

TEMA 04  
Ejercicio práctico 01  
Administración de las estructuras de Memoria

NOMBRE:

GRUPO:

FECHA DE ENTREGA:

CALIFICACION:

1.1. OBJETIVO

Conocer y familiarizarse con las principales vistas y parámetros de la base de datos que muestran o contienen información relevante acerca del uso de las diferentes áreas de la SGA.

1.2. EXPLORACIÓN DE LAS ÁREAS DE MEMORIA

- Crear un script `s-00-crea-usuario.sql` El script deberá crear un usuario `<nombre>0401` en caso de no existir. Otorgarle los privilegios necesarios para que este pueda crear tablas, sesiones, cuota ilimitada.
- Todas las tablas que serán creadas en este ejercicio práctico deberán ser asignadas al usuario `0401` creado anteriormente.
- Crear un script `s-01-sga-components.sql` Deberá realizar lo siguiente:
  - A. Crear una tabla llamada `t01_sga_components` y poblarla con la siguiente información

memory_target_param_mb	Fixed_size_mb	Variable_size_mb	Database_buffers_mb	Redo_buffers_mb	Total_sga_mb
Cantidad de memoria RAM en MB truncada a 2 decimales que se obtiene al consultar el valor del parámetro <code>memory_target</code>	Estas columnas contienen la memoria RAM en MB truncada a 2 decimales asignada a cada uno de los componentes de la SGA. Expresarlas en MB. Tip: emplear <code>v\$SGA</code> .				Calcular el total de RAM en MB truncada a 2 decimales con base a la distribución de los componentes de <code>v\$SGA</code> .

La consulta anterior contiene un resumen de los principales componentes de la SGA: Área de memoria fija (Fixed size), área de memoria variable, memoria asignada al db buffer caché, memoria asignada a los redo buffers.

- B. Crear una tabla `t02_sga_dynamic_components` con la siguiente estructura:

```
create table <nombre>0401.t02_sga_dynamic_components(  
  component_name varchar2(64),  
  current_size_mb number(10,2),  
  operation count number(10,0),  
  last_operation_type varchar2(13),  
  last_operation time date  
);
```

Empleando la vista `v$sga_dynamic_components`, incluir en la tabla el nombre del componente de la SGA, su tamaño actual en MB truncado a 2 decimales, el número de operaciones realizadas sobre cada componente, el nombre del tipo y la fecha de la última operación realizada.

- C. Crear una tabla `t03_sga_max_dynamic_component` con la siguiente estructura:

```
create table <nombre>0401.t03_sga_max_dynamic_component(  
  component_name varchar2(64),  
  current_size_mb number(10,2),  
);
```

Incluir en la tabla el nombre del componente y la cantidad de memoria asignada en MB. Incluir únicamente al componente con la mayor cantidad de memoria asignada. Tip: consultar `v$sga_dynamic_components` y extraer al componente que tenga la mayor cantidad de memoria asignada (atributo `current_size`).

- D. Crear una tabla `t04_sga_min_dynamic_component` con la siguiente estructura:

```
create table <nombre>0401.t04_sga_min_dynamic_component(  
  component_name varchar2(64),  
  current_size_mb number(10,2),  
);
```

Incluir en la tabla el nombre del componente y la cantidad de memoria asignada en MB. Incluir únicamente al componente con la menor cantidad de memoria asignada mayor a cero. Mismo Tip del ejercicio anterior.

E. Crear una tabla `t05_sga_memory_info` con la siguiente estructura:

```
create table <nombre>0401.t05_sga_memory_info(
  name varchar2(64),
  current_size_mb number(10,2),
);
```

Considerando la vista `v$sgainfo` incluir en la tabla 2 registros con nombres 'Maximum SGA Size' y 'Free SGA Memory Available'. En el campo `current_size_mb` incluir los valores de memoria en MB truncada a 2 decimales que indican la cantidad máxima de memoria que puede tener la SGA y la cantidad de memoria de la SGA que actualmente está disponible.

F. Crear una tabla `t06_sga_resizeable_components` con la siguiente estructura

```
create table <nombre>0401.t06_sga_resizeable_components(
  name varchar2(64)
);
```

Considerando la vista `v$sgainfo` incluir en la tabla los nombres de los componentes cuya cantidad de memoria puede ser ajustado.

- Hacer `commit` de los datos.

### 1.3. OPERACIONES TIPO RESIZE PARA LAS ÁREAS DE MEMORIA DE LA SGA.

Las vistas `v$sga_current_resize_ops` y `v$sga_resize_ops` muestran información de los cambios o reasignaciones en las diferentes áreas de memoria de la SGA que son requeridas para procesar cierta carga de trabajo. La primera vista muestra las operaciones de ajuste de memoria que se realizan en tiempo real y la segunda muestra un histórico de las últimas 800 operaciones de reasignación. Para ilustrar estas operaciones, crear un script `s-02-sga-resize-ops.sql` que será ejecutado por SYS con las siguientes instrucciones:

A. Reiniciar la instancia.

B. Crear una tabla `t07_sga_resize_ops` que contenga las columnas mostradas en la figura tomadas de las 2 vistas anteriores

COMPONENT	OPER_TYPE	PARAMETER	INITIAL_SIZE_MB	TARGET_SIZE_MB	FINAL_SIZE_MB	INCREMENT_MB	STATUS	START_TIME	END_TIME
1 DEFAULT buffer cache	INITIALIZING	db_cache_size	176	176	176		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16
2 DEFAULT buffer cache	STATIC	db_cache_size	0	176	176		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16
3 Shared IO Pool	STATIC	shared_io_pool_size	0	16	16		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16
4 java pool	STATIC	java_pool_size	0	4	4		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16
5 large pool	STATIC	large_pool_size	0	24	24		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16
6 shared pool	STATIC	shared_pool_size	0	264	264		0 COMPLETE	03-JUL-21 01:51:16	03-JUL-21 01:51:16

- Ordenar los registros por el nombre del componente y posteriormente por la fecha fin.
- Notar que algunas columnas se les agregó el sufijo “\_MB” ya que sus valores deben ser mostrados en MB.
- La consulta deberá ser obtenida justo después de haber iniciado la instancia para tomarla como referencia. Notar en la imagen que el default buffer cache solo cuenta con un ajuste inicial de 0 a 176 MB el cual se realizó al iniciar la instancia.

C. Crear una tabla `t08_random_data` con la siguiente estructura.

```
create table <nombre>0401.t08_random_data(
  random_string varchar2(1024)
);
```

- Esta tabla servirá para realizar la carga de aproximadamente 300 MB de datos (300,000 registros). Al realizar este proceso de carga seguramente las áreas de memoria mostradas anteriormente sufrirán ajustes para poder realizar el procesamiento de esta cantidad de datos.
- Para realizar la carga de datos, generar un bloque de código PL/SQL que genere cadenas aleatorias de 1024 bytes (1K), realizar las iteraciones necesarias para insertar alrededor de 300MB. Ejemplo:

```
declare
  v_rows number;
begin
  v_rows := 1000*300;
  for v_index in 1 .. v_rows loop
    insert into <nombre>0401.t08_random_data(random_string)
      values(dbms_random.string('P',1024));
  end loop;
end;
/
commit;
```

- No olvidar hacer `commit` para confirmar las inserciones.

- D. Realizar una consulta a los datos insertados para provocar una carga de bloques en el DB Buffer cache. Como consecuencia, la instancia podría realizar algunas operaciones de reajuste de memoria en especial en este buffer. Bastará con contar los registros de la tabla:

```
Prompt realizar una consulta para provocar carga de datos en BD Buffer cache
set timing on
select count(*) from jorge0401.t08_random_data;
set timing off
```

- Notar el uso de `set timing on/off`. Estas instrucciones permiten mostrar el tiempo que se tarda la instancia en realizar el conteo de registros. Debido a que no se cuentan con índices, la instancia deberá hacer un escaneo completo de toda la tabla.
- E. Repetir nuevamente la consulta para mostrar las operaciones de reajuste de memoria, pero ahora guardar la salida en una tabla `t09_sga_resize_ops`.
- F. Revisar el contenido de las tablas `t07_sga_resize_ops` y `t09_sga_resize_ops`. **C3. Incluir en el reporte** una descripción de los ajustes de memoria que sufrieron cada una de las áreas mostradas en los resultados. ¿Cuánto tiempo se tardó la base de datos en realizar el conteo de registros? Tip: En caso de no observar operaciones de reajuste de memoria revisar la cantidad de memoria asignada al DB buffer cache (columna `final_size_mb`). Si su valor es cercano a los 250M – 300M significaría que la instancia ha inicializado este buffer con la cantidad de memoria suficiente para procesar la consulta anterior. Si esta condición ocurre, agregar la explicación correspondiente.

#### 1.4. VALIDADOR.

- Copiar todos los scripts de validación en la misma carpeta donde se ubican los scripts del ejercicio. Ejecutar el validador empleando el usuario ordinario del sistema operativo.

```
sqlplus /nolog
start s-03-validador-main.sql
```

#### 1.5. CONTENIDO DE LA ENTREGA.

- C1. Código empleado para crear y poblar la tabla `t01_sga_components`
- C2. Respuesta del inciso E del punto 1.2
- C3. Descripción de los ajustes de memoria.
- C4. Mostrar el contenido de cada una de las tablas `t0*` creadas en este ejercicio excepto la tabla de cadenas aleatorias. Para el caso de la tabla `t02_sga_dynamic_components` mostrar su contenido ordenado con respecto al tamaño de la memoria asignada de forma descendente. La fecha deberá mostrarse hasta nivel de segundos. Analizar la salida y solo observar los componentes que NO tienen memoria asignada.
- C5. Resultado del validador.
- Elementos generales indicados en la rúbrica general de ejercicios prácticos (datos generales, conclusiones y comentarios).
- Entrega individual