

# 第六章 关系数据理论



## ❖ 教学目的

- 本章主要讨论如何进行关系数据库的逻辑设计。首先介绍函数依赖的概念，然后利用函数依赖和其他类型的依赖定义范式，并给出利用**Armstrong**公理系统确定范式级别的方法，最后介绍一些将关系模式分解为更高级范式的模式分解算法。

## ❖ 主要内容

- 关系模式的设计问题，函数依赖，范式，函数依赖的推理规则，模式分解

## ❖ 重点

- 函数依赖，范式及它们之间的关系，模式分解算法



# 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

- 数据库的逻辑设计问题
  - 给出一组数据，如何构造一个适合于它们的数据模式？
- 关系数据理论是关系数据库逻辑设计的工具
  - 关系模型有严格的数学理论基础
  - 可以向其它数据模型转换



- 关系模式的优劣分析

| 学号    | 活动  | 费用  |
|-------|-----|-----|
| 99001 | 羽毛球 | 30  |
| 99002 | 排球  | 50  |
| 99003 | 健美  | 100 |
| 99004 | 游泳  | 100 |
| 99005 | 游泳  | 100 |
| 99001 | 太极剑 | 100 |
| 99004 | 太极剑 | 100 |
| 99002 | 太极拳 | 60  |
| 99004 | 太极拳 | 60  |

- ❖ 信息的不可表示问题
  - 1) 插入异常
  - 2) 删除异常
- ❖ 信息的冗余问题
  - 3) 数据冗余
  - 4) 更新异常



# 解决问题的方法：模式分解

- 解决之道：分解! 分解!! 再分解!!!
- 按“一事一地”的模式设计原则进行模式的分解。让一个关系模式尽量只描述一个概念、一个实体或一种实体之间的联系---概念单一!

| 学号    | 活动  |
|-------|-----|
| 99001 | 羽毛球 |
| 99002 | 排球  |
| 99003 | 健美  |
| 99004 | 游泳  |
| 99005 | 游泳  |
| 99001 | 太极剑 |
| 99004 | 太极剑 |
| 99002 | 太极拳 |
| 99004 | 太极拳 |

| 活动  | 费用  |
|-----|-----|
| 羽毛球 | 30  |
| 排球  | 50  |
| 健美  | 100 |
| 游泳  | 100 |
| 太极剑 | 100 |
| 太极拳 | 60  |



# 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: 属性间数据的依赖关系集合



# 什么是数据依赖

## 数据依赖

- 一个关系内部属性与属性之间的约束关系

类似数学中的函数 $y=f(x)$ ,自变量 $x$ 确定后,相应的函数值 $y$ 也就被唯一地确定.

比如 $\text{student}(\text{sno}, \text{sname}, \text{sdept}, \text{sage}, \text{ssex})$  中,  $\text{sname}=f(\text{sno})$ ,  $\text{sdept}=f(\text{sno})$

而 $\text{sname} \neq f(\text{sdept})$

- 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

假设学生不重名,则 $\text{sdept}=f(\text{sname})$





# 什么是数据依赖（续）

## 数据依赖的类型

- 函数依赖（Functional Dependency，简记为FD）
- 多值依赖（Multivalued Dependency，简记为MVD）
- 其他(如连接依赖-5NF)



# 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：

$$R(U, F)$$

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



# 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)、课程名 (Cname)

成绩 (Grade)

单一的关系模式： Student <U、F>

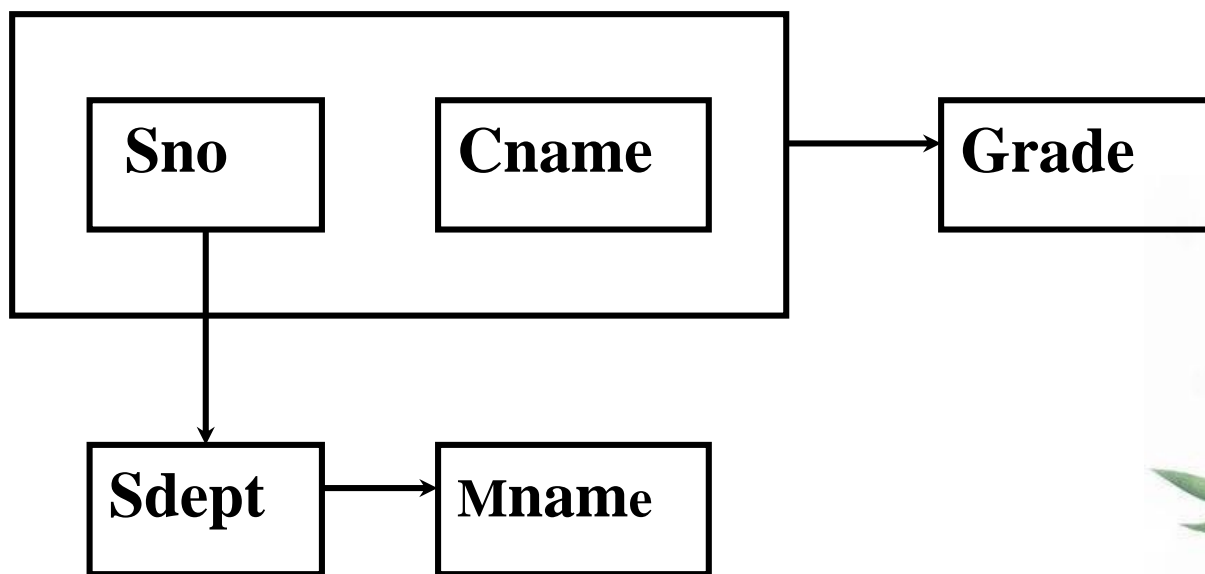
$U = \{ Sno, Sdept, Mname, Cname, Grade \}$



## 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:

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



## 关系模式Student<U, F>中存在的问题

1. 数据冗余太大
2. 更新异常 (Update Anomalies)
3. 插入异常 (Insertion Anomalies) (新开设课没人选就无法录入数据库)
4. 删除异常 (Deletion Anomalies)

| sno | sdept | mname | cname | grade |
|-----|-------|-------|-------|-------|
| s1  | 计算机   | 张明    | 数据库   | 95    |
| s2  | 计算机   | 张明    | 数据库   | 90    |
| s3  | 计算机   | 张明    | 数据库   | 88    |
| s4  | 计算机   | 张明    | 数据库   | 70    |



# 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



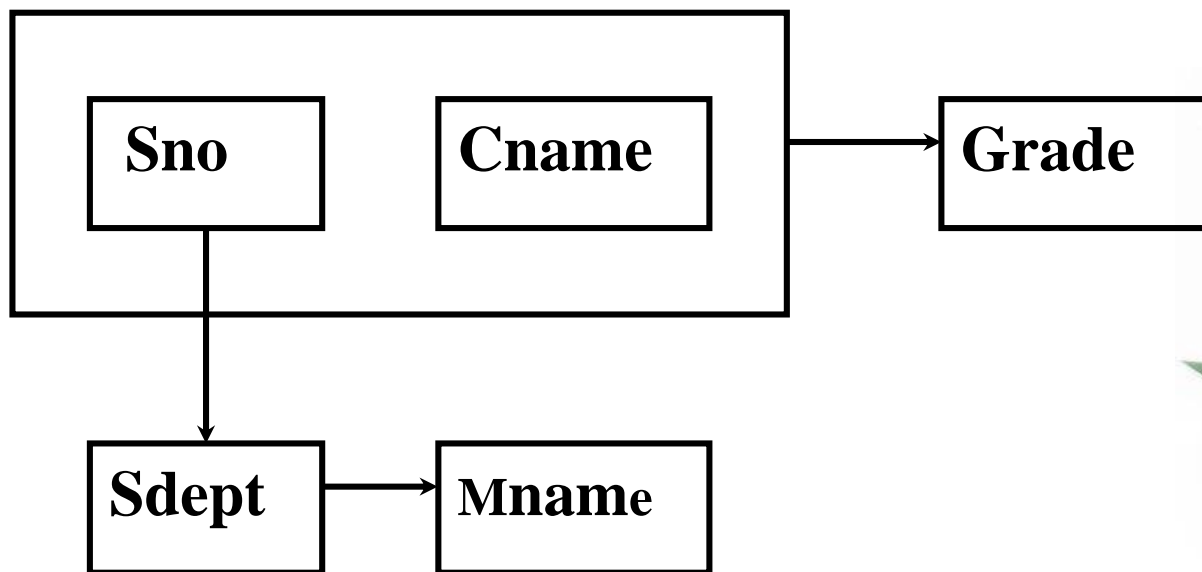
# 分解关系模式

- 把这个单一模式分成3个关系模式：

$S(Sno, Sdept, Sno \rightarrow Sdept)$  ;

$SC(Sno, Cname, Grade, (Sno, Cname) \rightarrow Grade)$  ;

$DEPT(Sdept, Mname, Sdept \rightarrow Mname)$



# 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 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

student

| Sno   | Sname | Ssex | Sage | Sdept |
|-------|-------|------|------|-------|
| 95001 | 李勇    | 男    | 20   | CS    |
| 95002 | 刘晨    | 女    | 19   | IS    |
| 95003 | 王敏    | 女    | 18   | MA    |
| 95004 | 张立    | 男    | 19   | IS    |

**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 $X \rightarrow Y$ 。



# 函数依赖

| A  | B  | C  | D  |
|----|----|----|----|
| a1 | b1 | c1 | d1 |
| a1 | b2 | c1 | d2 |
| a2 | b2 | c2 | d2 |
| a2 | b3 | c2 | d3 |
| a3 | b3 | c2 | d4 |

- 检验： $A \rightarrow C$ ?  $C \rightarrow A$ ?  $AB \rightarrow D$ ?
- 辨别：
  - 满足依赖的关系：依赖在模式的某个关系实例上成立
  - 模式上成立的依赖：依赖在模式的所有关系实例上都成立



# 函数依赖的确定

- 函数依赖是语义范畴的概念。只能根据属性对现实世界的描述及在现实世界中的关系确定函数依赖，是一个断言，不能证明。如 “**name**→**age**”
- 数据库设计者可以对现实世界作强制的规定。如规定不能重名。



| 工厂名 | 产品号 | 产品名 | 车间名 | 地点 | 单价  |
|-----|-----|-----|-----|----|-----|
| w1  | p1  | m1  | j1  | d1 | 100 |
| w2  | p2  | m2  | j2  | d3 | 80  |
| w2  | p3  | m3  | j3  | d2 | 80  |
| w2  | p1  | m1  | j3  | d2 | 100 |
| w2  | p2  | m2  | j4  | d3 | 80  |
| w3  | p1  | m1  | j5  | d4 | 75  |
| w3  | p2  | m2  | j5  | d4 | 90  |

**R表存在哪些函数依赖？**



## 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：  $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$



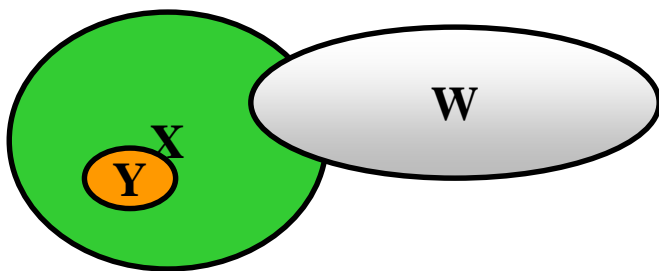


## 平凡函数依赖与非平凡函数依赖（续）

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



- 设 $X, Y, W$ 为关系 $R$ 的三个属性组,属性关系如下图所示,问 $X \longrightarrow Y, X \longrightarrow W, W \longrightarrow Y$ 属于何种依赖.



### 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 $X' \not\rightarrow Y$ , 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 $X \xrightarrow{P} Y$ 。

[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

**定义6.3** 在 $R(U)$ 中, 如果 $X \rightarrow Y$ ,  $(Y \not\subseteq X)$ ,  $Y \not\rightarrow X$   $Y \rightarrow Z$ , 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 $\text{Student}(\text{Sno}, \text{Sdept}, \text{Mname})$ 中, 有:

$\text{Sno} \rightarrow \text{Sdept}$ ,  $\text{Sdept} \rightarrow \text{Mname}$

$\text{Mname}$ 传递函数依赖于 $\text{Sno}$



## 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 规范化小结



## 6.2.2 码

**定义6.4** 设 $K$ 为 $R\langle U, F \rangle$ 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）。

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



候选码

候选码

主码

| 学号   | 身份证号        | 姓名 | 性别 | 系别 |
|------|-------------|----|----|----|
| 0101 | 4201111985  | 张  | 男  | CS |
| 0102 | 36020211986 | 李  | 女  | CS |
| 0203 | 3671111985  | 赵  | 男  | MA |
|      |             |    |    |    |

| 系号 | 系名  |
|----|-----|
| CS | 计算机 |
| EN | 英语  |
| MA | 数学系 |
|    |     |

外码

主码

主码

| 学号   | 课号    | 成绩 |
|------|-------|----|
| 0101 | CS145 | 88 |
| 0101 | CS148 | 90 |
| 0102 | CS180 | 87 |
| 0203 | CS145 | 78 |

| 课号    | 课名   |
|-------|------|
| CS145 | 数据库  |
| CS148 | 操作系统 |
| CS180 | 数据结构 |
|       |      |

# 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**）
  - 不包含在任何码中的属性称为非主属性（**Nonprime attribute**）或非主属性（**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(Sno, Sdept, Sage), 单个属性Sno是码

SC (Sno, 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

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  $SC$  (Sno, Cno, Grade) 中，Sno不是码，但Sno是关系模式  $S$  (Sno, Sdept, Sage) 的码，则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 规范化小结

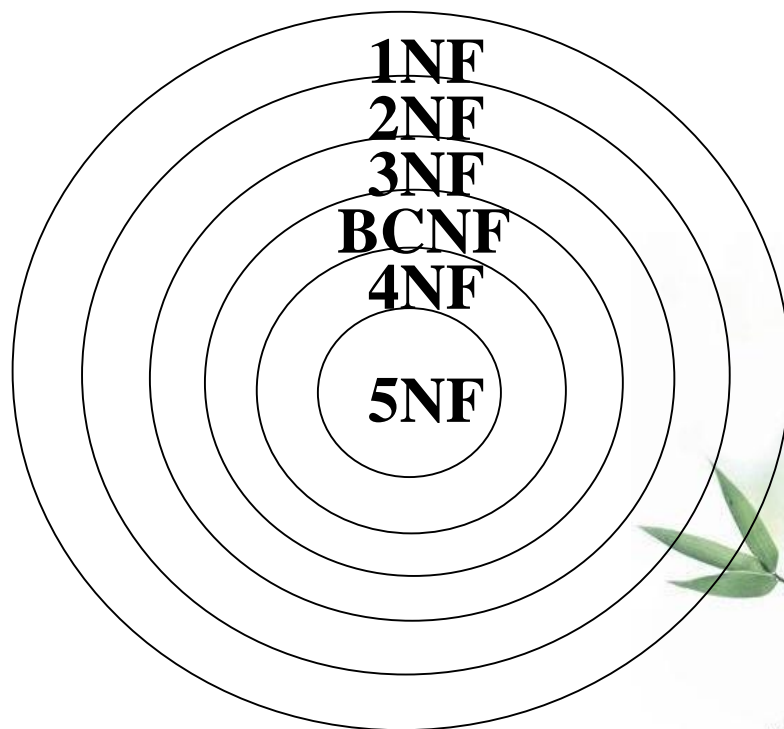


## 6.2.3 范式

- 定义

- 范式是对关系的不同数据依赖程度的要求，即符合某种级别的关系模式的集合，满足不同程度要求的为不同范式。
- 通过模式分解将一个低级范式转换为若干个高级范式的过程称作规范化（概念的纯粹化）

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



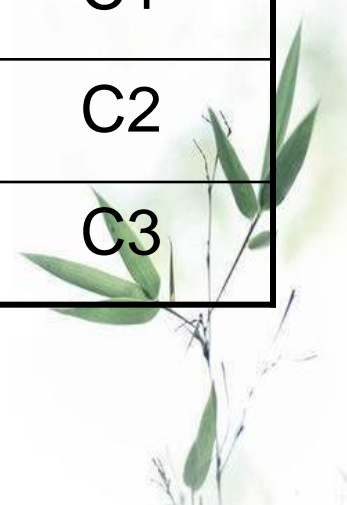
# 1NF

- 定义

关系中每一分量不可再分。即不能以集合、序列等作为属性值

| S# | C#           |
|----|--------------|
| S1 | {C1, C2, C3} |

| S# | C# |
|----|----|
| S1 | C1 |
| S1 | C2 |
| S1 | C3 |



# 1NF

- 原子粒度的选择

- 分量是否需要再分，与具体应用有关。如果用到值的一部分，则需要进一步分割

只是查询  
出生日期

| 姓名 | 生日      |
|----|---------|
| 王军 | 68.7.10 |
| 张立 | 69.7.10 |
| 李明 | 80.3.28 |

比较两人生  
肖是否相同

| 姓名 | 年  | 月日   |
|----|----|------|
| 王军 | 68 | 7.10 |
| 张立 | 69 | 7.10 |
| 李明 | 80 | 3.28 |

比较两人生  
日是否相同

| 姓名 | 年  | 月 | 日  |
|----|----|---|----|
| 王军 | 68 | 7 | 10 |
| 张立 | 69 | 7 | 10 |
| 李明 | 80 | 3 | 28 |

# 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

[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：

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

$Sno \rightarrow Sdept$

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

$Sno \rightarrow Sloc$

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

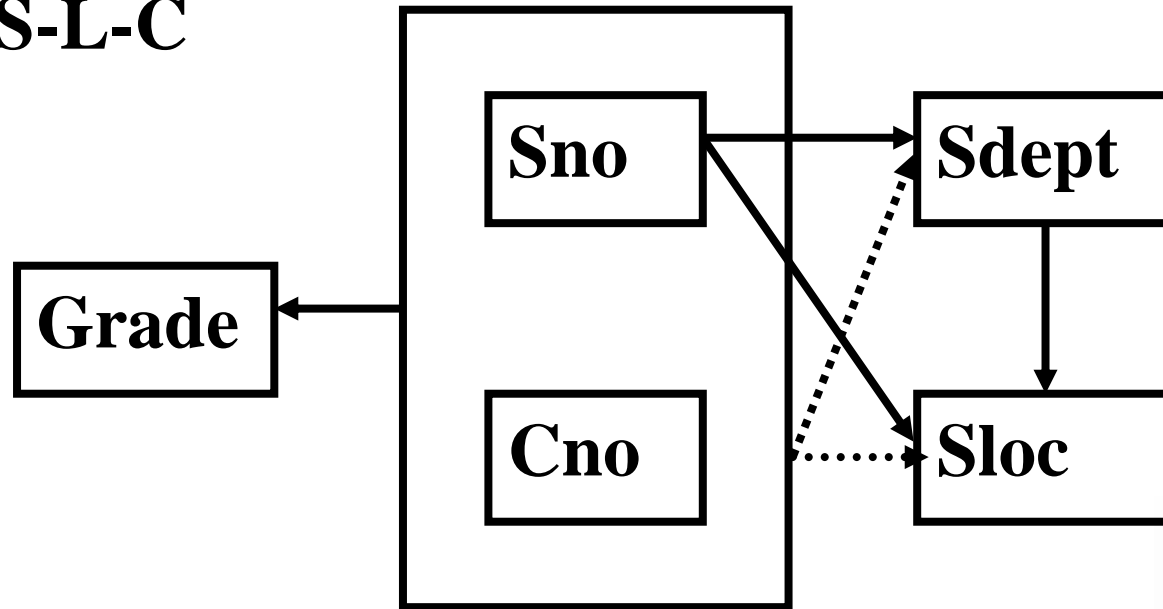
$Sdept \rightarrow Sloc$





## 2NF (续)

**S-L-C**



- S-L-C的码为(Sno, Cno)
- 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）

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

## 不良特性

- 插入异常：如果学生没有选课，关于他的个人信息及所在系的信息就无法插入
- 删除异常：如果删除学生的选课信息，则有关他的个人信息及所在系的信息也随之删除了
- 更新异常：如果学生转系，若他选修了k门课，则需要修改k次
- 数据冗余：如果一个学生选修了k门课，则有关他的所在系的信息重复



# 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。

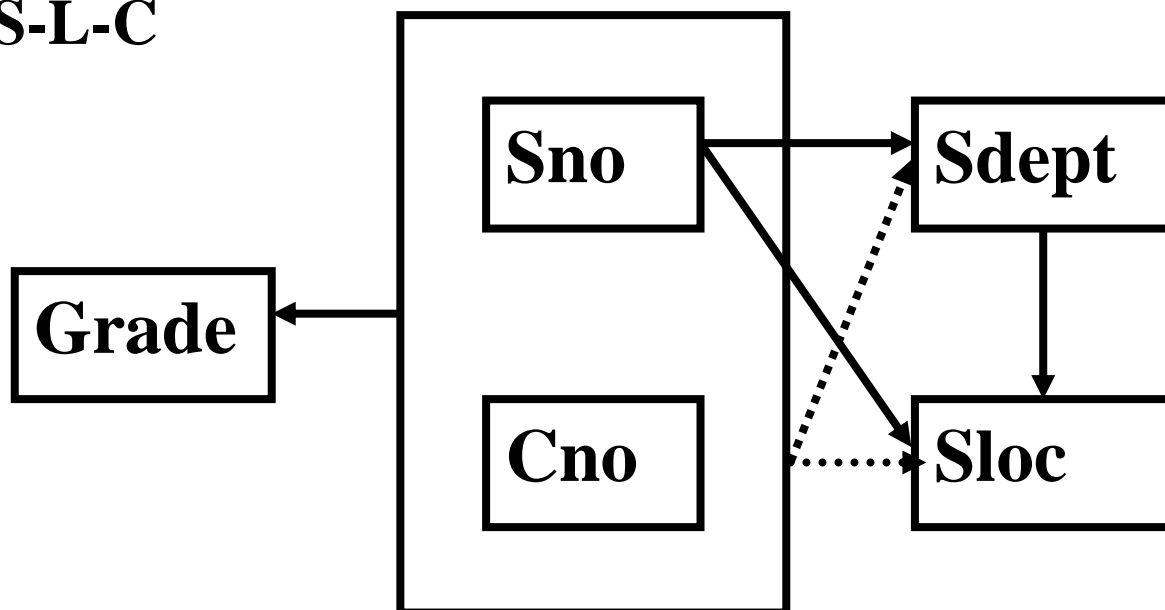
- 解决方法

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)

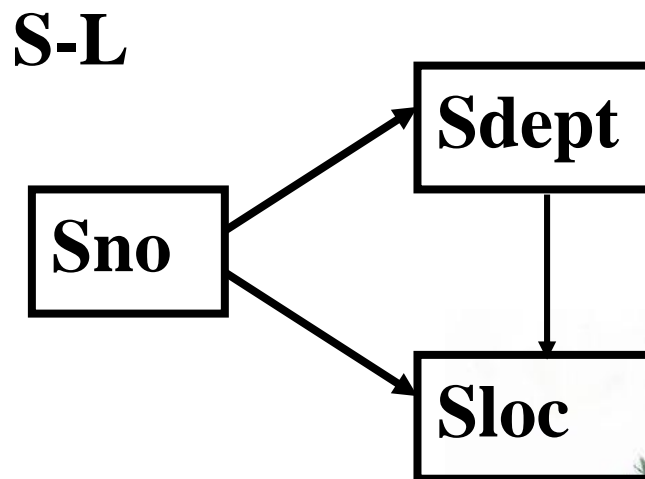
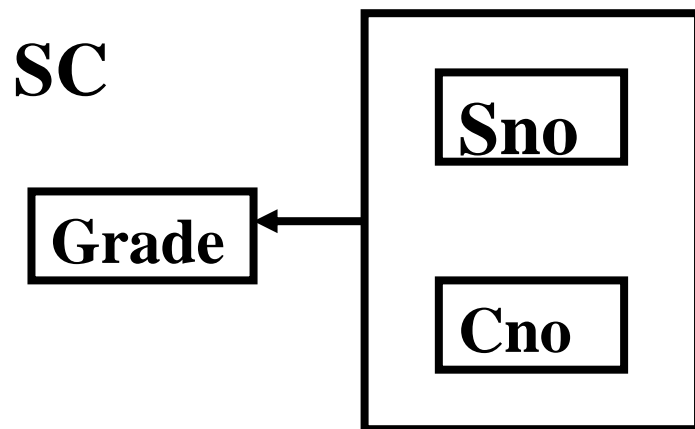
S-L (Sno, Sdept, Sloc)

**S-L-C**



## 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)  $\in 2NF$



# 3NF

S-L (Sno, Sdept, Sloc)

## ❖ 不良特性

- 插入异常：如果系中没有学生，则系的信息就无法插入
- 删除异常：如果学生全部毕业了，则在删除学生信息的同时系的信息也随之删除了



# 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 $R \in 3NF$ , 则每一个非主属性既不部分依赖于码也不传递依赖于码。



## 3NF (续)

例：2NF关系模式S-L(Sno, Sdept, Sloc)中

— 函数依赖：

$Sno \rightarrow Sdept$

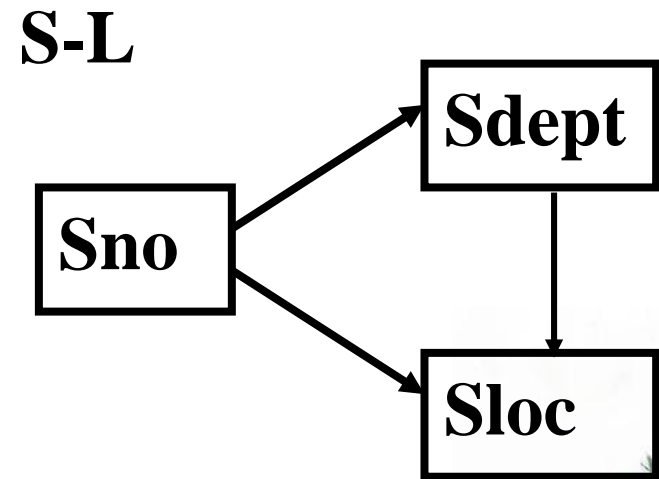
$Sdept \twoheadrightarrow Sno$

$Sdept \rightarrow Sloc$

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)

- 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)

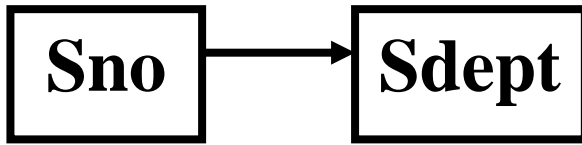
S-D的码为Sno， D-L的码为Sdept。

– 分解后的关系模式S-D与D-L中不再存在传递依赖

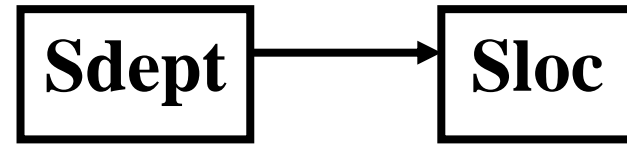


## 3NF (续)

S-D的码为Sno, D-L的码为Sdept



**S-D**



**D-L**

- ❖  $S-L(Sno, Sdept, Sloc) \in 2NF$   
 $S-L(Sno, Sdept, Sloc) \notin 3NF$   
 $S-D(Sno, Sdept) \in 3NF$   
 $D-L(Sdept, Sloc) \in 3NF$



## 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。



# BCNF

- 示例

STJ(S , T , J), 每个老师只教一门课. 每门课有若干老师, 某学生选定某门课, 就对应一个固定的老师. 由语义可得到下面的函数依赖.

$T \rightarrow J$

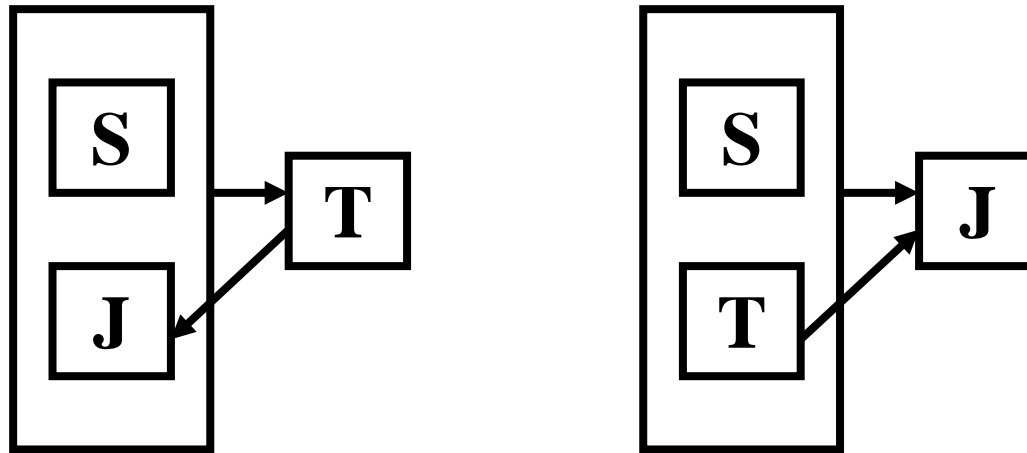
$(S, T) \rightarrow J$

$(S, J) \rightarrow T$

$(S, T)$  ,  $(S, J)$  为候选码。



# BCNF



STJ



# BCNF

## STC(S, T, J)

- 不良特性

- 插入异常：如果没有学生选修某位老师的任课，则该老师担任课程的信息就无法插入
- 删除异常：删除学生选课信息，会删除掉老师的任课信息
- 更新异常：如果老师所教授的课程有所改动，则所有选修该老师课程的学生元组都要做改动
- 数据冗余：每位学生都存储了有关老师所教授的课程的信息

| 学生 | 教师 | 课程   |
|----|----|------|
| 王军 | 李立 | 数据库  |
| 张立 | 吴海 | 网络   |
| 李明 | 王强 | 数据结构 |

主码：（学生，教师）

主码：（学生，课程）



## BCNF (续)

- **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 $R\langle U, F \rangle \in BCNF$ 。
  - BCNF的定义直接建立在1NF的基础上。但实质上BCNF是3NF的改进形式。3NF仅考虑了非主属性对码的依赖情况, BCNF把主属性对码的依赖情况也包括进去。
  - 等价于: 每一个决定属性因素都包含码
- $STJ \notin BCNF$ , 因为 $T \rightarrow J$ , 而 $T$ 不含有码**



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





# BCNF (续)

[例5] 关系模式C (Cno, Cname, Pcnno)

- $C \in 3NF$
- $C \in BCNF$



## BCNF（续）

[例6] 关系模式SJP（S，J，P）学生，课程，名次

每个学生选修每门课的成绩有一个名次，每门课中每一名次只有一个学生

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



# 3NF与BCNF的关系

**3NF**和**BCNF**是在函数依赖的条件下对模式分解所能达到的分离程度的测度。一个模式中的关系模式如果都属于**BCNF**，那么在函数依赖范畴内，它已经实现了彻底的分离，已消除了插入和删除的异常。



# 例：规范化的过程(1)

- 某书店购书情况汇总登记表：

| 订单号<br>NO | 订户代号<br>C# | 姓名<br>CN      | 地址<br>CA | 书号<br>B# | 书名<br>BN  | 出版单位<br>EU | 单价<br>UP | 定购数量<br>QUA |
|-----------|------------|---------------|----------|----------|-----------|------------|----------|-------------|
| 00001     | 0253       | 清华大学<br>自动化系  | 北京       | 02164    | 计算机语言     | 教育出版社      | 2.0      | 50          |
|           |            |               |          | 01003    | BASIC语言   | 科学出版社      | 1.1      | 50          |
|           |            |               |          | 06372    | FORTRAN语言 | 清华出版社      | 1.5      | 30          |
| 0002      | 0372       | 陈刚            | 上海       | 01083    | 电子线路      | 邮电出版社      | 1.4      | 1           |
|           |            |               |          | 02954    | 自动控制原理    | 国防出版社      | 0.95     | 1           |
| 0003      | 2234       | 天津大学<br>计算机系  | 天津       | 02954    | 自动控制原理    | 国防出版社      | 0.95     | 20          |
|           |            |               |          | 02164    | 计算机语言     | 教育出版社      | 2.0      | 50          |
|           |            |               |          | 01083    | 电子线路      | 邮电出版社      | 1.4      | 50          |
|           |            |               |          | 01003    | BASIC语言   | 科学出版社      | 1.1      | 20          |
| 0004      | 2523       | 北京钢铁学院<br>图书馆 | 北京       | 01003    | BASIC语言   | 科学出版社      | 1.1      | 30          |
|           |            |               |          | 06372    | FORTRAN语言 | 清华出版社      | 1.5      | 30          |



## 例：规范化的过程(2)

根据分析可以得到一组函数依赖：

$F = \{ NO \rightarrow C\#, C\# \rightarrow CN, C\# \rightarrow CA, B\# \rightarrow BN, B\# \rightarrow EU, B\# \rightarrow UP, (NO, B\#) \rightarrow QUA \}$ ，表中  $(NO, B\#)$  为主码。

消除重复组后，关系模式满足1NF的要求。

| 订单号<br>NO | 订户代号<br>C# | 姓名<br>CN  | 地址<br>CA | 书号<br>B# | 书名<br>BN  | 出版单位<br>EU | 单价<br>UP | 定购数量<br>QUA |
|-----------|------------|-----------|----------|----------|-----------|------------|----------|-------------|
| 00001     | 0253       | 清华大学自动化系  | 北京       | 02164    | 计算机语言     | 教育出版社      | 2.0      | 50          |
| 00001     | 0253       | 清华大学自动化系  | 北京       | 01003    | BASIC语言   | 科学出版社      | 1.1      | 50          |
| 00001     | 0253       | 清华大学自动化系  | 北京       | 06372    | FORTRAN语言 | 清华出版社      | 1.5      | 30          |
| 0002      | 0372       | 陈刚        | 上海       | 01083    | 电子线路      | 邮电出版社      | 1.4      | 1           |
| 0002      | 0372       | 陈刚        | 上海       | 02954    | 自动控制原理    | 国防出版社      | 0.95     | 1           |
| 0003      | 2234       | 天津大学计算机系  | 天津       | 02954    | 自动控制原理    | 国防出版社      | 0.95     | 20          |
| 0003      | 2234       | 天津大学计算机系  | 天津       | 02164    | 计算机语言     | 教育出版社      | 2.0      | 50          |
| 0003      | 2234       | 天津大学计算机系  | 天津       | 01083    | 电子线路      | 邮电出版社      | 1.4      | 50          |
| 0003      | 2234       | 天津大学计算机系  | 天津       | 01003    | BASIC语言   | 科学出版社      | 1.1      | 20          |
| 0004      | 2523       | 北京钢铁学院图书馆 | 北京       | 01003    | BASIC语言   | 科学出版社      | 1.1      | 30          |
| 0004      | 2523       | 北京钢铁学院图书馆 | 北京       | 06372    | FORTRAN语言 | 清华出版社      | 1.5      | 30          |

## 例：规范化的过程(3)

- 将其分解成三个关系，使每一个非主属性都完全依赖于主码，满足2NF的要求。

| 订单号<br>NO | 订户代号<br>C# | 姓名<br>CN  | 地址<br>CA |
|-----------|------------|-----------|----------|
| 00001     | 0253       | 清华大学自动化系  | 北京       |
| 0002      | 0372       | 陈刚        | 上海       |
| 0003      | 2234       | 天津大学计算机系  | 天津       |
| 0004      | 2523       | 北京钢铁学院图书馆 | 北京       |

| 书号<br>B# | 书名<br>BN   | 出版单位<br>EU | 单价<br>UP |
|----------|------------|------------|----------|
| 02164    | 计算机语言      | 教育出版社      | 2.0      |
| 01003    | BASIC语言    | 科学出版社      | 1.1      |
| 06372    | FORTTRAN语言 | 清华出版社      | 1.5      |
| 01083    | 电子线路       | 邮电出版社      | 1.4      |
| 02954    | 自动控制原理     | 国防出版社      | 0.95     |

| 订单号<br>NO | 书号<br>B# | 定购数量<br>QUA |
|-----------|----------|-------------|
| 00001     | 02164    | 50          |
| 00001     | 01003    | 50          |
| 00001     | 06372    | 30          |
| 0002      | 01083    | 1           |
| 0002      | 02954    | 1           |
| 0003      | 02954    | 20          |
| 0003      | 02164    | 50          |
| 0003      | 01083    | 50          |
| 0003      | 01003    | 20          |
| 0004      | 01003    | 30          |
| 0004      | 06372    | 30          |

## 例：规范化的过程(4)

- 进一步消除传递函数依赖，满足3NF的要求。

| 订户代号<br>C# | 姓名<br>CN | 地址<br>CA |
|------------|----------|----------|
| 0253       | 清华大学自    | 北京       |
| 0372       | 陈刚       | 上海       |
| 2234       | 天津大学计    | 天津       |
| 2523       | 北京钢铁学    | 北京       |

| 订单号<br>NO | 订户代号<br>C# |
|-----------|------------|
| 00001     | 0253       |
| 0002      | 0372       |
| 0003      | 2234       |
| 0004      | 2523       |

| 书号<br>B# | 书名<br>BN  | 出版单位<br>EU | 单价<br>UP |
|----------|-----------|------------|----------|
| 02164    | 计算机语言     | 教育出版社      | 2.0      |
| 01003    | BASIC语言   | 科学出版社      | 1.1      |
| 06372    | FORTRAN语言 | 清华出版社      | 1.5      |
| 01083    | 电子线路      | 邮电出版社      | 1.4      |
| 02954    | 自动控制原理    | 国防出版社      | 0.95     |

| 订单号<br>NO | 书号<br>B# | 定购数量<br>QUA |
|-----------|----------|-------------|
| 00001     | 02164    | 50          |
| 00001     | 01003    | 50          |
| 00001     | 06372    | 30          |
| 0002      | 01083    | 1           |
| 0002      | 02954    | 1           |
| 0003      | 02954    | 20          |
| 0003      | 02164    | 50          |
| 0003      | 01083    | 50          |
| 0003      | 01003    | 20          |
| 0004      | 01003    | 30          |
| 0004      | 06372    | 30          |



指出下列关系模式是第几范式？

➤  $R(X,Y,Z)$   $F=\{XY \rightarrow Z\}$

候选码XY

➤  $R(X,Y,Z)$   $F=\{Y \rightarrow Z, XZ \rightarrow Y\}$

候选码XY与XZ

➤  $R(X,Y,Z)$   $F=\{X \rightarrow Y, X \rightarrow Z\}$

候选码X

➤  $R(W,X,Y,Z)$   $F=\{X \rightarrow Z, WX \rightarrow Y\}$

候选码WX





# 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
  - 实质：概念的单一化



# 规范化小结（续）

- 关系模式规范化的基本步骤

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



# 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

