2、（9 分）假设将关系模式 R(ABCDE)分解为 R1(ABC)和 R2(ADE)，如果如下函数依赖 F 成立：A→BC，CD→E，B→D，E→A，问：

（1）（3 分）R 的候选码？并说明理由。

（2）（3 分）证明该分解是无损分解

（3）（3 分）求函数依赖 F 在 R1 和 R2 上的投影，证明该分解不保持函数依赖。

给定关系 R（A，B，C，D，E，F）和函数依赖集 F：{BC → D，BC →E, D → B, E → C, F → A}。问

（1）求出候选键，写出求解过程。

（2）R 最高是第几范式？为什么？

（3）有 R 的一个分解ρ={ R1(B,C,D,E)，R2(A,F)，R3(B,C,F)}，ρ是否保持函数依赖？ρ是否是无损联接分解？

（4）将 R 分解为一组既保持函数依赖又无损分解的 3NF。

给定关系 R（A，B，C，D，E）和函数依赖集 F：{A → E, BD → A, EC → B}。问：

（1）求出候选键，写出求解过程。

（2）R 最高是第几范式？为什么？

（3）有 R 的一个分解ρ={ R1(A,E)， R2(A,B,D)， R3(B,C,E)， R4(C,D,E)}，ρ是否保持函数依赖？ρ是否是无损联接分解？

（4）将 R 无损分解为一组 BCNF。

检查点：

1. 假定系统采用检查点方法，当系统崩溃时产生了以下并发事务的日志记录。

(1) 请给出系统在恢复后搜索日志时所形成的重做队列和撤销队列。

(2) 请给出恢复后各数据项的值。

<T3, start>
<T3, Update, A, 600, 60>
<T1, start>
<T1, Update, C, 50, -10>
<T2, start>
<T2, Update, B, 150, 190>
<T1, Commit>
<T4, start>
<checkpoint L{T2, T3, T4}>
<T4, Update, D, 200, 130>
<T4, Commit>
<T3, Update, B, 190, 320>
<T2, Update, C, -10, 90>
<T2, Commit>

2. 假定系统采用检查点方法，T1、T2 和 T3 是并发事务，在 t_{19} 时发生系统故障，最近的检查点在 t_8 时（见图）；A、B、C 和 D 都是数据库中的数据项，初值依次是 800、300、70 和 80，说明所需的恢复工作。

时刻	T1	T2	T3	说 明
t_1	read(B)			t_1：T1 开始
t_2	B=B*2			
t_3	write(B)			
t_4		read(A)		t_4：T2 开始
t_5		A=A-100		
t_6		write(A)		
t_7		COMMIT		
t_8				t_8：检查点
t_9			read(D)	t_9：T3 开始
t_{10}			D=D-100	
t_{11}			write(D)	
t_{12}	read(A)			
t_{13}	A=A+10			
t_{14}	write(A)			
t_{15}	COMMIT			
t_{16}			read(A)	
t_{17}			read(B)	
t_{18}			B=B+A	
t_{19}			write(B) …	t_{19}：发生故障

并发控制：

1. 判断下列并发调度是不是可串行化调度，为什么？修改如下并发调度为可串行化调度

时间	事务 T1	数据库中值	事务 T2
t_0		A=10, B=2	
t_1	Read A		
t_2			Read B
t_3			Read A
t_4	Update A=A+1		
t_5	Read B		
t_6			Update B=A+B
t_7	Update B=B+1		

1 说明串行的结果

2. 题中的结果说明

3. 结论

2. 操作序列 T1、T2、T3 对数据 A、B、C 并发操作如下所示，T1 与 T2 间并发操作（1），T2 与 T3 间并发操作（2）。修改如下并发调度为可串行化调度

时间	T1	T2	T3
t1	读 A=50		
t2	读 B=200		
t3	X1=A+B		
t4			读 B=200
t5		读 B=200	
t6		B=B-100	
t7		写 B	
t8	读 A=50		
t9	读 B=100		
t10	X1=A+B		
t11	验算不对		B=B+50
t12			写 B

3. 设 T1,T2,T3 是如下三个事务：

T1： A:=A+2； T2： A:=A*2； T3： A:=A^2； 设 A 的初始值为 0.

1）若三个事务允许并发执行，则有多少种可能的正确的结果，请分别列举出来
2）请给出一个可串行化的调度，并给出执行结果
3）请给出一个非串行化的调度，并给出执行结果
4）若三个事务都遵守两段锁协议，请给出一个产生死锁的调度。

完整性

一、某支付系统数据库有如下关系模式：

E	ENO	Ename	IDcard	PassWord	Balance	CalScore	MaxExpend
支付卡	卡编号	持卡人姓名	身份证号	密码	余额	积分	每笔最大消费额

EC	CNO	ENO	Date	Type	Expend		
消费明细	消费单号	卡编号	消费日期	消费类型	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．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。

```
CREATE ASSERTION  ASSE1  CHECK
            (5>=ALL (SELECT COUNT(DISTINCT  (Type )
             FROM  EC
             GROUP BY  ENO, Date))
或：CREATE ASSERTION  ASSE1  CHECK
        (NOT  EXISTS (SELECT  *
         FROM  EC
         GROUP BY  ENO, Date  HAVING COUNT(DISTINCT  (Type )>5)))
```

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%。

```
CREATE TRIGGER TRI1
 AFTER INSERT ON EC
  REFERENCING NEW AS NEWTUPLE
  WHEN (NEWTUPLE.ENO IS NOT NULL)
 BEGIN ATOMIC
  UPDATE E
      SET Balance = Balance -NEWTUPLE.Expend, CalScor=CalScor+NEWTUPLE.Expend*0.1
WHERE (E.ENO = NEWTUPLE.ENO)    AND (Balance-NEWTUPLE. Expend)>-1000))  AND
(Maxexpend>= NEWTUPLE. Expend)
   DELETE FROM EC
    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
```

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．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。

Unique（BNAME, AUTHER, PUBLISHER）↵

（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。↵

```
CREATE ASERSION MAX_AMOUNT CHECK↵
    (Not exists (SELECT * FROM BOOK↵
                WHERE AMOUNT < (SELECT COUNT(*) FROM BORROW↵
                                WHERE BORROW.BNO = BOOK.BNO↵
                                AND R_DATE IS NULL)) ↵
```

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。↵

```
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 BNO=NEWTUPLE.BNO AND
B_DATE=NEWTUPLE.B_DATE↵
FOR EACH ROW; ↵
```

三、设有一个数据库模式，其中有四个关系模式：↵

- 游戏（游戏编号，游戏名称，游戏类别,使用频度）↵
- 类别（类别编号，类别名称）↵
- 玩家（玩家编号，玩家名,性别，等级）↵
- 对决（游戏编号，发起玩家，接受玩家，赢家，对决日期）↵

↵

相应的英文模式定义如下↵

- GAME（GNO， GNAME， GCATE,GFREQ）↵
- CATEGORY (CNO， CNAME)↵
- PLAYER(PNO，PNAME，PSEX，PCLASS，PSCORE)↵
- BATTLE(GNO,APLAYER，RPLAYER，WINNER,BDATE)↵

（1） 自行设计一条完整性规则，条件至少涉及 2 个表或有聚合函数，分别用文字描述约束，并用断言来定义该约束。↵

（2） 你认为该数据库模式上可以设计哪些触发器，请写出一个。要求：先用文字描述该触发器的功能，再定义该触发器。↵