

TEMA 04
Ejercicio práctico 03
Administración de las estructuras de Memoria

NOMBRE:

GRUPO:

FECHA DE ENTREGA:

CALIFICACION:

1.1. OBJETIVO

Practicar y explorar el uso de las vistas del diccionario de datos asociadas con el uso, administración y monitoreo de las áreas de memoria que componen a la PGA

1.2. CREACIÓN DE USUARIO

- Crear un script `s-00-crea-usuario.sql` encargado de generar el usuario del ejercicio `<nombre>0403` con los privilegios necesarios para realizar las actividades del documento.

1.3. REDO LOG BUFFER

Crear un script `s-01-redo-log-buffer.sql` que realice las siguientes acciones:

A. Crear una tabla `<nombre>0403.t01_redo_log_buffer` que muestre la siguiente información:

- Cantidad de memoria en MB asignada a este buffer a partir del valor del parámetro correspondiente
- Cantidad de memoria en MB asignada a este buffer obtenido de las vistas del diccionario de datos (no emplear las vistas `v$parameter*`, `v$sppparameter*` y `v$system_parameter`, emplearlas para el punto anterior).
- Bandera que indica si la memoria asignada puede cambiar de tamaño.
- Emplear los nombres de columna indicados en la siguiente imagen.

REDO_BUFFER_SIZE_PARAM_MB	REDO_BUFFER_SGA_INFO_MB	RESIZEABLE
7.40625	7.73828125	No

C1. Incluir en el reporte la consulta y la salida obtenida.

1.4. SHARED POOL.

Crear un script `s-02-shared-pool.sql` que realice las siguientes acciones:

A. Crear una tabla `<nombre>0403.t02_shared_pool.sql` que muestre la siguiente información

- Cantidad de memoria en MB asignada al pool a partir del valor del parámetro correspondiente. ¿Qué sucede si este parámetro tiene el valor 0?
 - Cantidad de memoria en MB asignada a este pool obtenido de las vistas del diccionario de datos (no emplear las vistas `v$parameter*`, `v$sppparameter*` y `v$system_parameter`, emplearlas para el punto anterior).
 - Bandera que indica si la memoria asignada puede cambiar de tamaño.
 - Número total de componentes o procesos que están haciendo o que tienen memoria asignada del shared pool. Tip: revisar `v$sgastat`.
 - Cantidad de memoria libre del shared pool.
- B. Emplear los nombres de columna indicados en la siguiente imagen.

SHARED_POOL_PARAM_MB	SHARED_POOL_SGA_INFO_MB	RESIZEABLE	SHARED_POOL_COMPONENT_TOTAL	SHARED_POOL_FREE_MEMORY
0	264	Yes	1369	26.99935150146484375

1.5. USO Y ESTADÍSTICAS DE LA PGA

Crear un script `s-03-pga-stats.sql` que realice las siguientes acciones:

A. Crear una tabla `<nombre>0403.t03_pga_stats` que muestre la siguiente información

- La cantidad total de memoria máxima que se le ha asignado a la PGA desde el inicio de la instancia.

- La cantidad de memoria que la instancia ha calculado para el parámetro `pga_aggregate_target`. ¿Qué significaría si el valor de esta métrica es mayor a la memoria máxima calculada en el punto anterior?
- La cantidad de memoria total que tiene asignada actualmente la PGA
- La cantidad de memoria en MB que actualmente está en uso en la PGA. ¿Qué significado tendrá realizar la resta de la columna anterior con esta.
- Cantidad de memoria libre en MB que contiene la PGA y que puede ser liberada.
- Número de procesos activos que están siendo atendidos por la PGA hasta el momento.
- Cantidad de memoria en MB empleada para las llamadas "Work áreas" en modo automático.

B. Emplear los nombres de columna indicados en la siguiente imagen.

MAX_PGA_MB	PGA_TARGET_PARAMETER_MB	PGA_TOTAL_ACTUAL_MB	PGA_IN_USE_ACTUAL_MB	PGA_FREE_MEMORY_MB	PGA_PROCESS_COUNT	PGA_IN_USE_WORKAREAS_AUTO
237.5830078125	336	237.5830078125	129.7236328125	70.375	64	0

C2. Incluir en el reporte la consulta y la salida obtenida.

1.6. USO DE LA PGA EN V\$PROCESS

En este ejercicio se realizan algunas consultas que permiten visualizar la cantidad de memoria PGA que utilizan los procesos que se crean durante la existencia de la instancia. A nivel general se consideran 2 tipos de procesos:

- Procesos de background (`bkg-process`): Aquellos procesos cuyo valor de la columna `pname` de la vista `v$process` es no nula.
- Otros procesos que son creados para atender otras peticiones (`other-process`), por ejemplo, usuarios conectados a la instancia. Para estos casos el valor de la columna `pname` es `null`.

Crear un script `s-04-pga-process.sql` que realice las siguientes acciones:

A. Crear una tabla `<nombre>0403.t04_pga_process` que muestre la siguiente información:

Registro 1:

- Columna cuyo nombre sea `process_type` y cuyo valor sea la cadena `'bkg-process'`.
- Número total de procesos de background que actualmente existen (número de registros que obtiene la consulta, usar la función `count`).
- La cantidad total de memoria en MB truncada a 2 decimales asignada a los procesos de background.
- La cantidad total de memoria en MB truncada a 2 decimales que actualmente está en uso para todos los procesos de background.

La consulta deberá considerar únicamente a los procesos de background, es decir, la columna `p_name` es no nula.

Registro 2:

- Columna cuyo nombre sea `process_type` y cuyo valor sea la cadena `'other-process'`.
- Número total de procesos que no son de background (número de registros que obtiene la consulta, usar la función `count`).
- La cantidad total de memoria en MB truncada a 2 decimales asignada a los procesos que no son de background
- La cantidad total de memoria en MB truncada a 2 decimales que actualmente está en uso para todos los procesos que no son de background.

La consulta deberá considerar únicamente a los procesos que no son de background, es decir, la columna `p_name` es nula.

Tip: Es posible hacer uso del operador `union` para generar ambos registros.

C. Emplear los nombres de columna indicados en la siguiente imagen.

PROCESS_TYPE	TOTAL_PROCESS	TOTAL_PGA_ALLOCATED_MB	TOTAL_PGA_USED_MB
bkg-process	62	154.78	110.37
other-process	3	11.38	6.62

C3. Incluir en el reporte la consulta y la salida obtenida.

1.7. MONITOREO DE LA PGA

En este ejercicio se realizará una prueba empleando JMeter. El objetivo del ejercicio es simular la creación de un conjunto de sesiones que realicen consultas aleatorias que permitan el monitoreo del uso de la PGA.

Crear un plan de pruebas en JMeter que cumpla con las siguientes características.

- Crear un grupo de hilos con una duración de 4 minutos. El grupo estará formado por un solo hilo de ejecución que lanzará 2 sentencias SQL cada segundo para generar un histórico de uso de la PGA. Las sentencias deberán generar registros en las siguientes tablas:

```
create table <nombre>0403.t05 hist pga stats(
  sample_date          date,
  max pga mb           number ,
  pga_target_parameter_mb number ,
  pga_total_actual_mb  number ,
  pga_in_use_actual_mb  number ,
  pga free memory mb   number ,
  pga_process_count    number ,
  pga in use workareas auto number
);
```

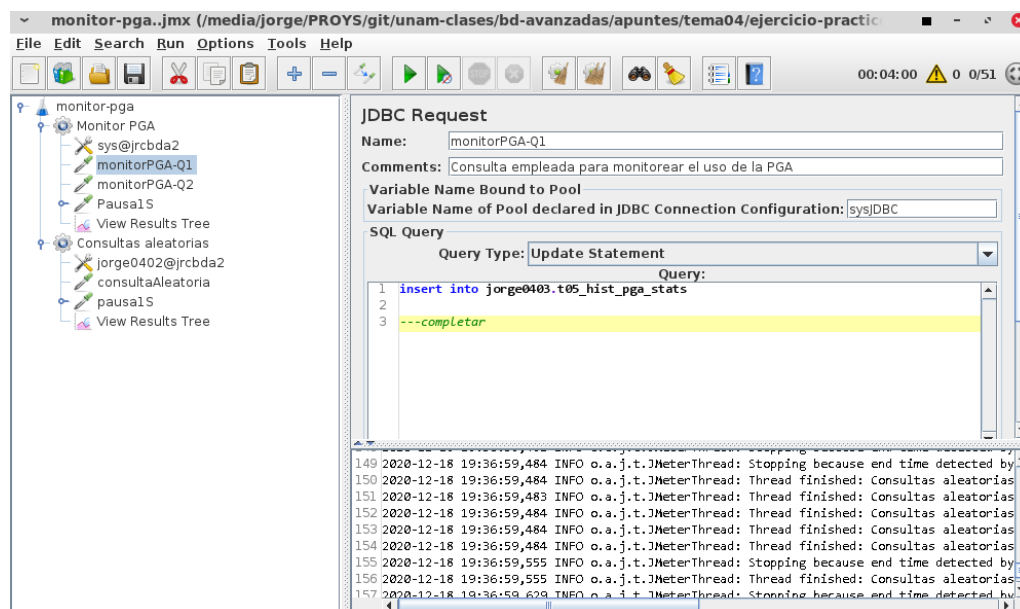
```
create table <nombre>0403.t06 hist pga process (
  sample_date          date,
  process type         varchar2(15),
  total_process        number,
  total_pga_allocated_mb number,
  total_pga_used_mb    number
);
```

Crear un script s-05-pga-monitoreo.sql que contenga la definición de estas 2 tablas. Ambas deberán pertenecer al usuario <nombre>0403. El contenido de la primera tabla es similar al contenido de la tabla t03_pga_stats. El contenido de la segunda tabla es similar al contenido de la tabla t04_pga_process. Cada segundo se deberán generar 3 registros. Un registro en la primera tabla y 2 registros en la segunda (proceso tipo other-process y proceso tipo bkg-process). La columna sample_date contiene la fecha a nivel de segundos en la que se realiza la inserción.

- Crear un segundo grupo de hilos de ejecución. El grupo estará formado por 50 hilos que simularán a 50 sesiones. Los hilos deberán comenzar a crearse a partir del minuto 1 y tendrán una duración de 2 minutos. La consulta es la misma realizada en el ejercicio práctico anterior:

```
select length(r_varchar)
from <nombre>0402.t01_random_data
where id = sys.dbms_random.value(?,?)
```

Notar que el usuario dueño de la tabla es el usuario creado en el ejercicio práctico anterior. Para poder acceder a los datos, el usuario <nombre>0402 deberá otorgar permisos de lectura al usuario <nombre>0403 para poder acceder a los datos.



C4. Incluir en el reporte la evidencia de la configuración de alguna de las 2 sentencias insert y log viewer durante la ejecución de la prueba (similar al a imagen anterior).

Una vez que la prueba en JMeter concluya, ejecutar una vez más las 2 sentencias SQL para generar los últimos 3 registros, esperar 10 minutos antes de ejecutarlas. Emplear SQL Developer o SQL*Plus.

1.8. GRÁFICAS DE MONITOREO.

Con base a los resultados obtenidos del monitoreo anterior generar las siguientes gráficas. Para cada caso proporcionar un breve texto que describa y justifique el comportamiento de la PGA. **C5. Incluir en el reporte** las gráficas y su correspondiente explicación

1.8.1. Uso de la PGA

- Fecha Vs PGA_TOTAL_ACTUAL_MB
- Fecha Vs PGA_IN_USE_ACTUAL_MB
- Fecha Vs PGA_PROCESS_COUNT

1.8.2. Uso de la PGA respecto a v\$process

- Fecha Vs TOTAL_PROCESS considerando únicamente a los procesos tipo `other-process`
- Fecha Vs TOTAL_PROCESS considerando únicamente a los procesos tipo `bkg-process`
 - ¿Qué diferencia se observa entre estas 2 gráficas?
- Fecha Vs TOTAL_PGA_USED_MB considerando únicamente a los procesos tipo `other-process`

1.9. 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-06-validador-main.sql
```

1.10. CONTENIDO DE LA ENTREGA.

- C1. Código y consulta de la tabla `t01_redo_log_buffer`
- C2. Código y consulta de la tabla `t03_pga_stats`
- C3. Código y consulta de la tabla `t04_pga_process`
- C4. Evidencia configuración sentencias `insert` en JMeter y Log Viewer
- C5. Gráficas y explicación de monitoreo.
- C6. Salida del validador.
- Entrega individual