

约束设计任务一

身份证校验码：第十八位数字的计算方法为：

1. 将前面的身份证号码17位数分别乘以不同的系数。从第一位到第十七位的系数分别为：7

9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2

2. 将这17位数字和系数相乘的结果相加

3. 用加出来和除以11，看余数是多少

余数0 1 2 3 4 5 6 7 8 9 10分别对应的最后一位身份证的号码为1 0 X 9 8 7 6 5 4 3 2

实现方式：定义一个函数，它接收一个身份证号，并计算校验码，正确返回1，错误返回0。

然后在一个表中定义身份证列，并为该列添加一个check约束，约束条件就是该校验函数。

约束设计任务二

Emp(eid, ename, age, salary)

Dept(did, budget, managerid)

1. 使用函数约束保证管理者的工资必须高于他所管理的任何一个员工
2. 使用触发器保证任何一个员工工资的变化额度，都应该体现在他所在部门的预算上面

触发器

```
my_stock(stock_id, volume, avg_price, profit):
```

表示所持有的股票编号、数量、持仓平均价格、利润

```
trans(trans_id, stock_id, date, price, amount, sell_or_buy):
```

表示一次交易的编号、股票编号、交易日期、成交价格、成交数量、买入还是卖出

使用触发器完成下面的工作：

1. 往trans里面插入一条记录时，根据其是买入还是卖出，调整my_stock中的volume以及avg_price。如果是初次插入的股票交易，就在my_stock中为该股票新建一条记录，profit置为0。注意，如果一笔卖出交易的amount大于my_stock中该股票的volume，说明是无效的下单交易，应该加以拒绝，直接抛弃。

触发器

2. profit的计算方式如下：每当有卖出交易发生时，将其与尽可能远的买入交易进行匹配，比如如果 trans 中现有的记录为 $\{(t01, s01, d01, 10, 1000, \text{buy}), (t02, s01, d02, 12, 500, \text{buy})\}$ ，如果现在插入 $\{(t03, s01, d03, 11, 700, \text{sold})\}$ ，本次交易产生的 $\text{profit} = (11 - 10) * 700$ ，如果再插入 $\{(t04, s01, d04, 9, 700, \text{sold})\}$ ，本次交易产生的 $\text{profit} = (9 - 10) * 300 + (9 - 12) * 400 = -1500$ 。将每次卖出交易的profit都累加到my_stock的profit上。

触发器

trans(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	7	800	S)