

# 回滚段

---

## 概念

- Oracle Database creates and manages information that is used to roll back, or undo, changes to the database. Such information consists of records of the actions of transactions, primarily before they are committed. These records are collectively referred to as undo.
- 类似于只保留一个历史状态的快照，用于在数据库数据出错等一些状况发生时用于恢复到某个时间点的数据状态
  - 1) 一个事务，使用一个undo段来保存该事务涉及的所有数据
  - 2) 多个并发事务，可以同时将数据写到同一个undo段
  - 3) undo段就是用来保存修改前的数据，直到数据被再次修改。
  - 4) undo段头，包含一个事务表信息，记录了当前事务的undo存储数据。

## 作用

- 1) 当使用ROLLBACK语句时，回滚事务。事务处理回退，当某事务处理修改表中某行时，被修改的列的旧映像（要还原的数据）将存储在还原段中。如果将该事务处理回退，则Oracle服务器通过将还原段中的值写回到该行。
- 2) 恢复数据库，撤销已经通过redo写入到数据文件的未提交数据。
- 3) 提供读一致性，当一个用户访问数据时，正好该数据正在被更改，undo通过保留之前数据的拷贝来保证读一致性。
- 4) 通过undo可以实现闪回查询，查询历史数据
- 5) 闪回数据库

UNDO EXTENT 三种状态： 1) UNEXPIRED 未过期的，可复用。 2) EXPIRED 过期的，可复用。 3) ACTIVE 活动的，不可用，必须等待事务提交后就立即变为unexpired。

```
SQL> select status,sum(bytes),count(status) from dba_undo_extents group by status;
```

STATUS	SUM(BYTES)	COUNT(STATUS)
UNEXPIRED	6619136	11
EXPIRED	3801088	43

```
SQL> insert into test values(354);
```

已创建 1 行。

```
SQL> select status,sum(bytes),count(status) from dba_undo_extents group by status;
```

STATUS	SUM(BYTES)	COUNT(STATUS)
UNEXPIRED	5570560	10
EXPIRED	3801088	43
ACTIVE	1048576	1

```
SQL> commit;
```

提交完成。

```
SQL> select status,sum(bytes),count(status) from dba_undo_extents group by status;
```

STATUS	SUM(BYTES)	COUNT(STATUS)
UNEXPIRED	6619136	11
EXPIRED	3801088	43

```
SQL> █
```

## 操作

# 创建新的undo表空间

```
SQL> create undo tablespace UNDOTBS2 datafile '/oradata/zzdb1/undotbs02.dbf' size 16M;
```

表空间已创建。

```
SQL> select SEGMENT_NAME,OWNER,TABLESPACE_NAME,STATUS from dba_rollback_segs;
```

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1_81441714\$	PUBLIC	UNDOTBS	ONLINE

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
_SYSSMU11_3650220049\$	PUBLIC	UNDOTBS2	OFFLINE

已选择21行。

# 切换undo表空间

```
SQL> Alter system set undo_tablespace=UNDOTBS2;
```

系统已更改。

```
SQL> select SEGMENT_NAME,OWNER,TABLESPACE_NAME,STATUS from dba_rollback_segs;
```

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1_81441714\$	PUBLIC	UNDOTBS	OFFLINE

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
_SYSSMU11_3650220049\$	PUBLIC	UNDOTBS2	ONLINE

已选择21行。

```
SQL> Alter system set undo_tablespace=UNDOTBS;
```

系统已更改。

# 查看undo表空间的一些信息

```
SQL> select SEGMENT_NAME,OWNER,TABLESPACE_NAME,STATUS from dba_rollback_segs;
```

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1_81441714\$	PUBLIC	UNDOTBS	ONLINE

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
-----	-----	-----	-----
_SYSSMU11_3650220049\$	PUBLIC	UNDOTBS2	OFFLINE

已选择21行。

# 删除undo表空间

SQL> drop tablespace undotbs2;

表空间已删除。

SQL> select SEGMENT\_NAME,OWNER,TABLESPACE\_NAME,STATUS from dba\_rollback\_segs;

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
-----	-----	-----	-----
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1_81441714\$	PUBLIC	UNDOTBS	ONLINE

已选择11行。

## UNDO生成与高水位线-实验

(1) 创建一个空表。(2) 对它做一个全部扫描，观察读表所执行的I/O数量。(3) 在表中填入许多行(但没有提交)。(4) 回滚这个工作，并撤销。(5) 再次进行全表扫描，观察所执行的I/O数。

# 创建新表

```
SQL> create table t as select * from all_objects where 1=0;
```

Table created.

# 查询这个表，SQL\*Plus中启用了AUTOTRACE，以便能测试I/O。

```
SQL> set autot
```

Usage: SET AUTOT[RACE] {OFF | ON | TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]

```
SQL> set autotrace trace stat
```

```
SQL> help set
```

# 插入数据

```
SQL> insert into t select * from all_objects;
```

```
SQL> select segment_name,extent_id,bytes/1024,status from dba_undo_extents where
status='ACTIVE' order by extent_id ;
```

# 查询历史undo是否还有事务(包含回滚事务)

```
SQL> SELECT a.tablespace_name,a.segment_name,b.ktuxesta,b.ktuxecfl,
b.ktuxeusn||'.'||b.ktuxeslt||'.'||b.ktuxesqn trans
FROM dba_rollback_segs a, x$ktuxe b
WHERE a.segment_id = b.ktuxeusn
AND a.tablespace_name = UPPER('&tsname')
AND b.ktuxesta <> 'INACTIVE';
```

```
SQL> select status, sum(bytes), count(status)
from dba_undo_extents
group by status
order by 1;
```

```
SQL> insert into t select * from t;
```

```
SQL> insert into t select * from t;
```

```
SQL> select segment_name,extent_id,bytes/1024,status
from dba_undo_extents
where status='ACTIVE'
order by extent_id ;
```

```
SQL> select status, sum(bytes), count(status)
from dba_undo_extents
group by status
order by 1;
```

```
SQL> rollback;
```

Rollback complete.

```
SQL> select status, sum(bytes), count(status)
from dba_undo_extents
group by status
order by 1;
```

# 打开统计

```
SQL> set autot trace stat;
```

# # 现在IO读为4069次，比之前34次多。

```
SQL> select * from t;
```

no rows selected

关闭

```
SQL> set autot off;
```

**回滚只是一个“将数据库还原”的逻辑操作。数据库并非真的还原成原来的样子，只是逻辑上相同而已。**