

# 数据库设计 课后作业 (共两大题)

2019.11.18

(参考答案 2019.12.15)

一、假设需要建立一个关于某个学期的期末考试监考安排的关系模式：

$R$ (课程号, 考试时间, 考试教室, 任课教师, 监考教师)

考试时间是由“日期+时间”构成的时间戳属性。如果规定：

- 一门课可以安排多位任课教师，一位教师可以担任多门课的任课教师；
- 每一门课的期末考试，可安排在多个教室中同时进行；
- 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

试回答下列问题：

1. 根据上述规定，请写出关系模式 $R$ 上的最小函数依赖集；
2. 请给出关系模式 $R$ 的所有关键字；
3. 请说明关系模式 $R$ 最高能够满足到第几范式；
4. 请将关系模式 $R$ 直接分解到满足3NF，并具有无损联接性和依赖保持性；
5. 第4小题的分解结果是否满足BCNF？如不满足请将其分解到BCNF；
6. 第5小题的分解结果是否满足4NF？如不满足请将其分解到4NF。

## 第一大题参考答案（计算过程附后）

1. 根据上述规定，请写出关系模式 $R$ 上的最小函数依赖集；

- ① 课程号 $\rightarrow$ 考试时间，
- ③ (任课教师, 考试时间) $\rightarrow$ 课程号
- ⑤ (监考教师, 考试时间) $\rightarrow$ 考试教室
- ⑥ (考试教室, 考试时间) $\rightarrow$ 课程号

2. 请给出关系模式 $R$ 的所有关键字；

(任课教师, 监考教师, 课程号), (任课教师, 监考教师, 考试时间)

3. 请说明关系模式 $R$ 最高能够满足到第几范式；

**1NF**

## 第一大题参考答案 (cont.)

4. 请将关系模式R直接分解到满足3NF，并具有无损联接性和依赖保持性；

子关系	函数依赖集	关键字
R1 (课程号, 考试时间, 任课教师)	课程号→考试时间 (任课教师, 考试时间)→课程号	(任课教师, 考试时间) (任课教师, 课程号)
R2 (课程号, 考试时间, 考试教室)	课程号→考试时间 (考试教室, 考试时间)→课程号	(考试教室, 考试时间) (考试教室, 课程号)
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (课程号, 任课教师, 监考教师)	/	(课程号, 任课教师, 监考教师)

5. 第4小题的分解结果是否满足BCNF？如不满足请将其分解到BCNF；

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (课程号, 任课教师, 监考教师)	/	(课程号, 任课教师, 监考教师)
R5 (课程号, 考试时间)	课程号→考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)

## 第一大题参考答案 (cont.)

6. 第5小题的分解结果是否满足4NF？如不满足请将其分解到4NF。

(选做题，不作要求)

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间) → 考试教室	(监考教师, 考试时间)
R5 (课程号, 考试时间)	课程号 → 考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)
R8 (课程号, 任课教师)	/	(课程号, 任课教师)
R9 (课程号, 监考教师)	/	(课程号, 监考教师)

二、设有一个关系模式**R(A, B, C, D, E, F, G)**，其上的函数依赖集**M**为：

**$\{ A \rightarrow CD, AB \rightarrow EF, AC \rightarrow E, AD \rightarrow E, BC \rightarrow DF, BCD \rightarrow ACE \}$**

1. 计算**M**的最小函数依赖集，并简要写出算法每一步的计算任务和计算结果。
2. 使用算法找出关系**R**的所有候选关键字，并简要写出计算过程。
3. 使用模式分解算法，将关系**R**直接分解到**3NF**，并满足分解的无损联结性和依赖保持性。
4. 上述分解结果是否满足**BCNF**？如不满足请将其分解到**BCNF**。

## 第二大题的参考答案(省略了计算过程)

1. 最小函数依赖集:  $M = \{ A \rightarrow CDE, BC \rightarrow AF \}$
2. 候选关键字:  $\{A,B,G\}$  和  $\{B,C,G\}$
3. 到 3NF 的分解:

子关系	函数依赖集	关键字
R1 (A, C, D, E)	$A \rightarrow CDE$	A
R2 (A, B, C, F)	$BC \rightarrow AF, A \rightarrow C$	两个关键字: BA 和 BC
R3 (A, B, G)	(无)	ABG

4. 到BCNF的分解结果如下:

子关系	函数依赖集	关键字
R1 (A, C, D, E)	$A \rightarrow CDE$	A
R2 (A, B, F)	(无)	ABF
R3 (A, B, G)	(无)	ABG





一、假设需要建立一个关于某个学期的期末考试监考安排的关系模式：

$R$ (课程号, 考试时间, 考试教室, 任课教师, 监考教师)

考试时间是由“日期+时间”构成的时间戳属性。如果规定：

- 一门课可以安排多位任课教师，一位教师可以担任多门课的任课教师；
- 每一门课的期末考试，可安排在多个教室中同时进行；
- 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

试回答下列问题：

1. 根据上述规定，请写出关系模式 $R$ 上的最小函数依赖集；
2. 请给出关系模式 $R$ 的所有关键字；
3. 请说明关系模式 $R$ 最高能够满足到第几范式；
4. 请将关系模式 $R$ 直接分解到满足3NF，并具有无损联接性和依赖保持性；
5. 第4小题的分解结果是否满足BCNF？如不满足请将其分解到BCNF；
6. 第5小题的分解结果是否满足4NF？如不满足请将其分解到4NF。

一、假设需要建立一个关于某个学期的期末考试监考安排的关系模式：

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

考试时间是由“日期+时间”构成的时间戳属性。如果规定：

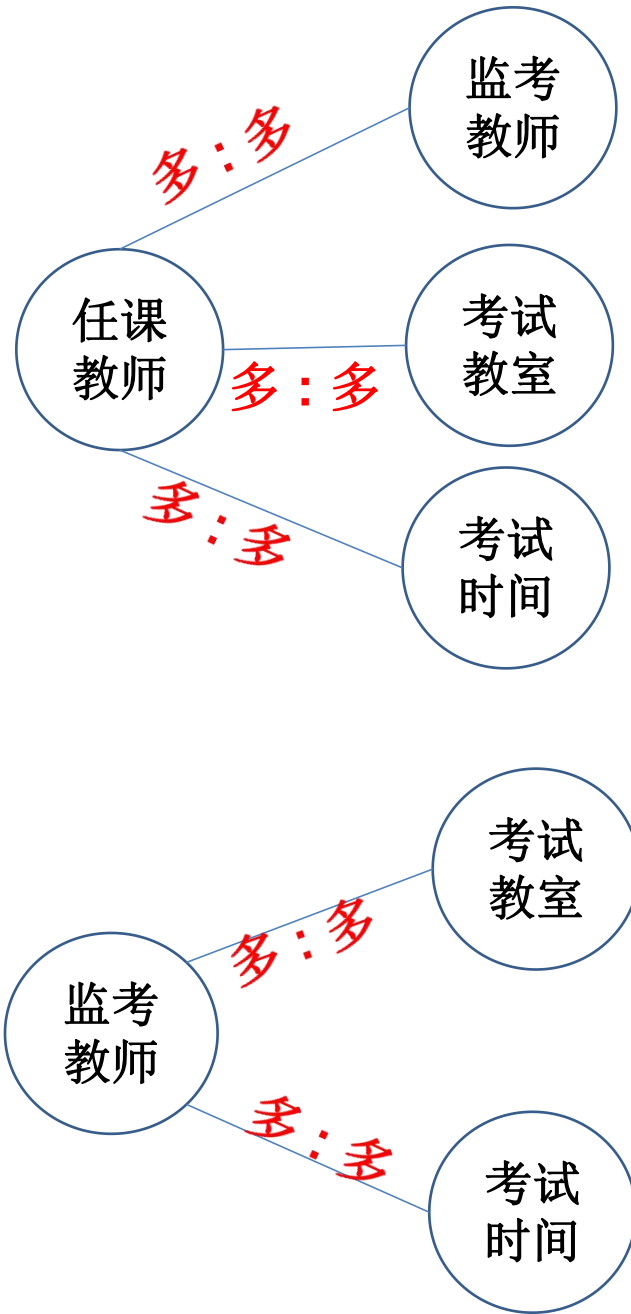
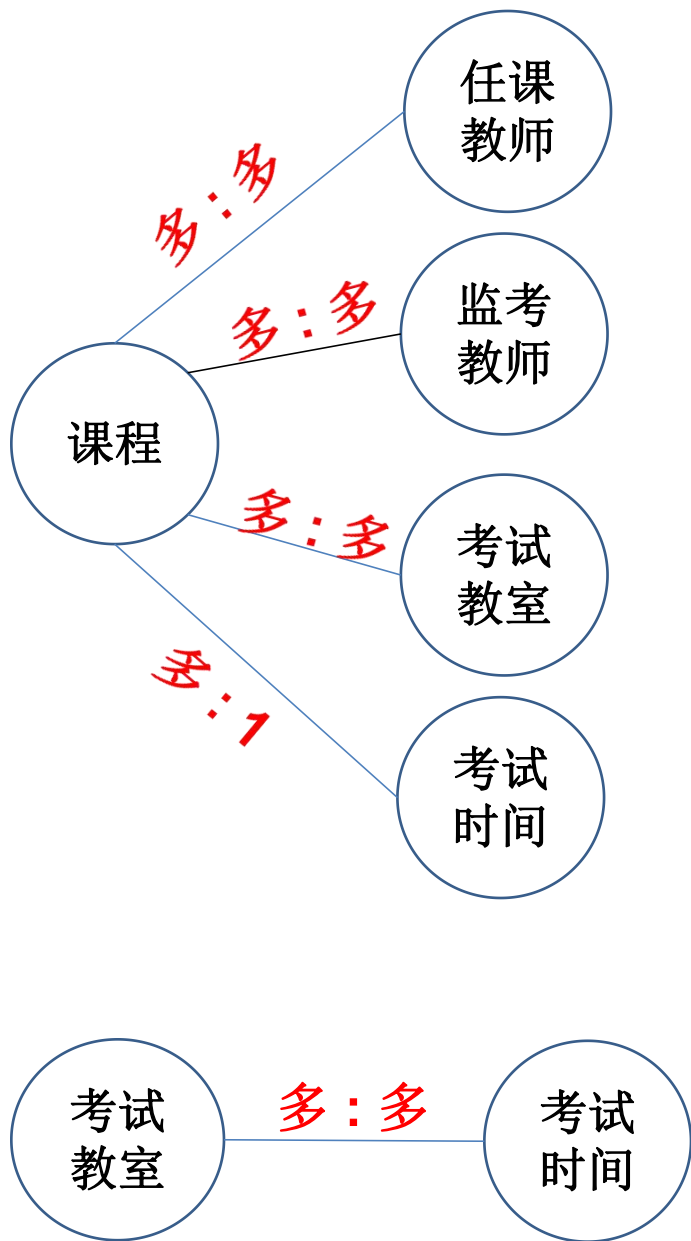
- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

试回答下列问题：

1. 根据上述规定，请写出关系模式R上的最小函数依赖集；

**解：**系统中包括课程（课程号），任课教师，监考教师，考试教室，考试时间等五个方面的数据信息。分别考虑他们相互之间的取值数量对应关系。

- 只有‘课程号’和‘考试时间’之间是“多对一”，其他都是“多对多” (next slide)
- 所以有函数依赖：**课程号→考试时间**



**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间 }**

**(cont.) 其次，任选两类数据组合在一起，检查是否存在以它们作为决定因素的函数依赖**

**一共有10种情况，分别讨论如下：**

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间 }**

**情况1：以“课程号+任课教师”作为决定因素**

- “课程号+任课教师” vs. “监考教师” 是 “多:多”，不存在函数依赖
- “课程号+任课教师” vs. “考试教室” 是 “多:多”，不存在函数依赖
- “课程号+任课教师” vs. “考试时间” 是 “多:1”

**但我们不需要再写出该函数依赖 (课程号, 任课教师) → 考试时间，为什么？**

**答：这是一个部分函数依赖，可以被简化为已经发现的“课程号 → 考试时间”**

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间 }**

**情况2：以“课程号+监考教师”作为决定因素**

- “课程号+监考教师” vs. “任课教师” 是 “多:多”，不存在函数依赖
- “课程号+监考教师” vs. “考试教室” 是 “多:1”，所以有函数依赖：  
**(课程号, 监考教师) → 考试教室**
- “课程号+监考教师” vs. “考试时间” 是 “多:1”

**但我们同样不需要再写出该函数依赖（理由同前）**

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号→考试时间, (课程号, 监考教师)→考试教室 }**

**情况3：以“课程号+考试教室”作为决定因素**

- “课程号+考试教室” vs. “任课教师” 是 “多:多”，不存在函数依赖
- “课程号+考试教室” vs. “监考教师” 是 “多:多”，不存在函数依赖
- “课程号+考试教室” vs. “考试时间” 是 “多:1”

**也不需要写出该函数依赖（理由同前）**

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室 }**

**情况4：以“课程号+考试时间”作为决定因素**

- **该种情况不需要再考虑了！ WHY？**

**∴ 课程号 → 考试时间**

**∴ 如果存在以“课程号+考试时间”为决定因素的函数依赖**

**那么该函数依赖也是一个部分函数依赖，一定可以被化简为以“课程号”为决定因素的函数依赖。**



**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室 }**

**情况5：以“任课教师+监考教师”作为决定因素**

- “任课教师+监考教师” vs. “课程号” 是 “多:多”，不存在函数依赖
- “任课教师+监考教师” vs. “考试教室” 是 “多:多”，不存在函数依赖
- “任课教师+监考教师” vs. “考试时间” 是 “多:多”，不存在函数依赖

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室 }**

**情况6：以“任课教师+考试教室”作为决定因素**

- “任课教师+考试教室” vs. “课程号” 是“多:多”，不存在函数依赖
- “任课教师+考试教室” vs. “监考教师” 是“多:多”，不存在函数依赖
- “任课教师+考试教室” vs. “考试时间” 是“多:多”，不存在函数依赖

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室 }**

**情况7：以“任课教师+考试时间”作为决定因素**

- “任课教师+考试时间” vs. “监考教师” 是“多:多”，不存在函数依赖
- “任课教师+考试时间” vs. “考试教室” 是“多:多”，不存在函数依赖
- “任课教师+考试时间” vs. “课程号” 是“多:1”，所以有：  
**(任课教师, 考试时间) → 课程号**

原因：同一位任课教师的不同课程，不会被安排在同一段时间进行考试！

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

{ 课程号→考试时间, (课程号, 监考教师)→考试教室  
(任课教师, 考试时间)→课程号 }

**情况8：以“监考教师+考试教室”作为决定因素**

- “监考教师+考试教室” vs. “课程号” 是“多:多”，不存在函数依赖
- “监考教师+考试教室” vs. “任课教师” 是“多:多”，不存在函数依赖
- “监考教师+考试教室” vs. “考试时间” 是“多:多”，不存在函数依赖

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

**{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室  
(任课教师, 考试时间) → 课程号 }**

**情况9：以“监考教师+考试时间”作为决定因素**

- “监考教师+考试时间” vs. “任课教师” 是“多:多”，不存在函数依赖
- “监考教师+考试时间” vs. “课程号” 是“多:1”，所以有：  
**(监考教师, 考试时间) → 课程号**
- “监考教师+考试时间” vs. “考试教室” 是“多:1”，所以有：  
**(监考教师, 考试时间) → 考试教室**

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

{ 课程号 → 考试时间, (课程号, 监考教师) → 考试教室  
(任课教师, 考试时间) → 课程号  
(监考教师, 考试时间) → 课程号 (监考教师, 考试时间) → 考试教室 }

**情况10：以“考试教室+考试时间”作为决定因素**

- “考试教室+考试时间” vs. “任课教师”是“多:多”，不存在函数依赖
- “考试教室+考试时间” vs. “监考教师”是“多:多”，不存在函数依赖
- “考试教室+考试时间” vs. “课程号”是“多:1”，所以有：  
(考试教室, 考试时间) → 课程号

**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1、已经发现的函数依赖有：**

<b>{ 课程号 → 考试时间,</b>	<b>(课程号, 监考教师) → 考试教室</b>
<b>(任课教师, 考试时间) → 课程号</b>	
<b>(监考教师, 考试时间) → 课程号</b>	<b>(监考教师, 考试时间) → 考试教室</b>
<b>(考试教室, 考试时间) → 课程号</b>	<b>}</b>

- **(cont.)**对发现的函数依赖集进行检查（即进行最小覆盖的计算），可以发现其中存在一些冗余函数依赖

- ① 课程号 $\rightarrow$ 考试时间,
- ② (课程号, 监考教师) $\rightarrow$ 考试教室
- ③ (任课教师, 考试时间) $\rightarrow$ 课程号
- ④ (监考教师, 考试时间) $\rightarrow$ 课程号
- ⑤ (监考教师, 考试时间) $\rightarrow$ 考试教室
- ⑥ (考试教室, 考试时间) $\rightarrow$ 课程号

- 如果按照如上的函数依赖的排列顺序进行检查, 可发现②和④是冗余的
  - a)  $\because$  课程号 $\rightarrow$ 考试时间  $\therefore$  (课程号, 监考教师) $\rightarrow$ (监考教师, 考试时间)  
由(课程号, 监考教师) $\rightarrow$ (监考教师, 考试时间)和函数依赖⑤可推到得到函数依赖②
  - b)  $\because$  (监考教师, 考试时间) $\rightarrow$ 考试教室  $\therefore$  (监考教师, 考试时间) $\rightarrow$ (考试教室, 考试时间)  
由(监考教师, 考试时间) $\rightarrow$ (考试教室, 考试时间)和函数依赖⑥可推到得到函数依赖④
- 所以, 最小函数依赖集的计算结果是:
  - ① 课程号 $\rightarrow$ 考试时间,
  - ③ (任课教师, 考试时间) $\rightarrow$ 课程号
  - ⑤ (监考教师, 考试时间) $\rightarrow$ 考试教室
  - ⑥ (考试教室, 考试时间) $\rightarrow$ 课程号



**R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

- ① 一门课可以安排多位任课教师，一位老师可以担任多门课的任课教师；
- ② 一门课的期末考试，可安排在多个教室中同时进行；
- ③ 在一场考试中，每个考试教室中可安排多位监考教师，每一位监考教师必须在指定的教室中进行监考；
- ④ 同一时间，在一间教室里只能安排一门课程的考试；同一位任课教师讲授的课程，不能被安排在同一时间里考试。

**1. 答：综上所述，其最小函数依赖集为：**

**{ 课程号 → 考试时间**

**(任课教师, 考试时间) → 课程号**

**(考试教室, 考试时间) → 课程号**

**(监考教师, 考试时间) → 考试教室 }**

关系模式:  $R(\text{课程号}, \text{考试时间}, \text{考试教室}, \text{任课教师}, \text{监考教师})$

1、最小函数依赖集: { 课程号  $\rightarrow$  考试时间

(任课教师, 考试时间)  $\rightarrow$  课程号

(考试教室, 考试时间)  $\rightarrow$  课程号

(监考教师, 考试时间)  $\rightarrow$  考试教室 }

2、请给出关系模式 $R$ 的所有候选关键字;

答: 分析上述的函数依赖关系, 得到:

- 只在函数依赖的左边出现过的属性是: 任课教师, 监考教师
- 只在函数依赖的右边出现过的属性是: (无)
- 在函数依赖的左右两边都出现过的属性是: 课程号, 考试教室, 考试时间
- 因此,
  - ① '任课教师'和'监考教师'是每一个关键字的组成部分, 先判断(任课教师, 监考教师)能否构成关键字?
  - ② 如果第①问的结论是否定的, 再用'任课教师+监考教师'和另外三个属性分别进行组合验证, 可以得到本关系的两个候选关键字, 分别是:  
(任课教师, 监考教师, 课程号), (任课教师, 监考教师, 考试时间)

关系模式:  $R(\text{课程号}, \text{考试时间}, \text{考试教室}, \text{任课教师}, \text{监考教师})$

1、最小函数依赖集:

$\{ \text{课程号} \rightarrow \text{考试时间}, (\text{任课教师}, \text{考试时间}) \rightarrow \text{课程号}$   
 $(\text{考试教室}, \text{考试时间}) \rightarrow \text{课程号}, (\text{监考教师}, \text{考试时间}) \rightarrow \text{考试教室} \}$

2. 关键字:

$(\text{任课教师}, \text{监考教师}, \text{课程号})$      $(\text{任课教师}, \text{监考教师}, \text{考试时间})$

3、请说明关系模式 $R$ 最高能够满足到第几范式;

答:

- 关系 $R$ 的主属性集是:  $\{ \text{任课教师}, \text{监考教师}, \text{课程号}, \text{考试时间} \}$
- 关系 $R$ 的非主属性集是:  $\{ \text{考试教室} \}$

$\therefore$  “ $(\text{监考教师}, \text{考试时间}) \rightarrow \text{考试教室}$ ” 不满足2NF的定义 (右边是单个非主属性, 而左边则是第二个候选关键字的真子集)

$\therefore$  所以关系 $R$ 不满足2NF

$\therefore$  所以关系 $R$ 最高只能满足1NF

(注: 我们从‘是否满足2NF’开始检查, 如果不满足2NF, 当然也可能满足3NF和BCNF。事实上, 函数依赖 “ $(\text{监考教师}, \text{考试时间}) \rightarrow \text{考试教室}$ ” 也不满足3NF的定义, 所有四个函数依赖都不满足BCNF的定义。)

**关系模式: R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

**1、最小函数依赖集:**

{ 课程号  $\rightarrow$  考试时间, (任课教师, 考试时间)  $\rightarrow$  课程号  
(考试教室, 考试时间)  $\rightarrow$  课程号, (监考教师, 考试时间)  $\rightarrow$  考试教室 }

**2. 关键字:**

(任课教师, 监考教师, 课程号) (任课教师, 监考教师, 考试时间)

**4、请将关系模式R直接分解到满足3NF，并具有无损联接性和依赖保持性；**

**答：根据我们上课介绍的直接到3NF的分解算法6.8.8，初步结果如下：**

**step 3:** 根据最小函数依赖集的分解结果如下（其中：第一个函数依赖中的两个属性，也同时出现在第二和第三个函数依赖中，就不必再单独分解为一个子关系了）

- **R1 (课程号, 考试时间, 任课教师)**
- **R2 (课程号, 考试时间, 考试教室)**
- **R3 (考试时间, 考试教室, 监考教师)**

**step 4:** 在step 3的分解结果中，没有一个子关系能够包含原关系的一个候选关键字，所以需要选择原关系的一个候选关键字，并将其作为一个新的子关系并加入到分解结果中去（这里，我们选择的是第一个候选关键字）

- **R4 (课程号, 任课教师, 监考教师)**

关系模式: **R**(课程号, 考试时间, 考试教室, 任课教师, 监考教师)

1、最小函数依赖集:

{ 课程号→考试时间, (任课教师, 考试时间)→课程号  
(考试教室, 考试时间)→课程号, (监考教师, 考试时间)→考试教室 }

2. 关键字:

(任课教师,监考教师,课程号) (任课教师,监考教师,考试时间)

4、请将关系模式**R**直接分解到满足**3NF**，并具有无损联接性和依赖保持性；

(cont.) 到**3NF**的最终分解结果如下:

子关系	函数依赖集	关键字
<b>R1 (课程号, 考试时间, 任课教师)</b>	课程号→考试时间 (任课教师, 考试时间)→课程号	(任课教师, 考试时间) (任课教师, 课程号)
<b>R2 (课程号, 考试时间, 考试教室)</b>	课程号→考试时间 (考试教室, 考试时间)→课程号	(考试教室, 考试时间) (考试教室, 课程号)
<b>R3 (考试时间, 考试教室, 监考教师)</b>	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
<b>R4 (课程号, 任课教师, 监考教师)</b>	/	(课程号, 任课教师, 监考教师)

按照**3NF**的定义逐一检查上述的四个关系，它们都能满足**3NF**的定义，并且该分解满足无损联接和依赖保持。

5、第4小题的分解结果是否满足BCNF？如不满足请将其分解到BCNF；

子关系	函数依赖集	关键字
R1 (课程号, 考试时间, 任课教师)	课程号→考试时间 (任课教师, 考试时间)→课程号	(任课教师, 考试时间) (任课教师, 课程号)
R2 (课程号, 考试时间, 考试教室)	课程号→考试时间 (考试教室, 考试时间)→课程号	(考试教室, 考试时间) (考试教室, 课程号)
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (课程号, 任课教师, 监考教师)	/	(课程号, 任课教师, 监考教师)

答：R1和R2不满足BCNF，R3和R4满足BCNF。

根据我们最后补充的一般化的分解算法，

- 可将 R1 分解为 R5(课程号,考试时间) 和 R6(课程号,任课教师)
- 可将 R2 分解为 R5(课程号,考试时间) 和 R7(课程号,考试教室)

其中，R6的属性集合是R4属性集合的一个子集，所以可将R6合并到R4中去，最终到BCNF的分解结果如下：

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (课程号, 任课教师, 监考教师)	/	(课程号, 任课教师, 监考教师)
R5 (课程号, 考试时间)	课程号→考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)

6、第5小题的分解结果是否满足4NF？如不满足请将其分解到4NF。

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间) → 考试教室	(监考教师, 考试时间)
R4 (课程号, 任课教师, 监考教师)	/	(课程号, 任课教师, 监考教师)
R5 (课程号, 考试时间)	课程号 → 考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)

答：（本小题是选做题，不作要求）R3, R5, R7都满足4NF，R4不满足4NF

∴ 在R4中存在不是函数依赖的非平凡多值依赖：

‘课程号 →→ 任课教师’ 和 ‘课程号 →→ 监考教师’

∴ R4不满足4NF，可继续分解为：R8(课程号,任课教师)和R9(课程号,监考教师)

最终到4NF的分解结果如下：

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间) → 考试教室	(监考教师, 考试时间)
R5 (课程号, 考试时间)	课程号 → 考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)
R8 (课程号, 任课教师)	/	(课程号, 任课教师)
R9 (课程号, 监考教师)	/	(课程号, 监考教师)

- 如果在28页第4小题，到3NF的分解算法中，我们选择第二个候选关键字(任课教师,监考教师,考试时间)来做分解，则后续设计结果如下。



**关系模式: R(课程号, 考试时间, 考试教室, 任课教师, 监考教师)**

**1、最小函数依赖集:**

{ 课程号  $\rightarrow$  考试时间, (任课教师, 考试时间)  $\rightarrow$  课程号  
(考试教室, 考试时间)  $\rightarrow$  课程号, (监考教师, 考试时间)  $\rightarrow$  考试教室 }

**2. 关键字:**

(任课教师, 监考教师, 课程号) (任课教师, 监考教师, 考试时间)

**4、请将关系模式R直接分解到满足3NF，并满足无损联接性和依赖保持性；**

**答：根据我们上课介绍的直接到3NF的分解算法，初步结果如下：**

**step 3:** 根据最小函数依赖集的分解结果如下（其中：第一个函数依赖中的两个属性，也同时出现在第二和第三个函数依赖中，就不必再单独分解为一个子关系了）

- **R1 (课程号, 考试时间, 任课教师)**
- **R2 (课程号, 考试时间, 考试教室)**
- **R3 (考试时间, 考试教室, 监考教师)**

**step 4:** 如果选择第二个候选关键字做分解，则结果如下：

- **R4 (任课教师, 监考教师, 考试时间)**

关系模式: **R**(课程号, 考试时间, 考试教室, 任课教师, 监考教师)

1、最小函数依赖集:

{ 课程号→考试时间, (任课教师, 考试时间)→课程号  
(考试教室, 考试时间)→课程号, (监考教师, 考试时间)→考试教室 }

2. 关键字:

(任课教师,监考教师,课程号)      (任课教师,监考教师,考试时间)

4、请将关系模式**R**直接分解到满足**3NF**，并满足无损联接性和依赖保持性；

(cont.) 到**3NF**的最终分解结果如下:

子关系	函数依赖集	关键字
<b>R1 (课程号, 考试时间, 任课教师)</b>	课程号→考试时间 (任课教师, 考试时间)→课程号	(任课教师, 考试时间) (任课教师, 课程号)
<b>R2 (课程号, 考试时间, 考试教室)</b>	课程号→考试时间 (考试教室, 考试时间)→课程号	(考试教室, 考试时间) (考试教室, 课程号)
<b>R3 (考试时间, 考试教室, 监考教师)</b>	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
<b>R4 (任课教师, 监考教师, 考试时间)</b>	/	(任课教师, 监考教师, 考试时间)

按照**3NF**的定义逐一检查上述的四个关系，它们都能满足**3NF**的定义，并且该分解满足无损联接和依赖保持。

子关系	函数依赖集	关键字
R1 (课程号, 考试时间, 任课教师)	课程号→考试时间 (任课教师, 考试时间)→课程号	(任课教师, 考试时间) (任课教师, 课程号)
R2 (课程号, 考试时间, 考试教室)	课程号→考试时间 (考试教室, 考试时间)→课程号	(考试教室, 考试时间) (考试教室, 课程号)
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (任课教师, 监考教师, 考试时间)	/	(任课教师, 监考教师, 考试时间)

5、第4小题的分解结果是否满足BCNF？如不满足请将其分解到BCNF；

答：R1和R2不满足BCNF，R3和R4满足BCNF。

根据我们最后补充的一般化的分解算法，

- 可将 R1 分解为 R5(课程号,考试时间) 和 R6(课程号,任课教师)
- 可将 R2 分解为 R5(课程号,考试时间) 和 R7(课程号,考试教室)

最终，到BCNF的分解结果如下：

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间)→考试教室	(监考教师, 考试时间)
R4 (任课教师, 监考教师, 考试时间)	/	(任课教师, 监考教师, 考试时间)
R5 (课程号, 考试时间)	课程号→考试时间	课程号
R6 (课程号, 任课教师)	/	(课程号, 任课教师)
R7 (课程号, 考试教室)	/	(课程号, 考试教室)

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间) → 考试教室	(监考教师, 考试时间)
R4 (任课教师, 监考教师, 考试时间)	/	(任课教师, 监考教师, 考试时间)
R5 (课程号, 考试时间)	课程号 → 考试时间	课程号
R6 (课程号, 任课教师)	/	(课程号, 任课教师)
R7 (课程号, 考试教室)	/	(课程号, 考试教室)

6、第5小题的分解结果是否满足4NF？如不满足请将其分解到4NF。

答：以上的所有关系都满足4NF

第一种模式分解方案的结果如下：

子关系	函数依赖集	关键字
R3 (考试时间, 考试教室, 监考教师)	(监考教师, 考试时间) → 考试教室	(监考教师, 考试时间)
R5 (课程号, 考试时间)	课程号 → 考试时间	课程号
R7 (课程号, 考试教室)	/	(课程号, 考试教室)
R8 (课程号, 任课教师)	/	(课程号, 任课教师)
R9 (课程号, 监考教师)	/	(课程号, 监考教师)

两者之间的区别仅仅是：

上表中的 R4(任课教师, 监考教师, 考试时间)  
变成了下表中的 R9(课程号, 监考教师)



二、设有一个关系模式**R(A, B, C, D, E, F, G)**，其上的函数依赖集**M**为：

**$\{ A \rightarrow CD, AB \rightarrow EF, AC \rightarrow E, AD \rightarrow E, BC \rightarrow DF, BCD \rightarrow ACE \}$**

1. 计算**M**的最小函数依赖集，并简要写出算法每一步的计算任务和计算结果。
2. 使用算法找出关系**R**的所有候选关键字，并简要写出计算过程。
3. 使用模式分解算法，将关系**R**直接分解到**3NF**，并满足分解的无损联结性和依赖保持性。
4. 上述分解结果是否满足**BCNF**？如不满足请将其分解到**BCNF**。

**R (A, B, C, D, E, F, G)**

**M = {  $A \rightarrow CD$ ,  $AB \rightarrow EF$ ,  $AC \rightarrow E$ ,  $AD \rightarrow E$ ,  $BC \rightarrow DF$ ,  $BCD \rightarrow ACE$  }**

**1. 计算M的最小函数依赖集，并简要写出算法每一步的计算任务和计算结果.**

**step 1: 将每一个函数依赖分解为右边只含单个属性**

**M1 = {  $A \rightarrow C$ ,  $A \rightarrow D$ ,  
 $AB \rightarrow E$ ,  $AB \rightarrow F$ ,  
 $AC \rightarrow E$ ,  
 $AD \rightarrow E$ ,  
 $BC \rightarrow D$ ,  $BC \rightarrow F$ ,  
 $BCD \rightarrow A$ ,  $BCD \rightarrow C$ ,  $BCD \rightarrow E$  }**

$M1 = \{ A \rightarrow C, A \rightarrow D, AB \rightarrow E, AB \rightarrow F, AC \rightarrow E, AD \rightarrow E, \\ BC \rightarrow D, BC \rightarrow F, BCD \rightarrow A, BCD \rightarrow C, BCD \rightarrow E \}$

step 2: 消除M1中的部分函数依赖

①  $BCD \rightarrow C$  是平凡函数依赖，可以直接被剔除

② 考虑对  $BCD \rightarrow A$  和  $BCD \rightarrow E$  的化简

- 去掉B，计算  $\{C, D\}_{M1}^+ = \{C, D\}$

- 去掉C，计算  $\{B, D\}_{M1}^+ = \{B, D\}$

- 去掉D，计算  $\{B, C\}_{M1}^+ = \{B, C, D, F, A, E\}$

$\therefore$  将  $BCD \rightarrow A$  化简为  $BC \rightarrow A$ ，将  $BCD \rightarrow E$  化简为  $BC \rightarrow E$

③ 再考虑左边的决定因素是单个属性情况（两个属性再化简就变成了单个属性）

- 计算:  $\{A\}_{M1}^+ = \{A, C, D, E\}$

$\{B\}_{M1}^+ = \{B\}$        $\{C\}_{M1}^+ = \{C\}$        $\{D\}_{M1}^+ = \{D\}$

$\therefore$  可以将  $AB \rightarrow E$ 、 $AC \rightarrow E$  和  $AD \rightarrow E$  全部化简为  $A \rightarrow E$



step 2: 因此, 步骤2的计算结果如右所示的 M2

M1 = {  $A \rightarrow C$ ,

$A \rightarrow D$ ,

~~$AB \rightarrow E$ ,~~

$AB \rightarrow F$ ,

~~$AC \rightarrow E$ ,~~

~~$AD \rightarrow E$ ,~~

$BC \rightarrow D$ ,

$BC \rightarrow F$ ,

~~$BCD \rightarrow A$ ,~~

~~$BCD \rightarrow C$ ,~~

~~$BCD \rightarrow E$~~

}

M2 = {  $A \rightarrow C$ ,

$A \rightarrow D$ ,

$A \rightarrow E$ ,

$AB \rightarrow F$ ,

$BC \rightarrow D$ ,

$BC \rightarrow F$ ,

$BC \rightarrow A$ ,

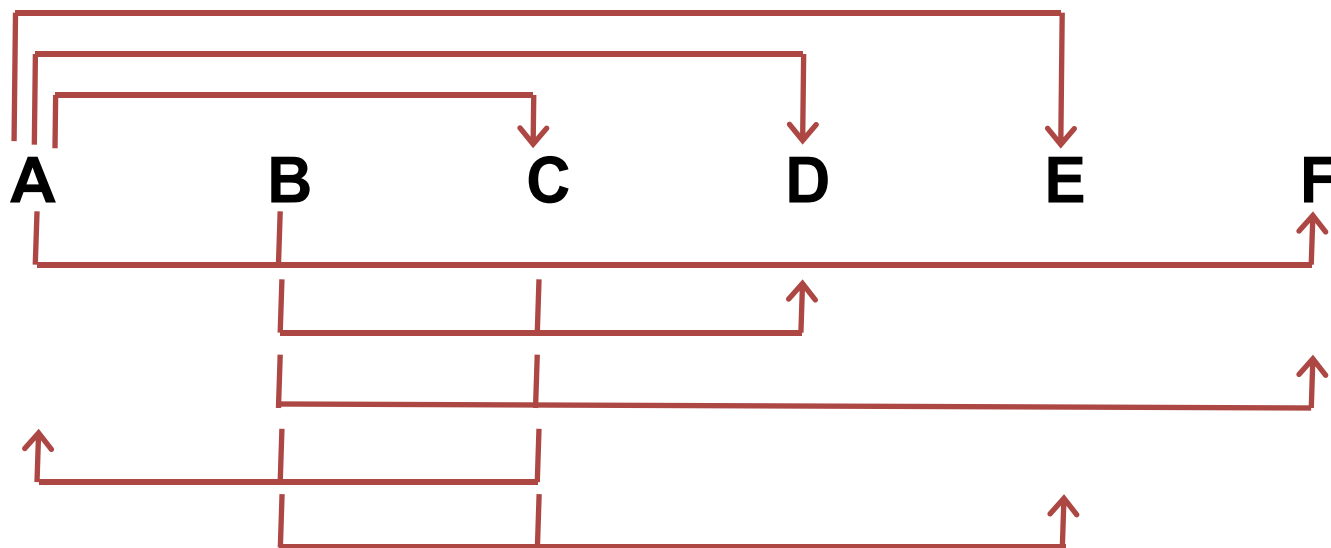
$BC \rightarrow E$  }

步骤2的计算结果 **M2** 如下：

$\{A \rightarrow C, A \rightarrow D, A \rightarrow E, AB \rightarrow F, BC \rightarrow D, BC \rightarrow F, BC \rightarrow A, BC \rightarrow E\}$

**step 3:** 消除**M2**中冗余的函数依赖

**分析：** 先将**M2**中的函数依赖画成如下所示的示意图



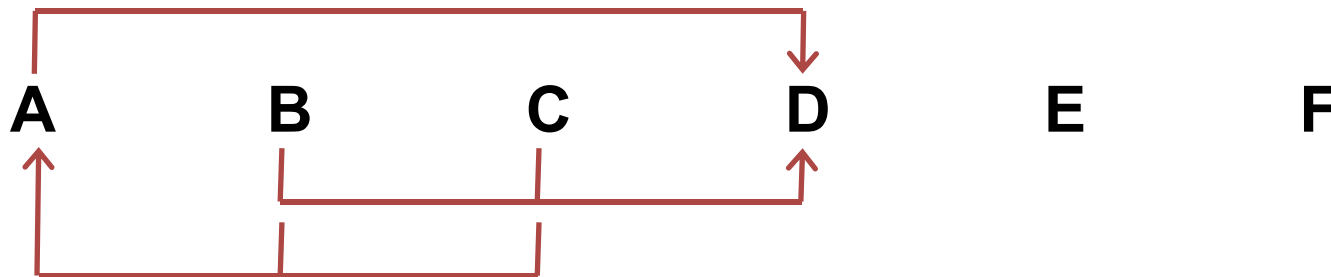
通过观察分析上图，可以得到如下结论：

- 冗余函数依赖只可能出现在分别以**D**、**E**、**F**为依赖属性的三组函数依赖中！

步骤2的计算结果 M2 如下:

$\{A \rightarrow C, A \rightarrow D, A \rightarrow E, AB \rightarrow F, BC \rightarrow D, BC \rightarrow F, BC \rightarrow A, BC \rightarrow E\}$

➤ 检查  $A \rightarrow D$  和  $BC \rightarrow D$ : 分析与之相关的三个函数依赖



观察上图易知:

由  $BC \rightarrow A$  和  $A \rightarrow D$

调用传递规则可得

$BC \rightarrow D$

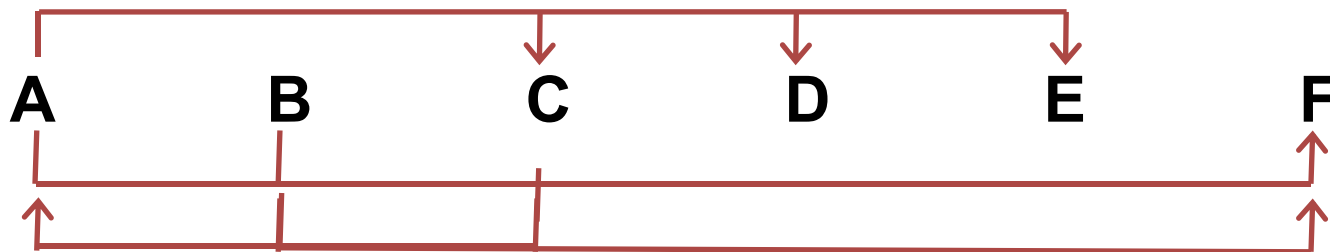
∴  $BC \rightarrow D$  是冗余的!

同理可得: 在  $A \rightarrow E$  和  $BC \rightarrow E$  中,  $BC \rightarrow E$  是冗余的!

新的函数依赖集 **M2** 如下：

**$\{ A \rightarrow C, A \rightarrow D, A \rightarrow E, AB \rightarrow F, BC \rightarrow F, BC \rightarrow A \}$**

➤ 再检查  **$AB \rightarrow F$**  和  **$BC \rightarrow F$**



➤ 删去  **$AB \rightarrow F$**  得到：  **$M3 = \{A \rightarrow C, A \rightarrow D, A \rightarrow E, BC \rightarrow F, BC \rightarrow A\}$**

判断：从 **M3** 中是否能够推导得到  **$AB \rightarrow F$** ？

在 **M3** 上计算  **$AB^+ = \{A, B, C, D, E, F\}$**

∵ 在闭包计算结果中有属性 **F**

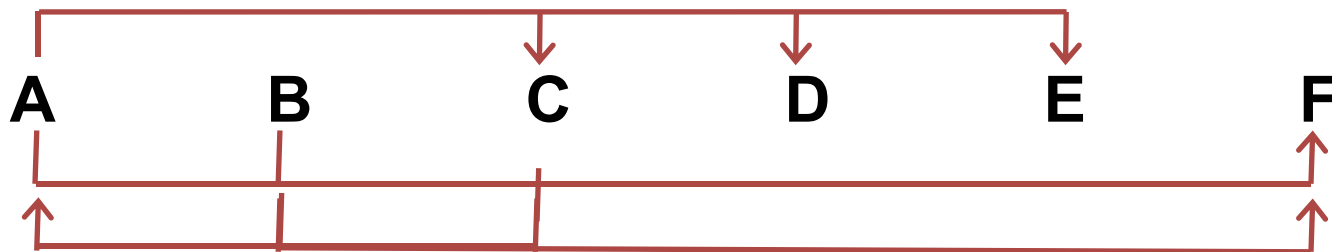
∴ 从 **M3** 中能够推导得到  **$AB \rightarrow F$**

∴ 可以用删除  **$AB \rightarrow F$**  后得到的 **M3** 来取代 **M2**

新的函数依赖集 **M2** 如下：

$\{ A \rightarrow C, A \rightarrow D, A \rightarrow E, AB \rightarrow F, BC \rightarrow F, BC \rightarrow A \}$

➤ 重新考虑  **$AB \rightarrow F$**  和  **$BC \rightarrow F$**



➤ 事实上，如果我们先考虑删去  **$BC \rightarrow F$**  得到另一个函数依赖集：

$M4 = \{ A \rightarrow C, A \rightarrow D, A \rightarrow E, AB \rightarrow F, BC \rightarrow A \}$

采用与前一页相同的推导过程，也可以证明 **M4** 和 **M2** 是等价的！

出现上述情况的根本原因，是我们可以从 **M2** 中推导得到：

$AB \rightarrow BC$       和       $BC \rightarrow AB$

➤ 在最小覆盖的计算中，既可以用前一页的 **M3**，也可以用本页的 **M4** 来作为 step 3 的计算结果！

使用 **M3** 作为 **step 3** 的计算结果如下：

$$\mathbf{M3} = \{ \mathbf{A \rightarrow C}, \mathbf{A \rightarrow D}, \mathbf{A \rightarrow E}, \mathbf{BC \rightarrow F}, \mathbf{BC \rightarrow A} \}$$

**step 4：** 合并M3中决定因素相同的函数依赖得到F的最小覆盖：

$$\mathbf{M} = \{ \mathbf{A \rightarrow CDE}, \mathbf{BC \rightarrow AF} \}$$

说明：

- ① 不论使用**M3**还是**M4**作为 **step 3** 的最终计算结果，后面的设计结果可能有所不同，但都不影响规范化设计希望达到的目标！
- ② **step 2**和**step 3**的执行顺序可以对调；
- ③ 如果先执行**step 3**（消除冗余函数依赖），再执行**step 2**（消除部分函数依赖），那么需要重新检查一下，是否产生新的冗余函数依赖；
- ④ 最后的**step 4**必不可少。

$R(A, B, C, D, E, F, G) \quad M = \{ A \rightarrow CDE, BC \rightarrow AF \}$

## 2. 使用算法找出关系R的所有候选关键字，并简要写出计算过程

- 检查最小覆盖M中的每一个函数依赖
  - 只在函数依赖的左边出现过的属性是：B
  - 只在函数依赖的右边出现过的属性是：D, E, F
  - 在函数依赖的左右两边都出现过的属性是：A, C
  - 在M中没有出现过的属性：G
- D、E、F不可能是关键字的组成部分，B和G是每一个关键字的组成部分，在FOR循环中只需要检查A和C (计算过程略)
- 事实上，我们还可以简化上述的计算过程：
  - ① 忽略D、E、F，从M中推导得到A,B,C,G四个属性之间的函数依赖有：N = { A → C, BC → A }
  - ② 按照同样的计算过程，可以得到该关系的两个关键字 {A,B,G} 和 {B,C,G}
  - ③ 经检验 {A,B,G} 和 {B,C,G} 是关系R的两个候选关键字

## $R(A, B, C, D, E, F, G)$

1. 最小函数依赖集:  $M = \{A \rightarrow CDE, BC \rightarrow AF\}$
2. 候选关键字:  $\{A, B, G\}$  和  $\{B, C, G\}$
3. 使用模式分解算法, 将关系  $R$  直接分解到  $3NF$ , 并满足分解的无损联结性和依赖保持性。

➤ 到  $3NF$  的分解结果如下:

子关系	函数依赖集	关键字
$R_1(A, C, D, E)$	$A \rightarrow CDE$	$A$
$R_2(A, B, C, F)$	$BC \rightarrow AF, A \rightarrow C$	两个关键字: $BA$ 和 $BC$
$R_3(A, B, G)$	(无)	$ABG$

注: 也可以用另一个候选关键字  $\{B, C, G\}$  组成关系  $R_3$



## R ( A, B, C, D, E, F, G )

1. 最小函数依赖集:  $M = \{ A \rightarrow CDE, BC \rightarrow AF \}$
2. 候选关键字:  $\{A,B,G\}$  和  $\{B,C,G\}$
3. 到 3NF 的分解:

子关系	函数依赖集	关键字
R1 (A, C, D, E)	$A \rightarrow CDE$	A
R2 (A, B, C, F)	$BC \rightarrow AF, A \rightarrow C$	两个关键字: BA 和 BC
R3 (A, B, G)	(无)	ABG

4. 上述分解结果是否满足BCNF? 如不满足请将其分解到BCNF。

➤ R2 不满足 BCNF , 可将其进一步分解如下:

- R21 (A, C) 函数依赖集是  $\{ A \rightarrow C \}$ , 关键字为 A
- R22 (A, B, F) 函数依赖集为空集, 关键字为 ABF

其中, R21 可以被合并到关系 R1 中去, R22 成为新的 R2

## $R(A, B, C, D, E, F, G)$

1. 最小函数依赖集:  $M = \{A \rightarrow CDE, BC \rightarrow AF\}$
2. 候选关键字:  $\{A, B, G\}$  和  $\{B, C, G\}$
3. 到 3NF 的分解:

子关系	函数依赖集	关键字
$R_1(A, C, D, E)$	$A \rightarrow CDE$	$A$
$R_2(A, B, C, F)$	$BC \rightarrow AF, A \rightarrow C$	两个关键字: $BA$ 和 $BC$
$R_3(A, B, G)$	(无)	$ABG$

4. 到BCNF的分解结果如下:

子关系	函数依赖集	关键字
$R_1(A, C, D, E)$	$A \rightarrow CDE$	$A$
$R_2(A, B, F)$	(无)	$ABF$
$R_3(A, B, G)$	(无)	$ABG$