

Oracle数据库中为什么会产生回滚与前退Oracle认证考试 PDF  
转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/637/2021\\_2022\\_Oracle\\_E6\\_95\\_B0\\_E6\\_c102\\_637996.htm](https://www.100test.com/kao_ti2020/637/2021_2022_Oracle_E6_95_B0_E6_c102_637996.htm) Oracle概念问题，假如数据没有提交

，但是却被dbwn进程写入了数据文件，会怎么样呢？案例分析：首先说明的是dbwn写脏数据跟commit提交没有关系！

在一个transaction发生的过程中，online redo log首先记录transaction中修改的数据块相关信息，修改的数据块会被缓存在database buffer cache中。由于database buffer cache写满或者checkpoint等等条件触发dbwn进程，会导致这些缓存的数据块写入数据文件，但此时可能该transaction仍然还没有提交。所以在数据文件中，可能会有committed 和 uncommitted 的数据块。而原有的数据块镜像会存放在undo segment。然而

，dbwn写脏数据时不管这个要写的transaction是否提交，也没有必要去管。这样就发生了所谓的已经提交的数据，但是还没有写入数据文件的现象。还有一种情况，数据没有提交，但是已经被写入数据文件，此时发生回退，撤销没有提交的数据。那么，引发Oracle前滚与回退的根本原因就是什么呢

？根本原因是commit后写redo buffer和触发lgwr写 redo buffer的区别。事务在执行完毕后，随即会被写入redo buffer和undo中，同时在redo buffer和undo中对该事务都有一个是否提交的标记。两者的默认状态都是active的，即没有提交时刻处于激活状态。commit操作执行时刻把此前的所有事务操作全部写入redo log file，commit成功后，redo buffer信息全部写入redo file，同时修改两者中的事务提交标识为inactive，表示此前事务已经递交。oracle的前滚和回退根据就是依据事务是否提交

而进行的。在触发lgwr进程后，oracle同样把此前的redo buffer信息写入redo file，但是与commit触发写日志不同的是，redo file本身对lgwr写日志操作不记录任何信息标识，lgwr写到哪里就是哪里，就算此时掉电也无妨，redo file就记录到掉电时刻的信息。lgwr是一个Oracle后台执行的进程，具体的日志写操作都有oracle去控制，这对于oracle来说是透明的，因此不用在redo file中写入任何标记信息，这也是正常的。commit操作是唯一一个可以前台操作与oracle后台通信的指令，因此当加入这个操作以后，oracle本身必须要了解各个事务的读写状况，那么怎么了解整个状况：在redo以及undo中加入是否递交的标识，对于已经提交的操作，但是还没有写入数据文件，那么就要前滚，相反，对于没有提交，执行回退！于是，Oracle崩溃恢复步骤如下：

首先rolling forward 前滚：由于oracle failure，sga中的内存信息丢失了，但是online redo log中还是存储了transaction信息，包括committed or uncommitted data。可能这些修改信息并没有被oracle正确的来处理，包含两种情况：已经提交的还没有写入数据文件，或者没有提交的却被写入了数据文件。针对已经提交的还没有写入数据文件就要发生前滚，在前滚过程中，smon会根据online redo log中的记录来完成对datafile的修改。保证已经提交的数据已经写入数据文件。接下来，前滚结束后，数据库正常open，此时用户可以正常连接，可以访问已经recover的committed data，但是对于那些属于unrecoverable transaction的uncommitted data，会被oracle加锁，是不可以访问的。

rolling back：假如有进程访问这些加锁的数据，此时smon会对这些数据块做rollback回滚，从数据文件中撤销没有提交却被写入数据文

件的数据。 100Test 下载频道开通，各类考试题目直接下载。  
详细请访问 [www.100test.com](http://www.100test.com)