

## 2012 年 11 月 7 日学习笔记-管理还原数据

### 1. 还原数据的管理方法

当某个进程修改数据时, `oracle` 首先将它的原始值 (还原数据) 存入一个还原段中。一个事务只能将它的全部还原数据存放在同一个还原段中。但是多个并行的事务可以写一个还原段。每个还原段都有一个段头, 在该段头中包含了一个事务表, 该表中存放着有关使用这一还原段的当前事务信息。

还原 (`undo`) 一词在以前的版本中被称为回滚 (`rollback`)

没提交可用 `rollback`, 提交了可用 `flashback` (闪回)。

### 2. 还原数据的管理方法

自动的还原数据管理

手动的还原数据管理

### 3. `oracle` 使用还原段共有 3 个目的:

事务回滚

当一个用户发了一些 DML 语句之后改了主意使用了 `ROLLBACK` 语句回滚了他所做的事务。此时 `oracle` 服务器就会将还原段中的原始数据回写到原来的数据行中。

事务恢复

如果当事务正在进行期间实例崩溃了, 则当数据库再次打开时 `oracle` 服务器就要还原 (回滚) 所有没有提交的变化。这种还原 (回滚) 就是事务恢复的一部分。

保证数据的读一致性

当事务在进行期间, 数据库中的其他用户不应看到任何这些事务所做的没有提交的变化。一个语句在执行期间所看到的数据是静止的。

### 4. 还原段的类型

`oracle` 数据库系统中的还原段分别具有以下不同的类型:

#### 4.1 系统还原段

仅为系统表空间中的对象变化所用。它是在数据库创建时在系统表空间中创建的。系统还原段即可以存在并工作在自动模式下, 也可以存在并工作在手动模式下。

#### 4.2 非系统还原段

为其他表空间中对象的变化所使用。当一个数据库具有多个表空间时就需要至少一个手动的非系统还原段或一个自动管理的还原表空间。

##### 4.2.1 自动管理的还原段

需要一个还原表空间 (`undo tablespace`), `oracle` 服务器将自动维护还原表空间中的还原数据。

##### 4.2.2 手动管理的还原段

数据库管理员负责创建非系统还原段 (回滚段), 所有非系统表空间中对象的变化都将使用这样的非系统还原段。

`oracle` 系统中包括了如下两类非系统还原段:

私有还原段: 为一个实例所用。

公有还原段: 为任何一个实例所用。

#### 4.3 延迟还原段

当一个表空间被设置为脱机时, 如果需要, 有 `oracle` 系统自动创建。

延迟还原段被用来在该表空间重新设为联机时回滚事务。当不需要他们时, 系统会自动将他们删除。

### 5. 自动还原数据管理的概念和配置

还原信息分为三种可能:

未提交还原信息: 属于当前事务, 在重做段中不能被覆盖。

已提交还原信息: 事务已经结束, 但仍在还原保持时间范围内, 一般不能被覆盖。

过期提交还原信息: 事务已经结束, 在还原保持时间范围外, 可以被覆盖。

如果要 `oracle` 自动管理还原数据, 数据库管理员就必须在初始化参数文件中配置一下两个参数

`UNDO_MANAGEMENT=AUTO`: 说明系统是使用自动 (`auto`) 还是手动 (`manual`) 模式。

`UNDO_TABLESPACE=UNDOTBS1`: 说明系统使用哪个还原表空间。

`UNDO_MANAGEMENT` 不是动态参数, 不能使用 `ALTER SYSTEM SET` 命令修改。但是 `UNDO_TABLESPACE` 是动态参数, 可以使用 `ALTER SYSTEM SET` 命令修改。

`ALTER SYSTEM SET UNDO_TABLESPACE=UNDOTBS1;`

#### 5.1 还原表空间的创建和维护

有两种方法创建还原表空间:

1) 通过在 `CREATE DATABASE` 命令中加入一个子句, 在创建数据库时建立还原表空间。

2) 在创建数据库之后, 使用 `CREATE UNDO TABLESPACE` 命令来建立还原表空间。

## 5.2 用 CREATE UNDO TABLESPACE 方式创建还原表空间

```
CREATE UNDO TABLESPACE UNDOTBS2 DATAFILE '/tmp/tbs2.dbf' SIZE 20M;
```

查看系统中已经存在的还原表空间。

```
SELECT TABLESPACE_NAME,STATUS,CONTENTS FROM DBA_TABLESPACES WHERE  
CONTENTS='UNDO';
```

## 5.3 动态修改还原表空间的配置

可以使用 ALTER TABLESPACE 命令来修改。oracle 允许使用如下子句：

```
RENAME  
ADD DATAFILE  
DATAFILE [ONLINE|OFFLINE]  
BEGIN BACKUP  
END BACKUP;
```

### 为还原表空间扩容

如果增删改的操作非常多,就要频繁的用到还原表空间.还原表空间的大小最好尽可能大.一般 10G 左右比较合适.(这只是建议值,需要根据不同的环境再定制)

```
ALTER TABLESPACE UNDOTBS2 ADD DATAFILE '/tmp/tbs21.dbf' SIZE 200M;
```

把表空间中的数据设置成自动扩展大小

```
ALTER DATABASE DATAFILE '/tmp/tbs2.dbf' AUTOEXTEND ON;
```

扩大还原表空间数据文件的大小

```
ALTER DATABASE DATAFILE '/tmp/tbs2.dbf' RESIZE 1000M;
```

## 5.4 还原表空间之间的切换

```
ALTER SYSTEM SET UNDO_TABLESPACE = UNDOTBS2;
```

## 5.5 删除还原表空间

```
DROP TABLESPACE UNDOTBS2;
```

```
DROP TABLESPACE UNDOTBS2 INCLUDING CONTENTS AND DATAFILES;
```

## 5.6 自动还原数据管理的一些参数。

动态参数 UNDO\_SUPPRESS\_ERRORS 保证在还原数据管理的自动模式中试图执行手动操作时将不显示错误信息。修改该参数的值：

```
ALTER SESSION SET UNDO_SUPPRESS_ERRORS = TRUE
```

动态参数 UNDO\_RETENTION 决定为了保证读一致行还原数据所保留的时间，单位为秒。900 秒就是 15 分钟。

```
ALTER SYSTEM SET UNDO_RETENTION = 900;
```

动态参数 UNDO\_RETENTION 虽然能在保证长查询语句的读一致性方面有所改进，但是加大 UNDO\_RETENTION 的也将会增加数据库的磁盘空间消耗，特别是 DML 操作很频繁时。

## 5.7 获得还原数据的信息

```
SQL> SELECT TO_CHAR(BEGIN_TIME, 'HH:MM:SS')  
BEGIN_TIME, TO_CHAR(END_TIME, 'HH:MM:SS') END_TIME, UNDOBLKS, TXNCOUNT, MAXQUERYLEN  
FROM V$UNDOSTAT;
```

BEGIN_TI	END_TIME	UNDOBLKS	TXNCOUNT	MAXQUERYLEN
02:11:15	02:11:53	11	17	0
01:11:15	02:11:15	17	74	1548
01:11:15	01:11:15	41	388	948
01:11:15	01:11:15	543	1179	347

视图 v\$undostat 显示系统还原数据的统计信息。oracle 实例每 10 分钟收集一次统计信息并存在该视图的某一行中。其中

BEGIN\_TIME: 为日期型，标识时间间隔的开始。

END\_TIME: 为日期型，表示时间间隔的结束。

UNDOBLKS: 为数据型，标识所消耗的还原数据块的总和。

TXNCOUNT: 为数据型，标识该时间段中事物的总和。

MAXQUERYLEN: 为数据型，标识该时间段中所执行的最长的查询（按描述）；

## 5.8 其他有用的视图

DBA\_ROLLBACK\_SEGS  
V\$ROLLNAME  
V\$ROLLSTAT  
V\$SESSION  
V\$TRANSACTION

## 6. 闪回

是 DML 操作提交后回滚段的数据不释放, 缺省留 900 秒, 详见 UNDO\_RETENTION 的值. 保留时间从 COMMIT 操作以后开始计算.

### 6.1 闪回表内容

查找误操作的时间. 一般情况下当发生误操作的时候这个时间是自己估的. 所以在真是环境中此步骤根本没有.

```
SQL> SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS') FROM DUAL;
```

删除表的内容

```
SQL> DELETE FROM EMP;  
SQL> COMMIT;
```

允许表迁移. 行迁移是表的数据从一个数据块挪到另一个数据块.

```
SQL> ALTER TABLE EMP ENABLE ROW MOVEMENT;
```

执行闪回

```
SQL> FLASHBACK TABLE EMP TO TIMESTAMP(TO_DATE('2012-08-12 12:15:02', 'YYYY-MM-DD  
HH24:MI:SS'))
```

查询数据是否已经被恢复.

```
SQL> SELECT * FROM EMP;
```

### 6.2 闪回表

查询当前用户环境下都有哪些表

```
SQL> SELECT * FROM TAB;
```

删除表

```
SQL> DROP TABLE EMP;
```

查询表的状态

```
SQL> SELECT * FROM TAB;
```

查看回收站中的信息

```
SQL> SHOW RECYCLEBIN
```

可以从回收站中查看被删除的表的内容

```
SQL> SELECT * FROM "BINXXXXXX";
```

回滚表

```
SQL> FLASHBACK TABLE EMP TO BEFORE DROP;
```

如果想永久删除表

```
SQL> DROP TABLE EMP PURGE;
```

## 7. "快照太旧"差错

快照太旧是要查询的数据不存在了. 在还原表空间特小并且设定成自动扩展非常容易引起"快照太旧"的问题. 解决方法就是把还原表空间改大.

## 8. 从 oracle10g 开始有一个新特性就是: 还原保持得到"保证"

在设置的 900 秒不到, 表空间的空间不够. 把不到 900 秒的数据覆盖.

查看表空间是否为保证状态

```
SQL> SELECT TABLESPACE_NAME,RETENTION FROM DBA_TABLESPACES;
```

设定表空间为保证状态。

```
SQL> ALTER TABLESPACE UNDOTBS1 RETENTION GUARANTEE;
```

设定表空间为不保证状态。

```
SQL> ALTER TABLESPACE UNDOTBS1 RETENTION NOGUARANTEE;
```