链滴

mysql 幻读问题

作者：AshShawn

原文链接：https://ld246.com/article/1648820289097

来源网站：链滴

1.什么是幻读

幻读指的是一个事务在前后两次查询同一个范围的时候，后一次查询看到了前一次查询没有看到的行,里的读取指的是**当前读**

2.幻读有什么问题

	session A	session B	session C
T1	begin; select * from t where d=5 for update; /*Q1*/ update t set d=100 where d=5;		
T2		update t set d=5 where id=0; update t set c=5 where id=0;	
T3	select * from t where d=5 for update; /*Q2*/		
T4			insert into t values(1,1,5); update t set c=5 where id=1;
T5	select * from t where d=5 for update; /*Q3*/		
T6	commit;		

2.1 破坏语义

Q1要锁住所有d=5的行,但是session B与session C破坏了这个语义

2.2 一致性问题

binlog日志实在提交时按顺序写入的

```
//1.session B先写入
update t set d=5 where id=0; /*(0,0,5)*/
update t set c=5 where id=0; /*(0,5,5)*/
//2. session C再写入
insert into t values(1,1,5); /*(1,1,5)*/
update t set c=5 where id=1; /*(1,5,5)*/
//3. session A最后写入
update t set d=100 where d=5;/*所有d=5的行，d改成100*/
```

如果binlog 同步到从库,就会产生主从不一致的问题

3.解决方案 间隙锁

innoDB为解决幻读引入了间隙锁, 间隙锁+行锁共同解决了RR级别下的幻读问题

间隙锁的出现是为了解决幻读,间隙锁只有再可重复读下才能使用

3.1加锁原则

1. 加锁基本单位为next-key lock(左开右闭);
2. 查找过程中访问的对象才会加锁(二级索引的间隙锁有可能会传递到主键上)
3. 唯一索引等值查询,next-key lock退化为行锁
4. 索引等值查询(包含唯一和普通索引),向右遍历时最后一个不满足等值条件的时候,next-key lock退为间隙锁(左开右开);
5. 范围查询会访问到不满足条件的第一个值为止.

简述:

- 等值查询:

若是唯一索引,退化成行锁;

若是普通索引,需要访问到第一个不满足条件的值,退化成间隙锁;

- 范围查询:

均需访问到不满足条件的第一个值为止

```
CREATE TABLE `t` (
  `id` int(11) NOT NULL,
  `c` int(11) DEFAULT NULL,
  `d` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `c` (`c`)
) ENGINE=InnoDB;

insert into t values(0,0,0),(5,5,5),
(10,10,10),(15,15,15),(20,20,20),(25,25,25);
```

3.2案例一

session A	session B	session C
begin; update t set d=d+1 where id=7;		
	insert into t values(8,8,8); (blocked)	
		update t set d=d+1 where id=10; (Query OK)

id=7不存在,扫描到10,加锁(5,10]

唯一索引等值查询,根据原则4,退化为(5,10)

3.3案例二

session A	session B	session C
begin; select id from t where c=5 lock in share mode;		
	update t set d=d+1 where id=5; (Query OK)	
		insert into t values(7,7,7); (blocked)

根据原则1,加锁范围为(0,5],(5,10]

根据原则4,加锁范围为(0,5],(5,10)

由于使用覆盖索引,id=5并不会加锁

3.4案例三

session A	session B	session C
begin; select * from t where id>=10 and id<11 for update;		
	insert into t values(8,8,8); (Query OK) insert into t values(13,13,13); (blocked)	
		update t set d=d+1 where id=15; (blocked)

id=10,唯一索引等值查询,加行锁10,

id>10,id<11,唯一索引范围查询,加锁(10,15],

综上,加锁范围为[10,15]

3.5案例四

session A	session B	session C
begin; select * from t where c>=10 and c<11 for update;		
	insert into t values(8,8,8); (blocked)	
		update t set d=d+1 where c=15; (blocked)

c>=10,=10为普通索引等值查询,加锁范围为(5,10],

c<11,为普通索引范围查询,加锁范围为(10,15]

综上,加锁范围为(5,15]

3.6案例五

session A	session B	session C
begin; select * from t where id>10 and id<=15 for update;		
	update t set d=d+1 where id=20; (blocked)	
		insert into t values(16,16,16); (blocked)

id>10,唯一索引范围查询,加锁范围(10,15]

id<15,唯一索引范围查询,加锁(10,15],

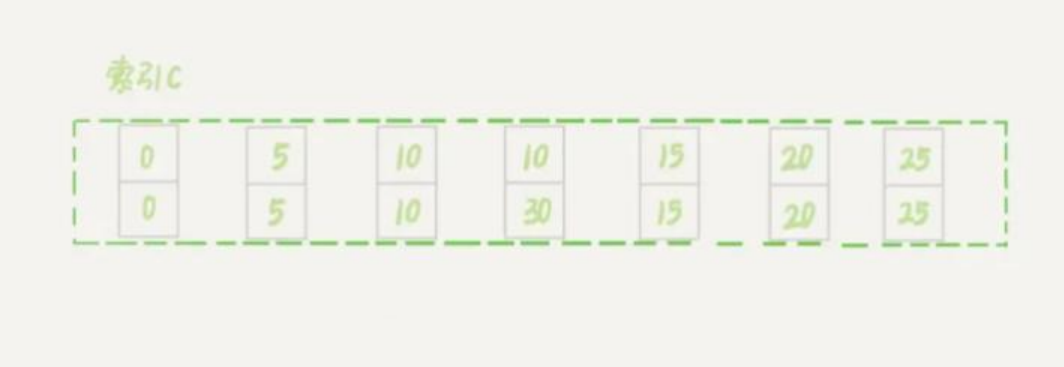
id=15,唯一索引等值查询,加行锁15,

根据原则5,唯一索引再扫描到不满足条件的第一个值,加锁(15,20],

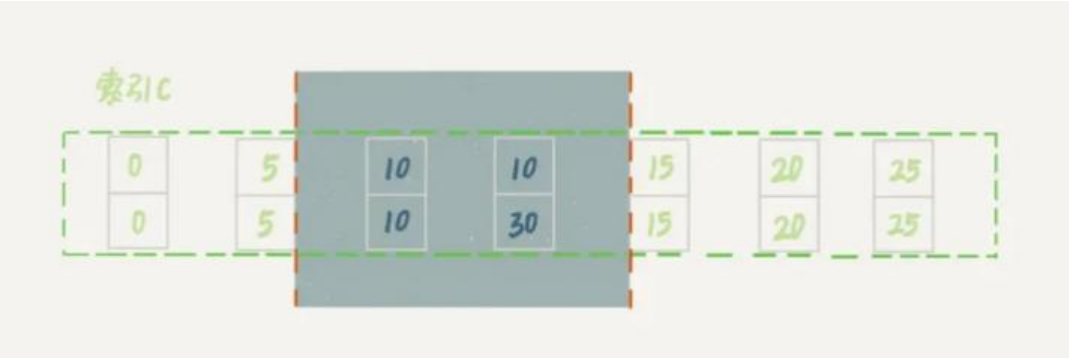
综上,加锁范围为(10,20]

3.7案例六

```
mysql> insert into t values(30,10,30);
```



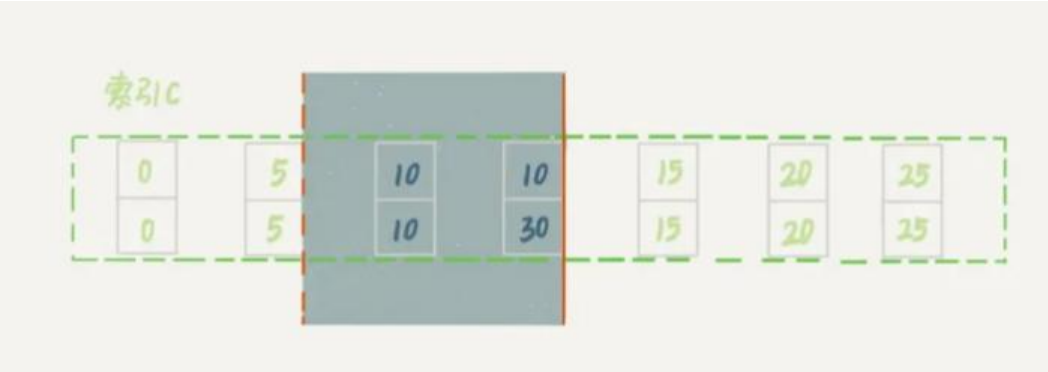
普通索引等值查询,加锁(5,10],
普通索引等值下扫到不满足条件的索引,加锁(10,15),
综上,加锁范围为(5,15)



3.8案例七

session A	session B
begin; delete from t where c=10 limit 2;	
	insert into t values(12,12,12); (Query OK)

受limit影响,加锁范围为



原文链接：mysql 幻读问题

所以,**删除数据时尽量加上limit,以减少加锁范围**

3.9案例八

session A	session B
begin; select id from t where c=10 lock in share mode;	
	update t set d=d+1 where c=10; (blocked)
insert into t values(8,8,8);	
	ERROR 1213 (40001): Deadlock found when trying to get lock; try restarting transaction

session A 加锁(5,15)

session B 等待加锁 (5,10)(分步骤加锁),阻塞等待session A的锁

session A insert语句等待加锁(5,10)(分步骤加锁),阻塞等待session B的锁,形成死锁

间隙锁可以共享,但是间隙锁已insert互斥, sessionA和sessionB其实是行锁10互斥

3.10案例九

session A	session B
begin; select * from t where c>=15 and c<=20 order by c desc lock in share mode;	
	insert into t values(6,6,6); (blocked)

c=20,普通索引等值查询,下扫到25,加锁(15,25)

c=15,普通索引等值查询,加锁(10,15]

由于索引c向左遍历,扫到10,加锁(5,10]

综上 加锁范围为c索引(5,25),主键id上是15和20两个行锁