

# Ejercicio 2 - tema 7

## Almacenamiento en data files

Vicente Romero Andrade

### I. OBJETIVO

EL objetivo es poner en práctica las tareas de administración que permitan el almacenamiento de datos de una tabla en un tablespace y data file específico configurado previamente

### II. DESARROLLO

#### A. sentencias

```

1  whenever sqlerror exit rollback
2  set serveroutput on
3  connect sys/system2 as sysdba
4  --A
5  SELECT
6      FILE_NAME,
7      FILE_ID,
8      TRUNC((BYTES/(1024*1024)),2) SIZE_MB
9  FROM DBA_DATA_FILES WHERE TABLESPACE_NAME = '
10     STORE_TBS_MULTIPLE';
11  --B
12  SELECT
13      TRUNC((SUM(DF.BYTES)-NVL(SUM(S.BYTES),0))
14      /(1024*1024),2) MB_LIBRES,
15      (SUM(DF.BLOCKS)-NVL(SUM(S.BLOCKS),0))
16      BLOQUES_DISPONIBLES
17  FROM DBA_DATA_FILES DF
18  LEFT JOIN DBA_SEGMENTS S
19      ON S.TABLESPACE_NAME = DF.TABLESPACE_NAME
20  WHERE DF.TABLESPACE_NAME='STORE_TBS_MULTIPLE';
21  --C
22  --CREATE USER VRA_TBS_MULTIPLE IDENTIFIED BY
23  VRA_TBS_MULTIPLE
24  --quota unlimited on store_tbs_multiple
25  --default tablespace store_tbs_multiple;
26  --D
27  declare
28      v_count number;
29      v_username varchar2(30) := 'VRA_TBS_MULTIPLE';
30      v_table varchar2(30) := 'VRA_TBS_MULTIPLE';
31  begin
32      --Verificar si la table existe
33      select count(*) into v_count
34      from all_tables
35      where table_name = v_table
36      and owner = v_username;
37
38      if v_count > 0 then
39          execute immediate 'drop table '||v_username||'
40          .'||v_table;
41      end if;
42      execute immediate 'create table '||v_username||'
43      .'||v_table||' (
44          str char(1024 byte)
45          ) segment creation immediate';
46  end;
47  /
48  --E
49  SELECT DF.FILE_NAME,
50      DF.FILE_ID,
51      COUNT(DE.SEGMENT_NAME) NUMERO_EXTENSIONES,
52      SUM(DE.BYTES/(1024*1024)) TOTAL_MB,
53      SUM(DE.BLOCKS) BLOQUES_RESERVADOS

```

```

48  FROM DBA_SEGMENTS DS
49  JOIN DBA_DATA_FILES DF
50      ON DS.HEADER_FILE = DF.FILE_ID
51  JOIN DBA_DATA_FILES DF
52      ON DS.HEADER_FILE = DF.FILE_ID
53  JOIN DBA_EXTENTS DE
54      ON DS.SEGMENT_NAME = DE.SEGMENT_NAME
55  WHERE DS.SEGMENT_NAME like '%VRA_TBS_MULTIPLE%'
56  GROUP BY DF.FILE_NAME, DF.FILE_ID;
57  --F
58  declare
59      v_count number := 0;
60  begin
61      while v_count < 512 loop
62          insert into VRA_TBS_MULTIPLE.VRA_TBS_MULTIPLE (
63              str) values('S');
64          v_count := v_count + 1;
65      end loop;
66      commit;
67  end;
68  /
69  --G
70  SELECT DF.FILE_NAME,
71      DF.FILE_ID,
72      COUNT(DE.SEGMENT_NAME) NUMERO_EXTENSIONES,
73      SUM(DE.BYTES/(1024*1024)) TOTAL_MB,
74      SUM(DE.BLOCKS) BLOQUES_RESERVADOS
75  FROM DBA_SEGMENTS DS
76  JOIN DBA_DATA_FILES DF
77      ON DS.HEADER_FILE = DF.FILE_ID
78  JOIN DBA_DATA_FILES DF
79      ON DS.HEADER_FILE = DF.FILE_ID
80  JOIN DBA_EXTENTS DE
81      ON DS.SEGMENT_NAME = DE.SEGMENT_NAME
82  WHERE DS.SEGMENT_NAME like '%VRA_TBS_MULTIPLE%'
83  GROUP BY DF.FILE_NAME, DF.FILE_ID;
84  --H
85  declare
86      v_count number := 0;
87  begin
88      while v_count < 512*5 loop
89          insert into VRA_TBS_MULTIPLE.VRA_TBS_MULTIPLE (
90              str) values('S');
91          v_count := v_count + 1;
92      end loop;
93      commit;
94  end;
95  /
96  SELECT DF.FILE_NAME,
97      DF.FILE_ID,
98      COUNT(DE.SEGMENT_NAME) NUMERO_EXTENSIONES,
99      SUM(DE.BYTES/(1024*1024)) TOTAL_MB,
100     SUM(DE.BLOCKS) BLOQUES_RESERVADOS
101  FROM DBA_SEGMENTS DS
102  JOIN DBA_DATA_FILES DF
103      ON DS.HEADER_FILE = DF.FILE_ID
104  JOIN DBA_DATA_FILES DF
105      ON DS.HEADER_FILE = DF.FILE_ID
106  JOIN DBA_EXTENTS DE
107      ON DS.SEGMENT_NAME = DE.SEGMENT_NAME
108  WHERE DS.SEGMENT_NAME like '%VRA_TBS_MULTIPLE%'
109  GROUP BY DF.FILE_NAME, DF.FILE_ID;
110  --I
111  SELECT
112      TRUNC((SUM(DF.BYTES)-NVL(SUM(S.BYTES),0))

```

```

111 / (1024*1024),2) MB_LIBRES,
112 (SUM(DF.BLOCKS)-NVL(SUM(S.BLOCKS),0))
113 BLOQUES_DISPONIBLES
114 FROM DBA_DATA_FILES DF
115 LEFT JOIN DBA_SEGMENTS S
116 ON S.TABLESPACE_NAME = DF.TABLESPACE_NAME
117 WHERE DF.TABLESPACE_NAME='STORE_TBS_MULTIPLE';
118
119 whenever sqlerror continue

```

Código 1. s-00-datafile.sql

FILE_NAME	FILE_ID	SIZE_MB
1 /u01/app/oracle/oradata/VRABDA2/store_tbs_multiple_01.dbf	6	10
2 /u02/app/oracle/oradata/VRABDA2/store_tbs_multiple_02.dbf	7	10
3 /u03/app/oracle/oradata/VRABDA2/store_tbs_multiple_03.dbf	8	10

Figure 1. Salida punto A

	MB_LIBRES	BLOQUES_DISPONIBLES
1	135.93	17400

Figure 2. Salida punto B

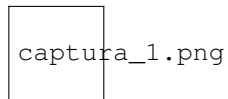


Figure 3. Salida punto E

### III. CONCLUSIONES

Se encontro una forma eficiente de consultar los segmentos creados en una tabla, el unico inconveniente es que estos dependen de que sean creados con el nombre de la tabla para su busqueda.