悲观锁、乐观锁、行级锁、表级锁 - kuyuyingzi的专栏 - 博客频道 - CSDN

更新丢失:新改的覆盖先改的,开发中有三种方法解决

1、将事务级别提高到最高级别TRANSACTION SERIALIZABLE

操作双方都要提升级别;查询使用共享锁;更新使用更新锁;一方查询,一方更新,共享锁和更新锁会冲突;当 双方更新锁,出现死锁,程序会自动回滚一方操作,从而避免更新丢失

2、悲观锁(加的是表级锁)

一方:查询语句加 for update;另一方:查询语句加 for update;当进行更新语句的时候,另一方不能进行更新操作

```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)
                                                                                                    id | name | money |
                                                                                                     1 | aaa | 800 |
   nysql> select * from account where name='aaa' for update;
                                                                                                    row in set (0.00 sec)
  id | name | money |
                                                                                                    ysql> start transaction;
uery OK, O rows affected (0.00 sec)
  | 1 | aaa | 800 |
  1 row in set (0.00 sec)
                                                                                                    ysql> select * from account where name='aaa' for update;
   mysgl> update account set money=600 where name='aaa';
                                                                                                    id | name | money |
  Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
                                                                                                     1 | aaa | 600 |
  mysql> commit;
Query OK, 0 rows affected (0.00 sec)
                                                                                                    row in set (13.01 sec)
nonev=1200
```

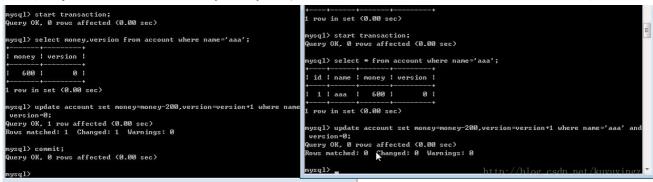
3、乐观锁

更新语句设置版本号,在指定版本中更新数据

一方: update account set money=money-200, version=version+1 where id=1 and version=0;

另一方操作同一个版本号,则不能更新数据

另一方: update account set money=money+200, version=version+1 where id=1 and version=0;



如果更新多,查询少,用悲观锁;反之,乐观锁

表级锁, where用的是非主键

行级锁,where用主键一般是id

如果用表级锁,其他客户将不能进行查询操作,因此开发中记得用行级锁