

# 第45期-事务隔离级别与死锁

## 显式事务与隐式事务

SQL Server 显式事务与隐式事务

SQL Server 事务类型

## 事务相关方法

### DbTransaction 或 IDbContextTransaction

Commit 和 Rollback

### context.Database.XXX

BeginTransaction、BeginTransactionAsync、CommitTransaction、CurrentTransaction、RollbackTransaction 和 UseTransaction 方法。

```
if (!context.Blogs.Any(b => b.Url.Length > 10))
{
    context.Database.RollbackTransaction();
}
```

## 四种隔离级别

01: Read uncommitted(读未提交): 最低级别, 任何情况都会发生。

02: Read Committed(读已提交): 可避免脏读的发生。

03: Repeatable read(可重复读): 可避免脏读、不可重复读的发生。

04: Serializable(串行化): 避免脏读、不可重复读、幻读的发生。

注: 四种隔离级别: Serializable 级别最高, 最低的是 Read uncommitted 级别, 级别越高, 执行效率就越低, 隔离级别的设置只对当前链接有效, 对 .NET 操作数据库来说, 一个 Connection 对象相当于一个链接。

01: MySQL 数据库的默认隔离级别是 Repeatable read 级别。

02: Oracle 数据库中, 只支持 Serializable 和 Read committed 级别, 默认的是 Read committed 级别。

03: SQL Server 数据库中, 默认的是 Read committed 级别。

## 四种隔离级别场景举例

01: Read uncommitted 读未提交, 公司发工资了, 领导把 5000 元打到老断的账号上, 但是该事务并未提交, 而老断正好去查看账户, 发现工资已经到账, 是 5000 元整, 非常高兴。可是不幸的是, 领导发现发给老断的工资金额不对, 应该是 2000 元, 于是迅速回滚了事务, 修改金额为 2000 后, 将事务提交, 最后老断实际的工资只有 2000 元, 老断空欢喜一场, 场面十分尴尬, 假如老段是神速, 在事务回滚前取走了 5000 元, 事情就更加难办了, 还得去公司财务退钱。

02: Read committed 读已提交，老断拿着工资卡去买衣服，系统读取到卡里确实有 2000 元，而此时他的媳妇也正好在网上转账，把老断工资卡的 2000 元转到另一账户，并在老断完成付款之前提交了事务，当老断扣款时，系统检查到老断的工资卡已经没钱，扣款失败，老断十分纳闷，就短短几毫秒，明明卡里有钱，怎么突然就没钱了，思考了半天，尴尬离场。

03: Repeatable read 重复读，当老断拿着工资卡去消费时，一旦系统开始读取工资卡信息（即事务开始），老断的媳妇就不可能对该记录进行修改，也就是老断的媳妇不能在此时转账。

04: 重复读可能出现幻读：老断的媳妇工作在银行部门，她时常通过银行内部系统查看老断的信用卡消费记录。有一天，她正在查询到老断当月信用卡的总消费金额为 80 元，而老断此时正好吃完火锅在收银台买单，消费 1000 元，即新增了一条 1000 元的消费记录，并提交了事务，老断的媳妇准备回家问问老断这 80 元是干啥花掉的，于是乎她将老断当月信用卡消费的明细打印到 A4 纸上，却发现消费总额为 1080 元，老断的老婆很诧异，刚刚查出来不是 80 元嘛，以为出现了幻觉，幻读就这样产生了。

Serializabale：最高的事务隔离级别，代价花费最高，性能很低，很少使用，在此级别下，事务顺序执行，以避免上述所有产生的情况。

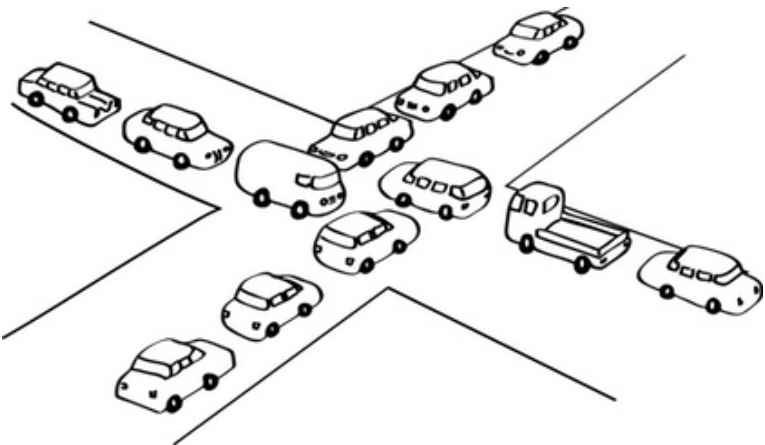
上面故事都是虚构的，请别信以为真，老断有媳妇是假设的，别当真，老断这种月薪 2000 的农村普通程序员是不可能

有媳妇的。

来自 <<https://www.cnblogs.com/phpfeng/p/9333829.html>>

SQL Server 事务隔离级别详解

数据库事务死锁



活动与监视器

概述
进程
资源等待
数据文件 I/O
最近耗费大量资源的查询
活动的耗费大量资源的查询

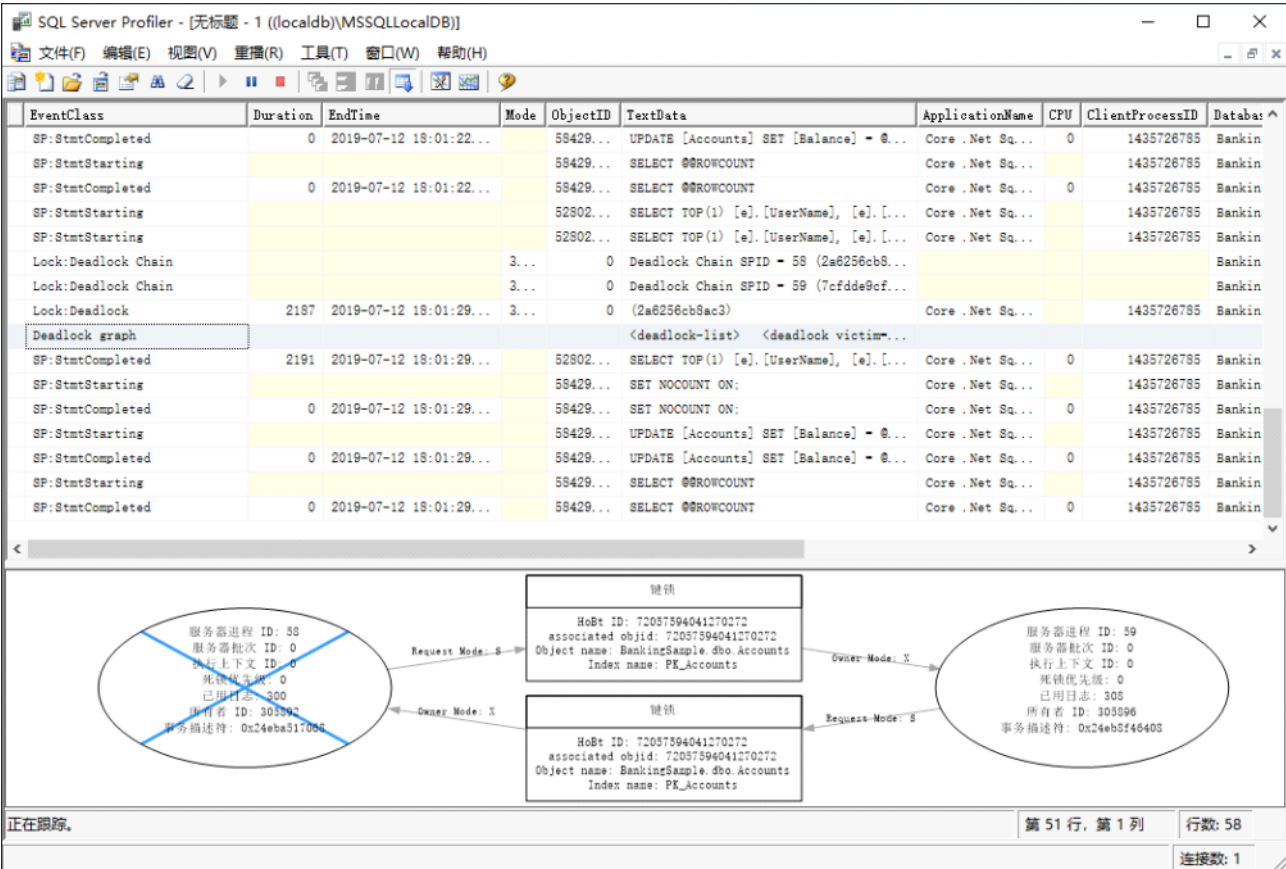
# sys.dm\_tran\_locks

向锁管理器发出的已授予锁或正等待授予锁的每个当前活动请求分别对应一行。

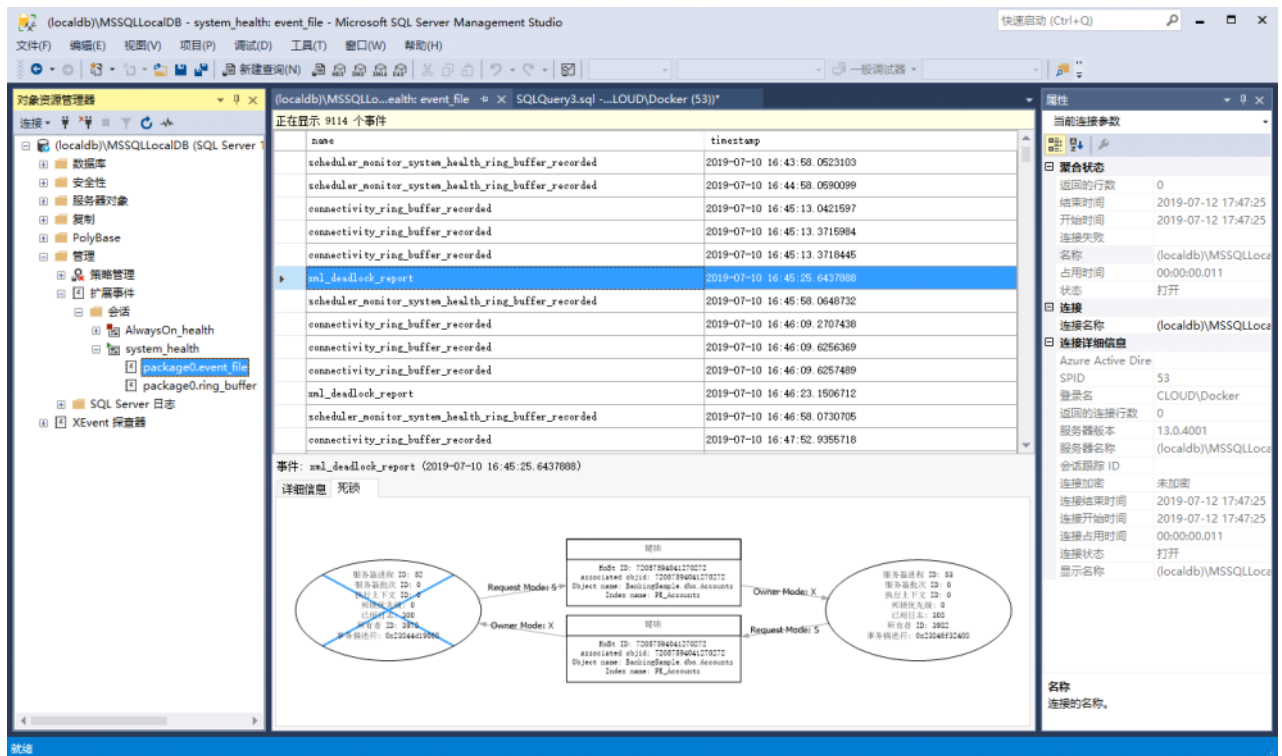
结果集中的列大体分为两组：资源组和请求组。资源组说明正在进行锁请求的资源，请求组说明锁请求。

<https://docs.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-locks-transact-sql>

## SQL Server Profiler



## 通过系统健康状况



## KILL

结束基于会话 ID 的用户进程，如果指定的会话 ID 有许多工作要撤消，KILL 语句可能需要一段时间才能完成，特别是在进程涉及回滚长事务时，此过程可能需要更长的时间才能完成。

<https://docs.microsoft.com/zh-cn/sql/t-sql/language-elements/kill-transact-sql>

## sp\_who

提供有关当前用户、会话和进程的实例中的信息 Microsoft SQL Server 数据库引擎。可以筛选信息以便只返回那些属于特定用户或特定会话的非空闲进程。

<https://docs.microsoft.com/zh-cn/sql/relational-databases/system-stored-procedures/sp-who-transact-sql>