



2025年春季学期

数据库系统概论

An Introduction to Database Systems

第六章 关系数据理论

中国科学技术大学

人工智能与数据科学学院

黄振亚, huangzhy@ustc.edu.cn



关于闭包的引理

147

□ 引理6.2（引理6.1得出）

设 F 为属性集 U 上的一组函数依赖， $X, Y \subseteq U$ ， $X \rightarrow Y$ 能由 F 根据Armstrong公理导出的充分必要条件是 $Y \subseteq X_F^+$

□ 引理6.2的用途

- 1. 判定 $X \rightarrow Y$ 是否能由 F 根据Armstrong公理导出的问题，转化为求出 X_F^+ 、判定 Y 是否为 X_F^+ 的子集的问题
- 2. 如果 $X_F^+ = U$ ，则 X 是 $R \langle U, F \rangle$ 的候选码
- X_F^+ 可以用算法求得



求属性集 X 关于 F 的闭包 X_F^+ 的算法

148

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^+

迭代

输入: X, F

输出: X_F^+

步骤:

对 $X^{(i)}$ 中的每个元素, 依次检查相应的函数依赖, 将依赖它的属性加入 B

(1) 令 $X^{(0)} = X, i=0$

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

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

(4) 判断 $X^{(i+1)} = X^{(i)}$ 吗?

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

(6) 若否, 则 $i=i+1$, 返回第(2)步。



函数依赖闭包

149

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

[例6.11] 已知关系模式 $R\langle U, F \rangle$ ，其中

$U = \{A, B, C, D, E\}$;

$F = \{AB \rightarrow C, B \rightarrow D, C \rightarrow E, EC \rightarrow B, AC \rightarrow B\}$ 。

求 $(AB)_F^+$ 。

解 设 $X^{(0)} = AB$;

(1) $X^{(1)} = AB \cup CD = ABCD$ 。

(2) $X^{(0)} \neq X^{(1)}$

$X^{(2)} = X^{(1)} \cup BE = ABCDE$ 。

(3) $X^{(2)} = U$ ，算法终止

$\rightarrow (AB)_F^+ = ABCDE$ 。

计算 $X^{(1)}$ ：逐一的扫描 F 集合中各个函数依赖，找左部为 A 、 B 或 AB 的函数依赖。得到两个： $AB \rightarrow C$ ， $B \rightarrow D$
于是 $X^{(1)} = AB \cup CD = ABCD$ 。

找出左部为 $ABCD$ 子集的那些函数依赖，又得到 $C \rightarrow E$ ， $AC \rightarrow B$ ，于是
 $X^{(2)} = X^{(1)} \cup BE = ABCDE$



算法6.1

150

- 算法6.1 一定会终止
- 对于算法6.1, 令 $a_i = |X^{(i)}|$, $\{a_i\}$ 形成一个步长大于1的严格递增的序列, 序列的上界是 $|U|$, 因此该算法最多 $|U| - |X|$ 次循环就会终止。

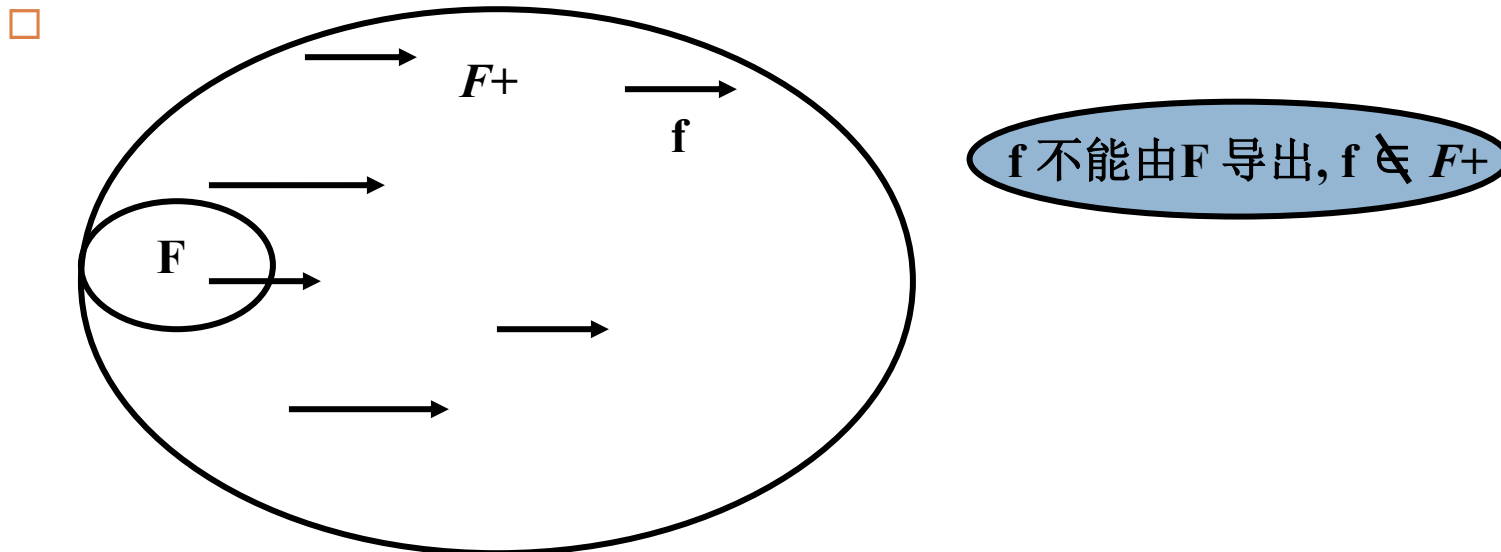


4. Armstrong公理系统的有效性与完备性

152

□ 定理6.2 Armstrong公理系统 ($A1, A2, A3$) 是有效的、完备的

□ 明确：1. 公理系统推导出来的 f 正确？
2. F^+ 中的每一个 f 都能推导出来？





4. Armstrong公理系统的有效性与完备性

153

- 定理6.2 Armstrong公理系统 $(A1, A2, A3)$ 是有效的、完备的
- 证明（了解）：
 1. 有效性：由 F 出发根据Armstrong公理推导出来的每一个函数依赖一定在 F^+ 中
可由定理6.1得证
定理6.1：Armstrong推理规则是正确的
 2. 完备性： F^+ 中的每一个函数依赖，必定可以由 F 出发根据Armstrong公理推导出来
完备性：所有不能用Armstrong公理推导出来 f , 都不为真
若 f 不能用Armstrong公理推导出来, $f \notin F^+$



4. Armstrong公理系统的有效性与完备性

154

□ 证明**完备性**： F^+ 中的每一个函数依赖，必定可以由 F 出发根据Armstrong公理推导出来

□ 只需证明**逆否命题**：若函数依赖 $X \rightarrow Y$ 不能由 F 从Armstrong公理导出，那么它必然不为 F 所蕴含

□ 证明分3步

□ **(1)** 若 $V \rightarrow W$ 成立，且 $V \subseteq X_F^+$ ，则 $W \subseteq X_F^+$

证 因为 $V \subseteq X_F^+$ ，所以有 $X \rightarrow V$ 成立；（ ? ）

因为 $X \rightarrow V$ ， $V \rightarrow W$ ，于是 $X \rightarrow W$ 成立

所以 $W \subseteq X_F^+$

□ 引理6.2 （引理6.1得出）

设 F 为属性集 U 上的一组函数依赖， $X, Y \subseteq U$ ， $X \rightarrow Y$ 能由 F 根据Armstrong公理导出的充分必要条件是 $Y \subseteq X_F^+$



4. Armstrong公理系统的有效性与完备性

155

- 证明完备性逆否命题: 若函数依赖 $X \rightarrow Y$ 不能由 F 从Armstrong公理导出, 那么它必然不为 F 所蕴含
 - (2) 构造一张二维表 r , 它由下列两个元组构成, 可证明 r 必是 $R\langle U, F \rangle$ 的一个关系, 即 F 中的全部函数依赖在 r 上成立

$$\begin{array}{cc} \underbrace{X_F^+} & \underbrace{U-X_F^+} \\ 11\dots\dots 1 & 00\dots\dots 0 \\ 11\dots\dots 1 & 11\dots\dots 1 \end{array}$$

若 r 不是 $R\langle U, F \rangle$ 的关系, 则必由于 F 中有某一个函数依赖 $V \rightarrow W$ 在 r 上不成立所致。由 r 的构成可知, V 必定是 X_F^+ 的子集, 而 W 不是 X_F^+ 的子集, 可是由第(1)步, $W \subseteq X_F^+$, 矛盾。

所以 r 必是 $R\langle U, F \rangle$ 的一个关系。



4. Armstrong公理系统的有效性与完备性

156

- 证明完备性逆否命题: 若函数依赖 $X \rightarrow Y$ 不能由 F 从Armstrong公理导出, 那么它必然不为 F 所蕴含
 - (3) 若 $X \rightarrow Y$ 不能由 F 从Armstrong公理导出, 则 Y 不是 XF^+ 的子集。(引理6.2)
 - 因此必有 Y 的子集 Y' 满足 $Y' \subseteq U - XF^+$,
 - 则 $X \rightarrow Y$ 在 r 中不成立,
 - 即 $X \rightarrow Y$ 必不为 $R \langle U, F \rangle$ 蕴含
- /* 因为 F^+ 中的全部函数依赖在 r 上成立。

□ 引理6.2 (引理6.1得出)

设 F 为属性集 U 上的一组函数依赖, $X, Y \subseteq U$, $X \rightarrow Y$ 能由 F 根据Armstrong公理导出的充分必要条件是 $Y \subseteq X_F^+$



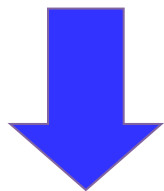
Armstrong公理系统完备性证明

157

□ Armstrong公理的完备性及有效性说明:

□ “导出” 与“蕴涵” 是两个完全等价的概念

□ F^+ : 为 F 所逻辑蕴涵的函数依赖的全体 (定义6.12)



定义6.12 在关系模式 $R<U, F>$ 中为 F 所逻辑蕴含的函数依赖的全体叫作 F 的闭包, 记为 F^+ 。

□ F^+ : 可以说成由 F 出发借助Armstrong公理导出的函数依赖的集合



5. 函数依赖集等价

□ 引理6.2 (引理6.1得出)

设 F 为属性集 U 上的一组函数依赖, $X, Y \subseteq U$, $X \rightarrow Y$ 能由 F 根据Armstrong公理导出的充分必要条件是 $Y \subseteq X_F^+$

定义6.14 如果 $G^+ = F^+$, 就说函数依赖集 F 覆盖 G (F 是 G 的覆盖, 或 G 是 F 的覆盖), 或 F 与 G 等价。

□ 两个函数依赖集等价是指它们的闭包等价

引理6.3 $F^+ = G^+$ 的充分必要条件是 $F \subseteq G^+$ 和 $G \subseteq F^+$

证明: 必要性显然: 因为 $F \subseteq F^+$, $F^+ = G^+$, 所以 $F \subseteq G^+$

充分性:

(1) 若 $F \subseteq G^+$, 则 $X_F^+ \subseteq X_{G^+}^+$

(2) 任取 $X \rightarrow Y \in F^+$ 则有 $Y \subseteq X_F^+ \subseteq X_{G^+}^+$

所以 $X \rightarrow Y \in (G^+)^+ = G^+$ 。即 $F^+ \subseteq G^+$

(3) 同理可证 $G^+ \subseteq F^+$, 所以 $F^+ = G^+$

如何判定 $F \subseteq G^+$?

只需逐一对 F 中的函数依赖 $X \rightarrow Y$ 考察 Y 是否属于 $X_{G^+}^+$



6. 最小依赖集

159

亦称为极小函数依赖集或最小覆盖

定义6.15 如果函数依赖集F满足下列条件，
则称F为一个最小依赖集。

- (1) F中任一函数依赖的右部仅含有一个属性
- (2) F中不存在这样的函数依赖 $X \rightarrow A$ ，使得F与 $F - \{X \rightarrow A\}$ 等价
- (3) F中不存在这样的函数依赖 $X \rightarrow A$ ， X 有真子集 Z 使得 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与F等价

即F中的函数依赖均不能由F中其他函数依赖导出

F中各函数依赖左部均为最小属性集（ X 不存在冗余属性）



最小依赖集

160

[例6.12] 关系模式 $S\langle U, F \rangle$, 其中:

$U = \{ \text{Sno}, \text{School}, \text{Mname}, \text{Cno}, \text{Grade} \},$

$F = \{ \text{Sno} \rightarrow \text{School}, \text{School} \rightarrow \text{Mname}, (\text{Sno}, \text{Cno}) \rightarrow \text{Grade} \}$

设 $F' = \{ \text{Sno} \rightarrow \text{School}, \text{Sno} \rightarrow \text{Mname}, \text{School} \rightarrow \text{Mname},$

$(\text{Sno}, \text{Cno}) \rightarrow \text{Grade}, (\text{Sno}, \text{School}) \rightarrow \text{School} \}$

F 是最小覆盖, 而 F' 不是。

因为: $F' - \{ \text{Sno} \rightarrow \text{Mname} \}$ 与 F' 等价

$F' - \{ (\text{Sno}, \text{School}) \rightarrow \text{School} \}$ 也与 F' 等价



7. 极小化过程

161

定理6.3 每一个函数依赖集 F 均等价于一个极小函数依赖集 F_m 。此 F_m 称为 F 的最小依赖集。

证明：构造性证明，找出 F 的一个最小依赖集。

极小化过程(定理6.3的证明)也是检验 F 是否为极小依赖集的一个算法



极小化过程（续）

162

证明（了解），但需要掌握构造求解方法：

(1) 逐一检查 F 中各函数依赖 $FD_i: X \rightarrow Y$ ，若 $Y = A_1 A_2 \dots A_k$ ， $k > 2$ ，

则用 $\{X \rightarrow A_j \mid j=1, 2, \dots, k\}$ 来取代 $X \rightarrow Y$ 。

引理6.1 $X \rightarrow A_1 A_2 \dots A_k$ 成立的充分必要条件是 $X \rightarrow A_i$ 成立（ $i=1, 2, \dots, k$ ）。

(2) 逐一检查 F 中各函数依赖 $FD_i: X \rightarrow A$ ，令 $G = F - \{X \rightarrow A\}$ ，

若 $A \in X_G^+$ ，则从 F 中去掉此函数依赖。

引理6.2

设 F 为属性集 U 上的一组函数依赖， $X, Y \subseteq U$ ， $X \rightarrow Y$ 能由 F 根据Armstrong公理导出的充分必要条件是 $Y \subseteq X_F^+$

(3) 逐一取出 F 中各函数依赖 $FD_i: X \rightarrow A$ ，设 $X = B_1 B_2 \dots B_m$ ，

逐一考查 B_i （ $i=1, 2, \dots, m$ ），若 $A \in (X - B_i)_F^+$ ，

则以 $X - B_i$ 取代 X 。

定义6.15 (3) F 中不存在这样的函数依赖 $X \rightarrow A$ ， X 有真子集 Z 使得 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与 F 等价。



例题一求F的最小函数依赖集

163

- $R\langle U, F\rangle$, $U=ABCD$,
- 函数依赖集 $F = \{A \rightarrow BD, AB \rightarrow C, C \rightarrow D\}$ 。
- 求：F最小函数依赖集
- 解法步骤：
 - 1. 将F中的所有函数依赖的右边化为单一属性
 - 2. 去掉F中的所有函数依赖左边的冗余属性
 - 3. 去掉F中所有冗余的函数依赖



例题图解 (1)

164

- (1) 将F中的所有函数依赖右边化为单一属性

$$F = \{A \rightarrow BD, AB \rightarrow C, C \rightarrow D\}$$

$$\Rightarrow F = \{A \rightarrow B, A \rightarrow D, AB \rightarrow C, C \rightarrow D\}$$



例题图解 (2)

165

- (2) 去掉F中的所有函数依赖左边的冗余属性

$$F = \{A \rightarrow B, A \rightarrow D, \mathbf{AB} \rightarrow \mathbf{C}, C \rightarrow D\}$$

$$\Rightarrow A^+ = \{A, B, \mathbf{C}, D\} \quad B^+ = \{B\}$$

$$\Rightarrow F = \{A \rightarrow B, A \rightarrow D, A \rightarrow C, C \rightarrow D\}$$



例题图解 (3)

166

- (3) 去掉F中所有冗余的函数依赖关系

$$F = \{ A \rightarrow B, \boxed{A \rightarrow D}, A \rightarrow C, C \rightarrow D \}$$

➡ $F = \{ A \rightarrow B, \quad , A \rightarrow C, C \rightarrow D \}$

➡ $A^+ = \{A, B, C, D\}$

➡ $F_m = \{ A \rightarrow B, A \rightarrow C, C \rightarrow D \}$



极小化过程（续）

167

F的最小依赖集 F_m 是否唯一？

[例6.13] $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A\}$

F_{m1} 、 F_{m2} 都是 F 的最小依赖集：

$$F_{m1} = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$$

$$F_{m2} = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A\}$$

- F 的最小依赖集 F_m 不唯一，它与对各函数依赖 FD_i 及 $X \rightarrow A$ 中 X 各属性的处置的顺序有关



候选码的求解(补充)

- 候选码如何求解？
 - 如果 $XF^+ = U$ ，则 X 是 $R<U, F>$ 的候选码
- 给定的关系 $R(A_1, A_2, \dots, A_n)$ 和函数依赖集 F ，可以将其属性分为4类：
 - L类：仅出现在 F 的函数依赖左边的属性
 - R类：仅出现在 F 的函数依赖右边的属性
 - N类：在 F 的函数依赖左右两边均未出现的属性
 - LR类：在 F 的函数依赖左右两边均出现的属性



候选码的求解(补充)

快速求解候选码

- 定理：对于给定的关系模式 $R(U, F)$ ，若 X （ $X \in R$ ）是L类属性，则 X 必为 R 的任一候选码的成员
- 定理：对于给定的关系模式 $R(U, F)$ ，如果 X （ $X \in R$ ）是R类属性，则 X 不在任何候选码中
- 定理：设有关系模式 $R(U, F)$ ， X （ $X \in R$ ）是N类属性，则 X 必包含在 R 的任一候选码中
 - 推论：对于给定的关系模式 $R(U, F)$ ，如果 X 是 R 的N类和L类组成的属性集，且 X_F^+ 包含了 R 的全部属性 U ，则 X 是 R 的唯一候选码



候选码的求解(补充)

- 例：关系模式 $R(A, B, C, D, E, P)$ ，函数依赖集 $F = \{A \rightarrow D, E \rightarrow D, D \rightarrow B, BC \rightarrow D, DC \rightarrow A\}$

- 解：
 - 1. F 中， C 、 E 均是 L 类属性，则 C 、 E 在 R 的任何候选码中。
 - 2. P 是 N 类属性，则 P 在 R 的任何候选码中
 - 3. $(CEP)^+ = ABCDEP$ ，则 CEP 是 R 的候选码



Armstrong公理系统—总结

171

- 问题引入：判断 $R\langle U, F \rangle$ 的好坏的关键是找到部分F，传递F
- 目标需求：找出 $R\langle U, F \rangle$ 的全部函数依赖 F_R^+
 - 但，直接通过语义找到F很难
 - 因此，建立一套推理系统，给定 $R\langle U, F \rangle$ 和已知的F，推导所有 F_R^+
- Armstrong公理系统引入
 - 第一步：给出逻辑蕴含F，
 - 第二步：函数依赖闭包 F^+
 - 第三步：Armstrong公理系统的导出规则。证明 $F_R^+ = F^+$
 - 第四步：如何方便的求出 F^+ ：求属性集X ($X \subseteq U$) 关于F的闭包 X_F^+
 - 第五步：如何计算 X_F^+ （算法6.1）？如何求R的码？
 - Armstrong公理系统的有效性与完备性
 - 函数依赖集等价的概念
 - 最小依赖集或最小覆盖



第六章 关系数据理论

172

6.1 问题的提出

6.2 规范化

6.3 数据依赖的公理系统

*6.4 模式的分解

6.5 小结



6.4 模式的分解

173

- 6.4.1 模式分解的3个定义
- 6.4.2 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 模式的分解

174

- 关系模式的规范化过程是通过对关系模式的分解来实现的
- 什么是模式分解

(定义6.16) 关系模式 $R\langle U, F \rangle$ 的一个分解是指: $\rho = \{ R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle \}$, 其中 $U = U_1 \cup U_2 \cup \dots \cup U_i$, 且不存在 $U_i \subseteq U_j$, F_i 为 F 在 U_i 上的投影

- 相应地, 将 R 存储在二维表 r 中的数据分散到二维表 $r_1, r_2, \dots, r_n$ 中去, 其中 r_i 是 r 在属性集 U_i 上的投影

定义6.17 函数依赖集合 $\{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq U_i\}$ 的一个覆盖 F_i 叫作 F 在属性 U_i 上的投影



关系模式分解的标准

175

- 把低一级的关系模式分解为若干个高一级的关系模式的方法不是唯一的
- 只有能够保证分解后的关系模式与原关系模式等价，分解方法才有意义
- 三种模式分解等价的定义：
 1. 分解具有无损连接性
 2. 分解要保持函数依赖
 3. 分解既要保持函数依赖，又要具有无损连接性(这三个定义是实行模式分解的三条准则)



例子

176

例子：S-L(Sno, Sdept, Sloc) 有下列函数依赖：

$$F = \{ \text{Sno} \rightarrow \text{Sdept}, \\ \text{Sdept} \rightarrow \text{Sloc},$$
$$\text{Sno} \xrightarrow{\text{传递}} \text{Sloc} \}$$

- 已知 $S-L \in 2NF$ ，该关系模式存在非主属性对码的传递函数依赖
- 存在插入异常、删除异常、数据冗余度大和修改复杂的问题。需要分解该关系模式，使成为更高范式的关系模式
- 下面**对4种不同的分解**情况进行分析，引出模式分解的3个定义



例子：第一种分解情况

177

例：S-L(Sno, Sdept, Sloc) $F=\{Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc\}$

□ 分解方法一. S-L分解为三个关系模式：

SN(Sno)、SD(Sdept)、SO(Sloc)

S-L			分解	SN		SD	SO
Sno	Sdept	Sloc		Sno	Sdept	Sloc	Sloc
95001	CS	A		95001	CS	A	
95002	IS	B		95002	IS	B	
95003	MA	C		95003	MA	C	
95004	IS	B		95004	PH		
95005	PH	B		95005			

- 分解后的关系，无法查询95001学生 所在的系 或 所住的宿舍了！
- 这种分解是不可取的。丢失了许多信息，丢失了数据之间联系的信息



例子：第二种分解情况

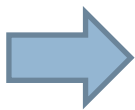
178

例：S-L(Sno, Sdept, Sloc) F={Sno→Sdept, Sdept→Sloc, Sno→Sloc}

□ 分解方法二：SL分解为下面二个关系模式：

NL(Sno, Sloc), DL(Sdept, Sloc)

S-L		
Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B



NL	
Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

DL	
Sdept	Sloc
CS	A
IS	B
MA	C
PH	B



例子：第二种分解情况（续）

179

□ 对NL和DL关系 进行自然连接的结果为

ND ⋈ DL			S-L		
Sno	Sloc	Sloc	Sno	Sdept	Sloc
95001	A	CS	95001	CS	A
95002	B	IS	95002	IS	B
95002	B	PH	95003	MA	C
95003	C	MA	95004	IS	B
95004	B	IS	95005	PH	B
95004	B	PH			
95005	B	IS			
95005	B	PH			

自然连接比原来的S-L关系多了三个元组，因此，也无法知道原来的S-L关系中究竟有哪些元组，从这个意义上说，此分解仍然丢失了信息



例子：第三种分解情况

180

例：S-L(Sno, Sdept, Sloc) $F=\{Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc\}$

□ 分解方法三：将SL分解为下面二个关系模式：

ND(Sno, Sdept), NL(Sno, Sloc)

分解后的关系为：

对ND和NL关系进行自然连接的结果为：

ND

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

ND $\bowtie$ NL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

第三种分解情况没有丢失信息，称为分解“具有无损连接性”



例子：第三种分解情况（续）

181

- 第3种分解方法具有无损连接性，但是它存在下面的问题
 - 例如，95001学生由CS系转到IS系，ND关系的(95001, CS)元组和NL关系的(95001, A)元组必须同时进行修改，否则会破坏数据库的一致性。

ND

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

IS

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

B

原因：

- ❑ S-L中的函数依赖 $Sdept \rightarrow Sloc$ 既没有投影到关系模式ND上，也没有投影到关系模式NL上。
- ❑ 这种分解没有保持原关系模式中的函数依赖



例子：第四种分解情况

182

例：S-L(Sno, Sdept, Sloc) $F=\{Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc\}$

□ 分解方法四：将SL分解为下面二个关系模式：

ND(Sno, Sdept) , $Sno \rightarrow Sdept$

DL(Sdept, Sloc) , $Sdept \rightarrow Sloc$

这种分解方法就保持了函数依赖，称为“具有保持函数依赖性”



例子：第四种分解情况（续）

继续分析第四种分解情况

ND(Sno, Sdept) , Sno→Sdept

DL(Sdept, Sloc) , Sdept→Sloc

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

ND ∝ DL		
Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

这种分解不仅 “具有保持函数依赖性” , 还 “具有无损连接性”



例子：分解情况分析

184

在给出的例子中：

- 第1种分解：既不具有无损连接性，也未保持函数依赖
 - 它不是原关系模式的一个等价分解
- 第2种分解：既不具有无损连接性，也未保持函数依赖
- 第3种分解：具有无损连接性，但未保持函数依赖
- 第4种分解：既具有无损连接性，又保持了函数依赖



6.4 模式的分解

185

- 6.4.1 模式分解的3个定义
- 6.4.2 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



1. 具有无损连接性的模式分解

186

[定义6.18]: $\rho=\{R_1, \dots, R_n\}$ 是 R 的一个分解,

□ $R\langle U, F \rangle$ 的一个分解 $\rho=\{R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle\}$

若对 R 的任何一个关系 r 均有 $\mathbf{r} = \mathbf{r}$ 在 ρ 中各关系模式上投影的自然连接成立, 则称分解 ρ 具有无损连接性。简称 ρ 为无损分解。

□ 只有具有无损连接性的分解才能够保证不丢失信息。

□ 无损连接性不一定能解决插入异常、删除异常、修改复杂、数据冗余等问题。

* 算法6.3 判别一个分解的无损连接性。 p.192



2. 保持函数依赖的模式分解

187

[定义6.17]: $\rho=\{R_1, \dots, R_n\}$ 是 R 的一个分解,

□ $R\langle U, F \rangle$ 的一个分解 $\rho=\{R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle\}$

若 F 所逻辑蕴含的函数依赖一定也为分解后所有的关系模式中的函数依赖 F_i 所逻辑蕴含, 即 $F^+ = (F_1 \cup F_2 \cup \dots \cup F_n)^+$, 则称关系模式 R 的这个分解是保持函数依赖的 (Preserve dependency)

$$F^+ = (F_1 \cup F_2 \cup \dots \cup F_n)^+$$

引理6.3 给出了判别 R 的分解是否保持函数依赖的方法。 P.188



3. 分解的无损连接性和保持函数依赖性

188

- 如果一个分解具有无损连接性，则它能够保证不丢失信息
- 如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况。
- 分解具有无损连接性和分解保持函数依赖是两个互相独立的标准。
 - 具有无损连接性的分解不一定能够保持函数依赖。
 - 保持函数依赖的分解也不一定具有无损连接性



6.4 模式的分解

189

- 6.4.1 模式分解的3个定义
- 6.4.2 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



6.4.3 模式分解的算法

190

- 算法6.2 判别一个分解的无损连接性
- 算法6.3（合成法）转换为3NF的保持函数依赖的分解。
- 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
- 算法6.5（分解法）转换为BCNF的无损连接分解
- 算法6.6 达到4NF的具有无损连接性的分解



6.4.3 模式分解的算法

191

- 按照不同的分解算法，关系模式所能达到的范式是不相同的
 - 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到BCNF
 - 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF
 - 若要求分解具有无损连接性，那么模式分解一定能够达到4NF

无损连接	保持依赖	达到的范式
	Y	3NF，不一定BCMF
Y	Y	3NF，不一定BCMF
Y		4NF



第六章 关系数据理论

192

6.1 问题的提出

6.2 规范化

6.3 数据依赖的公理系统

*6.4 模式的分解

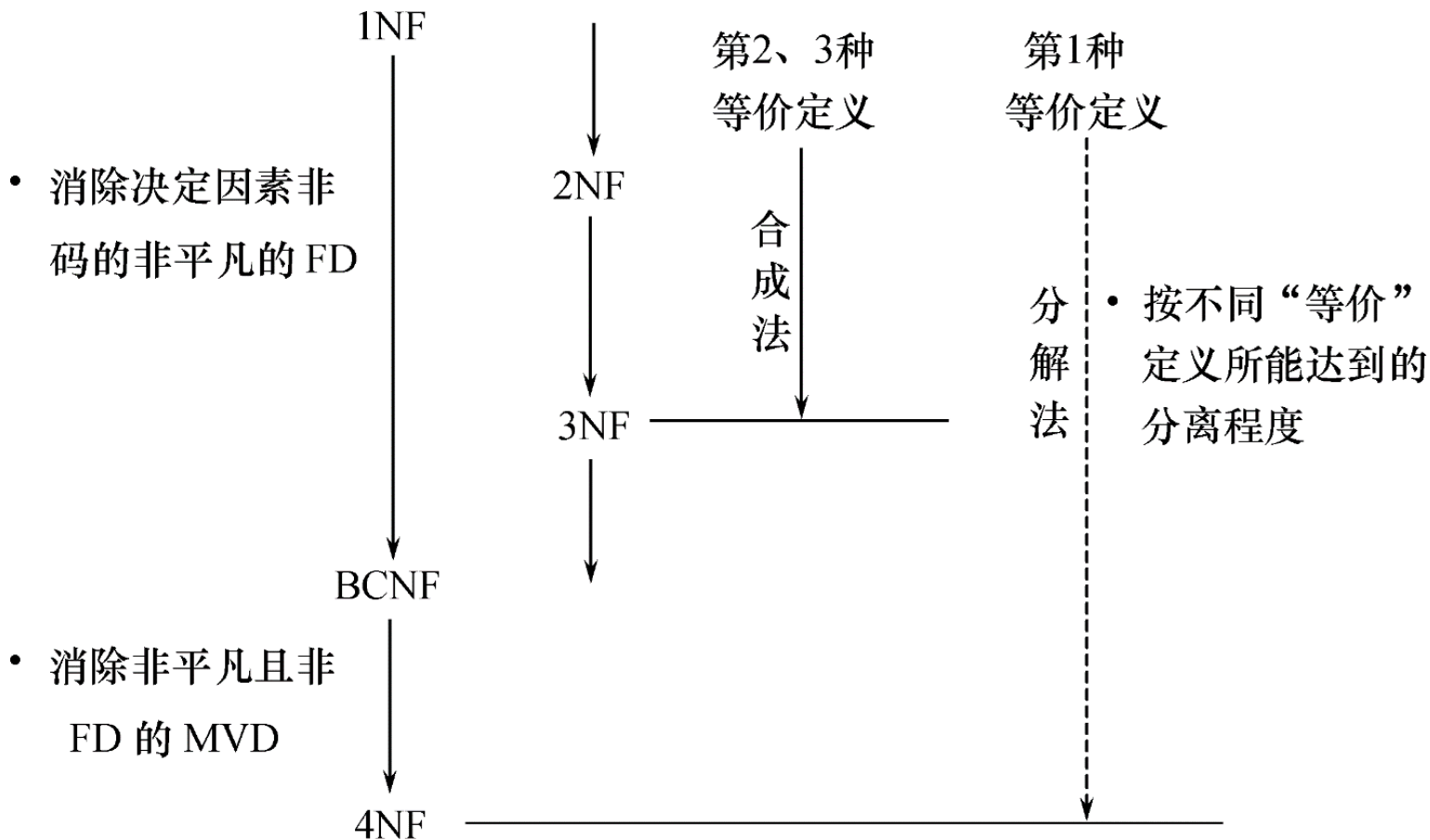
6.5 小结



6.5 小结

193

关系模式的规范化，其基本思想：





规范化小结（续）

194

□ 关系模式规范化的基本步骤

消除一些非平凡函数依赖，这些非平凡函数依赖的**决定属性集**不是码

1NF

↓ 消除非主属性对码的部分函数依赖

消除决定属性

2NF

集非码的非平

↓ 消除非主属性对码的传递函数依赖

凡函数依赖

3NF

↓ 消除主属性对码的部分和传递函数依赖

BCNF

↓ 消除非平凡且非函数依赖的多值依赖

4NF