monitor, SMON*)

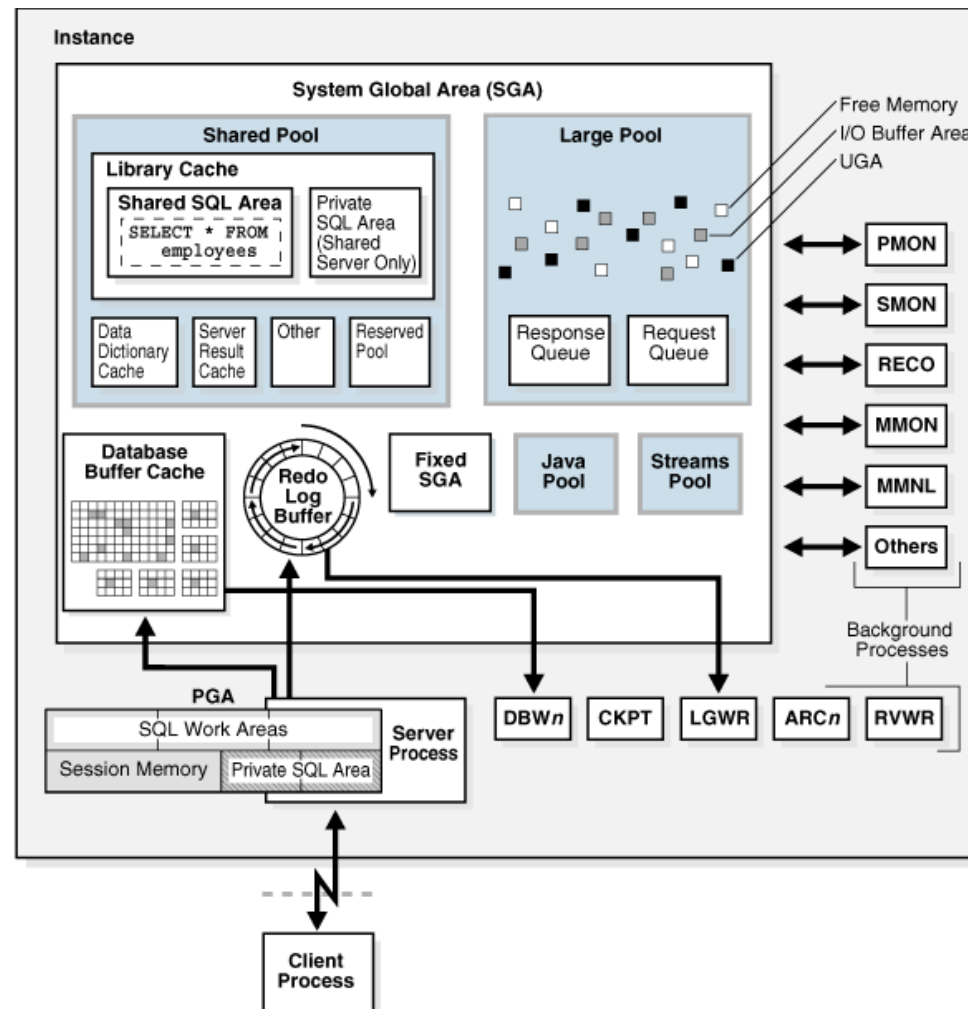
- ❑ Responsable de efectuar la recuperación de un error cuando se arranca la instancia después de algún fallo. Recupera las transacciones (registros de rehacer), purga los segmentos temporales de los espacios de tabla, etc.
- ❑ También se encarga de consolidar el espacio libre de los espacios de tabla combinando las extensiones libres, cuando los espacios de tabla están gestionados por el diccionario
- ❑ Además libera el espacio ocupado por segmentos temporales durante el procesamiento de sentencias SQL

Monitor de Procesos (*Process Monitor, PMON*)

- ❑ Responsable de controlar los procesos de usuario que acceden a la base de datos y recuperarlos después de producirse un error
- ❑ Por ejemplo, elimina los recursos que el usuario estuviera utilizando, deshace los cambios que se hubiesen realizado desde el último COMMIT, elimina los bloqueos de las filas afectadas de las tablas, etc.

Otros procesos en segundo plano

- Además de los procesos descritos, existen otros procesos en segundo plano, cada uno con su función correspondiente
- Consultar http://docs.oracle.com/cd/E11882_01/server.112/e40402/bgprocesses.htm#REFRN104 para ampliar información



EJEMPLOS DE EJECUCIÓN DE SENTENCIAS SQL



La gestión de la memoria compartida

- ❑ En la versión 11g, Oracle recomienda realizar una gestión automática de la memoria compartida
- ❑ El ABD asigna una cantidad de memoria total para la SGA y Oracle la reparte, automáticamente, entre sus componentes
- ❑ Se conoce como **Automatic Shared Memory Management** (ASMM)
- ❑ Oracle recomienda asignar un **máximo del 80% de la RAM de la máquina para el Servidor Oracle**. A su vez, de dicho 80%, recomienda asignar un **80% para la SGA** y un **20% para la PGA**
- ❑ Cuando la ASMM está activada, Oracle dimensiona automáticamente los componentes de la SGA (*Database Buffer Cache, Shared Pool, Large Pool, etc.*) en función de sus necesidades y los adapta, dinámicamente, en función de la carga del sistema

- ❑ Para activar la ASMM sólo es necesario definir el parámetro **SGA_TARGET**, que especifica la cantidad de memoria total asignada a la SGA

- ❑ Para que ASMM sea operativa es necesario que el parámetro **STATISTICS_LEVEL** tenga el valor **TYPICAL** (por defecto) o **ALL**

- ❑ Cuando la ASMM está activada, Oracle dimensiona automáticamente los siguientes componentes de la SGA en función de sus necesidades y los adapta, dinámicamente, en función de la carga del sistema:
 - Shared pool (parámetro **SHARED_POOL_SIZE**)
 - Large pool (parámetro **LARGE_POOL_SIZE**)
 - Java pool (parámetro **JAVA_POOL_SIZE**)
 - Database buffer caché (parámetro **DB_CACHE_SIZE**)
 - Streams pool (parámetro **STREAMS_POOL_SIZE**)
 - SGA fija (sin parámetro)

- ❑ El resto de componentes se ajustan manualmente y consumen espacio de SGA_TARGET:
 - Redo Log Buffer (parámetro **LOG_BUFFER**)
 - Otros *pools* de la *Database Buffer Cache*:
 - **LOG_BUFFER**
 - **DB_KEEP_CACHE_SIZE**
 - **DB_RECYCLE_CACHE_SIZE**
 - **DB_nK_CACHE_SIZE**

- ❑ En esta configuración, si los parámetros **SHARED_POOL_SIZE**, **LARGE_POOL_SIZE**, **JAVA_POOL_SIZE**, **DB_CACHE_SIZE** y **STREAMS_POOL_SIZE** no se especifican, Oracle los pone a 0 y ajusta su tamaño de forma automática y dinámica. Si se especifica un valor para estos parámetros, estamos indicando el valor mínimo que deben tener

La gestión de la memoria de la instancia

- ❑ La versión 11g introduce una gestión automática de la memoria de la instancia (SGA + PGA)
- ❑ Se conoce como **Automatic Memory Management** (AMM)
- ❑ Oracle reparte automáticamente la memoria asignada a la instancia entre la SGA (y sus componentes) y la PGA
- ❑ Para activar la AMM es suficiente con definir el parámetro **MEMORY_TARGET**, que especifica la cantidad de memoria total asignada a la instancia
- ❑ También se puede definir el tamaño máximo de memoria utilizable por la instancia mediante el parámetro **MEMORY_MAX_TARGET**
- ❑ Se puede modificar dinámicamente el tamaño de la memoria de la instancia hasta llegar al máximo
- ❑ La SGA se gestiona siguiendo los mismos principios que los utilizados en ASMM (por ejemplo, dimensionar manualmente ciertos componentes)

Ejemplo

```
MEMORY_MAX_TARGET = 900M
MEMORY_TARGET = 800M
SGA_MAX_SIZE = 500M
SGA_TARGET = 300M
DB_CACHE_SIZE = 128M
PGA_AGGREGATE_TARGET = 64M
```

Se trata de un ejemplo didáctico. En la práctica, Oracle aconseja no especificar ningún parámetro relativo a un componente de memoria gestionado automáticamente que no sean MEMORY_MAX_TARGET y MEMORY_TARGET

- ❑ La memoria disponible para el ajuste automático es igual a **MEMORY_TARGET – un gránulo (SGA fija y Redo Log Buffer)**, es decir, $800 - 4 = 796\text{M}$
- ❑ Oracle reparte esa cantidad entre la PGA y los componentes de la SGA gestionados automáticamente
- ❑ Con esta configuración se garantizan un mínimo de 128M para el Database Buffer Cache, 64M para la PGA y una SGA comprendida entre 300M y 500M
- ❑ La memoria libre es de 100M ($\text{MEMORY_MAX_TARGET} - \text{MEMORY_TARGET}$). En caso necesario se puede aumentar la MEMORY_TARGET para aprovechar esta memoria disponible

El fichero de Parámetros de Inicialización

- ❑ Una instancia Oracle se inicia usando un fichero de parámetros de inicialización. Los **parámetros de inicialización** permiten a la instancia asignar la cantidad de memoria a la SGA y PGA, indicar la ubicación de los ficheros de datos y de control, etc.
- ❑ Cada base de datos posee un fichero de parámetros que corresponde sólo a esa base de datos
- ❑ Hasta la versión 8, el fichero de parámetros era un fichero de texto (**init**)
 - puede estar en el cliente o en el servidor
 - por defecto, su nombre es **initSID.ora**
 - estático
- ❑ Desde la versión 9i, se utiliza un fichero de parámetros binario (**spfile**)
 - Se encuentra ubicado en el servidor
 - por defecto, su nombre es **spfileSID.ora**
 - dinámico
 - su ubicación por defecto es ORACLE_BASE\database

- ❑ El fichero de parámetros de servidor (*spfile*) se genera a partir de un fichero de parámetros de texto (*init*)
- ❑ Hay dos tipos de parámetros:
 - **Explícitos:** los que se indican en el fichero de parámetros
 - **Implícitos:** el resto, que tomarán un valor por defecto
- ❑ En un fichero de texto, la forma de indicar valor a los parámetros es **parametro = valor**
- ❑ El símbolo # indica el comienzo de un comentario, pudiendo estar al principio o en medio de la línea.
- ❑ Una lista de valores se indicará entre paréntesis, separando los valores por comas
- ❑ Para indicar un valor de tipo cadena de caracteres hay que encerrarlo entre comillas simples.

```
db_name=Onuba           # nombre de la base de datos
# ficheros de control de la base de datos
control_files=("C:\app\uhu\oradata\onuba\control01.ctl",
              "C:\app\uhu\flash_recovery_area\onuba\control02.ctl")
```


- ❑ Una vez generado el fichero binario, los valores de los parámetros pueden ser modificados. La mayoría de las veces, el cambio surtirá efecto cuando se vuelva a iniciar la base de datos
- ❑ Algunos parámetros son **dinámicos**, es decir, pueden modificarse mediante las sentencias **ALTER SYSTEM** y **ALTER SESSION** mientras que la instancia está en ejecución
- ❑ Sintaxis de ALTER SYSTEM para modificar los parámetros del fichero de inicialización:

```
ALTER SYSTEM SET parámetro = valor [SCOPE = SPFILE|MEMORY|BOTH] [DEFERRED]
```

```
ALTER SYSTEM SET parámetro = valor [SCOPE = SPFILE|MEMORY|BOTH] [DEFERRED]
```

❑ **SCOPE.** Especifica el ámbito del cambio

- **SPFILE.** Modifica los valores de los parámetros dinámicos pero los cambios, aunque quedan grabados en el fichero, sólo serán efectivos cuando se reinicie la base de datos. Se comporta de la misma forma para los parámetros estáticos (es la única opción válida para este tipo de parámetros)
- **MEMORY.** Modifica los valores de los parámetros dinámicos inmediatamente pero sus cambios no quedan registrados en el fichero de inicialización. Si se ha iniciado la base de datos utilizando un fichero de parámetros (*init*), ésta es la única cláusula SCOPE que se puede especificar
- **BOTH.** El cambio es inmediato y persistente
- Cuando se utiliza el *spfile* para iniciar la base de datos, **BOTH** es la opción por defecto. Si se ha iniciado con *init*, la cláusula predeterminada es **MEMORY**

❑ La cláusula **DEFERRED** indica que los cambios serán efectivos en futuras sesiones. Las sesiones actuales conservan el valor antiguo

- ❑ Algunos parámetros dinámicos se pueden modificar mediante la sentencia **ALTER SESSION**. Sólo se modifica el valor de los parámetros durante la sesión que invoca la sentencia. Los cambios no afectan a futuras sesiones de la instancia
- ❑ Un parámetro es modificable mediante la sentencia **ALTER SESSION** si su valor en el campo **ISSES_MODIFIABLE** de la vista dinámica **V\$PARAMETER** es TRUE
- ❑ Sintaxis de **ALTER SESSION**:

```
ALTER SESSION SET parámetro = valor
```

- ❑ Un fichero binario *spfile* se genera a partir de un fichero de parámetros de texto:

```
CREATE SPFILE [= 'nombre_spfile'] FROM PFILE [= 'nombre_pfile']
```

- ❑ También se puede realizar la operación inversa:

```
CREATE PFILE [= 'nombre_pfile'] FROM SPFILE [= 'nombre_spfile']
```

❑ Algunos de los parámetros y su significado (I)

DB_NAME	Estático	Nombre global de la base de datos. Este valor debe ser asignado antes de la creación y debe coincidir con el de la sentencia CREATE DATABASE. No debe tener más de 8 caracteres
INSTANCE_NAME	Estático	Nombre de la instancia
CONTROL_FILES	Estático	Sirve para asignar los nombres a los ficheros de control CONTROL_FILES = nombreFichero [, nombreFichero [...]]
BACKGROUND_DUMP_DEST	Dinámico ALTER SYSTEM	Directorio donde se escriben los ficheros de traza de los procesos en segundo plano y el alert Ej: background_dump_dest = C:\app\uhu\diag\rdbms\onuba\onuba\trace
USER_DUMP_DEST	Dinámico ALTER SYSTEM	Directorio donde se escriben los ficheros de traza de los procesos de usuario Ej: background_dump_dest = C:\app\uhu\diag\rdbms\onuba\onuba\trace
COMPATIBLE	Estático	Versión del servidor con la que debe ser compatible la instancia
DB_CACHE_SIZE	Dinámico ALTER SYSTEM	Tamaño de la caché de buffer de la base de datos
DB_BLOCK_SIZE	Estático	El tamaño del bloque de datos por defecto se especifica en este parámetro. Este parámetro no puede cambiar después de la creación, aunque se pueden definir tamaños de bloque distintos para los objetos de la BD
SHARED_POOL_SIZE	Dinámico ALTER SYSTEM	Tamaño de la memoria compartida

❑ Algunos de los parámetros y su significado (II)

SGA_MAX_SIZE	Estático	Tamaño máximo hasta el que puede crecer la SGA
DB_nK_CACHE_SIZE	Dinámico ALTER SYSTEM	Caché para espacios de tabla con otro tamaño de bloque Ej: DB_16K_CACHE_SIZE = 1000
NLS_LANGUAGE	Dinámico ALTER SESSION	Especifica el lenguaje. Si no se especifica ninguno, se toma el del S.O.
NLS_TERRITORY	Dinámico ALTER SESSION	Especifica el país. Al fijar se establece también la moneda, la fecha, los separadores de miles y decimales
NLS_NUMERIC_CHARACTERS	Dinámico ALTER SESSION	Especifica el separador decimal y el de miles Ej: NLS_NUMERIC_CHARACTERS = ",."
NLS_DATE_FORMAT	Dinámico ALTER SESSION	Especifica el formato de la fecha Ej: NLS_DATE_FORMAT = "MM/DD/YYYY"
NLS_DATE_LANGUAGE	Dinámico ALTER SESSION	Lenguaje de las fechas para abreviaturas, mes y día
DB_CREATE_FILE_DEST	Dinámico ALTER SESSION ALTER SYSTEM	Directorio predeterminado para todos los archivos (datos, temporales, control y redo). El directorio debe existir

❑ Algunos de los parámetros y su significado (III)

UNDO_MANAGEMENT	Estático	Indica qué sistema de "deshacer" se utilizará (automático o mediante segmento de rollback) Ej: undo_management = AUTO
UNDO_TABLESPACE	Dinámico ALTER SYSTEM	Indica el espacio de tablas que se utilizará para "deshacer" Ej: undo_tablespace = UNDOTBS1
UNDO_RETENTION	Dinámico ALTER SYSTEM	Indica el número de segundos que se debe mantener en el sistema de "deshacer" la información ya validada Ej: undo_retention = 10800
REMOTE_LOGIN_PASSWORDFILE	Estático	Especifica si Oracle valida por medio de un fichero de password o por medio del S.O. Ej: remote_login_passwordfile = EXCLUSIVE
PROCESSES	Estático	Número máximo de procesos del S.O. que pueden estar conectados a Oracle concurrentemente. Su valor mínimo debe ser 6 (5 para los procesos en background y 1 para un proceso de usuario). Si se prevé tener 50 usuarios concurrentemente, este valor debe ser 55
DB_FILES	Estático	Máximo número de ficheros de datos que pueden estar abiertos en la base de datos. Su máximo depende del S.O.

Todos los parámetros de inicialización en:

http://docs.oracle.com/cd/B28359_01/server.111/b28320/toc.htm

2.2. El Diccionario de Datos de Oracle

- ❑ El Diccionario de Datos es un conjunto de tablas y vistas asociadas donde se almacena toda la información sobre los objetos que componen la BD, así como la estructura lógica y física de la misma
- ❑ Las tablas del diccionario son propiedad del usuario **SYS** y se almacena en el espacio de tablas **SYSTEM**. El usuario **SYSTEM** es propietario de algunas vistas sobre esas tablas
- ❑ Cuando se crea una BD se crean muchos objetos del diccionario de datos. Estas tablas y vistas se pueden agrupar según su función. Las más utilizadas son:
 - Las que mantienen información de los objetos de los usuarios: tablas, vistas, secuencias, restricciones, índices, disparadores, procedimientos, funciones, paquetes, etc.
 - Las que mantienen información de seguridad: usuarios y privilegios, roles, etc.
 - Las que mantienen información sobre la asignación y uso del espacio
 - Las tablas de rendimiento dinámico o funcionamiento (**V\$**). Almacenan el estado actual de la BD. Se modifican continuamente (por ejemplo cuando se conecta un usuario, cuando se realiza un bloqueo de una tabla, etc.). Se utilizan para monitorizar el estado de la BD

- ❑ El Diccionario de Datos se carga en memoria en el componente **Dictionary Cache** de la **Shared Pool**
- ❑ Existen dos tipos de objetos:
 - **Tablas base**
 - Se crean en el momento de la creación de la base de datos y almacenan la descripción de todos los objetos de la base de datos
 - Poseen una estructura muy compleja y nunca se deben modificar sus datos ni su estructura
 - **Vistas del diccionario**
 - Son vistas sobre las tablas base que permiten acceder a toda la información de una manera más cómoda y sencilla. Se crean al lanzar el script **catalog.sql** y se clasifican en:
 - **USER_**. Describen los objetos que posee el usuario que las consulta. A través de ellas, cada usuario accede sólo a la información de sus objetos (aquellos que ha creado).
 - **DBA_**. Contienen información de toda la base de datos y sólo son accesibles por el administrador de la base de datos
 - **ALL_**. Describen los objetos que son accesibles por el usuario que las consulta
 - **V\$. (Vistas dinámicas)**. Reflejan el estado interno del sistema y son útiles para que el administrador de la base de datos pueda analizar las prestaciones. No se almacenan como tablas, pero muestran en forma de tablas las estructuras de datos del sistema. Se crean al arrancar la instancia y residen en memoria. Cuando se cierra la BD (y por tanto la instancia), desaparecen y, con ellas, su contenido

❑ **DICTIONARY (DICT)**

- Proporciona el nombre y un comentario de todas las tablas y vistas del DD a las que tenemos acceso desde nuestro usuario

```
SELECT Table_Name, SUBSTR(Comments,1,50)
FROM DICT
WHERE Table_Name LIKE '%VIEWS%';
```

❑ **DICT_COLUMNS**

- Proporciona información sobre las columnas de las vistas del DD. Tiene 3 columnas: nombre de la tabla, nombre de la columna y comentario

```
SELECT Table_Name, Comments
FROM DICT_COLUMNS
WHERE Column_Name = 'BLOCKS' AND
      Table_Name LIKE 'USER%';
```

❑ Algunas vistas del diccionario de datos:

- V\$DATABASE (Base de datos)
- V\$INSTANCE (Instancia)
- V\$SGA (SGA)
- V\$SGASTAT (SGA detallada)
- V\$BUFFER_POOL (Buffers en la caché de datos)
- V\$SQLAREA (Sentencias SQL)
- V\$PROCESS (Procesos)
- V\$BGPROCESS (Procesos background)
- V\$DATAFILE (Ficheros de datos de la BD)
- V\$CONTROLFILE (Ficheros de control de la BD)
- V\$LOGFILE (Ficheros redo log de la BD)
- V\$SESSION (SESIONES)
- DBA_TABLESPACES (Espacios de tablas de la BD)
- DBA_SEGMENTS (Segmentos que hay en los espacios de tablas)
- DBA_EXTENTS (Extensiones que componen los segmentos)
- DBA_USERS (Usuarios de la BD)

Más información:

<http://www.ss64.com/orad/>

<http://ss64.com/orav/index.html>

http://docs.oracle.com/cd/B28359_01/server.111/b28320/toc.htm