



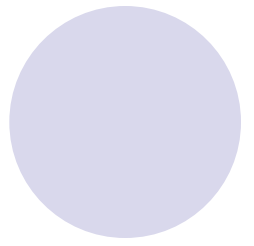
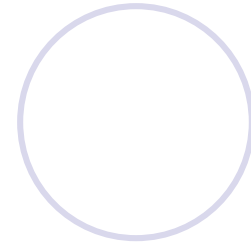
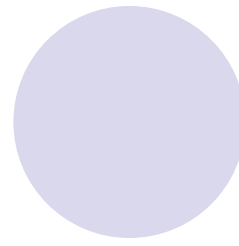
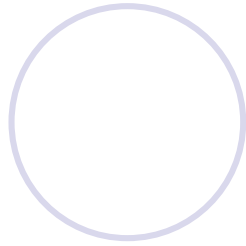
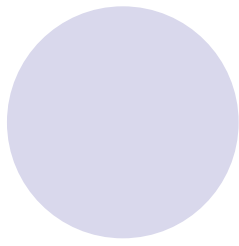
# 数据库系统

Database System

主讲：张仲楠 教授

Email: [zhongnan\\_zhang@xmu.edu.cn](mailto:zhongnan_zhang@xmu.edu.cn)

Office: 海韵A416



# 数据库系统 Database System

## 第六章 关系数据理论

# 第六章 关系数据理论

**6.1 问题的提出**

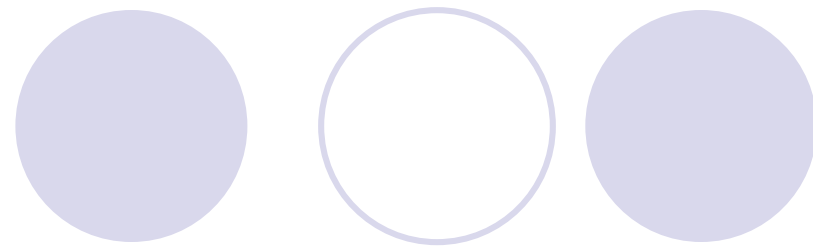
**6.2 规范化**

**6.3 数据依赖的公理系统**

**\*6.4 模式的分解**

**6.5 小结**

## 6.1 问题的提出



- 针对具体问题，如何构造一个适合于它的数据模式(几个关系模式，哪些属性组成)
- 数据库逻辑结构设计问题
- 数据库**逻辑设计**的工具——关系数据库的**规范化理论**

# 问题的提出

The slide features a decorative header with five circles. The first circle is solid light purple and partially overlaps the title. The second circle is an outline of the same color. To the right, there are three more circles: a solid light purple one, an outline one, and another solid light purple one.

一、概念回顾

二、关系模式的形式化定义

三、什么是数据依赖

四、关系模式的简化定义

五、数据依赖对关系模式影响

# 一、概念回顾

- **关系**：描述实体、属性、实体间的联系。
  - 从形式上看，它是一张二维表，是所涉及属性的笛卡尔积的一个子集。
- **关系模式**：对关系的描述。
  - 元组集合的结构，完整性约束条件
- **关系数据库**：基于关系模型的数据库，利用关系来描述现实世界。
  - 从形式上看，它由一组关系组成。
- **关系数据库的模式**：定义这组关系的关系模式的全体。

## 二、关系模式的形式化定义

关系模式由五部分组成，即它是一个五元组：

$$R(U, D, DOM, F)$$

R: 关系名

U: 组成该关系的属性名集合

D: 属性组U中属性所来自的域

DOM: 属性向域的映象集合

F: 属性间数据依赖集合

### 三、什么是数据依赖

- 一个**关系内部**属性与属性之间的**约束关系**
- 通过属性间值的**相等与否**体现出来的数据间相关联系
- 现实世界属性间相互联系的抽象
- 数据内在的性质
- **语义**的体现



# 什么是数据依赖（续）

## 主要的数据依赖类型

- 函数依赖（Functional Dependency, 简记为FD）  
 $y=f(x)$  学号(x) **确定** 学生所在系(y) 记作：学号  $\rightarrow$  所在系  
类似：学号(x) **确定** 学生姓名(y) 记作：学号  $\rightarrow$  学生姓名
- 多值依赖（Multivalued Dependency, 简记为MVD）

## 四、关系模式的简化表示

- 关系模式  $R(U, D, DOM, F)$

简化为一个三元组：

$$R\langle U, F \rangle$$

- 当且仅当  $U$  上的一个关系  $r$  满足  $F$  时， $r$  称为关系模式  $R\langle U, F \rangle$  的一个关系

## 五、数据依赖对关系模式的影响

[例1]建立一个描述学校教务的数据库：

学生的学号（Sno）、所在系（Sdept）

系主任姓名（Mname）、课程号（Cno）

成绩（Grade）

单一的关系模式： Student  $\langle U, F \rangle$

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

# 数据依赖对关系模式的影响（续）

数据库具有如下语义：

- 一个系有若干学生，但一个学生只属于一个系

$Sno \rightarrow Sdept$

- 一个系只有一名负责人（比如：系主任）

$Sdept \rightarrow Mname$

- 一个学生可以选修多门课程，每门课程有若干学生选修

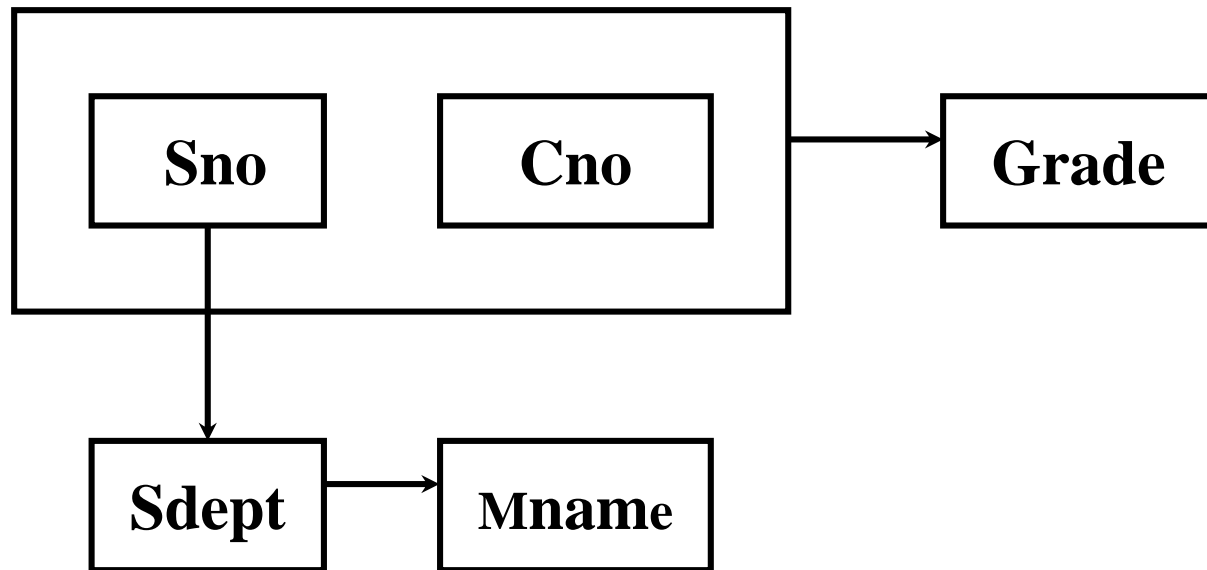
- 每个学生学习每一门课程有一个成绩

$(Sno, Cno) \rightarrow Grade$

# 数据依赖对关系模式的影响（续）

属性组U上的一组函数依赖F:

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



# 数据依赖对关系模式的影响（续）

Sno	Sdept	Mname	Cno	Grade
S1	计算机系	张明	C1	95
S2	计算机系	张明	C1	90
S3	计算机系	张明	C1	88
S4	计算机系	张明	C1	70
S5	计算机系	张明	C1	78
...				

# 关系模式Student<U, F>中存在的问题

- 1. 数据冗余太大(比如系主任)
- 2. 更新异常 (Update Anomalies)
  - 比如：更新系主任名

Sno	Sdept	Mname	Cno	Grade
S1	计算机系	张明	C1	95
S2	计算机系	张明	C1	90
S3	计算机系	张明	C1	88
S4	计算机系	张明	C1	70
S5	计算机系	张明	C1	78
...				

# 关系模式Student<U, F>中存在的问题

- 3. 插入异常（**Insertion Anomalies**）

- 该存储的数据插无法存储

例，如果一个系刚成立，尚无学生，我们就无法把这个系及其系主任的信息存入数据库。

- 4. 删除异常（**Deletion Anomalies**）

- 不该删除的数据不得不删

例，如果某个系的学生全部毕业了，我们在删除该系学生信息的同时，把这个系及其系主任的信息也丢掉了。



# 数据依赖对关系模式的影响（续）

结论：

- Student关系模式不是一个“好”的模式。
- “好”的模式：

不会发生插入异常、删除异常、更新异常，  
数据冗余应尽可能少

原因：由存在于模式中的某些数据依赖引起的

解决方法：通过分解关系模式来消除其中不合适的  
数据依赖

# 分解关系模式

- 把这个单一模式分成3个关系模式：

- 分解前

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

- 分解后

$S (U=\{Sno, Sdept\}, F=\{Sno \rightarrow Sdept\}) ;$

$SC (U=\{Sno, Cno, Grade\}, F=\{(Sno, Cno) \rightarrow Grade\}) ;$

$DEPT (U=\{Sdept, Mname\}, F=\{Sdept \rightarrow Mname\})$

# 第六章 关系数据理论

**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**

## 6.2 规范化

规范化理论正是用来改造关系模式，通过分解关系模式来消除其中不合适的数据依赖，以解决插入异常、删除异常、更新异常和数据冗余问题。

## 6.2 规范化

### 6.2.1 函数依赖

### 6.2.2 码

### 6.2.3 范式

### 6.2.4 2NF

### 6.2.5 3NF

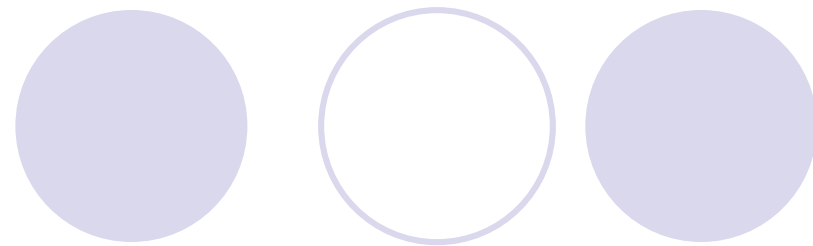
### 6.2.6 BCNF

### 6.2.7 多值依赖

### 6.2.8 4NF

### 6.2.9 规范化小结

## 6.2.1 函数依赖



- 函数依赖
- 平凡函数依赖与非平凡函数依赖
- 完全函数依赖与部分函数依赖
- 传递函数依赖

# 一、函数依赖

定义6.1 设 $R(U)$ 是一个属性集 $U$ 上的关系模式， $X$ 和 $Y$ 是 $U$ 的子集。若对于 $R(U)$ 的任意一个可能的关系 $r$ ， $r$ 中不可能存在两个元组在 $X$ 上的属性值相等，而在 $Y$ 上的属性值不等，则称“ $X$ 函数确定 $Y$ ”或“ $Y$ 函数依赖于 $X$ ”，记作 $X \rightarrow Y$ 。

换种说法：如果 $r$ 中任意两个元组在 $X$ 上的属性值相等，在 $Y$ 上的属性值也相等

# 说明

- 1. 函数依赖不是指关系模式**R**的某个或某些关系实例满足的约束条件，而是指**R**的所有关系实例均要满足的约束条件。
- 2. 函数依赖是语义范畴的概念。只能根据数据的语义来确定函数依赖。  
例如“姓名→年龄”这个函数依赖只有在不允许有同名人的条件下成立
- 3. 数据库设计者可以对现实世界作强制的规定。例如规定不允许同名人的出现，函数依赖“姓名→年龄”成立。所插入的元组必须满足规定的函数依赖，若发现有同名人的存在，则拒绝装入该元组。



## 二、平凡函数依赖与非平凡函数依赖

在关系模式 $R(U)$ 中，对于 $U$ 的子集 $X$ 和 $Y$ ，

如果 $X \rightarrow Y$ ，但 $Y \not\subseteq X$ ，则称 $X \rightarrow Y$ 是**非平凡的函数依赖**

若 $X \rightarrow Y$ ，但 $Y \subseteq X$ ，则称 $X \rightarrow Y$ 是**平凡的函数依赖**

- 例：在关系 $SC(Sno, Cno, Grade)$ 中，

非平凡函数依赖：  $(Sno, Cno) \rightarrow Grade$

平凡函数依赖：  $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$

## 平凡函数依赖与非平凡函数依赖（续）

○ 若  $X \rightarrow Y$ ，则  $X$  称为这个函数依赖的**决定属性组**，也称为**决定因素**（Determinant）。

○ 若  $X \rightarrow Y$ ， $Y \rightarrow X$ ，则记作  $X \leftrightarrow Y$ 。

比如：  
身份证号  $\leftrightarrow$  学号

○ 若  $Y$  不函数依赖于  $X$ ，则记作  $X \nrightarrow Y$ 。

○ 平凡的函数依赖**不反映新的语义**，都必然成立

○ 一般都讨论**非平凡的**函数依赖

### 三、完全函数依赖与部分函数依赖

**定义6.2** 在 $R(U)$ 中, 如果 $X \rightarrow Y$ , 并且对于 $X$ 的任何一个真子集 $X'$ , 都有 $X' \nrightarrow Y$ , 则称 $Y$ 对 $X$ 完全函数依赖, 记作 $X \xrightarrow{F} Y$ 。

若 $X \rightarrow Y$ , 但 $Y$ 不完全函数依赖于 $X$ , 则称 $Y$ 对 $X$ 部分函数依赖, 记作 $X \xrightarrow{P} Y$ 。

## 完全函数依赖与部分函数依赖（续）

[例1] 中  $(Sno, Cno) \xrightarrow{F} Grade$  是完全函数依赖，

$(Sno, Cno) \xrightarrow{P} Sdept$  是部分函数依赖

因为  $Sno \rightarrow Sdept$  成立，且  $Sno$  是  $(Sno, Cno)$  的真子集

## 四、传递函数依赖

非平凡

**定义6.3** 在 $R(U)$ 中, 如果 $X \rightarrow Y$ ,  $(Y \not\subseteq X)$ ,  $Y \nrightarrow X$   
 $Y \rightarrow Z$ ,  $Z \not\subseteq Y$  则称 $Z$ 对 $X$ **传递函数依赖**。

记为:  $X \xrightarrow{\text{传递}} Z$

注: 如果 $Y \rightarrow X$ , 即 $X \longleftrightarrow Y$ , 则 **$Z$ 直接依赖于 $X$** 。

例: 在关系 $Std(Sno, Sdept, Mname)$ 中, 有:

$Sno \rightarrow Sdept$ ,  $Sdept \rightarrow Mname$

$Mname$ 传递函数依赖于 $Sno$

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.2 码

定义6.4 设K为R<U,F>中的属性或属性组合。若

$K \xrightarrow{F} U$ , 则K称为R的**候选码** (Candidate

Key)。若 $K \rightarrow U$ , 则K称为R的**超码**

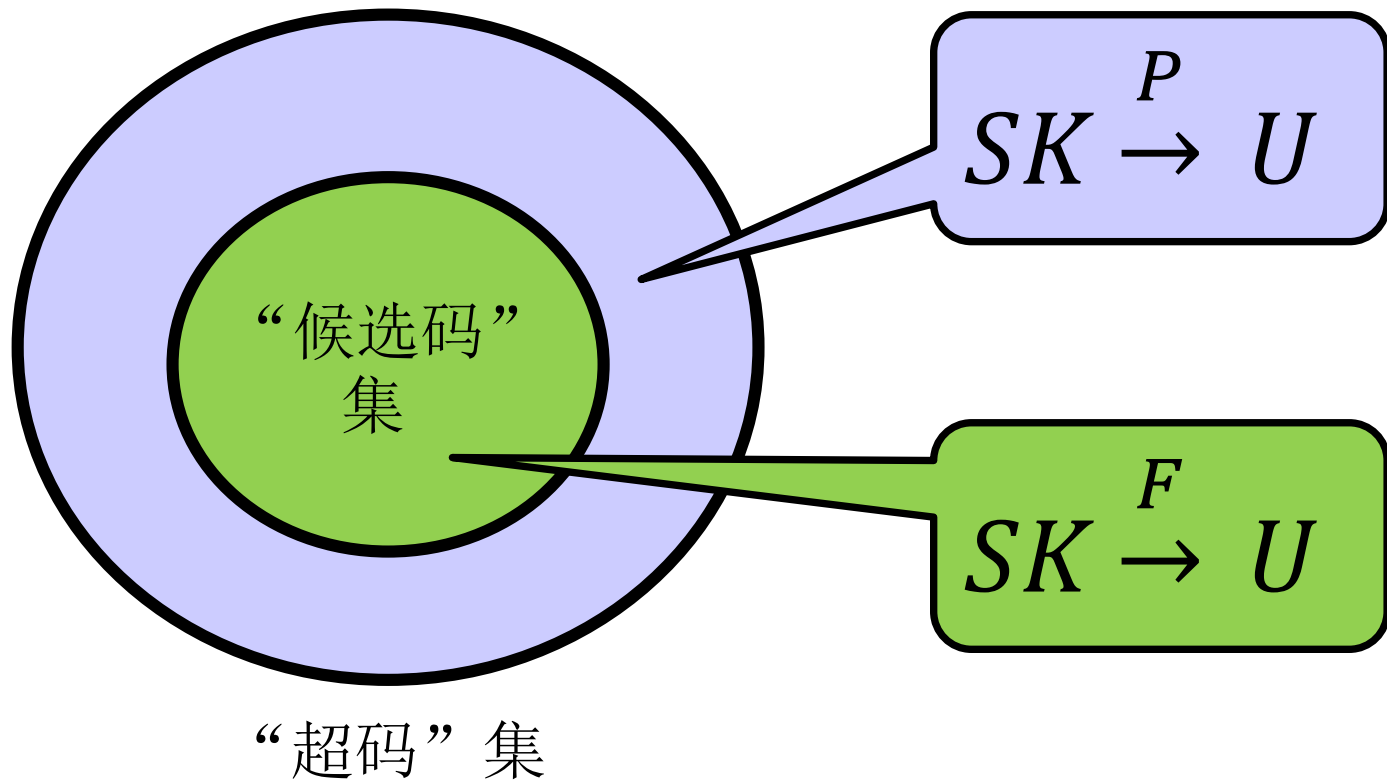
(Superkey)。

- 若候选码多于一个, 则选定其中的一个做为**主码** (Primary Key)。

码（续）

- 候选码与超码的关系

对于某一个超码  $SK$





## 码（续）

- 主属性与非主属性

- 包含在任何一个候选码中的属性，称为主属性

- （Prime attribute）

- 不包含在任何候选码中的属性称为非主属性

- （Nonprime attribute）或非码属性（Non-key attribute）

- 全码

- 整个属性组是码，称为全码（All-key）

# 码（续）

## [例2]

关系模式 $S(\underline{Sno}, Sdept, Sage)$ ，单个属性 $Sno$ 是码，  
 $SC(\underline{Sno}, \underline{Cno}, Grade)$ 中， $(Sno, Cno)$ 是码

## [例3]

关系模式 $R(P, W, A)$

$P$ : 演奏者     $W$ : 作品     $A$ : 听众

一个演奏者可以演奏多个作品

某一作品可被多个演奏者演奏

一个演奏者演奏的某个作品可以被不同听众欣赏

听众可以欣赏不同演奏者的不同作品

码为 $(P, W, A)$ ，即All-Key

# 外部码

**定义6.5** 关系模式  $R$  中属性或属性组  $X$  并非  $R$  的码，但  $X$  是另一个关系模式的码，则称  $X$  是  $R$  的**外部码**（**Foreign key**）也称**外码**

- 如在  $SC(\underline{Sno}, \underline{Cno}, Grade)$  中， $Sno$  不是码，但  $Sno$  是关系模式  $S(\underline{Sno}, Sdept, Sage)$  的码， $Sno$  是关系模式  $SC$  的外部码
- 主码与外部码一起提供了表示关系间联系的手段

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.3 范式

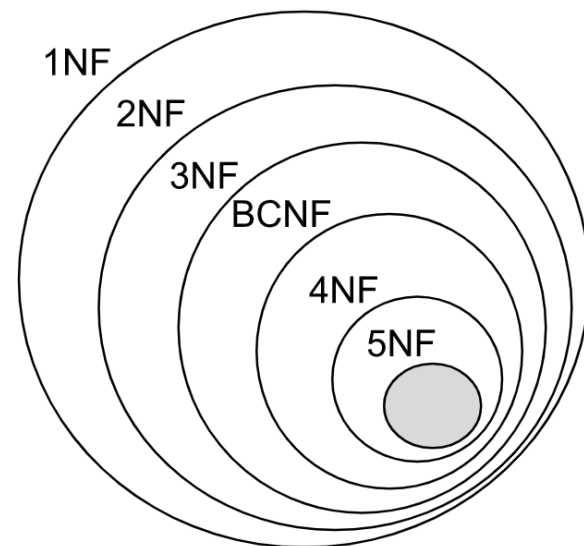
- 关系数据库中的关系必须满足一定的要求。满足不同程度要求的为不同范式
- 范式是符合某一种级别的关系模式的集合
- 范式的种类：

{ 第一范式(1NF)  
第二范式(2NF)  
第三范式(3NF)  
BC范式(BCNF)  
第四范式(4NF)  
第五范式(5NF)

要求不断提高

## 6.2.3 范式

- 各种范式之间存在联系：



$$1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$$

- 某一关系模式R为第n范式，可简记为 $R \in nNF$ 。
- 一个低一级范式的关系模式，通过模式分解可以转换为若干个高一级范式的关系模式的集合，这种过程就叫规范化（normalization）

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

**6.2.4 2NF**

**6.2.5 3NF**

**6.2.6 BCNF**

6.2.7 多值依赖

**6.2.8 4NF**

6.2.9 规范化小结

## 6.2.4 2NF

- 1NF的定义

如果一个关系模式R的所有属性都是不可分的基本数据项(原子值), 则 $R \in 1NF$

- 第一范式是对关系模式的最起码的要求。不满足第一范式的数据库模式不能称为关系数据库
- 但是满足第一范式的关系模式并不一定是一个好的关系模式



## 2NF (续)

[例4] 关系模式 S-L-C(Sno, Sdept, Sloc, Cno, Grade)  
Sloc为学生住处，假设每个系的学生住在同一个地方

- 函数依赖包括：

$(Sno, Cno) \xrightarrow{F} Grade$

$Sno \rightarrow Sdept$

$(Sno, Cno) \xrightarrow{P} Sdept$

$Sno \rightarrow Sloc$

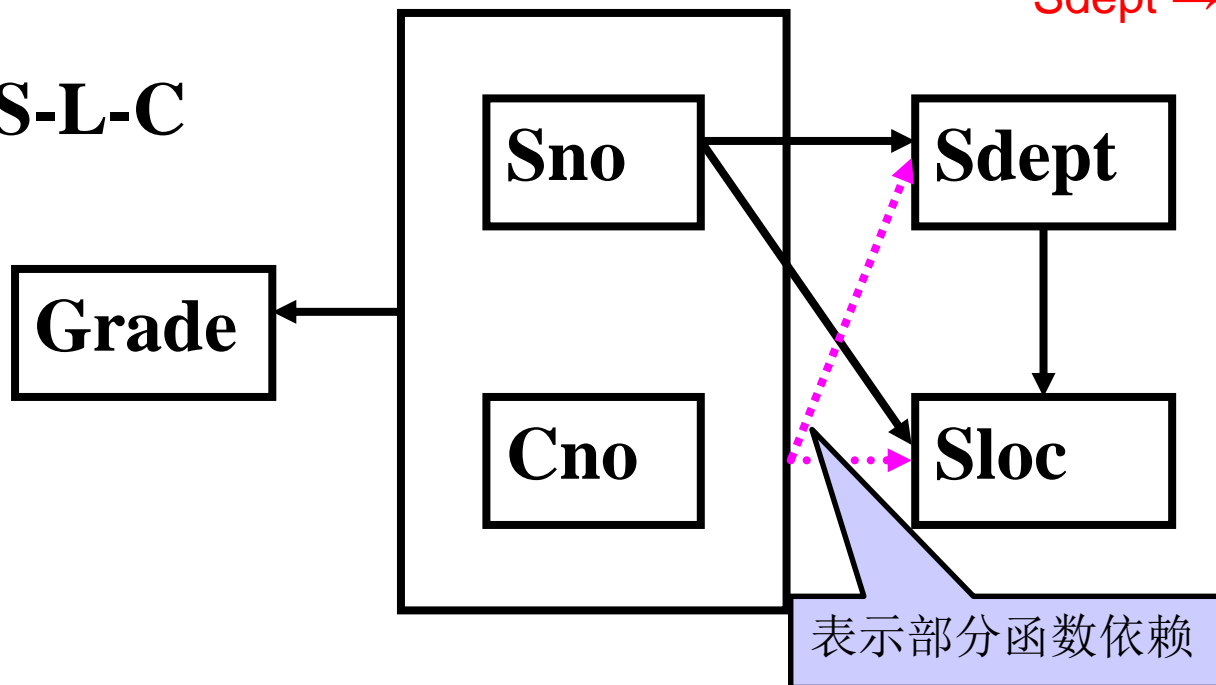
$(Sno, Cno) \xrightarrow{P} Sloc$

$Sdept \rightarrow Sloc$

# 2NF (续)

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

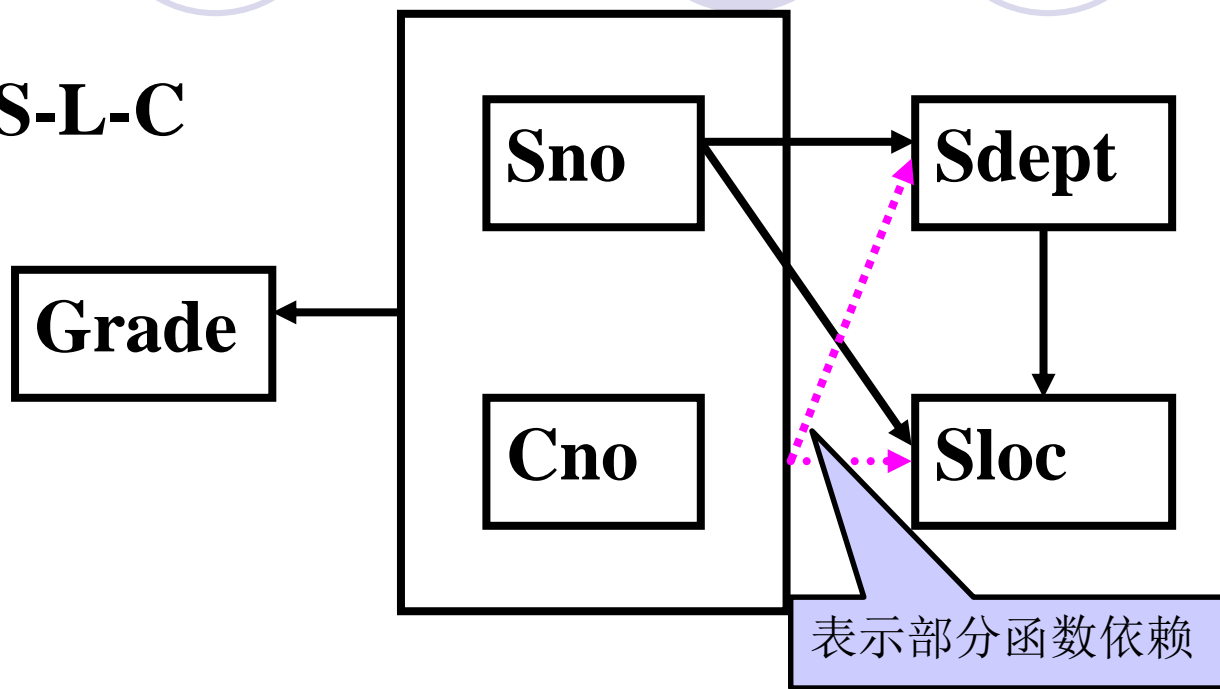
S-L-C



$(Sno, Cno) \xrightarrow{?} (Sno, Cno, Grade, Sdept, Sloc)$   
 $Sno \longrightarrow (Sno, Sdept, Sloc)$   
 $Cno \longrightarrow (Cno)$

## 2NF (续)

S-L-C



- S-L-C的码为(Sno, Cno) 。是否还有其他码？为什么？
- S-L-C满足第一范式。
- 非主属性Sdept和Sloc部分函数依赖于码(Sno, Cno)

# S-L-C不是一个好的关系模式（续）

- (1) 插入异常

- 假设Sno=95102, Sdept=IS, Sloc=HY的学生还未选课，因课程号是主属性，因此该学生的信息无法插入SLC。

- (2) 删除异常

- 假定某个学生本来只选修了3号课程这一门课。现在因身体不适，他连3号课程也不选修了。因课程号是主属性，此操作将导致该学生信息的整个元组都要删除。

关系模式 S-L-C(Sno, Sdept, Sloc, Cno, Grade)

# S-L-C不是一个好的关系模式（续）

- (3) 数据冗余度大

- 如果一个学生选修了10门课程，那么他的Sdept和Sloc值就要重复存储了10次。

- (4) 修改复杂

- 例如学生转系，在修改此学生元组的Sdept值的同时，还可能需要修改住处（Sloc）。如果这个学生选修了K门课，则必须无遗漏地修改K个元组中全部Sdept、Sloc信息。

关系模式 S-L-C(Sno, Sdept, Sloc, Cno, Grade)

# S-L-C不是一个好的关系模式（续）

- 原因

Sdept、Sloc部分函数依赖于码。

$(Sno, Cno) \xrightarrow{P} Sdept$   
 $(Sno, Cno) \xrightarrow{P} Sloc$

- 解决方法

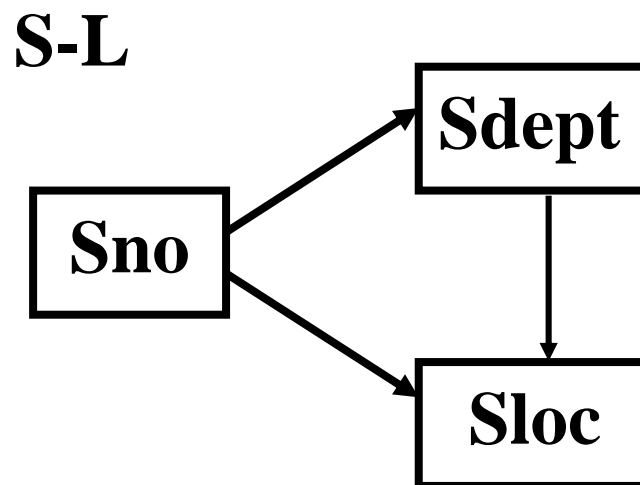
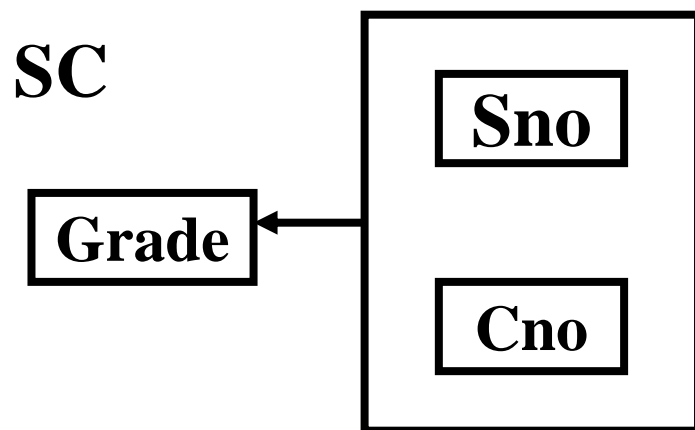
S-L-C分解为两个关系模式，以消除这些部分函数依赖

SC (Sno, Cno, Grade)

S-L (Sno, Sdept, Sloc)

## 2NF (续)

函数依赖图:



- 关系模式SC的码为 (Sno, Cno)
- 关系模式S-L的码为Sno
- 这样非主属性对码都是完全函数依赖

## 2NF (续)

### ● 2NF的定义

定义**6.6** 若 $R \in 1NF$ ，且**每一个非主属性完全**函数依赖于码，则 $R \in 2NF$ 。

例：S-L-C(Sno, Sdept, Sloc, Cno, Grade)  $\in 1NF$

S-L-C(Sno, Sdept, Sloc, Cno, Grade)  $\notin 2NF$

SC (Sno, Cno, Grade)  $\in 2NF$

S-L (Sno, Sdept, Sloc)  $\in 2NF$



## 2NF（续）

- 采用**投影分解法**将一个1NF的关系**分解**为多个2NF的关系，可以在**一定程度上**减轻原1NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- 将一个1NF关系分解为多个2NF的关系，**并不能完全消除**关系模式中的各种异常情况和数据冗余。

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.5 3NF

### ● 3NF的定义

定义6.7 关系模式 $R\langle U, F \rangle$ 中若不存在这样的码 $X$ 、属性组 $Y$ 及非主属性 $Z$  ( $Z \notin Y$ ), 使得 $X \rightarrow Y$ ,  $Y \rightarrow Z$ 成立,  $Y \not\rightarrow X$ , 则称 $R\langle U, F \rangle \in 3NF$ 。

- 若 $R \in 3NF$ , 则每一个非主属性完全依赖于码也不传递依赖于码。

## 6.2.5 3NF

- 证明：若  $R \in 3NF$ ，则每一个非主属性完全依赖于码也不传递依赖于码

根据定义6.7可得 不存在 (码  $X$ ，属性组  $Y$  及非主属性  $Z, Z \notin Y$ )，使得  $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z)$  成立。

因为  $\neg \exists x(P(x)) \equiv (\forall x)(\neg P(x))$ ，上述表述等价于任意的 (码  $X$ ，属性组  $Y$  及非主属性  $Z, Z \notin Y$ )， $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z)$  都不成立

## 6.2.5 3NF

- 证明：若  $R \in 3NF$ ，则每一个非主属性完全依赖于码也不传递依赖于码

任意的(码  $X$ ，属性组  $Y$  及非主属性  $Z$   $Z \not\subseteq Y$ )， $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z)$  都不成立

- (1) 当  $Y \subset X$ ， $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z) \equiv$  (非主属性  $Z$  部分依赖于码  $X$ )，意味着 (非主属性  $Z$  部分依赖于码  $X$ ) 不成立，即非主属性  $Z$  完全依赖于码  $X$

部分依赖的定义：  $X \rightarrow Y$ ， $X$  的子集  $X'(Y) \rightarrow Z$ ， $Z$  对  $X$  部分依赖

## 6.2.5 3NF

- 证明：若  $R \in 3NF$ ，则每一个非主属性完全依赖于码也不传递依赖于码

任意的(码  $X$ ，属性组  $Y$  及非主属性  $Z$   $Z \not\subseteq Y$ )， $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z)$  都不成立

(2) 当  $Y \not\subseteq X$ ， $(X \rightarrow Y, Y \nrightarrow X, Y \rightarrow Z) \equiv$  (非主属性  $Z$  传递依赖于码  $X$ )，意味着(非主属性  $Z$  传递依赖于码  $X$ ) 不成立，即非主属性  $Z$  不传递依赖于码  $X$

传递依赖的定义：  $X \rightarrow Y, Y \not\subseteq X, Y \nrightarrow X, Y \rightarrow Z, Z \not\subseteq Y$ ， $Z$  对  $X$  传递依赖

## 3NF (续)

例：2NF关系模式S-L(Sno, Sdept, Sloc)中

○ 函数依赖：

$Sno \rightarrow Sdept$

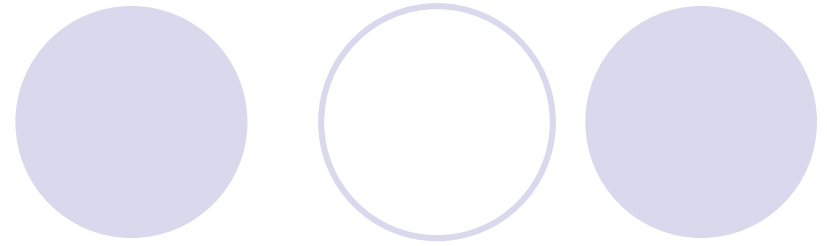
$Sdept \nrightarrow Sno$

$Sdept \rightarrow Sloc$

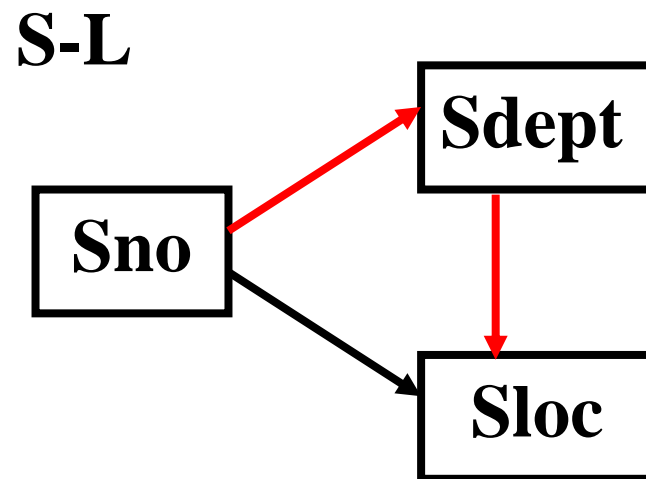
可得：

$Sno \xrightarrow{\text{传递}} Sloc$ ，即S-L中存在**非主属性**对码的**传递**函数依赖， $S-L \notin 3NF$

# 3NF (续)



函数依赖图:





## 3NF (续)

- 解决方法

采用投影分解法，把**S-L**分解为两个关系模式，以消除传递函数依赖：

**S-D (Sno, Sdept)**

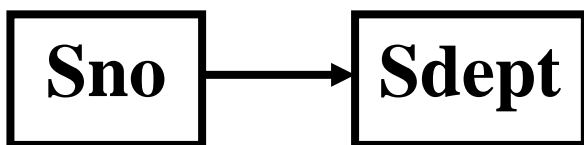
**D-L (Sdept, Sloc)**

**S-D**的码为**Sno**， **D-L**的码为**Sdept**。

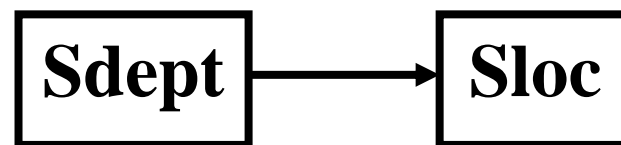
- 分解后的关系模式**S-D**与**D-L**中不再存在传递依赖

## 3NF (续)

S-D的码为Sno, D-L的码为Sdept



**S-D**



**D-L**

❖  $S-L(Sno, Sdept, Sloc) \in 2NF$

$S-L(Sno, Sdept, Sloc) \notin 3NF$

$S-D(Sno, Sdept) \in 3NF$

$D-L(Sdept, Sloc) \in 3NF$

❖ 但是, 原有的函数依赖  $Sno \rightarrow Sloc$  “丢失”

## 3NF（续）

- 采用投影分解法将一个2NF的关系分解为多个3NF的关系，可以在一定程度上解决原2NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- 将一个2NF关系分解为多个3NF的关系后，**仍然不能完全消除**关系模式中的各种异常情况和数据冗余。

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

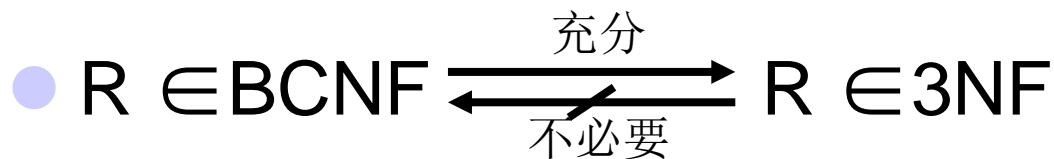
## 6.2.6 BC范式 (BCNF)

- 定义6.8 关系模式 $R\langle U, F \rangle \in 1NF$ , 若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ 时**X必含有码**, 则 $R\langle U, F \rangle \in BCNF$ 。
- 等价于: 每一个**决定因素**(属性集)都包含码

# BCNF (续)

- 若  $R \in \text{BCNF}$

- 所有 **非主属性** 对每一个码都是 **完全** 函数依赖
- 所有的主属性对每一个 **不包含它的码**，也是 **完全** 函数依赖
- 没有任何属性 **完全函数依赖于非码** 的任何一组属性



保证主属性、非主属性不会 **传递依赖** 于码

# BCNF (续)

- 证明：若  $R \in \text{BCNF}$ ，则  $R \in 3\text{NF}$

## 用反证法

- 假设  $R \in 3\text{NF}$  **不成立**，则关系模式  $R$  中 **存在** 这样的码  $X$ ，属性组  $Y$ ，及非主属性  $Z$  ( $Z \not\subseteq Y$ )，使得  $X \rightarrow Y$  ( $Y \nrightarrow X$ )， $Y \rightarrow Z$  成立
  - 因为  $R \in \text{BCNF}$ ，则在  $Y \rightarrow Z$  ( $Z \not\subseteq Y$ ) 条件下， $Y$  必含有码，则  $Y \rightarrow U$ ，进而  $Y \rightarrow X$ ，与假设矛盾，所以  $R \in 3\text{NF}$  成立.
- 后面会有例子：  $R \in 3\text{NF}$ ，但  $R \notin \text{BCNF}$

# BCNF (续)

- [例7] 关系模式SJP (S, J, P)  
S表示学生, J表示课程, P表示名次  
每一门课程没有并列名次

- 函数依赖:  $(S, J) \xrightarrow{F} P$ ;  $(J, P) \xrightarrow{F} S$
- (S, J) 与 (J, P) 都可以作为候选码, 属性相交
- $SJP \in 3NF$ ,
- $SJP \in BCNF$



# BCNF (续)

[例8]在关系模式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。  $T \rightarrow J$  (S, T)  $\xrightarrow{P} J$
- 每门课由若干教师教。  $J \nrightarrow T$
- 某一学生选定某门课, 就确定了一个固定的教师。  
 $(S, J) \xrightarrow{F} T$
- (S, J)和(S, T)都是候选码, 为什么?

# BCNF (续)

- $STJ \in 3NF$

- S, T, J 都是主属性

- 没有任何非主属性对码传递依赖或部分依赖

- $STJ \notin BCNF$

- $T \rightarrow J$

- T 是决定因素，但是 T 不包含码

# BCNF (续)

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

- 虽然  $STJ \in 3NF$ ，但仍然有数据冗余

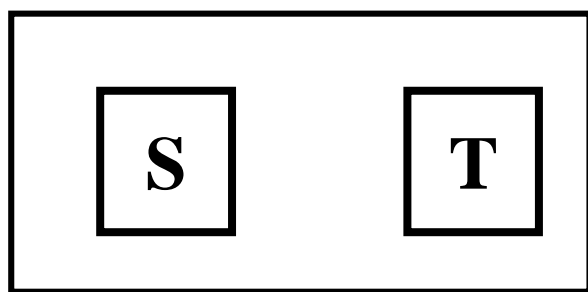
S(学生)	J(课程)	T(教师)
Smith	Database	Mark
Tom	Database	Navathe
Jack	Database	Mark
Smith	OS	Ommar

- 原因在于：存在主属性对不包含它的码部分依赖

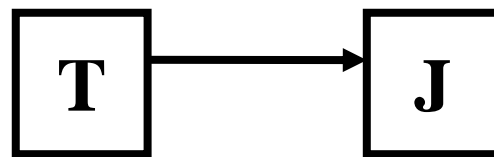
$(S, T) \xrightarrow{P} J$

# BCNF (续)

- 解决方法：将STJ分解为二个关系模式：  
 $ST(S, T) \in BCNF, TJ(T, J) \in BCNF$



ST



TJ

没有**任何属性**对码的部分函数依赖和传递函数依赖

# 3NF与BCNF的关系

●  $R \in \text{BCNF} \xrightleftharpoons[\text{不必要}]{\text{充分}} R \in 3\text{NF}$

- 如果  $R \in 3\text{NF}$ ，且  $R$  只有一个候选码

$$R \in \text{BCNF} \xrightleftharpoons[\text{必要}]{\text{充分}} R \in 3\text{NF}$$

- 总可以在满足**无损**并**保持依赖**的前提下经过分解得到3NF，而BCNF并不总能

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.7 多值依赖

[例9] 学校中某一门课程由多个教师讲授，他们使用相同的一套参考书。每个教员可以讲授多门课程，每种参考书可以供多门课程使用。

# 多值依赖（续）

## ❖ 非关系的二维表

课 程 C	教 员 T	参 考 书 B
物理	$\begin{Bmatrix} \text{李 勇} \\ \text{王 军} \end{Bmatrix}$	$\begin{Bmatrix} \text{普通物理学} \\ \text{光学原理} \\ \text{物理习题集} \end{Bmatrix}$
数学	$\begin{Bmatrix} \text{李 勇} \\ \text{张 平} \end{Bmatrix}$	$\begin{Bmatrix} \text{数学分析} \\ \text{微分方程} \\ \text{高等代数} \end{Bmatrix}$
计算数学	$\begin{Bmatrix} \text{张 平} \\ \text{周 峰} \end{Bmatrix}$	$\begin{Bmatrix} \text{数学分析} \\ \dots \\ \dots \end{Bmatrix}$
$\vdots$	$\vdots$	$\vdots$

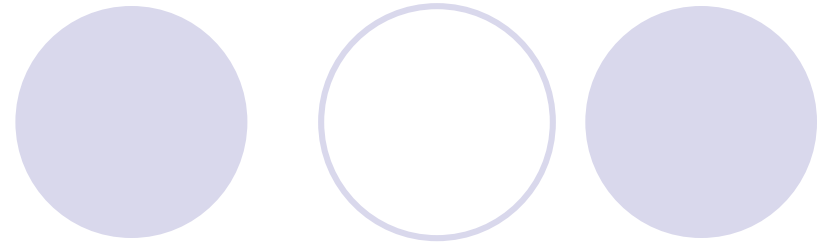


# 多值依赖（续）

❖ 用关系表示Teaching

课程C	教员T	参考书B
物理	李勇	普通物理学
物理	李勇	光学原理
物理	李勇	物理习题集
物理	王军	普通物理学
物理	王军	光学原理
物理	王军	物理习题集
数学	李勇	数学分析
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	数学分析
数学	张平	微分方程
数学	张平	高等代数
...	...	...

## 多值依赖（续）



- Teaching  $\in$  BCNF
- Teaching 具有唯一候选码(C, T, B), 即全码

# 多值依赖（续）



罪魁祸首  
**多值依赖**

## Teaching模式中存在的问题

- (1) 数据冗余度大：对于同一门课，有多少名任课教师，参考书就要存储多少次
- (2) 插入操作复杂：当某一课程增加一名任课教师时，该课程有多少本参考书，就必须插入多少个新元组
- (3) 删除操作复杂：某一门课要去掉一本参考书，该课程有多少名教师，就必须删除多少个元组
- (4) 修改操作复杂：某一门课要修改一本参考书，该课程有多少名教师，就必须修改多少个元组

# 多值依赖（续）

课程C	教员T	参考书B
物理	李勇	普通物理学
物理	李勇	光学原理
物理	李勇	物理习题集
物理	王军	普通物理学
物理	王军	光学原理
物理	王军	物理习题集

## ● 定义6.9

设 $R(U)$ 是一个属性集 $U$ 上的一个关系模式， $X$ 、 $Y$ 和 $Z$ 是 $U$ 的子集，并且 $Z=U-X-Y$ 。关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立，当且仅当对 $R(U)$ 的任一关系 $r$ ，给定的一对 $(x, z)$ 值，有一组 $Y$ 的值，这组值仅仅决定于 $x$ 值而与 $z$ 值无关

换句话说: 对于另外一对 $(x, z')$ 值，依然是这组 $Y$ 值

## ● 例 Teaching (C, T, B)

对于一个（物理，光学原理）有一组 $T$ 值{李勇，王军}，这组值仅由课程 $C$ 上的值（物理）决定，对于另一个（物理，物理习题集）对应的 $T$ 值仍是{李勇，王军}，因此 $T$ 多值依赖于 $C$ ，即 $C \twoheadrightarrow T$

# 多值依赖（续）

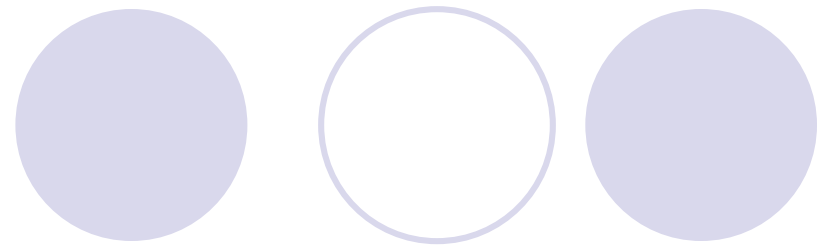
	X 课程C	Y 教员T	Z 参考书B
t	物理	李 勇	普通物理学
	物理	李 勇	光学原理
w	物理	李 勇	物理习题集
v	物理	王 军	普通物理学
	物理	王 军	光学原理
s	物理	王 军	物理习题集

- 多值依赖的另一个等价的形式化的定义：

在 $R(U)$ 的任一关系 $r$ 中，如果存在元组 $t, s$ 使得

$t[X]=s[X]$ ，那么就必然存在元组 $w, v \in r$ ，（ $w, v$ 可以与 $s, t$ 相同），使得 $w[X]=v[X]=t[X]=s[X]$ ，而 $t[Y]=w[Y]$ ， $s[Y]=v[Y]$ ， $s[Z]=w[Z]$ ， $t[Z]=v[Z]$ （即交换 $s, t$ 元组的 $Y$ 值所得的两个新元组必在 $r$ 中），则 $Y$ 多值依赖于 $X$ ，记为 $X \twoheadrightarrow Y$ 。这里， $X, Y$ 是 $U$ 的子集， $Z=U-X-Y$ 。

# 多值依赖（续）



- 平凡多值依赖和非平凡的多值依赖
  - 若 $X \twoheadrightarrow Y$ ，而 $Y \subseteq X$ 或者 $Z = \emptyset$ ，则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖
  - 否则称 $X \twoheadrightarrow Y$ 为非平凡的多值依赖

## 多值依赖 (续)

### [例10] 关系模式WSC (W, S, C)

- W表示仓库，S表示保管员，C表示商品
- 假设每个仓库有若干个保管员，有若干种商品
- 每个保管员保管所在的仓库的所有商品
- 每种商品被**所在仓库的**所有保管员保管

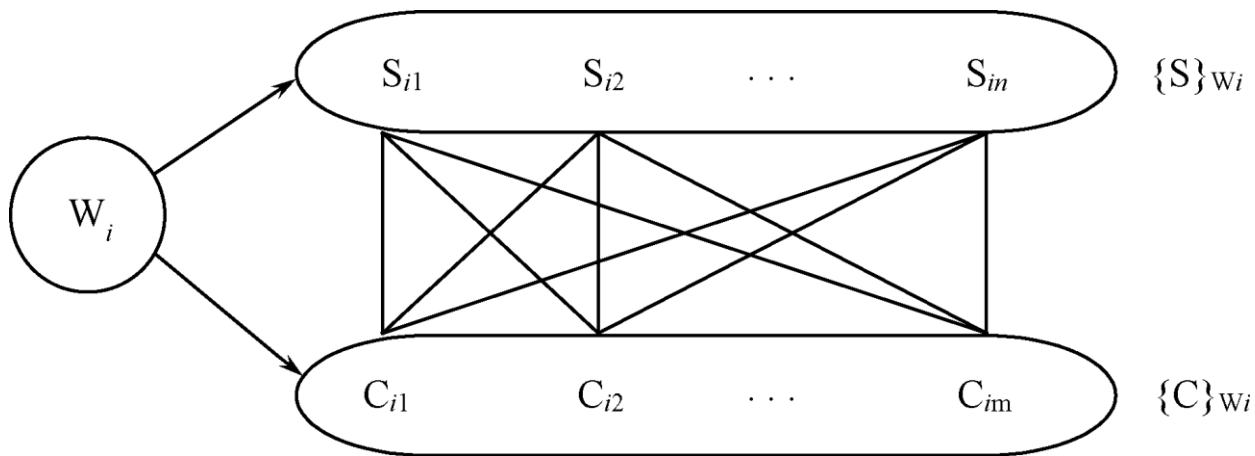
# 多值依赖（续）

	X	Y	Z
	W	S	C
t	W1	S1	C1
w	W1	S1	C2
	W1	S1	C3
v	W1	S2	C1
s	W1	S2	C2
	W1	S2	C3
	W2	S3	C4
	W2	S3	C5
	W2	S4	C4
	W2	S4	C5



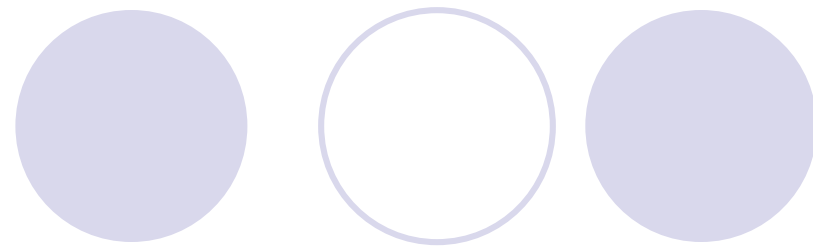
# 多值依赖（续）

用下图表示这种对应



$$W \twoheadrightarrow S \text{ 且 } W \twoheadrightarrow C$$

# 多值依赖的性质



## (1) 多值依赖具有对称性

若 $X \twoheadrightarrow Y$ , 则 $X \twoheadrightarrow Z$ , 其中 $Z = U - X - Y$

## (2) 多值依赖具有传递性

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

## (3) 函数依赖是多值依赖的特殊情况

若 $X \rightarrow Y$ , 则 $X \twoheadrightarrow Y$

(4) 若 $X \twoheadrightarrow Y$ ,  $X \twoheadrightarrow Z$ , 则 $X \twoheadrightarrow YZ$ 。

(5) 若 $X \twoheadrightarrow Y$ ,  $X \twoheadrightarrow Z$ , 则 $X \twoheadrightarrow Y \cap Z$ 。

(6) 若 $X \twoheadrightarrow Y$ ,  $X \twoheadrightarrow Z$ , 则 $X \twoheadrightarrow Y - Z$ ,  $X \twoheadrightarrow Z - Y$ 。

# 多值依赖与函数依赖的区别

(1) 多值依赖的有效性与属性集的范围有关

若  $X \twoheadrightarrow Y$  在  $U$  上成立，则在  $W (XY \subseteq W \subseteq U)$  上成立；

反之不一定

	U			
	X	Y	A	B
t	x	y1	a1	b1
s	x	y2	a2	b2
w	x	y1	a2	b2
v	x	y2	a1	b1

	W		
	X	Y	A
t	x	y1	a1
s	x	y2	a2
w	x	y1	a2
v	x	y2	a1

# 多值依赖与函数依赖的区别

- (2) a. 若函数依赖  $X \rightarrow Y$  在  $R(U)$  上成立, 则对于任何  $Y' \subset Y$  均有  $X \rightarrow Y'$  成立
- b. 多值依赖  $X \twoheadrightarrow Y$  若在  $R(U)$  上成立, 不能断言对于任何  $Y' \subset Y$  有  $X \twoheadrightarrow Y'$  成立

	X		Y		Z
	A	B	C	D	
t	a	b1	c1	d1	
s	a	b2	c2	d2	
w	a	b1	c1	d2	
v	a	b2	c2	d1	

	X	Y'	Z'	
	A	B	C	D
t	a	b1	c1	d1
s	a	b2	c2	d2
w	a	b1	c1	d2
v	a	b2	c2	d1

## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.8 4NF

- **定义6.10** 关系模式 $R\langle U, F \rangle \in 1NF$ ，如果对于R的每个**非平凡**多值依赖 $X \twoheadrightarrow Y$  ( $Y \not\subseteq X$ )，X都含有码，则 $R \in 4NF$ 。
- 如果 $R \in 4NF$ ， 则 $R \in BCNF$ 
  - 不允许有**非平凡**且**非函数依赖**的**多值依赖**
  - 允许的非平凡多值依赖是**函数依赖**

X都含有码， $X \rightarrow Y$

# 4NF (续)

例: Teaching(C,T,B)  $\notin$  4NF

存在非平凡的多值依赖  $C \twoheadrightarrow T$ , 且  $C$  不是码

■ 用投影分解法把 Teaching 分解为如下两个关系模式:

$$CT(C, T) \in 4NF$$

$$CB(C, B) \in 4NF$$

$C \twoheadrightarrow T$ ,  $C \twoheadrightarrow B$  是平凡多值依赖, 为什么?

## 4NF (续)

例:  $WSC(W, S, C) \notin 4NF$

存在**非平凡的**多值依赖 $W \twoheadrightarrow S$ , 且 $W$ 不是码

■ 用投影分解法把 $WSC$ 分解为如下两个关系模式:

$$WS(W, S) \in 4NF$$

$$WC(W, C) \in 4NF$$

$W \twoheadrightarrow S$ ,  $W \twoheadrightarrow C$ 是平凡多值依赖



## 6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

## 6.2.9 规范化小结

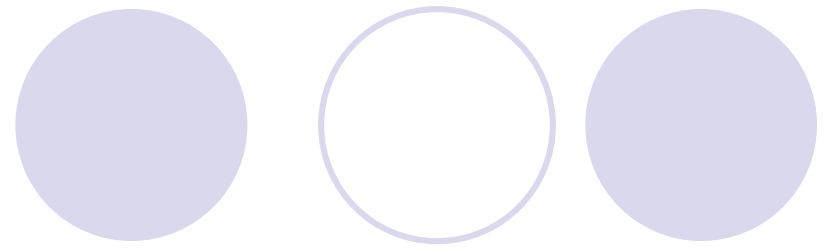
- 关系数据库的规范化理论是数据库逻辑设计的工具
- 目的：尽量消除插入、删除异常，修改复杂，数据冗余
- 基本思想：逐步消除数据依赖中**不合适**的部分
  - 各关系模式达到某种程度的“分离”
  - 采用“**一事一地**”的模式设计原则  
让一个关系描述一个概念、一个实体或者实体间的一种联系。若多于一个概念就把它“分离”出去
  - 所谓规范化实质上是概念的**单一化**

# 规范化小结（续）

## ● 关系模式规范化的基本步骤



# 规范化小结（续）



- 不能说规范化程度越高的关系模式就越好
- 在设计数据库模式结构时，必须对现实世界的实际情况和用户应用需求作进一步分析，确定一个合适的、能够反映现实世界的模式
- 上面的规范化步骤可以在其中任何一步终止

# 第六章 关系数据理论

**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**

## 6.3 数据依赖的公理系统

- 逻辑蕴含

**定义6.11** 对于满足一组函数依赖  $F$  的关系模式  $R$   $\langle U, F \rangle$ ，其任何一个关系  $r$ ，若函数依赖  $X \rightarrow Y$  都成立，（即  $r$  中任意两元组  $t, s$ ，若  $t[X] = s[X]$ ，则  $t[Y] = s[Y]$ ），则称  $F$  逻辑蕴含  $X \rightarrow Y$

# 1. Armstrong公理系统

关系模式 $R \langle U, F \rangle$ 来说有以下的推理规则：

- A1. 自反律 (Reflexivity) : 若  $Y \subseteq X \subseteq U$ , 则  $X \rightarrow Y$  为  $F$  所蕴含。
- A2. 增广律 (Augmentation) : 若  $X \rightarrow Y$  为  $F$  所蕴含, 且  $Z \subseteq U$ , 则  $XZ \rightarrow YZ$  为  $F$  所蕴含。
- A3. 传递律 (Transitivity) : 若  $X \rightarrow Y$  及  $Y \rightarrow Z$  为  $F$  所蕴含, 则  $X \rightarrow Z$  为  $F$  所蕴含。

## 2. 导出规则

1. 根据A1, A2, A3这三条推理规则可以得到下面三条推理规则:

○ 合并规则: 由 $X \rightarrow Y$ ,  $X \rightarrow Z$ , 有 $X \rightarrow YZ$ 。

(A2, A3)

○ 伪传递规则: 由 $X \rightarrow Y$ ,  $WY \rightarrow Z$ , 有 $XW \rightarrow Z$ 。

(A2, A3)

○ 分解规则: 由 $X \rightarrow Y$ 及  $Z \subseteq Y$ , 有 $X \rightarrow Z$ 。

(A1, A3)

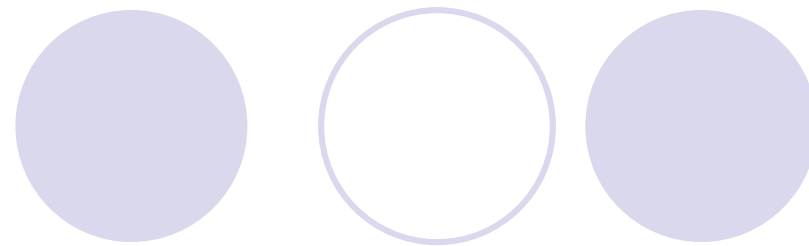


# 导出规则

2.根据合并规则和分解规则，可得引理6.1

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 ( $i=1, 2, \dots, k$ )

### 3. 函数依赖闭包



**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的闭包

$R\langle U, F \rangle, U=\{X, Y, Z\} \quad F=\{X \rightarrow Y, Y \rightarrow Z\}$

$F^+=\{$

$X \rightarrow \phi,$	$Y \rightarrow \phi,$	$Z \rightarrow \phi,$	$XY \rightarrow \phi,$	$XZ \rightarrow \phi,$	$YZ \rightarrow \phi,$	$XYZ \rightarrow \phi,$
$X \rightarrow X,$	$Y \rightarrow Y,$	$Z \rightarrow Z,$	$XY \rightarrow X,$	$XZ \rightarrow X,$	$YZ \rightarrow Y,$	$XYZ \rightarrow X,$
$X \rightarrow Y,$	$Y \rightarrow Z,$		$XY \rightarrow Y,$	$XZ \rightarrow Y,$	$YZ \rightarrow Z,$	$XYZ \rightarrow Y,$
$X \rightarrow Z,$	$Y \rightarrow YZ,$		$XY \rightarrow Z,$	$XZ \rightarrow Z,$	$YZ \rightarrow YZ,$	$XYZ \rightarrow Z,$
$X \rightarrow XY,$			$XY \rightarrow XY,$	$XZ \rightarrow XY,$		$XYZ \rightarrow XY,$
$X \rightarrow XZ,$			$XY \rightarrow YZ,$	$XZ \rightarrow XZ,$		$XYZ \rightarrow YZ,$
$X \rightarrow YZ,$			$XY \rightarrow XZ,$	$XZ \rightarrow XY,$		$XYZ \rightarrow XZ,$
$X \rightarrow XYZ,$			$XY \rightarrow XYZ,$	$XZ \rightarrow XYZ,$		$XYZ \rightarrow XYZ$

$\}$

# 第六章 关系数据理论

**6.1 问题的提出**

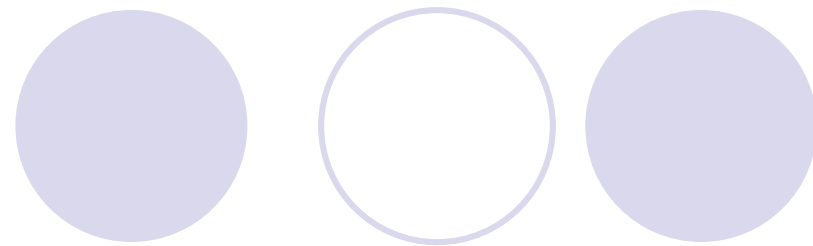
**6.2 规范化**

**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



- 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

# 模式的分解（续）

**定义6.16** 关系模式 $R\langle U, F \rangle$ 的一个分解：

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

其中 $U = \bigcup_{i=1}^n U_i$ ，且不存在 $U_i \subseteq U_j, 1 \leq i, j \leq n$ ， $F_i$ 为 $F$ 在 $U_i$ 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

三种模式分解等价的定义：

1. 分解具有**无损连接性**(lossless join)
2. 分解要**保持函数依赖**(preserve FD)
3. 分解既要保持函数依赖，又要具有无损连接性

# 具有无损连接性的模式分解

- 关系模式 $R\langle U, F \rangle$ 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 $R$ 与 $R_1$ 、 $R_2$ 、...、 $R_n$  **自然连接的结果相等**，则称关系模式 $R$ 的这个分解 $\rho$ 具有**无损连接性 (Lossless join)**
- 具有无损连接性的分解**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



# 保持函数依赖的模式分解

设关系模式 $R\langle U, F \rangle$ 被分解为若干个关系模式

$$R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$$

（其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$ ，且不存在 $U_i \subseteq U_j$ ， $F_i$ 为 $F$ 在 $U_i$ 上的投影），若 $F^+ = (U_{i=1}^n F_i)^+$ ，则称关系模式 $R$ 的这个分解是保持函数依赖的

# 模式的分解（续）

例：S-L(Sno, Sdept, Sloc)

$F = \{ Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc \}$

$S-L \in 2NF$

分解方法可以有多种：

1. 将S-L分解为三个关系模式：  
SN(Sno)  
SD(Sdept)    即不无损，也不依赖保持  
SO(Sloc)
2. 将SL分解为下面二个关系模式：  
NL(Sno, Sloc)    即不无损，  
DL(Sdept, Sloc)    也不依赖保持

如果两个关系没有公共属性，自然连接就退化为笛卡尔积

# 模式的分解（续）

第3种分解方法具有无损连接性

$ND(Sno, Sdept) \quad NL(Sno, Sloc)$

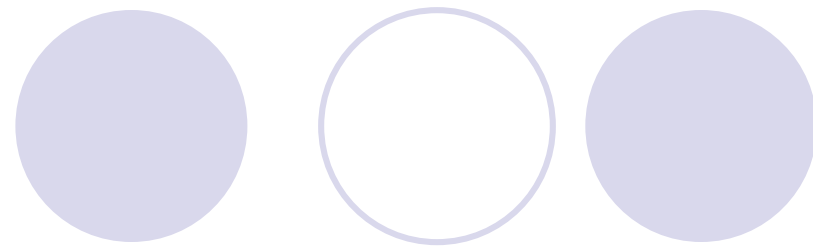
因为  $ND \bowtie NL \equiv S-L$

问题:这种分解方法**没有保持原关系中的函数依赖**

○ S-L中的函数依赖  $Sdept \rightarrow Sloc$  没有投影到关系模式

ND、NL上, 并且也无法由ND、NL上的依赖经过推导得出

# 模式的分解（续）



4. 将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法既保持了函数依赖，又具有无损  
连接

# 模式的分解（续）

- 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；同样，保持函数依赖的分解也不一定具有无损连接性。

# 第六章 关系数据理论

**6.1 问题的提出**

**6.2 规范化**

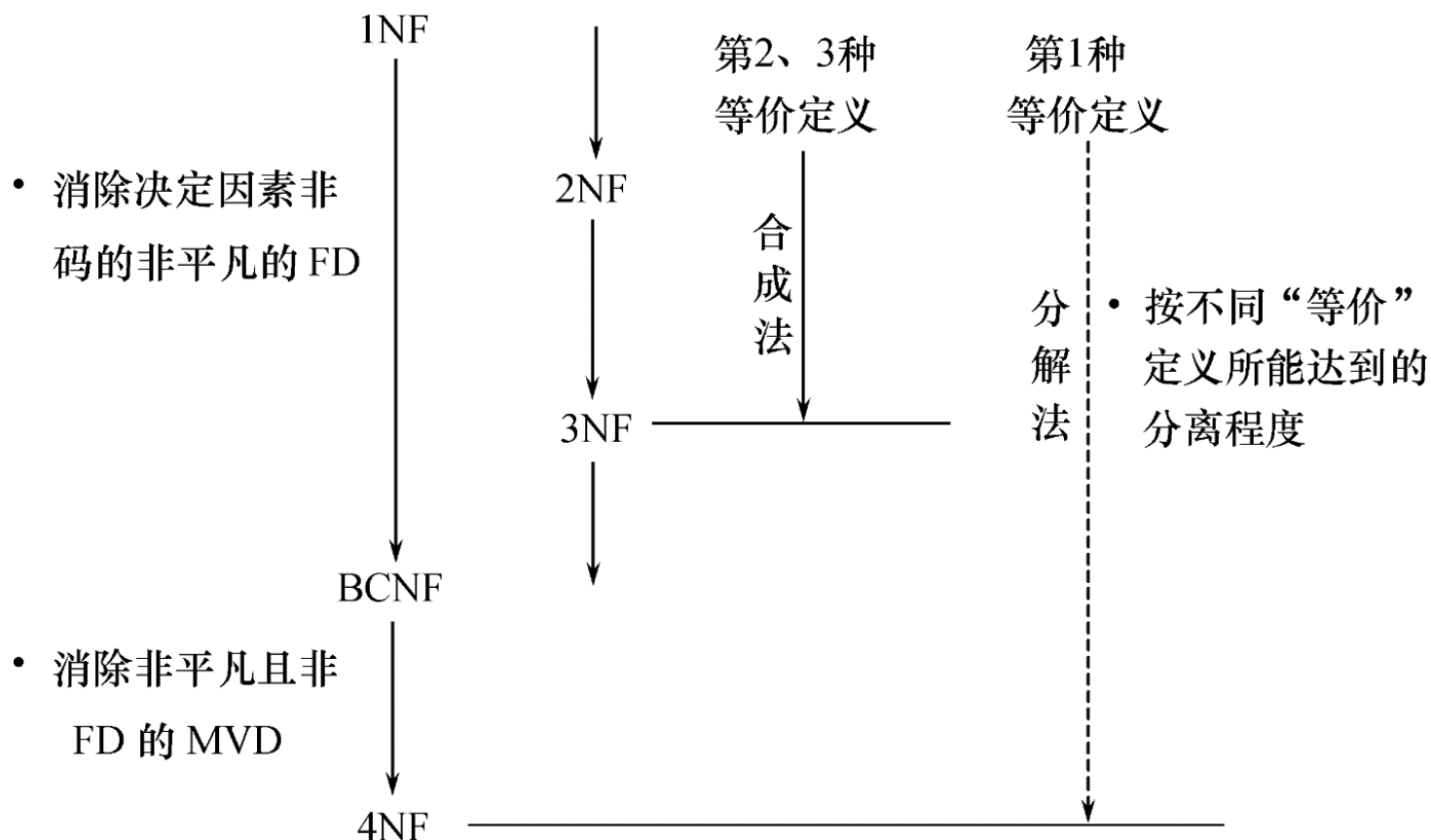
**6.3 数据依赖的公理系统**

**\*6.4 模式的分解**

**6.5 小结**

## 6.5 小结

关系模式的规范化，其基本思想：



## 小结(续)

- 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**
- 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