实习二之触发器 小组成员：1900013049陈福康、1900012455周裕涵、1900013011汪楚皓

In [1]:

```
%load_ext sql
```

In [2]:

```
import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu1900013011:stu1900013011@162.105.146.37:43306
```

In [3]:

```
%sql use stu1900013011;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[3]:

```
[]
```

In [4]:

```
%sql show tables;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[4]:

Tables_in_stu1900013011

实习要求的表my_stock

In [5]:

```
%%sql

CREATE TABLE IF NOT EXISTS my_stock
(
    stock_id INT,
    volume INT,
    avg_price FLOAT,
    profit INT
);
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[5]:

```
[]
```

实习要求的表trans

In [6]:

```
%%sql

CREATE TABLE IF NOT EXISTS trans
(
    trans_id INT,
    stock_id INT,
    date INT,
    price INT,
    amount INT,
    sell_or_buy CHAR
);
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[6]:

```
[]
```

自建的表transcount，用来说明当前的卖出记录和哪条买入记录匹配，匹配的买入记录中已经有多少份额股票被用于匹配之前的卖出记录

In [7]:

```
%%sql

CREATE TABLE IF NOT EXISTS transcount
(
    id_now INT,
    usedcount INT
);
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[7]:

```
[]
```

初始化表transcount，目前的匹配记录初始指向交易ID为1的交易，已用数量初始置为0

In [8]:

```
%%sql
INSERT INTO transcount(id_now, usedcount) VALUES (1, 0);
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[8]:

```
[]
```

触发器的定义。
根据插入的操作是买或卖执行不同分支。交易记录为买入时，如果my_stock表中没有该股票的信息，则新建一行表项，profit置为0。更新avg_price。
交易记录为卖出时，首先判断目前持股数量是否不小于卖出数量，若小于则视为交易无效，不做任何操作。

若满足要求，则从transcount表中取出匹配的买入交易信息，与当前的卖出交易进行匹配。具体做法为从transount表中的buyidnow开始以一个WHILE循环从小到大搜索交易ID，再用一个WHILE循环确定是否为买入交易，发现是买入交易则匹配。根据transcount表中的used_count比较可匹配的的数量和当前卖出交易的数量，如果可匹配的买入数（即买入的总股票数减去used_count）小于当前还需匹配的交易数量（left_count），则把这次买入的交易金额累加到buyprice中，把left_count减去已匹配的买入额，继续搜索下一次买入记录，直到当前卖出记录的所有额度都匹配完毕。根据得到的buyprice更新profit，更新avg_price，同时更新transcount表，更新当前匹配到的买入记录的信息。

In [9]:

```
%%sql

CREATE TRIGGER my_trigger
AFTER INSERT
ON trans
FOR EACH ROW
BEGIN
  DECLARE null_flag INT;
  DECLARE tmp_volume INT;
  DECLARE tmp_avgprice FLOAT;
  DECLARE tmp_profit INT;
  DECLARE buyidnow INT;
  DECLARE used_count INT;
  DECLARE left_count INT;
  DECLARE tmp_price INT;
  DECLARE buycount INT;
  DECLARE buyprice INT;
  DECLARE null_flag2 INT;

  IF NEW.sell_or_buy = 'B'
    THEN
    SELECT COUNT(*) INTO null_flag
      FROM my_stock WHERE stock_id = NEW.stock_id;
    IF null_flag = 0
    THEN INSERT INTO my_stock(stock_id, volume, avg_price, profit) VALUES (NEW.stock_id, NEW.amount,
    ELSE UPDATE my_stock SET avg_price = (volume * avg_price + NEW.price * NEW.amount)/(volume + NE
           WHERE stock_id = NEW.stock_id;
    END IF;
  ELSE
    SELECT volume, profit INTO tmp_volume, tmp_profit
      FROM my_stock WHERE stock_id = NEW.stock_id;
    IF tmp_volume >= NEW.amount
      THEN #match
      SELECT id_now, usedcount INTO buyidnow, used_count
        FROM transcount;
      SELECT amount, price INTO buycount, tmp_price
        FROM trans WHERE trans_id = buyidnow;
      IF NEW.amount > buycount - used_count #needs new match
        THEN BEGIN
        SET buyprice = (buycount - used_count) * tmp_price; #match now
        SET left_count = NEW.amount - (buycount - used_count); #match next
        WHILE left_count > 0 DO #still needs new match
          SET null_flag2 = 0;
          WHILE null_flag2 = 0 DO #search for buy record
            SET buyidnow = buyidnow + 1;
            SELECT COUNT(*) INTO null_flag2
              FROM trans WHERE trans_id = buyidnow AND sell_or_buy = 'B';
          END WHILE;
          SELECT amount, price INTO buycount, tmp_price
            FROM trans WHERE trans_id = buyidnow AND sell_or_buy = 'B';
          SET left_count = left_count - buycount;
          IF left_count < 0 #match over
            THEN BEGIN
            SET buyprice = buyprice + (buycount + left_count) * tmp_price; #new buyprice
            UPDATE transcount SET id_now = buyidnow, usedcount = buycount + left_count; #new transc
            END;
          ELSE SET buyprice = buyprice + buycount * tmp_price; #new buyprice
          END IF;
        END WHILE;
```

```
        END;
      ELSE BEGIN #doesn't need new match
        UPDATE transcount SET usedcount = used_count + NEW.amount; #new transcount
        SET buyprice = NEW.amount * tmp_price; #new buyprice
        END;
      END IF;
      UPDATE my_stock SET avg_price = (tmp_volume * avg_price - NEW.amount * NEW.price)/(tmp_volume
        WHERE stock_id = NEW.stock_id;
    END IF;
  END IF;
END;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[9]:

[]

插入样例数据，验证触发器效果

In [10]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (1, 1, 1, 10, 1000, '
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[10]:

[]

目前的持有情况

In [11]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[11]:

stock_id	volume	avg_price	profit
1	1000	10.0	0

In [12]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (2, 1, 2, 11, 500, 'B
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[12]:

[]

目前的持有情况

In [13]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[13]:

stock_id	volume	avg_price	profit
1	1500	10.3333	0

下面一条数据会和交易ID为1的记录的800股匹配

In [14]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (3, 1, 3, 12, 800, 'S
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[14]:

```
[]
```

目前的持有情况

In [15]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[15]:

stock_id	volume	avg_price	profit
1	700	8.42857	1600

In [16]:

```
%%sql
SELECT * FROM transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[16]:

id_now	usedcount
1	800

下面一条数据会因为交易量大于目前持有量而判定为无效交易，不产生任何效果

In [17]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (4, 1, 4, 12, 1000, '
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[17]:

[]

目前的持有情况

In [18]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[18]:

stock_id	volume	avg_price	profit
1	700	8.42857	1600

In [19]:

```
%%sql
SELECT * FROM transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[19]:

id_now	usedcount
1	800

In [20]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (5, 1, 5, 9, 1000, 'B
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[20]:

[]

目前的持有情况

In [21]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[21]:

stock_id	volume	avg_price	profit
1	1700	8.76471	1600

In [22]:

```
%%sql
SELECT * FROM transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[22]:

id_now	usedcount
1	800

下面一条数据会和交易ID为1的记录的200股匹配，交易ID为2的500股匹配，再和交易ID为5的100股匹配

In [23]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (6, 1, 6, 12, 800, 'S
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[23]:

```
[]
```

目前的持有情况

In [24]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[24]:

stock_id	volume	avg_price	profit
1	900	5.88889	2800

In [25]:

```
%%sql
SELECT * FROM transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[25]:

id_now	usedcount
5	100

下面一条数据会和交易ID为5的记录的800股匹配

In [26]:

```
%%sql
INSERT INTO trans(trans_id, stock_id, date, price, amount, sell_or_buy) VALUES (7, 1, 7, 8, 800, 'S'
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[26]:

```
[]
```

目前的持有情况

In [27]:

```
%%sql
SELECT * FROM my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[27]:

stock_id	volume	avg_price	profit
1	100	-11.0	2000

In [28]:

```
%%sql
SELECT * FROM transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.
```

Out[28]:

id_now	usedcount
5	900

In [29]:

```
%%sql
SELECT * FROM trans;
```

 * mysql://stu1900013011:***@162.105.146.37:43306
7 rows affected.

Out[29]:

trans_id	stock_id	date	price	amount	sell_or_buy
1	1	1	10	1000	B
2	1	2	11	500	B
3	1	3	12	800	S
4	1	4	12	1000	S
5	1	5	9	1000	B
6	1	6	12	800	S
7	1	7	8	800	S

清空表项，删除触发器和表

In [30]:

```
%%sql
DELETE FROM my_stock;
```

 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.

Out[30]:

[]

In [31]:

```
%%sql
DELETE FROM trans;
```

 * mysql://stu1900013011:***@162.105.146.37:43306
7 rows affected.

Out[31]:

[]

In [32]:

```
%%sql
DELETE FROM transcount;
```

 * mysql://stu1900013011:***@162.105.146.37:43306
1 rows affected.

Out[32]:

[]

In [33]:

```
%%sql
DROP TRIGGER my_trigger;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[33]:

[]

In [34]:

```
%%sql
DROP TABLE my_stock;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[34]:

[]

In [35]:

```
%%sql
DROP TABLE trans;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[35]:

[]

In [36]:

```
%%sql
DROP TABLE transcount;
```

```
 * mysql://stu1900013011:***@162.105.146.37:43306
0 rows affected.
```

Out[36]:

[]

In [ ]:

```

```