测试service事务

注意接口Service里面方法的Transaction注解

1. **Propagation.REQUIRED**

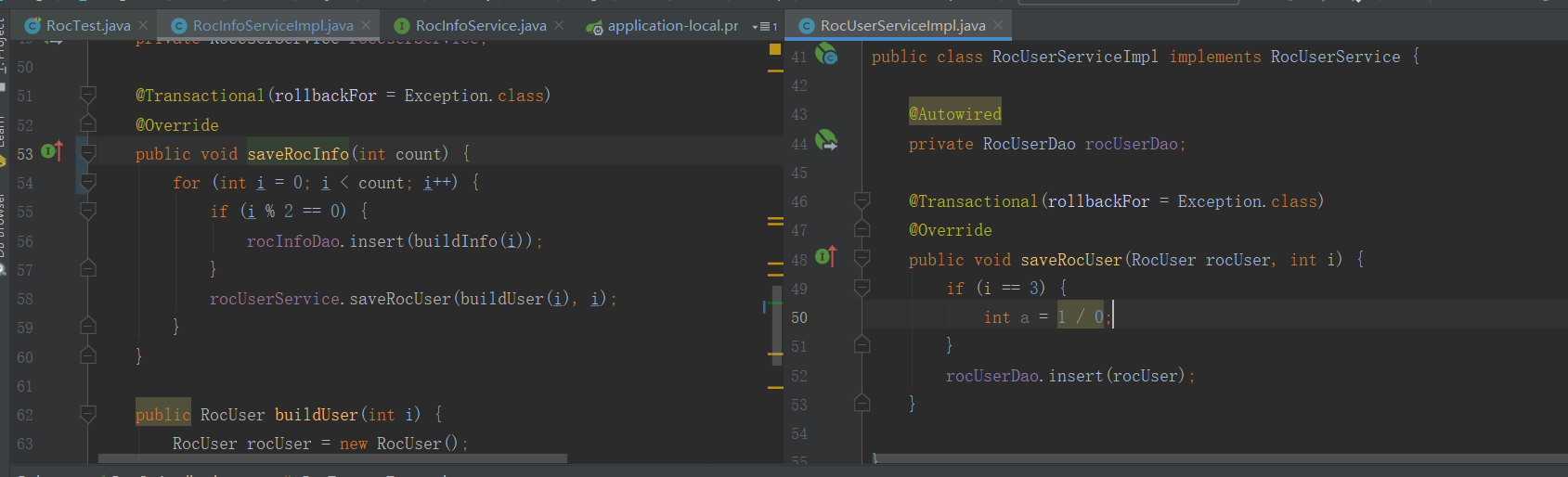
Spring默认的传播机制，能满足绝大部分业务需求，如果外层有事务，则当前事务加入到外层事务，一块提交，一块回滚。如果外层没有事务，新建一个事务执行

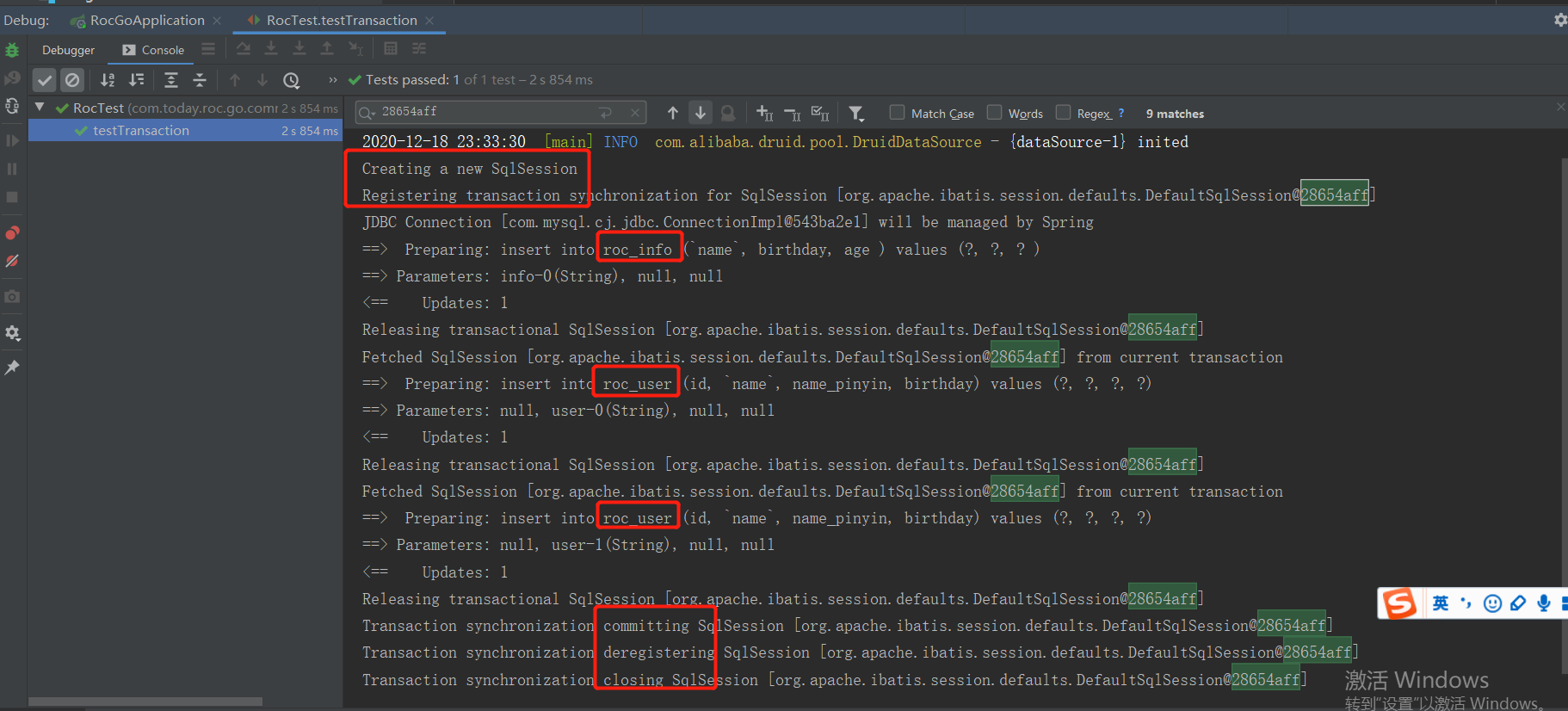
RocInfoService开启事务、RocUserService 开启事务

RocInfoService循环调用rocUserService新增RocUser方法

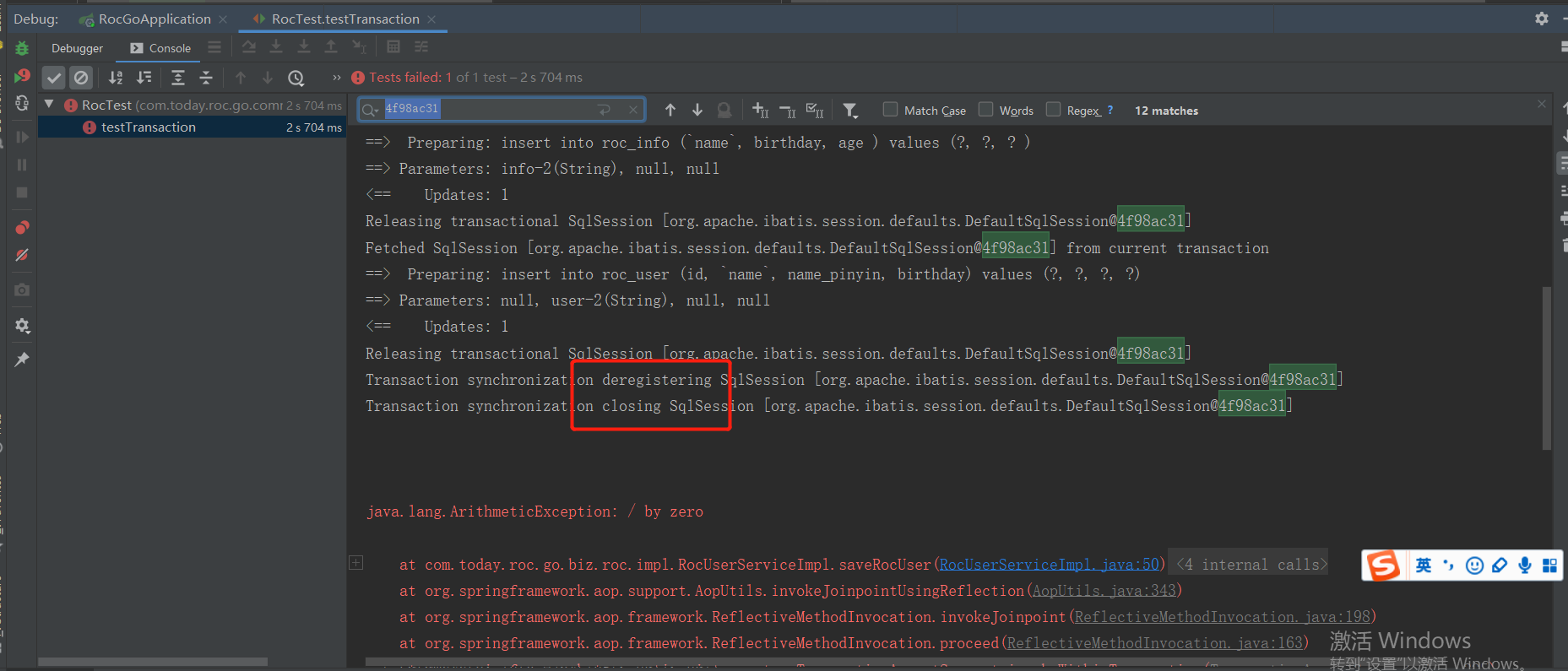
事务是嵌套的

Count = 2

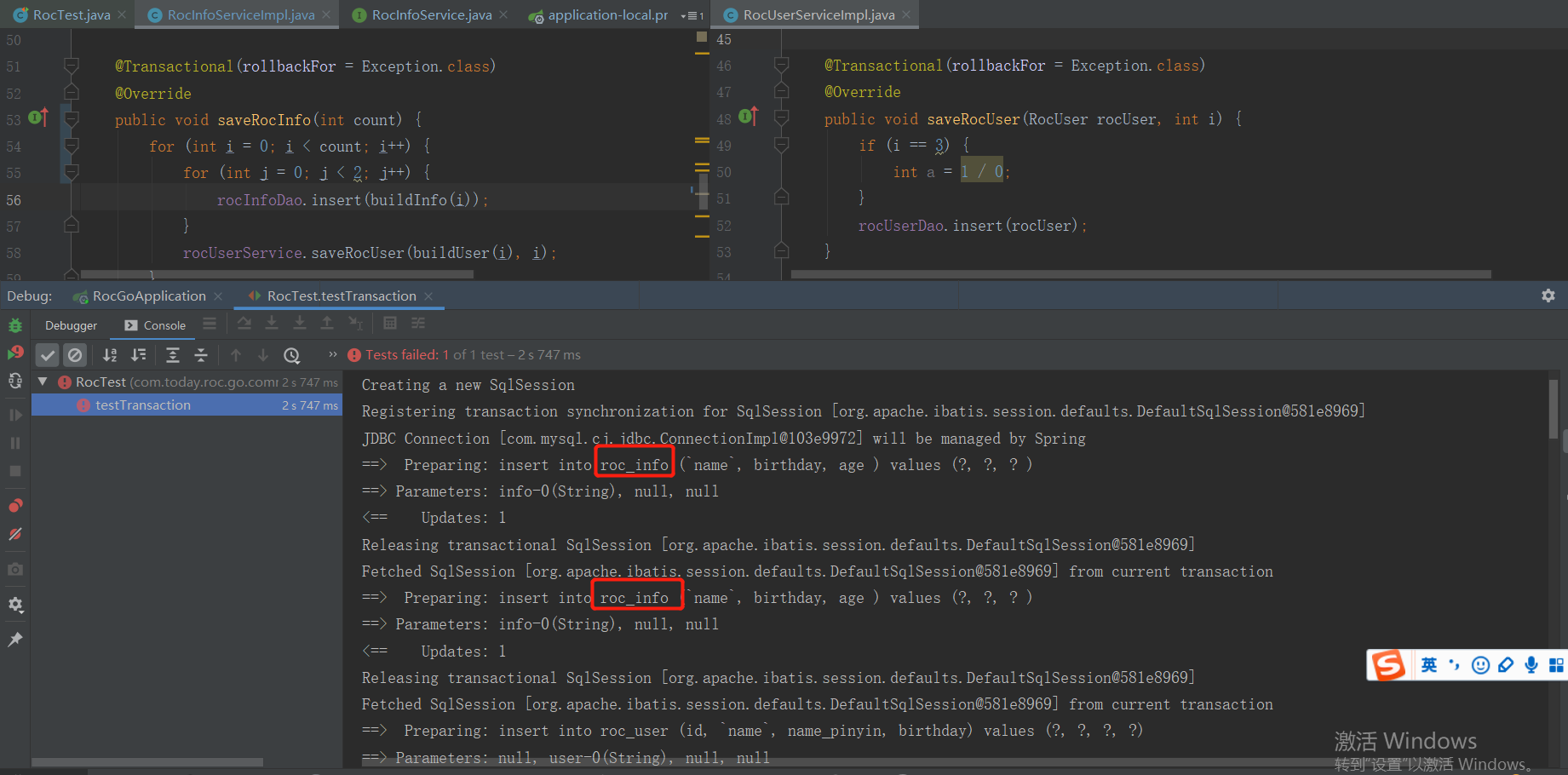




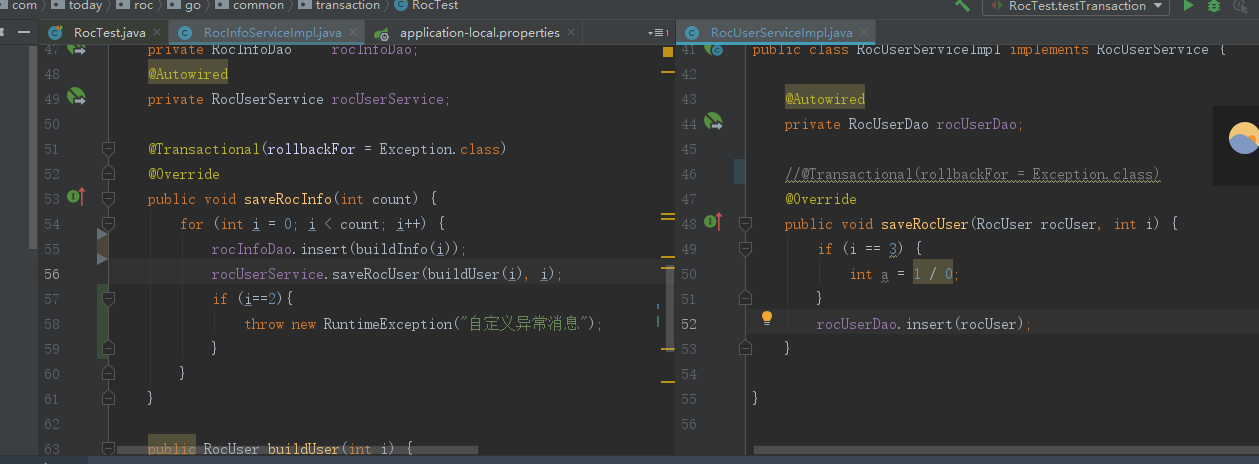
Count =4



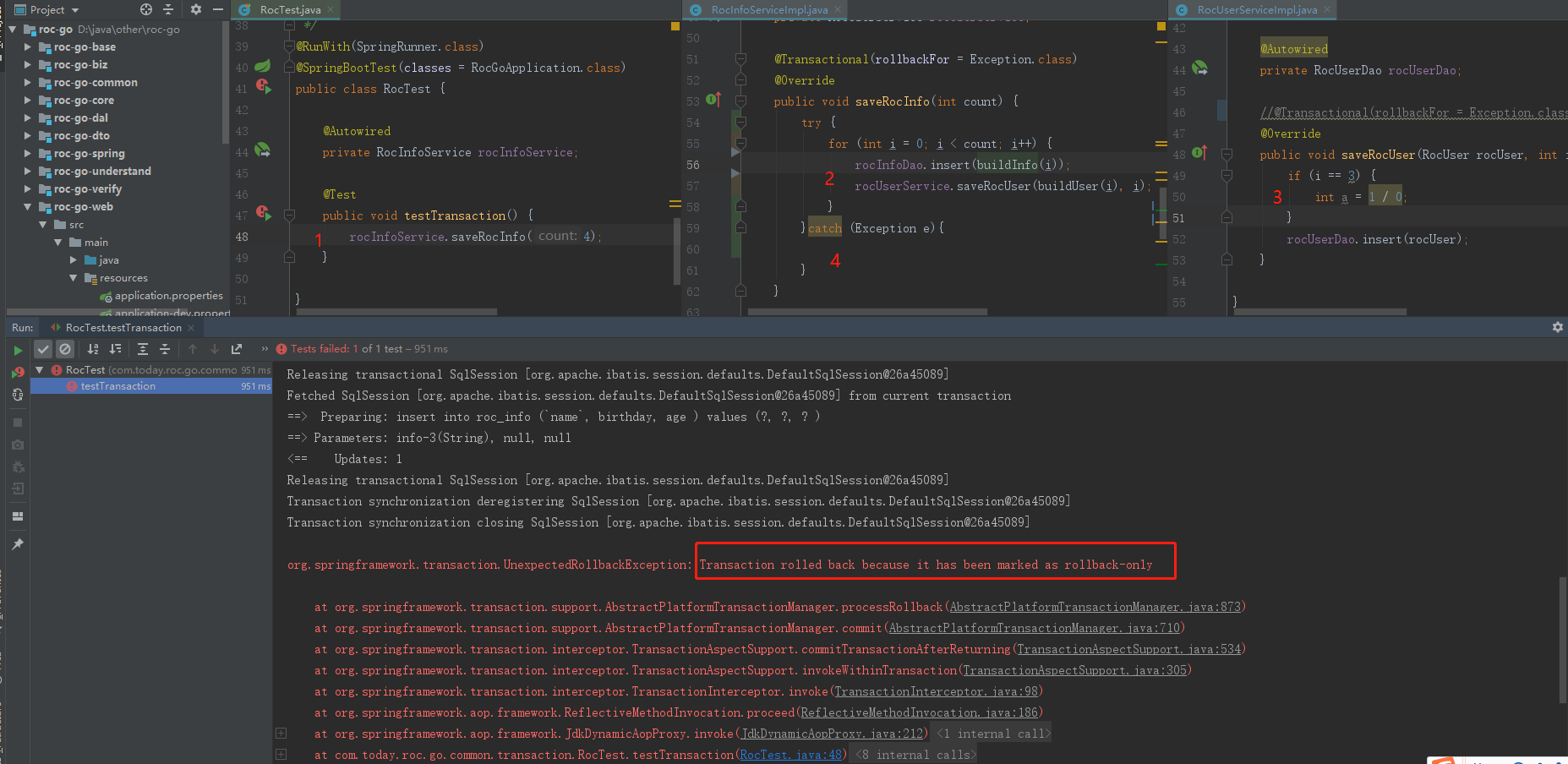
测试Fetched SqlSession



在外层RocInfoService抛出异常



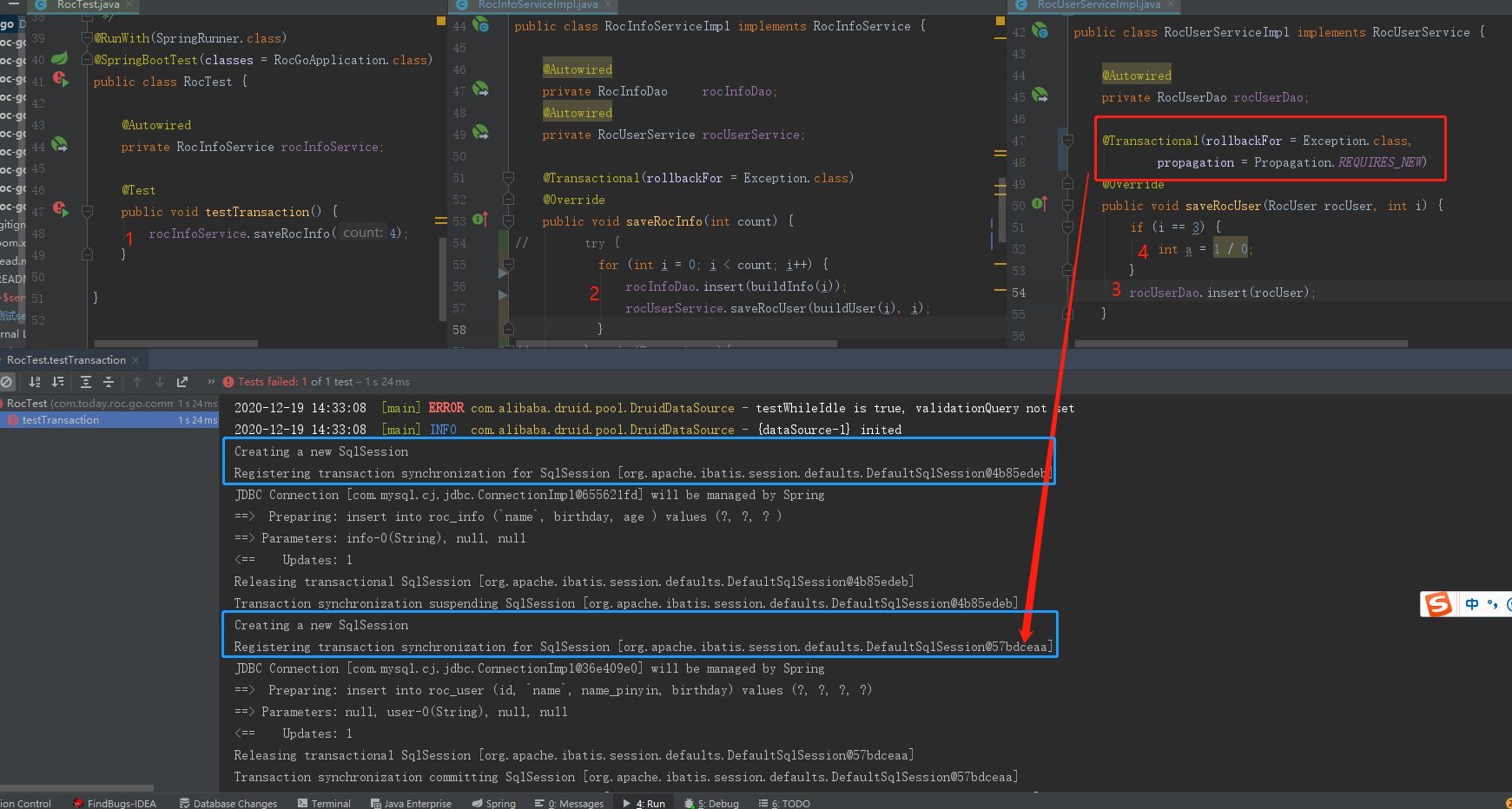
在嵌套事务场景中，内层事务的sql和外层事务的sql会在外层事务结束时进行提交或回滚。如果内层事务抛出异常e，在内层事务结束时，spring会把事务标记为“rollback-only”。这时如果外层事务捕捉了异常e，那么外层事务方法还会继续执行代码，直到外层事务也结束时，spring发现事务已经被标记为“rollback-only”，但方法却正常执行完毕了，这时spring就会抛出“org.springframework.transaction.UnexpectedRollbackException: Transaction rolled back because it has been marked as rollback-only”。

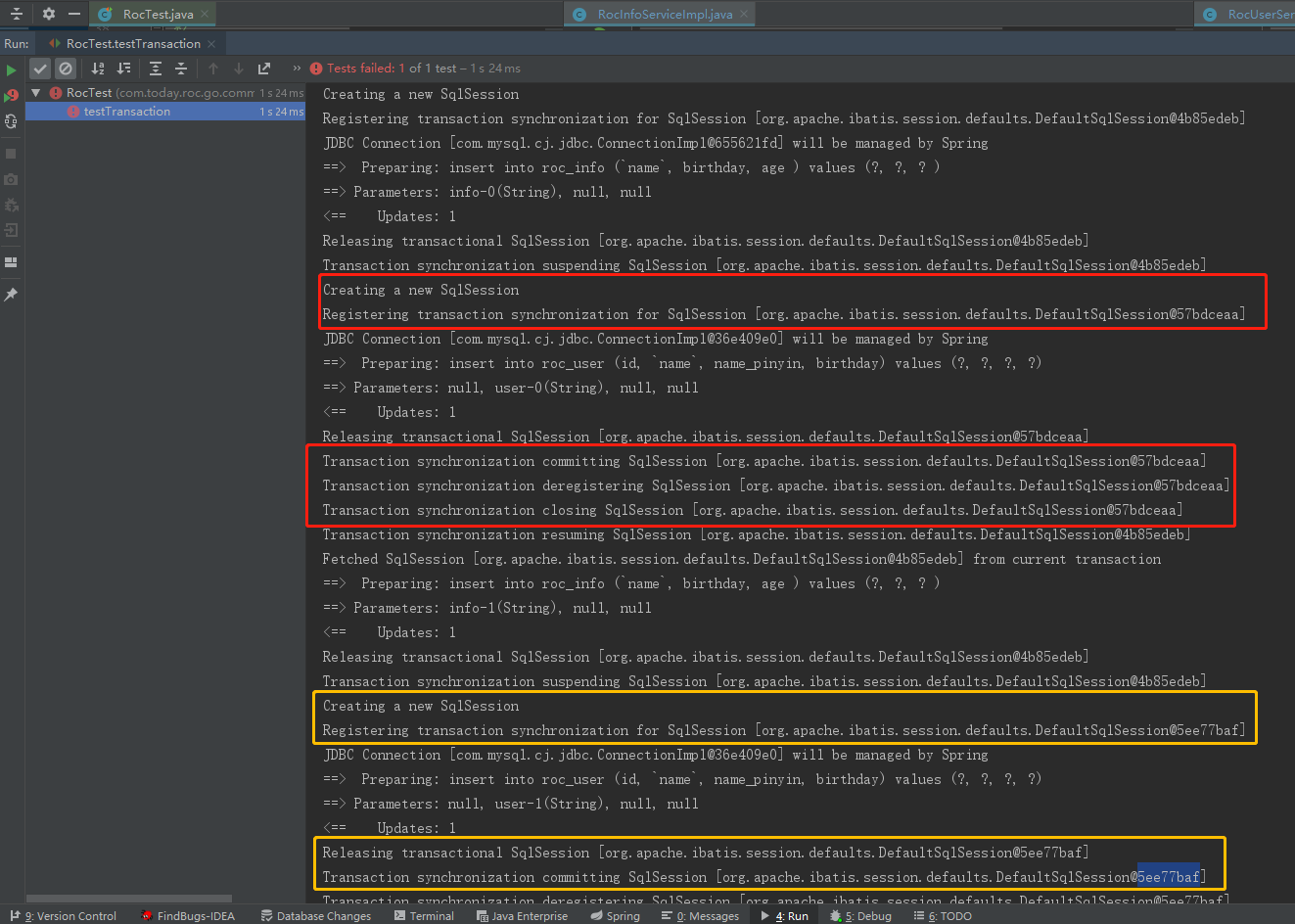


1. **Propagation.REQUIRES\_NEW**

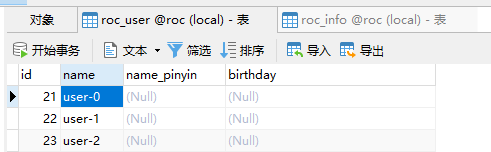
该事务传播机制是每次都会新开启一个事务，同时把外层事务挂起，当当前事务执行完毕，恢复上层事务的执行。如果外层没有事务，执行当前新开启的事务即可

RocUserService每次执行开启一个新事务，每次执行完都commit事务，如果内层事务报错未捕获，外层事务回滚



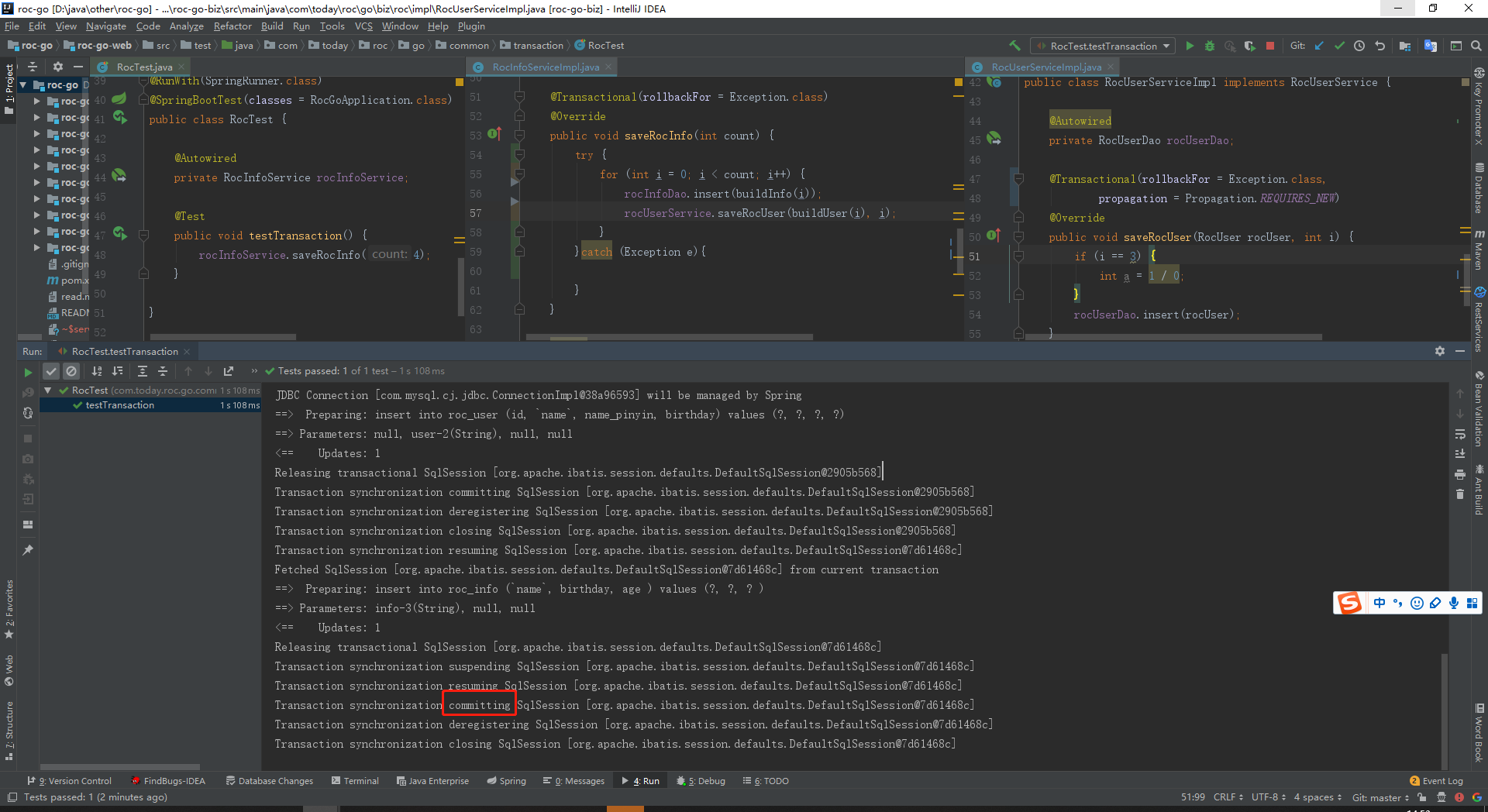


增加三条数据，第四条报错

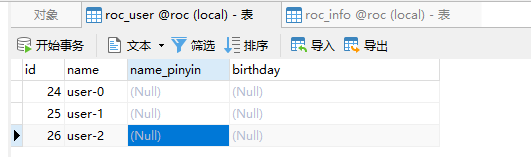


外层数据roc\_info为空

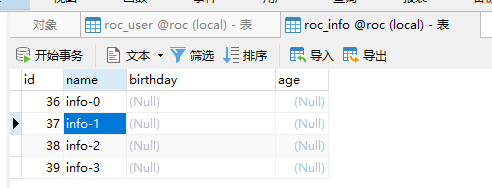
内层事务报错会回滚，如果外层捕获异常，外层事务不回滚，并插入数据



新增三条数据



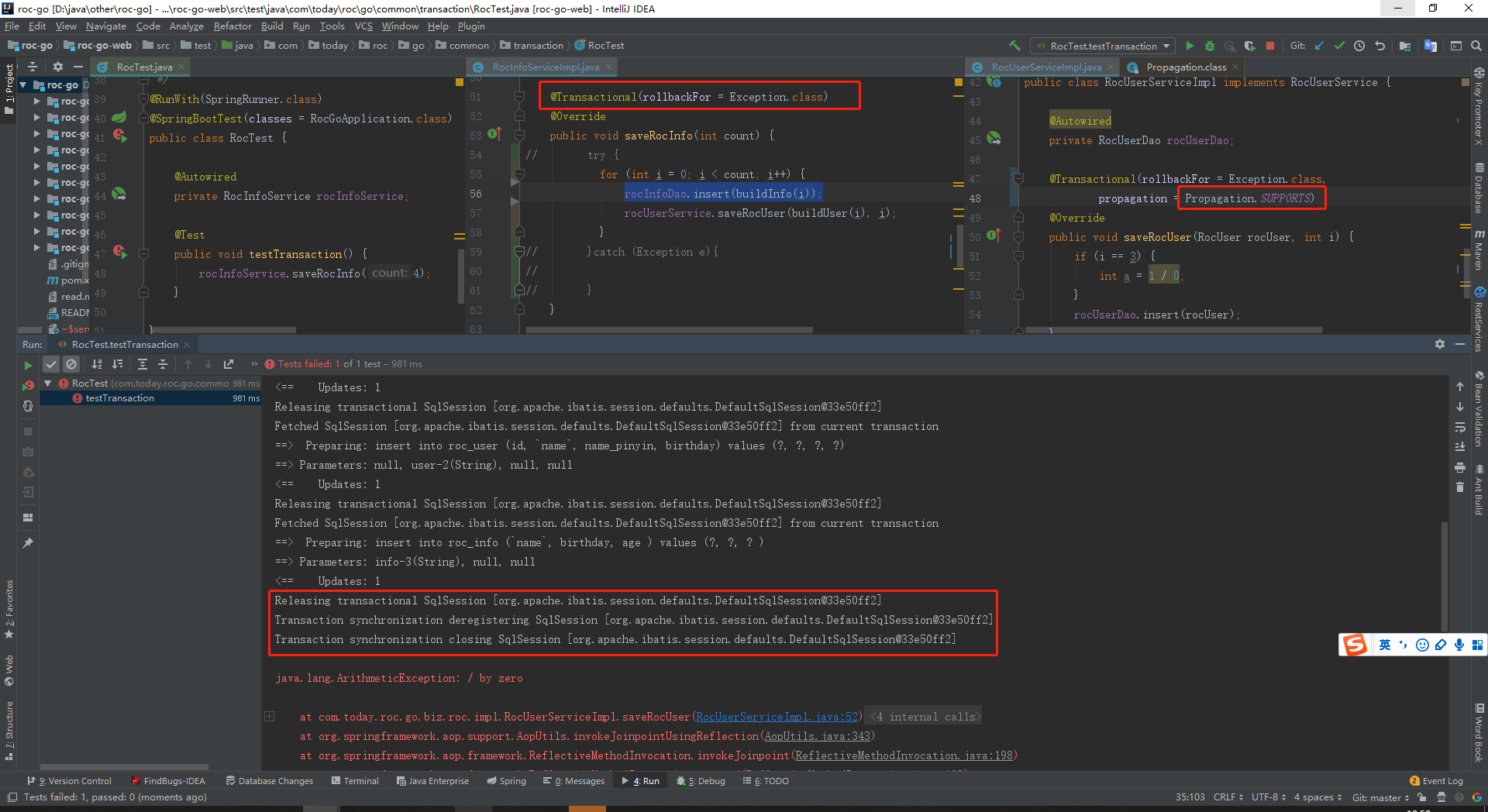
新增4条数据



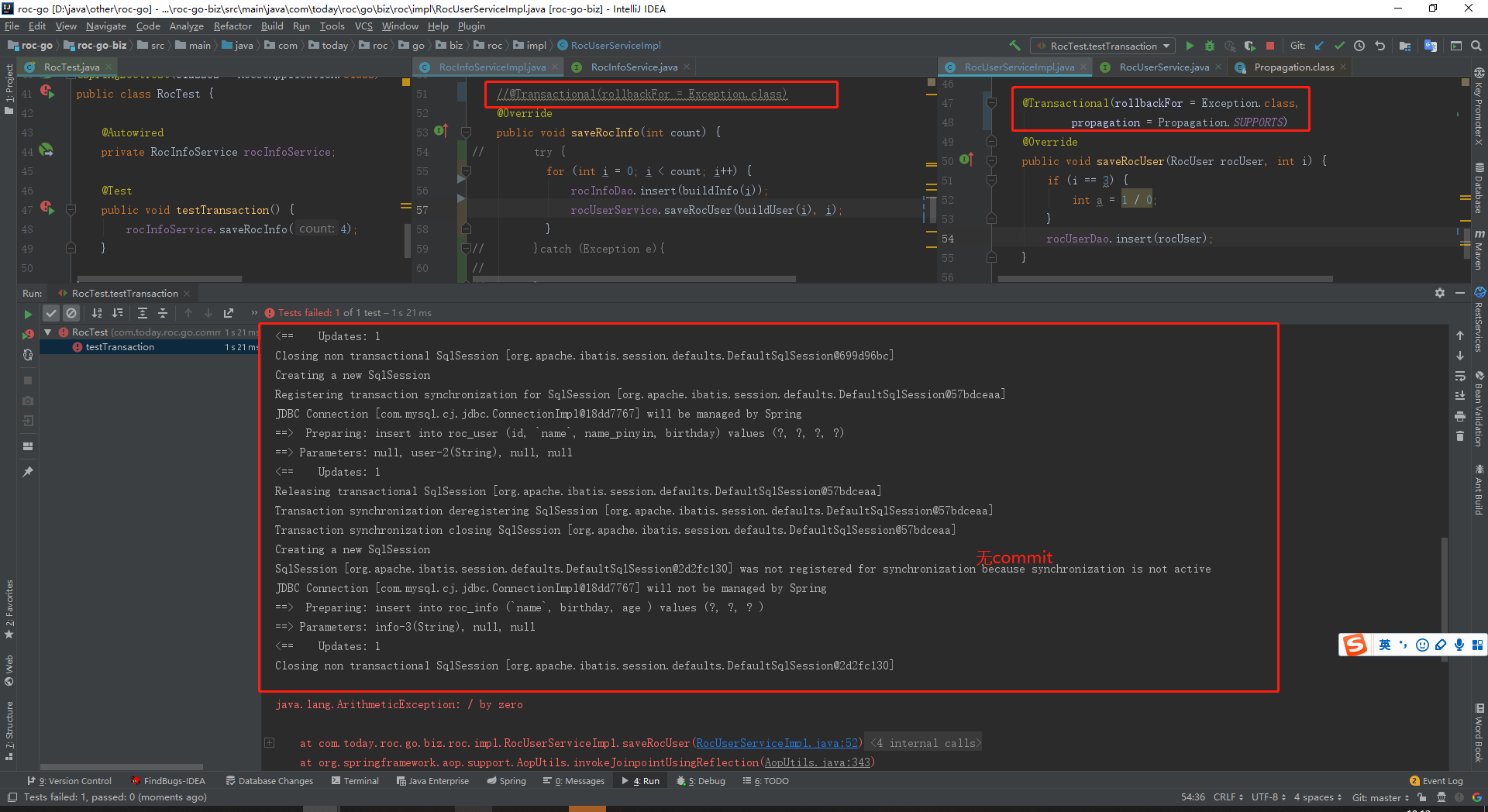
**3、Propagation.SUPPORTS**

如果外层有事务，则加入外层事务，如果外层没有事务，则直接使用非事务方式执行。完全依赖外层的事务

外层有事务，内层使用外层业务



外层没有事务，内层事务事务不作用



4、