

## **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 igual que la anterior, pero con compresión de datos.

3.5. Crear tablespace TSP2K 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.atica.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>4194304</b>
db_keep_cache_size	big integer	0
db_recycle_cache_size	big integer	0

```
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	4194304
db_8k_cache_size	big integer	0
db_cache_size	big integer	4194304
db_keep_cache_size	big integer	4194304
db_recycle_cache_size	big integer	0

### 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.
```

```
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:**

```
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
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
SQL> delete from tabla01;
3 rows deleted.
SQL> commit;
Commit complete.
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
      10240
SQL> truncate table tabla01;
Table truncated.
SQL> select bytes from dba_EXTENTS where segment_name='TABLA01';
      BYTES
-----
      10240
```

**3.4. Crear una tabla TABLA02 igual que la anterior, pero con compresión de datos.**

Debido a un bug, la compresión de datos 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). Hay que instalar patchset 9.2.0.3 o superior (ya va, al menos, por el 9.2.0.6).

**Solución:**

```
SQL> connect system/systcursoXY
Connected.
SQL> CREATE TABLE TABLA02 (C1 VARCHAR2(1000)) TABLESPACE SEGAUTO COMPRESS;
Table created.
```

### 3.5. Crear tablespace TSP2K 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:

```
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.
```

```
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>
  1 begin
  2   for i in 1..100
  3   loop
  4     INSERT INTO TABLA01 VALUES (rpad('a',1000,'a'));
  5   end loop;
  6 end;
  7 /
```

PL/SQL procedure successfully completed.

```
SQL> insert into tabla03 select * from tabla01;
100 rows created.
```

```
SQL> COMMIT;
```

Commit complete.

```
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 esto, 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

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

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.

```

**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 cabecera\_fila+rowid+cabecera\_col1+col1+cabecera\_col2+col2+...; donde cabecera\_fila=3, rowid=6 y cabecera\_coli=1 (ó 3 si la columna >= 250bytes).

El espacio útil de un bloque es tamaño\_bloque – overhead – transacciones - pctfree; donde overhead = 107bytes, y transacciones = 23 (23 bytes por cada transacción, mínimo habrá una); y 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>
1  begin
2  for i in 1..100
3  loop
4  INSERT INTO TABLA04 VALUES ('01', '12345',SYSDATE,12345678.12);
5  end loop;
6  end;
7 /

```

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           100           2             3           476           27

```