

Ejercicios de arquitectura de Oracle

10 de octubre de 2019

A continuación se describen algunos aspectos de la arquitectura de una instalación particular de Oracle y se proponen una serie de ejercicios. Crea un documento que contenga una memoria de los resultados de las cuestiones propuestas. **Antes de comenzar inicia la máquina virtual Ubuntu con la instancia de Oracle como se indica en el documento del campus virtual 00Ej-Arranque-Oracle-VB.**

1. Creación básica de un usuario de la base de datos.

Puedes saltar este apartado si ya está creado el usuario `usuario1` en la instalación local de Oracle. Si no recuerdas su contraseña, puedes cambiarla desde el usuario `SYSTEM` con la sentencia:

```
ALTER USER USUARIO1 IDENTIFIED BY adbd19;
```

Si no está creado el usuario, debes crearlo. Desde una sesión de SQLDeveloper conectada con el usuario `system`, utiliza el elemento Other Users del panel izquierdo de SQLDeveloper para crear un usuario `usuario1` con la siguiente información: el usuario debe crearse en el tablespace por defecto `USERS` y el tablespace temporal `TEMP`, y con una cuota de disco de 100M. El usuario debe tener las siguientes características:

1. permiso para conectarse,
2. privilegio para alterar la sesión,
3. privilegios para crear vistas, tablas, triggers, secuencias, procedimientos y vistas materializadas en su propio *schema*.

Ejercicio 1. *Indica en la memoria del ejercicio las instrucciones SQL DCL que deben ejecutarse (las que SQLDeveloper genera automáticamente).*

2. Creación de tablas de una base de datos de usuario.

Crea una conexión en SQLDeveloper con el usuario `usuario1`. A continuación, conéctate a la base de datos con esa conexión y ejecuta los scripts `Pedidos_SCH.sql` e `insert_pedidos.sql` que están en el Campus virtual. Haz lo mismo con los scripts `students_SCH.sql` e `insert_students.sql`.

Observación: ten en cuenta que el formato de fecha por defecto de Oracle no es el utilizado en las inserciones del fichero `insert_pedidos.sql`. Se puede ver el formato por defecto con la siguiente consulta:

```
SELECT * FROM nls_session_parameters;
```

Puedes cambiar el formato de la sesión actual con:

```
ALTER SESSION SET NLS_DATE_FORMAT='DD-MM-RR';
```

3. Consulta de los tablespaces y datafiles de la base de datos.

Para este apartado **utiliza la conexión del usuario SYSTEM**. Prácticamente toda la información del gestor de la base de datos está almacenada en tablas de los *schema* del sistema. En este y otros apartados posteriores utilizaremos algunas consultas sobre vistas del diccionario de datos y de las estructuras de almacenamiento. Las vistas del diccionario de datos tienen como prefijos `USER_`, `ALL_` o `DBA_` (para la información del diccionario de datos cuyo propietario es el usuario actual, a los que puede acceder el usuario actual, o toda la información del sistema, respectivamente). La información en estas vistas no se modifica directamente, sino que se realiza a través de instrucciones DDL.

También vamos a utilizar las llamadas “vistas v\$”, que son vistas de las tablas más internas de un sistema Oracle (las “tablas x\$” del usuario SYS, que no son accesibles directamente y que principalmente utilizan datos del controlfile). En la versión 11g de Oracle hay muchas vistas v\$ (más de 500). Solo veremos unas pocas.

3.1. Consulta de los tablespaces de la base de datos.

Utiliza la siguiente consulta para obtener la lista de los tablespaces de la base de datos.

```
SELECT * FROM v$tablespace;
```

Ejercicio 2. *Indica en la memoria del ejercicio el nombre de los tablespaces de la instalación de Oracle y los que están habilitados para realizar copia de seguridad.*

3.2. Consulta de los datafiles de la base de datos.

De forma similar a la anterior, utiliza la siguiente consulta para obtener la información de los datafiles de la base de datos.

```
SELECT * FROM v$datafile;
```

Comprueba en cada datafile si está online u offline y verifica cuál es el nombre del fichero con el que se corresponde en el sistema operativo.

Ejercicio 3. *Entra en el usuario oracle de la máquina virtual y verifica los nombres y tamaños de los ficheros del SO que corresponden a los datafiles. Incluye el listado del directorio (`ls -l`) en la memoria del ejercicio.*

3.3. Investigando los segmentos y extents de un *schema*.

Ejercicio 4. *Utiliza la siguiente consulta para ver los segmentos de un tablespace:*

```
SELECT * FROM dba_segments WHERE TABLESPACE_NAME = 'USERS';
```

Localiza los segmentos de los que es propietario el usuario usuario1 e indica en la memoria del ejercicio los nombres y tipos de segmento que aparecen en la lista. Hay muchos tipos de segmento en una base de datos. Investiga las columnas de la vista dba_segments para crear una consulta que muestre todos los tipos de segmento de todos los tablespaces de la base de datos. Incluye la consulta y el resultado en la memoria del ejercicio.

La vista dba_extents contiene los extents de la base de datos. Para conocer los extents de un *schema* podemos utilizar la siguiente consulta:

```
select * from dba_extents where owner = 'USUARIO1';
```

Vamos a crear una tabla muy sencilla en el *schema* de usuario1 con una cantidad de datos suficiente para forzar la creación de extents adicionales para el segmento de la tabla. Utiliza la siguiente instrucción para crear la tabla:

```
CREATE TABLE usuario1.mitabla (c1 INTEGER);
```

Ahora vamos a dar valores a esta tabla mediante el siguiente bloque anónimo:

```
declare
  n integer;
begin
  for n IN 1..15000 loop
    insert into usuario1.mitabla values (n);
  end loop;
  commit;
end;
/
```

Este bloque inserta 15000 filas con números de 1 a 15000 en la tabla recién creada. Ahora podemos consultar los extents de esta tabla:

```
SELECT * FROM dba_extents
WHERE owner = 'USUARIO1' AND segment_name='MITABLA';
```

De hecho, con los datos de la vista dba_extents podríamos saber la posición exacta de cada extent en el datafile. La columna bytes nos da el tamaño del extent, la columna blocks contiene el número de bloques en el extent, y el identificador del bloque inicial del extent, block_id, es realmente el número de bloque del datafile (comenzando por cero).

Ejercicio 5. *Indica en la memoria cuál sería la consulta que nos proporcione la posición de los bloques del extent en el datafile.*

Ejercicio 6. *Los extents de un segmento pueden estar en distintos datafiles. ¿Cómo podemos saber en qué datafile está cada extent? Indícalo en la memoria del ejercicio.*

3.4. Creación de tablespaces y datafiles.

En una instalación típica de Oracle, el programa de instalación crea una serie de tablespaces por defecto: al menos SYSTEM y SYSAUX, pero también uno para deshacer operaciones (*undo tablespace*) y otro para datos temporales. Los dos primeros contienen el diccionario de datos (los nombres de las tablas, las columnas de cada tabla, las propiedades de cada columna, de las claves, índices, vistas, etc.). Además, es necesario almacenar los datos propiamente dichos. En instalaciones muy sencillas existe un solo tablespace para los datos de aplicaciones (que manejan los usuarios de la base de datos que no son administradores del sistema), que normalmente se denomina USERS. Sin embargo, dependiendo de las aplicaciones que utilicen Oracle en el sistema que se está administrando y las diversas necesidades que estas tengan sobre el almacenamiento de los datos (frecuencia de copias de seguridad, disponibilidad on-line, encriptación, datos de solo-lectura, etc.), puede ser conveniente tener varios tablespaces para los datos de usuario.

Ejercicio 7. *Utiliza la siguiente instrucción para crear un tablespace adicional:*

```
CREATE TABLESPACE mitbs  
DATAFILE '/u01/app/oracle/oradata/orcl/midf01.dbf' SIZE 1m  
EXTENT MANAGEMENT local autoallocate  
SEGMENT SPACE MANAGEMENT auto;
```

No entraremos en el detalle de todas las opciones para la creación de un tablespace. En este caso, asociamos al tablespace un datafile de 1MB que no es extensible, con las opciones recomendadas por defecto: gestión de extents local y gestión de segmentos automática.

Ejercicio 8. *Comprueba que las vistas vtablespaceyvdatafile se han actualizado convenientemente y se ha creado un fichero en el directorio correspondiente del sistema operativo. Adjunta a la memoria el resultado de las consultas.*

Ahora crea una tabla en este tablespace:

```
CREATE TABLE mitbl (micol VARCHAR2(100)) TABLESPACE mitbs;
```

Puedes insertar registros en esta tabla con la siguiente sentencia (o con un bloque PLSQL como el del apartado 3.3):

```
INSERT INTO mitbl (SELECT 'dábale arroz a la zorra el abad'  
FROM DUAL CONNECT BY level <= 15000);  
COMMIT;
```

Ejercicio 9. *Consulta los extents del segmento correspondiente a los datos de esta tabla (reutiliza alguna consulta anterior).*

Ejercicio 10. *Contesta a las siguientes preguntas: ¿Qué ocurre si ejecutas por segunda vez la consulta de inserción anterior? ¿Por qué?*

Puedes utilizar Enterprise Manager para ver el estado y ocupación de los tablespaces: en la pestaña "Server", accede al enlace "Tablespaces" para ver cuál puede ser el motivo del mensaje de error proporcionado si vuelves a ejecutar la consulta de inserción.

4. Otras estructuras de almacenamiento.

Verifica en el mismo directorio del sistema donde están almacenados los datafiles la existencia de los archivos que almacenan el controlfile (posiblemente múltiples copias) y los redo log files.

Ejercicio 11. *Detalla cuáles son los ficheros del sistema operativo para estas estructuras y su tamaño.*

4.1. Consulta del controlfile de la base de datos.

El controlfile contiene información crítica de la base de datos. Es uno de los ficheros más importantes de cualquier instalación de Oracle y es fundamental que no se corrompa por ningún motivo, pues en ese caso la base de datos quedaría irrecuperable. Un administrador de base de datos debe tener identificado el controlfile de la instalación y saber cómo se puede recuperar si falla el disco sobre el que se almacena.

Ejercicio 12. *Utiliza la consulta siguiente para localizar a través de los comandos de Oracle el controlfile del sistema:*

```
SELECT * FROM v$controlfile;
```

¿Cuántos resultados se muestran? Indica en la memoria del ejercicio el nombre completo de los ficheros del sistema operativo que contienen controlfiles.

Este fichero es tan importante que lo habitual es que se mantengan varias copias (al menos dos) sincronizadas continuamente pero en distintos discos. Esto se denomina multiplexación en la terminología de Oracle. De esta forma, si una de ellas se corrompe, se puede recuperar el controlfile de otra de las copias.

La instancia de la base de datos solo puede funcionar si todos los controlfiles son correctos. Si un controlfile es incorrecto, se para la ejecución de la instancia de la base de datos y no puede operar de nuevo hasta que no se restablezcan todos los controlfiles.

4.2. Consulta de los redo log files.

Los redo log files contienen la secuencia de operaciones realizadas hasta el momento en la base de datos. Son muy importantes porque en caso de fallos de disco permiten recuperar el estado de la base de datos a partir de una copia de seguridad (más o menos antigua) de los datafiles.

Como en el caso del controlfile, es importante que no se pierda la información de los redo log files. Para ello, se utilizan grupos de log files, de forma que cada cambio en los datos se registra en todos los log files del grupo activo, y los distintos grupos se van alternando de forma rotatoria en el grupo activo, de forma que se puede hacer copia de seguridad de cada grupo mientras la base de datos esta operativa y registra las operaciones en otro de los grupos. Un grupo de redo log files es activo hasta que los ficheros del grupo se llenan (son de tamaño fijo).

Ejercicio 13. *Utiliza la siguiente consulta para ver los grupos de redo log files, cuántos ficheros (miembros, en la terminología de Oracle) forman cada grupo, y qué tamaño tienen:*

```
SELECT * FROM v$log;
```

Ejercicio 14. *A continuación, utiliza la siguiente consulta para saber cuáles son los ficheros del disco que se utilizan para almacenar los redo log files:*

```
SELECT * FROM v$logfile;
```

¿Cuántos grupos de redo log files hay en el sistema? ¿cuantos ficheros tiene cada grupo?

Ejercicio 15. *Los redo log files se pueden configurar sin necesidad de parar la base de datos. Prueba a añadir un miembro a la lista de log files del grupo activo con el siguiente comando:*

```
ALTER DATABASE ADD LOGFILE MEMBER  
'/u01/app/oracle/oradata/orcl/redo01b.log' TO GROUP 1;
```

Muestra la lista de redo log para comprobar que se ha añadido.

5. Manejo de las estructuras de datos en memoria.

Vamos a ver tres consultas sencillas para descubrir información sobre las estructuras de datos en memoria.

Ejercicio 16.

1. *Primero determina los tamaños de las estructuras de datos de la SGA (System Global Area) con la siguiente consulta SQL:*

```
select * from v$sgainfo;
```

2. *Para las estructuras de datos dinámicas, determina los tamaños mínimo y máximo que pueden tener las estructuras que has visto en clase, así como el tamaño que tienen actualmente:*

```
select * from v$sga_dynamic_components;
```

3. *Por último, consulta la información de las PGA (Program Global Areas) y determina: el tamaño máximo que se puede dedicar (allocate) a PGA, el tamaño total dedicado actualmente, y el espacio que se está utilizando actualmente. Utiliza la consulta:*

```
select * from v$pgastat;
```

Incluye los resultados en la memoria del ejercicio.