

oracle认证:数据库中日志问题详解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_ti2020/565/2021_2022_oracle_E8_A

E_A4_E8_c102_565771.htm 日志是日志记录的一个序列。在多事务的数据库系统中，每个事务有若干个操作步骤。每个日志记录记载有关某个事务已做的某些情况。几个事务的行为可以是“交错的”，因此可能是一个事务的某个步骤被执行，并且其效果被记录到日志中，接着执行另外一个事务的某个步骤并记入日志，接着可能接着做第一事务的下一个步骤，也可能执行另外一个事务的某个步骤。依次类推。事务的交错执行使日志更复杂，因为仅在事务结束后记载事务的全过程是不够的。如果系统崩溃，恢复管理器就被激活，检查日志以重建数据库的一致性状态。恢复时，有些事务的工作将会重做，它们写到数据库的新值会重写一次。而另外一些事务的工作将被撤消，数据库被恢复，将仿佛这些事务从来没执行过一样。Undo日志是日志类型的一种，这类日志仅仅进行第二类修复。对于要被撤消的事务，因为不能肯定它对数据库的修改是否已经写到磁盘中，所以对于该事务的所有更新都将被撤消，数据库恢复到事务发生以前的状态。

1.日志记录 日志只允许以附加的方式写入数据。日志块最初在主存中创建，像数据块一样也由缓冲区管理，在确当的时刻，日志块会从缓冲区写入到磁盘。关于undo记录形式有四种：

1): 这一记录表示事务T开始 2): 事务T已经完成。 3): 事务T不能成功执行。 4): 事务T改变了数据库元素X的值，元素X原来的值为v。

2.undo日志规则 要想让undo日志能使我们从系统故障中恢复，事务必须遵循两条规则。规则1)如果事务T

改变了数据库元素X，那么形如的日志记录必须在X的新值写到磁盘前写到磁盘 规则 2)如果事务提交，则其COMMIT日志记录必须在事务改变的所有数据库元素已写到磁盘后再写到磁盘，但应尽快。 简单概括，undo日志系统顺序如下： 1) 指明所改变数据库元素的日志记录 2) 改变的数据库元素自身 3) COMMIT日志记录。 3.应用 对于前面所举的例子(A转帐50元给B帐号)，如果使用了undo日志系统，则相关的工作流程如表2如下。 我们可以看到FLUSH LOG这个命令。该命令的用于强制将还没有刷盘日志记录写到磁盘上。对于步骤8执行之前，三个undo记录(,,)是存储在缓冲区中的(这样描述是为将问题简单化)，执行步骤8之后，三条undo记录便写入了磁盘日志文件。在步骤12再次执行FLUSH LOG命令时，只将未刷盘的日志记录写入磁盘。 步骤9、10，在前面已经描述过，是将数据库元素的修改从缓冲区写入到磁盘文件，因为WRITE操作仅仅是将修改反应到缓冲区中(这样描述也是为了将问题简单化)。 关注步骤8，执行完步骤8之后，将满足undo的规则1(如果事务T改变了数据库元素X，那么形如的日志记录必须在X的新值写到磁盘前写到磁盘)。在我们真正将数据库元素A与B的修改反应到磁盘前，我们已经将它们对应的与写入到磁盘日志文件。 关注步骤11与步骤12，执行它们将满足undo的规则2(如果事务提交，则其COMMIT日志记录必须在事务改变的所有数据库元素已写到磁盘后再写到磁盘，但应尽快)。我们已经将数据库元素A与B的修改反应到磁盘上，接着可以写入来表示事务T的成功执行，但是该日志记录还在缓冲区中，所以我们在步骤12执行FLUSH LOG进行日志刷盘。 4.使用undo日志进行数据库的恢复 现在假设系统故障发生

了。有可能给定事务的某些数据库更新已经写到磁盘上，而同一事务的另外一些更新尚未到达磁盘。如果这样，事务的执行就不是原子的，数据库状态就可能不一致。这时候，我们就有必要使用日志将数据库恢复到一致的状态。恢复管理的第一个任务就是将事务划分为已经提交事务和未提交事务。如果在日志中，根据undo的规则2，事务T所做的全部改变在此之前已经写到磁盘上，因此当故障发生时，该事务T不可能导致数据库的不一致状态。然而，假设在日志中，只有记录，而没有与之相匹配的记录。那么就有可能在崩溃前，事务的某些修改已经反应到磁盘上，而另外一些修改可能未发生或者还在缓冲区中。这种情况下，T是一个未完成的事务，因为必须被撤消。也就是说，T所做的任何修改都必须恢复为原来的值。Undo的规则1使该想法可以成为可能，因为在修改数据刷盘之前，日志文件中已经保存了修改数据的原先值。对于，只需要将X的值恢复为v就行了(我们不必检查数据库中X现有值是否为v)。日志中可能有一些未提交的事务，并且甚至可能有一些未提交的事务修改了X，所以恢复时采用从日志文件尾向前扫描。在扫描过程中记住所有有或记录的事务T。同时在随后的扫描中，如果它看见，则：1) 如果T的COMMIT记录已被扫描到，则什么也不做。2) 否则，将数据库中元素X的值改为v。在做完这些操作后，为以前未中止且未完成的每个事务T写入一个日志记录，然后刷新日志。对于表2，作如下分析：1)崩溃在第12步后发生。因为日志记录已经写入日志文件。当恢复时，不需要处理T事务。2)崩溃发生在11步和12步之间。日志中记录了三条记录：、以及。当恢复管理进行向后扫描时，首先遇到记录，于是它将B在

磁盘上的值存为500。接着它遇到记录，于是它将A在磁盘上的值存为1000。最后记录被写到日志中且日志被刷新。 3)崩溃发生在第8步和第11步之间。情况2一样。 4)崩溃发生在第8步之前。日志系统中只有一条记录：。记录被写到日志中且日志被刷新。(实际上，在实际系统中，可能其他的事务执行FLUSH LOG操作，而将事务T的日志记录写入磁盘中，如果是这样的话，日志中可能已经写入和，对于这种情况，执行情况2的恢复就行。实际上这些细节不影响undo的恢复效用，为了简单起见，忽视这些情况)。

5.静态检查点 正如我们所看到的那样，恢复时需要检查整个日志。当采用undo类型的日志时，一旦日志记录被写入日志文件，事务T的日志记录就可以忽视了。但是此时事务T却不能截断日志，因为事务是交替执行的，如果这时将日志截断，可能丢失活动着的事务的日志记录 解决该问题的方法是周期性地对日志做检查点。步骤如下： 1)停止接受新的事务 2)等待所有当前的活动事务提交或终止，并且在日志文件中写COMMIT或ABORT记录。 3)将日志刷新到磁盘 4)写入日志记录，并再次刷新记录。 5)重新开始接受新事务。

当恢复事，扫描到日志时，就不需要继续扫描日志记录了。例如：假设日志开始时是这样的：这时候，我们做一个检查点。等待事务T1和事务T2都完成，才在日志文件中写记录。检查点做完后，可以接受新的事务，这里为事务T3，T3执行了一些操作，此时系统崩溃了。恢复时，从文件尾部开始扫描，因为T3是未完成事务，则将磁盘上F的值存为前值30、磁盘上E的值存为前值25。当扫描到日志记录时，我们知道没有必要检查以前的日志记录了，并且数据库状态的恢复已经完成。 更多优质资料尽在百考试题论

坛 百考试题在线题库 oracle/认证更多详细资料 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