

MySQL for update 详解

📅 发表于 2020-05-14 | 👁 阅读次数: 23930

for update 的作用是在查询的时候为行加上 排它锁， 当一个事务的操作未完成时候， 其他事务可以读取但是不能写入或更新。

它的典型使用场景是高并发并且对于数据的准确性有很高要求， 比如金钱、库存等， 一般这种操作都是很长一串并且开启事务的， 假如现在要对库存进行操作， 在刚开始读的时候是1， 然后马上另外一个进程将库存更新为0了， 但事务还没结束， 会一直用1进行后续的逻辑， 就会有问题， 所以需要 for upate 加锁防止出错， 记住一个原则： 一锁二判三更新。

InnoDB行锁类型

共享锁(S-Lock)

允许多个事务对于同一数据可以共享一把锁， 都能访问到数据， 阻止其它事务对于同一数据获取 排它锁。

排它锁(X-Lock)

允许事务删除或者更新一行数据， 阻止其它事务对于同一数据获取其它锁， 包括 共享锁 和 排它锁。

select 语句默认不获取任何锁， 所以是可以读被其它事务持有 排它锁 的数据的！

for update 使用方式

for update 仅适用于 InnoDB， 并且必须开启事务， 在 begin 与 commit 之间才生效。

```
1 select * from table_name where ... for update;
```

行级锁？ 表级锁？

InnoDB 默认是行级锁， 当有明确指定的 主键/索引 时候， 是 行级锁， 否则是 表级锁。
假设表 user， 存在有 id 跟 name 字段， id 是主键， 有5条数据。

行级锁

明确指定主键， 并且有此记录， 行级锁

```
1 SELECT * FROM user WHERE id = 1 FOR UPDATE;
2 SELECT * FROM user WHERE id = 1 and name = 'segon' FOR UPDATE;
```

验证

进程 1	进程 2
begin;	
SELECT * FROM user WHERE id = 1 FOR UPDATE;	
	UPDATE user SET name = 'test' WHERE id = 2; - 成功
	UPDATE user SET name = 'test' WHERE id = 1; - 等待
commit;	
	执行等待的任务， 成功

表级锁

无主键/索引， 表级锁

```
1 SELECT * FROM user WHERE name = 'segon' FOR UPDATE;
```

主键/索引不明确， 表级锁

```
1 SELECT * FROM user WHERE id <> 3 FOR UPDATE;
2 SELECT * FROM user WHERE id LIKE '3' FOR UPDATE;
```

验证

进程 1	进程 2
begin;	
SELECT * FROM user WHERE id LIKE '3' FOR UPDATE;	
	UPDATE user SET name = 'test' WHERE id = 1; - 等待
commit;	
	执行等待的任务， 成功

无锁

明确指定主键/索引， 若查无此记录， 无锁

```
1 SELECT * FROM user WHERE id = -1 FOR UPDATE;
```

验证

进程 1	进程 2
begin;	
SELECT * FROM user WHERE id = -1 FOR UPDATE;	
	UPDATE user SET name = 'test' WHERE id = 2; - 成功
commit;	

相关文章

- [Docker安装MySql](#)
- [MySQL 统计每秒查询次数、插入次数、删除次数、更新次数](#)
- [MySQL查看库大小、表大小、索引大小](#)
- [使用 Jenkins 构建、部署 Spring Boot 应用](#)
- [Android 中 Handler 导致的内存泄漏](#)

本文作者：Segon

本文链接：https://segon.cn/mysql-for-update.html

版权声明： 本博客所有文章除特别声明外，均采用 ©BY-NC-SA 许可协议。 转载请注明出处！

MySQL