14 事务

事务是一个程序执行单元，它访问且可能更新不同的数据项。事务对于实现数据库中的数据更新是很关键的，只有这样才能保证并发执行与各种故障不会导致数据库处于不一致状态。

事务具有ACID特性：即原子性、一致性、隔离性和持久性

原子性保证事务的所有影响在数据库中要么全部反映出来，要么根本不反映。发生故障后数据库不能处于事物部分执行后的状态。

一致性保证若数据库一开始是一致的，则事务执行后数据库仍处于一致状态。

隔离性保证并发执行的事务相互隔离，使得每个事务感觉不到系统中其它事务的并发执行。

持久性保证一旦一个事务提交后，它对数据库的改变不会丢失，即使系统可能出现故障。

事务的并发执行提高了事务的吞吐量和系统利用率，也减少了事务的等待时间。

但多个事务在数据库中并发执行时，可能无法再保证数据的一致性。因此系统必须控制各并发事务之间的相互作用。

由于事务是保持一致性的单元，所以事务的串行执行能保证一致性。串行调度是可串行化的，但如果许多事务的步骤交错执行，则很难确定一个调度是否是可串行化的。确定一个事务有哪些操作、多个事务的操作如何相互作用是比较困难的。所以，通常不会考虑一个事务对某一数据项的的所有细节操作，而只考虑两种操作：read和write。

要求事务集的并发执行所产生的任何调度的执行效果等价于由这些事务按某种串行顺序执行的效果，保证这个特性的系统称为保证了可串行化。考虑一个调度S含有I，J两条指令，只有I和J全为read指令时，两条指令的执行顺序才是无关紧要的，而只要有至少一条是write指令时，I和J是冲突的。

如果调度S可以经过一系列非冲突指令交换转换成S1，则称S1是S的冲突等价。存在几种不同的等价概念，所以可串行化分为了冲突可串行化和视图可串行化。

事务并发执行所产生的调度的可串行化可以通过多种并发控制机制中的一种来加以保证。给定一个调度，可以通过为该调度构造优先图及搜索是否无环来判定它是否冲突可串行化。还有更好的并发控制机制可用来保证可串行化。

调度必须是可恢复的，这样如果事务a依赖于事务b，如果b中止时，a也要中止。调度也最好是无级联的，这样不会由于一个事务的中止引起其它事务的级联中止。无级联性是通过只允许事务读取已经提交的数据来保证的。