

数据库第3次作业

191220163 计算机科学与技术系 张木子苗

1. 以下是两个事务写入的 undo/redo-日志记录序列 T and U:

```
1 | <START U>
2 | <U, A, 10, 11>
3 | <START T>
4 | <T, B, 20, 21>
5 | <U, C, 30, 31>
6 | <T, D, 40, 41>
7 | <COMMIT T>
8 | <U, E, 50, 51>
9 | <COMMIT U>
```

描述恢复管理器的操作，包括对磁盘和日志的更改，如果发生崩溃并且出现在磁盘上的最后一条日志记录是：

(a) <START T>

在 <START T> 处，尚未有事务被提交，因此我们需要：

- (1) 逆向扫描，撤销事务T和U的操作，包括在磁盘上把A从11恢复到10；
- (2) 正向扫描，do nothing；
- (3) 在日志文件的尾部插入U和T的结束标志：

```
1 | <Abort U>
2 | <Abort T>
```

(b) <COMMIT T>

在 <COMMIT T> 处，事务T已经被提交，而事务U尚未被提交。所以我们需要：

- (1) 逆向扫描，撤销事务U的操作，包括在磁盘上把C从31恢复到30，把A从11恢复到10；
- (2) 正向扫描，重做事务T的操作，包括在磁盘上把21写入B，把41写入D；
- (3) 在日志文件的尾部插入U的结束标志：

```
1 | <Abort U>
```

(c) <U, E, 50, 51>

在 <U, E, 50, 51>处，事务T已经被提交，而事务U尚未被提交。所以我们需要：

- (1) 逆向扫描，撤销事务U的操作，包括在磁盘上把E从51恢复到50，把C从31恢复到30，把A从11恢复到10；
- (2) 正向扫描，重做事务T的操作，包括在磁盘上把21写入B，把41写入D；
- (3) 在日志文件的尾部插入U的结束标志：

```
1 | <Abort U>
```

(d) <COMMIT U>

在 <COMMIT T> 处，事务T和U都被提交，所以我们需要：

- (1) 逆向扫描，do nothing;
- (2) 正向扫描，重做事务T和事务U的操作，包括在磁盘上把11写入A，把21写入B，把31写入C，把41写入D，把51写入E;
- (3) 不需要在日志文件的尾部插入结束标志。

2. 请考虑下面两个事务：

1 T1: read(A); 2 read(B); 3 if A = 0 then B := B + 1; 4 write B;	T2: read(B); read(A); if B = 0 then A := A + 1; write A.
---	---

请给事务 T1 与 T2 增加封锁和解锁指令，使它们遵从两阶段封锁协议。这两个事务的执行会导致死锁吗？

(1) 增加封锁和解锁指令后如下：

$sl_i(A)$: 事务 T_i 申请数据对象A上的一个‘S锁’

$xl_i(A)$: 事务 T_i 申请数据对象A上的一个‘X锁’

$u_i(A)$: 事务 T_i 释放自己在数据对象A上所持有的锁

1 T1: $sl_1(A)$; 2 read(A); 3 $xl_1(B)$; 4 read(B); 5 if A = 0 then B := B + 1; 6 write B; 7 $u_1(A)$; 8 $u_1(B)$;	T2: $sl_2(B)$ read(B); $xl_2(A)$; read(A); if B = 0 then A := A + 1; write A. $u_2(A)$; $u_2(B)$;
---	---

上述所示的两个事务 T1 和 T2 的一个可能执行序列如下：

$sl_1(A); read_1(A); sl_1(B); read_2(B);$

到这之后，由于 T1 持有了 A 上的一个共享锁，T2 持有了 B 上的一个共享锁，此后 T1 对 B 和 T2 对 A 上的排他锁的申请均失败，不能往后执行，系统陷入死锁。