

# 数据管理基础

## 第6章 关系数据理论

### (模式分解)

智能软件与工程学院



## 6.4 模式的分解

- ❑ 定义6.16 关系模式的分解
- ❑ 定义6.17 函数依赖集在属性组上的投影
- ❑ 模式分解的三个定义
  - 准确说：两条性质，三种分解准则
- ❑ 引理6.4
- ❑ 无损连接
  - 定义6.18，定理6.5
  - （自学）算法6.2 & 定理6.4
- ❑ 保持函数依赖
  - 定义6.19
- ❑ 模式分解算法：算法6.3、6.4、6.5
  - 引理6.5、6.6
  - 定理6.6，算法6.6
- ❑ 补充算法

## 6.4 模式的分解

□ [定义6.16] 关系模式 $R(U, F)$ 的一个分解是指

$$\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)\}$$

其中：

- $U = \bigcup_{i=1}^n U_i$ ，并且没有 $U_i \subseteq U_j$ ， $1 \leq i, j \leq n$
- $F_i$ 是 $F$ 在属性集 $U_i$ 上的投影，也被称为‘ $F_i$ 是 $F$ 在关系 $R_i$ 上的投影’

□ [定义6.17] 与函数依赖集合 $\{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq U_i\}$ 等价的 $F_i$ 叫作 $F$ 在属性集 $U_i$ 上的投影。

- 在一个关系模式 $R(U, F)$ 中，可以用与 $F$ 等价的依赖集 $G$ 来取代 $F$ 。
- 原因：两个关系模式 $R_1(U, F)$ 和 $R_2(U, G)$ ，如果 $F$ 与 $G$ 等价，那么 $R_1$ 的关系一定是 $R_2$ 的关系；反过来， $R_2$ 的关系也一定是 $R_1$ 的关系。

## 6.4 模式的分解

- ❑ 6.4.1 模式分解的三个定义
- ❑ 6.4.2 模式分解的无损连接性和保持函数依赖性
- ❑ 6.4.3 模式分解的算法

## 6.4.1 模式分解的三个定义

- ❑ 对于一个模式的分解是多种多样的，但是分解后产生的模式应与原模式等价。
- ❑ 从不同的角度去观察，对‘等价’的概念形成了三种不同的定义：
  - 无损连接 (lossless join)
  - 保持函数依赖 (preserve functional dependency)
  - 既有‘无损连接’，又要‘保持函数依赖’
- ❑ 其中：
  - ‘无损连接’ 是进行模式分解必须满足的要求；
  - 单纯考虑一个分解 $\rho$ 满足‘保持函数依赖’很容易实现，但这样的分解不一定满足‘无损连接性’；
  - 模式分解的目标：既有‘无损连接’，又要‘保持函数依赖’（但不一定能满足）

# 不同分解的示例

❑ [6.14] 关系模式  $R(\{Sno, Sdept, Mname\}, \{Sno \rightarrow Sdept, Sdept \rightarrow Mname\})$

❑ 分解一：

$$\rho_1 = \{R_1(\{Sno\}, \emptyset), R_2(\{Sdept\}, \emptyset), R_3(\{Mname\}, \emptyset)\}$$

➤ 分解  $\rho_1$  既没有无损连接性，也不保持函数依赖；

❑ 分解二：

$$\rho_2 = \{R_1(\{Sno, Sdept\}, \{Sno \rightarrow Sdept\}), \\ R_2(\{Sno, Mname\}, \{Sno \rightarrow Mname\})\}$$

➤ 分解  $\rho_2$  具有无损连接性，但不保持函数依赖；

❑ 分解三：

$$\rho_3 = \{R_1(\{Sno, Sdept\}, \{Sno \rightarrow Sdept\}), \\ R_2(\{Sdept, Mname\}, \{Sdept \rightarrow Mname\})\}$$

➤ 分解  $\rho_3$  既具有无损连接性，又保持函数依赖。

Sno	Sdept	Mname
S1	D1	张五
S2	D1	张五
S3	D2	李四
S4	D3	王一

## 6.4.2 模式分解的无损连接性和保持函数依赖性

### □ 符号定义

- 设  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$  是  $R(U, F)$  的一个分解,  $r$  是  $R(U, F)$  的一个关系
- $r_i = \pi_{R_i}(r) = \{t.U_i \mid t \in r\}$  是关系  $r$  在关系模式  $R_i$  上的投影
- 定义  $m_\rho(r) = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$ , 即  $m_\rho(r)$  是  $r$  在  $\rho$  中各关系模式上投影的连接。

□ **[引理6.4]** 设  $R(U, F)$  是一个关系模式,  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$  是  $R$  的一个分解,  $r$  是  $R$  的一个关系,  $r_i = \pi_{R_i}(r)$ , 则

- ①  $r \subseteq m_\rho(r)$
- ② 若  $s = m_\rho(r)$ , 则  $\pi_{R_i}(s) = r_i$ , 即  $\pi_{R_i}(m_\rho(r)) = r_i$
- ③  $m_\rho(m_\rho(r)) = m_\rho(r)$

说明: ①可以对  $k$  用归纳法来证明; ②如果关系  $R$  和  $S$  不存在同名属性, 那么  $R \bowtie S = R \times S$ .

## □ [定义6.18]

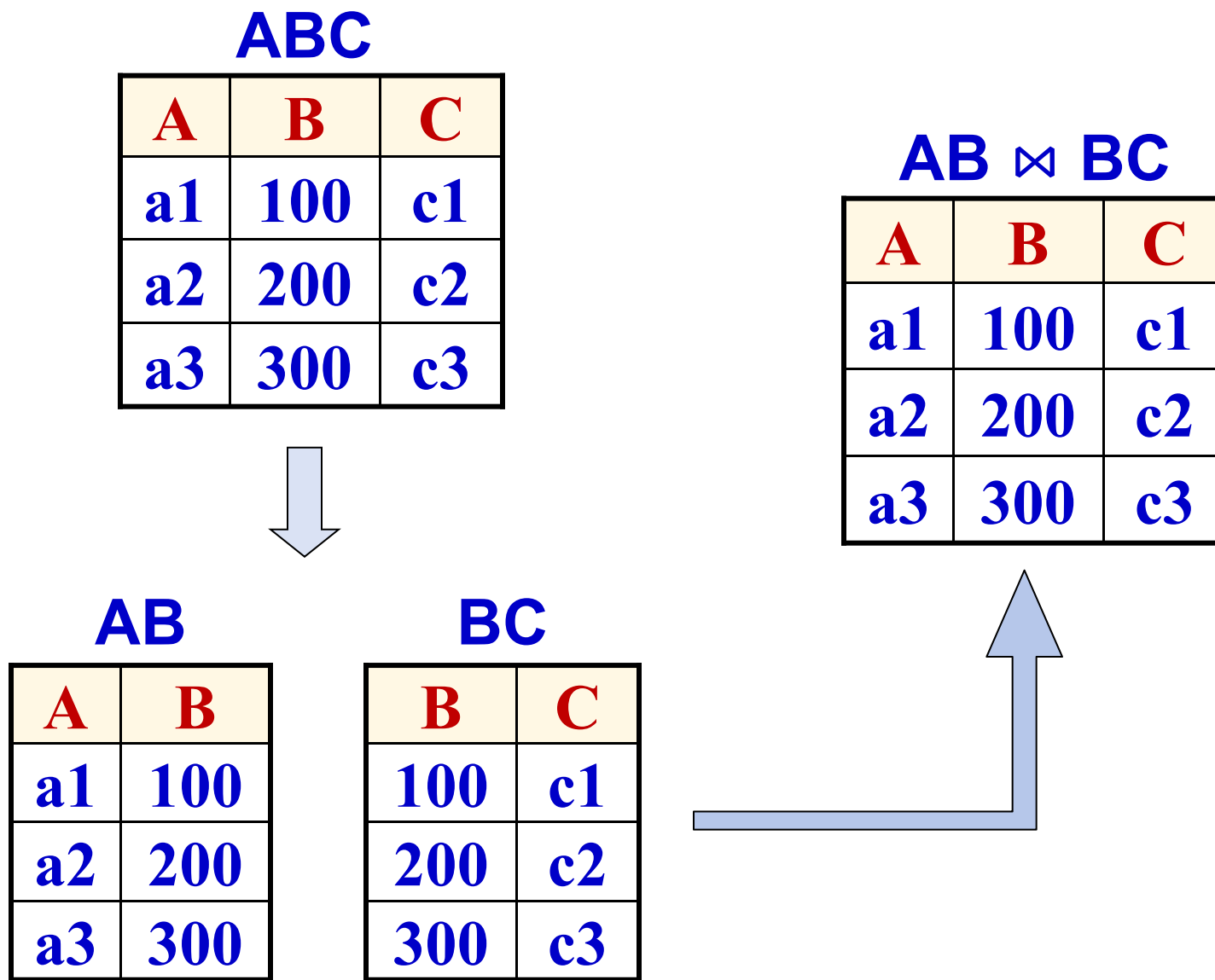
设  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$  是  $R(U, F)$  的一个分解, 若对  $R(U, F)$  的任何一个关系  $r$  均有  $r = m_\rho(r)$  成立, 则称分解  $\rho$  具有**无损连接性**。简称  $\rho$  为**无损分解**。

□ 对于不具备‘无损连接性’的分解来说, 关系  $r$  与分解后的子关系满足:

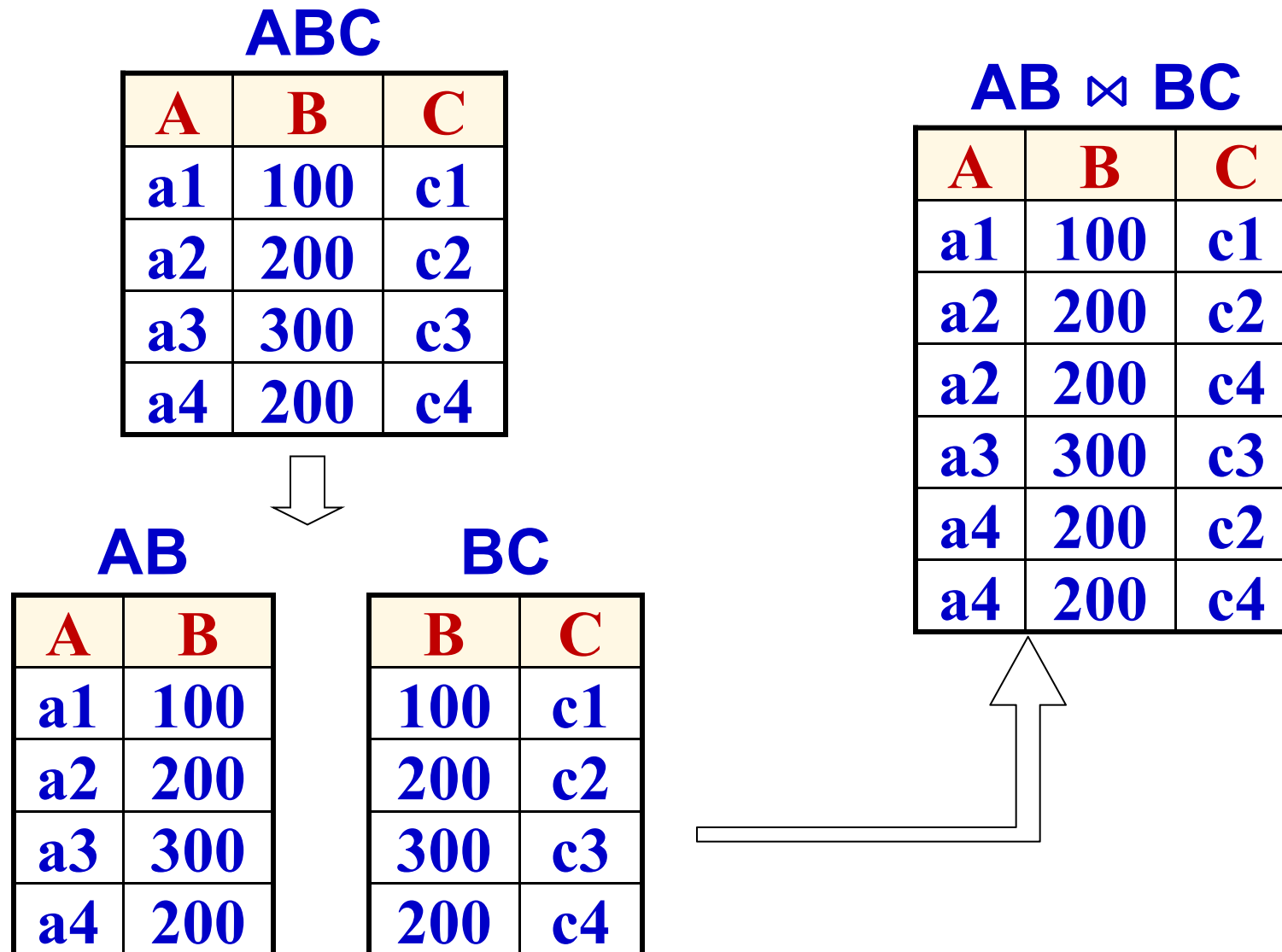
$$r \subset \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$$



# 一个具有 ‘无损连接性’ 的模式分解



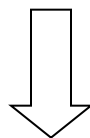
# 一个不具备‘无损连接性’的模式分解



## 无损分解

ABC

A	B	C
a1	100	c1
a2	200	c2
a3	300	c3



AB

A	B
a1	100
a2	200
a3	300

BC

B	C
100	c1
200	c2
300	c3

## 有损分解

ABC

A	B	C
a1	100	c1
a2	200	c2
a3	300	c3
a4	200	c4



AB

A	B
a1	100
a2	200
a3	300
a4	200

BC

B	C
100	c1
200	c2
300	c3
200	c4

# 模式分解的‘无损连接’探究 (1)

- ❑ 从关系模式**ABC**（如下左图）到关系模式**AB**和**BC**的分解为何具有‘无损连接性’？

ABC					BC	
A	B	C			B	C
a1	100	c1	⇒	a1	100	c1
a2	200	c2		a2	200	c2
a3	300	c3		a3	300	c3

- 在关系模式**ABC**中具有如下的函数依赖： $B \rightarrow C$
- 因此从模式**ABC**到模式**AB**和**BC**的分解具有无损连接性：

$$ABC \equiv AB \bowtie BC$$

WHY ?

## 模式分解的‘无损连接’探究 (2)

- 在满足依赖关系  $B \rightarrow C$  的前提下，向关系 **ABC** 中插入一个新的元组  $(a4, 200, c2)$ ，得到一个新的关系  $ABC^*$  (下图左)。从关系  $ABC^*$  到关系 **AB** 和关系 **BC** 的分解仍然具有无损连接性！

$ABC^*$			<b>AB</b>		<b>BC</b>	
A	B	C	A	B	B	C
a1	100	c1	a1	100	100	c1
a2	200	c2	a2	200	200	c2
a3	300	c3	a3	300	300	c3
a4	200	c2	a4	200		

## 模式分解的‘无损连接’探究 (3)

- ❑ 如果违反函数依赖  $B \rightarrow C$  的约束，强行向关系 **ABC** 插入一个新的元组  $(a4, 200, c4)$ ，得到的新关系  $ABC^{**}$  (下图左) 将不再满足  $B \rightarrow C$  的约束，因而该分解则不再具有无损连接性。 ( **WHY ?** )

$ABC^{**}$			$AB$		$BC$	
A	B	C	A	B	B	C
a1	100	c1	a1	100	100	c1
a2	200	c2	a2	200	200	c2
a3	300	c3	a3	300	300	c3
a4	200	c4	a4	200	200	c4

# 模式分解无损连接性的判定

□ [定理6.5] 对于 $R(U, F)$ 的一个分解 $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2)\}$ , 如果 $(U_1 \cap U_2) \rightarrow (U_1 - U_2) \in F^+$  或  $(U_1 \cap U_2) \rightarrow (U_2 - U_1) \in F^+$  成立, 则分解 $\rho$ 具有无损连接性。

□ 说明:

- 对 $k$ 用归纳法, 可以判断 $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$ 是不是 $R(U, F)$ 的无损分解;
- 下面给出定理6.5的证明。

## 定理6.5证明 (1)

□ [思路] 对 $R(U, F)$ 的任何一个关系 $r$ , 关系 $r$ 在 $R_1(U_1, F_1)$ 和 $R_2(U_2, F_2)$ 上的投影是 $r_1$  和  $r_2$ , 证明:  $r \subseteq r_1 \bowtie r_2$  and  $r_1 \bowtie r_2 \subseteq r$

□ 设:  $U_1 \cap U_2 = A$ ,  $U_1 - U_2 = B$ ,  $U_2 - U_1 = C$ , 则有:

➤  $U_1 = AB$ ,  $U_2 = AC$ ,  $U = ABC$

➤  $(U_1 \cap U_2) \rightarrow (U_1 - U_2)$  就是  $A \rightarrow B$

➤  $(U_1 \cap U_2) \rightarrow (U_2 - U_1)$  就是  $A \rightarrow C$

下面证明: 当  $A \rightarrow B$  成立时, 对 $R(U, F)$ 的任何一个关系 $r$ , 经分解投影得到两个子关系  $r_1 = r[A, B]$  和  $r_2 = r[A, C]$ , 必有  $r = (r_1 \bowtie r_2)$  成立, 即下述两个结论成立:

结论1: **if  $t \in r$  then  $t \in (r_1 \bowtie r_2)$**

结论2: **if  $t \in (r_1 \bowtie r_2)$  then  $t \in r$**



## 定理6.5证明 (2)

**结论1: if  $t \in r$  then  $t \in (r_1 \bowtie r_2)$**

□ 对关系 $r$ 中的任意一个元组 $t$ ，它在两个子关系中的投影结果分别是关系 $r_1$ 中的元组 $t_1$ 和关系 $r_2$ 中的元组 $t_2$ ，并满足：

$$t_1[A] = t[A] = t_2[A], t_1[B] = t[B], t_2[C] = t[C] \dots\dots\dots \textcircled{1}$$

□ 因为有  $t_1 \in r_1, t_2 \in r_2$  且  $t_1[A] = t_2[A]$ ，所以由 $t_1$ 和 $t_2$ 组合得到的元组  $(t_1[A], t_1[B], t_2[C]) \in (r_1 \bowtie r_2) \dots\dots\dots \textcircled{2}$

□ 依据式①中的等价关系，可将式②变换为

$$(t[A], t[B], t[C]) \in (r_1 \bowtie r_2) \quad \text{即:} \quad t \in (r_1 \bowtie r_2)$$

□ 所以，结论1 成立。

## 定理6.5证明 (3)

**结论2: if  $t \in (r_1 \bowtie r_2)$  then  $t \in r$**

□ 对 $(r_1 \bowtie r_2)$ 中的任意一个元组 $t$ , 必存在关系 $r_1$ 中的一个元组 $t_1$ 和关系 $r_2$ 中的一个元组 $t_2$ , 并满足:

$$t[A] = t_1[A] = t_2[A], t[B] = t_1[B], t[C] = t_2[C] \dots\dots\dots \textcircled{3}$$

□ 因为  $t_1 \in r_1$ , 所以在关系 $r$ 中必存在一个元组 $s_1$ ,  $t_1$ 是 $s_1$ 在关系 $R_1$ 上的投影, 即: 存在元组  $s_1 \in r$  且满足

$$s_1[A] = t_1[A], s_1[B] = t_1[B] \dots\dots\dots \textcircled{4}$$

□ 同理, 有: 存在元组  $s_2 \in r$  且满足

$$s_2[A] = t_2[A], s_2[C] = t_2[C] \dots\dots\dots \textcircled{5}$$

□ 由式 $\textcircled{3}\textcircled{4}\textcircled{5}$ 可得

$$s_1[A] = t_1[A] = t_2[A] = s_2[A], \text{ 即: } s_1[A] = s_2[A] \dots\dots\dots \textcircled{6}$$

□ 在关系 $r$ 中, 由 $s_1 \in r, s_2 \in r, s_1[A] = s_2[A]$ , 及 $A \rightarrow B$ 可知:

$$s_1[B] = s_2[B] \dots\dots\dots \textcircled{7}$$

## 定理6.5证明 (4)

$$t[A] = t_1[A] = t_2[A], t[B] = t_1[B], t[C] = t_2[C] \dots\dots\dots \textcircled{3}$$

$$s_1[A] = t_1[A], s_1[B] = t_1[B] \dots\dots\dots \textcircled{4}$$

$$s_2[A] = t_2[A], s_2[C] = t_2[C] \dots\dots\dots \textcircled{5}$$

$$s_1[A] = s_2[A] \dots\dots\dots \textcircled{6}$$

$$s_1[B] = s_2[B] \dots\dots\dots \textcircled{7}$$

□ 由式③④⑤⑥⑦可知：

$$\begin{aligned} s_2[A] &= t_2[A] = t[A] \\ s_2[B] &= s_1[B] = t_1[B] = t[B] \\ s_2[C] &= t_2[C] = t[C] \end{aligned}$$

即  $s_2$  和  $t$  是同一个元组

□ 既然  $s_2$  是关系  $r$  中的元组，那么  $t$  也是  $r$  中的元组，即：  $t \in r$

□ 所以，结论2也成立。

综上所述，结论1和结论2都成立，所以有：  $r = (r_1 \bowtie r_2)$  (证毕)

# 无损连接判断

□ [思考] 设有一个关系模式 $T(A, B, C)$ ，函数依赖集为 $F$ ，请判断下述的分解 $\rho = \{T_1, T_2\}$ 是否具有无损连接性？

	函数依赖集 $F$	分解 $\rho$	是否满足无损连接性？
1	$\{A \rightarrow B\}$	$T_1(A, B) \quad T_2(A, C)$	yes
2	$\{A \rightarrow C, B \rightarrow C\}$	$T_1(A, B) \quad T_2(A, C)$	yes
3	$\{A \rightarrow B\}$	$T_1(A, B) \quad T_2(B, C)$	no
4	$\{A \rightarrow B, B \rightarrow C\}$	$T_1(A, C) \quad T_2(B, C)$	no

# 模式分解的保持函数依赖

- [定义6.19] 若  $F^+ = (F_1 \cup F_2 \cup \dots \cup F_k)^+$ , 则  $R(U, F)$  的分解  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$  保持函数依赖。
- 可以根据引理6.3, 找到判断分解  $\rho$  是否保持函数依赖的方法, 即判断  $F$  和  $(F_1 \cup F_2 \cup \dots \cup F_k)$  是否等价的方法。
  - 显然,  $(F_1 \cup F_2 \cup \dots \cup F_k) \subseteq F^+$  成立
  - 只需要判断:  $F \subseteq (F_1 \cup F_2 \cup \dots \cup F_k)^+$  是否成立?
    - 如果  $F \subseteq (F_1 \cup F_2 \cup \dots \cup F_k)^+$  成立, 则分解  $\rho$  保持函数依赖;
    - 否则, 分解  $\rho$  不具有保持函数依赖性。

## 6.4.3 模式分解的算法

### □ 已经介绍的算法

- 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  $X_F^+$
- [补充算法1] 极小函数依赖集的计算
  - 寻找与函数依赖集  $F$  等价的极小函数依赖集  $G$

### □ 接下来

- [补充算法2] 候选码的计算
- 算法6.4: 转换为3NF既有无损连接性又保持函数依赖的分解。
- 引理6.5 & 引理6.6

## [回顾] 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: 函数依赖集  $F$ , 属性集  $X$

❑ 输出: 属性集闭包  $X_F^+$

① 令  $X^{(0)} = X$ ,  $i = 0$

② 求  $Y$ , 这里  $Y = \{ A \mid (\exists V)(\exists W)(V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W) \}$ .

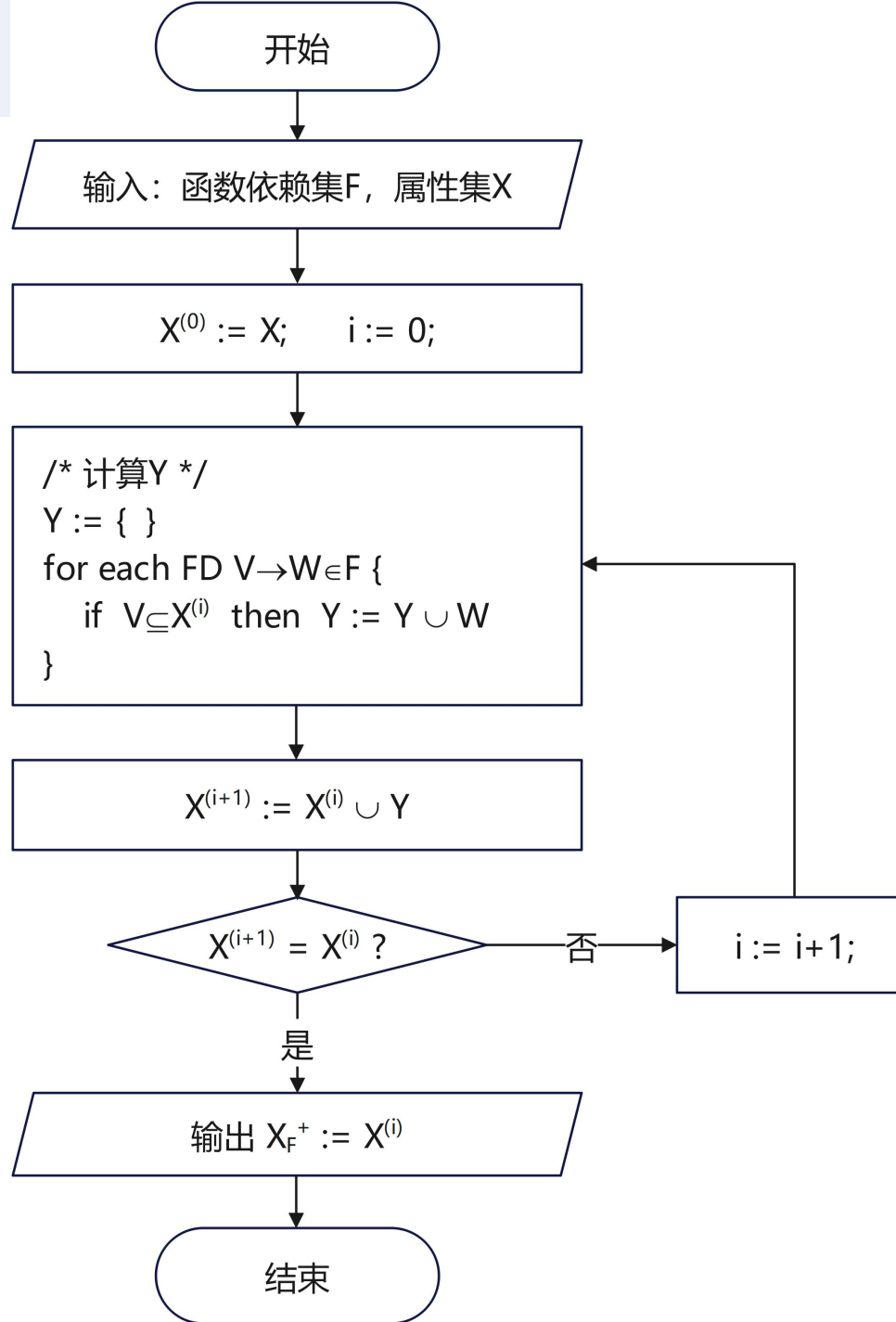
③  $X^{(i+1)} = Y \cup X^{(i)}$ .

④ 判断  $X^{(i+1)} = X^{(i)}$  是否成立?

⑤ 若  $X^{(i+1)} = X^{(i)}$  成立 或  $X^{(i)} = U$  成立, 则  $X^{(i)}$  就是  $X_F^+$ , 算法终止。

⑥ 若否, 则  $i = i + 1$ , 返回第②步。

## 算法6.1流程图





## [回顾]补充算法1：寻找与函数依赖集 $F$ 等价的极小函数依赖集 $G$

### 1. 令 $G = F$

- 将  $G$  中每一个形如  $X \rightarrow (A_1, A_2, \dots, A_n)$  的函数依赖替换为如下一组依赖因素为单个属性的函数依赖： $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$

### 2. 对 $G$ 中的每一个函数依赖 $X \rightarrow A$ 作如下的处理：

- 对决定因素  $X$  中的每一个属性  $B$  作如下处理：
  - 1) 计算属性集的闭包  $(X - B)_G^+$ ；
  - 2) 如果  $A \in (X - B)_G^+$ ，则用新的函数依赖  $(X - B) \rightarrow A$  替换原来的函数依赖  $X \rightarrow A$ ；

### 3. 对 $G$ 中的每一个函数依赖 $X \rightarrow A$ 作如下处理：

- 1) 令  $N = G - \{X \rightarrow A\}$ ；
- 2) 计算属性集的闭包  $X_N^+$ ；
- 3) 如果  $A \in X_N^+$ ，那么从  $G$  中删去函数依赖  $X \rightarrow A$ ；

### 4. 将 $G$ 中每一组形如 $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ (决定因素相同) 的函数依赖合并为一个函数依赖： $X \rightarrow (A_1, A_2, \dots, A_n)$

## [补充算法2] 候选码的计算 (1)

□ 如何寻找一个关系模式 $R(U, F)$ 的候选码？

方法一：运用Armstrong公理系统做如下推导： $F \models K \xrightarrow{f} U$

【困难】依赖于个人经验和对推理规则的熟练使用。

方法二：运用属性集闭包的概念，寻找满足条件 $(K_F^+ == U)$ 的最小属性集合 $K$

【优点】有算法支持（属性集闭包计算 & 候选码计算）

【缺点】计算工作量大

方法三：运用极小函数依赖集来优化方法二中的候选码计算。

## [补充算法2] 候选码的计算 (2)

- 设有关系模式 $R(U, F)$ ,  $U$ 是关系 $R$ 的属性集合,  $F$ 是关系上的极小函数依赖集。根据 $F$ 中的函数依赖, 可以将属性集合 $U$ 划分为以下的三个子集:
  1. 只在函数依赖的左边出现过的属性的集合  $U_L$  (包括没有出现在任何函数依赖中的属性)
  2. 只在函数依赖的右边出现过的属性的集合  $U_R$
  3. 在两边都出现过的属性的集合  $U_A$
- 其中:
  - $U_L$  中的属性是每一个候选码的组成部分;
  - $U_R$  中的属性不可能出现在任何一个候选码中;
  - 在候选码计算中, 只需要对  $U_A$  中的属性进行FOR循环检查。

## [补充算法2] 候选码的计算 (3)

输入：关系  $R$  的属性集合  $U$ ，极小函数依赖集  $F$

只在函数依赖的右边出现过的属性集合  $U_R$

在函数依赖的左右两边都出现过的属性集合  $U_A$

输出：候选码  $K$

```
set  $K := U - U_R$  ;
```

```
for each attribute  $A$  in  $U_A$ 
```

```
{
```

```
    compute  $(K - A)_F^+$  ;
```

```
    if  $(K - A)_F^+$  contains all the attributes in  $R$ 
```

```
    then set  $K := K - A$  ;
```

```
}
```

```
return  $K$  ;
```

*/\* 对 $U_A$ 中属性的不同处理顺序，  
可能得到不同的计算结果 $K$  \*/*

# 计算候选码 (1)

□ [例1] 寻找下述关系模式的候选码

$R(A, B, C)$   $F: \{A \rightarrow B, B \rightarrow A, A \rightarrow C\};$

解1:  $K = \{A, B, C\}$

$$\because (K-A)_F^+ = \{B, C\}_F^+ = \{A, B, C\} = U \quad \therefore K = K-A = \{B, C\}$$

$$\because (K-B)_F^+ = \{C\}_F^+ = \{C\} \neq U \quad \therefore \text{不能再从}\{B, C\}\text{中去掉属性}B$$

$$\because (K-C)_F^+ = \{B\}_F^+ = \{A, B, C\} = U \quad \therefore K = K-C = \{B\}$$

最后得到该关系的一个候选码  $\{B\}$

解2:  $K = \{A, B, C\}$

$$\because (K-B)_F^+ = \{A, C\}_F^+ = \{A, B, C\} = U \quad \therefore K = K-B = \{A, C\}$$

$$\because (K-A)_F^+ = \{C\}_F^+ = \{C\} \neq U \quad \therefore \text{不能再从}\{A, C\}\text{中去掉属性}A$$

$$\because (K-C)_F^+ = \{A\}_F^+ = \{A, B, C\} = U \quad \therefore K = K-C = \{A\}$$

最后得到该关系的另一个候选码  $\{A\}$

5. 设有一个期末考试监考安排关系R，其中的属性有：课程的课程号(cno)和课程名(cname)，授课教师的工作证编号(tno)和姓名(tname)，监考老师的工作证编号(in\_no)，每一场考试的开始时间(s\_date)、结束时间(e\_date)和考试教室(room)。其中：课程号和工作证编号分别是课程及教师的标识属性，开始时间和结束时间是date类型（含日期和时间）的字段，并且规定：

- ① 每一门课程至少有一位授课教师，也可能安排多位授课教师；
- ② 一位老师也可以担任多门课程的授课任务；
- ③ 每一门课的期末考试只安排一场，可分在多个教室中同时进行，除了授课教师外，在每一间考试教室中都必须安排一位或多位监考老师；
- ④ 同一时间段、同一间教室中只能安排一门课程的考试；
- ⑤ 一位老师可以担任多门课程的监考任务，但在同一时间段内，一位老师只能在指定的一间教室中监考一门课；
- ⑥ 授课教师必须参加自己承担授课任务的课程监考（不限定教室）。

R(cno, cname, tno, tname, in\_no, s\_date, e\_date, room)

请找出该关系中的所有函数依赖（非平凡的完全函数依赖）；

请找出该关系上的所有候选码。

## 算法6.3&6.4：转换为3NF既有无损连接性又保持函数依赖的分解

□ 输入：关系模式  $R(U, F)$

□ 输出：满足3NF且既有无损连接性又保持函数依赖的分解  $\rho$

1) 计算F的极小函数依赖集，并用来代替F进行后续的模式分解；

2)  $\rho = \emptyset$  ; // 初始化分解  $\rho$  为空集

3) 对 F 中的每一个函数依赖  $X \rightarrow Y$  做如下处理：

➤ 如果在分解  $\rho$  中找不到满足下述条件的关系模式  $Z(U_Z, F_Z) : XY \subseteq U_Z$

➤ 则由X和Y合并构成一个新的子关系模式并加入到分解  $\rho$  中；

4) 如果关系 R 的所有候选码都没有出现在分解  $\rho$  的关系模式中，即：找不到一个原关系R的候选码K和一个关系模式  $Z(U_Z, F_Z) (Z \in \rho)$ ，满足  $K \subseteq U_Z$

那么，就从关系R中任选一个候选码K, 由K中的属性单独构成一个关系模式并加入到分解  $\rho$  中去。

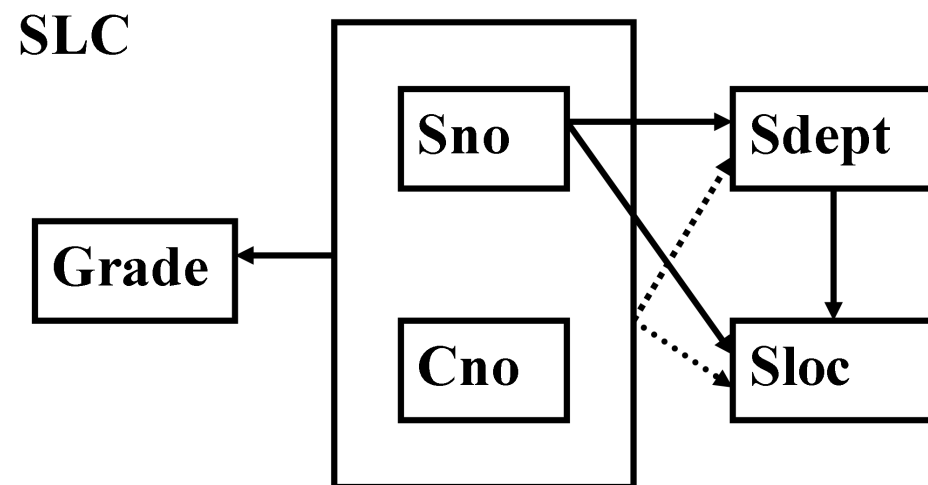


## 到3NF的模式分解示例 (1)

### □ [例6.4] SLC(Sno,Sdept,Sloc,Cno,Grade)

➤ 函数依赖有

- $(Sno, Cno) \xrightarrow{F} Grade$
- $Sno \rightarrow Sdept, (Sno, Cno) \xrightarrow{P} Sdept$
- $Sno \rightarrow Sloc, (Sno, Cno) \xrightarrow{P} Sloc$
- $Sdept \rightarrow Sloc$



- ① 计算关系**SLC**的极小函数依赖集 $F_{min}$ ;
- ② 写出关系**SLC**的所有候选码、主属性集、非主属性集;
- ③ 关系**SLC**是否满足**3NF**? 如不满足, 请给出到**3NF**且具有无损连接性和保持函数依赖的分解。



## 到3NF的模式分解示例 (2)

□ 关系  $SLC(U, F)$

$U = \{Sno, Sdept, Sloc, Cno, Grade\}$

$F = \{ (Sno, Cno) \xrightarrow{F} Grade, \\ Sno \rightarrow Sdept, \\ (Sno, Cno) \xrightarrow{P} Sdept, \\ Sno \rightarrow Sloc, \\ (Sno, Cno) \xrightarrow{P} Sloc, \\ Sdept \rightarrow Sloc \}$

① 计算关系 **SLC** 的极小函数依赖集  $F_{min}$ ;

➤ 首先消除 **F** 中的部分函数依赖, 结果是:

$F = \{ (Sno, Cno) \rightarrow Grade, Sno \rightarrow Sdept, Sno \rightarrow Sloc, Sdept \rightarrow Sloc \}$

➤ 再消除 **F** 中的冗余函数依赖, 得到 **F** 的最小覆盖  $F_{min}$ :

$F_{min} = \{ (Sno, Cno) \rightarrow Grade, Sno \rightarrow Sdept, Sdept \rightarrow Sloc \}$

## 到3NF的模式分解示例 (3)

□ 关系  $SLC(U, F_{min})$ ,  $U = \{Sno, Sdept, Sloc, Cno, Grade\}$   
 $F_{min} = \{ (Sno, Cno) \rightarrow Grade, Sno \rightarrow Sdept, Sdept \rightarrow Sloc \}$

② 写出关系 **SLC** 的所有候选码、主属性集、非主属性集;

□ 使用补充算法2, 候选码的计算过程如下:

- 只在函数依赖的右边出现过的属性集合  $U_R = \{Grade, Sloc\}$
- 在函数依赖的左右两边都出现过的属性集合  $U_A = \{Sdept\}$
- 首先排除掉属性 **Grade** 和 **Sloc**, 然后再检查属性 **Sdept** 是不是多余的, 可得到 **SLC** 的唯一一个候选码:  $(Sno, Cno)$
- 最后的结果是:
  - 候选码:  $(Sno, Cno)$
  - 主属性集:  $\{Sno, Cno\}$
  - 非主属性集:  $\{Sdept, Sloc, Grade\}$

## 到3NF的模式分解示例 (4)

- ❑ 关系  $SLC(U, F_{min})$ ,  $U = \{Sno, Sdept, Sloc, Cno, Grade\}$   
 $F_{min} = \{ (Sno, Cno) \rightarrow Grade, Sno \rightarrow Sdept, Sdept \rightarrow Sloc \}$   
候选码:  $(Sno, Cno)$   
主属性集:  $\{Sno, Cno\}$     非主属性集:  $\{Sdept, Sloc, Grade\}$

③ 关系 **SLC** 是否满足 **3NF**? 如不满足, 请给出到 **3NF** 且具有无损连接性和保持函数依赖的分解。

- ❑ 在关系 **SLC** 中, 存在非主属性对候选码的部分函数依赖和传递函数依赖, 所以  $SLC \notin 3NF$
- ❑ 调用到 **3NF** 的分解算法, 分解结果如下:  
 $SC(\{Sno, Cno, Grade\}, \{(Sno, Cno) \rightarrow Grade\})$   
 $SD(\{Sno, Sdept\}, \{Sno \rightarrow Sdept\})$   
 $DL(\{Sdept, Sloc\}, \{Sdept \rightarrow Sloc\})$

## 算法6.5：转换为BCNF的无损连接分解

- ❑ 输入：关系模式 $R(U, F)$
- ❑ 输出：到BCNF且具有无损连接性的分解 $\rho$
- ❑ 算法：
  - ① 令  $\rho = \{ R(U, F) \}$
  - ② 检查 $\rho$ 中各关系模式是否均属于BCNF。若是，则算法终止。
  - ③ 设 $\rho$ 中 $R_i(U_i, F_i) \notin BCNF$ ，那么必有  $X \rightarrow A \in F_i^+$  ( $A \notin X$ )，且 $X$ 非 $R_i$ 的候选码。对 $R_i(U_i, F_i)$ 进行如下分解：
    - $\sigma = \{S_1, S_2\}$ ,  $U_{S_1} = XA$ ,  $U_{S_2} = U_i - A$
    - 以 $\sigma$ 代替 $\rho$ 中的 $R_i(U_i, F_i)$
    - 返回步骤②。

## 到BCNF的模式分解示例

❑ [例6.8] 关系模式**STJ(S,T,J)**中，**S**表示学生，**T**表示教师，**J**表示课程。每一教师只教一门课。每门课有若干教师，某一学生选定某门课，就对应一个固定的教师。

➤ 由语义可得到函数依赖：

- $(S, J) \rightarrow T$  ,  $T \rightarrow J$

- $(S, T) \xrightarrow{P} J$

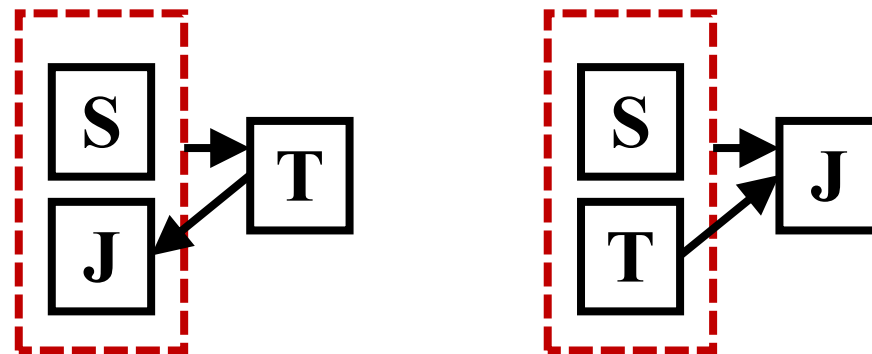
➤  $STJ \in 3NF$  但  $STJ \notin BCNF$

➤ 调用算法6.5，到BCNF的分解结果是：

- $TJ(\{T, J\}, \{T \rightarrow J\})$

- $ST(\{S, T\}, \{\})$

➤ 分解具有无损连接性，但不保持函数依赖。



## □ [引理6.5]

- 若  $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_k(U_k, F_k)\}$  是  $R(U, F)$  的一个无损连接分解
- $\sigma = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m\}$  是  $\rho$  中  $R_i(U_i, F_i)$  的一个无损连接分解
- 那么  $\rho' = \{R_1, \dots, R_{i-1}, \mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_m, R_{i+1}, \dots, R_k\}$  也是  $R(U, F)$  的无损连接分解。

□ [引理6.6]  $(R_1 \bowtie R_2) \bowtie R_3 = R_1 \bowtie (R_2 \bowtie R_3)$

# 关系规范化设计综合示例 (1)

`emps(emp_id, emp_name, emp_phone, dept_name, emp_cityst, emp_straddr, emp_zip)`

❑ 职工**emps**关系中各属性的语义如下：

- 每一位员工只有一个姓名**emp\_name**、电话**emp\_phone**和通讯地址，通讯地址由城市**emp\_cityst**、街道地址**emp\_straddr**、邮政编码**emp\_zip**组成；
- **emp\_cityst**属性值具体到省(州)和城市名称，确保每一个**emp\_cityst**值对应着唯一的一座城市，以解决不同地区的城市同名问题；
- 在不同城市中，可能存在同名的街道；同一条街道上的不同区域，可能对应不同的邮政编码；当城市和街道地址确定时，对应的邮政编码唯一确定；
- 在同一座城市中，不同的区域使用不同的邮政编码；每一个邮政编码，只能对应着唯一一座城市中的某个区域；
- 每一位员工只就职于一个部门**dept\_name**。

## 关系规范化设计综合示例 (2)

$\text{emps}(\text{emp\_id}, \text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}, \text{emp\_zip})$

□ 根据对上述语义信息的分析，初步得到该关系上的函数依赖集如下：

- ①  $\text{emp\_id} \rightarrow \{\text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}, \text{emp\_zip}\}$
- ②  $\{\text{emp\_cityst}, \text{emp\_straddr}\} \rightarrow \text{emp\_zip}$
- ③  $\text{emp\_zip} \rightarrow \text{emp\_cityst}$

□ 按照最小覆盖的要求进行检查。在函数依赖①中，**emp\_cityst** 和 **emp\_zip** 只要保留一个即可。

□ 保留**emp\_cityst**，计算得到的函数依赖集最小覆盖如下：

- ①  $\text{emp\_id} \rightarrow \{\text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}\}$
- ②  $\{\text{emp\_cityst}, \text{emp\_straddr}\} \rightarrow \text{emp\_zip}$
- ③  $\text{emp\_zip} \rightarrow \text{emp\_cityst}$



## 关系规范化设计综合示例 (3)

`emps(emp_id, emp_name, emp_phone, dept_name, emp_cityst, emp_straddr, emp_zip)`

❑ 保留 **emp\_cityst**, 计算得到的函数依赖集最小覆盖如下:

①  $\text{emp\_id} \rightarrow \{\text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}\}$

②  $\{\text{emp\_cityst}, \text{emp\_straddr}\} \rightarrow \text{emp\_zip}$

③  $\text{emp\_zip} \rightarrow \text{emp\_cityst}$

❑ 对 **emps** 关系进行规范化设计检查, 发现  $\text{emps} \notin 3\text{NF}$  and  $\text{emps} \notin \text{BCNF}$ .

❑ 理由如下:

➤ 候选码: **emp\_id**

➤ 函数依赖⑤和⑥既不符合 **3NF** 定义, 也不符合 **BCNF** 定义。

## 关系规范化设计综合示例 (4)

`emps(emp_id, emp_name, emp_phone, dept_name, emp_cityst, emp_straddr, emp_zip)`

□ 首先将关系 **emps** 分解到满足 **3NF**，且分解满足‘无损联接’和‘依赖保持’：

➤ `emps(emp_id, emp_name, emp_phone, dept_name, emp_cityst, emp_straddr)`

①  $\text{emp\_id} \rightarrow \{\text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}\}$

➤ `empadds(emp_cityst, emp_straddr, emp_zip)`

②  $\{\text{emp\_cityst}, \text{emp\_straddr}\} \rightarrow \text{emp\_zip}$

③  $\text{emp\_zip} \rightarrow \text{emp\_cityst}$

## 关系规范化设计综合示例 (5)

❑ `emps(emp_id, emp_name, emp_phone, dept_name, emp_cityst, emp_straddr)`

①  $\text{emp\_id} \rightarrow \{\text{emp\_name}, \text{emp\_phone}, \text{dept\_name}, \text{emp\_cityst}, \text{emp\_straddr}\}$

❑ `empadds(emp_cityst, emp_straddr, emp_zip)`

②  $\{\text{emp\_cityst}, \text{emp\_straddr}\} \rightarrow \text{emp\_zip}$

③  $\text{emp\_zip} \rightarrow \text{emp\_cityst}$

❑ 对分解后的两个子关系，分别用**3NF**和**BCNF**的定义进行检查，发现：

➤  $\text{emps} \in 3\text{NF}$  **and**  $\text{emps} \in \text{BCNF}$

➤  $\text{empadds} \in 3\text{NF}$  **and**  $\text{empadds} \notin \text{BCNF}$

## 关系规范化设计综合示例 (6)

❑ empadds(emp\_cityst, emp\_straddr, emp\_zip)

② {emp\_cityst, emp\_straddr} → emp\_zip

③ emp\_zip → emp\_cityst

❑ 对关系 empadds 作出判断的理由如下：

➤ 候选码：{emp\_cityst, emp\_straddr} 和 {emp\_zip, emp\_straddr}

➤ 函数依赖符合 3NF 的定义（不存在非主属性）

➤ 函数依赖 ③ 不符合 BCNF 定义。

❑ 可继续将 empadds 分解成为如下的两个子关系，结果关系满足 BCNF，分解满足‘无损联接’，但不满足‘保持函数依赖’：

➤ zipstr(emp\_zip, emp\_straddr)

➤ zipcit(emp\_zip, emp\_cityst)

③ emp\_zip → emp\_cityst

## 复习思考题 (1)

14. 给定关系模式 $R(A,B,C,D,E,F)$ 及其上的函数依赖集:

$$S = \{ A \rightarrow E, B \rightarrow ADE, DF \rightarrow AC, ADF \rightarrow B \}$$

- ① 请直接写出与 $S$ 等价的最小函数依赖集;
- ② 请直接写出关系模式 $R$ 的所有候选码、主属性集、非主属性集;
- ③ 请使用3NF模式分解算法对关系 $R$ 进行分解, 并满足无损联接性和依赖保持性;
- ④ 上题分解结果是否满足BCNF? 如果不满足, 请将其继续分解到满足BCNF并说明理由。

15. 给定关系 $R(A,B,C,D,E,F,G)$ 及其上的函数依赖集: (不需要写计算过程)

$$M = \{ ABC \rightarrow DEF, AC \rightarrow BG, D \rightarrow F, E \rightarrow GC \}$$

- ① 请直接写出与 $M$ 等价的最小函数依赖集;
- ② 请直接写出关系 $R$ 的所有候选码、主属性集、非主属性集;
- ③ 请使用3NF模式分解算法对关系 $R$ 进行分解, 并满足无损联接性和依赖保持性;
- ④ 上述的分解结果是否满足BCNF? 如满足BCNF, 请简单说明理由; 否则, 请将其继续分解到满足BCNF。

## 复习思考题 (2)

16. 设有一个机场跑道使用调度管理系统，其关系模式如下：

机场编号 跑道编号 飞机编号 使用开始时间 使用结束时间  
R ( ano, lno, pno, s\_time, e\_time )

其中：

- 机场编号ano和飞机编号pno分别是机场和飞机的标识属性；
- 在一个机场中，可能有多条用于飞机起降的跑道，每条跑道都有一个唯一的编号lno；分属于不同机场的跑道，可能有相同的跑道编号；
- 每一条跑道每次只能供一架飞机使用（供飞机起降用）；
- 使用开始时间和使用结束时间的数据类型是时间戳(timestamp)。

- ① 根据上述描述，请写出关系R上的最小函数依赖集。(不需要写计算过程)
- ② 关系R最高能够满足哪个范式的定义？请简单说明理由。

## 复习思考题 (3)

17. 设有一个大学生创新项目管理关系**P**，其属性包括：项目的编号**pno**、执行年份**pyear**、验收等级**ps**，项目负责学生的学号**mgrno**，项目参与学生的学号**sno**。

其中：①项目编号和学号分别是项目和学生的标识属性；②每个项目都有唯一的一名负责的学生，以及可能的若干名参与的学生；③每个项目的执行周期只有一年；④一个学生可以负责或参与过若干个项目，但每一年最多只能负责或参与一个项目。

- ① 请写出该关系上的最小函数依赖集（直接写出结果）
- ② 该关系最高能够满足到第几范式？请简单说明理由；
- ③ 关系**P**是否满足3NF？如不满足，请用到3NF的模式分解算法直接对其进行模式分解；
- ④ 上述分解结果是否满足BCNF？如不满足，请将其进一步分解到满足BCNF。