

Configuración de la base de datos en oracle

Índice

1.- INTRODUCCIÓN.....	1
2. INICIO Y PARADA DE LA BASE DE DATOS.....	1
2.1 INICIADA.....	2
2.2.-MONTADA.....	2
2.3.-ABIERTA.....	2
2.4 CERRADA.....	3
3.- FICHERO DE PARÁMETRO.....	3
3.1.- Archivo PFILE (initSID.ora).....	4
3.2- Modificación de los valores.....	4
4- ARCHIVOS DE DIAGNÓSTICOS.....	5

1.- INTRODUCCIÓN

Para realizar un instalación correcta y obtener un rendimiento óptimo de un sistema de gestor de base de datos es importante conocer tanto la estructura física como la estructura lógica de los componentes del sistema gestor de datos que vamos a usar para poder sacarle el máximo partido a su configuración.

También es fundamental darle los valores óptimos a los archivos de configuración del SGBD

Es importante conocer las distintos estados por lo que pasa la base de datos y conocer que operaciones se puede ejecutar en ella según el estado en el que se encuentre.

2. INICIO Y PARADA DE LA BASE DE DATOS

Para hacer que una base de datos sea accesible a todos los usuarios hay que arrancar una instancia y abrir la base de datos con dicha instancia. Se puede decir que hay tres grandes fases en el proceso de apertura de la base de datos:

- Inicio de la instancia.
- Montaje de la base de datos.
- Apertura de la base de datos.

Para el proceso de parada de la base de datos también podemos decir que existen tres fases:

- Cierre de la base de datos.
- Desmontaje de la base de datos
- Parada de la instancia.

Veamos los estados de la instancia y el proceso de apertura poco a poco.

2.1 INICIADA

En este estado, se arranca la instancia: lee el fichero de parámetro (spfile), localiza los ficheros de control, crea e inicializa la SGA y finalmente arranca los procesos servidores, pero no se monta la base de datos.

Con este estado se puede crear una base de datos manualmente o se puede restaurar un fichero de control.

Para llegar a este estado desde la instancia parada ejecutamos STARTUP NOMOUNT.

2.2.-MONTADA

En este estado los procesos y recursos de la instancia se asocian a la base de datos, se leen los archivos de control para localizar los ficheros de datos y los redolog pero no los abre.

La instancia monta la base de datos y la bloquea, verificando que ninguna otra instancia ha montado la misma base de datos. Esta situación se utiliza principalmente para el mantenimiento de la base de datos. Algunas de las operaciones a realizar cuando la base de datos esta montado son:

- Habilitar o deshabilitar el modo archive log de la base de datos.
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE NOARCHIVELOG;
- añadir, renombrar o eliminar archivos de REDOLOG
- Llevar a cabo un full database recovery

Desde la instancia parada llegamos a este estado con el comando STARTUP MOUNT. Y desde el estado anterior con el comando ALTER DATABASE MOUNT.

Sólo se pueden conectar la base de datos usuarios con permisos SYSDBA, SYSOPER O SYSBACKUP.

2.3.-ABIERTA

Durante esta etapa la instancia abre la base de datos, bloquea los ficheros de datos, y abre todos los ficheros redo log. Se comprueba la coherencia de la base de datos. Si alguno de los ficheros de datos que se va a abrir no se encuentra o está dañado, la base de datos genera un error y no se abre.

Si la instancia abre la base de datos después de una terminación anormal, o después de una caída, se ejecutará automáticamente el proceso SMON para realizar la restauración de la instancia.. Al final de esta etapa la instancia esta disponible para su uso.

Llegamos a este estado con STARTUP OPEN Si la base de datos esta cerrada y con ALTER DATABASE OPEN si estamos en el escalón anterior.

Existen otros estado de apertura restringida de la base de datos como:

- abrir la base de datos en modo lectura. Para ello se usa la orden
“STARTUP OPEN READ ONLY” o “ALTER DATABASE OPEN READ ONLY”

- abrir la base de datos en modo restringido. Sólo se permite las conexiones a usuario que tengan el privilegio create session y restricted session.
“STARTUP OPEN RESTRICT” o “ALTER SYSTEM ENABLE RESTRICTED SESSION”
- Para desactivar el modo restrictivo se usa la orden:
“ALTER SYSTEM DISABLE RESTRICTED SESSION”

La vista para obtener información de la instancia es V\$INSTANCE, y la columna que me dice el estado actual de la instancia es STATUS.

2.3.1. Arranque automático

Oracle proporciona herramientas para arrancar la base de datos automáticamente.

En plataformas Windows, el comando oradim permite parar/arrancar la instancia desde la consola de sistema operativo.

En sistemas Unix, existe dos ficheros de scripts: dbstart y dbshut, que permite automatizar la apertura y parada de la instancia

2.4 CERRADA

La instancia está inactiva y la base de datos cerrada. Para pasar a este estado se utiliza el comando shutdown independientemente del escalón en que nos encontremos. Hay 4 tipos de cierre: normal, inmediato, transaccional y abort.

- **SHUTDOWN NORMAL**: se impide el acceso a la base de datos, espera que todos los usuarios completen sus peticiones y se desconecten del servidor. Purga todo los buffers de datos y cachés de redo log, actualizando los ficheros de datos y de redo log, se elimina los bloqueos de ficheros, se completan las transacciones en marcha y sincroniza los ficheros de control y de datos. En resumen, cierra la base de dato, desmonta la base de datos y para la instancia con cuidado. Es la opción recomendada.
- **SHUTDOWN TRANSACTIONAL**: se impide el acceso a la base de datos, se espera que los usuarios terminen sus transacciones en curso, fuerza un checkpoint, además de cerrar todos los ficheros; Pero, desconecta automáticamente a los usuarios
- **SHUTDOWN IMMEDIATE**: también se impide el acceso a la base de datos, pero las sentencias en procesos son terminadas inmediatamente, cualquier transacción no confirmada con commit es vuelta atrás con un rollback y la base de datos es parada. La única desventaja de utilizar esta opción es que oracle no espera a que los usuarios se desconecten , sin embargo, la base de datos será consistente y no se necesitará recuperación en el siguiente arranque.
- **SHUTDOWN ABORT**: sólo se debe ejecutar en situaciones de emergencia. Cuando se realiza un shutdown de este tipo las sentencias SQL son terminadas bruscamente y las transacciones no confirmadas no son vuelta atrás. Para la base de datos con esta opción requiere la recuperación en la siguiente vez que arranque la base de datos (lo hace automáticamente el proceso SMON).

3.- FICHERO DE PARÁMETRO

Cuando se arranca la instancia oracle, el primer fichero que lee el servidor oracle es el fichero de inicialización SPFILE.ORA (ejemplo:spfileaulax.ora.).

Este fichero contiene los parámetros que crean la instancia inicial. Contiene la siguiente información:

- Tamaño y composición de la de la instancia. (zona de memoria SGA)
- Nombre de la base de datos asociada a la instancia.
- Ubicación y los nombres de los archivos de control.
- Información de los segmentos deshacer.

El Spfile es un fichero binario, que no se puede modificar de forma manual. Pero oracle permite gestionar el SPFILE a través del comando ALTER SYSTEM.

La ubicación por defecto del archivo spfile es en:

- Linux: \$ORACLE_HOME/dbs.
- Windows: \$ORACLE_HOME%\database

3.1.- Archivo PFILE (initSID.ora)

En las versiones anteriores de oracle 9i, no existe el archivo spfile.ora, sino que el archivo de parámetros de la instancia se llama initSID.ORA. Es un archivo de texto que se puede abrir con un editor de texto cualquiera y modificarlo de forma manual.

NOTA: En las versiones 9i y posteriores sigue existiendo este fichero por compatibilidad

Un Pfile especificado puede sustituir al spfile cuando se arranca la instancia (por supuesto, hay que indicarle en el arranque que use el pfile en vez del spfile.ora)

Para crear un PFILE de un SPFILE o viceversa, se usa la siguientes sentencia:

Crear pfile de spfile:

CREATE PFILE [=’nombre’] FROM SPFILE [=’nombre’]

Crear spfile de pfile:

CREATE SPFILE [=’nombre’] FROM PFILE [=’nombre’]

Si se emiten los nombres, toma los valores por defecto.

Cuando se arranque con el pfile, la base de dato no podrá abrir el nuevo spfile hasta el siguiente arranque.

3.2- Modificación de los valores

Para ver los parámetros y sus valores actuales, se usa las siguientes vistas: V\$PARAMETER y V\$SPPARAMETER.

Los campos *name* y *value*, muestran el nombre del parámetro y su valor respectivamente.

V\$PARAMETER: muestra los parámetros y valores actuales de la instancia que se está ejecutando.

V\$SPPARAMETER: muestra los parámetros y valores del spfile guardado en disco.

Por lo general, estos valores serán los mismos. Pero eso no ocurre siempre, ya que algunos parámetros pueden ser cambiados durante la ejecución de la instancia. Sin embargo, otros tienen valor fijo que se les asigna a la hora de iniciar la instancia, son los llamados parámetros estáticos.

Para saber si un valor de un parámetro se puede modificar hay que observar los valores (algunos booleanos) de los siguientes campos de la vista v\$parameter.

- ISMODIFIED.- Si ha sido modificado
- ISDEFAULT.- Si tiene un valor por defecto
- ISSES_MODIFIABLE.-
 - o true: Es modificable con alter session
 - o false: No es modificable con alter session
- ISSYS_MODIFIABLE .- Es modificable con Alter system
 - o IMMEDIATE: el cambio se realiza de manera inmediata (Parámetros dinámicos).
 - o DEFERRED: El cambio se realizados en “caliente” pero solo tendrá efecto para futuras sesiones, no para la actualmente conectada.
 - o FALSE: No puede cambiarse

Para modificar el valor de los parámetros se usa las ordenes:

ALTER SESSION SET parámetro = valor, o

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

- Scope = Spfile indica que sólo queremos modificar el parámetro en el spfile, con lo que el cambio no se realizará hasta el siguiente arranque
- Scope = memory, el cambio sólo se produce en memoria MEMORY
- Scope = BOTH, el cambio se produce en memoria y en el archivo.

Alter sesión, permite modificar el parámetro a nivel de sesión.

Los parámetros estáticos sólo se pueden cambiar con Alter system scope=spfile o both

4- ARCHIVOS DE DIAGNÓSTICOS

Hay muchas situaciones en las que como administradores es necesario tener información acerca de lo que está haciendo el servidor de la base de datos. Para realizar esta tarea se utilizan los archivos de diagnóstico.

Existen dos tipos de archivos de diagnósticos en los que el administrador puede obtener información sobre si algo sale mal, son los archivos TRACE y el archivo de alerta:

TRACE

Estos archivos se usan para rastrear información de los procesos background y actividad de los usuarios

Los archivos de traza pueden ser de procesos usuario o procesos servidores (background)

- trazas de los background

Registran errores producidos en los procesos background de la instancia: LGWR, DBWR, SMON, etc. Estos archivos tienen la extensión TRC.

La ruta de estos archivos, se generan en el directorios indicado por el parámetro de inicialización BACKGROUND_DUMP_DEST.

- trazas de procesos usuarios

Recoge estadísticas de seguimiento de sentencias SQL o errores en la sesiones de usuario. Se genera donde apunta el parámetro USER_DUMP_DEST.

El tamaño máximo de los ficheros trazas se puede limitar con el parámetro “max_dump_file_size”.

Otros procesos como el listener (proceso de red) también generan ficheros de traza.

Archivo de alerta

Este archivo es generado automáticamente por oracle cuando se crea una instancia. Es un archivo que registra errores y sucesos de la gestión de la base de datos cronológicamente.

En este archivo oracle escribe los siguientes mensajes de información o errores:

- Creación de la base de datos
- Todos los inicio y parada de la base de datos
- Parámetros del init sin valores por defecto
- Arranque de los procesos background
- Escritura o almacenamiento de los archivos redo log
- Creación de espacios de tabla y segmentos rollback
- Comandos Alter (alter database, alter tablespace, etc.)
- Errores críticos (incidencias)
- Errores de bloque corrupto (ORA-1578)

Existe dos versiones de este archivo: una versión xml “log.xml” y una versión texto “alert_SID.log”

El archivo log.xml está ubicado en un directorio dado por el parámetro de inicialización DIAGNOSTIC_DEST, el cual por defecto apunta a la ruta \$ORACLE_BASE/diag dentro de la cual encontrarán una carpeta denominada dblms.

El archivo xml se renombra automáticamente como log_n.xml cuando llega a un tamaño determinado, sin embargo el archivo de texto crece sin límite, por lo que se aconseja limpiarlo regularmente para que no sea muy grande.

Una de las funciones de un DBA es revisar periódicamente estos archivos para encontrar posibles errores que hayan ocurrido.