悲观锁：

是一种并发控制的方法，它可以阻止一个事务以影响其他用户的方式来修改数据，如果一个事务执行的操作对某行数据应用了锁，那只有当这个事务把锁释放后，其他事务才能够只想与该锁冲突的操作。悲观锁用于数据征用激烈的环境，以及发生并发冲突时使用锁保护数据的成本要低于回滚事务的成本环境中。

悲观锁的步骤如下：

在对任意记录进行修改前，先尝试为该记录加上排他锁。

如果加锁失败，说明该记录正在被修改，那么当前查询可能要等待或者跑出异常。

如果加锁成功，那么就可以记录做修改，事务完成后就会被解锁。

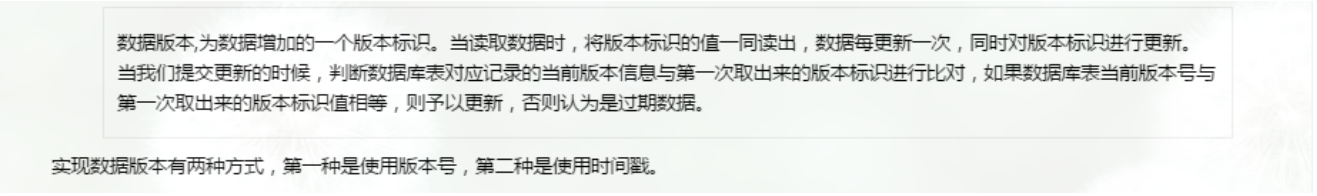
其间如果有其他对该记录做修改或者加排他锁的操作，都会等待解锁或者直接抛出异常。

优点与不足：

先获得锁再访问，为数据处理的安全提供了保证，但是在效率方面，处理加锁的机制会让数据库增加额外的开销，还有增加死锁的机会；另外，在只读型事务中由于不会产生冲突，也没有必要使用锁，这样做只能增加系统负载，还降低了并行性，一个事务如果锁定了某行数据，其他事务就必须等待改事务处理完后才可以处理那行数据。

乐观锁：

乐观锁是相对于悲观锁而言的，认为数据在一般情况下不会造成冲突，所以在数据进行提交更新的时候，才会正式对数据的冲突与否进行检测，如果发现冲突了，则返回给用户错误信息，让用户决定怎么去做。相对于悲观锁，在对数据库进行处理的时候，乐观锁并不会使用数据库提供的锁机制，一般的实现乐观锁的方式就是记录数据版本号。



优点与不足：

乐观并发控制相信事务之间的数据竞争的概率是比较小的，因此尽可能的直接做下去，直到提交的时候才去锁定，所以不会产生任何锁和死锁，但如果直接简单这么做，还有可能会遇到不可预期的结果，例如两个事务都读取了数据库的某一行，经过修改后写会数据库，就会出现麻烦。