



## 第6章 关系模式的规范化理论

---



## 6.3 函数依赖的公理系统

---

6.3.1 函数依赖的逻辑蕴涵

6.3.2 Armstrong公理系统

6.3.3 函数依赖集的等价与覆盖





## 6.3.1 函数依赖的逻辑蕴涵

例如在上述的传递函数依赖中，由 $X \rightarrow Y$ ， $Y \rightarrow Z$ ，推导出 $X \rightarrow Z$ ，这可以表示为：

$\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$  其中： $\models$ 表示逻辑蕴涵。

一般地讲，函数依赖的逻辑蕴涵定义如下：

**定义6.6（逻辑蕴涵）：** 设 $F$ 是由关系模式 $R(U)$ 满足的一个函数依赖集， $X \rightarrow Y$ 是 $R$ 的一个函数依赖，且不包含在 $F$ ，如果满足 $F$ 中所有函数依赖的任一具体关系 $r$ ，也满足 $X \rightarrow Y$ ，则称函数依赖集 $F$ 逻辑地蕴涵函数依赖 $X \rightarrow Y$ ，或称 $X \rightarrow Y$ 可从 $F$ 推出。可表示为：

$$F \models X \rightarrow Y$$

## 函数依赖集F的闭包 $F^+$

定义6.7：函数依赖集F所逻辑蕴涵的函数依赖的全体称为为F的闭包（Closure），记为 $F^+$ ，即 $F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$

例如，有关系R（X，Y，Z），它的函数依赖集 $F = \{X \rightarrow Y, Y \rightarrow Z\}$ ，则其闭包 $F^+$ 为：

$$F^+ = \left[ \begin{array}{cccccc} X \rightarrow \phi & XY \rightarrow \phi & XZ \rightarrow \phi & XYZ \rightarrow \phi & Y \rightarrow \phi & Z \rightarrow \phi \\ X \rightarrow X & XY \rightarrow X & XZ \rightarrow X & XYZ \rightarrow X & Y \rightarrow Y & Z \rightarrow Z \\ X \rightarrow Y & XY \rightarrow Y & XZ \rightarrow Y & XYZ \rightarrow Y & Y \rightarrow Z & \phi \rightarrow \phi \\ X \rightarrow Z & XY \rightarrow Z & XZ \rightarrow Z & XYZ \rightarrow Z & Y \rightarrow YZ & \\ X \rightarrow XY & XY \rightarrow XY & XZ \rightarrow XY & XYZ \rightarrow XY & YZ \rightarrow \phi & \\ X \rightarrow XZ & XY \rightarrow XZ & XZ \rightarrow XZ & XYZ \rightarrow XZ & YZ \rightarrow Y & \\ X \rightarrow YZ & XY \rightarrow YZ & XZ \rightarrow YZ & XYZ \rightarrow YZ & YZ \rightarrow Z & \\ X \rightarrow XYZ & XY \rightarrow XYZ & XZ \rightarrow XYZ & XYZ \rightarrow XYZ & YZ \rightarrow YZ & \end{array} \right]$$





## 6.3.2 Armstrong公理系统

### 1) 独立推理规则

即下面给出的Armstrong公理的三条推理规则是彼此独立的。

(1) A1: 自反律(Reflexivity)

如果 $Y \subseteq X$ , 则 $X \rightarrow Y$ 成立, 这是一个平凡函数依赖。

根据A1可以推出 $X \rightarrow \Phi$ 、 $U \rightarrow X$ 等平凡函数依赖 (因为 $\Phi \subseteq X \subseteq U$ )。

(2) A2: 增广律(Augmentation)

如果 $X \rightarrow Y$ , 且 $Z \subseteq W$ , 则 $XW \rightarrow YZ$ 成立。

根据A2可以推出 $XW \rightarrow Y$ 、 $XZ \rightarrow YZ$ 或 $XW \rightarrow YW$ 、 $X \rightarrow XY$ 、 $XY \rightarrow X$ 等。

(3) A3: 传递律(Transitivity)

如果 $X \rightarrow Y$ 且 $Y \rightarrow Z$ , 则 $X \rightarrow Z$ 成立。



## 2) 其他推理规则

推论1: 合并规则(The Union Rule)

$$\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$$

推论2: 分解规则(The Decomposition Rule)

如果  $X \rightarrow Y$ ,  $Z \subseteq Y$ , 则  $X \rightarrow Z$  成立

推论3: 伪传递规则(The Pseudo Transitivity Rule)

$$\{X \rightarrow Y, WY \rightarrow Z\} \models XW \rightarrow Z$$

证: (1)  $X \rightarrow Y \models X \rightarrow XY$  (A2增广律)

$X \rightarrow Z \models XY \rightarrow YZ$  (A2增广律)

由上可得  $X \rightarrow YZ$  (A3传递律)

(2)  $Z \subseteq Y \models Y \rightarrow Z$  (A1自反律)

$X \rightarrow Y$  (给定条件)

由上可得  $X \rightarrow Z$  (A3传递律)

(3)  $X \rightarrow Y \models WX \rightarrow WY$  (A2增广律)

$WY \rightarrow Z$  (给定条件)

由上可得  $XW \rightarrow Z$  (A3传递律)



## 一个重要定理

例 6.2：设有关系模式  $R(A,B,C,D,E)$  及其上的函数依赖集  $F=\{AB\rightarrow CD, A\rightarrow B, D\rightarrow E\}$ ，求证  $F$  必蕴涵  $A\rightarrow E$ 。

证明：  $\because A\rightarrow B$  (给定条件)  
  $\therefore A\rightarrow AB$  (A2增广律)  
  $\because AB\rightarrow CD$  (给定条件)  
  $\therefore A\rightarrow CD$  (A3传递律)  
  $\therefore A\rightarrow C, A\rightarrow D$  (分解规则)  
  $\because D\rightarrow E$  (给定条件)  
  $\therefore A\rightarrow E$  (A3传递律)  
 证毕。

定理6.1：若  $A_i (i=1, 2, \dots, n)$  是关系模式  $R$  的属性，  
 则  $X\rightarrow(A_1, A_2, \dots, A_n)$  成立的充分必要条件是  $X\rightarrow A_i$  均成立。



## 属性集闭包

定义 6.8 (属性集闭包) : 设有关系模式  $R(U), U = \{A_1, A_2, \dots, A_n\}$ ,  $X$  是  $U$  的子集,  $F$  是  $U$  上的一个函数依赖集, 则属性集  $X$  关于函数依赖集  $F$  的闭包  $X_F^+$  定义为:

$$X_F^+ = \{A_i \mid A_i \in U, \text{ 且 } X \rightarrow A_i \text{ 可用阿氏公理从 } F \text{ 推出}\}$$

例: 设关系模式  $R(A, B, C)$  的函数依赖集为  $F = \{A \rightarrow B, B \rightarrow C\}$ , 分别求  $A$ 、 $B$ 、 $C$  的闭包。

解: 若  $X = A$ ,

$$\begin{aligned} &\because A \rightarrow B, B \rightarrow C && \text{(给定条件)} \\ &\therefore A \rightarrow C && \text{(A2传递律)} \\ &\because A \rightarrow A && \text{(A1自反律)} \\ &\therefore A_F^+ = \{A, B, C\} && \text{(据定义)} \end{aligned}$$

若  $X = B$

$$\begin{aligned} &\because B \rightarrow B && \text{(A1自反律)} \\ &\quad B \rightarrow C && \text{(给定条件)} \\ &\therefore B_F^+ = \{B, C\} && \text{(据定义)} \end{aligned}$$

若  $X = C$ ,

$$\begin{aligned} &\quad C \rightarrow C && \text{(自反律)} \\ &\therefore C_F^+ = \{C\} && \text{(据定义)} \end{aligned}$$





**定理6.2:** 设 $F$ 是关系模式 $R(U)$ 上的函数依赖集, $U$ 是属性全集,  $X, Y \subseteq U$ , 则函数依赖 $X \rightarrow Y$ 是用阿氏公理从 $F$ 推出的, 充分必要条件是  $Y \subseteq X_F^+$ ;

反之, 能用阿氏公理从 $F$ 推出的所有 $X \rightarrow Y$ 的 $Y$ 都在 $X_F^+$ 中。

这个定理告诉我们, 只要 $Y \subseteq X_F^+$ , 则必有 $X \rightarrow Y$ 。于是, 一个函数依赖 $X \rightarrow Y$ 能否用阿氏公理从 $F$ 推出的问题, 就变成判断 $Y$ 是否为 $X_F^+$ 子集的问题。

下面介绍一下计算  $X_F^+$  的算法。



## 属性集的闭包计算

算法6.1：求属性集 $X(X \subseteq U)$ 关于 $U$ 上的函数依赖集 $F$ 的闭包 $X_F^+$ 。

输入：属性全集 $U$ ,  $U$ 上的函数依赖集 $F$ , 以及属性集 $X \subseteq U$ 。

输出： $X$ 关于 $F$ 的闭包  $X_F^+$ 。

方法：根据下列步骤计算一系列属性集合 $X^{(0)}, X^{(1)}, \dots$

(1) 令 $X^{(0)}=X, i=0$ ;

(2) 求属性集  $B = \{A \mid (\forall V)(\forall W) V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W\}$

/\*在 $F$ 中寻找满足条件 $V \subseteq X^{(i)}$ 的所有函数依赖 $V \rightarrow W$ ,并记属性 $W$ 的并集为 $B^*$ /\*

(3)  $X^{(i+1)} = X^{(i)} \cup B$

(4) 判断 $X^{(i+1)} = X^{(i)}$ 吗?

(4) 若 $X^{(i+1)} \neq X^{(i)}$ , 则用 $i+1$ 取代 $i$ , 返回(2);

(5) 若 $X^{(i+1)} = X^{(i)}$ , 则  $X_F^+ = X^{(i)}$ , 结束。

## 算法6.1的求解过程

例：设  $F = \{ AH \rightarrow C, C \rightarrow A, EH \rightarrow C, CH \rightarrow D, D \rightarrow EG, CG \rightarrow DH, CE \rightarrow AG, ACD \rightarrow H \}$ ，令  $X = DH$ ，求  $X_F^+$ 。

解：①  $X^{(0)} = X = DH$

②在F中找所有满足条件  $V \subseteq X^{(0)} = DH$  的函数依赖  $V \rightarrow W$ ，结果只有  $D \rightarrow EG$ ，则  $B = EG$ ，于是  $X^{(1)} = X^{(0)} \cup B = DEGH$ 。

③ 判断是否  $X^{(i+1)} = X^{(i)}$ ，显然  $X^{(1)} \neq X^{(0)}$ 。

④在F中找所有满足条件  $V \subseteq X^{(1)} = DEGH$  的函数依赖  $V \rightarrow W$ ，结果为  $EH \rightarrow C$ ，于是  $B = C$ ，则  $X^{(2)} = X^{(1)} \cup B = CDEGH$ 。

⑤ 判断是否  $X^{(i+1)} = X^{(i)}$ ，显然  $X^{(2)} \neq X^{(1)}$ 。

⑥在F中找所有满足条件  $V \subseteq X^{(2)} = CDEGH$  的函数依赖  $V \rightarrow W$ ，结果为  $C \rightarrow A, CH \rightarrow D, CG \rightarrow DH, CE \rightarrow AG$ ，则  $B = ADGH$ ，于是  $X^{(3)} = X^{(2)} \cup B = CDEGH \cup B = ACDEGH$ 。

⑦判断是否  $X^{(i+1)} = X^{(i)}$ ，这时虽然  $X^{(3)} \neq X^{(2)}$ 。但  $X^{(3)}$  已经包含了全部属性，所以不必再继续计算下去。

最后，  $X_F^+ (DH)^+ = \{ ACDEGH \}$ 。





# 属性集闭包计算结束判断方法

在判断计算何时结束时，可用下面四种方法：

- (1)  $X^{(i+1)} = X^{(i)}$ 。
- (2)  $X^{(i+1)}$ 已包含了全部属性。
- (3) 在F中再也找不到函数依赖的右部属性是  $X^{(i)}$ 中未出现过的属性。
- (4) 在F中再也找不到满足条件  $V \subseteq X^{(i)}$  的函数依赖  $V \rightarrow W$ 。



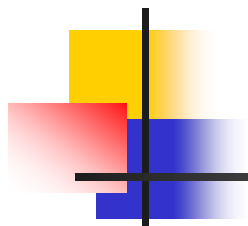
### 6.3.3 函数依赖集的等价和覆盖

定义6.9（函数依赖集的等价、覆盖）：设F和G是关系R（U）上的两个依赖集，若 $F^+=G^+$ ，则称F与G等价，记为 $F=G$ 。也可以称F覆盖G，或G覆盖F；也可说F与G相互覆盖。

检查两个函数依赖集F和G是否等价的方法是：

- 第一步：检查F中的每个函数依赖是否属于 $G^+$ ，若全部满足，则 $F \subseteq G^+$ 。如若 $X \rightarrow Y \in F$ ，则计算 $X_G^+$ ，如果 $Y \subseteq X_G^+$ ，则 $X \rightarrow Y \in G^+$ ；
- 第二步：同第一步，检查是否 $G \subseteq F^+$ ；
- 第三步：如果 $F \subseteq G^+$ ，且 $G \subseteq F^+$ ，则F与G等价。

由此可见,F和G等价的充分必要条件是： $F \subseteq G^+$ ，且 $G \subseteq F^+$ 。



引理6.1：设 $G$ 是一个函数依赖集，且其中所有依赖的右部都只有一个属性，则 $G$ 覆盖任一左部与 $G$ （左部）相同的函数依赖集。

证明：构造 $G = \{X \rightarrow A \mid X \rightarrow Y \in F \text{ 且 } A \in Y\}$

由 $A \in Y$ ， $X \rightarrow Y \in F$ 根据分解规则导出，从而等到 $G \subseteq F^+$ 。

反之，如果 $Y = A_1 A_2 \dots A_n$ ，而且 $X \rightarrow A_1$ ， $X \rightarrow A_2$ ，... $X \rightarrow A_n$ 在 $G$ 中可根据合并律等到 $F \subseteq G^+$ 。

由此可见， $F$ 与 $G$ 等价，即 $F$ 被 $G$ 覆盖。

一个函数依赖集 $F$ 可能有若干个与其等价的函数依赖集，我们可以从中选择一个较好以便应用的函数依赖集。标准至少是：

- 所有函数依赖均独立，即该函数依赖集中不存在这样的函数依赖，它可由这个集合中的别的函数依赖推导出来。
- 表示最简单，即每个函数依赖的右部为单个属性，左部最简单。



## 最小函数依赖集

**定义6.10（最小函数依赖集）**:函数依赖集F如果满足下列条件,则称F为最小函数覆盖,记为 $F_{\min}$ :

- (1) F中每一个函数依赖的右部都是单个属性。
- (2) 对F中任一函数依赖 $X \rightarrow A$ ,  $F - \{X \rightarrow A\}$ 都不与F等价。
- (3) 对于F中的任一函数依赖 $X \rightarrow A$ ,  $\{F - \{X \rightarrow A\}\} \cup \{Z \rightarrow A\}$ 都不与F等价, 其中Z为X的任一真子集。

求函数依赖集F的最小覆盖的方法是:

- (1) 检查F中的每个函数依赖 $X \rightarrow A$ ,若 $A = A_1, A_2, \dots, A_k$ ,则根据分解规则, 用 $X \rightarrow A_i (i=1, 2, \dots, k)$ 取代 $X \rightarrow A$ 。
- (2) 检查F中的每个函数依赖 $X \rightarrow A$ ,令 $G = F - \{X \rightarrow A\}$ ,  
若有  $A \in X_G^+$  则从F中去掉此函数依赖。
- (3) 检查F中各函数依赖 $X \rightarrow A$ , 设 $X = B_1, B_2, \dots, B_m$ ,检查 $B_i$ ,  
当 $A \in (X - B_i)_F^+$  时, 即以 $X - B_i$ 替换 $X$ 。



## 最小覆盖的求解事例

例6.5: 求下列函数依赖集的最小覆盖:

$F = \{AH \rightarrow C, C \rightarrow A, CH \rightarrow D, C \rightarrow EG, EH \rightarrow C, CG \rightarrow DH, CE \rightarrow AG, ACD \rightarrow H\}$ 。

解: (1) 用分解规则将F中的所有依赖的右部变成单个属性, 可以得到以下11个函数依赖:

$AH \rightarrow C, C \rightarrow A, CH \rightarrow D, ACD \rightarrow H$  (给定)

$C \rightarrow E, C \rightarrow G$  (由  $C \rightarrow EG$  分解得到)

$EH \rightarrow C$  (给定)

$CG \rightarrow H, CG \rightarrow D$  (由  $CG \rightarrow DH$  分解得到)

$CE \rightarrow A, CE \rightarrow G$  (由  $CE \rightarrow AG$  分解得到)

(2) 根据阿氏公理去掉F中的冗余依赖

由于从  $C \rightarrow A$  可推出  $CE \rightarrow A$ , 从  $C \rightarrow A, CG \rightarrow D, ACD \rightarrow H$  推出  $CG \rightarrow H$ , 因此  $CE \rightarrow A$  和  $CG \rightarrow H$  是冗余, 可从F删除。

(3) 用所含属性较少的依赖代替所含属性较多的依赖。

由于  $C \rightarrow A, ACD \rightarrow H$  中A是冗余属性, 因此, 可用  $CD \rightarrow H$  代替  $ACD \rightarrow H$ , 故删除  $ACD \rightarrow H$ 。

最后得到F的最小覆盖为:

$F = \{AH \rightarrow C, C \rightarrow A, CH \rightarrow D, CD \rightarrow H, C \rightarrow E, C \rightarrow G, EH \rightarrow C, CG \rightarrow D, CE \rightarrow G\}$





## 6. 候选关键字的求解方法

- 给定一个关系模式 $R(U, F)$ ,  $U = \{A_1, A_2, \dots, A_n\}$ ,  $F$ 是 $R$ 的函数依赖集, 那么, 可以见属性分为如下四类:
  - $L$ : 仅出现在函数依赖集 $F$ 左部的属性
  - $R$ : 仅出现在函数依赖集 $F$ 右部的属性
  - $LR$ : 在函数依赖集 $F$ 左右部都出现的属性
  - $NLR$ : 在函数依赖集  $F$ 左右部都未出现的属性



## 6. 候选关键字的求解方法

---

- 求候选码的规则：

(1) 如果有属性不在函数依赖集中出现，那么它必须包含在候选码中；

(2) 如果有属性不在函数依赖集中任何函数依赖的右边出现，那么它必须包含在候选码中；

(3) 如果有属性只在函数依赖集的左边出现，则该属性一定包含在候选码中。

(4) 如果有属性或属性组能唯一标识元组，则它就是候选码；

## 6. 候选关键字的求解方法

- 设有关系模式  $R(B, C, M, T, A, G)$ ，根据语义有如下函数依赖集， $F = \{B \rightarrow C, (M, T) \rightarrow B, (M, C) \rightarrow T, (M, A) \rightarrow T, (A, B) \rightarrow G\}$ ，则求关系的候选码？
- 根据第一条，发现所有的属性都在函数依赖集中出现了。所以不使用第一条规则  
根据第二条，只有  $M, A$  没有在函数依赖的右边出现，则  $A, M$  一定是候选码的一部分。  
根据第三条，只有  $M, A$  只在函数依赖的左边出现，因此可以判定  $AM$  一定是候选码的一部分。  
根据第四条，只有  $(M, A)$  这个属性组可以唯一确定属性集中的每一个属性，因此， $(M, A)$  一定是该关系模式的候选码。  
在根据第四条，没有发现还有别的属性或属性组能够确定属性集中的全部属性，因此可以确定，该关系模式只有一个候选码，该候选码就是  $(M, A)$ ，单独的  $A$  或  $M$  都不是候选码，候选码是  $M$  和  $A$  的组合。



## 举例

---

【例2.18】 设关系模式 $R(U, F)$ ，其中  
 $U = \{A, B, C, D\}$ ， $F = \{A \rightarrow C, C \rightarrow B,$   
 $AD \rightarrow B\}$ 。求 $R$ 的候选码。



## 举例

---

- 解:根据结论(1)可以求得R的候选码为AD, 而且AD是R唯一的候选码。分析如下:
  - (1)检查 F发现, A, D只出现在函数依赖的左部, 所以为L类属性, 而F包含了全属性, 即不存在NLR类的属性。
  - (2)根据求属性闭包的算法, F中 $A \rightarrow C$ ,  $AD \rightarrow B$ 可以求得  $(AD)_F^+ = ABCD = U$ , 而在 AD中不存在一个真子集能决定全属性, 故AD为R的候选码。



## 举例

---

【例2.19】 设关系模式 $R(U, F)$ ，其中 $U = \{H, I, J, K, L, M\}$ ， $F = \{H \rightarrow I, K \rightarrow I, LM \rightarrow K, I \rightarrow K, KH \rightarrow M\}$ 。求 $R$ 的候选码。



## 举例

---

- 解:根据结论(1)、(2)、(3)和(4) 可以求得R的候选码为HLJ, 而且HLJ是R唯一的候选码。分析如下:
  - (1)检查 F发现, H, L只出现在函数依赖的左部, 所以为L类属性, J为NLR类的属性。
  - (2)根据求属性闭包的算法, F中 $H \rightarrow I$ ,  $I \rightarrow K$ ,  $KH \rightarrow M$ 可以求得  $(HLJ)_F^+ = HIJKLM = U$ , 而在HLJ中不存在一个真子集能决定全属性, 故HLJ为R的候选码。



## 2) 结果为3NF的依赖保持分解

算法6-4: 结果为3NF的依赖保持分解算法

输入: 关系模式R和函数依赖集F

输出: 结果为3NF的一个依赖保持分解

步骤:

(1)如果R中有某些属性与F的最小覆盖F'中的每个依赖的左边和右边都无关, 原则上可由这些属性构成一个关系模式, 并从R中将它们消除;

(2)如果F'中有一个依赖涉及到R的所有属性, 则输出R;

(3)否则, 输出一个分解 $\rho$ , 它由模式XA组成, 其中 $X \rightarrow A \in F$ 。但当 $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ 均属于F'时, 则用模式 $XA_1A_2\dots A_n$ 代替 $XA_i (i=1,2, \dots, n)$ 。

例6-15: 对于上例,

$F' = \{ \{ C \rightarrow T, CS \rightarrow G, HT \rightarrow R, HR \rightarrow C, CH \rightarrow R, HS \rightarrow R \} , KEY = HS \}$

所以  $\rho = \{ CT, CSG, HRT, CHR, HSR \}$





### 3) 结果为3NF且具有依赖保持和连接不失真的分解

定理6-11: 设 $\delta$ 是由结果为3NF的依赖保持分解算法得到的3NF分解,  $X$ 为 $R$ 的一个候选关键字, 则 $\tau = \delta \cup \{X\}$ 是 $R$ 的一个分解, 且 $\tau$ 中的所有关系模式均满足3NF, 同时, 既具有连接不失真性, 又具有依赖保持性。

例: 已知 $R(C, T, H, R, S, G)$ ,

$F' = \{C \rightarrow T, HR \rightarrow C, CS \rightarrow G, HS \rightarrow R, HT \rightarrow R\}$ ,  $KEY = HS$ ,

则  $\tau = \{CT, CSG, HRT, CHR, HSR, HS\}$

但 $HS \subseteq HSR$ , 故

$\tau = \{CT, CSG, HRT, CHR, HSR\}$





## 6.4 关系模式的分解 及其问题

---

6.4.1 什么叫模式分解

6.4.2 分解的无损连接性

6.4.3 保持函数依赖性





## 6.4.1 什么叫模式分解

定义6.11（模式分解）：关系模式 $R(U, F)$ 的一个分解 $\rho$ 是若干个关系模式的一个集合： $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)\}$

式中：(1)  $U = \bigcup_{i=1}^n U_i$ 。

(2) 对每个 $i, j$  ( $1 \leq i, j \leq n$ ) 有  $U_i \not\subseteq U_j$ 。

(3)  $F_i (i=1, 2, \dots, n)$  是 $F$ 在 $U_i$ 上的投影，即

$$F_i = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \in U_i\}$$

例6.6：设在模式 $R(U, F)$ 中

$U = \{SNO, SNAME, DNAME, DADDR\}$

$F = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR\}$

如果对 $R$ 作如下分解（方法1）：

$\rho = \{R_1(\{SNO, SNAME\}, \{SNO \rightarrow SNAME\}),$   
 $R_2(\{DNAME, DADDR\}, \{DNAME \rightarrow DADDR\})\}$

# (1) 连接不失真问题

(a) 原关系R

SNO	SNAME	DNAME	DADDR
0001	张华	计算机	D1
0002	李明	信息管理	D2
0003	刘强	计算机	D1

(b) 方法1: 关系 $R_1$

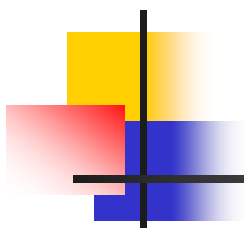
SNO	SNAME
0001	张华
0002	李明
0003	刘强

(c) 方法1: 关系 $R_2$

DNAME	DADDR
计算机	D1
信息管理	D2

(d) 方法1: 关系 $R_1 \times R_2$

SNO	SNAME	DNAME	DADDR
0001	张华	计算机	D1
0001	张华	信息管理	D2
0002	李明	计算机	D1
0002	李明	信息管理	D2
0003	刘强	计算机	D1
0003	刘强	信息管理	D2



方法2: 假设按下列方法对R进行分解

$\rho = \{R_1(\{SNO, SNAME, DNAME\}, \{SNO \rightarrow SNAME, SNO \rightarrow DNAME\}),$   
 $R_2(\{DNAME, DADDR\}, \{DNAME \rightarrow DADDR\})\}$   
(a) 原关系R

SNO	SNAME	DNAME	DADDR
0001	张华	计算机	D1
0002	李明	信息管理	D2
0003	刘强	计算机	D1

(g) 方法2:  $R_1 \bowtie R_2$

(e) 方法2: 关系 $R_1$

SNO	SNAME	DNAME
0001	张华	计算机
0002	李明	信息管理
0003	刘强	计算机

(f) 方法2: 关系 $R_2$

DNAME	DADDR
计算机	D1
信息管理	D2

SNO	SNAME	DNAME	DADDR
0001	张华	计算机	D1
0002	李明	信息管理	D2
0003	刘强	计算机	D1



## (2) 依赖保持问题

一个关系模式经分解后，其函数依赖集F也随之被分解，则分解后的依赖集 $F_i$ 并集是否能保持原有的函数依赖关系？即  $(\bigcup_{i=1}^n F_i)^+ = F^+$ ？

若出现  $F^+ \supset (\bigcup_{i=1}^n F_i)^+$ ，说明分解后有些函数依赖被丢失了。

上例方法1:

$F = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR\}$

$F_1 \cup F_2 = \{SNO \rightarrow SNAME, DNAME \rightarrow DADDR\}$

$F^+ = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR, SNO \rightarrow DADDR\}$

$(F_1 \cup F_2)^+ = \{SNO \rightarrow SNAME, DNAME \rightarrow DADDR\}$

上例方法2:

$F = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR\}$

$F_1 \cup F_2 = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR\}$

$F^+ = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR, SNO \rightarrow DADDR\}$

$(F_1 \cup F_2)^+ = \{SNO \rightarrow SNAME, SNO \rightarrow DNAME, DNAME \rightarrow DADDR, SNO \rightarrow DADDR\}$



## 6.4.2 分解的无损连接性

### 1) 无损连接分解的定义

定义6.12（无损连接分解，即连接不失真分解）：设关系模式R（U，F）上的一个分解为  $\rho = \{ R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k) \}$ ，F是R（U，F）上的一个函数依赖集。如果对R中满足F的任一关系r都有

$$r = \prod_{R_1}(r) \bowtie \prod_{R_2}(r) \bowtie \dots \bowtie \prod_{R_k}(r)$$

则称这个分解  $\rho$  相对于F的是连接不失真分解或称无损连接分解。

$$m_{\rho}(r) = \prod_{R_1}(r) \bowtie \prod_{R_2}(r) \bowtie \dots \bowtie \prod_{R_k}(r) = \bigwedge_{i=1}^k \prod_{R_i}(r)$$

对于关系模式R关于F的无损连接条件是：

任何满足F的关系r有  $r = m_{\rho}(r)$ 。



## r 和 $m_\rho(r)$ 之间的联系

定理6.4: 设R是一关系模式,  $\rho = \{ R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k) \}$  是关系模式R的一个分解, r是R的任一关系, ( $1 \leq i \leq k$ ), 那么有:

①  $r \subseteq m_\rho(r)$  ;

② 如果  $s = m_\rho(r)$ , 则  $\Pi_{R_i}(s) = r_i$ , 或  $\Pi_{U_i}(s) = \Pi_{U_i}(r)$

③  $m_\rho m_\rho(r) = m_\rho(r)$



## 定理6.4证明

证明：① 设任意一个元组  $t \in r, t_i = t[U_i] (i=1,2,\dots,k)$ ; 则  $t_i \in R_i$ 。根据自然连接定义, 可知  $t$  在  $\bowtie \prod_{i=1}^k R_i (r)$  中, 即  $t \in m_p(r)$ , 所以  $r \subseteq m_p(r)$ 。

该定理①说明, 一个关系模式经分解再连接恢复所得的新关系  $m_p(r)$  的元组一般比原关系的元组要多, 而且  $m_p(r)$  一定包括原关系的元组。

只有当  $r = m_p(r)$  时, 分解才是连接不失真分解。

② 由定理6-5①可知  $r \subseteq m_p(r)$ , 可得到  $\prod_{R_i} (r) \subseteq \prod_{R_i} (m_p(r)) = \prod_{R_i} (s)$ , 即  $r_i \subseteq \prod_{R_i} (s)$  (因为  $s = m_p(r)$ ) (也就是两边同时在  $U_i$  上投影, 得  $\prod_{U_i} (r) \subseteq \prod_{U_i} (s)$ )。

为了证明  $\prod_{R_i} (s) \subseteq r_i$ 。假设  $t_i \in \prod_{R_i} (s)$ , 则  $s$  中必存在满足  $t[R_i] = t_i$  的元组  $t$ 。

由于  $t \in s$ , 对每个  $j$ , 在  $r_j$  中必存在元组  $u_j$  满足  $t[R_j] = u_j (1 \leq j \leq k)$ , 即  $u_j \in \prod_{R_j} (r) = r_j$ 。

于是对那个特定的  $i$ , 亦有  $t[R_i] = u_i$ , 即  $t[R_i] \in r_i$ 。但  $t[R_i] = t_i$ , 所以  $t_i \in r_i$ , 从而得到  $\prod_{R_i} (s) \subseteq r_i$  (即  $\prod_{U_i} (s) \subseteq \prod_{U_i} (r)$ )。

由  $r_i \subseteq \prod_{R_i} (s)$  和  $\prod_{R_i} (s) \subseteq r_i$  可得  $\prod_{R_i} (s) = r_i$  (即  $\prod_{U_i} (s) = \prod_{U_i} (r)$ )。

③ 由定理6-5①可知  $\prod_{R_i} (s) = r_i (i=1,2,\dots,k)$ , 于是有  $\bowtie \prod_{i=1}^k R_i (s) = \bowtie \prod_{i=1}^k r_i$ 。

此式左式 =  $m_p(s) = m_p m_p(r)$  (由②得), 右式 =  $\bowtie \prod_{i=1}^k R_i (r) = m_p(r)$ ,

因此得:  $m_p m_p(r) = m_p(r)$

该定理③说明, 关系模式只有在第一次分解的连接恢复后有可能丢失信息, 此后的多次分解恢复均能使分解不失真

## 2) 无损连接的检验

方法1: 采用检验表格构造法

算法6.2: 连接不失真检验

方法 1:

(1)构造一个 $n$ 列 $k$ 行表, 每一行对应于一个模式 $R_i$  ( $1 \leq i \leq k$ ), 每一列对应于一个属性 $A_j$  ( $1 \leq j \leq n$ ), 如下表所示。

	$A_1$	$A_2$	...	$A_n$
$R_1$				
$R_2$				
...				
$R_k$				

(2) 初始表 (填表): 若 $A_j \in R_i$ , 则第 $i$ 行第 $j$ 列上填入 $a_j$ , 否则填入 $b_{ij}$ 。

(3) 修改表: 反复检查 $F$ 中的每一个函数依赖 $X \rightarrow Y$ , 按下方法修改表格中的元素: 取 $F$ 中的函数依赖 $X \rightarrow Y$ , 检查 $Y$ 中的属性所对应的列, 找出 $X$ 相等的那些行, 将这些 $X$ 的符号相同的行中的 $Y$ 的属性所对应的符号改成一致。即如果其中有 $a_j$ , 则将 $b_{ij}$ 改为 $a_j$ ; 若无 $a_j$ , 则将它们全改为 $b_{ij}$ 。一般取 $i$ 是为其中的最小行号值。

(4) 如发现某一行变成 $a_1, a_2, \dots, a_k$ , 则此分解 $\rho$ 具有连接不失真性。

# 事例说明

例：设有 $R(U, F)$ ，其中： $U = (A, B, C, D, E)$ ，

$F = (A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow C, CE \rightarrow A)$ ， $R$ 的一个分解为：

$\rho = \{R_1(AD), R_2(AB), R_3(BE), R_4(CDE), R_5(AE)\}$ 是否无损分解？

- 根据算法6.2中(1)和(2)构造初始表，如表(a)所示。
- 根据 $A \rightarrow C$ ，对表(a)进行处理，将 $b_{13}$ 、 $b_{23}$ 、 $b_{53}$ 改成同一符号 $b_{13}$ ，即 $b_{13} = b_{23} = b_{53}$ 。再根据 $B \rightarrow C$ ，将 $b_{33}$ 、 $b_{13}$  ( $R_2$ 中)改成同一符号 $b_{13}$ 。修改后如表(b)所示。
- 考虑 $C \rightarrow D$ ，根据上述修改原则，将 $D$ 所在的第4列的 $b_{24}$ 、 $b_{34}$ 、 $b_{54}$ 均修改成 $a_4$ ，其结果如表(c)所示。(因为 $A \rightarrow C, B \rightarrow C$ )
- 再考虑 $DE \rightarrow C$ ，根据修改原则，将 $C$ 所在的第3列第3、4、5行的 $b_{13}$ 、 $a_3$ 、 $b_{13}$ 均修改成 $a_3$ ，其结果如表(d)所示。(因为 $B \rightarrow C, A \rightarrow C, C \rightarrow D$ )
- 再考虑 $CE \rightarrow A$ ，根据修改原则，将 $A$ 所在的第1列第3、4、5行的 $b_{31}$  (由 $B \rightarrow C$ 推出)、 $b_{41}$  (由 $A \rightarrow C$ 推出)、 $a_1$ 均修改成 $a_1$ ，其结果如表(e)所示。

表(a)初始表 表(b) 表(c) 表(d) 表(e)

			A	A	A	B	B	A	B	C	B	B	C	D	D	C	D	E	D	D	E	E	E
$R_1$	1	1	$R_1: AD: AD$	$b_{11}$	$b_{12}$	$b_{13}$	$b_{14}$	$b_{15}$	$b_{16}$	$b_{17}$	$b_{18}$	$b_{19}$	$b_{110}$	$b_{111}$	$b_{112}$	$b_{113}$	$b_{114}$	$b_{115}$	$b_{116}$	$b_{117}$	$b_{118}$	$b_{119}$	$b_{120}$
$R_2$	2	2	$R_2: AB: AB$	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$	$a_{26}$	$a_{27}$	$a_{28}$	$a_{29}$	$a_{210}$	$a_{211}$	$a_{212}$	$a_{213}$	$a_{214}$	$a_{215}$	$a_{216}$	$a_{217}$	$a_{218}$	$a_{219}$	$a_{220}$
$R_3$	3	3	$R_3: BE: BE$	$b_{31}$	$b_{32}$	$b_{33}$	$b_{34}$	$b_{35}$	$b_{36}$	$b_{37}$	$b_{38}$	$b_{39}$	$b_{310}$	$b_{311}$	$b_{312}$	$b_{313}$	$b_{314}$	$b_{315}$	$b_{316}$	$b_{317}$	$b_{318}$	$b_{319}$	$b_{320}$
$R_4$	4	4	$R_4: CDE: CDE$	$b_{41}$	$b_{42}$	$b_{43}$	$b_{44}$	$b_{45}$	$b_{46}$	$b_{47}$	$b_{48}$	$b_{49}$	$b_{410}$	$b_{411}$	$b_{412}$	$b_{413}$	$b_{414}$	$b_{415}$	$b_{416}$	$b_{417}$	$b_{418}$	$b_{419}$	$b_{420}$
$R_5$	5	5	$R_5: AE: AE$	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55}$	$a_{56}$	$a_{57}$	$a_{58}$	$a_{59}$	$a_{510}$	$a_{511}$	$a_{512}$	$a_{513}$	$a_{514}$	$a_{515}$	$a_{516}$	$a_{517}$	$a_{518}$	$a_{519}$	$a_{520}$



## 简单的检验方法

### 方法2:

定理6.5: 设 $\rho = \{R_1, R_2\}$  是关系模式 $R$ 的一个分解,  $F$ 是 $R$ 的一个函数依赖集, 则对于 $F$ ,  $\rho$ 具有连接不失真性的充分必要条件是 $R_1 \cap R_2 \rightarrow R_1 - R_2 \in F^+$ , 或 $R_1 \cap R_2 \rightarrow R_2 - R_1 \in F^+$ 。

例6.8: 设有关系模式 $R(\{S\#, SN, C\#, G\}, \{S\# \rightarrow SN, (S\#, C\#) \rightarrow G\})$ 的一个分解为:

$\rho = \{R_1(\{S\#, SN\}, \{S\# \rightarrow SN\}), R_2(\{S\#, C\#, G\}, \{(S\#, C\#) \rightarrow G\})\}$

因为 $R_1 \cap R_2 = S\#$ ,  $R_1 - R_2 = SN$ , 故 $R_1 \cap R_2 \rightarrow R_1 - R_2$ , 且 $S\# \rightarrow SN$ 属于 $F$ , 所以该分解具有连接不失真性。

定理6-8和例6-9告诉我们一个事实: 如果两个关系模式间的公共属性集至少包含其中一个关系模式的关键字, 则此分解必定具有连接不失真性。





## 6.4.3 函数依赖保持性

定义6.13：设有关系模式R,F是R上的函数依赖集，Z是R上的一个属性集合，则称Z所涉及到的 $F^+$ 中的所有函数依赖为F在Z上的投影，记为 $\Pi_Z(F)$ 。

该定义实质上是，当 $X \rightarrow Y \in F^+$ 时，若 $XY \subseteq Z$ ，则有 $\Pi_Z(F)$ ，

可以定义为： $\Pi_Z(F) = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq Z\}$

定义 6-17：设关系模式R的一个分解为 $\rho = \{ \{ R_1, F_1 \}, \{ R_2, F_2 \}, \dots, \{ R_k, F_k \} \}$ ,F是R上的依赖集，如果对于所有的 $i=1,2, \dots, k$ ， $\Pi_{R_i}(F)$ 中的全部函数依赖的并集逻辑地蕴涵F中的全部依赖，则称分解 $\rho$ 具有依赖保持性。

$$F \subseteq \left( \bigcup_{i=1}^k \Pi_{R_i}(F) \right)^+$$



➤判断两个函数依赖集是否等价的方法也可以用来判断一个分解是否保持依赖。

下面以一个例子来说明一下。

: 设 $R(A,B,C,D)$ ,  $F = \{A \rightarrow B, C \rightarrow D\}$ ,  $\rho = \{R_1(\{A,B\}, \{A \rightarrow B\}), R_2(\{C,D\}, \{C \rightarrow D\})\}$ 。

因为  $F = \{A \rightarrow B, C \rightarrow D\}$ ,  $F_1 \cup F_2 = \{A \rightarrow B, C \rightarrow D\}$

所以  $F^+ = (F_1 \cup F_2)^+$

➤该例还说明，一个具有依赖保持性的分解不一定具有连接不失真性。反之，一个连接不失真分解也不一定具有依赖保持性。

例：设 $R(A,B,C)$ ,  $F = \{A \rightarrow B, C \rightarrow B\}$ ,  $\rho = \{R_1(\{A,B\}, \{A \rightarrow B\}), R_2(\{A,C\}, \{A \rightarrow C\})\}$ 。

$R_1 \cap R_2 = A$ ,  $R_1 - R_2 = B$ ,  $R_2 - R_1 = C$

$R_1 \cap R_2 \rightarrow R_1 - R_2 = A \rightarrow B \in F$

但 $F = \{A \rightarrow B, C \rightarrow B\}$ ,  $F_1 \cup F_2 = \{A \rightarrow B, A \rightarrow C\}$ , 即 $F^+ \neq (F_1 \cup F_2)^+$

可见具有连接不失真性，但不具有依赖保持性。



## 6.5 关系模式的规范化

### 6.5.1 范式

范式的概念是由E. F. Codd在1970年首先提出来的。

满足特定要求的模式称之为范式。

所谓模式规范化，就是对关系模式应当满足的条件的某种处理，其目的是：

- (1)消除异常现象。
- (2)方便用户使用，简化检索操作。
- (3)加强数据独立性。
- (4)使关系模式更灵活，更容易使用非过程化的高级查询语言。
- (5)更容易进行各种查询统计工作。

关系规范化的条件可以分成几级，每一级称为一个范式，记为XNF，其中X表示级别,NF是范式(Normal Form)，即关系模式满足的条件。

范式的级别越高，条件越严格，因此有：

$$\dots \subset 5NF \subset 4NF \subset BCNF \subset 3NF \subset 2NF \subset 1NF$$





## 1) 第一范式 (1NF)

**定义6.14 (1NF)：** 如果一个关系模式R的每个属性的域都只包含单纯值，而不是一些值的集合或元组，则称R是第一范式，记为  $R \in 1NF$ ,

把一个非规范化关系模式变为1NF有两种方法，一是把不含单纯值的属性分解为多个属性，并使它们仅含单纯值。

例如，设模式：

$P(PNO, PNAME, QOH, PJ(PJNO, PJNAME, PJMNO, PQC))$

将模式P变为：

$P(\underline{PNO}, PNAME, QOH, \underline{PJNO}, PJNAME, PJMNO, PQC)$

第二种方法是把关系模式分解，并使每个关系都符合1NF。则：

$P1(\underline{PNO}, PNAME, QOH)$

$PJ1(\underline{PJNO}, PJNAME, PJMNO, PQC)$

关系PJ1存在异常现象，例如，当一个新工程刚提出，仅有工程名，没有工程号，也没有使用零部件，此时工程数据就不能写入数据库。原因是存在部分函数依赖：

$$(PNO, PJNO) \xrightarrow{f} PQC \quad (PNO, PJNO) \xrightarrow{p} PJNAME, PJMNO$$





## 2) 第二范式 (2NF)

**定义6.15 (2NF)：** 如果关系模式 $R \in 1NF$ ，且它的任一非主属性都完全函数依赖于任一候选关键字，则称 $R$ 满足第二范式，记为 $R \in 2NF$ 。

把一个1NF的关系模式变为2NF的方法是，通过模式分解，使任一非主属性都完全函数依赖于它的任一候选关键字。

例如对上例，若把PJ1进一步分解成：

PJ2 (PNO,PJNO,PQC)

J(PJNO,PJNAME,PJMNO)



### 3) 第三范式 (3NF)

考察关系模式S (SNO, SNAME, DNAME, DADDR), SNO为候选关键字。但若假定一个系的学生们的所在系地址相同, 即一个系的学生们的DADDR值一样。显然,  $SNO \rightarrow DNAME, DNAME \rightarrow DADDR$ , 故 $SNO \rightarrow DADDR$ , 该关系模式在DADDR列存在高度数据冗余。

这是由于原关系模式中存在传递函数依赖。因此, 要消除数据冗余这种异常现象, 必须使关系模式中不出现传递函数依赖。

**定义6.16 (3NF):** 如果关系模式 $R \in 2NF$ , 且每一个非主属性不传递依赖于任一候选关键字, 则称 $R \in 3NF$ 。

例如把关系模式S分解成:

ST (SNO, NAME, DNAME)

DEPT(DNAME, DADDR)

3NF定义告诉我们, 一个关系模式满足3NF的充分必要条件是, 它的每个非主属性既不部分依赖也不传递依赖于候选关键字。



## 4 ) Boyce-Codd范式 ( BCNF)

**定义6.17 (BCNF)：** 设有关系模式R及其函数依赖集F,X和A是R的属性集合，且 $A \not\subseteq X$ 。如果只要R满足 $X \rightarrow A$ ，X就必包含R的一个候选关键字，则称R满足BCNF，记为 $R \in \text{BCNF}$ 。

该定义主要有三点：

- (1)所有非主属性A对键都是完全函数依赖的( $R \in 2\text{NF}$ )。
- (2)没有属性完全函数依赖于非键的任何属性组( $R \in 3\text{NF}$ )。
- (3)所有主属性对不包含它的键是完全函数依赖的（新增加条件）。

例如，模式S (NAME,SEX,BIRTH,ADDR,DNAME)的主属性为：NAME,SEX,BIRTH 和 ADDR，候选关键字为：(NAME,SEX)、(NAME,BIRTH) 以及 (NAME, ADDR)。定义中的A为(ADDR,DNAME)。显然有：

$$(\text{NAME}, \text{SEX}) \xrightarrow{f} \text{ADDR} \qquad (\text{NAME}, \text{BIRTH}) \xrightarrow{f} \text{ADDR}$$



## 事例

例6.13：关系模式STC（SNO，TNO，CNO），SNO表示学号，TNO表示教师编号，CNO表示课程号。每一个教师只教一门课，每门课有若干教师，某一个学生选定某门课，就对应一个固定教师。试判断ST的最高范式。

解 由语义可得到如下的函数依赖：

$(SNO, CNO) \rightarrow TNO$ ,  $(SNO, TNO) \rightarrow CNO$ ,  $TNO \rightarrow CNO$

这里（SNO，CNO），（SNO，TNO）都是候选关键字。

因为没有任何非主属性对候选关键字部分依赖，所以 $STC \in 2NF$ 。

没有任何非主属性对候选关键字传递依赖，所以 $STC \in 3NF$ 。

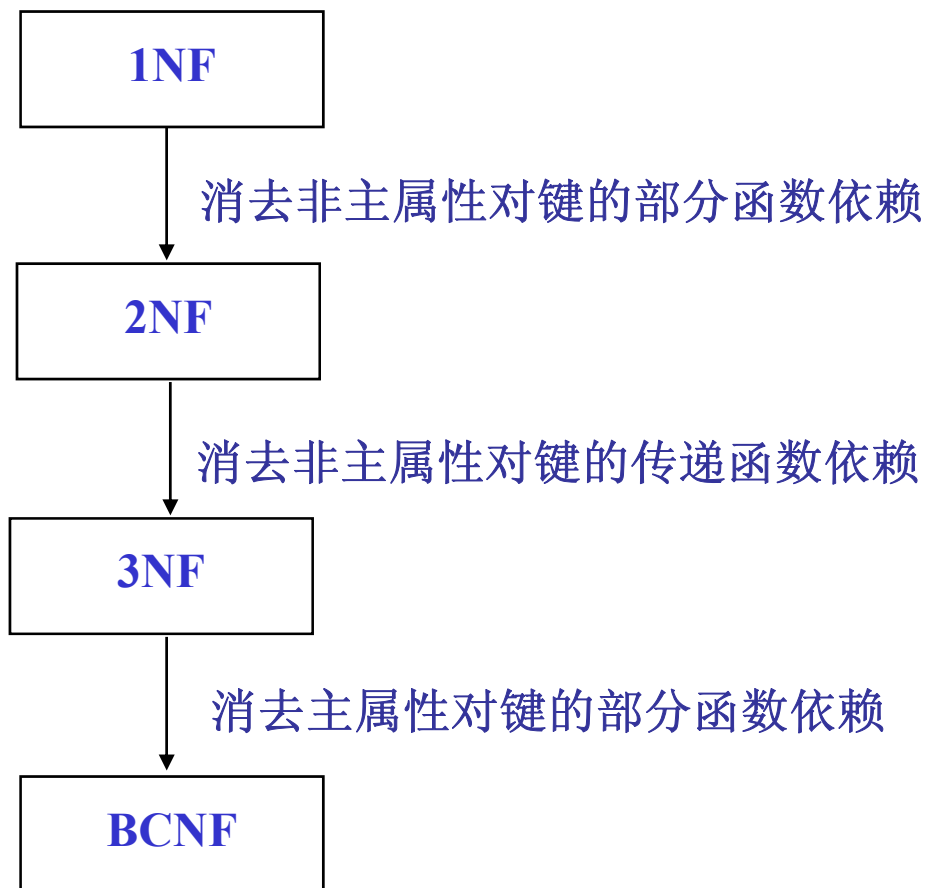
但在F中有 $TNO \rightarrow CNO$ ，而TNO不包含候选关键字，所以STC不是BCNF关系

这里我们可以将STC（SNO，TNO，CNO）分解成ST（SNO，TNO）和TC（TNO，CNO），它们都是BCNF。



# 范式之间的关系

---





## 6.5.2 模式分解的算法

按照上面讨论的模式分解理论，一个模式分解必须满足：

- ①连接不失真性；
- ②依赖保持性
- ③某一级范式。

但事实上不能顺利地同时满足上述三个条件。一般而言：

- (1)若要求连接不失真，分解可达到BCNF；
- (2)若要求依赖保持，则分解可达到3NF，但不一定能达到BCNF。
- (3)若同时要求连接不失真和依赖保持，则分解可达到3NF，但不一定能达到BCNF。



## 1) 结果为BCNF的连接不失真分解

### 定理6.6: 分解定理

(1) 设 $F$ 是关系模式 $R$ 的函数依赖集,  $\rho = \{R_1, R_2, \dots, R_k\}$  是 $R$ 的一个分解, 且对于 $F$ 有连接不失真性。设 $F_i$ 为 $F$ 在 $R_i$ 上的投影, 即:

如果 $X$ 和 $Y$ 均为 $R_i$ 的子集, 则 $X \rightarrow Y \in F^+$ 。又设 $\rho_1 = \{S_1, S_2, \dots, S_m\}$  为 $R_i$ 的一个分解, 且对于 $F_i$ 具有连接不失真性。如果将 $R$ 分解为

$$\{R_1, R_2, \dots, R_{i-1}, S_1, S_2, \dots, S_m, R_{i+1}, \dots, R_k\}$$

则这一分解相对于 $F$ 的一个连接不失真性分解。

(2) 设 $\rho_2 = \{R_1, R_2, \dots, R_k, R_{k+1}, \dots, R_n\}$  为 $R$ 的一个分解, 其中包含了 $\rho$ 的那些关系模式, 则 $\rho_2$ 相对于 $F$ 的一个连接不失真性分解。



## 结果为BCNF的连接不失真分解算法

输入：  $R(U, F)$

输出： 分解  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$ ，且，满足BCNF。

方法： 反复应用定理6—10（分解定理），逐步分解关系模式  $R$ ，使每次分解具有连接不失真性，并且分解出来的模式是BCNF。

- ① 置初值  $\rho = \{R\}$ ；
- ② 如果  $\rho$  中所有关系模式都是BCNF，则转④；
- ③ 如果  $\rho$  中有一个关系模式  $S$  不是BCNF，则  $S$  中必能找到一个函数依赖  $X \rightarrow A$  有  $X$  不是  $S$  的键，且  $A \notin X$ ，设  $S_1 = XA$ ， $S_2 = S - A$ ，用分解  $\{S_1, S_2\}$  代替  $S$ ，则转②；
- ④ 分解结束，输出  $\rho$ 。





## 事例

例6.14：设有关系模式CTHRSG(C,T,H,R,S,G)及其函数依赖集 $F = \{ CS \rightarrow G, C \rightarrow T, HR \rightarrow C, HS \rightarrow R, TH \rightarrow R \}$ 。

### (1) 求所有候选关键字

如果直接根据候选关键字的定义来求一个关系模式的所有关键字：

- 若属性A仅出现在所有函数依赖的右部，则它一定不包含在任何候选关键字中；
- 若属性A仅出现在所有函数依赖的左部，则它一定包含在某个候选关键字中；
- 若属性A既出现在函数依赖的右部，又出现在左部，则它可能包含在候选关键字中；
- 在上述基础上求属性集闭包。

对本例,G仅出现在函数依赖的右部，则它不包含在候选关键字中；又属性H和S仅出现在函数依赖的左部，则H和S必包含在候选关键字中。

计算 $(HS)^+$ 为： $(HS)^{(0)} = HS$   $(HS)^{(1)} = HSR$   $(HS)^{(2)} = HSRC$   
 $(HS)^{(3)} = CTHRSG$   $(HS)^{(4)} = CTHRSG$

即 $(HS)^+ = CTHRSG$ ，故HS是模式CTHRSG的唯一关键字。

## (2) 分解

首先在F中找出这样一个函数依赖 $X \rightarrow A$ ，其中X不包含R的任何候选关键字，也不包含A。把R分解成 $R_1(X, A)$ 和 $R_2(S - A)$ 。

➤ 对本例首先考虑 $CS \rightarrow G$ ，则 $CTHRSG = \{CSG, CTHRS\}$ 。

为进一步分解，需求 $F^+$ 在CSG和CTHRS上的投影：

$$\Pi_{CSG}(F) = \{CS \rightarrow G\}; \quad \Pi_{CTHRS}(F) = \{C \rightarrow T, TH \rightarrow R, HR \rightarrow C, HS \rightarrow R\} = F_1$$

很显然，模式CSG是BCNF。模式CTHRS不是BCNF，还要继续分解。

➤ (2-1)求得CTHRS的候选关键字为HS。

➤ (2-2)再分解CTHRS，选 $C \rightarrow T$ ，将CTHRS分解为  $CTHRS = \{CT, CHRS\}$ 。

函数依赖集CT上投影的最小覆盖是 $C \rightarrow T$ ，在CHRS上的投影的最小覆盖是 $CH \rightarrow R, HS \rightarrow R, HR \rightarrow C$ 。记作：

$$\Pi_{CT}(F_1) = \{C \rightarrow T\}; \quad \Pi_{CHRS}(F_1) = \{CH \rightarrow R, HS \rightarrow R, HR \rightarrow C\} = F_2$$

显然，模式CT为BCNF，但模式CHRS不是BCNF，还要继续分解。

➤ (2-3)求得CHRS的唯一关键字为HS。

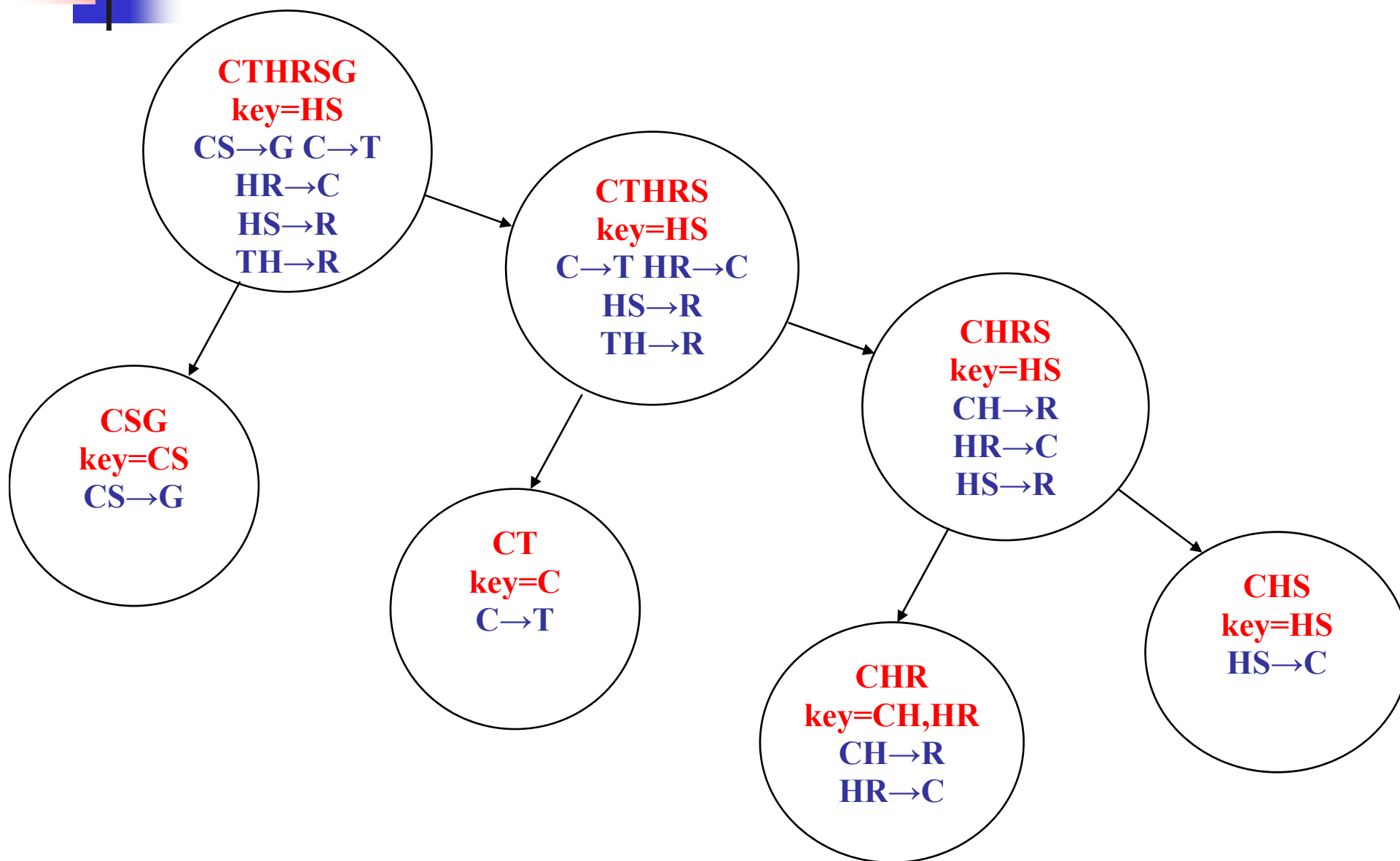
➤ (2-4)再分解CHRS，选 $CH \rightarrow R$ ，将CHRS分解为  $CHRS = \{CHR, CHS\}$ 。

$F_2$ 在CHR、CHS上投影的最小覆盖为：

$$\Pi_{CHR}(F_2) = \{CH \rightarrow R, HR \rightarrow C\}; \quad \Pi_{CHS}(F_2) = \{HS \rightarrow C\}$$

在模式CHR中，HC、HR为键，其所有决定因素都是键，在模式CHS中，HS为键，显然CHR、CHS都为BCNF。

# 分解树





## 2) 结果为3NF的依赖保持分解

算法6-4: 结果为3NF的依赖保持分解算法

输入: 关系模式R和函数依赖集F

输出: 结果为3NF的一个依赖保持分解

步骤:

(1)如果R中有某些属性与F的最小覆盖F'中的每个依赖的左边和右边都无关, 原则上可由这些属性构成一个关系模式, 并从R中将它们消除;

(2)如果F'中有一个依赖涉及到R的所有属性, 则输出R;

(3)否则, 输出一个分解 $\rho$ ,它由模式XA组成, 其中 $X \rightarrow A \in F$ 。但当 $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ 均属于F'时, 则用模式 $XA_1A_2\dots A_n$ 代替 $XA_i(i=1,2, \dots,n)$ 。

例6-15: 对于上例,

$F' = \{ \{ C \rightarrow T, CS \rightarrow G, HT \rightarrow R, HR \rightarrow C, CH \rightarrow R, HS \rightarrow R \} , KEY = HS \}$

所以  $\rho = \{ CT, CSG, HRT, CHR, HSR \}$



### 3) 结果为3NF且具有依赖保持和连接不失真的分解

定理6-11: 设 $\delta$ 是由结果为3NF的依赖保持分解算法得到的3NF分解,  $X$ 为 $R$ 的一个候选关键字, 则 $\tau = \delta \cup \{X\}$ 是 $R$ 的一个分解, 且 $\tau$ 中的所有关系模式均满足3NF, 同时, 既具有连接不失真性, 又具有依赖保持性。

例: 已知 $R(C, T, H, R, S, G)$ ,

$F' = \{C \rightarrow T, HR \rightarrow C, CS \rightarrow G, HS \rightarrow R, HT \rightarrow R\}$ ,  $KEY = HS$ ,

则  $\tau = \{CT, CSG, HRT, CHR, HSR, HS\}$

但 $HS \subseteq HSR$ , 故

$\tau = \{CT, CSG, HRT, CHR, HSR\}$





## 6.6 多值函数依赖与4NF

### 6.6.1 BCNF关系模式存在的问题(CTB是关键字)

C	T	B
高等数学	张华民	高等数学
高等数学	张华民	高等数学教程
高等数学	王天华	高等数学
高等数学	王天华	高等数学教程
高等数学	林静	高等数学
高等数学	林静	高等数学教程
普通物理	吴刚	物理学
普通物理	吴刚	普通物理
普通物理	吴刚	普通物理基础
普通物理	李晓芳	物理学
普通物理	李晓芳	普通物理
普通物理	李晓芳	普通物理基础

## 6.6.2 多值函数依赖

为了形式地定义多值依赖，根据上例，构造一个抽象关系 $R(U)$ （如下表），并设 $X, Y$ 是 $U$ 的子集，其余属性为 $Z=U-X-Y$ 。又设 $s, t, u, v$ 是该关系中的任意元组。

$R(U)$	$X$	$Y$	$U-X-Y$
$s$	$s[X]$	$s[Y]$	$s[U-X-Y]$
...	...	...	...
$t$	$t[X]$	$t[Y]$	$t[U-X-Y]$
...	...	...	...
$u$	$u[X]=s[X]$	$u[Y]=s[Y]$	$u[U-X-Y]=t[U-X-Y]$
...	...	...	...
$v$	$v[X]=t[X]$	$v[Y]=t[Y]$	$v[U-X-Y]=s[U-X-Y]$

定义6-23：设有 $R(U), X, Y$ 是 $U$ 的子集， $Z=U-X-Y$ 。多值依赖 $X \twoheadrightarrow Y$ 成立，当且仅当对 $R$ 的任一具体关系 $r$ ，给定一对 $(X, Z)$ 值，有一组 $Y$ 值与之对应，且这种对应关系与 $Z$ 值无关。



## 1) 多值依赖的独立推理规则

(1) MVD0: 多值互补性或对称性(Complementation)

$X \twoheadrightarrow Y$ , 则  $X \twoheadrightarrow U - X - Y$

(2) MVD1: 多值依赖自反性(Reflexivity)

$Y \subseteq X \subseteq U$ , 则  $X \twoheadrightarrow Y$

(3) MVD2: 多值依赖增广性(Augmentation)

$X \twoheadrightarrow Y$ , 且  $V \subseteq W$ , 则  $WX \twoheadrightarrow VY$

(4) MVD3: 多值依赖传递性(Transitivity)

$X \twoheadrightarrow Y$ , 且  $Y \twoheadrightarrow Z$ , 则  $X \twoheadrightarrow Z - Y$





## 2) 多值依赖的其他推理规则

---

(1) MVD4: 伪传递性

$X \twoheadrightarrow Y$ , 且  $WY \twoheadrightarrow Z$ , 则  $XW \twoheadrightarrow Z - WY$

(2) MVD5: 多值依赖的合并规则

$X \twoheadrightarrow Y$ , 且  $X \twoheadrightarrow Z$ , 则  $X \twoheadrightarrow YZ$

(3) MVD6: 多值依赖的分解或投影规则

$X \twoheadrightarrow Y$ , 且  $X \twoheadrightarrow Z$ , 则  $X \twoheadrightarrow Y \cap Z$ ,  $X \twoheadrightarrow Y - Z$ ,  $X \twoheadrightarrow Z - Y$



## 6.6.3 4NF

从上面的例子可以看到，一个存在多值依赖的关系模式，其数据冗余量特别大，而且还有其它异常现象。

如果把上面的关系TEACH分解成两个关系C\_T和C\_B，如下表。

**(a) C\_T关系**

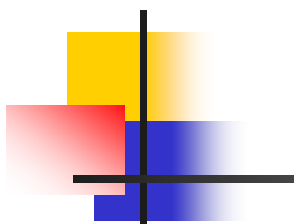
C	T
高等数学	张华民
高等数学	王天华
高等数学	林静
普通物理	吴刚
普通物理	李晓芳

**(b) C\_B关系**

C	B
高等数学	高等数学
高等数学	高等数学教程
普通物理	物理学
普通物理	普通物理
普通物理	普通物理基础

把关系分解成仅含平凡多值依赖的多个关系,定义一个其条件比BCNF更苛刻的范式，即4NF。一个满足4NF的关系模式的特点是：

- 该关系模式满足BCNF；
- 该关系模式只允许出现平凡多值依赖。



**定义6.19 (4NF) :**关系模式 $R(U) \in 1NF$ ,若 $X \twoheadrightarrow Y (Y \notin X)$ 是非平凡的多值依赖, 且 $X$ 包含关键字, (或只出现平凡多值依赖, 不存在非平凡多值依赖), 则称 $R(U)$ 满足第四范式, 记为 $R \in 4NF$ 。

上例中的关系模式TEACH为例, 虽然是BCNF, 但不是4NF, 因为在TEACH (C,T,B)中有: $C \twoheadrightarrow T$ 和 $C \twoheadrightarrow B$ , 关键字是CTB。虽然 $TEACH \in BCNF$ , 但C不是关键字, 即 $TEACH \notin 4NF$ 。

将TEACH分解后产生的 $C\_T(C,T)$ 和 $C\_B(C,B)$ , 显然因为有 $C \twoheadrightarrow T$ 和 $C \twoheadrightarrow B$ , 故不存在非平凡多值依赖, 从而有 $C\_T \in 4NF$ 和 $C\_B \in 4NF$





## 本章小结

- 本章讨论如何设计关系模式问题。关系模式设计得好与坏，直接影响到数据冗余度、数据一致性问题。要设计好的数据库模式，必须有一定的理论为基础。这就是模式规范化理论。
- 在数据库中，数据冗余是指同一个数据存储了多次，由数据冗余将会引起各种操作异常。通过把模式分解成若干比较小的关系模式可以消除冗余。
- 函数依赖 $X \rightarrow Y$ 是数据之间最基本的一种联系，在关系中有两个元组，如果X值相等那么要求Y值也相等。FD有一个完备的推理规则集。
- 关系模式在分解时应保持“等价”，有数据等价和语义等价两种，分别用无损分解和保持依赖两个特征来衡量。前者能保持泛关系在投影联接以后仍能恢复回来，而后者能保证数据在投影或联接中其语义不会发生变化，也就是不会违反FD的语义。但无损分解与保持依赖两者之间没有必然的联系。



## 本章小结（续）

- 范式是衡量模式优劣的标准，范式表达了模式中数据依赖之间应满足的联系。如果关系模式R是3NF，那么R上成立的非平凡FD都应该左边是超键或右边是非主属性。如果关系模式R是BCNF，那么R上成立的非平凡的FD都应该左边是超键。范式的级别越高，其数据冗余和操作异常现象就越少。
- 分解成BCNF模式集的算法能保持无损分解，但不一定能保持FD集。而分解成3NF模式集的算法既能保持无损分解，又能保持FD集。
- 关系模式的规范化过程实际上是一个“分解”过程：把逻辑上独立的信息放在独立的关系模式中。分解是解决数据冗余的主要方法，也是规范化的一条原则：“关系模式有冗余问题就分解它”。