

TEMA 3

Iniciar y detener una Base de Datos.

3.1. EL COMANDO STARTUP

En una operación normal de inicio, el comando `startup`, inicia la instancia, monta y posteriormente, abre una base de datos. Un usuario puede acceder a la BD una vez que esta se encuentra abierta.

Existen varios mecanismos para levantar una instancia:

- A través de SQL *Plus
- A través de Oracle restart
- A través del uso de Recovery Manager.
- Empleando Oracle Enterprise Manager Cloud Control

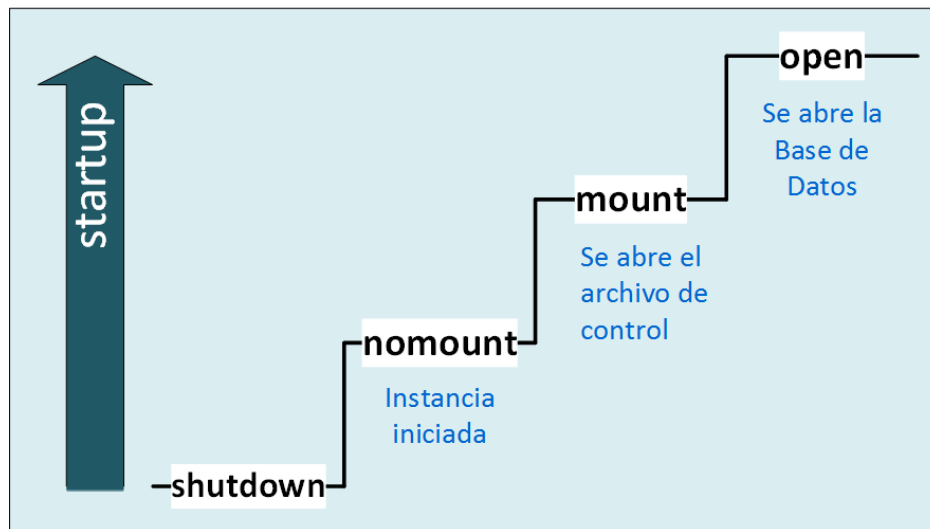


Para el caso del comando `startup`, como se comentó anteriormente, este puede ir acompañado del parámetro `pfile` para especificar la ruta absoluta donde se encuentra el SPFILE:

```
startup pfile = /u01/oracle/dbs/mySpfile.ora
```

3.1.1. Modos de inicio

El inicio de una instancia puede ser realizada en varios modos descritos en la siguiente imagen y tabla:



Modo	Descripción
<code>startup nomount</code>	<p>Inicia la instancia sin montar la BD. La instancia aún no se asocia a una BD.</p> <ul style="list-style-type: none">• Se crean todas las áreas de memoria• Los procesos de background son creados• Se ubica y se lee el archivo de parámetros considerando el siguiente orden y nombres de archivos: <p>Orden 1: <code>spfileSID.ora</code> Orden 2: <code>spfile.ora</code></p>

Modo	Descripción
	<p>Orden 3: <code>initSID.ora</code></p> <p>En caso de no encontrar alguno de estos archivos, se produce error.</p> <ul style="list-style-type: none"> En este modo no es posible acceder a la base de datos, inclusive puede no existir. Se emplea generalmente para: <ul style="list-style-type: none"> Crear una nueva base de datos Re-crear archivos de control. Se escribe en el <i>alert.log</i> los parámetros configurados. En este modo se utilizan únicamente 2 archivos: <i>archivo de parámetros</i> y el <i>alert log</i>.
startup mount	<p>Inicia la instancia, monta la BD. La instancia se asocia con una BD.</p> <ul style="list-style-type: none"> Ubica y lee el contenido del archivo de control y lo lee. Su ubicación lo obtiene del parámetro <code>control_files</code>. Si existen varias copias estos deben ser idénticos, todos disponibles. En este estado la BD permanece cerrada, no se tiene acceso aun a ella. El DBA puede realizar algunas tareas administrativas. Se leen las ubicaciones y nombres de los data files y los online redo logs. <p>Nota: Solo se obtienen las ubicaciones. Los archivos no se validan aun. Esto se hace en la siguiente etapa.</p>
startup open	<p>Inicial la instancia, monta la BD y se abre:</p> <ul style="list-style-type: none"> A partir de las ubicaciones y nombres de los data files y online redo logs, se lee su contenido y se revisa si estos archivos están sincronizados. Es decir, todas las transacciones fueron sincronizadas. Si esto ocurre, se procede a abrir la BD. De lo contrario inicia el proceso de recuperación (se revisará más adelante). Se obtiene un tablespace tipo undo para ser utilizado. Si existen múltiples tablespaces undo configurados, en el parámetro <code>undo_tablespace</code> indica cuál de ellos emplear. Si el parámetro no fue configurado, el primer tablespace tipo undo existente es el que se empleará. La BD puede ser accedida a partir de este momento.
startup force	<ul style="list-style-type: none"> Realiza un inicio forzado posterior a la ocurrencia de algún problema al detener o iniciar.
startup open recover	<ul style="list-style-type: none"> Abre la BD y procede inmediatamente a realizar una posible recuperación de la base de datos: <i>media recovery</i>.

El comando `startup` sin especificar el modo es equivalente a `startup open`.

3.1.2. Problemas que pueden ocurrir al inicio.

El inicio normal de la instancia y apertura de la BD puede verse afectada por problemas comunes como son:

- Archivo de control dañado o no disponible
- Si alguno de los archivos especificados en el archivo de control no están disponibles, la BD no se podrá montar.

3.1.3. Restringir el acceso durante el inicio.



- Para cualquiera de los modos anteriores, es posible restringir el acceso a usuarios comunes.
- El acceso en modo restringido solo se otorga a usuarios administradores.
- El modo restringido es generalmente útil para realizar alguna de las siguientes actividades:
 - Realizar operaciones de `import` o `export` de datos.
 - Realizar cargas masivas de datos, por ejemplo, empleando SQL *Loader
 - De manera temporal restringir el acceso a datos por parte de los usuarios.
 - Realizar operaciones de migración y actualización del software de la BD.
- Un usuario debe contar con los siguientes privilegios para poder acceder a una base de datos en modo restrictivo.

```
create session
create restricted session
```



- Para iniciar y/o montar una base de datos en modo restringido se emplea:

```
startup restrict
```

- Opcionalmente se puede emplear en cualquiera de las fases `nomount`, `mount`.
- Para deshabilitar el modo restringido se emplea:

```
alter system disable restricted session
```

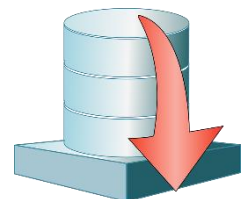
3.1.4. Modificando el modo de inicio de la instancia.

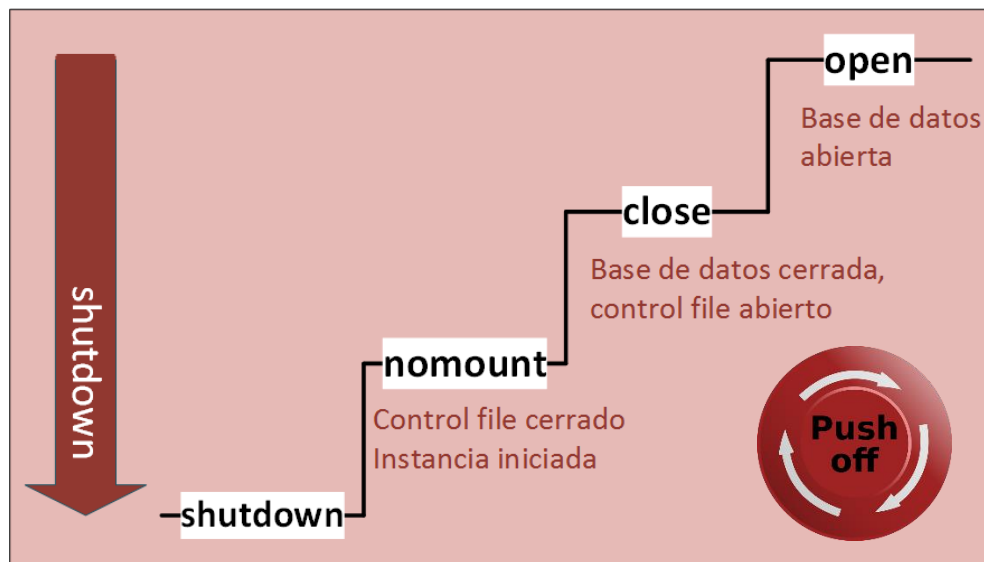
Los siguientes comandos permiten moverse a un nuevo modo durante el inicio de una base de datos:


Comando	Descripción
<code>alter database mount</code>	Cambia el estado actual de la instancia a modo <code>mount</code>
<code>alter database open</code>	Cambia el estado actual de la instancia a modo <code>open</code>
<code>alter database open read only</code>	<ul style="list-style-type: none"> • Permite realizar consultas a la base de datos pero evita cualquier cambio o modificación a los Online Redo Logs files, o a los data files, es decir, se evita que transacciones de usuarios modifiquen datos • Existe un escenario donde sí se pueden modificar estos archivos: Recovery.
<code>alter database open read write</code>	<ul style="list-style-type: none"> • Es el valor por default, equivalente a la instrucción <code>alter database open</code>.

3.2. EL COMANDO SHUTDOWN.

- Empleado para detener una instancia de base de datos.
- Bajo un escenario normal de operación, la base de datos pasa por varias etapas o fases para ser detenida.
- La siguiente imagen y tabla ilustra estas etapas.








Fase	Status de la BD	Descripción
1	Base de datos cerrada. 	<ul style="list-style-type: none"> En cualquier modo excepto el modo <code>abort</code>, todos los datos que se encuentren en la SGA se escriben o se sincronizan con los data files y online redo logs. Posteriormente, estos data files y online redo logs son cerrados. Los archivos de control permanecen abiertos. En modo <code>abort</code>, los datos no se sincronizan. La sentencia <code>alter database close</code> se emplea para cerrar una base de datos.
2	Base de datos no montada.	<ul style="list-style-type: none"> En este estado, la base de datos es desasociada con la instancia. El archivo de control es cerrado. La instancia permanece en memoria.
3	Base de datos detenida.	<ul style="list-style-type: none"> La instancia se detiene, la memoria se libera y los procesos de background son terminados. En algunos casos la instancia puede no ser detenida de forma completa (existen remanentes), puede emplearse en este caso <code>shutdown abort</code>.

3.2.1. Modos para detener una instancia.

De forma similar al comando `startup`, existen varios modos para detener una instancia. Las 3 etapas o estados descritos anteriores aplican a todos los siguientes modos excepto el modo `abort`.

Comando	Descripción
<code>shutdown normal</code>	Representa la opción por default. <ul style="list-style-type: none"> No más conexiones se permiten Si existen usuarios actualmente conectados, la instancia espera a que cierren su sesión para detener la instancia. Se realiza el cierre ordenado de la BD y sincronización, por lo que no se requerirá proceso de recuperación.
<code>shutdown immediate</code>	<ul style="list-style-type: none"> No más conexiones se permiten Para las sesiones actuales no nuevas transacciones son permitidas. Para las transacciones no concluidas se aplica <code>rollback</code> y se concluyen.

	<ul style="list-style-type: none"> No espera a que usuarios conectados se desconecten. Las sesiones se cierran inmediatamente. Se realiza el cierre ordenado de la BD y sincronización, por lo que no se requerirá proceso de recuperación.
<p>shutdown transactional</p> 	<ul style="list-style-type: none"> No más conexiones se permiten Para las sesiones actuales no nuevas transacciones son permitidas. A diferencia del modo anterior, el manejador espera a que transacciones en curso terminen. Si al término de todas las transacciones aún existen usuarios conectados, cierra sus sesiones y los desconecta. Se realiza el cierre ordenado de la BD y sincronización, por lo que no se requerirá proceso de recuperación.
<p>Shutdown abort</p> 	<p>Cierra la base de datos de forma instantánea.</p> <ul style="list-style-type: none"> Esta opción debería emplearse únicamente en casos de emergencia, ante la inminente ocurrencia de un problema o falla. Este modo realiza las mismas acciones que shutdown immediate. La diferencia es que archivos y buffers no son sincronizados por lo que el próximo inicio seguramente va a requerir proceso de <u>recuperación</u>.

- Para los modos que requieren esperar a que usuarios terminen sus transacciones o sus sesiones, la instancia espera hasta una hora para que terminen. De no hacerlo, la instancia las terminará y producirá el error `ORA-01013: user requested cancel of current operation`.

Los modos se pueden resumir en la siguiente tabla:

Comportamiento de la BD	abort	immediate	transactional	normal
¿Permite nuevas conexiones?	NO	NO	NO	NO
¿Espera hasta que las sesiones terminen?	NO	NO	NO	SI
¿Espera a que las transacciones actuales terminen?	NO	NO	SI	SI
Realiza una operación de checkpoint y cierra archivos	NO	SI	SI	SI

Ejercicio en clase 1

Considerar la vista `v$instance` empleada para mostrar el estado de una instancia y su base de datos.



- Generar las sentencias SQL necesarias para mostrar el nombre de la instancia, la fecha de inicio a nivel de segundos, el status de la instancia y el status de la base de datos cuando la instancia se encuentre iniciada en modo `nomount`.
- Cambiar el status de la instancia a `mounted`, ejecutar nuevamente la consulta del punto anterior.
- En el modo `mounted`, ¿qué valor deberá contener el parámetro `undo_tablespace` a nivel de sesión y a nivel de instancia?
- Cambiar el status de la instancia a `open`, e indicar el valor del parámetro `undo_tablespace` a nivel de sesión y a nivel de instancia?

- E. Cerrar la base de datos empleando la instrucción `alter database close`. Revisar nuevamente el status de la instancia. Con base a este valor, ¿Qué otra instrucción SQL permitiría cerrar una base de datos?
- F. Detener la instancia. Posteriormente abrirla pero en modo de solo lectura. Ejecutar sentencias SQL para demostrar que efectivamente la base de datos se encuentra en modo solo de lectura. Posteriormente detener la instancia empleando el modo `abort`.

3.3. INACTIVAR UNA BASE DE DATOS.

- Permite a una base de datos entrar a un estado en el que únicamente es posible realizar operaciones de administración. Por ejemplo, ejecutar transacciones, consultas, ejecutar programas PL/SQL pero realizadas **únicamente** por usuarios con **privilegios de administración**.
- Inactivar una base de datos impide a usuarios que no cuenten con privilegios de administración conectarse a la instancia.
- Si al momento de intentar inactivar existen sesiones activas (ejecutando transacciones, consultas, programas PL/SQL), a estas sesiones se les permite continuar hasta que termine la actividad en ejecución.
- Si al momento de intentar inactivar existen sesiones inactivas, estas no se desconectan, pero si intentan realizar consultas, transacciones, programas PL/SQL, la sesión será puesta en modo de espera hasta que la BD sea activada nuevamente. En resumen, *una BD inactiva impide a sesiones volverse activas*.
- Usuarios **sin** privilegios de administración que intenten entrar a sesión serán puestos en modo de espera hasta que la BD sea reactivada.
- Usuarios **con** privilegios de administración pueden entrar y crear sesiones de forma normal.
- El status de inactividad de la instancia se puede consultar en `v$instance` empleando el atributo `active_state`.
- Únicamente el usuario `sys` o `system` pueden inactivar la instancia ejecutando la siguiente instrucción:

```
alter system quiesce restricted; --cuayes
```

- Si existen sesiones activas al momento de ejecutar la instrucción anterior, la sesión del usuario `sys` o `system` entrará en modo de espera hasta que la sesión se vuelva inactiva.
- La siguiente instrucción puede ser útil para determinar las sesiones activas que pudieran impedir la inactivación de la BD:

```
select bl.sid, user, osuser, type, program
from v$blocking_quiesce bl, v$session se
where bl.sid = se.sid;
```

- Para eliminar una sesión se puede emplear el siguiente comando:

```
alter system kill session 'sid,serial#';
```



3.3.1. ¿Para qué sirve inactivar una BD?

- Posibilidad de realizar tareas que pudieran fallar si otros usuarios se conectan mientras las actividades se están ejecutando. Ejemplos:

- Intentar cambiar el esquema de una tabla mientras existen transacciones en ejecución de otros usuarios.
- Operaciones de administración que implica la ejecución de varios pasos. Por ejemplo: El usuario `sys` desea reorganizar los datos de una tabla. Para ello necesita realizar los siguientes pasos:
 - 1. Exportar datos,
 - 2. Eliminar la tabla,
 - 3. Importar los datos a una nueva tabla.

Si durante la ejecución de estas actividades otros usuarios modifican los datos de la tabla que se está reorganizando, se pueden obtener resultados no esperados, e inclusive bloqueos al existir concurrencia de actualización sobre un mismo registro.

- Para restaurar el estado normal de la base de datos:

```
alter system unquiesce;
```

- Antes de la existencia de esta funcionalidad, el DBA realizaba las siguientes acciones:
 - Detener la instancia
 - Iniciarla en modo restrictivo.
- Ejecutar las 2 acciones anteriores puede ser complicado, en especial para sistemas de misión crítica que deben estar prácticamente disponibles todo el tiempo: 24x7
- Inactivar la BD no requiere detener la instancia.

Ejercicio en clase 2



Para los incisos de este ejercicio:

- Abrir una terminal a la que llamaremos T1. Iniciar sesión como usuario `sys`
 - Abrir una terminal a la que llamaremos T2. Iniciar sesión con un usuario ordinario (el creado en temas anteriores).
 - Abrir una terminal a la que llamaremos T3. Iniciar también como usuario `sys`.
- A. Abrir la BD en T1. En T2 ejecutar las sentencias necesarias para impedir que en T1 el usuario `sys` pueda inactivar la instancia.
- B. En T1 ejecutar las sentencias necesarias para inactivar la instancia. Describir lo que sucede.
- C. En T3 ejecutar la consulta vista anteriormente que permita ver quién está impidiendo la inactivación de la BD.
- D. En T3, generar una consulta en `v$instance` que muestre el valor del campo `active_state`. ¿qué significa dicho valor?
- E. En T2, ejecutar las sentencias necesarias para que sea posible inactivar la instancia en T1
- F. En T3, nuevamente, generar una consulta en `v$instance` que muestre el valor del campo `active_state`. ¿qué significa dicho valor?
- G. En T2, ¿Qué sucede si el usuario general intenta consultar los datos de alguna de sus tablas? , explicar dicho evento.
- H. Abrir una nueva terminal T5. Conectarse a SQL *Plus como usuario general. ¿Qué sucede?, explicar el evento.
- I. Abrir una nueva terminal T6. Conectarse a SQL *Plus como usuario `sys`. ¿Qué sucede?, explicar el evento.

- J. En T1, ejecutar la sentencia SQL para activar nuevamente la instancia. ¿Qué sucede en las terminales T2 y T5?

3.4. SUSPENDER UNA BASE DE DATOS.

- Cuando una base de datos se suspende a través de la sentencia `alter system suspend`, toda la actividad de lecturas y escrituras hacia los **data files** y **archivos de control** se detiene.
- Al momento de ejecutar la sentencia, todas las operaciones I/O existentes se les permite terminar, pero nuevas peticiones para realizar operaciones I/O son detenidas y se les asigna el status `queued`.
- A diferencia de inactivar una BD, las sesiones no se inactivan, pueden realizar cualquier operación siempre y cuando no requieran leer o escribir en data files el archivo de control.
 - Instrucciones `select` podrían ser exitosas si los datos están en cache.
 - Instrucciones `insert` podrían ser exitosas cuando la BD no requiere escribir a Redo Logs (esta escritura provocaría actualizar el archivo de control).
 - Instrucciones `commit` fallarán ya que requieren actualizar el archivo de control.
- La suspensión de una BD a diferencia de inactivar, no se basa en privilegios de administración.
- Cualquier usuario podría entrar a sesión siempre y cuando no se requieran hacer lecturas a los data files (diccionario de datos). Por ejemplo:
 - Un usuario con el privilegio `sysdba` podría crear una sesión ya que solo se lee el archivo de passwords para autenticar.
 - Un usuario ordinario no podría autenticar si se requiere leer sus datos del diccionario para autenticar.
 - Un usuario ordinario podría autenticar si previamente sus datos de autenticación fueron leídos y puestos en caché.

- Para suspender una base de datos se emplea

```
alter system suspend;
```

- A diferencia de inactivar, la instrucción anterior no entra en modo de espera aunque existen operaciones de lectura y escritura en ejecución.
- Para resumir o terminar la suspensión se emplea:

```
alter system resume;
```

- La siguiente sentencia se emplea para verificar si una instancia está suspendida.

```
select database_status from v$instance;
```

3.4.1. ¿Para qué sirve suspender una BD?

- Esta característica permite por ejemplo, realizar copias de los data files sin posibles interferencias o cambios realizados a estos archivos durante el proceso de copiado o respaldo.

Ejercicio en clase 3

Realizar las siguientes actividades para entender el concepto de suspensión e identificar diferencias con un proceso de inactivación.



1. Abrir una terminal T1, crear sesión con el usuario sys.
 2. Abrir una terminal T2, no autenticar aun.
-
- A. En T1 ejecutar la instrucción SQL que permita suspender la actividad I/O.
 - B. En T2 intentar autenticar con un usuario ordinario existente ¿Qué sucederá?
 - C. En T1 ejecutar la sentencia SQL que permita reanudar la actividad I/O. ¿Qué ocurre en T2?
 - D. En T2, salir de SQL*Plus. En T1 suspender nuevamente la suspensión de actividad I/O. En T2 intentar autenticar nuevamente con el mismo usuario ordinario. ¿Qué sucede?
 - E. En T1 terminar la suspensión. En T2 crear una tabla `test(id number)` e insertar un registro, hacer `commit`, salir de SQL*Plus. En T1 volver a suspender la operación I/O. En T2 iniciar sesión nuevamente con el usuario ordinario. ¿Qué sucederá al intentar consultar los datos de la tabla `test`?
 - F. En T1 terminar la suspensión, reiniciar la instancia. En T2 iniciar sesión nuevamente con el usuario ordinario. En T1 suspender nuevamente la actividad I/O. ¿Qué sucederá en T2 si el usuario intenta mostrar los datos de la tabla `test`?
 - G. En T2, reanudar las operaciones I/O.

Ejercicio
práctico 01

