Chapter 8 Exercise

例1

一、某支付系统数据库有如下关系模式：

E 支付卡	ENO 卡编号	Ename 持卡人姓名	IDcard 身份证号	PassWord 密码	Balance 余额	CalScore 积分	MaxExpend 每笔最大消费额
EC 消费明细	CNO 消费单号	ENO 卡编号	Date 消费日期	Type 消费类型	Expend 消费金额		

提示：
- 消费明细中的每条消费金额必须满足该支付卡所限定的每笔最大消费额。
- 只有在消费明细表中成功地插入了一条消费明细记录后，才表示此次消费有效。
- 假定消费明细的日期为当前的系统日期。

请用指定的方法定义下列约束：

1. 用断言实现：不允许持卡人同一天消费不同的消费类型超过 5 种。
2. 用 SQL3 触发器完成如下操作：若发现某笔消费金额使支付卡余额透支超过 1000 元，则使此次消费无效。若此次消费有效，积分累加本次消费金额的 10%。

1. 断言实现

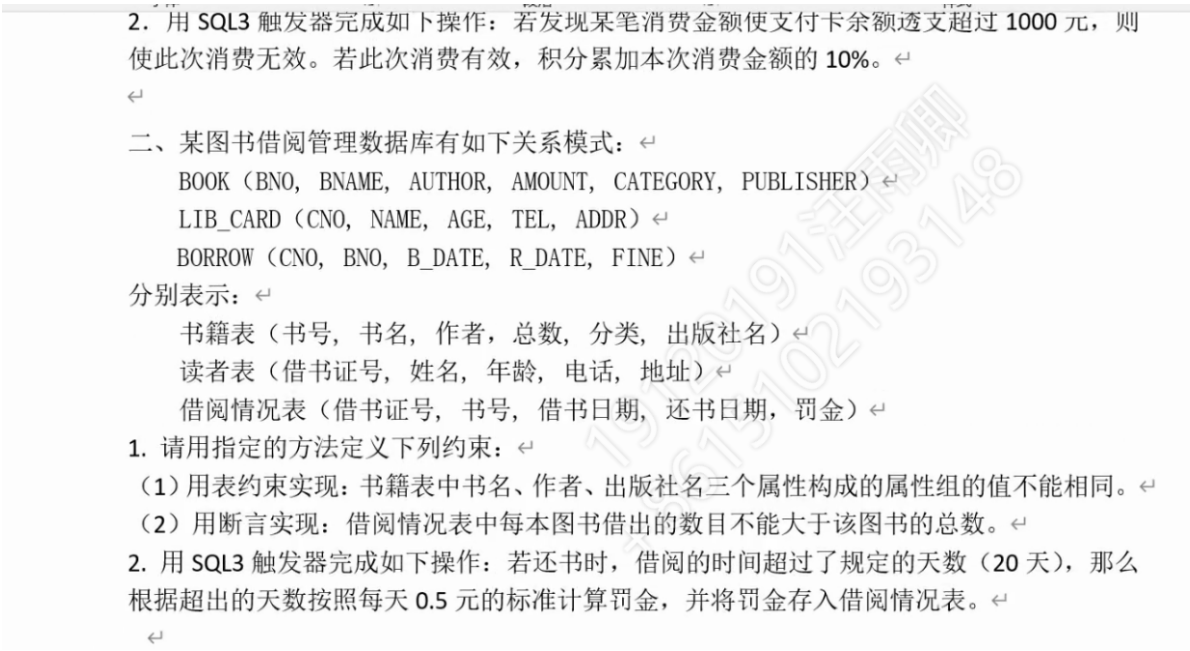
```
CREATE ASSERTION ASSE1 CHECK(
    5 >= ALL(SELECT COUNT(DISTINCT (TYPE)
             FROM EC
             GROUP BY ENO, DATE));

CREATE ASSERTION ASSE2 CHECHK(
    (NOT EXISTS (SELECT * FROM EC
                 GROUP BY ENO,DATE
                 HAVING COUNT(DISTINCT (TYPE) > 5)));
```

2. SQL3触发器

```
CREATE TRIGGER TRIG1
AFTER INSERT ON EC
REFERENCING
    NEW AS NEWTUPLE
WHEN (NEWTUPLE.ENO IS NOT NULL)
BEGIN ATOMIC
    UPDATE E     # 未超支1000，且小于单笔最大支出时，执行更新
    SET BALANCE = BALANCE - NEWTUPLE.EXPAND,
    CALSCORE = CALSCORE + NEWTUPLE.EXPEND * 0.1
    WHERE (NEWTUPLE.ENO = E.ENO) AND
          (E.BALANCE - NEWTUPLE.EXPEND > -1000) AND
          (MAXEXPEND >= NEWTUPLE.EXPEND)
    DELETE FROM EC  # 超支1000，或者单笔支出大于单笔时，撤销更新
        WHERE CNO = NEWTUPLE.CNO AND
            NEWTUPLE.EXPEND >= (SELECT MAXEXPEND FROM E
```

```
                                    WHERE E.ENO = NEWTUPLE.ENO) OR
            (-1000 > (SELECT BALANCE - NEWTUPLE.EXPEND FROM E
                        WHERE E.ENO=NEWTUPLE.ENO))
END
FROM EACH ROW;
```

例2


2. 用 SQL3 触发器完成如下操作：若发现某笔消费金额使支付卡余额透支超过 1000 元，则使此次消费无效。若此次消费有效，积分累加本次消费金额的 10%。

二、某图书借阅管理数据库有如下关系模式：

BOOK（BNO, BNAME, AUTHOR, AMOUNT, CATEGORY, PUBLISHER）

LIB_CARD（CNO, NAME, AGE, TEL, ADDR）

BORROW（CNO, BNO, B_DATE, R_DATE, FINE）

分别表示：

书籍表（书号，书名，作者，总数，分类，出版社名）

读者表（借书证号，姓名，年龄，电话，地址）

借阅情况表（借书证号，书号，借书日期，还书日期，罚金）

1. 请用指定的方法定义下列约束：

（1）用表约束实现：书籍表中书名、作者、出版社名三个属性构成的属性组的值不能相同。

（2）用断言实现：借阅情况表中每本图书借出的数目不能大于该图书的总数。

2. 用 SQL3 触发器完成如下操作：若还书时，借阅的时间超过了规定的天数（20 天），那么根据超出的天数按照每天 0.5 元的标准计算罚金，并将罚金存入借阅情况表。

1. 表约束

```
UNIQUE(BNAME,AUTHOR,PUBLISHER) # 利用UNIQUE实现
```

2. 断言

```
CREATE ASSERTION MAX_AMOUNT CHECK(
    NOT EXISTS (SELECT * FROM BOOK
                    WHERE AMOUNT < SELECT COUNT(*) FROM BORROW
                                    WHERE BORROW.BNO = BOOK.BNO
                                        AND R_DATE IS NULL))     # 注意要未还书
```

3. SQL3

```
CREATE TRIGGER TRIG1
AFTER UPDATE OF R_DATE ON BORROW
REFERENCING
    OLD AS OLDTUPLE
    NEW AS NEWTUPLE
WHEN (NEWTUPLE.R_DATE - NEWTUPLE.B_DATE > 20)
    UPDATE BORROW
        SET FINE = (NEWTUPLE.R_DATE-NEWTUPLE.B_DATE-20)*0.5
        WHERE CNO = NEWTUPLE.CNO AND     # 要加全
            BNO = NEWTUPLE.BNO AND
            B_DATE = NEWTUPLE.B_DATE
FOR EACH ROW;
```