# 21.更新丢失问题

## 21.1.更新丢失问题的产生

两个并发的事务基于同一个查询结果进行修改，后提交的事务忽略了先提交的事务对数据库的影响，造成了先提交的事务对数据库的影响丢失，这个过程就叫做更新丢失。

案例1：easymall删除商品而未删除商品分类案例

案例2：重复充值案例

案例3：秒杀抢购案例

## 21.2.更新丢失解决方案:

将数据库隔离级别设置为Serializable就可以防止更新丢失问题的产生 - 但是很少这样解决，因为效率太低。

**乐观锁和悲观锁：**

乐观锁和悲观锁并不是数据库中真实存在的锁,而是两种解决方案的名字。

**悲观锁：**

悲观锁悲观的认为，每次查询都会造成更新丢失，所以在查询时，手动添加排它锁，排斥查询，从而解决更新丢失问题。

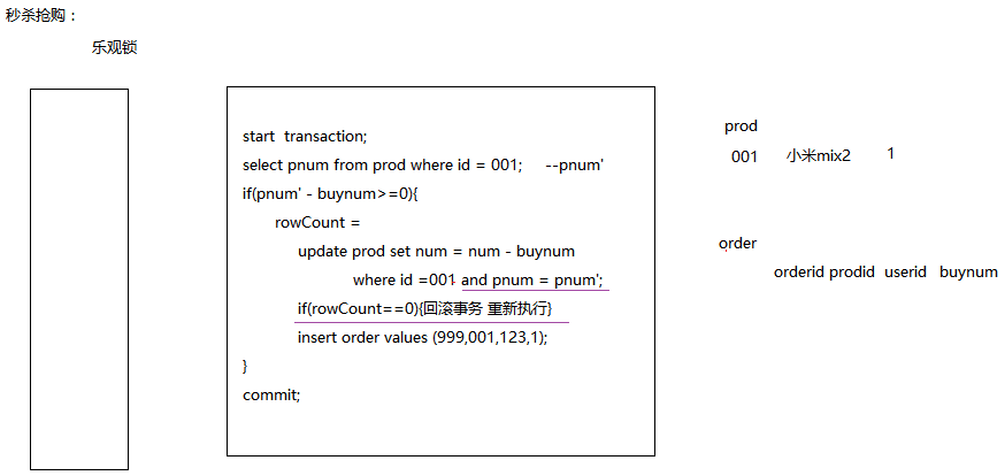
for update：在查询时手动增加排他锁。

例如：select \* from xxx for update;

**乐观锁：**

乐观锁乐观的认为，每次查询都不会造成更新丢失，在修改时，检测更新丢失的发生来进行纠正。

例如：在秒杀抢购的时候，将更新库存代码改为如下代码：



如果查询多，而更新少，用乐观锁。

如果更新多，而查询少，用悲观锁。

## 21.3.EasyMall删除商品更新丢失问题解决

|  |
| --- |
| 在ProdDaoImpl实现类中，findProdByCid方法中的sql语句后面加上for update即可：  String sql = "select \* from prod where cid = ? for update"; |

## 21.4.作业练习

1. 乐观锁和悲观锁的区别

答：悲观锁：

悲观锁悲观的认为，每次查询都会造成更新丢失，所以在查询时，手动添加排它锁， 排斥查询，从而解决更新丢失问题。

乐观锁：

乐观锁乐观的认为，每次查询都不会造成更新丢失，在修改时，会判断一下在此期间别人有没有去更新这个数据。

如果查询多，而更新少，用乐观锁。如果更新多，而查询少，用悲观锁。

1. web.xml 的作用？

答：用于配置Web应用的相关信息，如：监听器（listener）、过滤器（filter）、 Servlet、 相关参数、会话超时时间、错误页面等。

1. 线程安全指什么？

答：如果你的代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这 段代码。如果每次运行结果和单线程运行的结果是一样的，而且其他的变量的值也和预 期的是一样的，就是线程安全的。