

第10章 事务与并发控制

10.1 事务

10.2 并发控制

10.2 并发控制

- 并发操作产生的问题
- 封锁协议
- 事务隔离

并发操作产生的问题 (1)

丢失修改 当两个事务读取相同数据并且都处理该数据（修改了它的值），然后都尝试更新原来的数据成新的值时，这种行为就会发生了。第二个事务可能完全覆盖掉第一个所完成的更新。

时间	取款事务A	取款事务B
T1	开始事务	
T2		开始事务
T3		查询账户余额为1000
T4	查询账户余额为1000	
T5		取出100，存款余额为900
T6	取出300，存款余额为700	
T7	提交事务	
T8		提交事务

最终账户余额为900，取款事务A的更新丢失了。丢失更新是这些行为中唯一一个用户可能在所有情况下都想避免的行为。

并发操作产生的问题 (2)

读脏数据 在一个事务读取未提交数据时会发生，如果一个事务修改了数据但是尚未提交修改，而另一个正在读取数据的事务会读到这个修改从而导致一种不一致的状态发生

时间	查询事务A	取款事务B
T1	开始事务	
T2		开始事务
T3		查询账户余额为1000
T4		取出100，存款余额为900
T5	查询账号余额为900	
T6		撤销事务，恢复为1000
T7	提交事务	

查询事务A读取到取款事务B还未提交的余额900。

默认情况下，脏读是不允许的。

注意：更新数据的事务是无法控制别的事务在它提交之前读取其数据的，这是由读取数据的事务来决定是否想要读取未必会被提交的数据。

并发操作产生的问题 (3)

不可重复读 如果同一事务分别以两个读操作读取相同资源时，可能会得到不同的值，这就是不可重复读

时间	查询事务A	取款事务B
T1	开始事务	
T2		开始事务
T3	查询账号余额为1000	
T4		取出100，存款余额为900
T5	查询账户余额为900	
T6		提交事务
T7	提交事务	

查询事务A两次读取余额获取到不同结果

并发操作产生的问题 (4)

幻影读 产生于一个数据集内的部分数据被修改时。如果事务A读取与搜索条件相匹配的若干行。事务B以插入或删除行等方式来修改事务A的结果集，然后再提交

时间	取款记录处理事务A	取款事务B
T1	开始事务	
T2		开始事务
T3	查询到5条取款记录	
T4		查询账户余额为1000
T5		取出100，存款余额为900
T6	查询到6条取款记录	
T7	提交事务	
T8		提交事务

对于取款记录处理事务A，两次查询的结果集不同

- 出现上述问题的原因是并发操作破坏了事务的隔离性。

- 资源锁定模式

- 基本锁
 - 排他锁 (X锁)
 - 共享锁 (S锁)
- 专用锁
 - 更新锁
 - 意向锁
 - 架构锁
 - 大容量更新锁

- 可锁定的资源类型 (粒度)

- RID
- KEY (行)
- PAGE(页)
- 对象 (如表等)
- 数据库

- 封锁协议

- 一级封锁协议：没有丢失修改
- 二级封锁协议：不读脏数据
- 三级封锁协议：可重复读

- 事务隔离级别（由高到低）

- **SERIALIZABLE**（可串行化）：允许事务与其他事务并发执行，但系统必须保证并发调度是可串行化，不致发生错误
- **REPEATABLE READ**（可重复读）：只允许事务读已提交的数据，并且在两次读同一个数据时不允许其他事务修改此数据
- **READ COMMITTED**（提交读）：允许事务读已提交的数据，但不要求“可重复读”
- **READ UNCOMMITTED**（未提交读）：允许事务读已提交或未提交的数据

隔离级别	脏数据	丢失修改	不可重复读	幻影读
未提交读	是	是	是	是
提交读	否	是	是	是
可重复读	否	否	否	是
可串行读	否	否	否	否

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
/READ COMMITTED/REPEATABLE READ/SERIALIZABLE