

数据库原理

- 宋安平
- 上海大学计算机学院
- Apsong@shu.edu.cn
- 计算机学院楼1015
- 第1-4周





《数据库原理二》课堂教学（20学时)内容及安排:

| | | |
|----------|----------|-------|
| 第五章 | 规范化设计 | (8学时) |
| 第七章 | 数据库设计 | (1学时) |
| 第八章 | 数据库的管理 | (6学时) |
| 第九章 | 分布式数据库系统 | 自学 |
| 第十章 | 对象关系数据库 | (3学时) |
| 习题分析、总复习 | | (2学时) |



《数据库原理二》实验内容和安排（20学时）

实验五： 编程实施学分制教务管理信息系统（14学时）

... .. (教材P.373)

实验六： SQL Server 2000 高级技术使用（4学时）

... .. (教材P.303)

验 收： （2学时）

第5章 规范化设计

- 关系模式的设计问题
- 函数依赖
- 关系模式的分解特性
- 关系模式的范式
- 模式的进一步规范化（自学）

第一节 关系模式的设计问题

- 关系模式的外延和内涵
- 泛关系模式与数据库模式
- 关系模式的冗余和异常问题

一、关系模式的外延和内涵

- 一个关系模型包括外延和内涵两个方面的内容。
- 外延就是关系、表或当前值。
- 内涵就是对数据的定义以及对数据完整性约束的定义。

二、泛关系模式与数据库模式

- 关系模式 $R(U)$ ，关系 r 是关系模式 $R(U)$ 的当前值，是元组的集合。这样的关系模式和关系称为泛关系模式和泛关系。
- 实际使用时，用一个关系模式的集合 $\rho=\{R_1, \dots, R_k\}$ 来代替 $R(U)$ ，其中每个 R_i 是 U 的子集， ρ 称为数据库模式。

三、关系模式的冗余和异常问题

- 例 有三个属性的工资表（姓名,级别,工资）关系模式。对应此模式建立的表如下表5 — 1所示。

| 姓 名 | 级 别 | 工 资 |
|-----|-----|-----|
| A | 10 | 650 |
| B | 10 | 650 |
| C | 7 | 680 |
| D | 8 | 665 |
| E | 11 | 630 |
| F | 11 | 630 |

三、关系模式的冗余和异常问题

- 1. 数据冗余度大
- 工资是从级别推导出的,但却重复存放。数据在数据库中的重复存放称为数据冗余。
- 冗余度大,不仅浪费存储空间,重要的是在对数据进行修改时,又易造成数据的不一致性。如10级的工资变化时,如果表中有K个职工的工资为10级,就需要修改K次,一旦遗漏就使数据不一致。

三、关系模式的冗余和异常问题

- 2. 插入、修改与删除异常
- 无法插入某部分信息或删除掉不应删除的信息称为插入或删除异常。
- 例如, 9级工资为660元的信息无法插入表。因为该表的码是姓名, 而目前无职工工资级别为9级, 表中不能插入码为空值的记录。即在插入一行时, 此关系模式强迫同时增加关于两个实体的数据。
- 又如, 要删除姓名为C的职工记录时, 又将7级工资的信息一起删去了。即在删除一行时, 删除了关于两个实体的数据。

三、关系模式的冗余和异常问题

- 解决方法
- 上述现象的产生,是由于关系模式不合理。如果一个关系中,存储了两个或两个以上实体的数据,一般应将它分解为多个关系,使每个关系只有一个实体。将表 5—1 分解为两个模式表达: 职工级别 (姓名, 级别), 级别工资 (级别, 工资), 如表 5—2、 表5—3 所示。

| 姓名 | 级别 |
|----|----|
| A | 10 |
| B | 10 |
| C | 7 |
| D | 8 |
| E | 11 |
| F | 11 |

表 5 — 2 职工级别

| 级别 | 工资 |
|----|-----|
| 7 | 680 |
| 8 | 665 |
| 9 | 660 |
| 10 | 650 |
| 11 | 630 |

表 5 — 3 级别工资



第二节 函数依赖

- 函数依赖的定义
- FD的逻辑蕴涵
- 函数依赖的推理规则
- FD与关键码的联系
- 属性集的闭包
- FD的最小依赖集

一、函数依赖FD的定义

- 定义5.1 设有关系模式 $R(A_1, A_2, \dots, A_n)$, X, Y 是 U 的子集, r 是 R 的任一具体关系, 如果对 r 的任意两个元组 u, v , 由 $u[X]=v[X]$ 导致 $u[Y]=v[Y]$, 则称 X 函数决定 Y , 记为 $X \rightarrow Y$ 。
- 在关系 R 中, X, Y 为 R 的两个属性或属性组, 如果对于 R 的所有关系 r 都存在: 对于 X 的每一个具体值, Y 都只有一个具体值与之对应, 则称属性 Y 函数依赖于属性 X 。

一、函数依赖的定义

- 或者说, 属性X函数决定属性Y, 记作 $X \rightarrow Y$ 。其中X叫决定因素, Y叫被决定因素。
- 此定义可简单表述为: 如果属性X的值决定属性Y的值, 那么属性Y函数依赖于属性X。换一种说法是, 如果知道X的值, 就可以获得Y的值。
- (1) 若Y函数不依赖于X, 记作 $X \nrightarrow Y$ 。
- (2) 若 $X \rightarrow Y, Y \rightarrow X$, 记作 $X \leftrightarrow Y$ 。

一、函数依赖的定义

- 以关系模式学生课程为例, 来说明属性间的函数依赖。
- 设关系模式为: 学生课程 (学生号, 课程号, 成绩, 教师, 教师办公室)
- 在该关系中, 成绩要由学生号和课程号共同确定, 但教师和教师办公室由课程号决定, 所以此关系中包含了以下四种函数依赖关系:
 - ◆ (学生号, 课程号) \rightarrow 成绩
 - ◆ 课程号 \rightarrow 教师
 - ◆ 课程号 \rightarrow 教师办公室
 - ◆ 教师 \rightarrow 教师办公室

二、FD的逻辑蕴涵

- **定义5.2** 设**F**是在关系模式**R (U)**上成立的函数依赖集，**X**和**Y**是属性集**U**上的子集，如果从**F**推导出 **$X \rightarrow Y$** 也在**R (U)**上成立，那么称**F**逻辑蕴涵 **$X \rightarrow Y$** ，记为 **$F \models X \rightarrow Y$** 。
- **定义5.3** 设**F**是关系模式**R (A1,A2,...,An)**上成立的函数依赖集，**X**和**Y**是属性集**(A1,A2,...,An)**的子集，**F**的所有逻辑蕴涵组成的集合称为函数依赖集**F**的闭包,记为 **F^+** 。

$$F^+ = \{ X \rightarrow Y \mid F \models X \rightarrow Y \}$$

即：从给定的函数依赖集合**F**推出的所有函数依赖组成的集合，称为**F**的闭包。

三、函数依赖的推理规则

- **Armstrong**公理系统：设有关系模式 $R(A_1, A_2, \dots, A_n)$ 和属性集 $U = A_1A_2\dots A_n$, X, Y, Z, W 是 U 的一个子集, F 是 R 的一个函数依赖集, 推理规则如下:
 - ◆ 自反律: 如果 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 在 R 上成立。
 - ◆ 增广律: 如果 $X \rightarrow Y$ 为 F 所蕴涵, $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 在 R 上成立。
 - ◆ 传递律: 如果 $X \rightarrow Y$ 和 $Y \rightarrow Z$ 在 R 上成立, 则 $X \rightarrow Z$ 在 R 上成立。

三、函数依赖的推理规则

- FD的其它三个推理规则：
 - ◆ 合并律：如果 $X \rightarrow Y$ 和 $X \rightarrow Z$ 成立，则 $X \rightarrow YZ$ 也成立。
 - ◆ 伪传递律：如果 $X \rightarrow Y$ 和 $WY \rightarrow Z$ 成立，则 $WX \rightarrow Z$ 也成立。
 - ◆ 分解律：如果 $X \rightarrow Y$ 和 $Z \subseteq Y$ ，则 $X \rightarrow Z$ 成立。
- 定义5.4 对于FD $X \rightarrow Y$ 如果 $Y \subseteq X$ ，那么称 $X \rightarrow Y$ 是一个平凡的FD；否则称为一个非平凡的FD
- 例：已知关系模式 $R(ABC)$, $F = \{ A \rightarrow B, B \rightarrow C \}$ ，求 F^+ 。
- 答：根据函数依赖公理系统，可推出 F 的 F^+ 有43个FD。

四、FD与关键码的联系

- 定义5.5 设有关系模式 $R(A_1, A_2, \dots, A_n)$ ， F 是 R 的一个函数依赖集， X 是 $\{A_1, A_2, \dots, A_n\}$ 的一个子集。如果
 - ◆ ① $X \rightarrow A_1 A_2 \dots A_n \in F^+$ ，且
 - ◆ ② 不存在 X 真子集 Y ，使得 $Y \rightarrow A_1 A_2 \dots A_n$ 成立，则称 X 是 R 的候选键。
- 包含在任何一个候选键中的属性称为主属性
- 不包含在任何一个候选键中的属性称为非主属性。

■ 补充：候选码的求解算法（自学）

设关系模式 $R\langle U, F \rangle$

- (1) 将R的所有属性分为 L、R、N和 LR 四类，并令X代表L、N两类，Y代表LR类。

L类：仅出现在F的函数依赖左部的属性；

R类：.....右.....；

N类：在F的函数依赖左右两边都不出现的属性；

LR类：.....都出现的属性。

- (2) 求属性集闭包 X^+ ，若 X^+ 包含了R的全部属性则X即为R的唯一候选码，转(5)；

- (3) 否则, 在 Y 中取一属性 A , 求属性集闭包 $(XA)^+$, 若 $(XA)^+$ 包含了 R 的全部属性, 则转(4); 否则, 调换一属性反复进行这一过程, 直到试完所有 Y 中的属性。
- (4) 如果已找出了所有的候选码, 则转(5); 否则在 Y 中依次取2个、3个、...属性, 求 X 与它们的属性集闭包, 直到其闭包包含 R 的全部属性。
- (5) 停止, 输出结果。

例如：已知 $R(ABCDE)$ ， $F=\{A \rightarrow B, BC \rightarrow A, A \rightarrow D\}$ ，求 R 的全部非主属性。

- 现将所有属性分类
- L: C
- R: D
- N: E
- LR: A、B
- 则 X 代表L、N两类 Y 代表LR类
- 则首先 X 为CE， $(CE)^+=\{C,E\}$ 依次将 X 设为(ACE)和(BCE)得到 $(ACE)^+=\{A,B,C,D,E\}$
- $(BCE)^+=\{A,B,C,D,E\}$
- 由上得，非主属性为D

五、属性集的闭包

- 定义5.6 设关系模式 $R(U, F)$, U 为 R 的属性集合, F 为其函数依赖集, 则称所有用Armstrong公理从 F 推出的函数依赖 $X \rightarrow A_i$ 中 A_i 的属性集合, 为 X 的属性闭包, 记作 X^+ , 读作 X 关于函数依赖集 F 的闭包。
- 定理5.3 设关系模式 $R(U, F)$, U 为 R 的属性集合, F 为其函数依赖集, $X, Y \subseteq U$, 则从 F 推出 $X \rightarrow Y$ 的充要条件是 $Y \subseteq X^+$ 。

五、属性集的闭包

- 算法 5.1 求属性集 X 关于函数依赖 F 的属性闭包 X^+ 。
- 输入：关系模式 R 的全部属性集 U , U 的子集 X , U 上的函数依赖集 F 。
- 输出： X 关于 F 的属性闭包 X^+ 。

- 步骤： 设 $i=0,1,2,\dots$ 。
- (1) 初始化： $i=0, X(0) = X$ 。
- (2) $X(i+1) = X(i) \cup A$
求属性集 A 。 A 是这样的属性： 在 F 中寻找尚未用过的左边是 $X(i)$ 子集的函数依赖： $Y(j) \rightarrow Z(j)$ ($j=1,\dots,k$), 其中 $Y(j) \subseteq X(i)$, 并且在 Z 中寻找 $X(i)$ 中未出现过的属性集合 A , 若无这样的 A , 则转(4)
- (3) 判断是否有 $X(i+1) = X(i)$, 若是则转(4), 否则转(2)。
- (4) 输出 $X(i)$, 即为 X^+ 。

- 遇到下列四种情况之一，可以退出：
 - ◆ $X(i+1) = X(i)$;
 - ◆ $X(i)$ 中已包含了 R 的全部属性;
 - ◆ 在 F 中的每个函数依赖的右边属性中已没有 $X(i)$ 中未出现过的属性;
 - ◆ 在 F 中未用过的函数依赖的左边属性已没有 $X(i)$ 的子集。

- 例 设关系模式 $R(U, F)$, 其中,
 $U = \{A, B, C, D, E, I\}$, $F = \{A \rightarrow D, AB \rightarrow C, BI \rightarrow C, E$
 $D \rightarrow I, C \rightarrow E\}$, 求 $(AC)^+$.
- 解:
- (1) 令 $X = \{AC\}$, 则 $X(0) = AC$.
- (2) 在 F 中找出左边是 AC 子集的函数依赖:
 $A \rightarrow D, C \rightarrow E$.
- (3) $X(1) = X(0) \cup D \cup E = ACDE$.

- (4) 很明显 $X(1) \neq X(0)$,所以 $X(i) = X(1)$,并转向算法中的步骤(2)。
- (5) 在F中找出左边是ACDE子集的函数依赖: $ED \rightarrow I$ 。
- (6) $X(2) = X(1) \cup I = ACDEI$ 。
- (7) 虽然 $X(2) \neq X(1)$,但是F中未用过的函数依赖的左边属性已没有 $X(2)$ 的子集,所以,可停止计算,输出 $(AC)^+ = X(2) = ACDEI$ 。

六、FD的最小依赖集

- 定义 设**F**和**G**是关系模式**R** (**U**) 上的两个函数依赖集,如果 **$F^+ = G^+$** , 则称**F**和**G**是等价的,记作 **$F \equiv G$** 。也可称为**F**覆盖**G**,或**G**覆盖**F**,或**F**、**G**相互覆盖。
- 引理 **$F \equiv G$** 的充分必要条件是 **$F \subseteq G^+$** 、 **$G \subseteq F^+$** 。
- 引理 任一函数依赖集总可以为右边都为单属性的函数依赖集所覆盖。

六、FD的最小依赖集

- 定义5.7 如果函数依赖集**F**满足下列条件,则称**F**为一个极小函数依赖集,也称为最小依赖集或最小覆盖。
 - ◆ (1) **F**中任一函数依赖的右部都是单属性。
 - ◆ (2) **F**中任一函数依赖 $X \rightarrow A$,都不会使**F**与**F**- $\{X \rightarrow A\}$ 等价。
 - ◆ (3) **F**中任一函数依赖 $X \rightarrow A$,**X**的任一真子集**Z**,不会使**F**- $\{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与**F**等价。
- 条件(2)保证了**F**中不存在多余的函数依赖,条件(3)保证了**F**中每个函数依赖的左边没有多余的属性。

- 例 设有函数依赖集
 $F=\{A\rightarrow C, C\rightarrow A, B\rightarrow AC, D\rightarrow AC, BD\rightarrow A\}$, 计算它等价的最小依赖集 F_{\min} 。
- 解:
- (1) 化单依赖右边的属性, 结果为
- $F1=\{A\rightarrow C, C\rightarrow A, B\rightarrow A, B\rightarrow C, D\rightarrow A, D\rightarrow C, BD\rightarrow A\}$
- (2) 去除 $F1$ 的依赖中左边多余的属性。对于 $BD\rightarrow A$, 由于有 $B\rightarrow A$, 所以 D 是多余的。结果为
- $F2=\{A\rightarrow C, C\rightarrow A, B\rightarrow A, B\rightarrow C, D\rightarrow A, D\rightarrow C\}$

- (3) 去除 **F2** 中多余的依赖。 因为：
 $A \rightarrow C, C \rightarrow A$ ，所以 **$A \leftrightarrow C$** 。
- 故： **$B \rightarrow A$** 、 **$B \rightarrow C$** 以及 **$D \rightarrow A$** 、 **$D \rightarrow C$** 中之一为多余的。
- 取 **$F3 = \{ A \rightarrow C, C \rightarrow A, B \rightarrow A, D \rightarrow A \}$** 。 在**F3**中：
 - ◆ 对于 **$A \rightarrow C$** , **$F3 - \{A \rightarrow C\}$** 中 **$A^+ = A$** ;
 - ◆ 对于 **$C \rightarrow A$** , **$F3 - \{C \rightarrow A\}$** 中 **$C^+ = C$** ;
 - ◆ 对于 **$B \rightarrow A$** , **$F3 - \{B \rightarrow A\}$** 中 **$B^+ = B$** ;
 - ◆ 对于 **$D \rightarrow A$** , **$F3 - \{D \rightarrow A\}$** 中 **$D^+ = D$** ;
- 所以,**F3**中已没有多余的函数依赖。
- 即**F**的等价最小依赖集 **F_{\min}** 为：
 $\{ A \rightarrow C, C \rightarrow A, B \rightarrow A, D \rightarrow A \}$ 。

例：设函数依赖集 $F=\{A\rightarrow C, C\rightarrow A, A\rightarrow B, B\rightarrow A, B\rightarrow C\}$ ，试求 F_{\min} 。

解：第一步： F 已满足最小函数依赖集的第①个条件中。

第二步：对于 F 中每一个函数依赖 $X\rightarrow A$ ，分别检验 F 与
 $F-\{X\rightarrow A\}$ 是否等价；

① F 中的 $A\rightarrow C$ （ $\because A\rightarrow B, B\rightarrow C$ ，据传递性）是多余的，所以

$A\rightarrow C$ 可删去。得 $F=\{A\rightarrow B, B\rightarrow A, B\rightarrow C, C\rightarrow A\}$

② 同样 F 中 $B\rightarrow A$ 也是多余的（ $\because B\rightarrow C, C\rightarrow A$ ）， $B\rightarrow A$ 可删去。

得 $F=\{A\rightarrow B, B\rightarrow C, C\rightarrow A\}$

第三步：消去函数依赖左端的多余属性：在本例中， F 中每一个函数依赖左端的都是单属性，不可再分。

$\therefore F_{\min}=\{A\rightarrow B, B\rightarrow C, C\rightarrow A\}$

例：已知 $F = \{ A \rightarrow D, B \rightarrow D, BD \rightarrow CA, CD \rightarrow B \}$ ，求 F_{\min} 。

解：第一步：应用分解规则得：

$$F_1 = \{ A \rightarrow D, B \rightarrow D, BD \rightarrow C, BD \rightarrow A, CD \rightarrow B \}$$

第二步：消去函数依赖左端的冗余属性(应用伪传递)

由 $B \rightarrow D, BD \rightarrow C$ ，可推出 $B \rightarrow C$ ，所以 $BD \rightarrow C$ 的左端的D是多余的；

由 $B \rightarrow D, BD \rightarrow A$ ，可推出 $B \rightarrow A$ ，所以 $BD \rightarrow A$ 的左端的D是多余的；

$$\therefore F_2 = \{ A \rightarrow D, B \rightarrow D, B \rightarrow C, B \rightarrow A, CD \rightarrow B \}$$

第三步：去除F中冗余的FD (应用传递性)；



由 $B \rightarrow A, A \rightarrow D$ ，可推出 $B \rightarrow D$ ，所以 $B \rightarrow D$ 是多余的。

$$\therefore F_{\min} = \{ A \rightarrow D, B \rightarrow C, B \rightarrow A, CD \rightarrow B \}$$

第三节 关系模式的分解特性

- 模式分解问题
- 无损分解
- 无损分解的测试方法
- 保持函数依赖的分解

一、模式分解问题

- 定义5.8 关系模式 $R(U, F)$ 的一个分解是指 $\rho = \{R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)\}$,

$$\text{其中 } U = \bigcup_{i=1}^n U_i$$

并且没有 $u_i \subseteq u_j$, $1 \leq i, j \leq n$, F_i 是 F 在 u_i 上的投影。

- 定义 函数依赖集合 $\{X \rightarrow Y \mid X \rightarrow Y \in F^+ \text{ 且 } XY \subseteq u_i\}$ 的一个覆盖 F_i 叫做 F 在属性 u_i 上的投影。

一、模式分解问题

- 关系模式经分解后,应与原来的关系模式等价。所谓“等价”是指两者对数据的使用者来说应是等价的。即对分解前后的关系,做相同内容的查询,应产生同样的结果。这是对模式分解的基本要求。
- 历年来,人们对等价的概念形成了三种不同的定义:
 - ◆ 分解具有“无损连接性” (**Lossless join**) ;
 - ◆ 分解具有“函数依赖保持性” (**Preserve dependency**) ;
 - ◆ 分解既要具有“无损连接性”,又要具有“函数依赖保持性”。

二、无损分解

- 所谓无损连接性是指对关系模式分解时，原关系模式下的任一合法关系实例，在分解之后，应能通过自然连接运算恢复起来。无损连接性有时也称为无损分解。

- 定义5.9 设 $\rho=\{R_1, R_2, \dots, R_k\}$ 是关系模式 R (U, F) 的一个分解，如果对于 R 的任一满足 F 的关系 r ，都有

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

- 则称分解 ρ 满足函数依赖集 F 的无损连接。
- 根据算法5.2可以测试一个分解具有无损连接性（是否为无损分解）。

三、无损分解的测试方法

- 算法 5.2 检验分解的无损连接性。
- 输入： 关系模式 $R(A_1, A_2, \dots, A_n)$ ； R 上的函数依赖集 F ； R 上的分解 $\rho = \{R_1, R_2, \dots, R_k\}$ 。
- 输出： ρ 是否具有无损连接性。
- 步骤：
 - (1) 构造一 k 行 n 列的表（或矩阵），第 i 行对应于分解后的关系模式 R_i ，第 j 列对应于属性 A_j 。
 - 表中各分量的值由下面的规则确定：

$$M_{ij} = \begin{cases} a_j & A_j \in R_i \\ b_{ij} & A_j \notin R_i \end{cases}$$

三、无损分解的测试方法

- (2) 对 F 中的每一个函数依赖进行反复的检查和处理。具体处理为：取 F 中一个函数依赖 $X \rightarrow Y$ ，在 X 的分量中寻找相同的行，然后将这些行中的 Y 分量改为相同的符号。即如果其中之一为 a_j ，则将 b_{ij} 改为 a_j ；若其中无 a_j ，则改为 b_{ij} ，如：两个符号分别为 b_{23} 和 b_{13} ，则将它们统一改为 b_{23} 或 b_{13} 。
- (3) 如此反复进行，直至 M 无可改变为止。如果发现某一行变成了 a_1, a_2, \dots, a_n ，则 ρ 具有无损连接性；否则， ρ 不具有无损连接性。

- 例 设关系模式 $R(U, F)$ 中, $U = \{A, B, C, D, E\}$, $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow E\}$, R 的一个分解 $\rho = \{R_1(A, B, C), R_2(C, D), R_3(D, E)\}$ 。试判断 ρ 具有无损连接性。

| | A | B | C | D | E |
|----------------|----------|----------|----------|----------|----------|
| $R_1(A, B, C)$ | a_1 | a_2 | a_3 | b_{14} | b_{15} |
| $R_2(C, D)$ | b_{21} | b_{22} | a_3 | a_4 | b_{25} |
| $R_3(D, E)$ | b_{31} | b_{32} | b_{33} | a_4 | a_5 |

(a)

| | A | B | C | D | E |
|----------------|----------|----------|----------|-------|-------|
| $R_1(A, B, C)$ | a_1 | a_2 | a_3 | a_4 | a_5 |
| $R_2(C, D)$ | b_{21} | b_{22} | a_3 | a_4 | a_5 |
| $R_3(D, E)$ | b_{31} | b_{32} | b_{33} | a_4 | a_5 |

(b)

- 解： (1) 首先构造初始表,如表 (a)所示。
- (2) 按下列次序反复检查函数依赖和修改M:
- $AB \rightarrow C$,属性A、 B (第1、 2列) 中都没有相同的分量值,故M值不变;
- $C \rightarrow D$,属性C中有相同值,故应改变D属性中的M值, b_{14} 改为 a_4 ;
- $D \rightarrow E$,属性D中有相同值, b_{15} 、 b_{25} 均改为 a_5 。
- 结果如表 (b) 所示。
- (3) 此时第一行已为 a_1, a_2, a_3, a_4, a_5 ,所以 ρ 具有无损连接性。

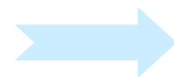
例：设有R（ABCDE）， $F=\{A\rightarrow C, B\rightarrow C, C\rightarrow D, CE\rightarrow A, DE\rightarrow C\}$ ，
R 的一个分解 $\rho=\{AD, AB, BE, CDE, AE\}$ ，
求 ρ 的分解无损性。

答：①根据已知条件构造一张 5 行 5 列的表格如下：

| | A | B | C | D | E |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|
| AD | a ₁ | b ₁₂ | b ₁₃ | a ₄ | b ₁₅ |
| AB | a ₁ | a ₂ | b ₂₃ | b ₂₄ | b ₂₅ |
| BE | b ₃₁ | a ₂ | b ₃₃ | b ₃₄ | a ₅ |
| CDE | b ₄₁ | b ₄₂ | a ₃ | a ₄ | a ₅ |
| AE | a ₁ | b ₅₂ | b ₅₃ | b ₅₄ | a ₅ |

② 根据 $A \rightarrow C$ ，对表一进行处理，将 b_{23} ， b_{53} 改为 b_{13} ，

再考虑 $B \rightarrow C$ ，将 b_{33} 改为 b_{13} ；



| $\begin{matrix} j \\ i \end{matrix}$ | A | B | C | D | E |
|--------------------------------------|----------|----------|-------------------|----------|----------|
| AD | a_1 | b_{12} | b_{13} | a_4 | b_{15} |
| AB | a_1 | a_2 | b_{23} b_{13} | b_{24} | b_{25} |
| BE | b_{31} | a_2 | b_{33} | b_{34} | a_5 |
| CDE | b_{41} | b_{42} | a_3 | a_4 | a_5 |
| AE | a_1 | b_{52} | b_{53} b_{13} | b_{54} | a_5 |

② 根据 $A \rightarrow C$ ，对表一进行处理，将 b_{23} ， b_{53} 改为 b_{13} ，
再考虑 $B \rightarrow C$ ，将 b_{33} 改为 b_{13} ；

| $\begin{matrix} j \\ i \end{matrix}$ | A | B | C | D | E |
|--------------------------------------|----------|----------|-------------------|----------|----------|
| AD | a_1 | b_{12} | b_{13} | a_4 | b_{15} |
| AB | a_1 | a_2 | b_{23} b_{13} | b_{24} | b_{25} |
| BE | b_{31} | a_2 | b_{33} b_{13} | b_{34} | a_5 |
| CDE | b_{41} | b_{42} | a_3 | a_4 | a_5 |
| AE | a_1 | b_{52} | b_{53} b_{13} | b_{54} | a_5 |

② 根据 $A \rightarrow C$ ，对表一进行处理，将 b_{23} ， b_{53} 改为 b_{13} ，

再考虑 $B \rightarrow C$ ，将 b_{33} 改为 b_{13} ；

然后考虑 $C \rightarrow D$ ，将 b_{24} ， b_{34} ， b_{54} 改为 a_4 。修改后的表格如下：

| $\begin{matrix} j \\ i \end{matrix}$ | A | B | C | D | E |
|--------------------------------------|----------|----------|-------------------|----------|----------|
| AD | a_1 | b_{12} | b_{13} | a_4 | b_{15} |
| AB | a_1 | a_2 | b_{23} b_{13} | b_{24} | b_{25} |
| BE | b_{31} | a_2 | b_{33} b_{13} | b_{34} | a_5 |
| CDE | b_{41} | b_{42} | a_3 | a_4 | a_5 |
| AE | a_1 | b_{52} | b_{53} b_{13} | b_{54} | a_5 |

在上表的基础上，考虑 $DE \rightarrow C$ ，将 b_{13} 改为 a_3 ；

考虑 $CE \rightarrow A$ ，将 b_{31} ， b_{41} 改为 a_1 。

| | A | B | C | D | E |
|-----|--------------|----------|--------------|-------|----------|
| AD | a_1 | b_{12} | b_{13} | a_4 | b_{15} |
| AB | a_1 | a_2 | b_{13} | a_4 | b_{25} |
| BE | $b_{31} a_1$ | a_2 | $b_{13} a_3$ | a_4 | a_5 |
| CDE | $b_{41} a_1$ | b_{42} | a_3 | a_4 | a_5 |
| AE | a_1 | b_{52} | $b_{13} a_3$ | a_4 | a_5 |

③ 由于 第三行为全a，所以 ρ 是无损联接。

三、无损分解的测试方法

- 定理5.4 设 $\rho=\{R_1, R_2\}$ 是关系模式 R 的一个分解， F 是 R 上成立的FD集，那么分解 ρ 相对于 F 是无损分解的充分必要条件是：
 - $(R_1 \cap R_2) \rightarrow R_1 - R_2$
或 $(R_1 \cap R_2) \rightarrow R_2 - R_1$

例：试分析下列分解是否具有无损联接性：

①设 $R(ABC)$ ， $F=\{A \rightarrow B\}$ 在 R 上成立， $\rho_1=\{AB, AC\}$ 。

答： ρ_1 相对于 F 具有无损联接性。

$$R_1 \cap R_2 = AB \cap AC = A;$$

$$R_1 - R_2 = AB - AC = B;$$

满足 $A \rightarrow B$;

∴ 模式 $\rho_1=\{AB, AC\}$ 具有无损联接性。

②设 $R(ABC)$, $F=\{A \rightarrow B\}$ 在 R 上成立, $\rho_2=\{AB, BC\}$ 。

答: ρ_2 相对于 F 是有损分解:

$$\bullet \quad R_1 \cap R_2 = AB \cap BC = B; \quad R_1 - R_2 = AB - BC = A;$$

不满足 $A \rightarrow B$;

∴ 模式 $\rho_2=\{AB, BC\}$ 是有损分解。

四、保持函数依赖的分解

- 定义5.10 设有关系模式R，F是R的函数依赖集，Z是R的一个属性集合，则Z所涉及到的F中所有函数依赖为F在Z上的投影，记为 $\pi_Z(F)$ ，有

$$\pi_Z(F) = \{ X \rightarrow Y \mid X \rightarrow Y \in F^+ \text{ 且 } XY \subseteq Z \}$$

- 定义5.11 设关系模式R的一个分解 $\rho = \{R_1, R_2, \dots, R_k\}$ ，F是R的依赖集，如果F等价于

$$\bigcup_{i=1}^k \pi_{R_i}(F)$$

则称分解 ρ 具有函数依赖保持性。

例：试分析下列分解是否具有无损联接性和保持函数依赖性：

设 $R(ABC)$ ， $F=\{A\rightarrow C, B\rightarrow C\}$ 在 R 上成立， $\rho=\{AB, AC\}$ 。

解： ρ 相对于 F 是无损联接分解但不保持函数依赖性。

$$\because R1 \cap R2 = AB \cap AC = A; \quad R2 - R1 = AC - AB = C;$$

满足 $A\rightarrow C$;

\therefore 模式 $\rho = \{AB, AC\}$ 具有无损联接性。

又 \because 已知 $F=\{A\rightarrow C, B\rightarrow C\}$

$$\because A^+=AC \quad B^+=BC \quad (AB)^+ = ABC$$

$$\therefore \pi_{AB}(F) = \{\phi\};$$

$$\pi_{AC}(F) = \{A\rightarrow C\};$$

而 $\pi_{AB}(F) \cup \pi_{AC}(F) = \{A\rightarrow C\}$ ， $B\rightarrow C$ 在该分解中丢失了。

\therefore 模式 ρ 不具有保持函数依赖性。

举 例1：试分析下列分解是否保持函数依赖性：

设 $R(ABCD)$, $F=\{A\rightarrow B, B\rightarrow C, C\rightarrow D, D\rightarrow A\}$, $\rho = \{AB, BC, CD\}$

请运用保持函数依赖集的测试算法验证 ρ 是否保持函数依赖性。

解：算法的第一步是计算F在每一个关系模式Ri上的投影：

$$\because A^+=ABCD, \quad B^+=ABCD, \quad C^+=ABCD, \quad D^+=ABCD$$

$$\therefore \pi_{AB}(F) = \{A\rightarrow B, B\rightarrow A\} \quad \pi_{BC}(F) = \{B\rightarrow C, C\rightarrow B\}$$

$$\pi_{CD}(F) = \{C\rightarrow D, D\rightarrow C\}$$

算法的第二步是逐步验证F中每个FD是否被 $\bigcup_{i=1}^k \pi_{R_i}(F)$ 逻辑蕴涵。

$$\because \bigcup_{i=1}^k \pi_{R_i}(F) = \{A\rightarrow B, B\rightarrow A, B\rightarrow C, C\rightarrow B, C\rightarrow D, D\rightarrow C\} \quad \text{只要判断其是否逻辑蕴涵 } D\rightarrow A?$$

由 $D\rightarrow C, C\rightarrow B, B\rightarrow A$ 可推出 $D\rightarrow A$,

$$\therefore \rho = \{ AB, BC, CD \} \text{ 保持 FD.}$$

举例2：设关系模式R（ABC）分解成 $\rho = \{ AB, BC \}$ ，如果R上的FD集 $F = \{ A \rightarrow B \}$ ，那么这个分解是损失分解。试举出R的一个关系r，不满足 $m_\rho(r) = r$ 。

解：反例 r 可以用测试时的初始表格：即满足F的泛关系r如下所示

| r | $\pi_{AB}(r)$ | $\pi_{BC}(r)$ |
|---|--------------------------------|--------------------------------|
| AB a ₁ a ₂ b ₁₃ | a ₁ a ₂ | a ₂ b ₁₃ |
| BC b ₂₁ a ₂ a ₃ | b ₂₁ a ₂ | a ₂ a ₃ |

$\pi_{AB}(r) \bowtie \pi_{BC}(r)$ 有四个元组：

| A |
|-----------------|
| a ₁ |
| a ₁ |
| b ₂₁ |
| b ₂₁ |

| |
|----------------|
| a ₂ |
| a ₂ |
| a ₂ |
| a ₂ |

| |
|-----------------|
| b ₁₃ |
| a ₃ |
| b ₁₃ |
| a ₃ |

$\therefore m_\rho(r) \neq r$

第四节 关系模式的范式

- 第一范式1NF
- 第二范式2NF
- 第三范式3NF
- 巴克斯范式BCNF
- 分解成BCNF模式集的方法
- 分解成3NF模式集的方法

一、第一范式1NF

- 非规范化的关系
- 当一个关系中的所有分量都是不可分的数据项时,该关系是规范化的。表 **5 — 4** 具有组合数据项,表 **5 — 5** 具有多值数据项,因此都不是规范化的表。

表5 — 4 具有组合数据项的非规范化的表

| 职工号 | 姓名 | 工资 | | |
|-----|----|------|------|------|
| | | 基本工资 | 职务工资 | 工龄工资 |
| | | | | |

表 5—5 具有多值数据项非规范化表

| 职工号 | 姓名 | 职称 | 系名 | 系办公地址 | 学历 | 毕业年份 |
|-----|----|----|-----|---------|-----|------|
| 001 | 张三 | 教授 | 计算机 | 1--305 | 大学 | 1963 |
| | | | | | 研究生 | 1982 |
| 002 | 李四 | 讲师 | 信电 | 2---204 | 大学 | 1989 |

一、第一范式1NF

- 定义**5.12** 如果关系模式**R**的所有的属性的值域中每一个值都是不可分解的值，则**R**满足第一范式,简称**1NF** (**First Normal Form**) , 记作 **$R \in 1NF$** 。
- **1NF**是对关系的最低要求,不满足**1NF**的关系是非规范化关系,如表 **5—4**、 表 **5—5** 所示。
- 非规范化关系转化为**1NF**的方法很简单,当然也不是唯一的。 对表 **5 — 4**、 表 **5 — 5** 分别进行横向和纵向展开,可分别转化为如表 **5 — 6**、 表 **5 — 7** 所示的符合**1NF**的关系。

表 5 — 6 消除组合数据项后的表

| 职工号 | 姓名 | 基本工资 | 职务工资 | 工龄工资 |
|-----|----|------|------|------|
| | | | | |

表 5 — 7 消除多值数据项后的表

| 职工号 | 姓名 | 职称 | 系名 | 系办公地址 | 学历 | 毕业年份 |
|-----|----|----|-----|--------|-----|------|
| 001 | 张三 | 教授 | 计算机 | 1- 305 | 大学 | 1963 |
| 001 | 张三 | 教授 | 计算机 | 1- 305 | 研究生 | 1982 |
| 002 | 李四 | 讲师 | 信电 | 2- 204 | 大学 | 1989 |

二、第二范式2NF

- 表 5 — 7 虽然已符合1NF的要求,但表中存在大量的数据冗余和潜在的数据更新异常。原因是该关系的码是 (职工号, 学历), 但姓名、 职称、 系名、 系办公地址却与学历无关, 即它们只与码的一部分有关。
- 关系模式会产生以下几个问题:
 - ◆ (1) 插入异常。
 - ◆ (2) 删除异常。
 - ◆ (3) 修改异常。
- 定义5.13 设X、Y是关系R的两个不同的属性或属性组, 且 $X \rightarrow Y$ 。如果存在X的某一个真子集 X' , 使 $X' \rightarrow Y$ 成立, 则称Y部分函数依赖于X, 记作 $X \xrightarrow{p} Y$ 。 反之, 则称Y完全函数依赖于X, 记作 $X \xrightarrow{f} Y$ 。

二、第二范式2NF

- 定义5.15 如果关系模式R为1NF，并且R中的每一个非主属性完全函数依赖于R的某个候选键，则称R是属于2NF。
- 对于表 5 — 7 所示的关系模式：
- 职工信息（职工号 ,姓名,职称,系名,系办公地址,学历,毕业年份）
- 由于也包含了部分函数依赖,不符合**2NF**的定义,可将其分解为下面两个关系模式：
 - ◆ 职工信息（职工号 ,姓名,职称,系名,系办公地址）
 - ◆ 学历（职工号,学历 ,毕业年份）
- 推论： 如果关系模式 $R \in 1NF$,且它的每一个候选码都是单码,则 $R \in 2NF$ 。

三、第三范式3NF

- 符合第二范式的关系模式仍可能存在数据冗余、更新异常等问题。
- 如前面从表 5— 7 分解出的职工信息关系模式： 职工信息（职工号，姓名，职称，系名，系办公地址）。 如果一个系有100位职工，那么系办公地址就要重复存储100次，存在着较高的数据冗余。
- 原因是此关系模式的码是职工号，而系办公地址通过系名函数依赖于职工号，即职工号→系名，系名→系办公地址，是一个传递依赖的过程。

三、第三范式3NF

- 定义5.16 在关系R中, X、 Y、 Z是R的三个不同的属性或属性组, 如果 $X \rightarrow Y$, $Y \rightarrow Z$, 但 $Y \not\rightarrow X$ 且Y不是X的子集, 则称Z传递依赖于X。
- 定义5.17 如果关系模式R为2NF, 并且R中的每一个非主属性都不传递依赖于R的候选键, 则称R是属于3NF。
- 上述职工关系不属于**3NF**, 可把它分解如下, 使其符合**3NF**:
 - ◆ 职工 (职工号, 姓名, 职称, 系名)
 - ◆ 系 (系名, 系办公地址)

举例：设有关系模式R（职工编号，日期，日营业额，部门名，部门经理），该模式统计商店里每个职工的日营业额，以及职工所在的部门和经理信息。

如果规定：每个职工每天只有一个营业额；每个职工只在一个部门工作；每个部门只有一个经理。

试回答下列问题：

- （1）根据上述规定，写出模式R的基本FD和关键码；
- （2）说明R不是2NF的理由，并把R分解成2NF模式集；
- （3）进而分解成3NF模式集。

解：（1）据： 每个职工每天只有一个营业额，

每个职工只在一个部门工作， 每个部门只有一个经理。

得到基本的函数依赖有三个：（职工编号，日期）

职工编号 \rightarrow 部门名， 部门名 \rightarrow 部门经理

R的关键码为（职工编号，日期）

（2）R中有两个这样的FD：

（职工编号，日期） \rightarrow （部门名，部门经理）

职工编号 \rightarrow （部门名，部门经理）

由于前一个函数依赖是部分函数依赖，所以R不是2NF模式。

R应分解成 R1（职工编号，部门名，部门经理）

R2（职工编号，日期，日营业额）

此时，R1和R2都是2NF模式。

R1 (职工编号, 部门名, 部门经理)

R2 (职工编号, 日期, 日营业额)

(3) **R2** 已是3NF模式。

但在R1中存在传递依赖：职工编号 \rightarrow 部门经理

\therefore 职工编号 \rightarrow 部门名, 部门名 \rightarrow 部门经理

\therefore **R1** 不是3NF模式。

将 **R1** 分解为： **R11** (职工编号, 部门名)

R12 (部门名, 部门经理)

这样， $\rho = \{ \mathbf{R11}, \mathbf{R12}, \mathbf{R2} \}$ 是一个3NF模式集。

四、巴克斯范式BCNF

- 第三范式的修正形式是**Boyce—Codd**范式（简称**BCNF**），是由**Boyce**与**Codd**提出的。
- 定义**5.18** 设关系模式 $R(U, F) \in 1NF$, 若 F 的任一函数依赖 $X \rightarrow Y$ ($Y \not\subseteq X$) 中 X 都包含了 R 的一个码, 则称 $R \in BCNF$ 。
- 换言之, 在关系模式 R 中, 如果每一个决定因素都包含码, 则 $R \in BCNF$ 。

四、巴克斯范式BCNF

- 由**BCNF**的定义可以得到以下推论： 如果 **$R \in \text{BCNF}$** , 则
 - ◆ **R**中所有非主属性对每一个码都是完全函数依赖;
 - ◆ **R**中所有主属性对每一个不包含它的码, 都是完全函数依赖;
 - ◆ **R**中没有任何属性完全函数依赖于非码的任何一组属性。

例如：关系模式 $R(C, S, Z)$ 其中

C ：城市名称 S ：街道名称 Z ：邮政编码

假定： $F = \{ CS \rightarrow Z, Z \rightarrow C \}$

$CS \rightarrow Z$ ：表示地址(城市和街道)决定邮政编码；

$Z \rightarrow C$ ：表示邮政编码决定城市名称。

R 的候选码为： CS 和 SZ 。

由于F中存在 $Z \rightarrow C$:

主属性C对不包含它的码SZ不是完全函数依赖,

所以 $R(C, S, Z)$ 不是BCNF模式, 但 $R(C, S, Z)$ 是3NF,

因为没有任何非主属性对码传递函数依赖或部分函数依赖。

如果把 $R(C, S, Z)$ 分解为: $R_1 (S, Z)$

$R_2 (Z, C)$

能解决冗余问题, 而且 R_1, R_2 都是BCNF模式集。

但丢失了 $CS \rightarrow Z$, 数据语义将会引起新的矛盾。

五、分解成BCNF模式集的方法

- 算法**5.3** 将关系模式分解为**BCNF**，使它具有无损连接。
- 输入：关系模式**R**和函数依赖**F**。
- 输出：具有无损连接性的分解，使得分解中的每一个关系模式对于**F**在这个模式上的投影都满足**BCNF**。
- 方法：1、构造 **ρ** ，开始 **$\rho = \{R(U, F)\}$**
- 2、分解 **ρ** ，检查 **ρ** 中各关系模式是否均属于**BCNF**，若是，则算法终止。

五、分解成BCNF模式集的方法

- 3、如果 **S** 是 **p** 的一个关系模式，而 **S** 不是 **BCNF**，又设 $X \rightarrow A$ 为 **S** 所满足的函数依赖，其中 **X** 不包含 **S** 的码，**A** 不包含在 **X** 中，则在 **S** 中除 **X**，**A** 外还有一些属性存在，否则 **X** 就是 **S** 的码。将 **X**，**A** 组成模式 **S1**，将 **S** 中除去 **A** 外其他属性组成模式 **S2**。由于 $S1 \cap S2 = X$ ， $S1 - S2 = A$ ，而且满足 $X \rightarrow A$ ，所以 **S** 分解为 **S1**，**S2** 具有无损连接性，分解 **p** 中将由 **S1** 和 **S2** 代替，**S1** 为 **BCNF**。
- **S1** 和 **S2** 中包含的属性个数都比 **S** 中的属性个数少，经过有限次迭代 **p** 的每个模式都是 **BCNF**。

六、分解成3NF模式集的方法

- 算法**5.4** 将一个关系模式分解为**3NF**，使它具有保持函数依赖。
- 输入：关系模式**R**和最小函数依赖集 **F_{min}** 。
- 输出：**R**的一个分解 **$\rho=\{R_1, R_2, \dots, R_K\}$** ， **$R_i$ 为3NF**， **ρ 保持函数依赖**。
- 方法：
 - 1、如果 **F_{min}** 只有一个函数依赖 **$X \rightarrow A$** ，且 **$XA=R$** ，则输出 **$\rho=\{R\}$**

六、分解成3NF模式集的方法

- 2、如果 \mathbf{R} 中某些属性与 \mathbf{F}_{\min} 中所有函数依赖的左部和右部都无关系，则将他们组成一个关系模式，从 \mathbf{R} 中将它们分出去。
- 3、对于 \mathbf{F}_{\min} 中的每一个 $\mathbf{X}_i \rightarrow \mathbf{A}_i$ ，都构成一个关系子模式 $\mathbf{R}_i = \mathbf{X}_i \mathbf{A}_i$
- 4、停止分解，输出 \mathbf{R} 。

例：关系模式

最小函数依赖集 $F_{\min} = (C \rightarrow T, HR \rightarrow C, HT \rightarrow R,$
 $CS \rightarrow G, HS \rightarrow R)$

根据算法，生成的分解为：

$\rho = (CT, HRC, HTR, CSG, HSR)$

这样的分解，显然将原有的函数依赖都保持下来，而且
每个分解模式均为3NF模式。

六、分解成3NF模式集的方法

- 算法**5.5** 将一个关系模式分解为**3NF**，使它既具有无损连接又保持函数依赖的分解。
- 方法：
- 1、算法**5.4**中关系模式**R**分解为 $\rho = \{R_1, R_2, \dots, R_k\}$ ，设**X**是**R**的码，则 $\tau = \rho \cup \{X\}$ 是**R**的一个分解。
- 2、若有某个 u_i ， $X \subseteq u_i$ ，将**X**从 τ 中去掉。
- 3、 τ 就是所求的分解。

例: $\rho = (CT, HRC, HTR, CSG, HSR)$

$\because HS^+ = CTHSRG$

$\tau = \rho \cup \{X\} = (CT, HRC, HTR, CSG, HSR)$

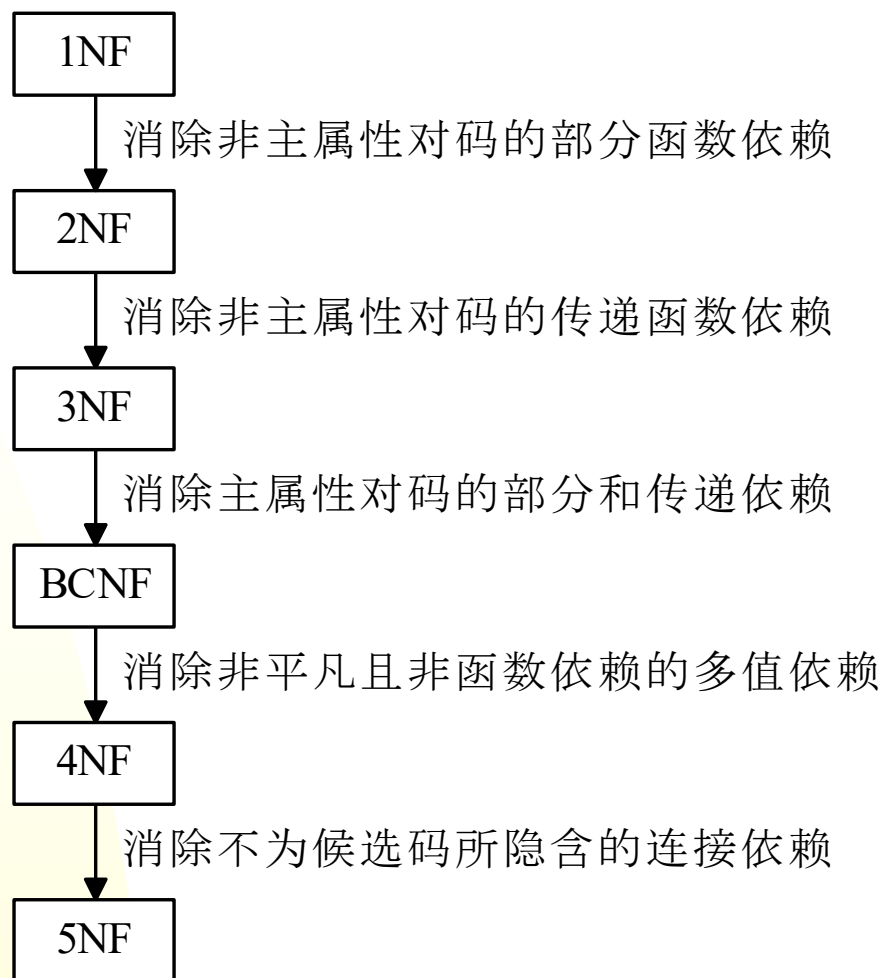


图 5—1 范式递进过程示意图