

## 67\_事务的基础知识筑基（一）：事务的隔离级别

---

**儒猿架构官网上线**，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：[www.ruyuan2020.com](http://www.ruyuan2020.com)（建议 PC 端访问）

---

### 1、开篇

上节课先通过复习事务 ACID 的特性，对事务的本质有所了解，然后返回创建订单的业务流程中，对照事务的特性将符合其特性的三个操作：创建订单、使用优惠券和扣除积分放到一个事务中处理。下节课会给大家补充事务的基础知识：事务的隔离级别。今天课程的内容包括以下几个部分：

- 事务的隔离级别

### 2、事务的隔离级别

上节课说了下 ACID，这里继续聊事务，谈谈事务的隔离性。隔离性是指，多个用户的并发事务访问同一个数据库时，一个用户的事务不应该被其他用户的事务干扰，多个并发事务之间要相互隔离。根据数据库的隔离性，定义了以下 4 种隔离级别：

- 读未提交（Read Uncommitted）一个事务还没提交，被另外一个事务读取了数据库中的数据，也把这种情况称为脏读。如图 1 所示，客户端 B 开启了事务 C 修改 MySQL 数据库中的 name 字段，把原本的“name=李四”修改成“name=张三”。但是在事务 B 还没有完成提交之前，客户端 A 又启动了事务 A 对这条记录发起查询操作，此时返回的数据是“name=李四”，并不是“name=张三”，因此产生了脏读。

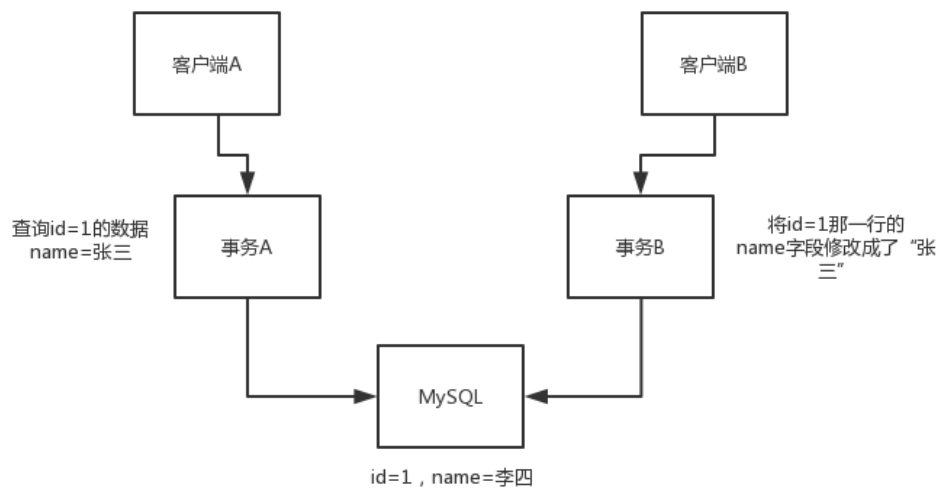


图 1

- 读已提交（Read Committed）：一个事务内对同一条数据进行两次读操作，返回的值不一样，也称作不可重复读。如图 2 所示，客户端 A 启动事务 A，第一次从 MySQL 数据库中读取记录“name=李四”。此时客户端 B 启动事务 B，对该条记录进行修改，将其修改为“name=张三”，并且成功提交事务 B。此时，事务 A 第二次读该条记录，记录从“name=李四”变成了“name=张三”，虽然这条记录的“id=1”，但是由于事务 B 的介入导致事务 A 在一个事务操作中读两次相同的记录返回了不同的值。这个级别可以避免脏读的发生。

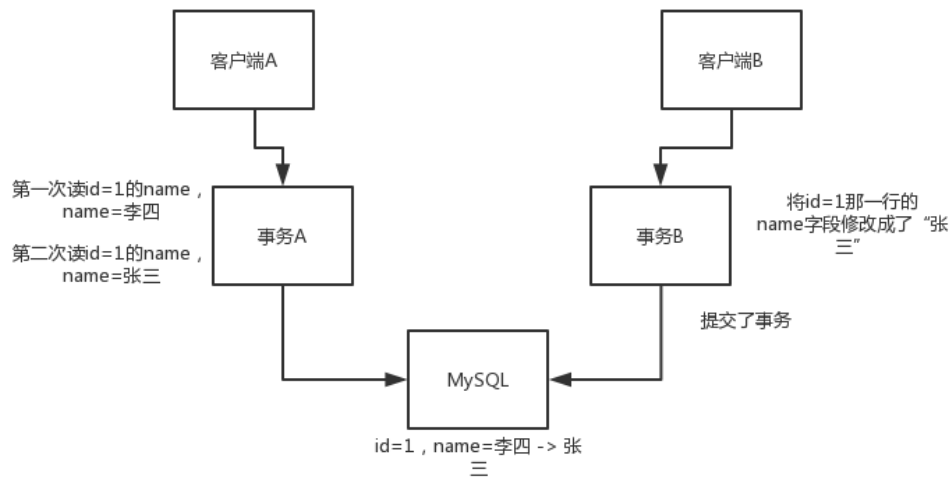


图 2

- 可重复读（Read Repeatable）：它就好比上面的不可重复读要好很多，也就是说在一个事务中对同一条记录进行多次读取，返回的值都是一样的，避免了不可重复读的窘境。如图 3 所示，事务 A 读取“id=1”的记录返回“name=李四”。此时事务 B 将“id=1”修改为“name=张三”并且成功提交事务。此时事务 B 再读取“id=1”的记录，返回值依旧是“name=李四”，并且没有因为事务 B 的介入返回的值产生变化。这个级别可以避免脏读和不可重复读的发生。

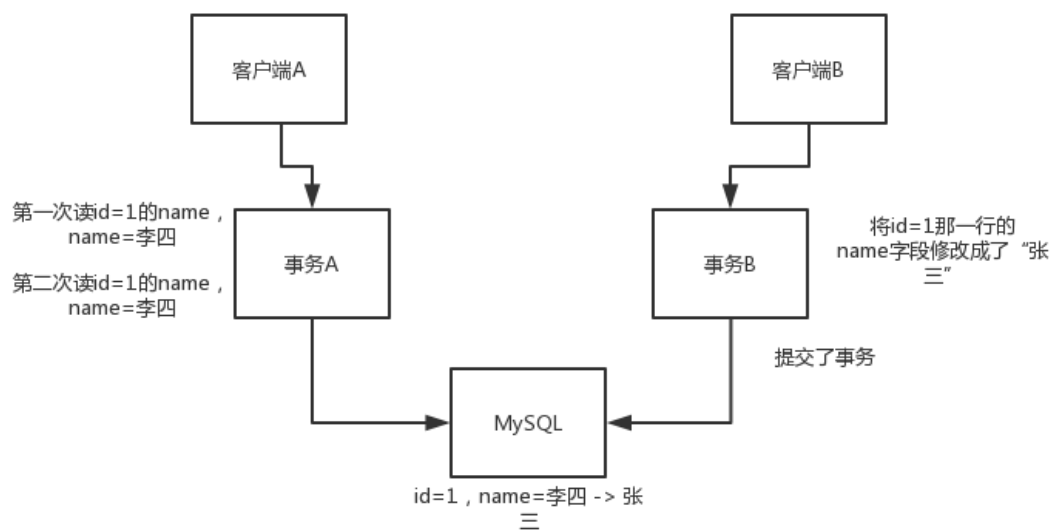


图 3

- 串行化（**Serializable**）：它是事务执行的最高级别，思路是将事务执行串行化，也就是说所有的事务执行排队一个一个执行，不允许多个事务并行操作。这样不仅可以避免上面说到的读未提交（脏读）、不可重复读，还可以阻止幻读的发生。说到幻读这里说明一下，如图 4 所示，事务 A 第一次从 MySQL 获取表中所有数据，此时数据只有 1 条就是“id=1”。此后事务 B 往表中插入一条数据“id=2”，返回到事务 A 再次获取表中所有数据，发现有 2 条：“id=1”和“id=2”，这种情况就是幻读。

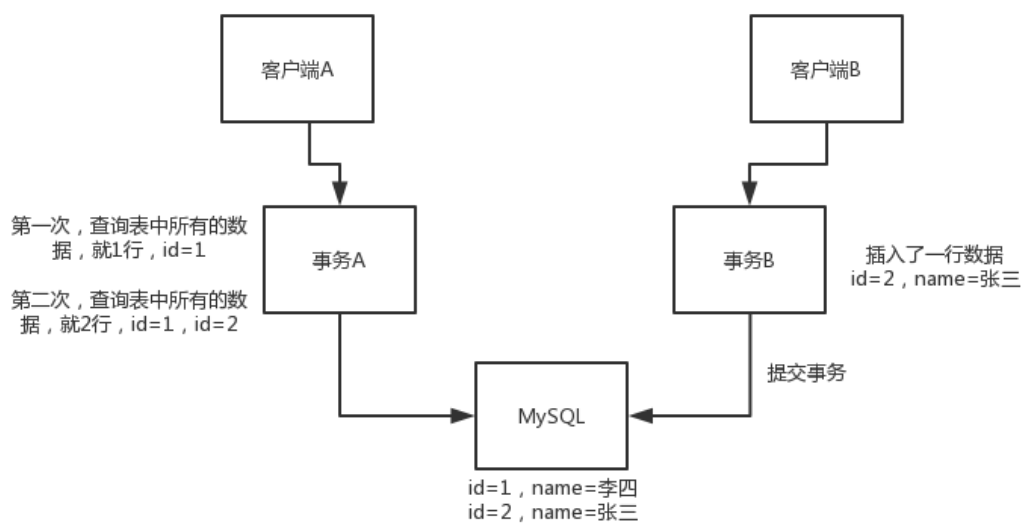


图 4

为了阻止幻读串行化，将事务执行逐个进行不能并行，如图 5 所示，当事务 A 执行完毕之前事务 B 是不让执行的，这样就不会出现幻读的情况。当然这种高级别事务机制也导致了执行效率的降低，在高可靠性的场景可以使用，例如：金融。

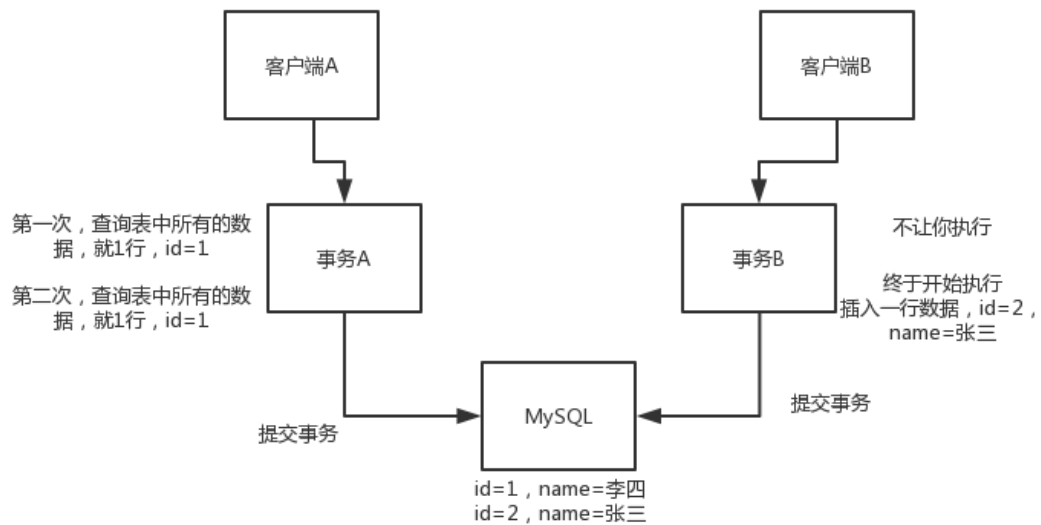


图 5

上面提交的四个隔离级别，从低到高排序是：读未提交、读已提交、可重复读、串行化。级别越高事务的安全性越高，效率就越低。**MySQL** 的默认隔离级别是可重复读，就是说每个事务都会开启一个自己要操作的某个数据的快照，事务期间，读到的都是这个数据的快照罢了，对一个数据的多次读都是一样的。

### 3、总结

本节课介绍了事务的四种隔离级别，分别是读未提交、读已提交、可重复读、串行化。它们按照隔离级别从低到高排序，级别越高事务的安全性越高，效率就越低。下节课说说 **Spring** 的事务支持以及传播特性。下期见，拜拜。