Spring 的Transactional注解，默认使用数据源的隔离级别。  
Mysql是 可重复读（Read Repeatable，避免了脏读和不可重复读，可能出现幻读）  
Oracle和SqlServer是 读已提交（Read Committed，避免了脏读，可能出现不可重复读）

1. 查看全局隔离级别

|  |
| --- |
| show global variables like '%isolation%' |

1. 设置全局隔离级别

|  |
| --- |
| set global transaction\_isolation ='read-committed'; |

1. 查看会话隔离级别

|  |
| --- |
| show session variables like '%isolation%' |

1. 设置会话隔离级别

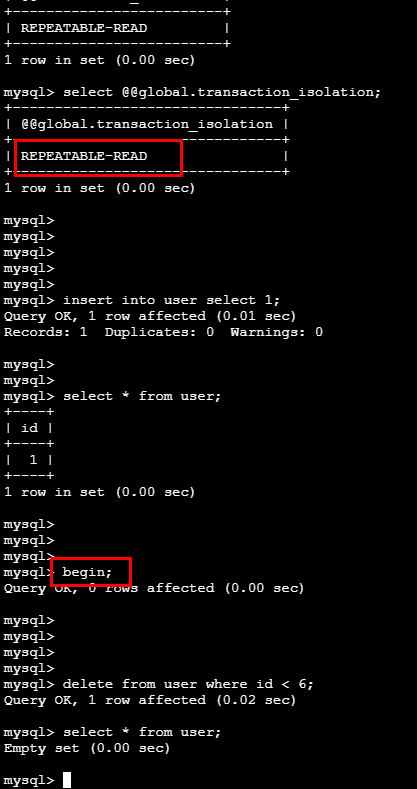
|  |
| --- |
| set session transaction\_isolation ='read-committed'; |

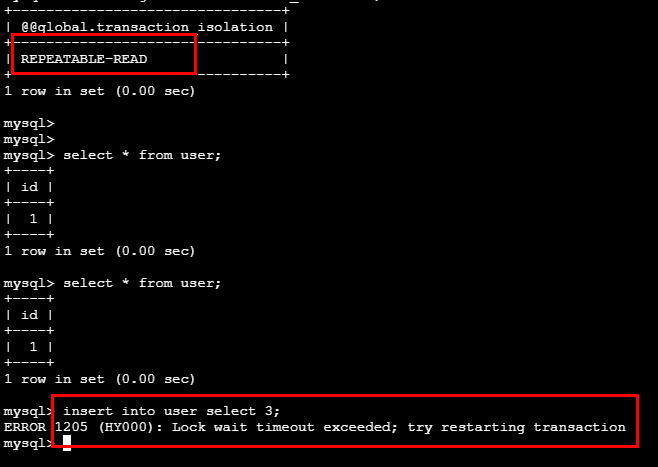
1. 直接通过变量名查看

|  |
| --- |
| select @@global.transaction\_isolation,session.transaction\_isolation; |

正常情况下，可重复读可能会出现了幻读问题，但是Mysql的可重复读通过**间隙锁**解决了幻读，

开启了可重复读，一个线程的事务没有完成时，别的连接的数据库insert或者delete会阻塞。





Mysql使用读已提交隔离级别和binlog使用statement时会出现主从数据不一致问题

binlog的记录顺序是按照事务commit顺序为序的

简单理解：

1、采用**RR隔离级别**时，A事务未提交时，B事务执行insert会阻塞

2、A在事务中使用for update，B未开启事务，这是B执行insert会阻塞