**1.为什么要使用事务？**

我们在执行SQL语句的时候，**某些业务要求**，多个DML语句需要全部执行成功；

例如转账操作：

账号1 转账给账户2；

update account set money = money -100 where userid = 1;

update account set money = money +100 where userid = 2;

这两个DML语句必须同时执行成功才能完成转账这个操作，如果由于某些原因，其中一条没有执行

成功，则两条语句都必须执行不成功；

**2.概念：**

这样把多条语句作为一个整体进行操作的功能，被称作数据库的事务；

在一个事务范围内的操作要么全部成功，要么全部失败，如果失败则会对数据库没有任何的影响。

**3.事务特征**

有刚才的讲解我们得出事务的四个特征：

A：atomic 原子性；

C:consistent 一致性；

I：isolation 隔离性；

D:durability 持久性；

4.事务的分类

隐式事务：对于单条的SQL语句，数据库系统默认的它当成一个事务执行；

显式事务：如果想要把多条SQL语句作为一个事务单元执行则必须使用显式的事务；显式事务以

begin开始，以commit结束；

如果commit成功，则事务对数据库的影响则是永久的；

如果commit失败，则事务对数据库是没有影响的；

如果想主动让一个事务失败则可以使用rollback;

数据库事务是由数据库系统保证的，我们只需要根据业务逻辑使用它就可以；

**5.隔离级别**

对于两个并发执行的事务，如果涉及到操作同一条记录的时候，可能会发生问题；

因为并发操作会带来数据的不一致性，包括脏读、不可重复读、幻读等。数据库系统提供了事务的

隔离级别，避免数据的不一致的问题；

1. read uncommitted 未提交读

是隔离级别最低的一种事务级别。在这种隔离级别下一个事务会读取到另一个事务更新后但是未提交的数据。如果另一个事务回滚，则当前事务读取到的数据就是脏读；

1. 提交读read committed

该隔离级别一个事务会遇到不可重复读的问题；

不可重复读是指的是，在一个事务内，多次读取同一数据，在这个事务还没有结束时，如果另一个事务恰好修改了这个数据，那么在第一个事务中，再次读取的数据就可能不一致。同一个事务，两次读取的数据不一致，就是不可重复读；



1. repeatable read 可重复读级别

在这种隔离级别下，可能会出现幻读的问题；

幻读是在一个事务中去查询某个记录，不存在，但是去更新这个记录时竟然可以更新，然后再次查询竟然奇迹出现了，这就是幻读；



1. serializable read 序列化级别

这个隔离级别是最高的，但是由于事务是串行执行，这样就大大降低了程序的执行性能，所以我们一般不会采取这种级别；