



链滴

mysql 锁随记

作者: [yechuan](#)

原文链接: <https://ld246.com/article/1678464524627>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



mysql锁 (innodb)

测试数据库

```
create database if not exists `mysql_demo`;  
use `mysql_demo`;  
create table if not exists `user`  
(  
  `id` int primary key,  
  `age` int,  
  `name` varchar(20),  
  index index_age(age)  
) charset utf8mb4 engine innodb;  
insert into user (id, age, name)  
values (1, 18, 'name_1')  
  , (2, 18, 'name_2')  
  , (3, 18, 'name_3')  
  , (4, 18, 'name_4')  
  , (5, 20, 'name_5')  
  , (7, 19, 'name_7')  
  , (8, 19, 'name_8');
```

锁的级别

- 排他锁 (X)

对于所有的写操作默认均会加上排他锁，获取到锁后会禁止其他事务获取此资源的任意锁
也可使用 **for update** 显式上锁

```

# 事务A对id=1的上排他锁
select * from user where id = 1 for update;
----- 事务B -----
# 尝试加共享锁 阻塞
select * from user where id = 1 lock in share mode;
# 尝试加排他锁 阻塞
select * from user where id = 1 for update;
update user set name = 'name_1_1' where id = 1;
# 尝试加排他锁 阻塞 此时上的是表锁 包含有 id=1的数据
update user set name = 'name_2_2' where name = 'name_2';

```

- 共享锁 (S)

读锁，会阻止其他事务获取此资源**排他锁**，允许获取**共享锁**

可以使用**lock in share mode**进行显式上锁

select语句不会默认上锁 为快照读

```

# 事务A对id=1的上共享锁
select * from user where id = 1 lock in share mode;
----- 事务B -----
# 未获取共享锁 快照读
select * from user where id = 1;
# 尝试获取共享锁 成功
select * from user where id = 1 lock in share mode;
# 尝试获取排他锁 阻塞
select * from user where id = 1 for update;

```

- 意向锁

当事务对某行数据使用了行锁，那么在这之前数据库会自动给此事务申请一个表的意向锁（意向共享锁S、意向排他锁IX）

当其他事务进行申请行级别锁时，则会根据类型被阻塞

意向锁是一种表级别的锁

- 几种锁的互斥性

级别)	意向排他锁IX 共享锁S (行级别)	意向共享锁IS	排他锁X (
意向排他锁IX 斥	兼容	兼容	互斥
意向共享锁IS 容	兼容	兼容	互斥
排他锁X (行级别) 斥	互斥	互斥	互斥
共享锁S (行级别) 容	互斥	兼容	互斥

锁的粒度

只有命中索引才可能执行行锁，否则是表锁

- 行锁

锁加在索引上，并非数据行

```
# 索引命中 但数据不存在 此时为间隙锁 锁住(7,+∞)
select * from user where id = 8 for update;
```

```
# 命中索引且数据存在 此时锁住 1
select * from user where id = 1 for update;
```

```
# 命中索引且数据存在 此时锁住 (-∞,2)
select * from user where id < 2 for update;
```

```
# 命中索引且数据存在 此时锁住 age (-∞,18]、 (18,19) ;id(-∞,4]、 (4,7)
select * from user where age = 18 for update;
```

- 表锁

```
# 索引失效 即索引未命中 表锁
select * from user where id + 1 = 2 for update;
```

```
# 索引未命中 表锁
select * from user where name = 'name_1' for update;
```

上锁过程（行锁）

- 命中聚簇索引

对命中的聚簇索引行加锁

- 命中非聚簇

对命中的非聚簇索引行加锁

对对应的聚簇索引行加锁

锁的区间

- 普通行锁（record lock）

锁住的是一行数据

- 间隙锁（gap lock）

锁住一个范围区间（开区间）

- 临键锁（next-key lock）

间隙锁 + 普通行锁（左开右闭）

- sql分析

```
# 事务A
select * from user where age = 18 for update;
----- 事务B -----
# sql_1 阻塞
select * from user where id = 4 for update;
# sql_2 阻塞
insert into user (id,age,name) values (6,19,'name_8')
# sql_3 阻塞
update user set name='name_5_5' where id =5;
# sql_4 阻塞
select * from user where age = 18 for update;
```

命中的索引是非聚簇非唯一索引

对index_age索引加临键锁 $(-\infty, 18]$ 、间隙锁(18,19)

sql_4阻塞

对聚簇索引加临键锁 $(-\infty, 4]$ 、间隙锁(4,7)

sql_1、sql_2、sql_3阻塞