隔离性：两个事务操作同一数据

事务1

1121a → 11

↓

改成

1131a

① 读未提交、Read Uncommit

设为读未提交

开始事务 → 就可以读取

不用提交

会读出脏数据，脏读

② 读已提交，(READ COMMITTED)

幻读，不可重复读

(两次重复读不一样)。

读 提交了更改 读

当一个事务插入，另一个读取

来时，出现幻读（两次不相

③ 可重复读 ☆

无论别的事务是否修改，

两次事务一致。

④ 串行化 （用的不多）

两个事务读十读可以

无法修改 右边不能

这几种都是SQL的标准定义，
但是不同数据库的实现
解决了这些问题

读已提交

例：

事务

A(id:100)　　B(id:200)

A读1　　B把1改为2.

则

不可重复读 → 未提交时 → 事务1 ... B读 ✓

原理：行格式中3个隐藏列 row_id → 行id

transaction_id ← 事务id

roll_pointer → 回滚指针

transaction_id 记录最近一次修改的事务的id，

Read View

活跃事务（没提交的事务） m_id：（81，82，200，300）

找到提交了的事务，根据 m_id 判断当不在其中时，取数据。

H3 ### ReadView

对于使用READ UNCOMMITTED隔离级别的事务来说，直接读取记录的最新版本就好了，对于使用SERIALIZABLE隔离级别的事务来说，使用加锁的方式来访问记录。对于使用READ COMMITTED和REPEATABLE READ隔离级别的事务来说，就需要用到我们上边所说的版本链了，核心问题就是：需要判断一下版本链中的哪个版本是当前事务可见的。

ReadView中主要包含4个比较重要的内容：

1. m_ids：表示在生成ReadView时当前系统中活跃的读写事务的事务id列表。
2. min_trx_id：表示在生成ReadView时当前系统中活跃的读写事务中最小的事务id，也就是m_ids中的最小值。
3. max_trx_id：表示生成ReadView时系统中应该分配给下一个事务的id值。
4. creator_trx_id：表示生成该ReadView的事务的事务id。