

1. 在Repeatable Read (RR) 的隔离级别下, Innodb使用MVVC和next-key locks解决幻读, MVVC解决的是普通读(快照读)的幻读, next-key locks解决的是当前读情况下的幻读。

2. 幻读是什么

事务A, 先执行:

```
update table set name= "hh" where id>3;
```

结果为:

OK row xx 表名成功影响多少行数据

事务B, 后执行, 并且提交:

```
insert into table values(11, uu);
```

```
commit;
```

事务A, 然后再select一下:

```
select * from table where id>3
```

结果集为:

...

11, uu

...

事务A懵了, 我特么不是id>3全部更新了吗

这次是已提交事务B对事务A产生的影响, 这个影响叫做“幻读”。

幻读和不可重复读的区别是, 前者是一个范围, 后者是本身(在一个事务中多次读取同一条数据, 出现结果不一致)

3. 怎么解决的?

3.1. 当前读

所谓当前读, 指的是加锁的select(S或者X), update, delete等语句。在RR的事务隔离级别下, 数据库会使用next-key locks来锁住本条记录以及索引区间。

拿上面那个例子来说, 在RR的情况下, 假设使用的是当前读, 加锁了的读

`select * from table where id>3` 锁住的就是id=3这条记录以及id>3这个区间范围，锁住索引记录之间的范围，避免范围间插入记录，以避免产生幻影行记录。

3.2. 普通读

因为普通读是不会加锁的读，故不会有next-key locks的使用，解决幻读的手段是MVVC

mvvc会给每行元组加一些辅助字段，记录创建版本号和删除版本号。

而每一个事务在启动的时候，都有一个唯一的递增的版本号。每开启一个新事务，事务的版本号就会递增。

默认的隔离级别（REPEATABLE READ）下，增删查改变成了这样：

SELECT

读取创建版本小于或等于当前事务版本号，并且删除版本为空或大于当前事务版本号的记录。这样可以保证在读取之前记录是存在的

INSERT

将当前事务的版本号保存至行的创建版本号

UPDATE

新插入一行，并以当前事务的版本号作为新行的创建版本号，同时将原记录行的删除版本号设置为当前事务版本号

DELETE

将当前事务的版本号保存至行的删除版本号

比如我插入一条记录，事务id 假设是1，那么记录如下：也就是说，创建版本号就是事务版本号。

id	name	createversion	deleteversion
1	wxt	1	

如果我更新的话，事务id假设是2

id	name	createversion	deleteversion
1	wxt	1	2
1	taotao	2	

这里是把name更新为taotao，原来的元组deleteversion版本号为这个事务的id，并且新增一条

如果我删除的话，假设事务是id=3

id	name	createverison	deleteversion
1	taotao	2	3

重点来了

现在我读取的话，必须同时满足两个条件的：

- 读取创建版本小于或等于当前事务版本号 这意味着数据在这个事务之前被创建
- 删除版本为空或大于当前事务版本号的记录。 这意味着删除操作在这个事务之后发生

假设事务A的id=10（当前事务A的版本号）

现在update table set name= “hh” where id>3;执行这条语句

id	name	createverison	deleteversion
4	a	2	10
5	b	5	10
4	hh	10	
5	hh	10	

事务B的id=11（事务B的版本号）

insert into table values(11, uu);

id	name	createverison	deleteversion
4	a	2	10
5	b	5	10
4	hh	10	
5	hh	10	
11	uu	11	

select * from table where id>3

根据上述的规则，

读取创建版本号小于等于当前事务的→那么（4，a）（5，b）（4，hh）（5，hh）

上面规则的输出作为下面规则的输入的话

删除版本为空或大于当前事务版本号的记录→（4，hh）（5，hh）

如此读取就没有读取到事务B新插入的那行，解决幻读。

如果事务B是更新id=4 的元组name=cc呢

同理，根据update的规则

id	name	createversion	deleteversion
4	a	2	10
5	b	5	10
4	hh	10	11
5	hh	10	
4	cc	11	

然后根据select的规则去读取的话，得到的还是（4，hh）（5，hh）

可重复读的隔离级别下使用了MVCC机制，select操作不会更新版本号，是快照读（历史版本）；insert、update和delete会更新版本号，是当前读（当前版本）。

“不可重复读”和“幻读”都是读的过程中数据前后不一致，只是前者侧重于修改，后者侧重于增删。

三者的场景介绍完，它们到底有什么区别：

脏读:指读到了其他事务未提交的数据.

不可重复读：读到了其他事务已提交的数据(update).

不可重复读与幻读都是读到其他事务已提交的数据,但是它们针对点不同.

不可重复读:update.

幻读:delete,insert.

隔离级别	脏读 (Dirty Read)	不可重复读 (NonRepeatable Read)	幻读 (Phantom Read)
未提交读 (Read uncommitted)	可能	可能	可能
已提交读 (Read committed)	不可能	可能	可能
可重复读 (Repeatable read)	不可能	不可能	可能
可串行化 (Serializable)	不可能	不可能	不可能

互联网项目建议使用读已提交RC

