事务隔离级别

2021年10月6日 20:22

@Transaction的isolation属性

事务并发的问题

1. 脏读
   事务之间的隔离性没有做好，事务之间完全没有隔离
   例：
   A事务在事务过程中间修改数据，B事务会读到A事务在过程中所修改的数据
   如果说A事务最后回滚了，它把它修改的数据改回和原来一样的
   但是B的事务读到了A事务的过程中间的数据
   我们就说B事务读到了A的脏数据

2. 不可重复读
   如果处理了事务之间不可读修改过程中的数据，就会出现不可重复读问题
   例：
   如果A事务的时间是比较长的，它首先读了一个数据
   A事务读完数据以后，我们开启了另外一个并行的B事务
   B事务去修改这个数据，在B事务结束了以后，它会把它的修改的数据值提交
   A事务如果在B事务结束以后再去读数据的，就会发现A事务在B事务开始之前读的数据和B事务结束之后读的数据是不一样的
   这就是我们称之为不可重复读问题

   如果要解决不可重复读问题，最简单的方式是对数据进行加锁，即读锁
   但加锁不能保证解决幻读的问题

3. 幻读
   例：
   A事务要统计在系统中间所有的商品的总数，它在读取了系统中所有的商品的数量后，把所有的商品都加了一个读锁
   但是B事务可以向这个系统中中间去新增一个新的商品
   如果A事务再去读一遍商品的总数的时候，就会发现多了一个商品
   这就是幻读的问题

   如果要解决幻读问题，就要做到完全串行，即一个事务在读数据的时候，另一个事务不能对其进行任何改动

isolation可用的值

为了解决事务的并行问题，就要对isolation属性进行规定

1. DEFAULT
   默认级别，即采用数据库的隔离级别
   MySQL的默认隔离级别是repeatable-read，可重复读
2. READ_UNCOMMITTED
   最低级别，完全没有隔离事务，一个事务可以读到另一个事务在过程中产生的数据，会产生所有问题

3. READ_COMMITTED
   进行有限隔离，一个事务不会读到另一个事物在中间过程中修改的数据，但是会导致不可重复读和幻读问题
4. REPEATEABLE_READ
   不论怎么读数据，都是一样的，这是对数据增加了读锁
5. SERIALIZABLE
   完全串行，把所有对数据的操作都变成串行的