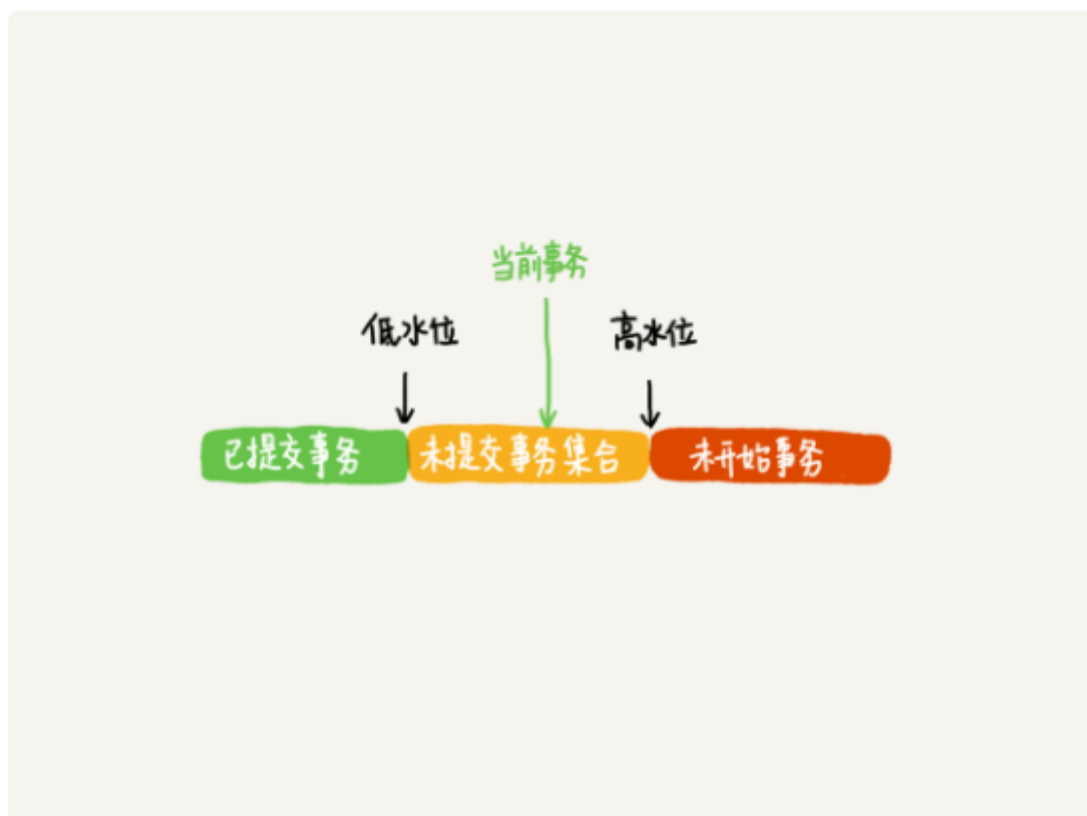


1."快照"在MVCC里的工作原理

mysql是基于一致性视图(read-view)来实现MVCC的，InnoDB里面每个事务都有一个唯一的id，transation id。它是在事务开始前向InnoDB申请的，申请过程中是加锁的，保证了transaction id是按照申请顺序递增的。每行数据都是有多个版本的，每次事务更新数据的时候就会生成一个新的版本，并且把事务的transaction id赋值给这个版本的事务ID，记为row trx_id。同时旧的数据保留，并且可以拿到它，这个就是undo log。每个事务启动时会保存一个事务数组，里面存了在这个事务启动瞬间当前已启动但未提交的事务id集合。数据里事务id最小的记为低水位，当前系统已创建过的事务ID的最大值+1记为高水位。这个数组就组成了当前事务的一致性视图(read-view)。



如图所示，对于当前启动的事务，一个数据版本的row trx_id就有以下几种情况：

- 小于低水位，处于绿色区域，就是已经提交过的事务，可见。
- 大于高水位，处于红色区域，就是未启动的事务，不可见。
- 落在黄色区域，那么有两种情况：1、在数组里，那么是未提交的事务，不可见。2、不在数组里，说明是已提交的事务，可见。

对于不可见的版本，当前事务会根据undo log去找历史版本，直到找到可见版本的数据，就取出来。这样就可以保证在整个事务执行期间读到的都是一致的数据。

2.更新逻辑

更新事务的做更新操作时，都是先读后写，读的数据都是当前值，称为**当前读(current read)**。当前读是为了避免更新事务在满足RR(Repeatable read)的隔离级别情况下丢失其他更新事务的修改记录的处理。如果把查询事务的sql加上锁，也是执行当前读。如果两条select语句分别加上了读锁(S锁，共享锁)和写锁(X锁，排他锁)。

```
select k from t where id=1 lock in share mode;  
select k from t where id=1 for update;
```

当前读保证了其他事务对同行数据的修改记录不会丢失，行锁保证了同一时刻只能有一个线程对同行记录进行操作，一致性视图保证了在整个事务执行期间，事务看到的数据都是一致的，他们共同构成了InnoDB的可重复读能力。RC(read committed,读提交)和RR的区别就是：

- RR是是事务开始就创建一致性视图，伴随整个事务的执行生命周期。
- RC是在每个sql语句执行前都会重新计算创建一致性视图。