

Temporary UNDO en Oracle Database 12c

Por Francisco Riccio

Introducción

En previas versiones a Oracle Database 12c si uno realizaba operaciones DML (Insert/Delete/Update) sobre una tabla temporal, dichas operaciones generaban entradas en el UNDO tablespace ocasionando que se quede con menos espacio e implícitamente generando mayor cantidad de información en los redo logs.

La versión Oracle Database 12c permite solucionar este problema de una manera sencilla que se verá en la implementación.

Implementación

La versión 12c ha implementado el parámetro TEMP_UNDO_ENABLED que permite modificar el comportamiento por default que se tenía en las versiones previas.

Su valor por default es FALSE, cambiándolo a TRUE permite que toda operación DML sobre una tabla temporal no genere entradas en el UNDO tablespace sino en el tablespace TEMPORAL haciendo que implícitamente no se genere información de redo logs.

El único requisito solicitado es que nuestra base de datos tenga configurado el parámetro COMPATIBLE con el valor 12.0.0 o superior.

En el siguiente ejemplo se está modificando el parámetro al valor de TRUE en pro de utilizar el nuevo feature que nos presenta Oracle.

```
SQL> show parameter undo
```

NAME	TYPE	VALUE
temp_undo_enabled	boolean	FALSE
undo_management	string	AUTO
undo_retention	integer	900
undo_tablespace	string	UNDOTBS1

```
SQL> alter system set temp_undo_enabled=true;
```

Sistema modificado.

El parámetro puede ser modificado a nivel de instancia o sesión.

A los segmentos creados por las operaciones DML sobre tablas temporales que se almacenarán en el tablespace TEMPORAL se definen con el nombre: TEMPORARY UNDO SEGMENTS.

Si deseamos monitorear y afinar la información de tipo Temporary Undo debemos consultar la vista: V\$TEMPUNDOSTAT;

```
SELECT TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,  
TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,  
UNDOTSN, UNDOBLKCNT, MAXCONCURRENCY  
FROM V$TEMPUNDOSTAT;
```

Ejemplo:

Primero validamos que no tenemos ninguna información de tipo TEMP UNDO y posteriormente procedemos a crear una tabla temporal llamada TABLA_TEMP.

```
SQL> SELECT TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,  
TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,  
UNDOTSN, UNDOBLKCNT, MAXCONCURRENCY  
FROM V$TEMPUNDOSTAT;
```

BEGIN_TIME	END_TIME	UNDOTSN	UNDOBLKCNT	MAXCONCURRENCY
02/13/2013 17:28:44	02/13/2013 17:56:50	0	0	0

```
SQL> create global temporary table TABLA_TEMP (campo1 number)  
on commit preserve rows;
```

Tabla creada.

Poblaremos la tabla temporal con 90,609 filas y validamos que efectivamente si se ha logrado crear segmentos de tipo TEMP UNDO donde incluso ha utilizado 28 data blocks en un rango de tiempo determinado.

```
SQL> insert into TABLA_TEMP select rownum from dba_objects;
```

90609 filas creadas.

```
SQL> commit;
```

Confirmación terminada.

```
SQL> SELECT TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,  
TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,  
UNDOTSN, UNDOBLKCNT, MAXCONCURRENCY  
FROM V$TEMPUNDOSTAT;
```

BEGIN_TIME	END_TIME	UNDOTSN	UNDOBLKCNT	MAXCONCURRENCY
02/13/2013 17:48:44	02/13/2013 17:58:53	3	28	0
02/13/2013 17:28:44	02/13/2013 17:48:44	2147483647	0	0

Este nuevo feature nos provee los siguientes beneficios:

- Reduce la cantidad de espacio que se utiliza en el tablespace UNDO permitiendo que el tablespace solo mantenga información de segmentos permanentes dando la posibilidad que el parámetro UNDO_RETENTION se pueda configurar de mejor manera.

- Reduce la cantidad de espacio utilizado por los redo logs, permitiendo un ahorro de espacio y mejora de performance en generar menos operaciones de I/O.
- Está soportado sobre un Oracle Active Data Guard.

Publicado por Ing. Francisco Riccio. Es un IT Oracle Specialist e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application / Oracle Database / Oracle Developer.

e-mail: francisco@friccio.com

web: www.friccio.com