

数据库事务的隔离级别有 4 种，由低到高分别为 **Read uncommitted**、**Read committed**、**Repeatable read**、**Serializable**。而且，在事务的并发操作中可能会出现脏读，不可重复读，幻读。下面通过事例一一阐述它们的概念与联系。

Read uncommitted

读未提交，顾名思义，就是一个事务可以读取另一个未提交事务的数据。

事例：老板要给程序员发工资，程序员的工资是 3.6 万/月。但是发工资时老板不小心按错了数字，按成 3.9 万/月，该钱已经打到程序员的户口，但是事务还没有提交，就在这时，程序员去查看自己这个月的工资，发现比往常多了 3 千元，以为涨工资了非常高兴。但是老板及时发现了不对，马上回滚差点就提交了的事务，将数字改成 3.6 万再提交。

分析：实际程序员这个月的工资还是 3.6 万，但是程序员看到的是 3.9 万。他看到的是老板还没提交事务时的数据。这就是脏读。

那怎么解决脏读呢？**Read committed**！读提交，能解决脏读问题。

Read committed

读提交，顾名思义，就是一个事务要等另一个事务提交后才能读取数据。

事例：程序员拿着信用卡去享受生活（卡里当然是只有 3.6 万），当他埋单时（程序员事务开启），收费系统事先检测到他的卡里有 3.6 万，就在这个时候！！程序员的妻子要把钱全部转出充当家用，并提交。当收费系统准备扣款时，再检测卡里的金额，发现已经没钱了（第二次检测金额当然要等待妻子转出金额事务提交完）。程序员就会很郁闷，明明卡里是有钱的...

分析：这就是读提交，若有事务对数据进行更新（**UPDATE**）操作时，读操作事务要等待这个更新操作事务提交后才能读取数据，可以解决脏读问题。但在这个事例中，出现了一个事务范围内两个相同的查询却返回了不同数据，这就是不可重复读。

那怎么解决可能的不可重复读问题？**Repeatable read**！

Repeatable read

重复读，就是在开始读取数据（事务开启）时，不再允许修改操作。

事例：程序员拿着信用卡去享受生活（卡里当然是只有 3.6 万），当他埋单时（事务开启，不允许其他事务的 **UPDATE** 修改操作），收费系统事先检测到他的卡里有 3.6 万。这个时候他的妻子不能转出金额了。接下来收费系统就可以扣款了。

分析：重复读可以解决不可重复读问题。写到这里，应该明白的一点就是，不可重复读对应的是修改，即 **UPDATE** 操作。但是可能还会有幻读问题。因为幻读问题对应的是插入 **INSERT** 操作，而不是 **UPDATE** 操作。

什么时候会出现幻读？

事例：程序员某一天去消费，花了 2 千元，然后他的妻子去查看他今天的消费记录（全表扫描 FTS，妻子事务开启），看到确实是花了 2 千元，就在这个时候，程序员花了 1 万买了一部电脑，即**新增 INSERT** 了一条消费记录，并提交。当妻子打印程序员的消费记录清单时（妻子事务提交），发现花了 1.2 万元，似乎出现了幻觉，这就是幻读。

那怎么解决幻读问题？Serializable！

Serializable 序列化

Serializable 是最高的事务隔离级别，在该级别下，事务**串行化顺序执行**，可以避免脏读、不可重复读与幻读。但是这种事务隔离级别效率低下，比较耗数据库性能，一般不使用。

值得一提的是：大多数数据库默认的事务隔离级别是 Read committed，比如 Sql Server ， Oracle。Mysql 的默认隔离级别是 Repeatable read。