实习二：数据库约束设计

本次实习的目标是请同学们体验如何在数据库中利用各种手段完成数据库约束设计，当然一般的主外码约束等比较简单，我们就不做实习练习了。我们主要练习一下如何使用存储过程、函数、触发器等实现一些相对复杂的业务约束。主要包括如下两个练习：

练习一：生成身份证号

身份证各位含义

1、第17位数字（也即身份证号倒数第二位）表示性别：

奇数表示男性，偶数表示女性；

2、第1、2位数字表示：所在省份的代码；

3、第3、4位数字表示：所在城市的代码；

4、第5、6位数字表示：所在区县的代码；

5、第7~14位数字表示：出生年、月、日；

6、第15、16位数字表示：所在地的派出所的代码；

7、第18位数字是校检码。

身份证校验码生成

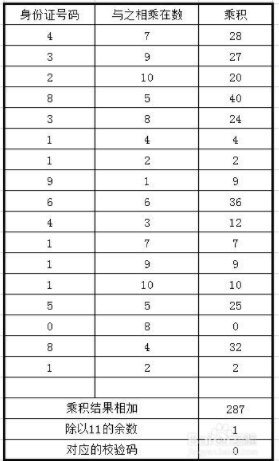
• 将身份证号码17位数分别乘以不同的系数。从第一位到第十七位的系数分别为：7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2

• 将这17位数字和系数相乘的结果相加

• 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

练习一：生成身份证号



有了前述知识准备之后，我们的实习是这样设计的：

• 编写一个函数，它能够根据输入的前17位，生成一个校验码。

• 构造一个存储过程，它接受的参数包括(省、市、区、出生日期、性别)，返回一个身份证号，我们在平台上提供了一个行政区划表，可以查到行政区划代码，第15和16位的派出所代码自己可以随机生成，再调用上面的校验码函数，这样就可以完成身份证的构造了。

练习二：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。

平均价格的计算

1. 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,buy), (t02,s01,d02,12,500,buy)},如果现在插入{(t03,s01,d03,11,700,sold)}，本次交易产生的profit=(11-10)\*700,如果再插入{(t04,s01,d04,9,700,sold)}，本次交易产生的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。

练习二：触发器

练习二：触发器的测试数据

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 )