



Prácticas de Administración de Bases de Datos

Grado en Ingeniería Informática

PRÁCTICA 1

Arquitectura del SGBD Oracle

SOLUCIONES

OBJETIVOS

- Saber consultar el diccionario de datos para conocer la arquitectura y el rendimiento de una BD Oracle

Ejercicios

Obteniendo información de la BD

Debéis guardar la información que se solicita en un fichero de texto para que, posteriormente, podáis verificar las respuestas.

Hay diferentes vistas que permiten visualizar los parámetros de inicialización:

- V\$SYSTEM_PARAMETER: valor actual de los valores de la instancia
- V\$SYSTEM_PARAMETER2: igual que la anterior pero con una visualización en varias líneas de los parámetros que tienen una lista de valores
- V\$PARAMETER y V\$PARAMETER2: tienen la misma estructura que las anteriores pero muestran los valores actuales de los parámetros de la sesión actual. Muestran el valor de los parámetros que han sido eventualmente modificados en la sesión mediante la sentencia ALTER SESSION
- V\$SPPARAMETER: contenido actual del fichero de parámetros de servidor activo

1. Haciendo consultas a las vistas del fichero de inicialización, responded a las siguientes preguntas:

a) ¿Se gestiona de forma automática la memoria de la instancia?. ¿Por qué?

Sí. Porque el parámetro memory_target tiene asignado un valor

b) ¿Cuánta memoria asigna Oracle a la SGA + PGA?

select * from V\$SPPARAMETER where name = 'memory_target';

c) ¿Hay un valor máximo para el tamaño de la memoria de la instancia? ¿Ha sido asignado explícitamente?

Sí. El parámetro memory_max_target tiene el mismo valor que memory_target. No

d) ¿Se asegura un valor mínimo para la SGA?

No. El parámetro sga_target tiene valor 0

e) ¿Cuántos parámetros del fichero de inicialización son explícitos?

```
select count(*)  
from V$SPPARAMETER  
where value IS NOT NULL;
```

f) ¿Cuál es el tamaño del bloque en la BD Onuba?

8M

```
select value  
from V$SPPARAMETER  
where name = 'db_block_size';
```

2. Asignar un valor de 256M a la memoria compartida pero mantener dicho valor únicamente en memoria (no guardar en el fichero spfile)

```
ALTER SYSTEM SET shared_pool_size = 256M SCOPE = MEMORY;
```

3. Comprobar que el cambio se ha realizado únicamente en la sesión actual. Volver a asignar el mismo valor pero almacenándolo en el fichero de inicialización

```
select * from V$SPPARAMETER where name='shared_pool_size'  
  
select * from V$PARAMETER where name='shared_pool_size'  
  
ALTER SYSTEM SET shared_pool_size = 256M SCOPE = SPFILE;
```

4. Reiniciar la base de datos y comprobar que arranca con el nuevo valor para la memoria compartida

5. ¿Cómo obtenemos los parámetros que son modificables a nivel de sesión con alter sesión?

```
select NAME, ISSES_MODIFIABLE from V$PARAMETER WHERE  
ISSES_MODIFIABLE='TRUE'
```

6. Editar el archivo de inicialización de la base de datos (estará ubicado en c:\app\usuario\admin\\${ORACLE_SID}\pfile y se llamará **init.ora** seguido de una serie de números). Observar algunos de los parámetros de inicialización e intentar entender su significado

7. Añadir una entrada en el fichero init asignando un tamaño máximo (algo mayor que el actual) a la memoria de la instancia. Comprobar que el cambio se ha llevado a cabo correctamente (Nota, es posible que alguna operación deba hacerla bajo Sql Plus con el usuario SYS (o SYSDBA))

1. Añadimos al fichero de inicialización de texto la línea
memory_max_target = xxM
2. Creamos el nuevo fichero binario a partir del fichero de texto con otro nombre

```
CREATE SPFILE =
  'c:\app\uhu\product\11.2.0\dbhome_1\database\spfileonuba.ora.new
FROM PFILE =
  'c:\app\uhu\admin\onuba\pfile\init.ora.1520141922143'
```

3. Renombramos el fichero que hemos creado con el nombre de "spfileonuba.ora" y mantenemos también el original
4. Reiniciamos la bd
5. Comprobamos select * from v\$parameter where name like 'mem%'

Hay varias vistas dinámicas de la BD que nos dan información sobre el tamaño y la estructura de la SGA:

- V\$SGAINFO
- V\$SGA_DYNAMIC_COMPONENTS
- V\$SGA_TARGET_ADVICE
- V\$SGA
- V\$SGASTAT
- V\$MEMORY_DYNAMIC_COMPONENTS
- V\$SGA_DYNAMIC_FREE_MEMORY

8. ¿Qué tamaño tiene la SGA fija? ¿Qué tamaño tiene el buffer del registro de deshacer?

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

9. ¿Qué tamaño se le ha asignado a la SGA y a la PGA de forma automática? ¿Por qué el tamaño de la shared_pool es distinto al que se especificó en el ejercicio 3 (PISTA: debéis mostrar las columnas current_size y user_specified_size)?

```
select      component,      current_size/(1024*1024)      MBCurrent,
user_specified_size/(1024*1024) MbUser
from v$memory_dynamic_components;
```

10. Consultar información sobre la base de datos y la instancia. Indique las instrucciones necesarias para obtener la siguiente información ¿En qué fecha fue creada la base de datos ONUBA?. ¿Cómo se llama la máquina en la que corre la base de datos?

```
select created from v$database;
select host_name from v$instance
```

11. Localizar el proceso “servidor” asociado a la sesión que estáis utilizando (pista: estas accediendo con el programa Sql Developer). ¿Es un servidor dedicado o compartido? ¿Cómo se llama el usuario del sistema operativo y el ordenador desde el que se está realizando la conexión?

```
select  a.SERVER,    a.username  dbuser, a.OSUSER,    a.PROCESS,
a.machine,  a.terminal,  a.program from v$session a where
a.username='SYSTEM' or a.username='SYS'
```

12. Ver las sentencias SQL que guarda la *SharedPool* (vista **v\$sqlarea**)

En la vista V\$SQLAREA podemos ver el contenido del “área SQL” de la SharedPool. Además podemos obtener información útil para el ajuste de cada una de las sentencias SQL (Shared Pool consumida, nº de veces que se ha ejecutado, nº de veces que se ha salido de la caché, lecturas físicas, tiempo de CPU, tiempo total incluyendo compilación, etc.).

Si consultamos desde el usuario SYS o SYSTEM esta vista obtendremos un número elevado de consultas internas que se hacen en el sistema. Para poder hacer bien el ejercicio, haremos lo siguiente:

- Desbloquear la cuenta del usuario scott (clave: tiger)
Ejecute las siguientes instrucciones
alter user scott account unlock;
alter profile default limit password_life_time unlimited;
alter profile DEFAULT limit FAILED_LOGIN_ATTEMPTS UNLIMITED;
alter user scott identified by tiger;
- Desde el usuario scott, crear una tabla cualquiera e insertar algunas filas
- También desde el usuario scott, realizar 2 o 3 consultas SELECT
- Desde el usuario SYS/SYSTEM, consultar el área SQL de las consultas realizadas por el usuario scott
- El campo que contiene el nombre del usuario que ha realizado la consulta es **parsing_schema_name**
- Visualizar las consultas del usuario SCOTT que se encuentran en la SharedPool.

```
select distinct PARSING_SCHEMA_NAME, SQL_TEXT,
PERSISTENT_MEM, EXECUTIONS, LOADS, DISK_READS,
CPU_TIME, ELAPSED_TIME, parsing_user_id
from v$sqlarea
where PARSING_SCHEMA_NAME = 'SCOTT';
```

13. Comprobar el funcionamiento de la caché de redolog, como protectora del contenido de la caché de datos. Para ello, se creará una transacción y se provocará una caída de la BD. Comprobar que, al arrancarla de nuevo, se mantiene la integridad de la misma.

- Crear una tabla cualquiera en el espacio de tablas del usuario SCOTT
- Insertar una fila sin hacer commit y forzar la caída de la BD
- Arrancar de nuevo la BD y comprobar que la fila insertada no está
- Repetir la inserción de la fila y hacer commit. Forzar de nuevo la caída de la BD
- Arrancar la BD y comprobar que ahora la fila si está insertada

14. Con este ejercicio vamos a comprobar el funcionamiento de la caché de datos.

- Activar la medición de tiempos mediante SET TIMING ON
- Lanzar la consulta `SELECT COUNT(*) FROM DBA_SOURCE`
- Volver a lanzar la misma consulta
- Comprobar que la segunda ejecución tarda mucho menos, ya que los datos ya se cargaron en la caché de datos al lanzarla la primera vez y, por tanto, se accede directamente a memoria y no a disco