20 MVCC怎么实现的?

更新时间：2019-09-19 15:23:29

> 宝剑锋从磨砺出，梅花香自苦寒来。
>
> ——佚名

本节跟大家一起聊聊 MVCC。

1 什么是 MVCC?

在说 MVCC 之前，大家先看看下面这个例子。

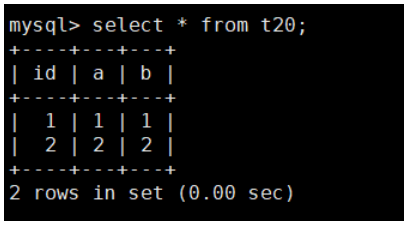
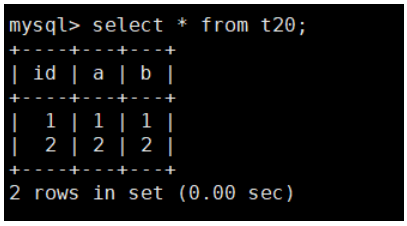
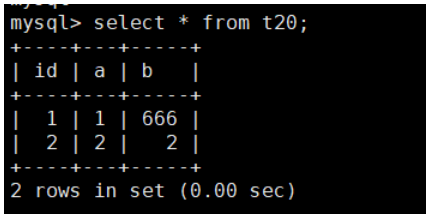
首先创建表并写入测试数据：

```
use muke;
drop table if exists t20;

CREATE TABLE `t20` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`a` int(11) NOT NULL,
`b` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `idx_c` (`a`)
) ENGINE=InnoDB CHARSET=utf8mb4;
insert into t20(a,b) values (1,1),(2,2);
```

进行实验：

	session1	session2
1	set session transaction_isolation='READ-COMMITTED';/* 设置会话隔离级别为 RC*/	set session transaction_isolation='READ-COMMITTED';/* 设置会话隔离级别为 RC*/

	session1	session2
2	select * from t20; 	
3	begin;	
4	update t20 set b=666 where a=1;	
5		begin;
6		select * from t20; 
7	commit;	
8		select * from t20; 
9		commit;

在 session1 更新了 a=1 这行记录，但还没提交的情况下，在 session2 中，满足 a=1 这条记录，b 的值还是原始值 1，而不是 session 1 更新之后的 666，那么在数据库层面，这是怎么实现的呢？

其实 InnoDB 就是通过 MVCC 和 UNDO LOG 来实现的。

什么是 MVCC 呢？

MVCC， 即多版本并发控制。MVCC 的实现，是通过保存数据在某个时间点的快照来实现的，也就是说，不管需要执行多长时间，每个事务看到的数据都是一致的。根据事务开始的时间不同，每个事务对同一张表，同一时刻看到的数据可能是不一样的。

也就是上面实验第 6 步中，为什么 session2 查询的结果还是 session1 修改之前的记录。

MCCC 只在 RC 和 RR 两个隔离级别下工作。因此在上面的实验中，改成 RR 隔离级别，第 6 步中，得到的结果还是 session1 修改之前的记录（但是在第 8 步，结果不一样哦，感兴趣的可以把上面操作放在 RR 隔离级别下实验一下）

上面提到了 undo log，那么什么是 undo log 呢？

2 Undo log

上节我们讲到了 redo log，它记录了事务操作变化。但是事务有时是需要回滚的，这时，undo log 就发挥了作用。undo log 是逻辑日志，将数据库逻辑地恢复到原来的样子，所有修改都被逻辑地取消了。

也就是如果是 insert 操作，其对应的回滚操作就是 delete；

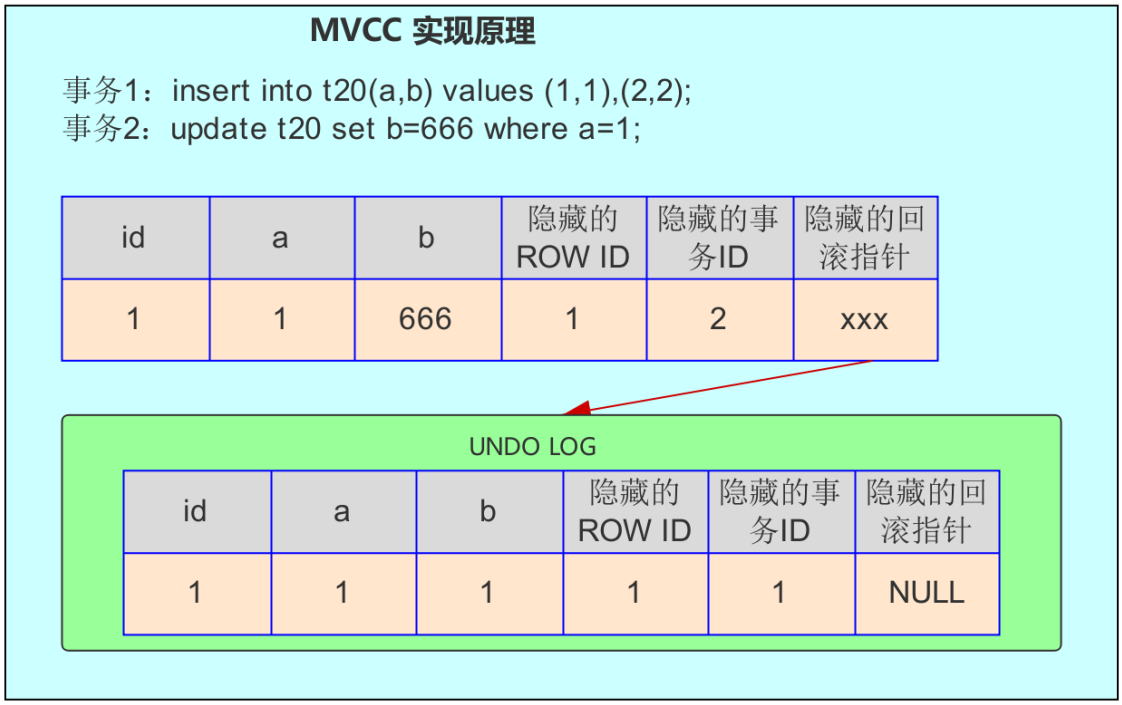
如果是 delete，则对应的回滚操作是 insert；

如果是 update，则对应的回滚操作是一个反向的 update 操作。

除了回滚操作，undo log 的另一个作用是 MVCC，InnoDB 存储引擎中 MVCC 的实现是通过 undo 来完成的。当用户读取一行记录时，若该记录已经被其它事务占用，当前事务可以通过 undo log 读取之前的行版本信息，以此实现非锁定读取。

3 MVCC 的实现原理

InnoDB 每一行数据都有一个隐藏的回滚指针，用于指向该行修改前的最后一个历史版本（存放在 UNDO LOG 中）。



如图，执行 update t20 set b=666 where a=1; 时，会将原记录放到 undo 表空间中，并通过隐藏的回滚指针指向该记录。在本节开始的实验中的第 6 步中，session2 查询的结果是 session1 修改之前的记录，这个记录就是来自 undo log 中。

4 MVCC 的优势

MVCC 最大的好处是读不加锁，读写不冲突，极大地增加了 MySQL 的并发性。

通过 MVCC，保证了事务 ACID 中的 I（隔离性）特性。

5 总结

在 RC 和 RR 隔离级别下，如果需要查询一些被其它事务正在更新的行，看到的是这些记录被更新之前的值。而这就是用 MVCC 实现的。

MVCC 实现的原理大致是：

InnoDB　　每一行数据都有一个隐藏的回滚指针，用于指向该行修改前的最后一个历史版本，这个历史版本存放在 undo log 中。如果要执行更新操作，会将原记录放入 undo log 中，并通过隐藏的回滚指针指向 undo log 中的原记录。其它事务此时需要查询时，就是查询 undo log 中这行数据的最后一个历史版本。

MVCC 最大的好处是读不加锁，读写不冲突，极大的增加了 MySQL 的并发性，通过 MVCC，也保证了事务 ACID 中的 I（隔离性）特性。

6 问题

MCCC 为什么只在 RC 和 RR 两个隔离级别下工作？

7 参考

《MySQL 技术内幕》第 2 版：7.2.2 undo

《高性能 MySQL》第 3 版：1.4 多版本并发控制

}

19 数据库忽然断电会丢数据吗？

21 不同事务隔离级别有哪些区别？