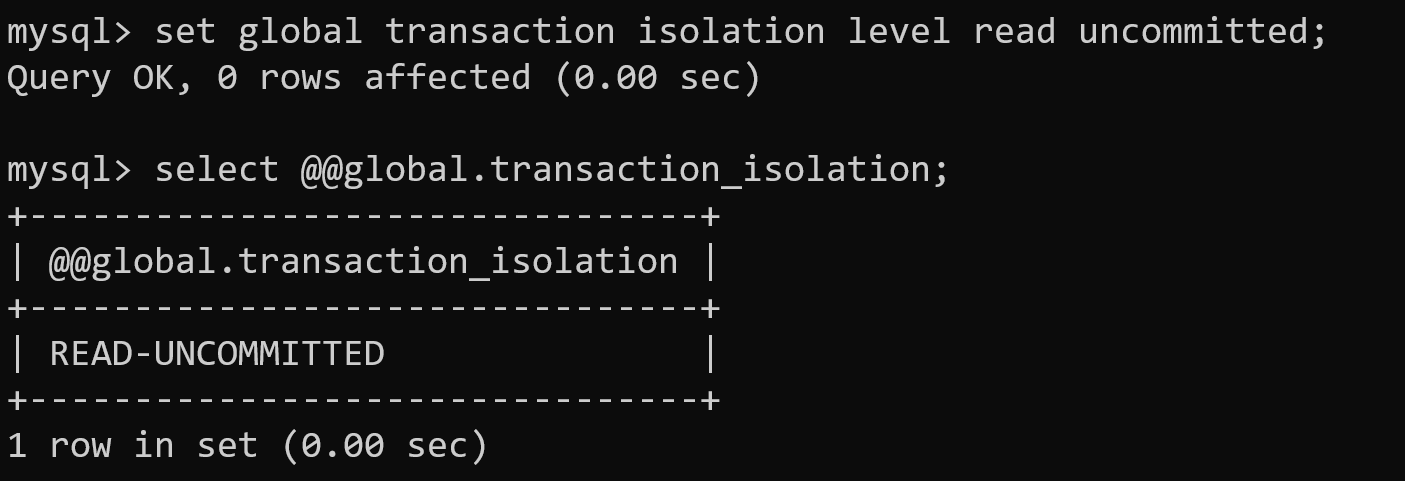
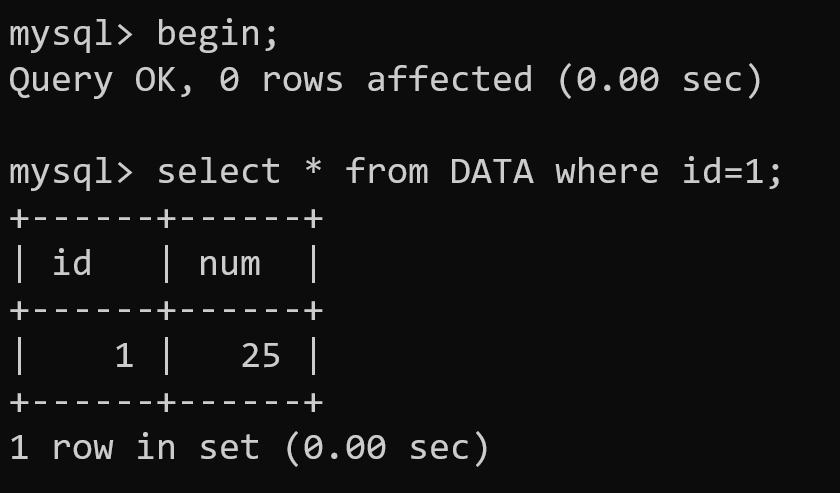
一、未提交读

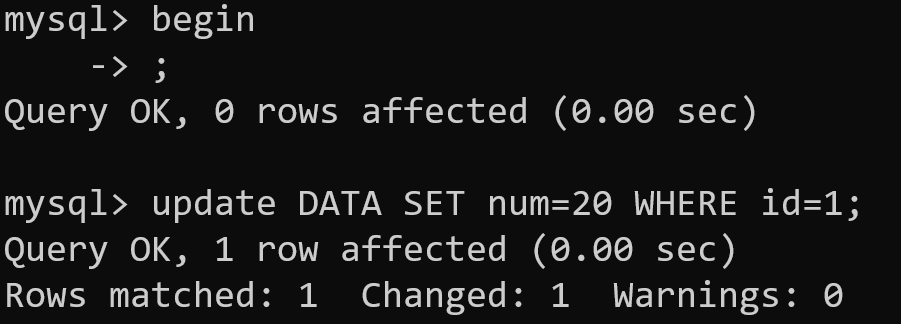
修改全局为未提交读



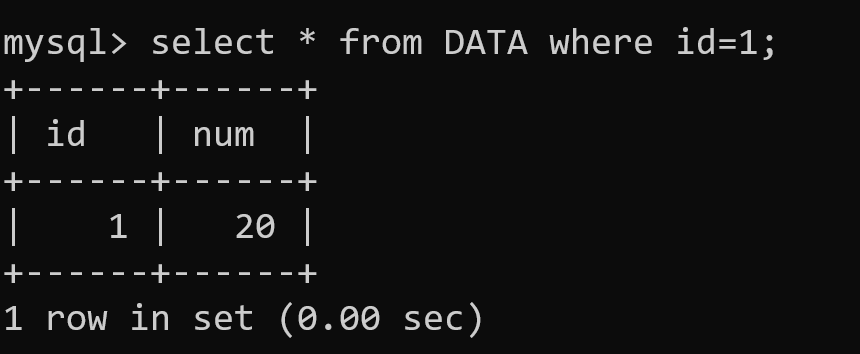
1. 开启一个事务，读取id为1的数据，值为25



1. 开启另一事务，修改id为1的数据为20



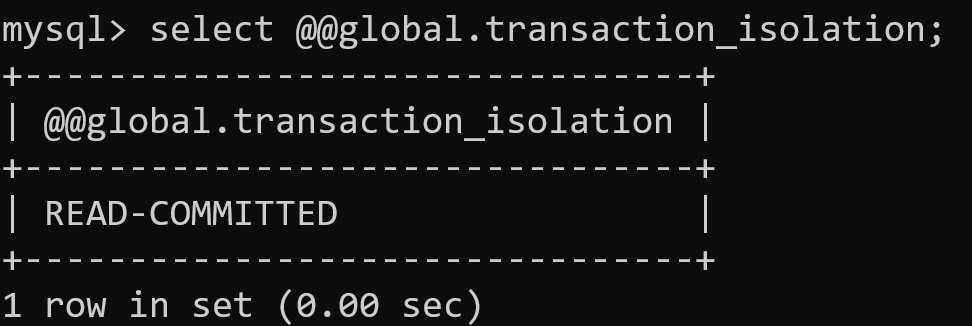
1. 再从第一个事务中读取id为1的数据，发现值变为20



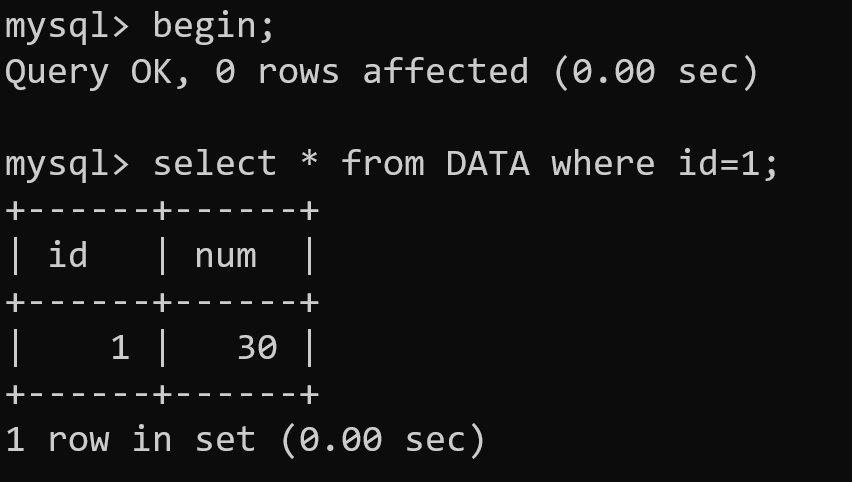
实现原理：读不加锁“当前读”，写加排它锁（X锁）

二、提交读

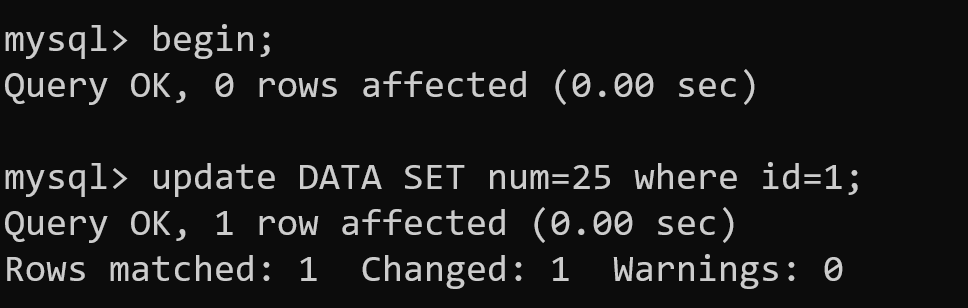
修改全局变量为提交读



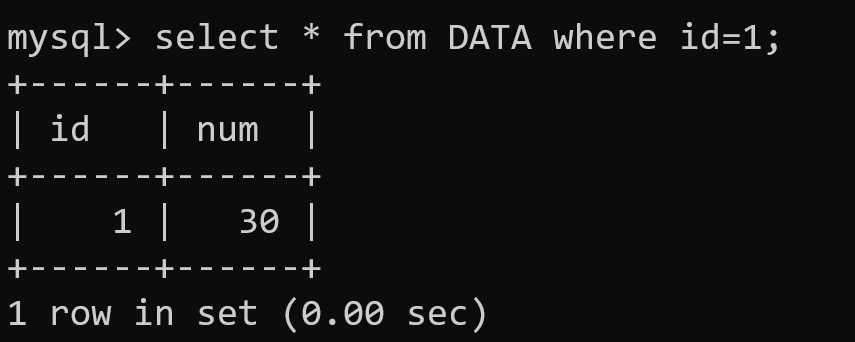
(1)开启一个事务，查询id1数据为30



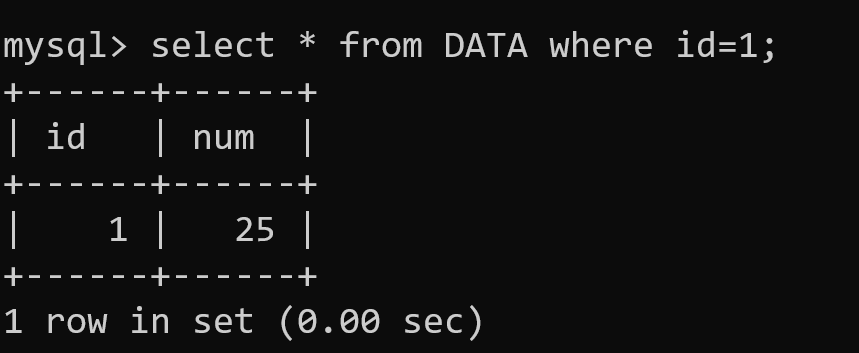
(2)开启另一事务，修改id1数据为25



(3)在第一个事务中查询id1数据仍为30



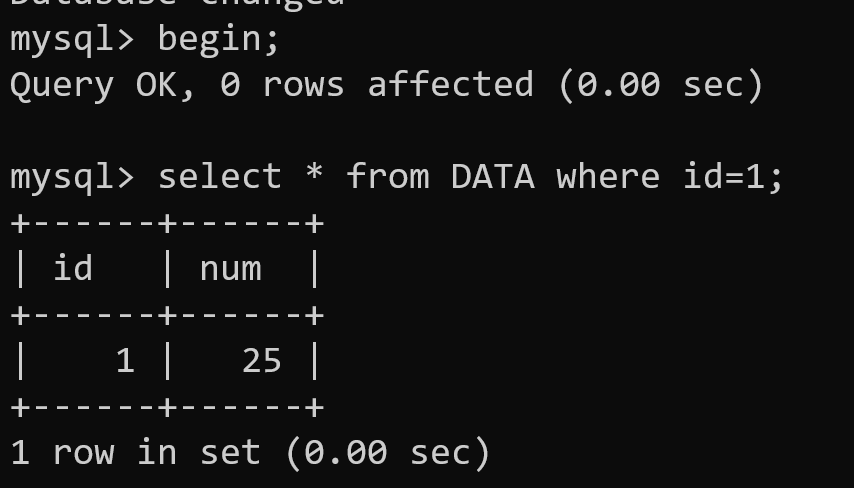
1. 第二事务提交后，在第一事务中查询id1数据为25



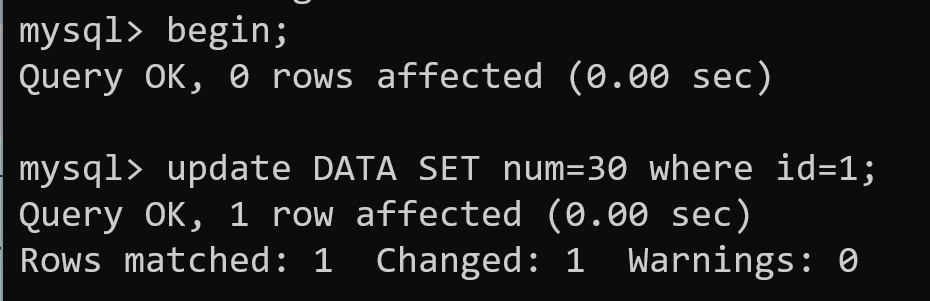
实现原理：读不加锁,做“快照读”并总是读最新的快照，写加排它锁(X锁)

三、可重复读

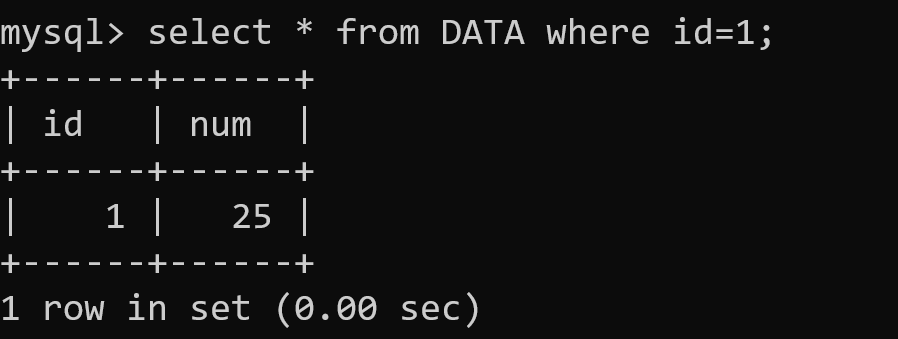
(1) 开启一个事务，查询id1数据为25



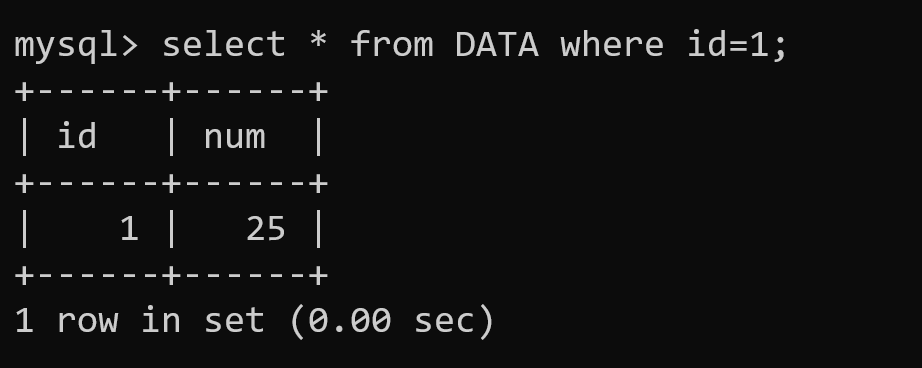
(2)开启另一个事务，修改id1数据为30



(3)在事务一中查询id1仍为25



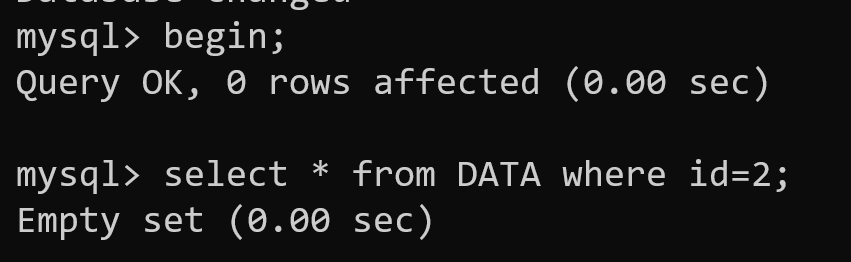
(4)事务二提交之后，事务一查询id1数据仍为25



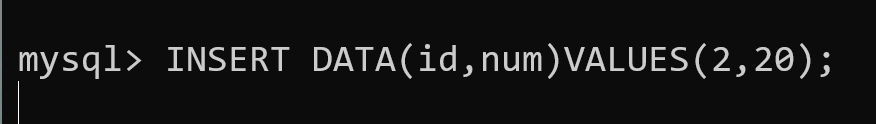
实现原理：读不加锁,做“快照读”并总是读事务开始第一次读的那个版本的快照。写加排它锁(X锁)

四、序列化

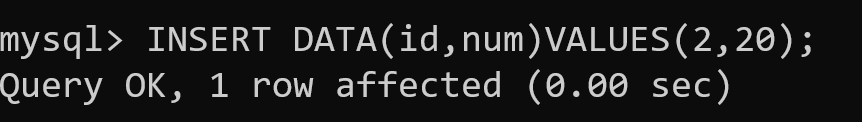
(1)在第一个事务中查询id2数据为空



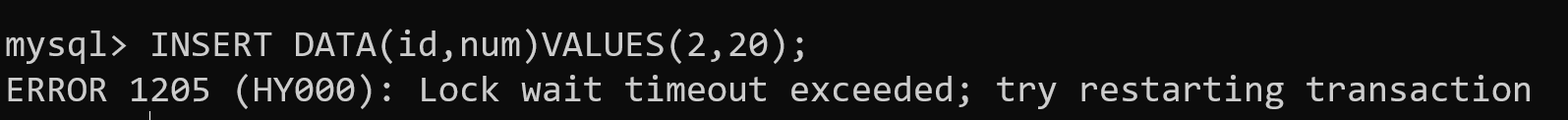
(2)在另一个事务中插入id2，发现被阻塞



(3)在事务一中插入id2数据并提交



(4)事务二报错，id2数据已经存在



实现原理：读加共享锁（S锁)，一致性的锁定读，写加排它锁(X锁)