

触发器实验

16337113

劳马东

计算机科学与技术（超算方向）

一、 实验目的

二、 实验内容和要求

三、 实验环境

系统	Windows 10
SQL	MySQL 8.0
工具	MySQL workbench

四、 实验过程

（一） 实体完整性实验

1. AFTER 触发器

1) 在 lineitem 表上定义一个 UPDATE 触发器

```
DELIMITER //  
CREATE TRIGGER TRI_Lineitem_Price_UPDATE  
AFTER UPDATE ON lineitem  
FOR EACH ROW  
BEGIN  
    DECLARE L_valuediff REAL;  
    SET L_valuediff = NEW.extendedprice * (1 - NEW.discount) * (1  
+ NEW.tax) - OLD.extendedprice * (1 - OLD.discount) * (1 +  
OLD.tax);  
    UPDATE orders SET totalprice = totalprice + L_valuediff  
        WHERE orderkey = NEW.orderkey;  
END//
```

2) 在 lineitem 表上定义一个 INSERT 触发器

```
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);
    UPDATE orders SET totalprice = totalprice + L_valuediff
                WHERE orderkey = NEW.orderkey;
END//




```

3) 在 lineitem 表上定义一个 DELETE 触发器

```

CREATE TRIGGER TRI_Lineitem_Price_INSERT
AFTER INSERT ON lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff REAL;
    SET L_valuediff = -OLD.extendedprice * (1 - OLD.discount) *
(1 + OLD.tax);
    UPDATE orders SET totalprice = totalprice + L_valuediff
                WHERE orderkey = OLD.orderkey;
END//

```

Name	Event	Timing	Created
 TRI_Lineitem_Price_INSERT	INSERT	AFTER	2018-11-14 23:1...
 TRI_Lineitem_Price_UPDATE	UPDATE	AFTER	2018-11-14 23:1...
 TRI_Lineitem_Price_DELETE	DELETE	AFTER	2018-11-14 23:2...

4) 验证触发器

```
SELECT totalprice FROM orders WHERE orderkey=1;
```

totalprice
172799.49

A. TRI_Lineitem_Price_UPDATE

```

UPDATE lineitem SET tax = tax + 0.005
WHERE orderkey = 1 AND linenummer = 1;

```

totalprice
172918.09967999998

B. TRI_Lineitem_Price_INSERT

```

INSERT INTO lineitem VALUES(1, 1552, 93, 100, 17, 24710.35, 0.04,
0.025, 'N', 'O', '1996-03-13', '1996-02-12', '1996-03-22',
'DELIVER IN PERSON', 'TRUCK', 'egular courts above the');

```

totalprice
197114.47439999998

C. TRI_Lineitem_Price_DELETE

```
DELETE FROM lineitem WHERE orderkey=1 AND linenummer=1;
```

	totalprice
▶	148603.11528

2. BEFORE 触发器

- 1) 在 lineitem 表上定义一个 UPDATE 触发器

```
DELIMITER //
CREATE TRIGGER TRI_Lineitem_Quantity_UPDATE
BEFORE UPDATE ON lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff, L_availqty INTEGER;

    SET L_valuediff = NEW.quantity - OLD.quantity;
    SELECT availqty INTO L_availqty FROM partsupp
        WHERE partkey = NEW.partkey AND suppkey = NEW.suppkey;
    IF (L_availqty - L_valuediff >= 0) THEN
        BEGIN
            UPDATE partsupp SET availqty = availqty - L_valuediff
                WHERE partkey = NEW.partkey AND suppkey = NEW.suppkey;
        END;
    ELSE
        SIGNAL SQLSTATE '45000' SET message_text='Available quantity is NOT
ENOUGH';
    END IF;
END//
```

- 2) 在 lineitem 表上定义一个 INSERT 触发器

```
CREATE TRIGGER TRI_Quantity_INSERT
BEFORE INSERT ON lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff, L_availqty INTEGER;

    SET L_valuediff = NEW.quantity;
    SELECT availqty INTO L_availqty FROM partsupp
        WHERE partkey = NEW.partkey AND suppkey = NEW.suppkey;
    IF (L_availqty - L_valuediff >= 0) THEN
        BEGIN
            UPDATE partsupp SET availqty = availqty - L_valuediff
                WHERE partkey = NEW.partkey AND suppkey = NEW.suppkey;
        END;
    ELSE
```

```

        SIGNAL SQLSTATE '45000' SET message_text='Available quantity is NOT
ENOUGH';
    END IF;
END//

```

3) 在 lineitem 表上定义一个 DELETE 触发器

```

CREATE TRIGGER TRI_Lineitem_Quantity_DELETE
BEFORE DELETE ON lineitem
FOR EACH ROW
BEGIN
    DECLARE L_valuediff, L_availqty INTEGER;

    SET L_valuediff = -OLD.quantity;
    UPDATE partsupp SET availqty = availqty - L_valuediff
        WHERE partkey = OLD.partkey AND suppkey = OLD.suppkey;
END//

```

Name	Event	Timing	Created
TRI_Lineitem_Quantity_UPDATE	UPDATE	BEFORE	2018-11-15 22:1...
TRI_Lineitem_Quantity_INSERT	INSERT	BEFORE	2018-11-15 22:1...
TRI_Lineitem_Quantity_DELETE	DELETE	BEFORE	2018-11-15 20:4...

4) 验证触发器

```

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 = 1 AND L.linenum=1;

```

	partkey	suppkey	quantity	availqty
▶	1552	93	17	7030

A. TRI_Lineitem_Price_UPDATE

```

UPDATE lineitem SET quantity = quantity + 5
WHERE orderkey = 1 AND linenum = 1;

```

	partkey	suppkey	quantity	availqty
▶	1552	93	22	7025

```

UPDATE lineitem SET quantity = quantity + 10000
WHERE orderkey = 1 AND linenum = 1;

```

✖ 42 22:27:57 UPDATE lineitem SET quantity = quantity + 10000 WHERE orderkey = 1 A... Error Code: 1644. Available quantity is NOT ENOUGH

B. TRI_Lineitem_Price_INSERT

```

INSERT INTO lineitem VALUES(1, 1552, 93, 100, 17, 24710.35, 0.04, 0.025,
'N', 'O', '1996-03-13', '1996-02-12', '1996-03-22', 'DELIVER IN PERSON',
'TRUCK', 'egular courts above the');

```

	partkey	suppkey	quantity	availqty
▶	1552	93	22	7008

```

INSERT INTO lineitem VALUES(1, 1552, 93, 101, 10000, 24710.35, 0.04, 0.025,
'N', 'O', '1996-03-13', '1996-02-12', '1996-03-22', 'DELIVER IN PERSON',

```

```
'TRUCK', 'egular courts above the');
```

✖ 45 22:32:18 INSERT INTO lineitem VALUES(1, 1552, 93, 101, 10000, 24710.35, 0.04, ... Error Code: 1644. Available quantity is NOT ENOUGH

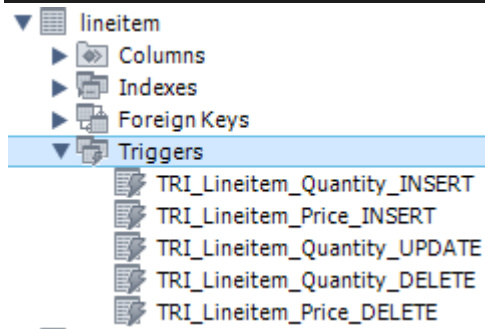
C. TRI_Lineitem_Price_DELETE

```
DELETE FROM lineitem WHERE orderkey = 1 AND linenumber = 100;
```

	partkey	suppkey	quantity	availqty
▶	1552	93	22	7025

3. 删除触发器

```
DROP TRIGGER TRI_Lineitem_Price_UPDATE;
```



五、 实验总结