

Ejercicio 10 - tema 7

Administración de datos Undo

Vicente Romero Andrade

I. OBJETIVO

El objetivo es comprender e ilustrar los efectos que produce un tablespace undo sin espacio durante la operación de la base de datos.

II. DESARROLLO

A. Generar una sentencia SQL que muestre el tablespace usando que actualmente está en uso

```
1 SELECT value FROM v$parameter WHERE name='
    undo_tablespace';
```

Código 1. sentencia consulta tablespace undo

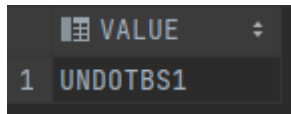


Figure 1. Salida punto A

B. Generar una sentencia SQL que genere un nuevo tablespace undos2, cuya característica principal es contar con un espacio muy limitado y por lo tanto la base de datos estará en riesgo de generar errores al requerir más espacio undos del disponible. Nombrar al tablespace undots2, asignarle un solo data file ubicado en 'u01/app/oracle/oradata/;ORACLE_SID;/undots2.dbf' con una capacidad únicamente de 30 MB. El data file no deberá auto extenderse. La administración de sus extensiones debe ser local

```
1 create undo tablespace undotbs2
2 datafile '/u01/app/oracle/oradata/VRABDA2/undotbs2.
   dbf' size 30M
3 autoextend off
4 extent management local autoallocate;
```

Código 2. sentencia generar tablespace undo

C. Generar la sentencia SQL necesaria para que la instancia ahora haga uso del nuevo tablespace en lugar el actual. Aplicar el cambio únicamente mientras la instancia esté iniciada

```
1 alter system set undo_tablespace='UNDOTBS2' scope=
    memory;
```

Código 3. sentencia generar tablespace undo

D. Crear una nueva sesión con el usuario SYS, ejecutar nuevamente la sentencia del inciso A para verificar que la instancia ahora está haciendo uso del tablespace undotbs2

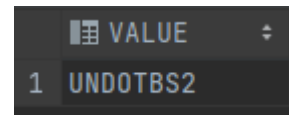


Figure 2. Salida punto D

E. Considerar la vista v\$undostat. Generar la siguiente consulta. Mostrar la fecha inicio y fin del periodo de la muestra hasta nivel de segundos, el id del tablespace que se está empleando, total de bloques undo empleados, total de transacciones, id de la consulta que tardó el mayor tiempo en ejecutarse, tiempo en segundos de la consulta que más tiempo tardó en ejecutarse durante el periodo de la muestra, número de bloques undo activos, no expirados y expirados; el valor del parámetro undo_retention calculado por la instancia. Ordenar los registros con base a la fecha de inicio de forma descendente. Mostrar únicamente los primeros 20 registros

```
1 SELECT BEGIN_TIME,
2        END_TIME,
3        UNDOTSN,
4        UNDOBLKS as TOTAL_BLOQUES_UNDO_USADOS,
5        TXNCOUNT as NUM_TRANSACCIONES,
6        MAXQUERYID,
7        MAXQUERYLEN,
8        ACTIVEBLKS,
9        UNEXPIREDBLKS,
10       EXPIREDBLKS,
11       TUNED_UNDORETENTION
12 FROM V$UNDOSTAT WHERE ROWNUM<=20 ORDER BY BEGIN_TIME
13        DESC;
```

Código 4. sentencia consulta tablespace stats

[illegible]

Figure 3. Salida punto E

F. Considerar el periodo de muestreo más reciente de la consulta realizada en el inciso E

1) ¿Cuántos bloques podrían ser sobrescritos sin causar mayores inconvenientes?: Los que son EXPIRED pueden ser sobrescritos por lo tanto es 0 así que ningún bloque se sobrescribiría.

2) ¿Cuántos bloques NO pueden ser sobrescritos aun?:
160 bloques que corresponden a los activos

G. Observar los valores de la columna undotsn de la consulta del inciso E. Confirmar que a partir del cambio del tablespace undo, el valor de esta columna cambia. Generar una sentencia SQL que permita mostrar los nombres de los tablespaces asociados a los valores de esta columna. Tip: auxiliarse de v\$tablespace. En la consulta incluir únicamente el periodo de la muestra, id del tablespace y su nombre

```
1 SELECT BEGIN_TIME,
2        END_TIME,
3        UNDOTSN,
4        T.NAME
5 FROM V$UNDOSTAT U JOIN V$TABLESPACE T ON U.UNDOTSN=T
   .TS# WHERE ROWNUM<=20 ORDER BY BEGIN_TIME DESC;
```

Código 5. sentencia consulta tablespace id-nombre

	BEGIN_TIME	END_TIME	UNDOTSN	NAME
1	2021-07-21 20:38:13	2021-07-21 20:48:10	8	UNDOTBS2
2	2021-07-21 20:28:13	2021-07-21 20:38:13	8	UNDOTBS2
3	2021-07-21 20:18:13	2021-07-21 20:28:13	8	UNDOTBS2
4	2021-07-21 20:08:13	2021-07-21 20:18:13	8	UNDOTBS2
5	2021-07-21 19:58:13	2021-07-21 20:08:13	2	UNDOTBS1
6	2021-07-21 19:48:13	2021-07-21 19:58:13	2	UNDOTBS1
7	2021-07-21 19:38:13	2021-07-21 19:48:13	2	UNDOTBS1
8	2021-07-21 19:28:13	2021-07-21 19:38:13	2	UNDOTBS1
9	2021-07-21 19:18:13	2021-07-21 19:28:13	2	UNDOTBS1
10	2021-07-21 19:08:13	2021-07-21 19:18:13	2	UNDOTBS1

Figure 4. Salida punto G

H. Generar una consulta que muestre las siguientes columnas: Nombre del tablespace undo que fue creado en pasos anteriores, total de bloques que contiene, total de bloques libres, porcentaje de bloques libres. Confirmar que el porcentaje de bloques libres es bajo, menor al 20%. Esta condición pone en riesgo la operación correcta que requieren hacer uso de los datos Undo

```
1 SELECT T.TABLESPACE_NAME,
2        DF.BLOCKS TOTAL_BLOQUES,
3        (DF.BLOCKS-SUM(E.BLOCKS)) BLOQUES_LIBRES,
4        TRUNC(((DF.BLOCKS-SUM(E.BLOCKS))/DF.BLOCKS *100)
5        ,2) PORCENTAJE_LIBRE FROM DBA_TABLESPACES T
6 JOIN DBA_DATA_FILES DF
7 ON T.TABLESPACE_NAME=DF.TABLESPACE_NAME
8 JOIN DBA_UNDO_EXTENTS E
9 ON T.TABLESPACE_NAME=E.TABLESPACE_NAME
10 WHERE T.TABLESPACE_NAME='UNDOTBS2'
11 GROUP BY T.TABLESPACE_NAME, DF.BLOCKS;
```

Código 6. sentencia consulta tablespace bloques libres

TABLESPACE_NAME	TOTAL_BLOQUES	BLOQUES_LIBRES	PORCENTAJE_LIBRE
UNDOTBS2	3840	3080	80.2

Figure 5. Salida punto H

I. Empleando el usuario creado en temas anteriores, generar una tabla con la siguiente estructura

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.