**第12章 事务**

**1. 什么是事务**

所谓事务，就是逻辑上的一组操作，要么同时完成，要么同时不完成。在我们的MySQL里面默认就是自带事务，但是MySQL自带的事务就是一条语句就是一个事务

**2. 为什么需要事务**

举个例子，现实生活中，人们经常转账。我们的转账可以分为两个部分：一个是转入，一个是转出。只有两个部分同时完成的时候，我们才能认为转账成功。任何一方如果不成功，就不算这个事务成功。例如张三给李四转钱，张三卡上扣了1000元，但是就在这个时候，ATM机坏掉了，如果没有事务，那么张三卡上就已经被扣掉了1000元，但是李四并没有收到这1000，这对张三来说就是一种损失，所以我们需要事务。

**3. 使用事务**

如果要使用事务，首先需要开启事务

开启事务的语法：

start transaction

因为在MySQL中一条语句默认就是一个事务，所以当我们要执行一组操作的时候，必须先要开启事务，然后书写操作语句，最后需要提交事务。

提交事务的语法：commit

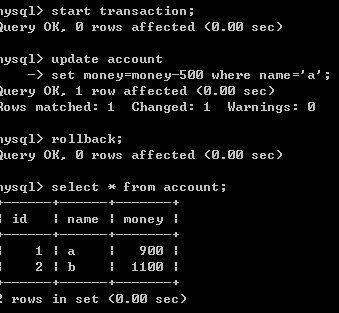
如果开启了事务以后不提交，退出MySQL，之前在事务中所做的操作就全部回滚回去了。

**4. 事务的回滚**

如果不想提交当前的事务，我们可以对事务进行回滚，回滚事务的关键字

rollback

例如：a又给b转账500，转账出去后悔了，想要结束转账，这个时候就可以使用关键字rollback回滚事务



注意点：我们的commit和rollback都可以结束事务

commit：结束事务，执行事务中的所有操作

rollback：结束事务，回滚事务中的所有操作（开启事务之后的所做的操作全部失效）

**5. 事务的特性**

我们的事务有很严格的定义，需要同时满足4个特性：原子性，一致性，隔离性，持久性，也就是人们常说的ACID标准

原子性：所谓原子性，就是指事务是一组不可分割的单位，要么同时成功，要么同时不成功

一致性：所谓一致性，就是指事务前后的数据必须要一致

隔离性：所谓事务的隔离性，就是指多个用户并发访问数据库时，一个用户的事务不能被其他用户的事务所干扰

持久性：所谓持久性，就是指一个事务一旦被提交，那么它对数据库中的数据的改变就是永久的。

**6. 事务的隔离级别**

事务的特性中，原子性，一致性和持久性都是数据库设计者可以为我们定下来的，唯独事务的隔离性，数据库设计者不能决定，所以交给了数据库用户来决定事务的隔离性

事务的隔离级别分为4个：读未提交（read uncommitted），读提交（read commited），可重复读（repeatable read），可串行化（serializable）

如果两个事务并发的查询：完全不用隔离

如果两个事务并发的修改：必须要隔离开

设置事务的隔离级别

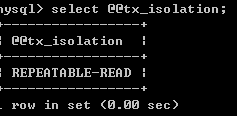
语法：

set session transaction isolation level 级别；

显示当前的隔离级别

select @@tx\_isolation；

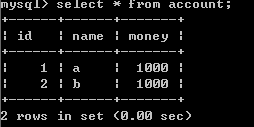
在我们的MySQL中，默认的隔离级别为repeatable read



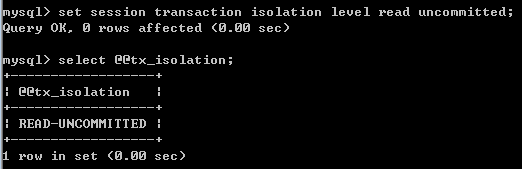
① 读未提交

容易产生脏读（dirty read）的现象：读了别人没有提交的事务

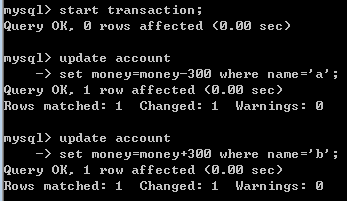
首先两边卡上各有1000元



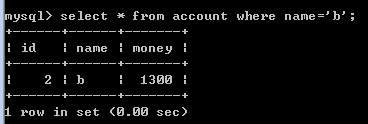
然后将控制台2的隔离级别修改为read uncommitted



接下来控制台1开启事务，进行转账，但是，不提交



接下来在控制台2进行查询



在接下来控制台1进行回滚操作



再到控制台2进行查询，发现money重新回到了1000

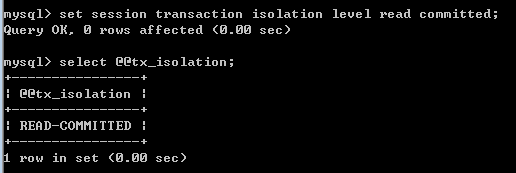


上面的情况就是脏读（dirty read）

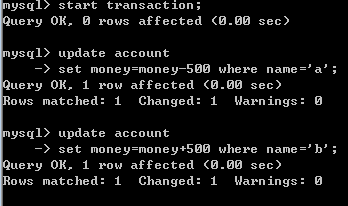
② 读提交

读提交的隔离级别可以防止脏读，但是读提交不能防止不可重复读和幻读

首先将小b的隔离级别设置为读提交（read committed）



接下来小a又开始使坏了，开启事务，转账500给小b，但是同样不提交事务



小b开始进行查询，发现卡上钱并没有多，你特么唬谁呢

****