# 隔离级别：

1：只加一个@Transactional

存在脏读

@Transactional(isolation = Isolation.***READ\_UNCOMMITTED***)

**能读取还未提交的事务，不能解决脏读、不可重复度和幻读**

**其他事务可以看到。但是必须是事务，非事务还是看不到**

@Transactional(isolation = Isolation.***READ\_COMMITTED***)

**只能读取已经提交的事务，能解决脏读，不能解决不可重复读和幻读，一般使用此隔离级别。**运行成功后数据才其他事务的看到

DEFAULT：使用数据库本身使用的隔离级别 ORACLE（读已提交） MySQL（可重复读）

数据库同spring中都有下面四种隔离级别

READ\_UNCOMITTED：读未提交（脏读）最低的隔离级别，一切皆有可能。

READ\_COMMITED：读已提交，ORACLE默认隔离级别，有幻读以及不可重复读风险。

REPEATABLE\_READ：可重复读，解决不可重复读的隔离级别，但还是有幻读风险。我觉得解决了脏读的

SERIALIZABLE：串行化，最高隔离级别，杜绝一切隐患，缺点是效率低。

## 脏读

 指当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据， 那么另外一 个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。

## 不可重复读

 指在一个事务内，多次读同一数据。在这个事务还没有结束时，另外一个事务也访问该同一数据。 那么，在第一个事务中的两次读数据之间，由于第二个事务的修改，那么第一个事务两次读到的数据可能是不一样的。这样就发生了在一个事务内两次读到的数据是不一样的，因此称为是不可重复读。

## 幻觉读

一个事务(同一个read view)在前后两次查询同一范围的时候，后一次查询看到了前一次查询没有看到的行。两点需要说明：  
　　1、在可重复读隔离级别下，普通查询是快照读，是不会看到别的事务插入的数据的，幻读只在当前读下才会出现。  
　　2、幻读专指新插入的行，读到原本存在行的更新结果不算。因为当前读的作用就是能读到所有已经提交记录的最新值。

# 传播行为

