



实 验      伍建霖      学      20337251      日 期: 2022.11.30  
人:                          号:

联系方 式: QQ773542531

### 一. 实验目的

## 二. 实验内容和要求

### 三. 实验重点和难点

难点：利用触发器实现较为复杂的用户自定义完整性。

#### 四. 实验工具

MySQL、SQL Server、Navicat

## 五. 实验过程

- 在 MySQL 环境下，执行以下操作

- 参阅 1 (触发器):

1 / 13

- 参阅 2(异常处理):

- ◆ <http://www.yiidian.com/mysql/mysql-signal-resignal.html>

- ◆ <https://ziruchu.com/art/198>

## (1) AFTER 触发器

①在 Lineitem 表上定义一个 UPDATE 触发器, 当修改订单明细 (即修改订单明细价格 extendedprice、折扣 discount、税率 tax) 时, 自动修改订单 Orders 的 TotalPrice, 以保持数据一致性。

```
/* totalprice = totalprice + extendedprice * (1-discount) * (1+tax) ) */
CREATE TRIGGER TRI_Lineitem_Price_UPDATE
AFTER UPDATE ON Lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff REAL;
    /*订单明细修改后, 计算订单含税折扣价总价的修正值*/
    set L_valuediff = NEW.extendedprice*(1-NEW.discount) *(1+NEW.tax)
                    - OLD.extendedprice*(1-OLD.discount) *(1+OLD.tax) ;
    /*更新订单的含税折扣价总价*/
    UPDATE Orders
    SET totalprice=totalprice+L_valuediff
    WHERE orderkey=NEW.orderkey;
END;
```

验证截图如下:

show TRIGGERS;

信息	摘要	结果 1	剖析	状态			
Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer
► TRI_Lineitem_Price_UPDATE	UPDATE	lineitem	BEGINDECLARE L_valuediff	AFTER	2022-10-30 21:58:21.77	STRICT_TRANS_TABLES,N	root@localhost

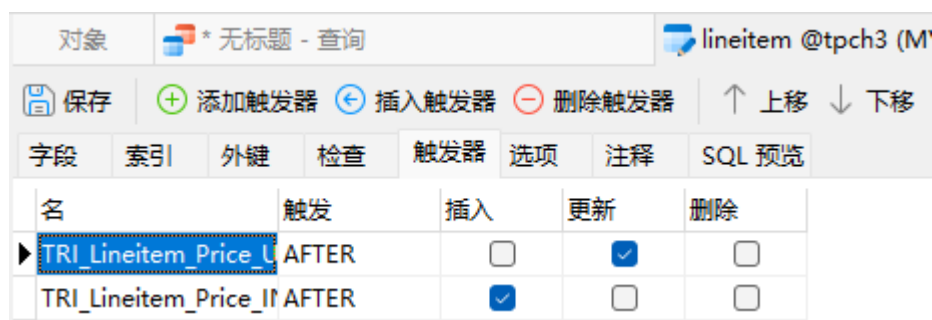
或者

字段	索引	外键	检查	触发器	选项	注释	SQL 预览
名				触发	插入	更新	删除
► TRI_Lineitem_Price_UPDATE				AFTER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

②在 Lineitem 表上定义一个 INSERT 触发器，当增加一项订单明细时，自动修改订单 Orders 的 TotalPrice，以保持数据一致性。

```
CREATE TRIGGER TRI_Lineitem_Price_INSERT
AFTER INSERT ON Lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff REAL;
    SET L_valuediff=NEW.extendedprice*(1-NEW.discount) *(1+NEW.tax) ;
    /*增加订单明细项后，计算订单含税折扣价总价的修正值*/
    UPDATE Orders SET TotalPrice=TotalPrice+L_valuediff
    /*更新订单的含税折扣价总价*/
    WHERE orderkey=NEW.orderkey;
END;
```

验证截图如下：



③在 Lineitem 表上定义一个 DELETE 触发器，当删除一项订单明细时，自动修改订单 Orders 的 TotalPrice，以保持数据一致性。

```
CREATE TRIGGER TRI_Lineitem_Price_DELETE
AFTER DELETE ON Lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff REAL;
    Set L_valuediff = -OLD.extendedprice*(1-OLD.discount) *(1+OLD.tax) ;
    /*删除订单明细项后，计算订单含税折扣价总价的修正值*/
    UPDATE Orders SET TotalPrice=TotalPrice+L_valuediff
    /*更新订单的含税折扣价总价*/
    WHERE orderkey=OLD.orderkey;
END;
```

验证截图如下：

对象	* 无标题 - 查询				lineitem @tpch3 (M)
保存	添加触发器	插入触发器	删除触发器	上移 下移	
字段	索引	外键	检查	触发器	SQL 预览
名	触发	插入	更新	删除	
TRI_Lineitem_Price_UPDATE	AFTER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TRI_Lineitem_Price_INSERT	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TRI_Lineitem_Price_DELETE	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

④验证触发器 TRI\_Lineitem\_Price\_UPDATE。

```
/*查看 1854 号订单的含税折扣总价 TotalPrice*/
SELECT TotalPrice
FROM Orders
WHERE orderkey=1854;
```

验证截图如下：

对象	* 无标题 - 查询			
保存	查询创建工具	美化 SQL		
MYSQL	tpch3			
<pre>1  /*查看1854号订单的含税折扣总价 2  SELECT TotalPrice 3  FROM Orders 4  WHERE orderkey=1854; 5   </pre>				
信息	摘要	结果 1	剖析	状态
TotalPrice		10908633.55		

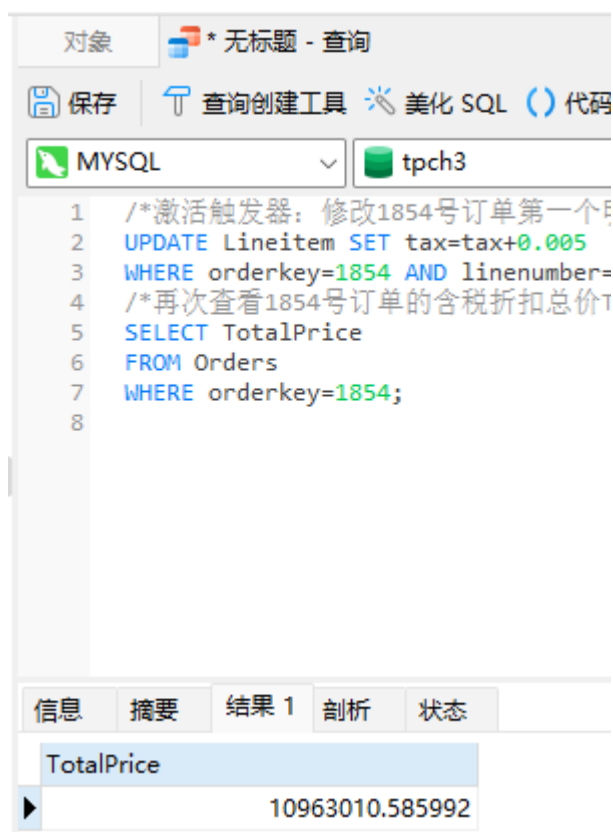
```

/*激活触发器：修改 1854 号订单第一个明细项的税率，该税率增加 0.5%*/
UPDATE Lineitem SET tax=tax+0.005
WHERE orderkey=1854 AND linenum=1;

/*再次查看 1854 号订单的含税折扣总价 TotalPrice 是否有变化， 如有变化， 则是触
发器起作用了， 否则触发器没有起作用*/
SELECT TotalPrice
FROM Orders
WHERE orderkey=1854;

```

验证截图如下：



## (2) BEFORE 触发器

①在 Lineitem 表上定义一个 BEFORE UPDATE 触发器， 当修改订单明细中的数量 (quantity) 时， 先检查供应表 PartSupp 中的可用数量 availqty 是否足够。

```

CREATE TRIGGER TRI_Lineitem_Quantity_UPDATE
BEFORE UPDATE ON Lineitem

```

```

FOR EACH ROW
BEGIN
    DECLARE      L_valuediff INTEGER;
    DECLARE      L_availqty INTEGER;
    /*计算订单明细项修改时，订购数量的变化值*/
    set L_valuediff=NEW.quantity-OLD.quantity;
    /*查询当前订单明细项对应零件供应记录中的可用数量*/
    SELECT availqty INTO L_availqty
    FROM PartSupp
    WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
    IF(L_availqty-L_valuediff>=0) THEN
    BEGIN
        /*如果可用数量可以满足订单订购数量，则提示 ENOUGH*/
        SELECT 'Available quantity is ENOUGH' into @user_prompt;
        /*修改当前订单明细项对应零件供应记录中的可用数量*/
        UPDATE PartSupp
        SET availqty=availqty-L_valuediff
        WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
    END;
    ELSE
        /*如果可用数量不能满足订单订购数量，则更新过程异常中断*/
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Available quantity is NOT ENOUGH';
    END IF;
END;

```

验证截图如下：

对象		* 无标题 - 查询		lineitem @tpch3	
保存		添加触发器	插入触发器	删除触发器	上移 下移
字段	索引	外键	检查	触发器	选项
名	触发	插入	更新	删除	
TRI_Lineitem_Price_U	AFTER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TRI_Lineitem_Price_I	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TRI_Lineitem_Price_D	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
TRI_Lineitem_Quantit	BEFORE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

②在 Lineitem 表上定义一个 BEFORE INSERT 触发器，当插入订单明细，先检查供应表 PartSupp 中的可用数量 availqty 是否足够。

```
CREATE TRIGGER TRI_Lineitem_Quantity_INSERT
BEFORE INSERT ON Lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff, L_availqty INTEGER;
    set L_valuediff = NEW.quantity; /*获得插入订单明细项的订购数量*/
    /*查询当前订单明细项对应零件供应记录中的可用数量*/
    SELECT availqty INTO L_availqty
    FROM PartSupp
    WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
    IF(L_availqty - L_valuediff >= 0) THEN
    BEGIN
        /*如果可用数量可以满足订单订购数量，则提示 ENOUGH*/
        SELECT 'Available quantity is ENOUGH' into @user_prompt;
        /*修改当前订单明细项对应零件供应记录中的可用数量*/
        UPDATE PartSupp
        SET availqty = availqty-L_valuediff
        WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
    END;
    ELSE
        /*如果可用数量不能满足订单订购数量，则插入过程异常中断。*/
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Available quantity is NOT ENOUGH';
    END IF;
END;
```

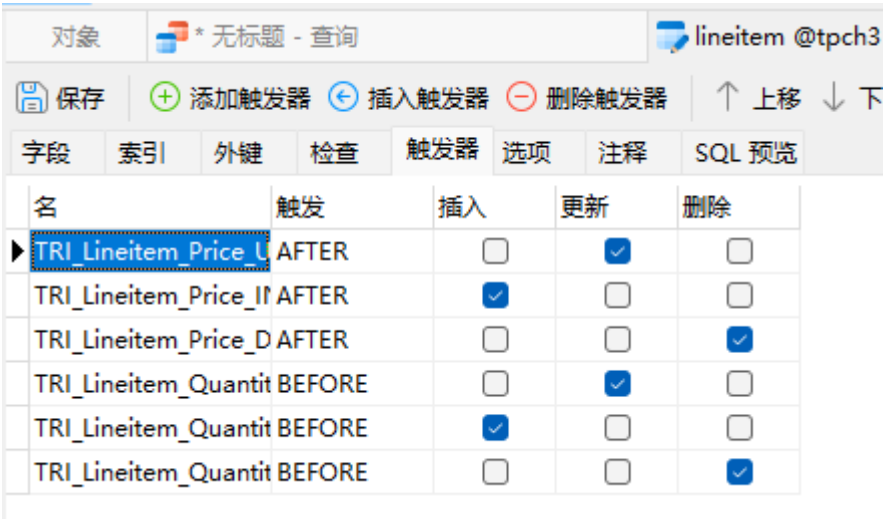
验证截图如下：

对象		* 无标题 - 查询			lineitem @tpch										
保存		添加触发器		插入触发器		删除触发器		上移		下移					
字段		索引		外键		检查		触发器		选项		注释		SQL 预览	
名		触发		插入		更新		删除							
▶ TRI_Lineitem_Price_U		AFTER		<input type="checkbox"/>		<input checked="" type="checkbox"/>		<input type="checkbox"/>							
TRI_Lineitem_Price_I		AFTER		<input checked="" type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>							
TRI_Lineitem_Price_D		AFTER		<input type="checkbox"/>		<input type="checkbox"/>		<input checked="" type="checkbox"/>							
TRI_Lineitem_Quantit		BEFORE		<input type="checkbox"/>		<input checked="" type="checkbox"/>		<input type="checkbox"/>							
TRI_Lineitem_Quantit		BEFORE		<input checked="" type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>							

③在 Lineitem 表上定义一个 BEFORE DELETE 触发器， 当删除订单明细时， 该订单明细项订购的数量要归还对应的零件供应记录。

```
CREATE TRIGGER TRI_Lineitem_Quantity_DELETE
  BEFORE DELETE ON Lineitem
  FOR EACH ROW
  BEGIN
    DECLARE L_valuediff, L_availqty INTEGER;
    /*获得删除订单明细项的订购数量*/
    set L_valuediff = OLD.quantity;
    /*修改当前订单明细项对应零件供应记录中的可用数量*/
    UPDATE PartSupp
    SET availqty = availqty+L_valuediff
    WHERE partkey=OLD.partkey AND suppkey=OLD.suppkey;
  END;
```

验证截图如下：



名	触发	插入	更新	删除
TRI_Lineitem_Price_U	AFTER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TRI_Lineitem_Price_I	AFTER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TRI_Lineitem_Price_D	AFTER	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
TRI_Lineitem_Quantity	BEFORE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TRI_Lineitem_Quantity	BEFORE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TRI_Lineitem_Quantity	BEFORE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

④验证触发器 TRI\_Lineitem\_Quantity\_UPDATE。

■ 验证 1

```
/*查看 1854 号订单第 1 个明细项的零件和供应商编号、订购数量、可用数量*/
SELECT L.partkey, L.suppkey, L.quantity, PS.availqty
FROM Lineitem L, PartSupp PS
WHERE L.partkey=PS.partkey AND L.suppkey=PS.suppkey
      AND L.orderkey=1854 AND L.linenum=1;
```

验证截图如下：



对象

\* 无标题 - 查询

保存

查询创建工具

美化 SQL

代码段

文本

导出结果

创建图表

MYSQL

tpch3

运行

停止

解释

```
1  /*查看1854号订单第1个明细项的零件和供应商编号、订购数量、可用数量*/
2  SELECT L.partkey, L.supkey, L.quantity, PS.availqty
3  FROM Lineitem L, PartSupp PS
4  WHERE L.partkey=PS.partkey AND L.supkey=PS.supkey
5         AND L.orderkey=1854 AND L.linenum=1;
6
```

信息

摘要

结果 1

剖析

状态

partkey	supkey	quantity	availqty
19966	14762	73	516

```
/*激活触发器：修改 1854 号订单第 1 个明细项的订购数量*/
UPDATE Lineitem
SET quantity =quantity+ 5
WHERE orderkey= 1854 AND linenum=1;
/*再次查看 1854 号订单第 1 个明细项的相关信息,以验证触发器是否起作用*/
SELECT L.partkey, L.supkey, L.quantity, PS.availqty
FROM Lineitem L, PartSupp PS
WHERE L.partkey=PS.partkey AND L.supkey=PS.supkey
      AND L.orderkey=1854 AND L.linenum=1;
```

验证截图如下：

对象

\* 无标题 - 查询

保存

查询创建工具

美化 SQL

代码段

文本

导出结果

创建图表

MYSQL

tpch3

运行

停止

解释

```
1  /*激活触发器：修改1854号订单第1个明细项的订购数量*/
2  UPDATE Lineitem
3  SET quantity =quantity+ 5
4  WHERE orderkey= 1854 AND linenum=1;
5  /*再次查看1854号订单第1个明细项的相关信息,以验证触发器是否起作用*/
6  SELECT L.partkey, L.supkey, L.quantity, PS.availqty
7  FROM Lineitem L, PartSupp PS
8  WHERE L.partkey=PS.partkey AND L.supkey=PS.supkey
9         AND L.orderkey=1854 AND L.linenum=1;
10
```

信息

摘要

结果 1

剖析

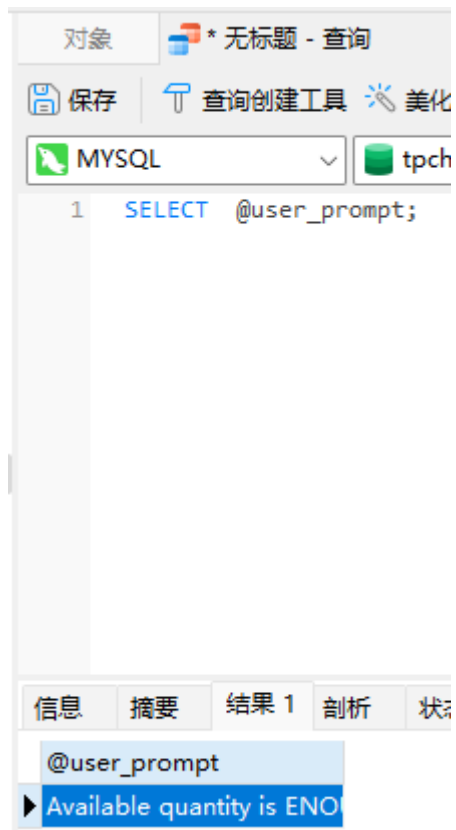
状态

partkey	supkey	quantity	availqty
19966	14762	78	511

■ 验证 2: 可用数量足够的提示?

```
SELECT @user_prompt;
```

验证截图如下:



■ 验证 3: 可用数量不够的提示?

```
UPDATE Lineitem  
SET quantity =quantity+ 50000  
WHERE orderkey= 1854 AND linenum=1;
```

验证截图如下:



(3) 删除触发器

删除触发器 TRI\_Lineitem\_Price\_UPDATE。

```
SHOW TRIGGERS
```

验证截图如下：

信息	摘要	结果 1	剖析	状态			
Trigger	Event	Table	Statement	Timing	Created	sql_modi	
▶ TRI_Lineitem_Quantity_INSERT	INSERT	lineitem	BEGINDECLARE L_valuedi	BEFORE	2022-11-30 19:07:02.39	STRICT_T	
TRI_Lineitem_Price_INSERT	INSERT	lineitem	BEGINDECLAREL_valuedif	AFTER	2022-11-30 19:03:28.19	STRICT_T	
TRI_Lineitem_Quantity_UPDATE	UPDATE	lineitem	BEGINDECLAREL_valuedif	BEFORE	2022-11-30 19:06:17.60	STRICT_T	
TRI_Lineitem_Price_UPDATE	UPDATE	lineitem	BEGINDECLARE L_valuedi	AFTER	2022-11-30 19:02:33.74	STRICT_T	
TRI_Lineitem_Quantity_DELETE	DELETE	lineitem	BEGINDECLARE L_valuedi	BEFORE	2022-11-30 19:07:37.86	STRICT_T	
TRI_Lineitem_Price_DELETE	DELETE	lineitem	BEGINDECLARE L_valuedi	AFTER	2022-11-30 19:04:19.46	STRICT_T	

```
DROP TRIGGER TRI_Lineitem_Price_UPDATE;
SHOW TRIGGERS;
```

验证截图如下：

信息	摘要	结果 1	剖析	状态				
Trigger	Event	Table	Statement	Timing	Created	sql_modi		
▶ TRI_Lineitem_Quantity_INSERT	INSERT	lineitem	BEGINDECLARE L_valuediff	BEFORE	2022-11-30 19:07:02.39	STRICT_T		
TRI_Lineitem_Price_INSERT	INSERT	lineitem	BEGINDECLARE L_valuediff	AFTER	2022-11-30 19:03:28.19	STRICT_T		
TRI_Lineitem_Quantity_UPDATE	UPDATE	lineitem	BEGINDECLARE L_valuediff	BEFORE	2022-11-30 19:06:17.60	STRICT_T		
TRI_Lineitem_Quantity_DELETE	DELETE	lineitem	BEGINDECLARE L_valuediff	BEFORE	2022-11-30 19:07:37.86	STRICT_T		
TRI_Lineitem_Price_DELETE	DELETE	lineitem	BEGINDECLARE L_valuediff	AFTER	2022-11-30 19:04:19.46	STRICT_T		

## 六. 与实验结果相关的文件

无

## 七. 思考题

试设计一个 AFTER 触发器，当 Lineitem 表中的 quantity 变化时，自动计算 Lineitem 表中的 extendedprice 值，同时也要修改 PartSupp 中的 availqty 值(提示:  $\text{extendedprice} = \text{quantity} * \text{Part}.\text{retailprice}$ )。

答:

```
CREATE TRIGGER TRI_Lineitem_Quantity_UPDATE
  BEFORE UPDATE ON Lineitem
  FOR EACH ROW
  BEGIN
    DECLARE L_valuediff INTEGER;
    DECLARE L_availqty INTEGER;
    DECLARE Q_QUANDIFF REAL;
    /*计算订单明细项修改时，订购数量的变化值*/
    set L_valuediff=NEW.quantity-OLD.quantity;
    set Q_QUANDIFF =
    ((NEW.quantity-OLD.quantity)/OLD.quantity)*OLD.extendedprice
    /*查询当前订单明细项对应零件供应记录中的可用数量*/
    SELECT availqty INTO L_availqty
    FROM PartSupp
    WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
    IF(L_availqty-L_valuediff>=0) THEN
    BEGIN
      /*如果可用数量可以满足订单订购数量，则提示 ENOUGH*/
      SELECT 'Available quantity is ENOUGH' into @user_prompt;
      /*修改当前订单明细项对应零件供应记录中的可用数量*/
      UPDATE PartSupp
      SET availqty=availqty-L_valuediff
      WHERE partkey=NEW.partkey AND suppkey=NEW.suppkey;
```

```
        UPDATE Lineitem
        SET extendedprice=extendprice+Q_QUANDIFF
        WHERE orderkey=NEW.orderkey;

    END;
ELSE
    /*如果可用数量不能满足订单订购数量，则更新过程异常中断*/
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Available quantity is NOT ENOUGH';
END IF;
END;
```

## 八. 实验总结