MySQL 下的 MVCC 模型

前言

MVCC 只工作在 REPEATABLE READ 和 READ COMMITED 隔离级别下。READ UNCOMMITED 每次都只能读到最新的版本。SERIABLABLE 通过加锁的方式使事务串行运行。

版本链

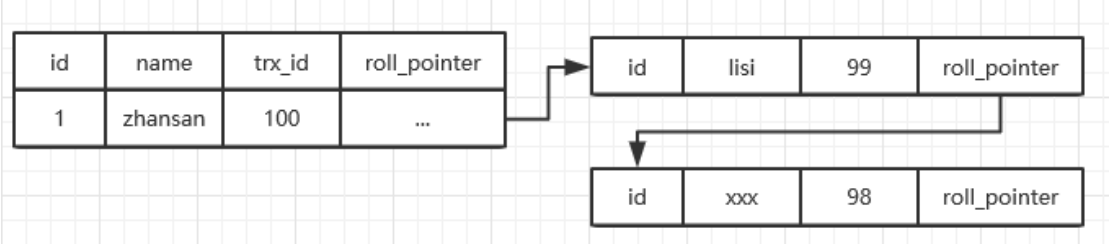
InnoDB 为每一行数据添加两个属性（在没有主键和非 NULL 唯一键的情况下还会存在一列隐藏列 row_id）：

- trx_id：每次对数据进行修改的时候就会将修改当前事务的 id 赋值给此列。
- roll_pointer：相当于一个指针，指向上个版本的记录。

例如：

1. A 事务 id 为 98，UPDATE user set name='xxx' WHERE id=1
2. B 事务 id 为 99，UPDATE user set name='lisi' WHERE id=1
3. C 事务 id 为 100，UPDATE user set name='zhansan' WHERE id=1

以上三个步骤执行以后形成下图的版本链。



ReadView

ReadView 记录了当前系统中未提交的事务，主要有一下三个成员：

- Low_trx_id：表示当前事务链中的最大事务 id。
- Up_trx_id：表示当前事务链中的最小事务 id。
- Trx_ids：事务链中所有事务的 id 集合。

访问某条记录时，只需要按照下边的步骤判断记录的某个版本是否可见：

- 如果被访问版本的 trx_id 属性值小于 Up_trx_id，表明生成该版本的事务在生成 ReadView 前已经提交，所以该版本可以被当前事务访问。
- 如果被访问版本的 trx_id 属性值大于 Low_trx_id, 表明生成该版本的事务在生成 ReadView 后才生成，所以该版本不可以被当前事务访问。
- 如果被访问版本的 trx_id 属性值在 m_ids 列表中最大的事务 id 和最小事务 id 之间，那就需要判断一下 trx_id 属性值是不是在 m_ids 列表中，如果在，说明创建 ReadView 时生成该版本的事务还是活跃的，该版本不可以被访问；如果不在，说明创建 ReadView 时生成该版本的事务已经被提交，该版本可以被访问。

Mysql 下的 MVCC 模型的实现

1. READ COMMITED

   1) Transaction 100 执行 UPDATE user set name='lisi' WHERE id=1

   2) Transaction 200 执行 UPDATE user set name='张三' WHERE id=1

   情况一：

   步骤一和二执行但未提交时执行 select * from user where id=1，此时

ReadView 的结构为{100,200,[100,200]}，最小事务 id 为 100，最大事务 id 为 200。此时 id 为 1 的这条数据 trx_id 的值为 200，因为 trx_ids 包含 200，所以此版本是读不到的。以此类推 100 也读不到，只能读到 100 上一个版本的值。

情况二：

步骤一和二执行但一已提交二未提交时执行 select * from user where id=1，此时 ReadView 的结构为{200,200,[200]}，最小事务 id 为 200，最大事务 id 为 200。此时 id 为 1 的这条数据 trx_id 的值为 200，因为 trx_ids 包含 200，所以此版本是读不到的。但是此时 trx_ids 不包含 100，说明 id 为 100 的事务已经提交，所以可以读到。

2. REPEATABLE READ

与 READ COMMITTED 不同的是 REPEATABLE READ 只在第一次读的时候生成 ReadView，而 READ COMMITTED 是在每次读的时候都生成 ReadView。

继续沿用以上的步骤：

情况一：

步骤一和二执行但未提交时执行 select * from user where id=1，此时 ReadView 的结构为{100,200,[100,200]}，最小事务 id 为 100，最大事务 id 为 200。此时 id 为 1 的这条数据 trx_id 的值为 200，因为 trx_ids 包含 200，所以此版本是读不到的。以此类推 100 也读不到，只能读到 100 上一个版本的值。

情况二：

步骤一和二执行但一已提交二未提交时执行 select * from user where id=1，

此时 ReadView 的结构依然为第一次查询时生成的{100,200,[100,200]}，最小事务 id 为 100，最大事务 id 为 200。此时 id 为 1 的这条数据 trx_id 的值为 200，因为 trx_ids 包含 200，所以此版本是读不到的。以此类推 100 也读不到，只能读到 100 上一个版本的值。

总结

- 在 MVVC 模型的情况下 REPEATABLE READ 隔离级别可以避免幻读的产生。
- 优点在于多大多数读来说不需要加锁，速度更快，缺点是需要为每一行存储更多的数据