[mysql创建触发器](https://www.cnblogs.com/zzlback/p/12660560.html)

首先，我们来了解一下什么是触发器，触发器，就是在对一张表数据进行增（insert），删（delete），改（update）的时候，为了保持数据的一致性，对别的表也要进行相应的数据修改。

我们都知道mysql最后事务提交后，数据是会保存到磁盘上的，那么每次在insert，delete，update时候旧数据和新数据，会在内存中生成临时的行数据，分别叫old和new。例如要inset插入一条数据的时候，会先将这行数据放在内存中，叫new临时表。update的时候，会先将更新之前的数据放在内存old表中，即将更新的数据放在内存new表中。然后通过内存，把数据持久化保存到磁盘中，触发器的before和after就是相对于数据是从内存持久化到数据库之前还是之后触发相对而言的。

举个例子吧：

假设有商品goods表：商品id，商品名字，商品数量

good\_id  good\_name  num

    1            books         10

    2            phones       20

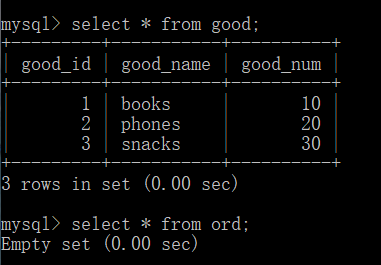
    3            snacks        30

有订单ord表：订单id，商品id，订单数量。

假设购买了10部手机，订单表中插入10部手机，那么商品中对应手机的数量是不是要减少10部。这完全可以通过编写高级程序代码来实现，但是高级程序要多次与数据库交互，浪费时间。

这也可以通过触发器来实现，就减少了程序与数据库的交互，节省时间。

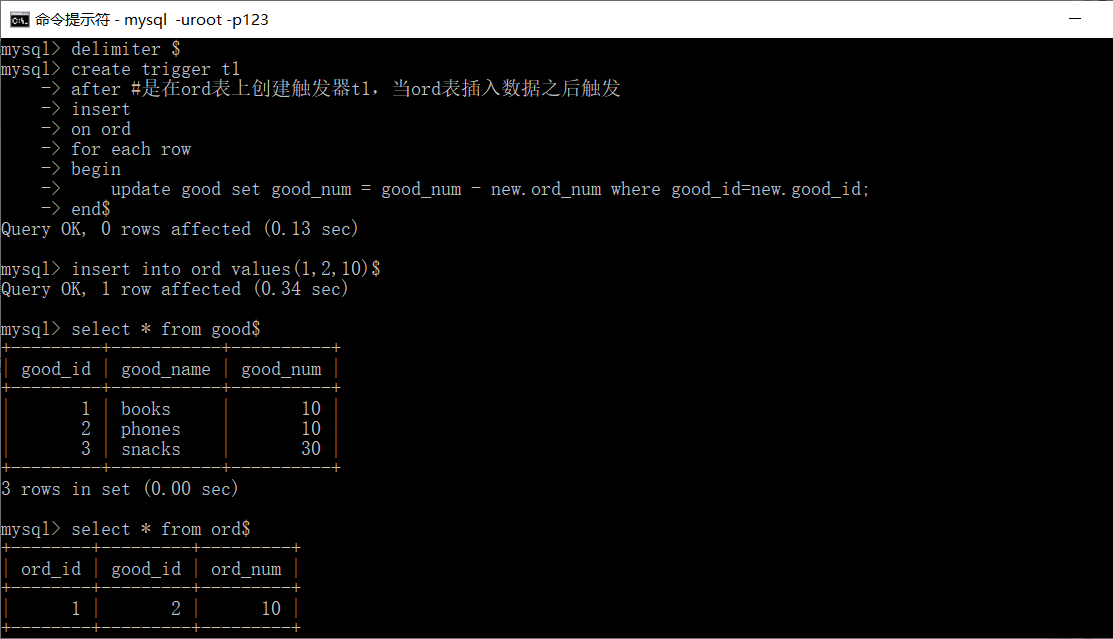
翻译一下上面的业务，就是在ord表中插入一条订单时，要在商品表中，对应的商品的对应num数量要减少，怎么写这个触发器呢？



1.创建插入数据时候的触发器

先修改mysql默认的结束符号位$

delimiter $  //结尾不要带分号



代码如下：

[复制代码](javascript:void(0);)

create trigger t1

after #是在ord表上创建触发器t1，当ord表插入数据之后触发

insert #在创建触发器t1，在内存数据持久化到磁盘，insert ord 表之后操作

on ord

for each row #固定写法，为的是批量操作

begin

update good set good\_num = good\_num - new.ord\_num where good\_id=new.good\_id;

end$

[复制代码](javascript:void(0);)

当我向订单表插入数据时，商品表数据自动更新了。这就是触发器的作用。其中for each row是固定代码，就是如果做批量更新，每行都是这样操作。

其中new.ord\_num 和new.good\_id 是表示在内存中的临时表，即将要插入的ord表中的那一行数据。本质是就是

ord\_id  good\_id   ord\_num

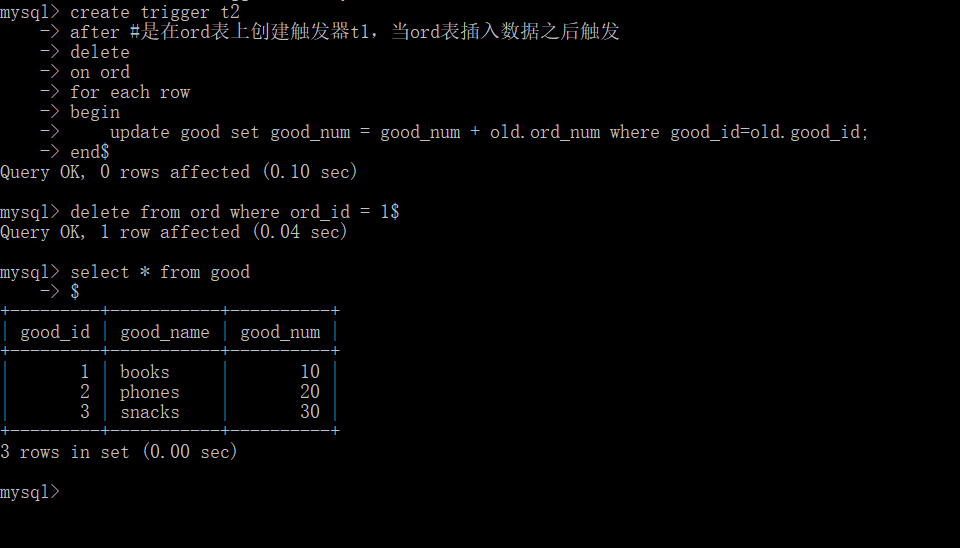
   1           2          10

 这行数据，只不过是暂时在内存new表中中存放。

2.创建删除时候的触发器

如何这时候订单1，撤销了，删除了，那么对应的2号商品Phone是不是要增加10，回到20。

这就是删除触发器，



 代码如下：

[复制代码](javascript:void(0);)

create trigger t2

after #是在ord表上创建触发器t1，当ord表插入数据之后触发

delete

on ord

for each row

begin

update good set good\_num = good\_num + old.ord\_num where good\_id=old.good\_id;

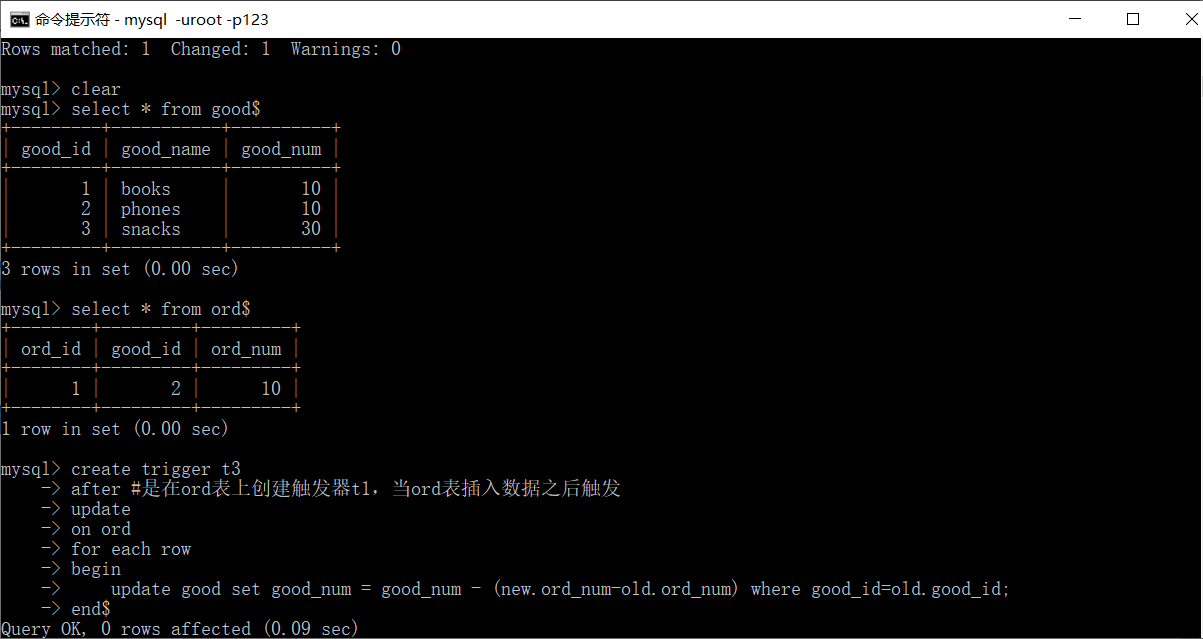
end$

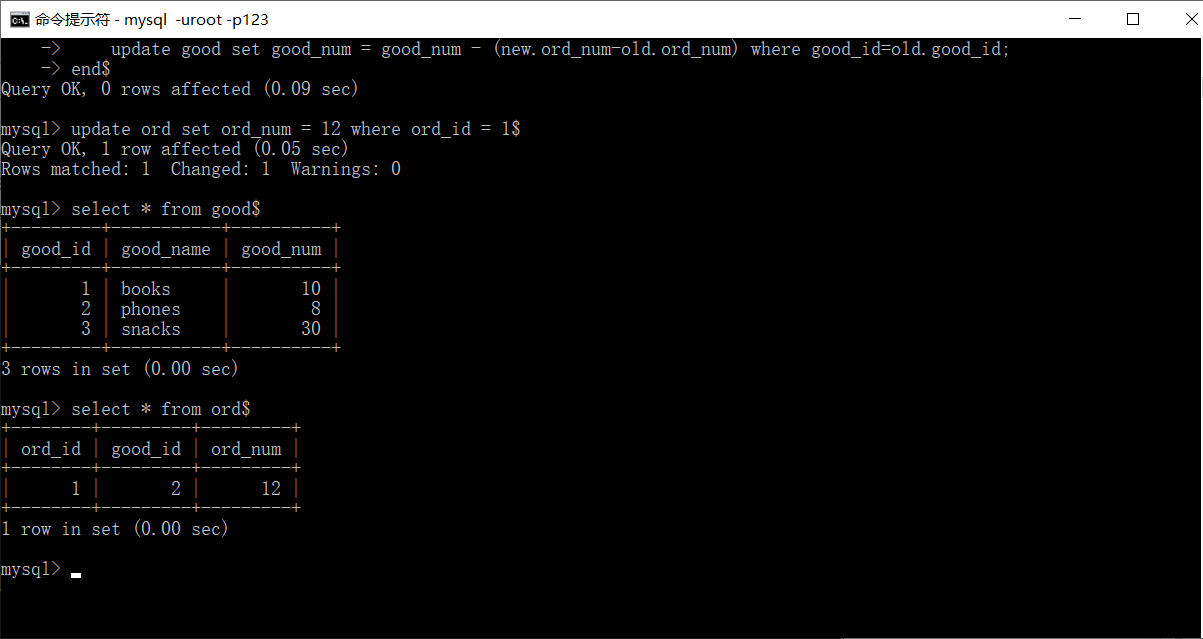
[复制代码](javascript:void(0);)

3.创建更新时候的触发器

数据库的更新，其实是分两步走，第一步是先将旧数据删除，保存在old临时表，再将新数据插入，保存在new临时表中。

那么我更新订单的数量，本来是买10部手机，那我想买12部手机了，这时候商品good表应该再减2部手机，这两部是new.12 - old.10，新表保存了12，旧表保存了10





代码如下:

[复制代码](javascript:void(0);)

create trigger t3

after #是在ord表上创建触发器t1，当ord表插入数据之后触发

update

on ord

for each row

begin

update good set good\_num = good\_num - (new.ord\_num-old.ord\_num) where good\_id=old.good\_id;

end$

[复制代码](javascript:void(0);)

这里where good\_id = old.good\_id 也可以写成 good\_id = new.good\_id 应为这两张表中保存的都是这种这个商品的id。

到此为止，我就把这三个操作的触发器都创建了一遍。

现在还没有说 在创建触发器时，before和after有什么区别？

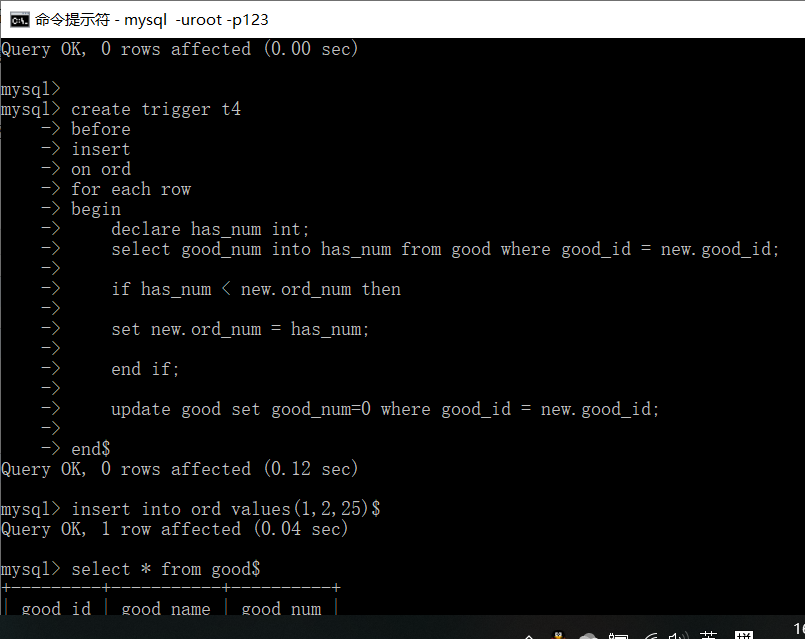
其实要理解这个区别，一定要牢记的就是，mysql在插入操作的时候，是先将数据保存在内存new表中，再将数据写入磁盘，删除操作时，是先将要删除的数据保存在内存的old表中，再写入磁盘完成删除。

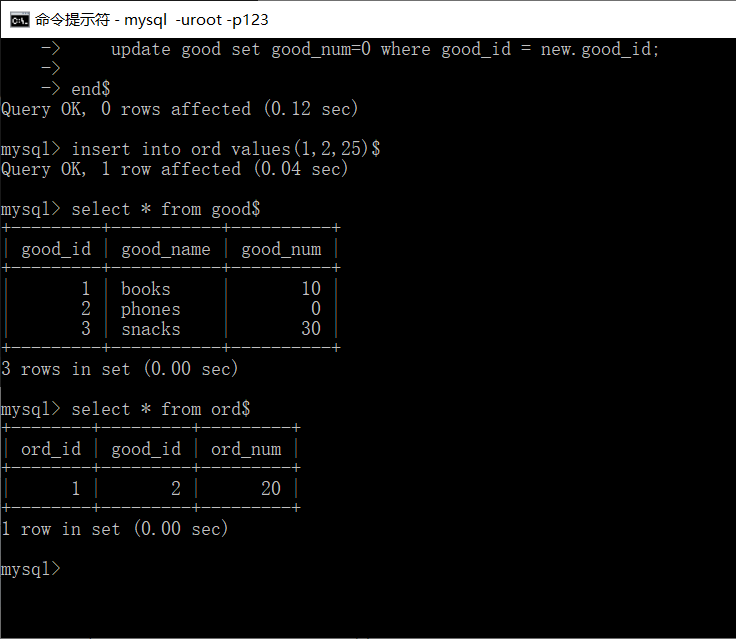
这个before和after是相对这个临时表 new或old相对而言的。定义after就是指在临时表从内存更新到磁盘之后才触发，定义before就是指，在临时表更新到磁盘之前触发。

如下面的业务场景，如果我用户购买商品，想购买25部手机，但是库存中只有10部手机，我想通过数据库来判断如果想购买的手机超过了库存量，就只让用户购买库存量这么多手机，不能超额购买。

针对这个问题怎么实现呢？

是不是当用户购买25部手机时，先去查询good表，手机还剩多少部手机，如果库存手机数大于 25部，就让购买。如果不够时，将要购买的手机数量个更新为当前的库存最大数量。





 初始phone是有20部，但是要买25部，只能让购买20部，代码如下：

[复制代码](javascript:void(0);)

create trigger t4

before

insert

on ord

for each row

begin

declare has\_num int;

select good\_num into has\_num from good where good\_id = new.good\_id;

if has\_num < new.ord\_num then

set new.ord\_num = has\_num;

end if;

update good set good\_num =0 where good\_id = new.good\_id;

end$

[复制代码](javascript:void(0);)

我这个地方调了很久，一直是下完订单之后超过库存数量后，good表更新的值有问题，不是为0。

原来是这个触发器是before发生，与之前的after触发器冲突了。两个触发器都起了作用，更新了两遍good表。