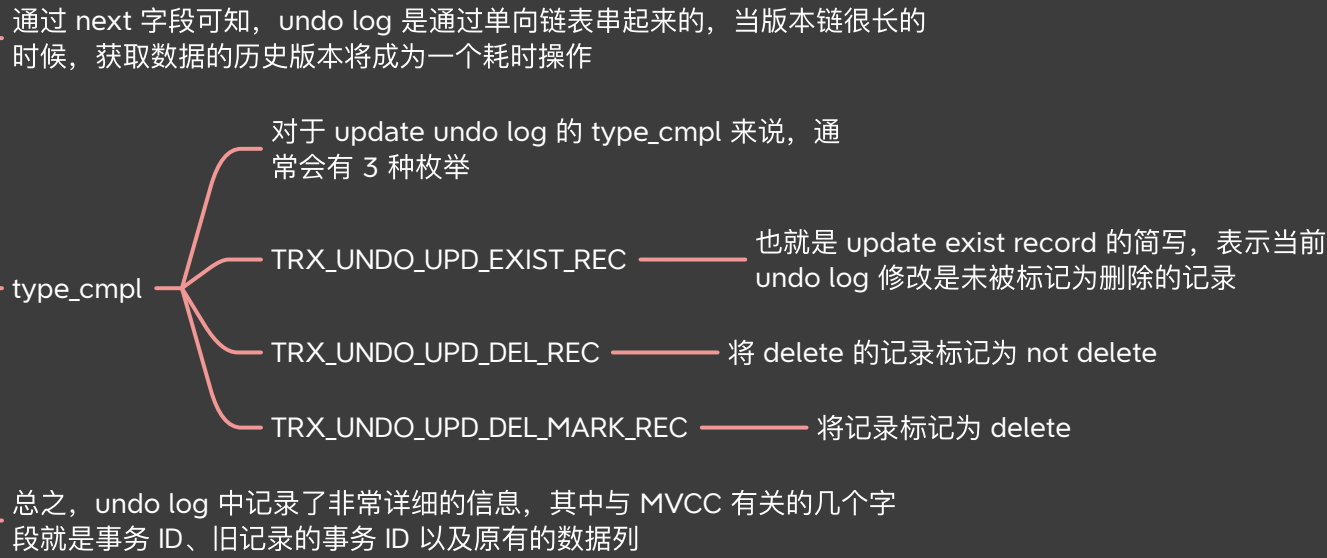
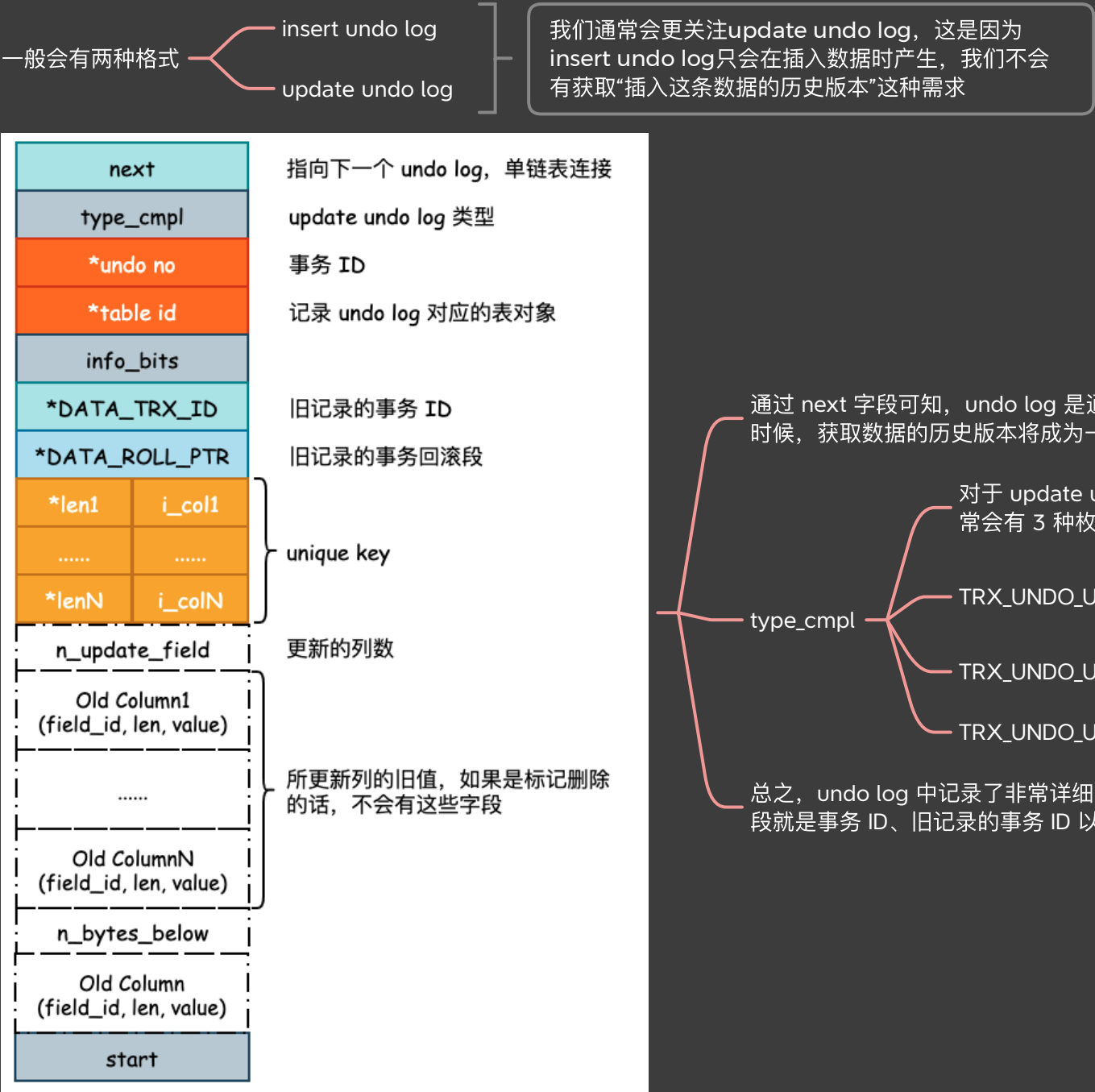


undo log与MVCC

基本概念

- undo log和bin log一样都属于逻辑日志，其作用是用来实现事务的回滚和MVCC，undo log在5.6之前保存在共享表空间的回滚段中，5.6之后undo表空间可以配置成独立的文件，但是提前需要在配置文件中配置，完成数据库初始化后生效且不可改变undo log文件的个数
- 当我们对数据进行修改的时候，除了会产生redo log，还会产生undo log；当事务执行失败或者是客户端要求回滚事务时，InnoDB就可以通过undo log逻辑性地将数据回滚到事务开始时候的样子
- 当事务回滚时，每个INSERT对应一个DELETE，每个DELETE会对应一个INSERT、对于UPDATE，InnoDB会执行一个相反的UPDATE。
- 另外，InnoDB通过undo log来实现MVCC，也就是把数据的历史版本也记录下来。只不过这些数据的历史版本是服务于InnoDB自身事务执行需要的，并不会永久的持久化
- etcd同样也实现了MVCC，并且我们可以随时查看某个数据的历史版本
- 事务的原子性是通过 undo log 实现的
- MVCC是有undo log实现的，事务的隔离性是由MVCC实现的

undo log 格式



MVCC

- 通过undo log中记录的值，我们就可以实现MVCC了，在MySQL中，MVCC主要作用是实现事务的隔离性，这里尤其指Repeated Read (RR) 可重复读事务隔离级别，同时也给一致性非锁定读提供了实现的可能
- 基本概念
  - MVCC , Multiversion Concurrency Control，即多版本并发控制，主要用来实现事务的隔离性以及一致性非锁定读，后这被称为快照读。通过快照读可以极大的提高数据库的并发性能
- 实现
  - 上图即为 InnoDB Compact 行记录格式，与 MVCC 相关的字段就是事务 ID 和回滚指针，回滚指针指向 undo log 中的记录
  - 当我们进行数据读取时，也会有一个事务 ID 产生，若某一行数据存在多个版本，那么当前事务只会去读取小于该事务 ID 的最新数据
  - MVCC 解决幻读也是同样的原理，在 RR 隔离级别下，读取范围数据时需要对事务 ID 进行比较，只要是超过了当前事务 ID 的数据行，就不在返回结果中进行记录

