数据库概论作业 7

23.12.18

14.1

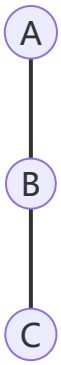
考虑满足两阶段锁协议的一个调度序列, 我们证明它是冲突可串行化的.

在调度对应的优先图中, 若有边 $T_i \to T_j$, 意味着对某个数据项 R, 出现了下述三种情况之一: T_i 读 R, 随后 T_j 写 R; T_i 写 R, 随后 T_j 读 R; T_i 写 R, 随后 T_j 写 R. 根据读写锁的性质, 一定是 T_i 先进行读/写, 然后释放锁, 之后 T_j 才能获取锁并读/写. 因此, T_i 的封锁点一定在 T_j 之前.

假设调度不是冲突可串行化的, 那么优先图有环, 那么封锁点的前后关系就有环, 这是不可能的. 因此, 两阶段锁能保证冲突可串行化. 封锁点的前后顺序就对应了优先图的一个拓扑排序, 因此事务可以根据封锁点来串行化.

14.5

考虑以下树结构:



满足树型封锁但不满足两阶段锁的调度: 单个事务执行 l(A) -> l(B) -> r(A) -> l(C) -> r(C) -> r(B).

满足两阶段锁但不满足树型封锁的调度: 单个事务执行 l(B) -> l(A) -> r(A) -> r(B).

其中 l 表示加锁, r 表示解锁.

14.30

满足两阶段锁但不满足时间戳排序的调度:

T1 (时间戳更小)	**T2**
	lock(A)
	write(A)
	unlock(A)
lock(A)	
read(A)	
unlock(A)	

满足时间戳排序但不满足两阶段锁的调度: 单个事务执行 lock(A) -> unlock(A) -> lock(A) -> unlock(A).

幻象

幻象现象: 事务 1 按一定条件读取关系后, 事务 2 插入/删除了满足该条件的元组, 事务 1 再按相同条件读取时, 得到的元组数发生了非预期的变化.

如果锁都是元组级别的, 那么即使采用两阶段锁, 也会出现幻象现象, 因为我们无法提前得知哪些元组需要上锁.