

三级数据库第九章考试要点 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/137/2021\\_2022\\_\\_E4\\_B8\\_89\\_E7\\_BA\\_A7\\_E6\\_95\\_B0\\_E6\\_c98\\_137735.htm](https://www.100test.com/kao_ti2020/137/2021_2022__E4_B8_89_E7_BA_A7_E6_95_B0_E6_c98_137735.htm) 第九章 一、事务概念和事务的特性 事务是构成单一逻辑工作单元的操作集合。不论有无故障，数据库系统必须保证事务的正确执行，即要么执行该事务的整个操作集合，要么属于该事务的操作一个也不执行。为了保证事务的正确执行，维护数据库的完整性，我们要求数据库系统维护以下事务特性：原子性（atomicity）事务的所有操作在数据库中要么全部正确反映出来，要么全部不反映。一致性（consistency）从事务的隔离执行（即没有并发执行的其他事务）来保持数据库的一致性。隔离性（isolation）尽管多个事务可以并发执行，但系统必须保证，对任一事务 $T_i$ 和 $T_j$ ，在 $T_i$ 看来 $T_j$ 或者在 $T_i$ 开始之前已经停止执行，或者在 $T_i$ 完成之后开始执行。这样，每个事务都感觉不到系统中有其他事务在并发地执行。持久性（durability）一个事务成功完成后，它对数据库的改变必须是永久的，即使系统可能出现故障。这些特性通常被称为ACID特性。解决事务并发执行问题的一种方法是串行地执行事务，即一个接一个地执行。但这样性能较低，事务并发执行能显著地改善性能。因此，数据库管理系统中有一个并发控制部件，它保证事务隔离性，即确保事务并发执行后的系统状态与这些事务以某种次序一个接一个地执行后的状态是等价的。成功完成执行的事务称为已提交事务。已提交事务使数据库进入一个新的一致状态，即使出现系统故障，这个状态也必须保持。未能成功完成的事务称为中止事务，中止事务必须对数

数据库的状态不造成影响，即中止事务对数据库所做的任何改变必须撤销。一旦中止事务造成的变更被撤销，则称事务已回滚。

## 二、故障恢复

### (一) 故障的类型

#### (1) 事务故障

用两种错误可能造成事务执行失败：逻辑错误。系统错误。事务故障意味着事务没有达到预期的终点，因此，数据库可能处于不一致状态。恢复子系统需要强行回滚该事务，即撤销该事务对数据库已做的所有的修改，这称做事务的撤销或回滚。

#### (2) 系统故障

硬件故障或者是数据库软件或操作系统的漏洞，导致系统停止运行。主存储器内容丢失，而外存储器仍完好无损。发生系统故障时，可能有些未完成事务所做的数据库修改已经写入到外存的数据库中，因此，数据库可能处于不一致状态。为保证一致性，在系统重新启动时必须强行回滚由于系统故障而非正常终止的事务（UNDO）。另一方面，发生系统故障时，可能有些已完成事务所做的数据库修改仍部分或全部留在系统缓冲区，尚未写入到外存的数据库中，这也会使数据库处于不一致状态。在系统重新启动时恢复子系统必须重做（REDO）所有已提交的事务，以保证事务的持久性和数据库的一致性。

#### (3) 磁盘故障

在数据传送操作过程中由于磁头损坏或故障造成磁盘块上的内容丢失。这类故障比前两类故障发生的可能性小，但危害大。发生磁盘故障时，可以利用其他磁盘上的数据拷贝，或三级介质（如磁带）上的备份来进行恢复。

### (二) 基于日志的恢复

保证在故障发生后仍保持数据库一致性以及事务的原子性的算法称为恢复算法，它由两部分组成：

#### (1) 在正常事务处理时采取措施，记录数据库中的更新活动，保证有足够的信息可用于故障恢复。

#### (2) 故障发生后采取措施，将数据库内容恢

复到某个保证数据库一致性、事务原子性及持久性的状态。使用最为广泛的记录数据库中更新活动的结构是日志。日志是日志记录的序列，它记录了数据库中的所有更新活动。1. 日志的登记日志文件在数据库恢复中起着非常重要的作用，它能够很好地支持事务故障的恢复和系统故障的恢复。利用更新日志记录中的改前值可以进行UNDO，撤销已做的修改操作，将数据项恢复到修改以前的旧值.利用更新日志记录中的改后值可以进行REDO，重做已完成的操作，将数据项置为修改后的新值。登记日志记录时必须遵循以下原则:登记的顺序严格按照事务的并发执行中各操作发生的实际顺序.必须先把日志记录写到外存的日志文件中，再把相应的数据库修改写到外存的数据库中。这称做先写日志的原则，是为了发生故障后保持数据库的原子性和持久性所必须遵循的原则。

发生故障后的恢复（1）事务故障恢复的步骤 反向扫描日志文件，查找该事务的更新操作。 对该事务的每一个更新操作执行逆操作，即将日志记录中“改前值”写入数据库。于是，若日志记录对应的是插入操作，则这时做删除操作.若日志记录对应的是删除操作，则这时做插入操作.若日志记录对应的是修改操作，则这时用修改前的值代替修改后的值。

如此处理下去，直至读到该事务的开始日志，则事务故障恢复结束。（2）系统故障恢复的步骤 正向扫描日志文件，找出在故障发生前已提交的事务，将其事务标识记入REDO队列.找出在故障发生时尚未完成的事务，将其事务标识记入UNDO队列。 对UNDO队列中的事务进行UNDO处理，即反向扫描日志文件，对每一个需要UNDO的事务的更新操作执行逆操作。 对RREDO队列中的事务进行REDO

处理，即正向扫描日志文件，对每一个需要REDO的事务的更新重新执行日志记录登记的操作。磁盘的恢复需要有DBA介入，装入最新的数据库后备副本，和有关的日志文件副本，然后由系统进行恢复工作。

### 三、并发控制

#### (一) 事务的并发执行

丢失更新 对未提交更新的依赖 不一致的分析

#### (二) 并发事务的调度

这两个调度都是串行的。串行调度由来自各事务的指令序列组成，其中属于同一事务的指令在调度中紧挨在一起。当多个事务并发执行时，相应的调度不一定是串行的。若有两个并发执行的事务，操作系统可能先选其中的一个事务执行一小段时间，然后切换上下文，执行第二个事务一段时间，接着又切换到第一个事务执行一段时间，如此下去。执行顺序可能会有多种，因为来自两个事务的各条指令可以交\*执行。如果并发执行的控制完全由操作系统负责，许多调度都是可能的，包括像上述调度那样使数据库处于不一致状态的调度。保证任何调度执行后数据库总处于一致状态是数据库系统的职责。数据库系统中完成此任务的部件是并发控制部件。

#### (三) 可串行化

直观地说，如果多个事务在某个调度下的执行结果与这些事务在某个串行调度下的执行结果相同，则称这个调度为可串行化的调度。下面用等价的概念来定义可串行化的概念。如果某个调度等价于一个串行调度，则称这个调度是可串行化的。并发调度3等价于串行调度1，于是称调度3是一个可串行化的调度。可串行化是多个事务并发执行的正确性准则。多个事务在某个调度下的执行是正确的，是能保证数据库一致性，当且仅当该调度是可串行化的。

#### (四) 可恢复性

因一个事务故障导致一系列事务回滚的现象称为级联回滚。由于级联回滚导致撤销大量工

作，因此不希望发生级联回滚。我们希望对调度加以限制，避免级联回滚发生，这样的调度称为无级联调度。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)