问题：业务中出现行锁记录无法立即解决。

* 解决方案：A.应急方法：show processlist; kill掉出现问题的进程

B.根治方法：select \* from innodb\_trx 查看有是哪些事务占据了表资源。

         C.方法：设置MySQL锁等待超时 innodb\_lock\_wait\_timeout=50 ，autocommit=on

* 该类问题导致原因

据我分析,Mysql的 InnoDB存储引擎是支持事务的，事务开启后没有被主动Commit。导致该资源被长期占用，其他事务在抢占该资源时，因上一个事务的锁而导致抢占失败！因此出现 Lock wait timeout exceeded 。

详细解读

MySQL InnoDB存储引擎，实现的是基于多版本的并发控制协议，一些列概念不累赘。

并发控制中，读操作可以分成两类：快照读 (snapshot read)与当前读 (current read)。快照读，读取的是记录的可见版本 (有可能是历史版本)，不用加锁。当前读，读取的是记录的最新版本，并且，当前读返回的记录，都会加上锁，保证其他事务不会再并发修改这条记录。

快照读：简单的select操作，属于快照读，不加锁。(当然，也有例外，下面会分析)

select \* from table where ?;

当前读：特殊的读操作，插入/更新/删除操作，属于当前读，需要加锁。

select \* from table where ? lock in share mode;

select \* from table where ? for update;

insert into table values (…);

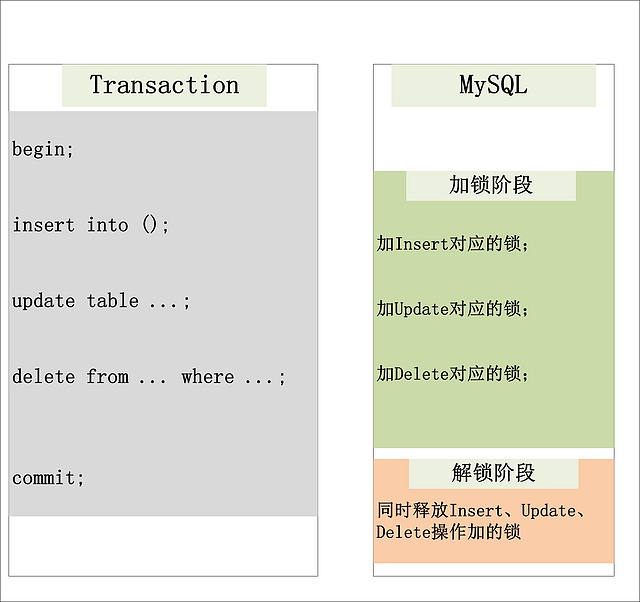
update table set ? where ?;

delete from table where ?;

所有以上的语句，都属于当前读，读取记录的最新版本。并且，读取之后，还需要保证其他并发事务不能修改当前记录，对读取记录加锁。其中，除了第一条语句，对读取记录加S锁 (共享锁)外，其他的操作，都加的是X锁 (排它锁)。

Cluster Index：聚簇索引（个人贬义的理解，有主键就是主键，没有主键就是唯一索引，没有唯一索引，隐形建立一条类似主键，保存一系列数据i/o操作，加快读取的速度），mysql官方解读：[Clustered and Secondary Indexes](http://dev.mysql.com/doc/refman/5.0/en/innodb-index-types.html)

2PL：Two-Phase Locking：锁操作分为两个阶段：加锁阶段与解锁阶段，并且保证加锁阶段与解锁阶段不相交。下面，仍旧以MySQL为例，来简单看看2PL在MySQL中的实现。



Isolation Level

隔离级别：Isolation Level，也是RDBMS的一个关键特性。对于4种隔离级别：Read Uncommited，Read Committed，Repeatable Read，Serializable，都有了深入的认识。本文不打算讨论数据库理论中，是如何定义这4种隔离级别的含义的，如何定义这4种隔离级别的。

MySQL/InnoDB定义的4种隔离级别：

Read Uncommited

可以读取未提交记录。此隔离级别，不会使用，忽略。

Read Committed (RC)

快照读忽略。

针对当前读，RC隔离级别保证对读取到的记录加锁 (记录锁)，存在幻读现象。

Repeatable Read (RR)

快照读忽略。

针对当前读，RR隔离级别保证对读取到的记录加锁 (记录锁)，同时保证对读取的范围加锁，新的满足查询条件的记录不能够插入 (间隙锁)，不存在幻读现象。

Serializable

从MVCC并发控制退化为基于锁的并发控制。不区别快照读与当前读，所有的读操作均为当前读，读加读锁 (S锁)，写加写锁 (X锁)。

Serializable隔离级别下，读写冲突，因此并发度急剧下降，在MySQL/InnoDB下不建议使用。

贴死锁情况图

