

TEMA 04  
Ejercicio práctico 04  
Administración de las estructuras de Memoria

NOMBRE:

GRUPO:

FECHA DE ENTREGA:

CALIFICACION:

### 1.1. OBJETIVO

Comprender y poner en práctica el cálculo y las configuraciones de los parámetros de una base de datos para modificar los modos de administración de la memoria: automática, automática compartida y manual.

### 1.2. CREACIÓN DE USUARIO.

- Crear un script `s-00-crea-usuario.sql` encargado de generar el usuario del ejercicio `<nombre>0404` con los privilegios necesarios para realizar las actividades del documento.

### 1.3. OBJETOS INICIALES

Crear un script `s-01-objetos-iniciales.sql` El script deberá contener los siguientes objetos.

A. Crear una tabla `<nombre>0404.t01_memory_areas` con la siguiente información:

- Número de renglón
  - Fecha de inserción (fecha actual)
  - Memoria asignada al redo log buffer
  - Memoria asignada al db buffer cache
  - Memoria asignada al shared pool
  - Memoria asignada al large pool size
  - Memoria asignada al java pool size
  - Memoria total asignada a la SGA
  - Memoria libre o disponible para aumentar el tamaño de los componentes de la SGA. Tip: emplear `v$sga_dynamic_free_memory`
  - Cantidad de memoria máxima que ha tenido la PGA asignada hasta el momento. Tip: emplear `v$pgastat`
  - Cantidad de memoria disponible que puede ser asignada la PGA (aggregate PGA target parameter). Tip: emplear `v$pgastat`. Este valor es calculado por la instancia, mismo que correspondería con el valor del parámetro `pga_aggregate_target` en caso de tener una configuración de memoria manual.
- Ojo: no confundir con el valor actual que tiene asignado el parámetro `pga_aggregate_target` el cual es cero para una administración automática. El valor de este atributo es calculado como parte de las estadísticas del uso de la PGA en la vista `v$pgastat`. Su valor es el que podría usarse en caso de cambiar el modo de administración automática a Automatic Shared Memory Management.

Todas las cantidades de memoria deberán estar expresadas en MB truncadas a 2 decimales. Emplear los nombres de columnas mostrados en la figura. La tabla deberá contener un primer registro con `Id = 1`.

A nivel general, la tabla muestra la distribución de memoria en los principales componentes de la SGA y la PGA.

ID	SAMPLE_DATE	REDO_BUFFER_SIZE	BUFFER_CACHE_SIZE	SHARED_POOL_SIZE	LARGE_POOL_SIZE	JAVA_POOL_SIZE	SGA_SIZE	SGA_FREE_MEMORY	MAX_PGA_ALLOCATED	AGGREGATE_PGA_TARGET_PARAM
1	20-12-2020 21:49:09	7.73	184	288	20	4	767.99	256	166.39	256

B. Crear una tabla `nombre<0404>.t02_memory_param_values` que muestre los valores actuales de los parámetros asociados con la distribución de la memoria en la SGA y PGA.

- Número de renglón
- Fecha de inserción (fecha actual)
- Valor del parámetro que indica el total de memoria disponible (memory target)
- Valor de los parameters `sga_target`, `pga_aggregate_target`, `shared_pool_size`, `large_pool_size`, `java_pool_size`, `db_cache_size`

- Todas las cantidades de memoria deberán estar expresadas en MB truncadas a 2 decimales. Emplear los nombres de columnas mostrados en la figura. La tabla deberá contener un primer registro con `Id = 1`.

ID	SAMPLE_DATE	MEMORY_TARGET	SGA_TARGET	PGA_AGGREGATE_TARGET	SHARED_POOL_SIZE	LARGE_POOL_SIZE	JAVA_POOL_SIZE	DB_CACHE_SIZE
1	20-12-2020 14:06:27	768	0	0	0	0	0	0

- C. Crear una tabla nombre<0404>.t03\_memory\_param\_info. La tabla deberá contener un renglón para cada uno de los siguientes parámetros con la siguiente información:
- Número del parámetro
  - Nombre del parámetro
  - Valor actual.
  - Valor por default (valor asignado en caso de no haber sido configurado de forma explícita).
  - Bandera que indica si el valor del parámetro es el valor por default.
  - Bandera que indica si el valor del parámetro puede ser modificado a nivel de sesión.
  - Valores que indiquen si el valor del parámetro puede ser modificado a través de la instrucción `alter system`.

Lista de parámetros:

'memory\_target', 'memory\_max\_target', 'sga\_target', 'sga\_max\_size', 'shared\_pool\_size', 'large\_pool\_size', 'java\_pool\_size', 'db\_cache\_size', 'streams\_pool\_size', 'log\_buffer'

Ejemplo:

NUM	NAME	VALUE	DEFAULT_VALUE	ISDEFAULT	ISSES_MODIFIABLE	ISSYS_MODIFIABLE
142	sga_max_size	805306368	0	TRUE	FALSE	FALSE
248	shared_pool_size	0	134217728	TRUE	FALSE	IMMEDIATE
250	large_pool_size	0	134217728	TRUE	FALSE	IMMEDIATE
252	java_pool_size	0	25165824	TRUE	FALSE	IMMEDIATE
254	streams_pool_size	0	0	TRUE	FALSE	IMMEDIATE
1436	sga_target	0	0	TRUE	FALSE	IMMEDIATE
1438	memory_target	805306368	0	FALSE	FALSE	IMMEDIATE
1439	memory_max_target	805306368	0	TRUE	FALSE	FALSE
1506	db_cache_size	0	0	TRUE	FALSE	IMMEDIATE
1867	log_buffer	7766016	0	TRUE	FALSE	FALSE

- D. Crear un procedimiento almacenado llamado `usuario<0404>.spv_consulta_random_data`. Este procedimiento realiza algunas operaciones para simular el uso máximo de la PGA. Suponer que se desea ejecutar un algoritmo simple que calcule el total de ocurrencias de un carácter dentro de la columna `r_varchar` de la tabla `<usuario>0204.t01_random_data` empleada en ejercicios anteriores. Para ello, el procedimiento deberá realizar las siguientes acciones:
- Definir una variable cuyo valor sea el carácter a buscar. El carácter a buscar es a libre elección (cualquier carácter imprimible excepto acentos).
  - A través de un cursor obtener todos los registros de la tabla `<usuario>0204.t01_random_data` ordenados por su `id` (no olvidar otorgar permisos de lectura).
  - Para cada registro se deberá leer el contenido de la columna `r_varchar` y se deberá analizar para contar el número de ocurrencias del carácter seleccionado. Es posible emplear funciones para manejo de cadenas para realizar este análisis. El número de ocurrencias se deberá acumularse en una variable; es decir, la variable contiene la suma total de ocurrencias del carácter en todos los registros.
  - Al final, el procedimiento deberá mostrar el valor del carácter seleccionado, el total de ocurrencias, y el número total de registros procesados. Tip: No imprimir mensajes a consola dentro del ciclo que itera al cursor ya que por la cantidad de registros que contiene la tabla pueden obtenerse problemas con la capacidad del buffer empleado para enviar los mensajes a consola una vez que el procedimiento se ejecuta.
  - Ejecutar el script. Solo crear el procedimiento, no ejecutarlo aún.
- E. Crear un script `s-02-obten-datos-reinicio.sql`. El script deberá realizar las siguientes acciones:
- Detener la instancia.
  - Iniciar la instancia.
  - Generar un nuevo registro en `<nombre>0404.t01_memory_areas` con `id = 2`.
  - Hacer `commit`.

Ejecutar el script. El registro con `ID = 2` permitirá visualizar el uso de la memoria mínimo justo después de reiniciar. Sus valores permitirán comparar la cantidad de memoria que se requiere para simular un proceso considerado como "proceso de carga máxima" el cual se describe a continuación.

#### 1.4. SIMULACIÓN CARGA MÁXIMA.

Suponer que la carga máxima que recibirá la BD es la ejecución del procedimiento almacenado creado anteriormente. Para realizar esta simulación realizar las siguientes acciones:

- A. Crear un script `s-03-carga-maxima.sql`.
  - Crear una sesión con el usuario `sys`
  - Habilitar la variable `timing` para mostrar el tiempo que tarda el procedimiento en ejecutarse.
  - Ejecutar el procedimiento.
  - Generar una nueva sentencia `insert` en la tabla `t01_memory_areas` con `id = 3`

```
set timing on
exec spv_consulta_random_data
set timing off
insert ...
commit;
```

### **C1. Incluir en el reporte:**

- El código del script `s-01-objetos-iniciales.sql`
- La evidencia que muestra la ejecución del procedimiento y su salida.
- Responder las siguientes preguntas:
  - A. ¿Cuánto tiempo tardó en ejecutar el procedimiento que simula una carga máxima?
  - B. ¿Por qué razón los valores de los parámetros en `t02_memory_param_values` es cero, y en la tabla `t01_memory_areas`, la memoria asignada es mayor a cero?
  - C. Considerar los datos de la tabla `t01_memory_areas` con `id = 2`. Calcular la cantidad de memoria de la SGA que está actualmente en uso ( es decir, no considerar la memoria libre de la SGA).
  - D. Considerar el valor de la última columna de la tabla `t01_memory_areas` con `id = 2`. Este valor fue calculado por la instancia y corresponde al valor recomendado de memoria para ser asignado a la PGA en una BD tradicional. ¿Qué porcentaje del total de la SGA le corresponde a este valor?
  - E. Describir las diferencias encontradas en los 3 registros de `t01_memory_areas`. Si se deseara realizar una configuración manual de la memoria máxima para ser asignada a la PGA, con base a los valores de estos 3 registros, ¿Qué valor tendría esta configuración manual?

## **1.5. CONFIGURACIÓN AUTOMATIC SHARED MEMORY MANAGEMENT**

Para habilitar esta configuración se requiere configurar 2 parámetros `sga_target` y `pga_aggregate_target`. Para determinar el valor de estos 2 parámetros se realizará el siguiente procedimiento suponiendo la carga máxima realizada en la sección anterior.

### **1.5.1. Calcular sga\_target**

Existe la fórmula vista en clase que permite calcular el valor del parámetro `sga_target` para habilitar una configuración de memoria "Automatic Shared Memory Management".

```
select (
  (select sum(value) from v$sga) -
  (select current size from v$sga dynamic free memory)
) "sga_target"
from dual;
```

Básicamente la fórmula obtiene la cantidad total de memoria asignada a la SGA sin considerar la memoria libre. Emplear esta fórmula para realizar la configuración del parámetro `sga_target`. Redondear el valor obtenido al siguiente número entero. Notar que la memoria libre o disponible representaría la cantidad máxima que podría asignarse a la PGA.

### **1.5.2. Calcular pga\_aggregate\_target**

El ejercicio de la sección anterior permitió simular la ejecución de un proceso que hace uso de la PGA. En la tabla `t01_memory_areas` contiene varios registros que permiten mostrar la cantidad máxima de memoria que ha usado la PGA. Analizar esta tabla y determinar el valor que será empleado para configurar `pga_aggregate_target`.

- A. Una vez que se haya determinado el valor de estos 2 parámetros, aplicar las configuraciones correspondientes para que la administración de la memoria sea "Automatic shared memory management". Especificar estas configuraciones en un script llamado `s-04-automatic-shared.sql`. Los cambios deben aplicarse a nivel de instancia, no permanentes. Tip: La suma de estos 2 valores debe ser aproximadamente igual al valor del parámetro `memory_target` el cual es de 768M configurado en ejercicios anteriores. Agregar los comentarios correspondientes al

script para justificar los valores calculados. No es necesario calcular estos valores de forma dinámica, solo se requiere documentar y justificar correctamente los cálculos realizados para obtener dichos valores.

- B. Agregar al script una instrucción `insert` para generar un nuevo registro en `t02_memory_param_values` con `id = 2`. Antes de ejecutar la instrucción `insert`, esperar 5 segundos `exec dbms_session.sleep(5);`

### **C2. Incluir en el reporte:**

- El contenido del script `s-04-automatic-shared.sql`
  - Responder las siguientes preguntas.
- A. ¿Cuál fue el valor obtenido al ejecutar la fórmula anterior?
- B. ¿Para qué se utilizó el valor obtenido al evaluar la fórmula?
- C. ¿Cuánta memoria libre tiene la SGA?
- D. Calcular la diferencia entre la memoria libre de la SGA y el valor máximo de memoria PGA que se obtuvo al ejecutar el script `s-03-carga-maxima.sql`. ¿Qué interpretación o significado representa esta diferencia?

## **1.6. CONFIGURACIÓN MANUAL SHARED MEMORY MANAGEMENT**

Con base a los valores de la tabla `t01_memory_areas` con `id = 1`, realizar las configuraciones necesarias para que la administración de la memoria ahora sea "Manual shared memory management".

- A. Especificar las configuraciones en un script llamado `s-05-manual-shared.sql`. Los cambios deben aplicarse a nivel instancia, no permanentes.
- B. Agregar al script una instrucción `insert` para generar un nuevo registro en `t02_memory_param_values` con `id = 3`. Antes de ejecutar la instrucción `insert`, esperar 5 segundos `exec dbms_session.sleep(5);`

### **C3. Incluir en el reporte** el contenido del script `s-05-manual-shared.sql`

## **1.7. CONFIGURACIÓN AUTOMÁTICA.**

- A. Considerando los valores de la tabla `t01_memory_areas` con `id = 3` generar las configuraciones necesarias para regresar a una configuración automática. Realizar estas configuraciones en un script llamado `s-06-automatic-memory.sql`. Los cambios deben aplicarse a nivel de instancia, no permanentes.
- B. Agregar al script una instrucción `insert` para generar un nuevo registro en `t02_memory_param_values` con `id = 4`. Antes de ejecutar la instrucción `insert`, esperar 5 segundos `exec dbms_session.sleep(5);`

### **C4. Incluir en el reporte**

- El contenido del script `s-06-automatic-memory.sql`
- El contenido de las 3 tablas creadas

## **1.8. 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-07-validador-main.sql
```

## **1.9. CONTENIDO DE LA ENTREGA.**

### **C1.**

- El código del script `s-01-objetos-iniciales.sql`
- La evidencia que muestra la ejecución del procedimiento
- Respuestas de las preguntas A,B,C y D

### **C2.**

- El contenido del script `s-04-automatic-shared.sql`
- Respuestas de las preguntas A,B,C y D

C3.

- Contenido del script `s-05-manual-shared.sql`

C4.

- El contenido del script `s-06-automatic-memory.sql`
- El contenido de las 3 tablas creadas