**数据库作业 第10章**

**22920212204392 黄勖**

**1. 试述事务的概念及事务的四个特性，恢复技术能保证事务那些特性？**

答：

**概念**：事务是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。

四个特性：（ACID）

**原子性**：事务是数据库的逻辑工作单位，事务中包括的诸操作要么都做，要么都不做

**一致性**：事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态

**隔离性**：一个事务的执行不能被其他事务干扰，即一个事务的内部操作及使用的数据对其他并发事务是隔离的，并发执行的各个事务之间不能互相干扰

**持久性**：一个事务一旦提交，它对数据库中数据的改变就是永久性的，接下来的其他操作或故障不应该对其执行结果有任何影响.

能保证事务的原子性，一致性

1. **为什么事务非正常结束的时候会影响数据库数据的正确性？请举例说明。**

答：事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。如果数据库系统运行中发生故障,有些事务尚未完成就被迫中断,这些未完成事务对数据库所做的修改有一部分已写入物理数据库,这时数据库就处于不一致的状态，就不正确了。

下面是一个例子，以说明事务非正常结束可能会影响数据库数据的正确性：

假设有一个银行数据库，包含两个表：账户表（Account）和交易记录表（Transaction）。账户表中存储了用户的账户信息，交易记录表中存储了用户的交易历史。

考虑以下两个操作作为一个事务：

1. 将账户A的余额减少100元。
2. 将账户B的余额增加100元。

正常情况下，这两个操作应该是作为一个事务执行的，以保证数据库的一致性。如果事务正常执行，账户A的余额减少100元，账户B的余额增加100元，数据库的状态保持一致。

然而，如果事务非正常结束（例如，系统崩溃或电力故障），可能会导致部分操作生效而部分操作未生效。假设在执行第一个操作后，系统发生了故障，事务未能完成。此时，账户A的余额已经减少100元，但账户B的余额仍然未被更新。

这种情况下，数据库的状态就不再一致。账户A的余额减少了，但账户B的余额并没有增加。这违反了数据库的一致性，导致数据的正确性受到了影响。

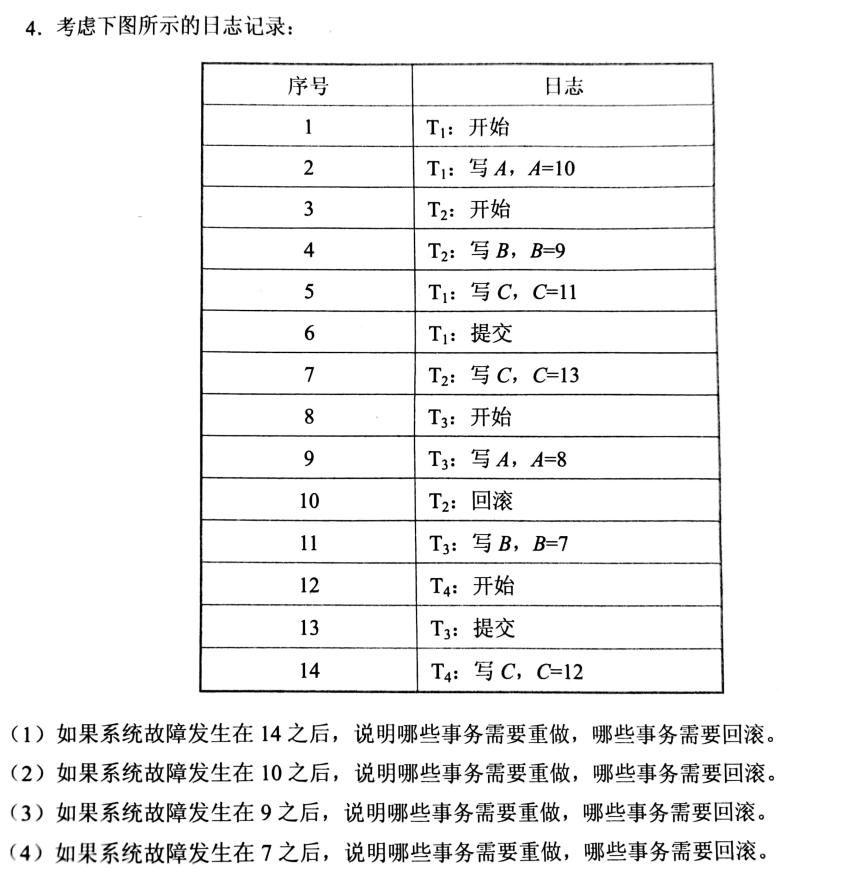
正常情况下，当事务非正常结束时，数据库管理系统会使用一些机制（例如日志记录和恢复操作）来保证数据的正确性和一致性。但在某些特殊情况下，如硬件故障或系统错误，这些机制可能无法完全防止数据不一致性的问题。因此，事务的正常结束对于确保数据库数据的正确性非常重要。

1. **登记日志文件为什么要先写入文件，后写入数据库？**

答：把对数据的修改写到数据库中和把表示这个修改的日志记录写到日志文件中是两个不同的操作。有可能在这两个操作之间发生故障，即这两个写操作只完成了一个。

如果先写了数据库修改，而在运行记录中没有登记这个修改，则以后就无法恢复这个修改。如果先写日志，但没有修改数据库，按日志文件恢复时只不过多执行一次不必要的UNDO操作，并不会影响数据库的正确性。

综上所述，一定要先写日志文件,即首先把日志记录写到日志文件中,然后写数据库的修改。



（1）T1，T3需要重做，T2，T4需要回滚

（2）T1需要重做，T2，T3需要回滚

（3）T1需要重做，T2，T3需要回滚

（4）T1需要重做，T2需要回滚

1. **考虑第四题所示的日志，假设在开始的时候ABCD都是0：**
2. 如果系统故障发生在14之后，请写出恢复后ABC的值

A=8，B=7，C=11

1. 如果系统故障发生在12之后，请写出恢复后ABC的值

A=10，B=0，C=11

1. 如果系统故障发生在10之后，请写出恢复后ABC的值

A=10，B=0，C=11

1. 如果系统故障发生在9之后，请写出恢复后ABC的值

A=10，B=0，C=11

1. 如果系统故障发生在7之后，请写出恢复后ABC的值

A=10，B=0，C=11

1. 如果系统故障发生在5之后，请写出恢复后ABC的值

A=0，B=0，C=0

1. **针对不同的故障，试着给出恢复的策略和方法。（如何进行事务故障的恢复，如何进行系统故障的恢复，如何进行介质故障的恢复）**

答：

**事务故障的恢复**

事务故障是指事务中运行至正常终点前被终止。恢复策略是由恢复子系统利用日志文件撤消（UNDO）此事务已对数据库进行的修改。事务故障的恢复由系统自动完成，对用户透明。

**事务故障的恢复步骤：**

①反向扫描文件日志,查找该事务的更新操作。

②对该事务的更新操作执行逆操作。即将日志记录中“更新前的值”写人数据库。直至读到此事务的开始标记,该事务故障的恢复就完成了。

**系统故障的恢复：**

系统故障造成数据库不一致状态的原因有未完成事务对数据库的更新可能已写入数据库，已提交事务对数据库的更新可能还留在缓冲区没来得及写入数据库。恢复策略是撤消（UNDO）故障发生时未完成的事务，重做（REDO）已完成的事务。系统故障的恢复由系统在重新启动时自动完成，对用户透明。

**系统故障的恢复步骤：**

①正向扫描日志文件,找出在故障发生前已经提交的事务队列(REDO队列)和未完成的事务队列(UNDO队列)。

②对未完成的事务队列中的各个事务进行UNDO处理。3对已经提交的事务队列中的各个事务进行REDO处理。

**介质故障的恢复：**

发生介质故障后，磁盘上的物理数据和日志文件被破坏，这是最严重的一种故障。

恢复策略是重装数据库，然后重做已完成的事务。介质故障的恢复需要DBA介入。DBA只需要重装最近转储的数据库副本和有关的各日志文件副本，然后执行系统提供恢复命令即可，具体的恢复操作仍由DBMS完成。

**介质故障的恢复步骤：**

①装入最新的数据库后备副本(离故障发生时刻最近的转储副本),使数据库恢复到最近一次转储时的一致性状态。

②装入转储结束时刻的日志文件副本。

③启动系统恢复命令,由DBMS完成恢复功能,即重做已完成的事务。

1. **什么是检查点记录，检查点记录包括哪些内容？**

答：检查点记录是一类新的日志记录。内容包括：建立检查点时刻所有正在执行的事务清单、这些事务最近一个日志记录的地址。

1. **具有检查点的恢复技术有什么优点？试着举一个例子加以说明。**

答：具有检查点的恢复技术是一种用于数据库管理系统的故障恢复方法，它通过在事务执行过程中创建检查点（checkpoint），在系统故障发生时能够从检查点处快速恢复，具有以下优点：

1. 快速恢复：具有检查点的恢复技术可以在系统故障后快速恢复到事务执行的某个检查点，而不需要从头重新执行所有操作。这可以大大缩短恢复时间，减少系统不可用的时间窗口。
2. 减少数据丢失：通过定期创建检查点，数据库可以将事务执行过程中的数据变更写入到稳定的存储介质（如磁盘），即使发生故障，也可以从最近的检查点处恢复。这样可以减少数据丢失的风险，提高数据的可靠性。
3. 降低恢复成本：相比于完全的日志回滚恢复技术，具有检查点的恢复技术可以减少恢复时需要读取和处理的日志量。这可以降低恢复操作的成本，提高系统的恢复效率。

例子： 假设有一个电子商务网站的数据库，其中包含订单表（Order）和库存表（Inventory）。在某一天的交易高峰期，数据库管理系统定期创建检查点。

在交易过程中，顾客下了一笔订单并支付成功。此时，数据库会将订单数据写入订单表，并减少库存表中对应商品的库存量。然而，正当订单数据写入订单表之后，系统突然发生故障，导致数据库无法继续正常运行。

由于具有检查点的恢复技术，系统管理员可以将系统恢复到最近的检查点。在恢复过程中，数据库管理系统会回滚到最近的检查点处，并重新执行从该检查点开始之后的事务操作。

在这个例子中，具有检查点的恢复技术的优点是：

* 快速恢复：系统管理员可以将数据库迅速恢复到最近的检查点，避免了长时间的系统不可用。
* 减少数据丢失：由于订单数据已经写入订单表，因此在故障发生后的恢复过程中，该订单数据不会丢失。
* 降低恢复成本：由于只需回滚到最近的检查点，而不需要逐个回滚所有事务，恢复操作的成本和时间大大降低。

因此，具有检查点的恢复技术可以有效提高数据库系统的可用性和数据可靠性，并降低故障恢复的成本和时间。

1. **试述用检查点方法进行恢复的步骤。**

**答：**

1.从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中找到最后一个检查点记录

2.由该检查点记录得到检查点建立时刻所有正在执行的事务清单ACTIVE-LIST

UNDO-LIST：需要执行UNDO操作的事务集合

REDO-LIST：需要执行REDO操作的事务集合

3.从检查点开始正向扫描日志文件：

如有新开始的事务Ti，把Ti暂时放入UNDO-LIST队列；

如有提交的事务Tj，把Tj从UNDO-LIST队列移到REDO-LIST队列;

继续以上过程，直到日志文件结束。

4.对UNDO-LIST中的每个事务执行UNDO操作，对REDO-LIST中的每个事务执行REDO操作。

1. **什么是数据库镜像？它有什么用途？**

数据库镜像(Mirror)是解决介质故障、提高数据库可用性的一种常用方法。

即根据DBA的要求，自动把整个数据库或其中的关键数据复制到另一个磁盘上。每当主数据库更新时，DBMS自动把更新后的数据复制过去，由DBMS自动保证镜像数据与主数据库的一致性。

**用途：**①用于数据库恢复。当出现介质故障时,镜像磁盘可继续使用,同时DBMS自动利用镜像磁盘数据进行数据库的恢复,不需要关闭系统和重装数据库副本。

②提高数据库的可用性。在没有出现故障时,当一个用户对某个数据加排他锁进行修改时,其他用户可以读镜像数据库上的数据,而不必等待该用户释放锁。