

## **PRACTICAS TEMA 3.**

### **ESTRUCTURAS DE ALMACENAMIENTO.**

3.1. Comprobar el tamaño del bloque de la BD. ¿Cuántas cachés de datos hay definidas?. Definir la caché keep con 1M. Definir también una caché para bloques de 4Kb, con 1M.

3.2. Crear un tablespace SEGAUTO con gestión automática de segmentos.

3.3. Crear una tabla TABLA01 en el tablespace anterior, con una columna VARCHAR2(1000). Insertar tres filas de forma que ocupe varias extensiones. Borrar todas las filas con “delete” y comprobar las extensiones de la tabla. Hacer “truncate table” y comprobar de nuevo el espacio ocupado por la tabla.

3.4. Crear una tabla TABLA02 con una sola columna varchar2(100); y otra llamada TABLA02COMPRI igual que la anterior, pero con compresión de datos. Insertar 1000 filas en TABLA02 y, luego, esas mismas 1000 en TABLA02COMPRI. ¿Cuántos bloques necesita cada tabla para almacenar las mismas filas?

3.5. Crear tablespace TSP4K con tamaño de bloque de 4Kb. Crear una tabla TABLA03, igual que TABLA01, y comprobar cuantos bloques ocupan las dos y comparar los datos.

3.6. Comprobar los parámetros pga\_aggregate\_target, workarea\_size\_policy y sort\_area\_size. ¿Se puede poner pga\_aggregate\_target=0? ¿cómo desactivar pga\_aggregate\_target?

3.7. Crear una TABLA04 en el tablespace SEGAUTO, con cuatro columnas: c1 char(2), c2 varchar2(10), c3 date y c4 number(10,2). Estimar el tamaño medio de la fila, y el número medio de filas por bloque. Insertar 100 filas en la tabla , calcular las estadísticas y comprobar los datos calculados (número de filas, número de bloques y longitud media de la fila).

#### Recursos.

Máquina: [cursos.atca.um.es](http://cursos.atca.um.es)

Usuario unix: curso51 a curso70

Conexión máquina unix: ssh con Secure Shell

Bases de datos: CURSO51 a CURSO70

Conexión a bd: sqlplus desde sesión unix, usuario “/ as sysdba”

### 3.1. Comprobar el tamaño del bloque de la BD. ¿Cuántas cachés de datos hay definidas?. Definir la caché keep con 1M. Definir también una caché para bloques de 4Kb, con 1M.

#### Solución:

```
SQL> show parameter db_block_size
```

NAME	TYPE	VALUE
db_block_size	integer	2048

```
SQL> show parameter cache_size
```

NAME	TYPE	VALUE
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	0
db_8k_cache_size	big integer	0
<b>db_cache_size</b>	<b>big integer</b>	<b>0</b>
db_keep_cache_size	big integer	0
db_recycle_cache_size	big integer	0

(Está a cero pq está activada ASMM con sga\_target, pero se puede ver su tamaño actual en v\$sgainfo o v\$sga\_dynamic\_components)

```
SQL> select rpad(component,30),CURRENT_SIZE,MIN_SIZE,MAX_SIZE,oper_count
from v$sga_dynamic_components
where component like '%buffer cache%';
```

RPAD(COMPONENT,30)	CURRENT_SIZE	MIN_SIZE	MAX_SIZE	OPER_COUNT
DEFAULT buffer cache	<u>29360128</u>	29360128	0	16
KEEP buffer cache	0	0	0	0
RECYCLE buffer cache	0	0	0	0
DEFAULT 2K buffer cache	0	0	0	0
DEFAULT 4K buffer cache	0	0	0	0
DEFAULT 8K buffer cache	0	0	0	0
DEFAULT 16K buffer cache	0	0	0	0
DEFAULT 32K buffer cache	0	0	0	0

(Sólo está definida la caché por defecto, las demás están desactivadas. Ahora vamos a asignar la caché recycle y la de tamaño de bloque de 4K.)

```
SQL> alter system set db_keep_cache_size=1M;
```

System altered.

```
SQL> alter system set db_4K_cache_size=1M;
```

System altered.

```
SQL> show parameter cache_size
```

NAME	TYPE	VALUE
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	<u>8M</u>
db_8k_cache_size	big integer	0
db_cache_size	big integer	0
db_keep_cache_size	big integer	<u>8M</u>
db_recycle_cache_size	big integer	0

(Aunque hemos indicado 1M de tamaño para cada una de ellas, Oracle las ha dimensionado con 8M cada una; en este caso pq el tamaño del gránulo es de 4M y hay 2 cpus)

### 3.2. Crear un tablespace SEGAUTO con gestión automática de segmentos.

#### Solución:

```
SQL> create tablespace SEGAUTO
      datafile '/u02/oradata/CURSOxy/segauto01.dbf' size 1M
      extent management local uniform size 10K
      segment space management auto;
Tablespace created.
```

(¡Ojo!, cambiar "xy" por el número de tu usuario)

```
SQL> select bytes from dba_free_space where tablespace_name='SEGAUTO';
      BYTES
-----
      983040
```

### 3.3. Crear una tabla TABLA01 en el tablespace anterior, con una columna VARCHAR2(1000). Insertar tres filas de forma que ocupe varias extensiones. Borrar todas las filas con "delete" y comprobar las extensiones de la tabla. Hacer "truncate table" y comprobar de nuevo el espacio ocupado por la tabla.

#### Solución:

(Vamos a conectarnos como el usuario SYSTEM, y creamos la tabla en el tsp SEGAUTO)

```
SQL> connect system/systcursoXY
Connected.
SQL> create table TABLA01 (col01 VARCHAR2(1000)) tablespace SEGAUTO;
Table created.
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
```

(Hemos visto q estando vacía ocupa una extensión. Ahora insertamos 3 filas para provocar la asignación de otra extensión más)

```
SQL> insert into tabla01 values (rpad('A',1000,'A'));
1 row created.
SQL> insert into tabla01 values (rpad('A',1000,'A'));
1 row created.
SQL> insert into tabla01 values (rpad('A',1000,'A'));
1 row created.
SQL> commit;
Commit complete.
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
      10240
```

(Ahora vamos a borrar todas las filas y veremos q el bloque q se asignó para la tercera fila, no se libera sólo con borrar filas)

```
SQL> delete from tabla01;
3 rows deleted.
SQL> commit;
```

Commit complete.

```
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
      10240
```

(Para dejar la tabla con la extensión inicial q tenía después de ser creada, hay q hacer un "truncate" de la misma, ¡¡¡ OJO, q "truncate" borrará todas las filas !!!)

```
SQL> truncate table tabla01;
Table truncated.
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
```

### 3.4. Crear una tabla TABLA02 con una sola columna varchar2(100); y otra llamada TABLA02COMPRI igual que la anterior, pero con compresión de datos. Insertar 1000 filas en TABLA02 y, luego, esas mismas 1000 en TABLA02COMPRI. ¿Cuántos bloques necesita cada tabla para almacenar las mismas filas?

La funcionalidad de compresión de datos está disponible desde la versión 9i, pero debido a un bug, no funciona con versiones < 9.2.0.3 (no da errores al crear la tabla, pero no aparece columna correspondiente en dba\_tables, y no comprime datos).

Con esta práctica vamos a comprobar como Oracle (>=9i) puede comprimir los datos en cada bloque de una tabla, siempre y cuando haya valores repetidos de una misma columna dentro de un mismo bloque.

#### Solución:

(Conectamos como usuario SCOTT)

```
SQL> connect scott/tigercursoXY
Connected.
```

(Creamos la tabla TABLA02 sin comprimir, y otra igual, llamada TABLA02COMPRI, en este caso con compresión de datos)

```
SQL> create table TABLA02 (c1 varchar2(100)) tablespace users;
Table created.
SQL> create table TABLA02COMPRI (c1 varchar2(100)) tablespace users compress;
Table created.
SQL> select rpad(segment_name,20),blocks,extents from user_segments where segment_name
like 'TABLA02%';
RPAD(SEGMENT_NAME,20      BLOCKS      EXTENTS
-----
TABLA02                    32          1
TABLA02COMPRI              32          1
```

(Cargamos TABLA02 con 1000 filas iguales, relleno la única columna q tiene con el carácter 'a')

```
SQL> begin
      for i in 1..1000
      loop
          INSERT INTO TABLA02 VALUES (rpad('a',100));
```

```

        end loop;
    end;
/
PL/SQL procedure successfully completed.
SQL> commit;
Commit complete.
(Replicamos el contenido de TABLA02 en TABLA02COMPRI)

```

```

SQL> insert into TABLA02COMPRI select * from TABLA02;
1000 rows created.
SQL> commit;
Commit complete.

```

(Comprobamos los bloques ocupados en cada una de las tablas)

```

SQL> select rpad(segment_name,20),blocks,extents from user_segments where segment_name
like 'TABLA02%';
RPAD(SEGMENT_NAME,20      BLOCKS      EXTENTS
-----
TABLA02                    96          3
TABLA02COMPRI             64          2

```

(Como se observa, la compresión de datos funciona.)

### 3.5. Crear tablespace TSP4K con tamaño de bloque de 4Kb. Crear una tabla TABLA03, igual que TABLA01, y comprobar cuantos bloques ocupan las dos y comparar los datos.

#### Solución:

(Nos conectamos como SYSTEM y creamos el tablespace TSP4K con tamaño de bloque de 4K)

```

SQL> connect system/systcursoXY
Connected.
SQL> create tablespace tsp4k
        datafile '/u02/oradata/CURSOxy/tsp4k01.dbf' size 1M
        extent management local segment space management auto blocksize 4K;
Tablespace created.

```

(Ahora vamos a crear la tabla03 a partir de la tabla01; y vamos a vaciarlas las dos, y a volver a llenarlas con las mismas 100 filas)

```

SQL> create table tabla03 tablespace tsp4k as select * from tabla01;
Table created.

```

```

SQL> truncate table tabla01;
Table truncated.
SQL> truncate table tabla03;
Table truncated.
SQL> begin
        for i in 1..100
        loop
            INSERT INTO TABLA01 VALUES (rpad('a',1000,'a'));
        end loop;
    end;
/

```

```

PL/SQL procedure successfully completed.
SQL> insert into tabla03 select * from tabla01;
100 rows created.
SQL> COMMIT;
Commit complete.

```

(Si ahora comprobamos los bloques que ocupan ambas tablas, veremos q TABLA03 ocupa la mitad pues su tamaño de bloque es el doble de grande.)

```
SQL> select RPAD(SEGMENT_NAME,30),bytes,blocks
       from dba_segments where segment_name in ('TABLA01','TABLA03');
RPAD(SEGMENT_NAME,30)          BYTES          BLOCKS
-----
TABLA01                        225280          110
TABLA03                        196608           48
```

### 3.6. Comprobar los parámetros pga\_aggregate\_target, workarea\_size\_policy y sort\_area\_size. ¿Se puede poner pga\_aggregate\_target=0? ¿cómo desactivar pga\_aggregate\_target?

Con la versión 9i, Oracle introduce la posibilidad de gestionar de forma compartida las PGAs (área de memoria asociada a cada proceso oracle, fuera de la SGA, y que contiene, por ejemplo, el área de sort). Asignando el parámetro pga\_aggregate\_target (valor mínimo 10M), se crea una zona de memoria limitada, para todos los procesos oracle; de forma que cada sesión tomará el espacio que necesite, y lo liberará al desconectarse. El parámetro workarea\_size\_policy=auto determina el uso de dicha funcionalidad. Si lo asignamos a “manual”, se usará la gestión tradicional basada en el parámetro sort\_area\_size. Todo ésto, referido sólo a las sesiones con servidor dedicado.

#### Solución:

```
SQL> show parameter pga
NAME                                TYPE          VALUE
-----
pga_aggregate_target                big integer   10485760
```

```
SQL> show parameter workarea
NAME                                TYPE          VALUE
-----
workarea_size_policy                string        AUTO
```

```
SQL> show parameter sort_area_size
NAME                                TYPE          VALUE
-----
sort_area_size                      integer       65536
```

(Como workarea\_size\_policy vale AUTO, se activa el uso de pga\_aggregate\_target; que no podrá valer menos de 10M. De este modo, para las sesiones con servidores dedicados, se ignorará sort\_area\_size.)

```
SQL> alter system set pga_aggregate_target=9M;
ORA-02097: parameter cannot be modified because specified value is invalid
ORA-00093: pga_aggregate_target must be between 10M and 4096G-1
SQL> alter system set pga_aggregate_target=11M;
System altered.
```

```
SQL> show parameter pga
NAME                                TYPE          VALUE
-----
pga_aggregate_target                big integer   11534336
```

(Si queremos desactivar pga\_aggregate\_target, no lo dejaremos a cero, sino q pondremos work\_area\_policy a MANUAL.)

```
SQL> alter system set pga_aggregate_target=0;
ORA-02097: parameter cannot be modified because specified value is invalid
ORA-00093: pga_aggregate_target must be between 10M and 4096G-1
```

```
SQL> alter system set workarea_size_policy=manual;
System altered.
```

(Lo volvemos a dejar en AUTO)

```
SQL> alter system set workarea_size_policy=auto;
System altered.
```

### 3.7. Crear una TABLA04 en el tablespace SEGAUTO, con cuatro columnas: c1 char(2), c2 varchar2(10), c3 date y c4 number(10,2). Estimar el tamaño medio de la fila, y el número medio de filas por bloque. insertar 100 filas en la tabla, calcular las estadísticas y comprobar los datos calculados (número de filas, número de bloques y longitud media de la fila).

El tamaño del bloque de la BD lo indica el parámetro db\_block\_size; y el tamaño del bloque concreto para un tablespace lo podemos ver en DBA\_TABLESPACES.BLOCK\_SIZE.

EL tamaño de una fila es

$\text{cabecera\_fila} + \text{rowid} + \text{cabecera\_col1} + \text{col1} + \text{cabecera\_col2} + \text{col2} + \dots$ ; donde  $\text{cabecera\_fila} = 3$ ,  $\text{rowid} = 6$  y  $\text{cabecera\_col1} = 1$  (ó 3 si la columna  $\geq 250$  bytes).

El espacio útil de un bloque es  $\text{tamaño\_bloque} - \text{overhead} - \text{transacciones} - \text{pctfree}$ ; donde  $\text{overhead} = 107$  bytes, y  $\text{transacciones} = 23$  (23 bytes por cada transacción, mínimo habrá una); y  $\text{pctfree}$  es el porcentaje de espacio libre que dejamos en cada bloque para "updates" (en este caso vale 10).

#### Solución:

```
SQL> connect system/systcursoXY
Connected.
```

```
SQL> select BLOCK_SIZE from dba_tablespaces where tablespace_name='SEGAUTO';
BLOCK_SIZE
```

```
-----
2048
```

```
SQL> create table tabla04 (c1 char(2), c2 varchar2(10), c3 date, c4 number(10,2))
tablespace segauto;
Table created.
```

```
SQL> select PCT_FREE, INI_TRANS, NUM_ROWS, BLOCKS, EMPTY_BLOCKS, AVG_SPACE, AVG_ROW_LEN
from dba_tables where table_name='TABLA04';
PCT_FREE  INI_TRANS  NUM_ROWS  BLOCKS  EMPTY_BLOCKS  AVG_SPACE  AVG_ROW_LEN
-----
10         1
```

#### ¿Qué ocupa cada fila?

Fila=cabecera\_fila+rowid+cabecer\_col+long\_col+ ...

Fila\_mas\_corta=3+6+1+2+1+1+1+8+1+6=29

Fila\_mas\_larga=3+6+1+2+1+10+1+8+1+6=38

Fila\_media=33bytes

**¿Espacio útil en cada bloque?**

Bloque=2048bytes.

Util=2048-overhead-23-pctfree

Util=2048-107-23-205=1713bytes

**¿Cuántas filas caben en un bloque?**

1713/33=51 filas

Ahora vamos a insertar 100 filas en tabla04 y comprobaremos los datos estimados con los datos calculados por Oracle.

```
SQL> begin
      for i in 1..100
      loop
        INSERT INTO TABLA04 VALUES ('01', '12345', SYSDATE, 12345678.12);
      end loop;
    end;
  /
```

PL/SQL procedure successfully completed.

```
SQL> commit;
Commit complete.
```

```
SQL> analyze table tabla04 compute statistics;
Table analyzed.
```

```
SQL> select PCT_FREE, INI_TRANS, NUM_ROWS, BLOCKS, EMPTY_BLOCKS, AVG_SPACE, AVG_ROW_LEN from
dba_tables where table_name='TABLA04';
```

PCT_FREE	INI_TRANS	NUM_ROWS	BLOCKS	EMPTY_BLOCKS	AVG_SPACE	AVG_ROW_LEN
10	1	<u>100</u>	<u>2</u>	3	476	<u>27</u>