

课后自由练习 #5 参考解答

1. 什么是事务的 ACID 性质？请给出违背事务 ACID 性质的具体例子，每个性质举一个例子。

原子性 (Atomicity)

事务是不可分的原子，其中的操作要么都做，要么都不做。

反例：转账中 A 余额减去 100 并且写入磁盘，但此时系统崩溃，且未做恢复。

一致性 (Consistency)

事务的执行保证数据库从一个一致状态转到另一个一致状态（满足所有完整性约束）。

反例：举一个违反完整性约束的例子，如上例转账。

隔离性 (Isolation)

多个事务一起执行时相互独立。

反例：同一个事务对同一个数据的两次读的值不同。

持久性 (Durability)

事务一旦成功提交，就在数据库永久保存。

反例1：事务已提交，而改变的数据仍然还在磁盘缓冲区中排队等待写入磁盘中，导致系统重启后 commit 的结果无效。

反例2：磁盘损坏导致数据丢失，且没有备份。

2. 目前许多 DBMS 例如 MySQL 都默认不支持嵌套事务（即在一个事务内部又开始另一个事务），请分析一下：如果 DBMS 支持嵌套事务，将面临哪些问题（至少写出 2 点以上并且要给出自己的分析）？

假设事务A调用事务B

(1) 原子性和持久性矛盾

如果 B commit，但之后 A rollback，此时 A 的原子性与 B 的持久性冲突。

(2) 一致性影响

B 开始的时候可能不处于一致性状态。

(3) 隔离性

A 事务需要先修改一条数据 v 再调用 B，B 事务也需要修改数据 v。如果系统采用了锁机制，则会陷入死锁。否则如果 B 读的是事务 A 修改后的数据，则发生了脏读；如果读的是修改前的数据，则发生了脏写，两种情况均破坏了隔离性。

(4) 死循环

A、B 相互调用，可能陷入死循环。

3.

3. 下面是一个数据库系统开始运行后的日志记录，该数据库系统支持检查点。

- 1) <T1, Begin Transaction>
- 2) <T1, A, 49, 20>
- 3) <T2, Begin Transaction>
- 4) <T1, B, 250, 20>
- 5) <T1, A, 75, 49>
- 6) <T2, C, 35, 20>
- 7) <T2, D, 45, 20>
- 8) <T1, Commit Transaction>
- 9) <T3, Begin Transaction>
- 10) <T3, E, 55, 20>
- 11) <T2, D, 46, 45>
- 12) <T2, C, 65, 35> ①
- 13) <T2, Commit Transaction>
- 14) <T3, Commit Transaction>
- 15) <CHECKPOINT> ②
- 16) <T4, Begin Transaction>
- 17) <T4, F, 100, 20>
- 18) <T4, G, 111, 20>
- 19) <T4, F, 150, 100>
- 20) <T4, Commit Transaction> ③

设日志修改记录的格式为 <Tid, Variable, New value, Old value>，请给出对于题中所示①、②、③三种故障情形下，数据库系统恢复的过程以及数据元素 A, B, C, D, E, F 和 G 在执行了恢复过程后的值。

① T1: redo

T2、T3: undo

先undo: D=45, E=20, D=20, C=20

再redo: A=49, B=250, A=75

所以, A=75, B=250, C=20, D=20, E=20, F=20, G=20

② T1、T2、T3: redo

redo: A=49, B=250, A=75, C=35, D=45, E=55, D=46, C=65

所以, A=75, B=250, C=65, D=46, E=55, F=20, G=20

③ 检查点前: A=75, B=250, C=65, D=46, E=55, F=20, G=20

检查点后:

T4: undo

undo: F=100, G=20, F=20

所以, A=75, B=250, C=65, D=46, E=55, F=20, G=20