

练习二：触发器编写

</h1>

游震邦, 区子锐, 夏景涛

1800012963, 1800012945, 1800012912

```
In [1]: %load_ext sql
```

```
In [2]: import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu1800012945:stu1800012945@162.105.146.37:43306
```

```
In [3]: %sql use stu1800012945;

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
```

```
Out[3]: []
```

```
In [4]: %sql show tables;

* mysql://stu1800012945:***@162.105.146.37:43306
3 rows affected.
```

```
Out[4]: Tables_in_stu1800012945
```

my_stock

trans

trans_logs

触发器练习

触发器工作流程

1. 在一开始的时候建立三张表，分别为my_stock, trans和trans_logs, 其中前两张表是按照题目要求所建立，最后一张表用于插入所有的买入记录，方便后续对获取利润profit进行计算。
2. 存储过程get_current_profit对trans_logs有副作用，具体而言，对于每一次合法的卖出记录，我们会调用get_curent_profit来按照要求对买入的记录进行匹配，并对匹配上的买入记录的剩余amount进行修改，直到为0为止，同时记录此买入记录对本次卖出记录的profit的贡献（差价*消耗的amount）。
3. 存储过程update_all对表my_stock和trans_logs有副作用，用于更新股票对应的volume, average price与profit。
4. 触发器before_insert_trans是主要的逻辑。
 - (1) 在每一次插入之前会对交易记录进行检查，如果发现此股票首次出现，会往my_stock中插入一个初始记录。
 - (2) 如果所插入的记录是卖出记录，触发器还需要进行额外的sanity check：如果此次

交易卖出的数量大于my_stock中记录的剩余数量，则说明此次交易记录不合法，通过signal state来阻止记录插入，否则继续进行第（3）步。

（3）经过了第（2）步后，说明此次的交易记录是合法的，此时调用存储过程update_all来维护my_stock中对应的股票的对应数量，平均价格和利润。

```
In [5]: %%sql

set @@foreign_key_checks=0;
drop table if exists my_stock;

CREATE TABLE my_stock
(
    stock_id INT PRIMARY KEY,
    volume INT,
    avg_price FLOAT,
    profit INT DEFAULT 0
);
set @@foreign_key_checks=1;

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[5]: []

```
In [6]: %%sql

set @@foreign_key_checks=0;
drop table if exists trans;

CREATE TABLE trans
(
    trans_id INT PRIMARY KEY,
    stock_id INT,
    date INT,
    price INT,
    amount INT,
    sell_or_buy CHAR(1)
);
set @@foreign_key_checks=1;

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[6]: []

```
In [7]: %%sql

set @@foreign_key_checks=0;
drop table if exists trans_logs;

CREATE TABLE trans_logs
(
    trans_id INT PRIMARY KEY,
    stock_id INT,
    date INT,
```

```

    price INT,
    amount INT
);
set @@foreign_key_checks=1;

```

```

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.

```

Out[7]: []

In [8]: %%sql

```

drop procedure if exists get_current_profit;
create procedure get_current_profit(
    in stock_id INT,
    in sold_volume INT,
    in sold_price INT,
    out delta_total_price FLOAT,
    out current_profit INT
)
begin
    declare rest_volume INT;
    declare current_amount INT default 0;
    declare rest_amount INT default 0;

    declare temp_trans_id INT default 0;
    declare temp_price INT default 0;
    declare temp_amount INT default 0;
    declare delta_price FLOAT default 0;
    declare total_profit INT default 0;
    declare flag INT default 0;

    declare trans_list cursor for
        select
            trans_id, price, amount
        from trans_logs
        where trans_logs.stock_id = stock_id
        order by date
        asc;
    declare continue handler for not found set flag=1;

    set rest_volume = sold_volume;

    open trans_list;
    fetch trans_list into temp_trans_id, temp_price, temp_amount;
    while flag <> 1 do
        if temp_amount > 0 then
            set current_amount = temp_amount;
            if current_amount > rest_volume then
                set current_amount = rest_volume;
            end if;
            set rest_amount = temp_amount - current_amount;
            update trans_logs
            set amount = rest_amount
            where trans_logs.trans_id = temp_trans_id;
            set delta_price = delta_price + current_amount * temp_price;
            set total_profit = total_profit + current_amount * (sold_price - temp_price);
            set rest_volume = rest_volume - current_amount;
            if rest_volume = 0 then
                set flag = 1;
            end if;
        end if;
    end while;
    set delta_total_price = delta_price;
    set current_profit = total_profit;
end procedure;

```

```

        end if;
        fetch trans_list into temp_trans_id, temp_price, temp_amount;
    end while;
    close trans_list;

    set delta_total_price = delta_price;
    set current_profit = total_profit;
end;

```

* mysql://stu1800012945:***@162.105.146.37:43306

0 rows affected.

0 rows affected.

Out[8]: []

In [9]: %%sql

```

drop procedure if exists update_all;
create procedure update_all(
    in stock_id INT,
    in amount INT,
    in price INT,
    in sell_or_buy char(1)
)
begin
    declare old_volume INT default 0;
    declare old_avg_price FLOAT default 0;
    declare old_profit FLOAT default 0;
    declare new_avg_price FLOAT default 0;
    declare new_volume INT default 0;
    declare delta_price FLOAT default 0;
    declare current_profit INT default 0;
    declare new_profit FLOAT default 0;

    declare my_stock_list cursor for
    select volume, avg_price, profit
    from my_stock
    where my_stock.stock_id = stock_id;

    open my_stock_list;
    fetch my_stock_list into old_volume, old_avg_price, old_profit;

    if sell_or_buy = 'B' then
        set new_volume = old_volume + amount;
        set new_avg_price = (old_avg_price * old_volume + price * amount) / new_volume;
        set new_profit = old_profit;
    else
        set new_volume = old_volume - amount;
        call get_current_profit(
            stock_id,
            amount,
            price,
            delta_price,
            current_profit
        );
        set new_avg_price = (old_avg_price * old_volume - price * amount) / new_volume;
        set new_profit = old_profit + current_profit;
    end if;
    close my_stock_list;

    update my_stock
    set
        volume = new_volume,
        avg_price = new_avg_price,

```

```

        profit = new_profit
    where my_stock.stock_id = stock_id;
end

```

```

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.

```

Out[9]: []

In [10]: %%sql

```

drop trigger if exists before_trans_insert;
create trigger before_trans_insert
    before insert on trans
    for each row
begin
    declare current_volume INT default 0;
    if not exists(select * from my_stock where stock_id=new.stock_id) then
        insert into my_stock(stock_id, volume, avg_price, profit)
        values(new.stock_id, 0, 0, 0);
    end if;
    if new.sell_or_buy = 'S' then
        set current_volume = (select volume from my_stock where stock_id=new.stock_id);
        if current_volume < new.amount then
            signal sqlstate '45000' set message_text = "Invalid sold out record.";
        else
            call update_all(new.stock_id, new.amount, new.price, new.sell_or_buy);
        end if;
    elseif new.sell_or_buy = 'B' then
        insert into
            trans_logs(trans_id, stock_id, date, price, amount)
            values(new.trans_id, new.stock_id, new.date, new.price, new.amount);
        call update_all(new.stock_id, new.amount, new.price, new.sell_or_buy);
    end if;
end;

```

```

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.

```

Out[10]: []

```

truncate trans;
truncate my_stock;
truncate trans_logs;
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(1, 1, 1, 10, 1000, 'B');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(2, 1, 2, 11, 500, 'B');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(3, 1, 3, 12, 800, 'S');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(4, 1, 4, 12, 1000, 'S');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(5, 1, 5, 9, 1000, 'B');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(6, 1, 6, 12, 800, 'S');
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(7, 1, 7, 7, 800, 'S');

```

```
In [11]: %%sql
truncate trans;
truncate my_stock;
truncate trans_logs;

* mysql://stu1800012945:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[11]: []

```
In [12]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(1, :

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[12]: []

```
In [13]: %%sql
select * from my_stock;

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[13]: **stock_id volume avg_price profit**

1	1000	10.0	0
---	------	------	---

```
In [14]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(2, :

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[14]: []

```
In [15]: %%sql
select * from my_stock;

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[15]: **stock_id volume avg_price profit**

1	1500	10.3333	0
---	------	---------	---

```
In [16]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(3, :

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[16]: []

```
In [17]: %%sql
select * from my_stock;

* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[17]: **stock_id volume avg_price profit**

1	700	8.42857	1600
---	-----	---------	------

In [18]: %%sql

```
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(4, 1, 4, 12, 1000, 'S');]
* mysql://stu1800012945:***@162.105.146.37:43306
(pymysql.err.OperationalError) (1644, 'Invalid sold out record.')
[SQL: insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(4, 1, 4, 12, 1000, 'S');]
(Background on this error at: https://sqlalche.me/e/14/e3q8)
```

In [19]: %%sql
select * from my_stock;

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[19]: **stock_id volume avg_price profit**

1	700	8.42857	1600
---	-----	---------	------

In [20]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(5, 1, 5, 12, 1000, 'S');

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[20]: []

In [21]: %%sql
select * from my_stock;

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[21]: **stock_id volume avg_price profit**

1	1700	8.76471	1600
---	------	---------	------

In [22]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(6, 1, 6, 12, 1000, 'S');

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[22]: []

In [23]: %%sql
select * from my_stock;

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[23]: **stock_id volume avg_price profit**

1	900	5.88889	2800
---	-----	---------	------

In [24]: %%sql
insert into trans(trans_id, stock_id, date, price, amount, sell_or_buy) values(7, 1, 7, 12, 1000, 'S');

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[24]: []

In [25]: %%sql
select * from my_stock;

```
* mysql://stu1800012945:***@162.105.146.37:43306
1 rows affected.
```

Out[25]:

stock_id	volume	avg_price	profit
1	100	-3.0	1200

1	100	-3.0	1200
---	-----	------	------