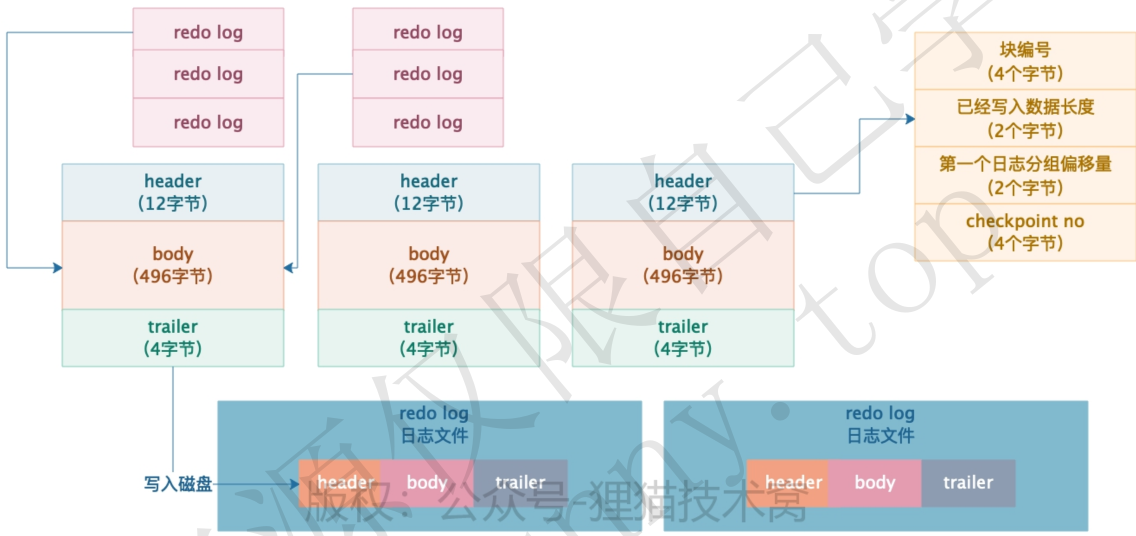
45 如果事务执行到一半要回滚怎么办？再探undo log回滚日志原理！

之前我们已经给大家深入讲解了在执行增删改操作时候的redo log的重做日志原理，其实说白了，就是你对buffer pool里的缓存页执行增删改操作的时候，必须要写对应的redo log记录下来你做了哪些修改

如下图所示，redo log都是先进入redo log buffer中的一个block，然后事务提交的时候就会刷入磁盘文件里去。

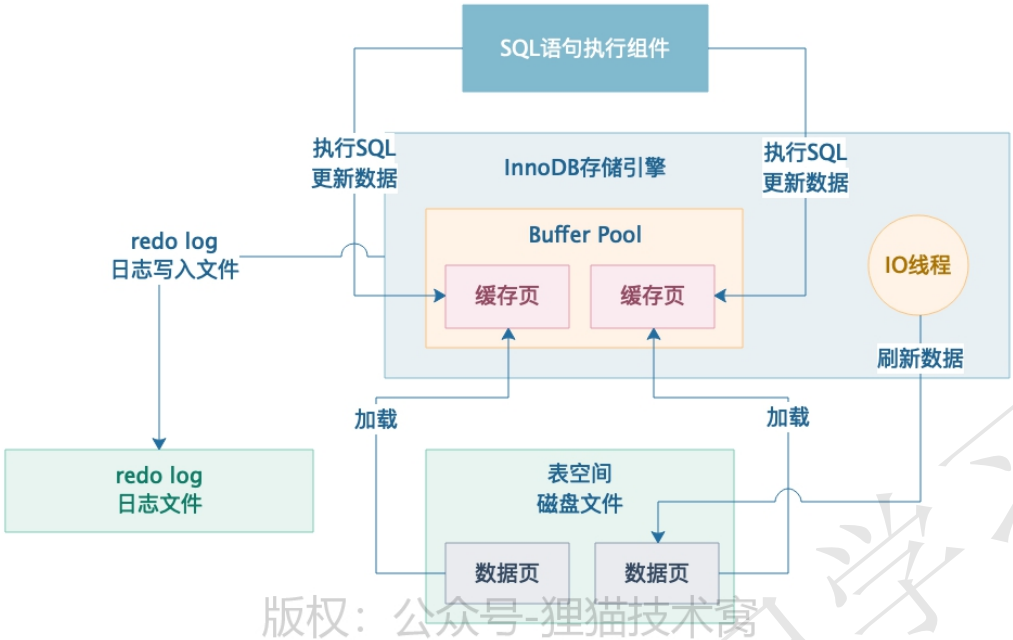


这样万一要是你提交事务了，结果事务修改的缓存页还没来得及刷入磁盘上的数据文件，此时你MySQL关闭了或者是宕机了，那么buffer pool里被事务修改过的数据就全部都丢失了！

但是只要有redo log，你重启MySQL之后完全是可以把那些修改了缓存页，但是缓存页还没来得及刷入磁盘的事务，他们所对应的redo log都加载出来，在buffer pool的缓存页里重做一遍，就可以保证事务提交之后，修改的数据绝对不会丢！

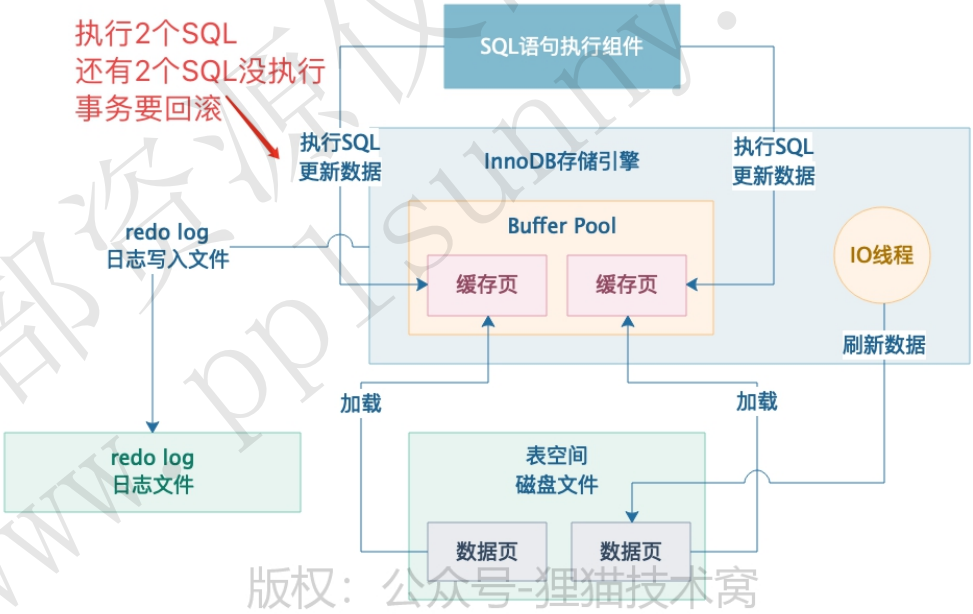
相信之前讲解了redo log日志之后，大家对这块都理解的更加深刻了，那么今天我们就带着大家来探索另外一种日志，就是undo log日志，也就是回滚日志，这种日志要应对的场景，就是事务回滚的场景！

那么首先大家先思考一个问题，假设现在我们一个事务里要执行一些增删改的操作，那么必然是先把对应的数据页从磁盘加载出来放buffer pool的缓存页里，然后在缓存页里执行一通增删改，同时记录redo log日志，对吧？如下图。



但是现在问题来了，万一要是一个事务里的一通增删改操作执行到了一半，结果就回滚事务了呢？

比如一个事务里有4个增删改操作，结果目前为止已经执行了2个增删改SQL了，已经更新了一些buffer pool里的数据了，但是还有2个增删改SQL的逻辑还没执行，此时事务要回滚了怎么办？看图



这个时候就很尴尬了，如果你要回滚事务的话，那么必须要把已经在buffer pool的缓存页里执行的增删改操作给回滚了

但是怎么回滚呢？毕竟无论是插入，还是更新，还是删除，该做的都已经做了啊！

所以在执行事务的时候，才必须引入另外一种日志，就是undo log回滚日志

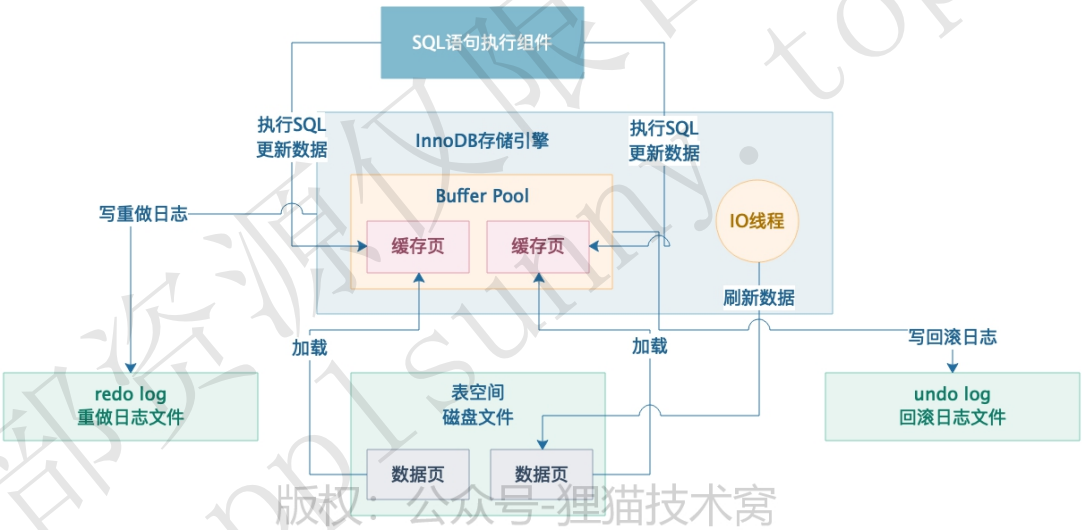
这个回滚日志，他记录的东西其实非常简单，比如你要是在缓存页里执行了一个insert语句，那么此时你在undo log日志里，对这个操作记录的回滚日志就必须是有一个主键和一个对应的delete操作，要能让你把这次insert操作给回退了。

那么比如说你要是执行的是delete语句，那么起码你要把你删除的那条数据记录下来，如果要回滚，就应该执行一个insert操作把那条数据插入回去。

如果你要是执行的是update语句，那么起码你要把你更新之前的那个值记录下来，回滚的时候重新update一下，把你之前更新前的旧值给他更新回去。

如果你要是执行的是select语句呢？不好意思，select语句压根儿没有在buffer pool里执行任何修改，所以根本不需要undo log！

好，所以我们来看下图，其实你在执行事务期间，之前我们最开始的几篇文章就讲过，你除了写redo log日志还必须要写undo log日志，这个undo log日志是至关重要的，没有他，你根本都没办法回滚事务！



明天我们继续来看看insert、delete和update几种操作的undo log到底长什么样，相信大家看完了，就会对undo log这块机制有一个更加深刻的理解了。

End