

# 数据库系统及应用作业 5

傅申 PB20000051

## 题 1.

嵌套事务可能带来以下的问题:

- 嵌套事务会导致系统复杂性增加.
- 两个事务互相调用, 会造成死循环.
- 嵌套事务可能会导致死锁. 若事务 A 先对某一资源加锁, 再开始事务 B, 但事务 B 需要等待该资源的锁, 此时事务 A 又需要等待事务 B 的完成, 造成死锁.
- 可能会破坏事务的原子性和一致性. 若事务 A 调用了事务 B, 且事务 B 成功提交了, 但事务 A 最后选择回滚. 由于事务 B 已经提交, 因此事务 A 无法撤销事务 B 的操作, 从而导致事务 A 的原子性和一致性被破坏.

## 题 2.

① 数据库系统恢复的过程如下:

(1) 正向扫描日志, T1 已经提交, 加入 Redo 列表; T2 和 T3 还未提交, 加入 Undo 列表.

(2) 反向扫描日志, 依次处理 Undo 列表中的事务 (T2 和 T3) 的操作:

$D = 45, E = 20, D = 20, C = 20$

(3) 正向扫描日志, 依次处理 Redo 列表中的事务 (T1) 的操作:

$A = 49, B = 250, A = 75$

恢复之后数据元素的值分别为:  $A = 75, B = 250, C = 20, D = 20, E = 20, F = 20, G = 20$ .

② 数据库系统恢复的过程如下:

(1) 正向扫描日志, T1, T2 和 T3 已经提交, 加入 Redo 列表.

(2) 正向扫描日志, 依次处理 Redo 列表中的事务 (T1, T2 和 T3) 的操作:

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

恢复之后数据元素的值分别为:  $A = 75, B = 250, C = 65, D = 46, E = 55, F = 20, G = 20$ .

③ 存在检查点 15), 从检查点之后开始恢复, 检查点之前即为 ② 的状态:

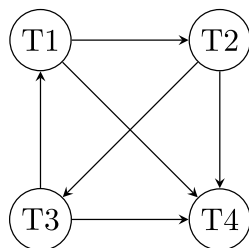
(1) 正向扫描日志, T4 还未提交, 加入 Undo 列表.

(2) 反向扫描日志, 依次处理 Undo 列表中的事务 (T4) 的操作:  $F = 100, F = 20, G = 20$

恢复之后数据元素的值分别为:  $A = 75, B = 250, C = 65, D = 46, E = 55, F = 20, G = 20$ .

## 题 3.

做出优先图如下, 图中显然存在环路, 并发调度不是冲突可串.



**题 4.**

对于优先图的一条边  $T_i \rightarrow T_j$ , 它说明存在  $T_i$  中的操作  $o_i$  和  $T_j$  中的操作  $o_j$  满足  $o_i \prec_S o_j$  且  $o_i$  和  $o_j$  冲突. 由于  $T_i$  和  $T_j$  都遵循 2PL, 则  $o_i$  和  $o_j$  需要获取同一个资源的锁, 且  $T_i$  释放该资源的锁后  $T_j$  才能获取该资源的锁. 因此,  $T_i$  一定在  $T_j$  获取全部锁之前开始释放锁, 即  $T_i$  释放第一个锁的时间早于  $T_j$  释放第一个锁.

使用反证法, 假设存在一个并发调度  $S$  中的所有事务都遵循 2PL, 但该调度不是冲突可串调度, 则其对应的优先图中必定存在环, 设其中一个环为  $T_1 \rightarrow T_2 \rightarrow \dots \rightarrow T_n \rightarrow T_1$ , 由上面的讨论, 记  $T_i$  释放第一个锁的时间为  $t_i$ , 则有  $t_1 < t_2, t_2 < t_3, \dots, t_{n-1} < t_n, t_n < t_1$ . 前  $n-1$  项说明  $t_1 < t_n$ , 与最后一项  $t_n < t_1$  矛盾. 因此, 该环不存在, 从而说明优先图中不存在环, 进一步得到并发调度  $S$  是冲突可串的.

**题 5.**

如果一个并发调度中的所有事务都遵循 2PL, 该并发调度还是会出现脏读问题. 如下, 第 7 行出现了脏读, T2 读取到 T1 修改过但未提交的数据.

t	T1	T2
1	xL1(A)	sL1(A) Read(A) # dirty read
2	Read(A)	
3	A = A + 100	
4	Write(A)	
5	U1(A)	
6		
7		
8	Rollback	