



2024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 大数据学院

黄振亚, [huangzhy@ustc.edu.cn](mailto:huangzhy@ustc.edu.cn)



# 学生-课程-选课 数据库

2

## □ 学生-课程-选课模式 S-T

学生表: Student (Sno, Sname, Ssex, Sage, Sdept)

课程表: Course (Cno, Cname, Cpno, Ccredit)

学生选课表: SC (Sno, Cno, Grade)

□ 问题: 为什么这样设计? 有没有别的方案?

□ 如何评价一个设计的好坏? 标准是什么?

□ 冗余, 操作异常? 效率?

□ 如何判断?



## 第二篇 设计与应用开发篇

3

- 基于某个数据库管理系统设计数据库，如何基于数据库系统编程
  - 第6章 关系数据理论
  - 第7章 数据库设计
  - 第8章 数据库编程
- 完成课程设计



# 第六章 关系数据理论

4

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

5

## 关系数据库逻辑设计

- 针对具体问题，如何构造一个适合于它的数据库模式
- 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

6

- 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

7

- 关系
- 关系模式
- 关系数据库
- 关系数据库的模式



# 关系 (Relation)

8

$D_1 \times D_2 \times \dots \times D_n$  的子集叫作在域  $D_1, D_2, \dots, D_n$  上的关系, 表示为  $R(D_1, D_2, \dots, D)$

- $R$ : 关系名
- $n$ : 关系的目或度 (Degree)

表 2.2 SAP 关系

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
刘逸	信息专业	王敏





# 关系模式

9

- 关系模式（Relation Schema）是型
- 关系模式是对关系的描述
  - 元组集合的结构
    - 属性构成
    - 属性来自的域
    - 属性与域之间的映象关系
  - 元组语义以及完整性约束条件
  - 属性间的数据依赖关系集合



# 关系模式与关系

10

- 关系模式
  - 对关系的描述
  - 静态的、稳定的
- 关系
  - 关系模式在某一时刻的状态或内容
  - 动态的、随时间不断变化的
- 关系模式和关系往往统称为关系
  - 通过上下文加以区别



## 二、关系模式的形式化定义

11

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



## 三、什么是数据依赖

12

回顾：1. 完整性约束的表现形式

- 限定属性取值范围
  - 例如学生成绩必须在0-100之间
- 定义属性值间的相互关联（主要体现于值的相等与否），

这就是数据依赖，它是数据库模式设计的关键



# 什么是数据依赖（续）

13

## 2. 数据依赖

- 一个关系内部属性与属性之间的约束关系
- 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）

14

## 3. 数据依赖的类型

- 函数依赖（Functional Dependency, 简记为FD）
  - 属性值之间的联系可直接用函数来表示（1:1）
- 多值依赖（Multivalued Dependency, 简记为MVD）
  - 属性值之间存在1:n的关系
- 其他



# 什么是数据依赖（续）

15

函数依赖普遍存在于现实生活中

- 描述一个学生关系，可以有学号、姓名、系名等属性。
  - 一个学号只对应一个学生，一个学生只在一个系中学习
  - “学号”值确定后，学生的姓名及所在系的值就被唯一确定。
- $Sname = f(Sno), Sdept = f(Sno)$ 
  - 即Sno函数决定Sname
  - Sno函数决定Sdept
  - 记作 $Sno \rightarrow Sname, Sno \rightarrow Sdept$



## 四、关系模式的简化表示

16

- 关系模式由五部分组成，即它是一个五元组：

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

- 简化为一个三元组（仅保留属性名、依赖集合）：

$$R(U, F)$$

- 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
- 关系二维表，需符合一个最基本的条件
  - 每个分量必须是不可分开的数据项
  - 满足了这个条件的关系模式就属于第一范式（1NF）





## 五、数据依赖对关系模式的影响

17

[例6.1] 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)

➤ 语义

- 一个系有若干学生， 但一个学生只属于一个系；
- 一个系只有一名（正职）负责人；
- 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。

➤ 假设使用**单一的关系模式**： Student <U、 F>

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

思考 教务处数据库的语义？



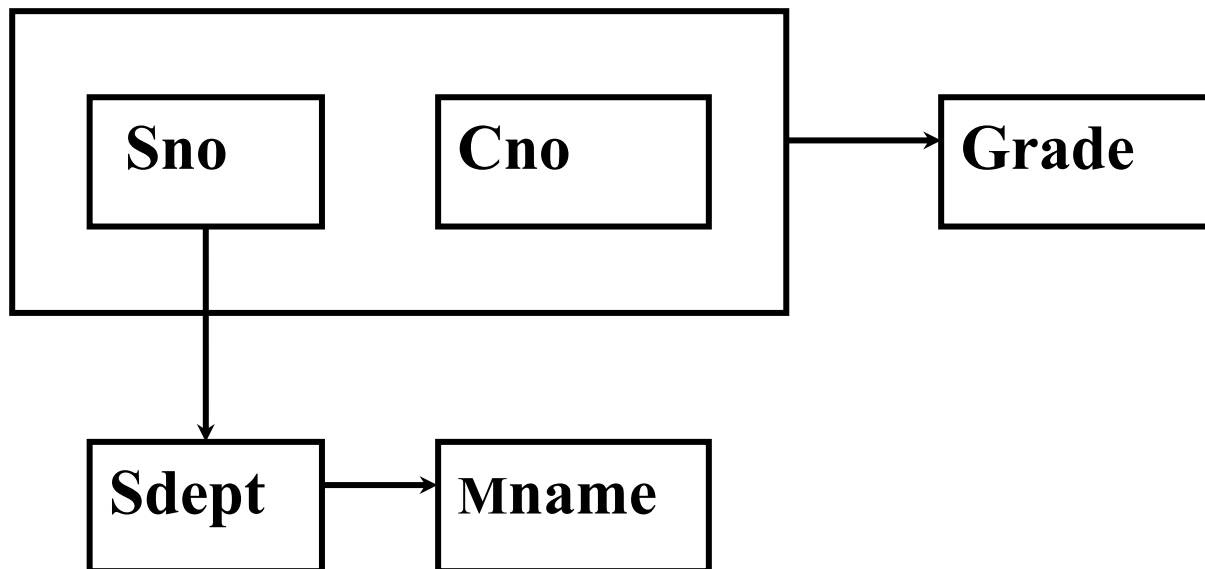
## 数据依赖对关系模式的影响（续）

18

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

- 现实世界的语义
- 属性组U上的一组函数依赖F:

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





# 数据依赖对关系模式的影响（续）

19

表6.1 Student表

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
....	...	...	...	...

1. 数据冗余：浪费存储空间，维护数据完整性代价大
2. 更新异常：更新系主任时
3. 插入异常：插入新的系
4. 删除异常：删除某系的学生



# 数据依赖对关系模式的影响（续）

20

结论：

- 单一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

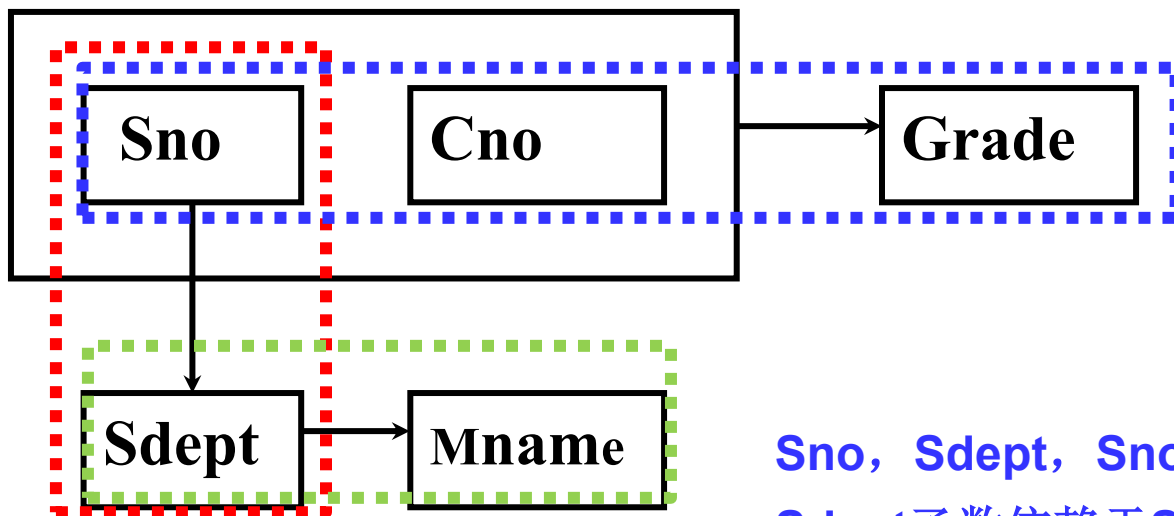
21

- 把这个单一模式分成3个关系模式:

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

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

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



Sno, Sdept, Sno函数确定 Sdept  
Sdept函数依赖于Sno, Sdept, Sno



# 第六章 关系数据理论

22

6.1 问题的提出

6.2 规范化（重点）

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

23

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

24

### 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

25

- 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

26

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 $X \rightarrow Y$ 。

➤  $X$ 与 $Y$ 之间是1:1的关系



## 函数依赖

27

- 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 $X \nrightarrow Y$ 。



## 函数依赖（续）

28

❖ [例] Student (Sno, Sname, Ssex, Sage, Sdept),  
假设不允许重名，则有：

$Sno \rightarrow Ssex, \quad Sno \rightarrow Sage$

$Sno \rightarrow Sdept, \quad Sno \longleftrightarrow Sname$

$Sname \rightarrow Ssex, \quad Sname \rightarrow Sage$

$Sname \rightarrow Sdept$

但  $Ssex \not\rightarrow Sage, Ssex \not\rightarrow Sdept$

若  $X \rightarrow Y$ ，并且  $Y \rightarrow X$ ，则记为  $X \longleftrightarrow Y$ 。

若  $Y$  不函数依赖于  $X$ ，则记为  $X \not\rightarrow Y$ 。



## 函数依赖（续）

29

❖ 由下面的关系表，能否得出  $Sno \rightarrow Sname$

Sno	Sname	Ssex	Sage	Sdept
S1	张三	男	20	计算机系
S1	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

违背了  $Sno \rightarrow Sname$



## 函数依赖（续）

30

❖ 由下面的关系表，能否得出  $Sno \rightarrow Sname$

Sno	Sname	Ssex	Sage	Sdept
S1	张三	男	20	计算机系
S2	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

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。



# 函数依赖（补充说明）

31

## 1. R的所有关系实例均要满足

- 函数依赖不是指关系模式R的某个或某些关系实例r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. 语义范畴的概念

- 需要根据语义来确定函数依赖。
  - 如，姓名 $\rightarrow$ 年龄只有在该部门没有同名人的条件下成立，如果允许有同名，则年龄不再依赖于姓名

## 3. 数据库设计者可以对现实世界作强制的规定

例如，规定不允许有同名，使得函数依赖“姓名 $\rightarrow$ 年龄”成立



## 二、平凡函数依赖与非平凡函数依赖

32

定义6.1补充:

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

如果 $X \rightarrow Y$ , 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:  $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$

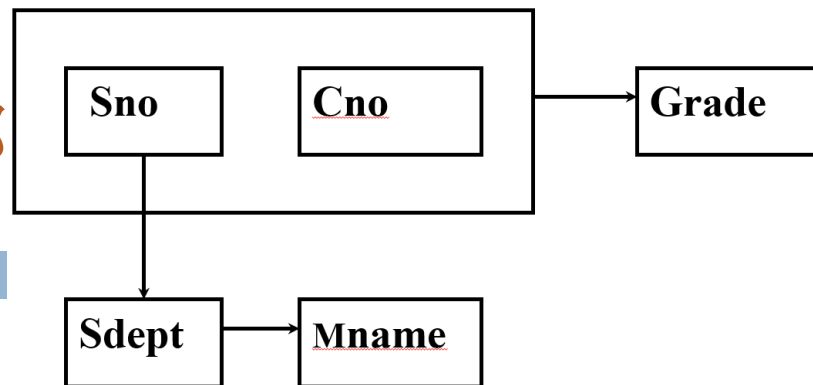
对于任一关系模式, 平凡函数依赖都是必然成立的, 它不反映新的语义。

若不特别声明, 我们总是讨论非平凡函数依赖。





### 三、完全函数依赖与部



33

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

$(Sno, Cno) \xrightarrow{F} Grade$ 是完全函数依赖,

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

?  $\xrightarrow{F} U$

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



### 三、完全函数依赖与部分函数依赖

34

❖ [例] 在关系SC(Sno, Cno, Grade)中, 有:

■ 由于:  $Sno \twoheadrightarrow Grade$ ,  $Cno \twoheadrightarrow Grade$ ,

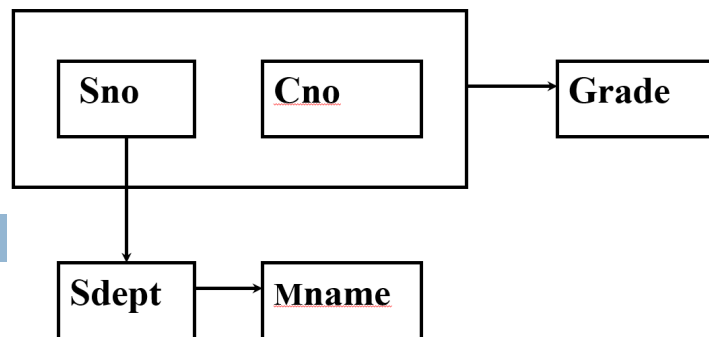
因此:  $(Sno, Cno) \xrightarrow{F} Grade$

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

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



## 四、传递函数依赖



35

定义6.3 在 $R(U)$ 中,

➤ 如果 $X \rightarrow Y$ ,  $Y \not\subseteq X$ ,  $Y \not\rightarrow X$ ,  $Y \rightarrow Z$ ,  $Z \not\subseteq Y$

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 $X \leftrightarrow Y$ , 则 $Z$ 直接依赖于 $X$ , 而不是传递函数依赖。

例6.1:  $U = \{ \text{Sno}, \text{Sdept}, \text{Mname}, \text{Cno}, \text{Grade} \}$ , 有:

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

$\text{Sno} \xrightarrow{\text{传递}} \text{Mname}$



## 6.2 规范化

36

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

38

从函数依赖的角度：

定义6.4 设 $K$ 为 $R\langle U, F \rangle$ 中的属性或属性组合。若 $U$ 完全函数依赖于 $K$ ，即 $K \xrightarrow{F} U$ ，则 $K$ 称为 $R$ 的候选码(Candidate Key)

- 若候选码多于一个，则选定其中一个做为主码(Primary Key)
- 注意：
  - 如果 $U$ 部分函数依赖于 $K$ ，即 $K \xrightarrow{P} U$ ，则 $K$ 称为超码(Surpkey)
  - 候选码是最小的超码
    - 候选码的超集一定是超码
    - 候选码的任意一个真子集都不是候选码



## 码（续）

39

### □ 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)或非码属性（Non-key attribute）

### □ 全码

- 整个属性组是码，称为全码（All-key）

[例1] S { Sno, Sdept, Mname, Cno, Grade }

[例2] S { Sno, Sname, Ssex, Sage, Sdept}, 假设不重名

候选码？ 主属性？ 非主属性？



## 码（续）

40

[例6.2]

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6.3]

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

41

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，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

42

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

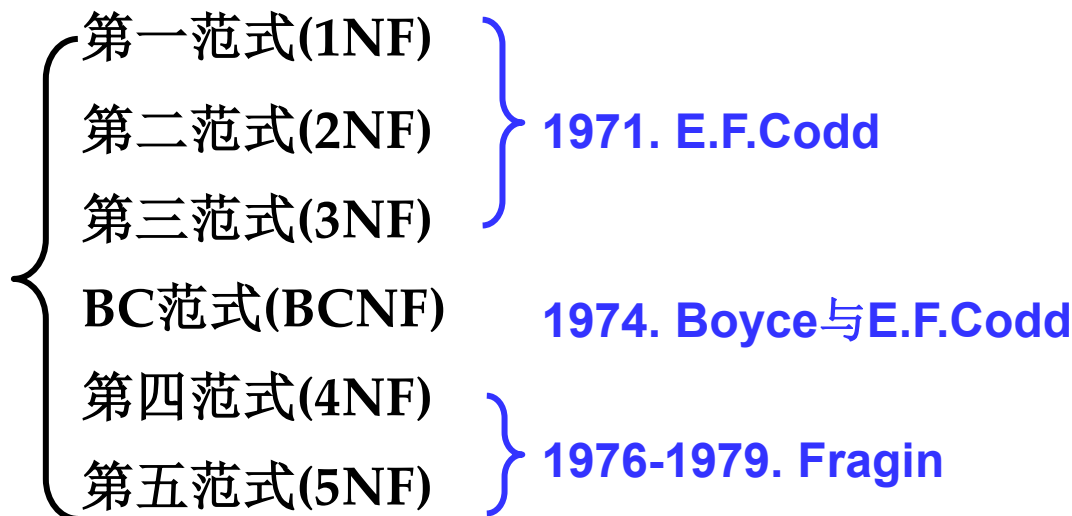
6.2.9 规范化小结



## 6.2.3 范式

43

- 范式是符合某一种级别的关系模式的集合
- 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
- 范式的种类：





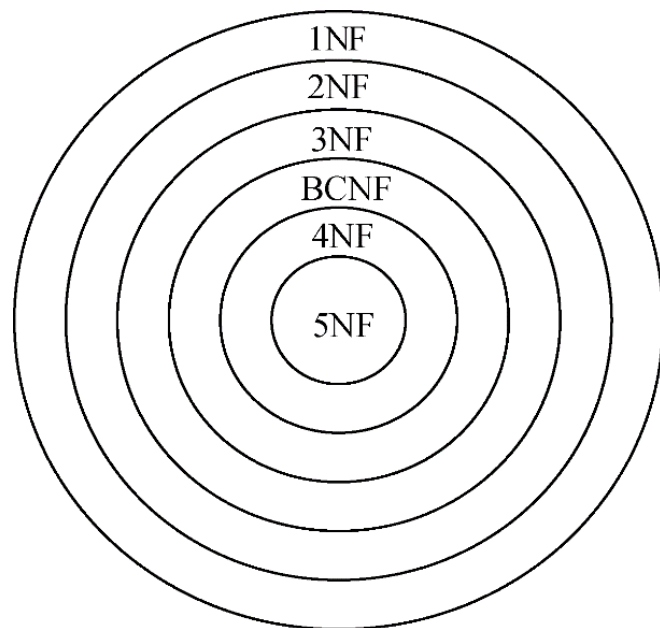
## 6.2.3 范式

44

- 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**





## 6.2 规范化

45

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

46

### □ 1NF的定义

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

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



## 6.2.4 2NF

47

### □ 1NF的例子

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

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
....	...	...	...	...



## 2NF—1NF的问题

48

[例6.4] 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：

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

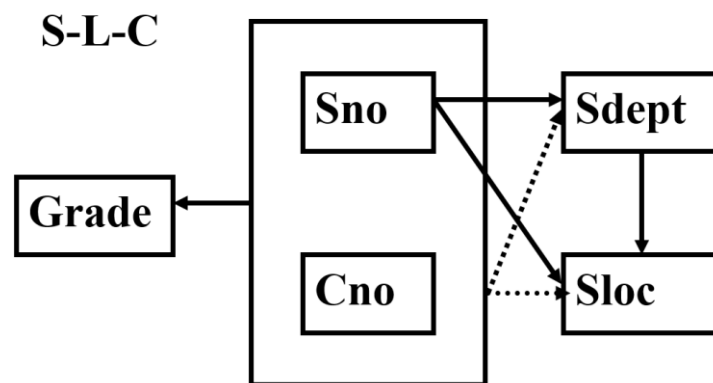
$Sno \rightarrow Sdept$

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

$Sno \rightarrow Sloc$

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

$Sdept \rightarrow Sloc$



- S-L-C的码为(Sno, Cno)
- S-L-C满足第一范式1NF。
- 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）

49

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

## (1) 插入异常

- 插入新学生，未选课（无Cno）的学生无法插入

## (2) 删除异常

- 删除只选了一门课的学生的选课信息时，必须把整个元组一起删除，造成学生的Sdept Sloc等信息丢失。

## (3) 数据冗余度大

- Sdept, Sloc重复存储

Sno	Sdept	Sloc	Cno	Grade
....	...	...	...	...

## (4) 修改复杂

- 修改学生系别Sdept时需要同时修改Sloc。如果一个学生选  
了多门课，则Sdept, Sloc被存储了多次。如果该生转系，  
则需要修改所有相关的Sdept和Sloc，造成修改的复杂化





## S-L-C不是一个好的关系模式（续）

50

### □ 原因

- 存在一类：如Grade，完全函数依赖于码（Sno，Cno）
- 存在另一类：如Sdept、Sloc，部分函数依赖于码（Sno）

### □ 解决方法：投影分解法

S-L-C分解为两个关系模式，以消除这些部分函数依赖

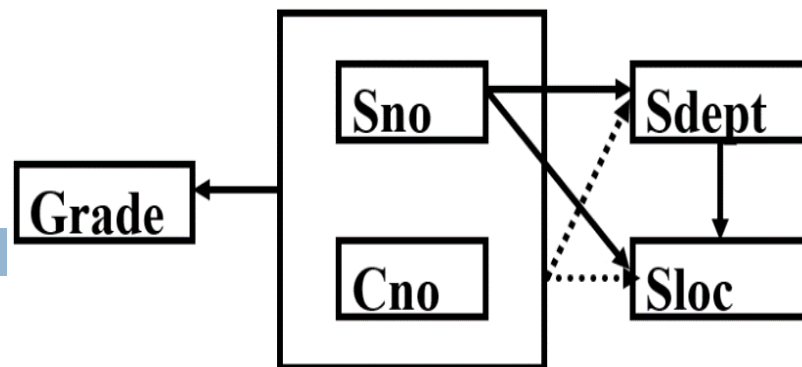
SC (Sno, Cno, Grade)

S-L (Sno, Sdept, Sloc)

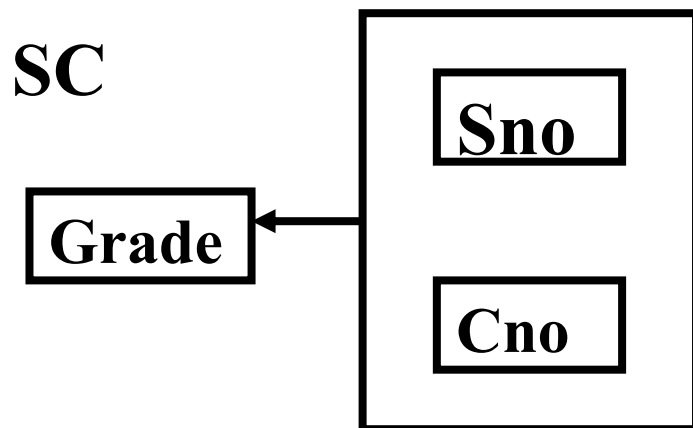


## 2NF (续)

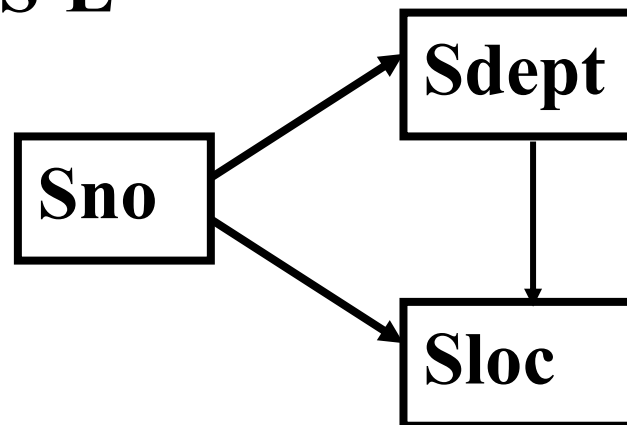
51



函数依赖图:



**S-L**



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



## 2NF (续)

52

### □ 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 $R \in 2NF$ 。

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

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

S-C (Sno, Cno, Grade)  $\in 2NF$

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



## 2NF (续)

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

53

### □ 2NF vs. 1NF (问题解决)

#### □ 插入异常?

- 插入新学生, 未选课 (无Cno), 可以插入SL关系中

#### □ 删除异常?

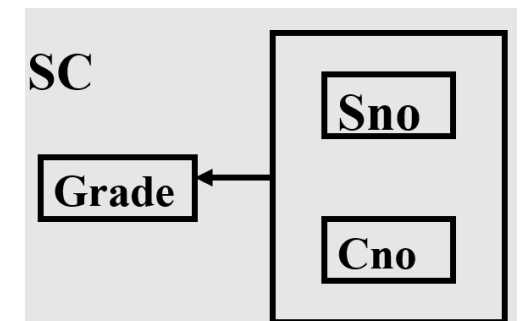
- 删除一个学生所有选课记录, 只是SC关系没有记录。SL关系关于该学生不受影响

#### □ 数据冗余度?

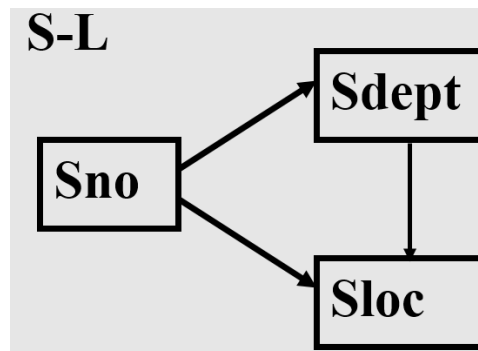
- 不论一个学生选多少课, Sdept和Sloc值都只存储1次 (SL关系中)

#### □ 修改复杂?

- 学生转系: 只修改SL关系即可。



sno	Cno	Grade
...	...	...



Sno	Sdept	Sloc
....	...	...



## 2NF (续)

54

- 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。
- 2NF消除了非主属性对码的部分函数依赖
  - 判断2NF：1. 函数依赖 2. 码，非主属性 3. 部分函数依赖
- 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

55

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

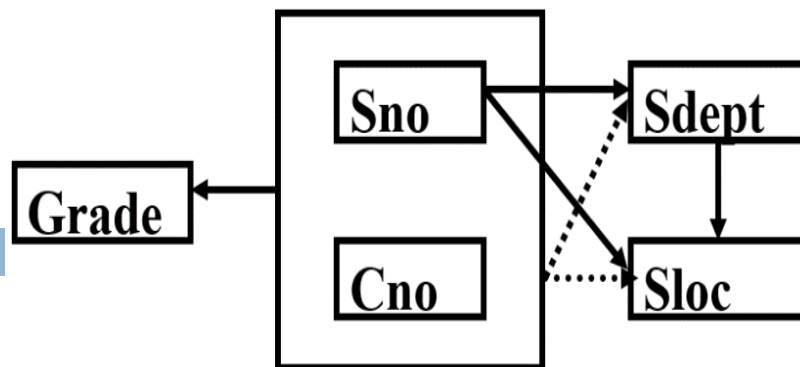
6.2.8 4NF

6.2.9 规范化小结

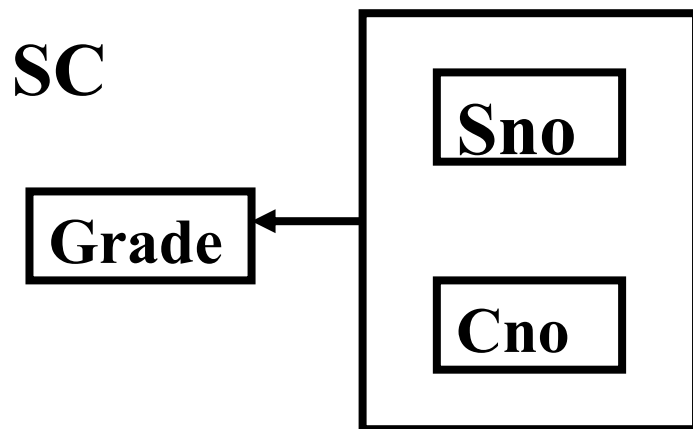


# 再想想2NF

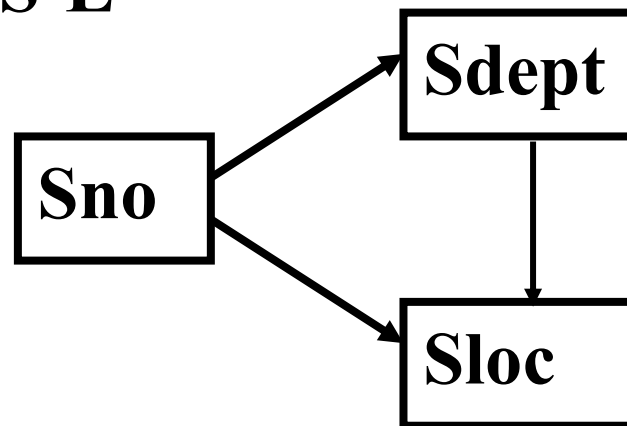
56



函数依赖图：



S-L



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



## 再想想2NF

定义6.3 在 $R(U)$ 中,

➤ 如果 $X \rightarrow Y$ ,  $Y \not\subseteq X$ ,  $Y \not\rightarrow X$ ,  $Y \rightarrow Z$ ,  $Z \not\subseteq Y$

则称 $Z$ 对 $X$ 传递函数依赖。记为:  $X \xrightarrow{\text{传递}} Z$

57

例: 2NF关系模式S-L(Sno, Sdept, Sloc)中

□ 函数依赖:

$Sno \rightarrow Sdept$

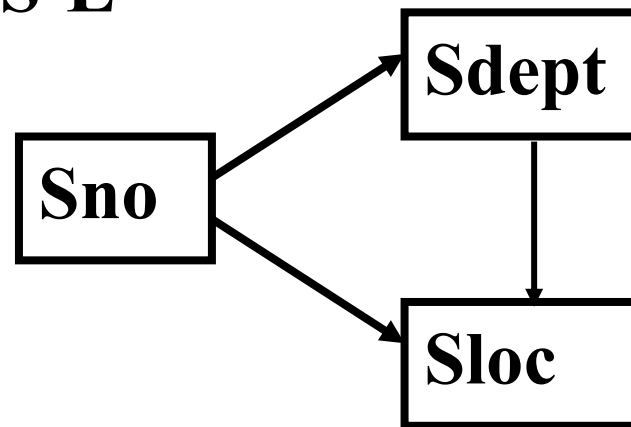
$Sdept \rightarrow Sloc$

$Sno \rightarrow Sloc$

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



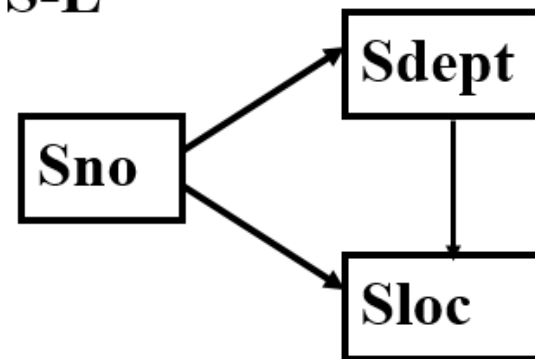




## 再想想2NF

Sno	Sdept	Sloc
....	...	...

S-L



58

### □ 在S-L关系中S-L(Sno, Sdept, Sloc)

#### □ 插入异常

- 如果某个系(DS)刚成立，没有学生，则：无法插入

#### □ 删除异常

- 如果某个系(DS)的学生全部毕业，则删除所有学生，则：删除了系

#### □ 数据冗余

- 每个系的学生都住在同一个地方，该系的sloc信息重复出现

#### □ 修改复杂

- 如果学校调整学生住处(DS)时，则需要同时修改所有学生的loc

### □ SL不是一个理想的关系模式



## 6.2.5 3NF

59

### □ 3NF的定义

定义6.7 关系模式 $R \langle U, F \rangle$ 中, 若不存在这样的情况: 码 $X$ 、属性组 $Y \subseteq U$ 及非主属性 $Z$ , 且 $Z \not\subseteq Y, Y \twoheadrightarrow X$ , 使得 $X \rightarrow Y, Y \rightarrow Z$ 成立, 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?
- 若 $R \in 3NF$ , 则必有 $R \in 2NF$ , 证明?



## 3NF (续)

60

例：2NF关系模式S-L(Sno, Sdept, Sloc)中

□ 函数依赖：

$Sno \rightarrow Sdept, Sno \rightarrow Sloc$

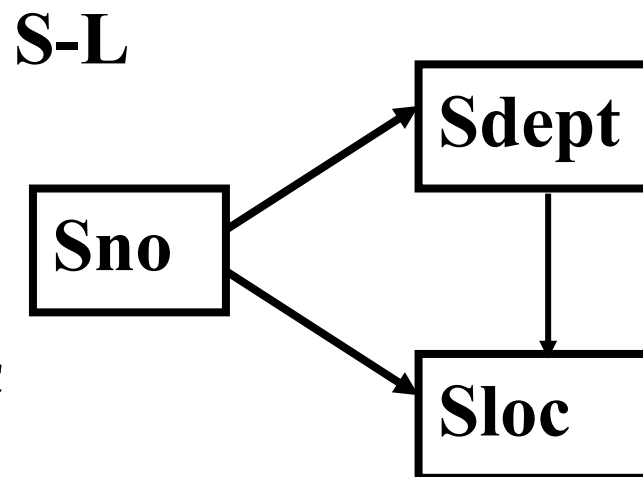
$Sdept \not\rightarrow Sno, Sdept \rightarrow Sloc$

码：Sno，非主属性：Sdept, 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，

因此， $S-L \notin 3NF$

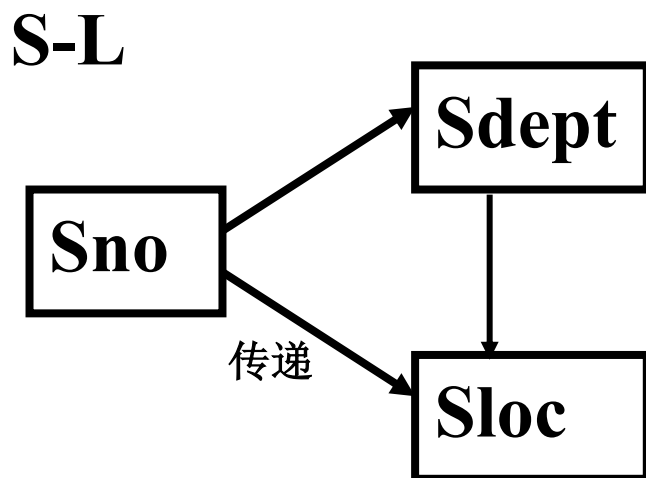




## 3NF (续)

61

函数依赖图:



- SL不是一个好的关系模式



## 3NF (续)

62

### □ 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。

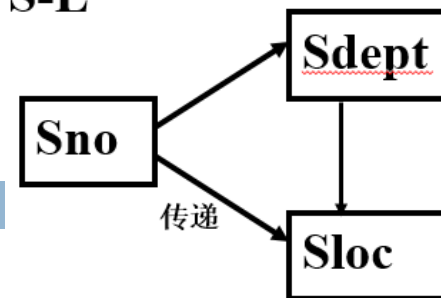
■ 分解后的关系模式S-D与D-L中不再存在传递依赖



# 3NF (续)

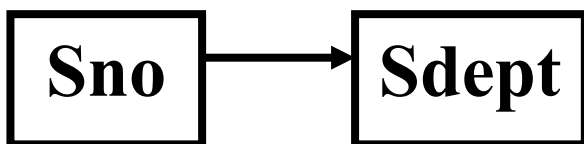
63

S-L



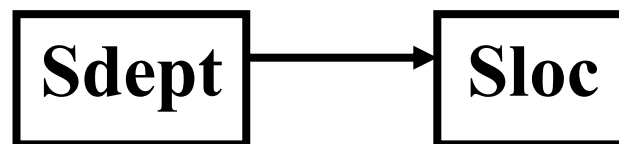
S-D的码为Sno, D-L的码为Sdept

S-D



Sno	Sdept
....	...

D-L



Sdept	Sloc
...	...

- ❖ 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

解决插入异常: **DS**刚成立—**DL**

解决删除异常: **DS**的学生全部毕业—**SD**

解决数据冗余: **DS**学生住在同一**sloc**—**DL**

解决修改复杂: 学校调整**DS**学生**sloc**—**DL**



## 3NF (续)

64

- 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。
- 3NF消除了非主属性对码的传递函数依赖
  - 判断3NF: 1. 函数依赖 2. 码, 非主属性 3. 传递函数依赖
- 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。



# 关系模式规范化的基本步骤

65

## □ 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  
凡函数依赖

↓ 消除非主属性对码的传递函数依赖

**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**





## 6.2 规范化

66

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



# 再想想3NF

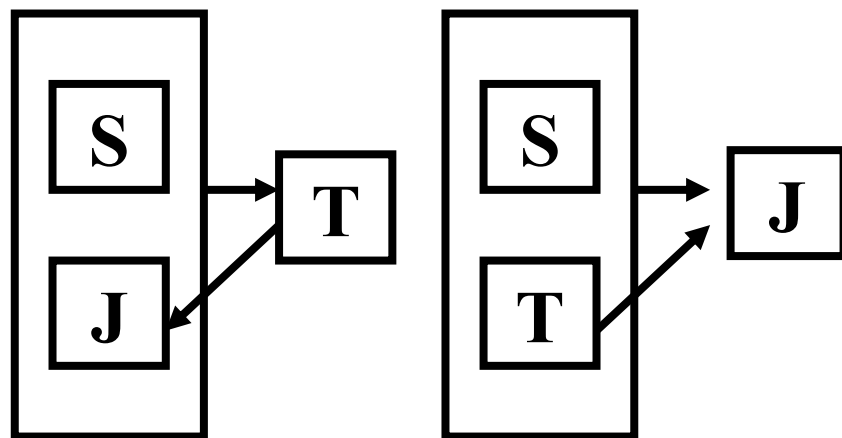
67

[例6.8]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。有如下函数依赖

- 语义: 每个教师只教一门课, 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。某个学生选修某个教师的课就确定了所选课的名称

- 函数依赖形式化:

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





## 再想想3NF

68

□ 函数依赖:

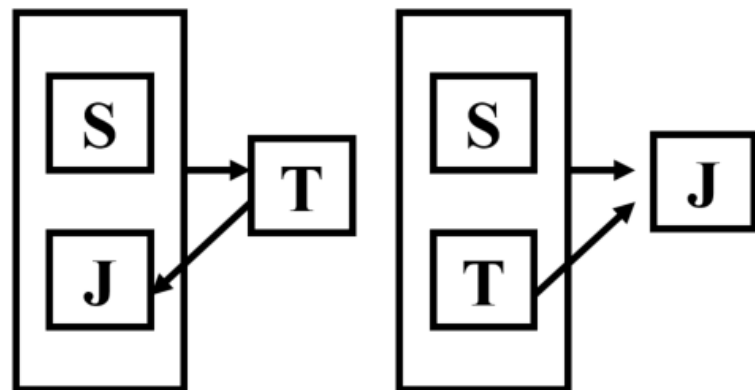
$T \rightarrow J$ ,  $(S, J) \rightarrow T$ ,  $(S, T) \rightarrow J$

□ 码:  $(S, J)$  和  $(S, T)$

□ 主属性:  $S, T, J$  都是主属性

□  $T \rightarrow J$ , 即  $T$  是决定因素, 可是  $T$  只是主属性, 它既不是候选码, 也不包含候选码

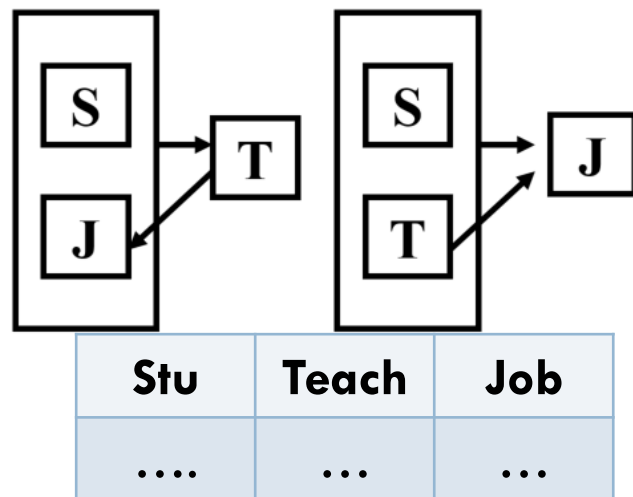
□  $STJ \in 3NF$ , 不存在非主属性对码的部分和传递函数依赖





# 再想想3NF

69



□ STJ (S, T, J) 是否有问题？

□ 插入异常

- 如果某个教师开设某门课，但尚未有学生选修，则信息无法插入

□ 删除异常

- 如果选修过某门课程的学生全部毕业，删除这些学生元组同时，相应教师开设的该课程也删除了

□ 数据冗余

- 虽然一个教师只教一门课，但是每个选修该教师该门课程的学生元组都要记录这一信息

□ 修改复杂

- 某个教师开设课程改名，所有选修该教师该门课学生元组都要进行修改

课后思考：给出数据，验证STJ的异常情况

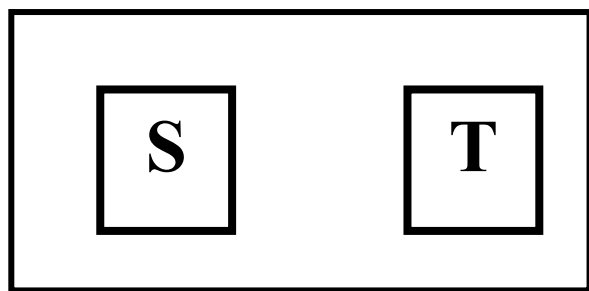
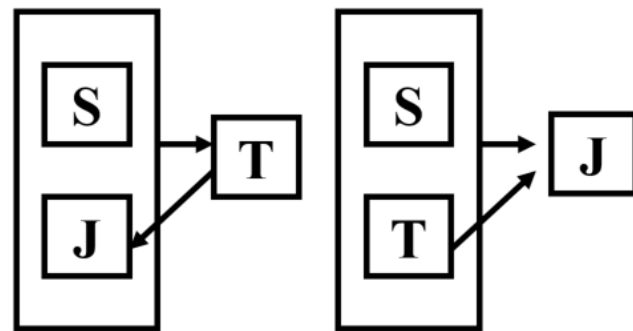
□  $STJ \in 3NF$ ，但不是一个理想的关系模式



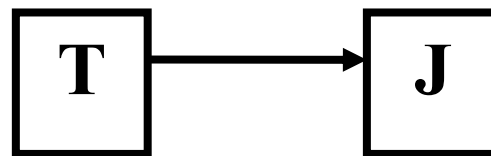
## 再想想3NF

70

- STJ (S, T, J) 不是理想的模式
  - 原因：主属性J依赖于T，即主属性J部分依赖于 (S, T)
  - 解决方法：
    - 采用投影分解法，将STJ分解为关系模式：



ST

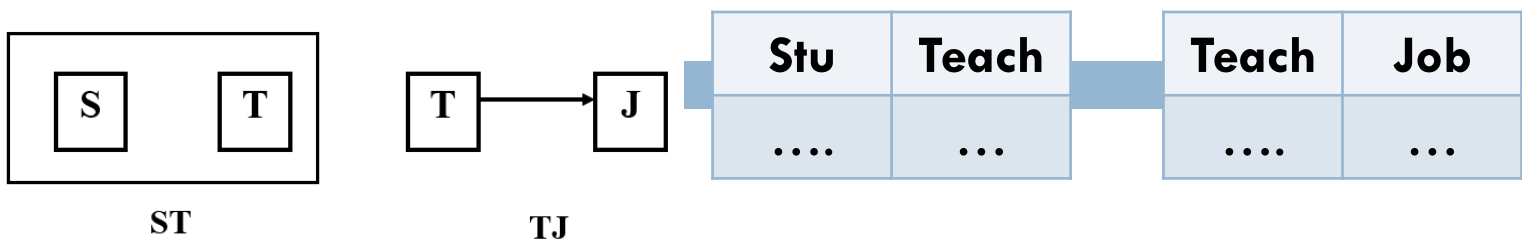


TJ



# 再想想3NF

S表示学生，T表示教师，J表示课程。



- 分解后的R没有**任何属性对码**的部分函数依赖和传递函数依赖
  - 解决插入异常
    - TJ关系中可以存储教师开课信息，但无学生选修
  - 解决删除异常
    - 若选修某课程的学生全部毕业，只删除了ST关系的元组，不会影响TJ关系中教师开设课程的信息
  - 解决数据冗余
    - 每个教师开设课程的信息只在TJ关系中存储一次
  - 解决更新异常
    - 某个教师开设的某课程改名，只需要修改TJ关系中的一个元组



## 6.2.6 BC范式 (BCNF)

72

- 定义6.8 关系模式 $R\langle U, F \rangle \in 1NF$ , 若 $X \rightarrow Y \in F$ , 且 $Y \not\subseteq X$ 时, **X必含有码**, 则 $R\langle U, F \rangle \in BCNF$ .
  - 1974年, 由Boyce与Codd提出, 比3NF更进了一步。通常认为BCNF是修正的第三范式, 有时也称为扩充的第三范式
- 换言之, 在关系模式 $R\langle U, F \rangle$ 中, 如果每一个**决定因素都包含码 (候选码)**, 则 $R \in BCNF$



## BCNF (续)

73

[例6.5] 关系模式C (Cno, Cname, Pcno)

- $C \in 3NF$ : 只有一个码Cno, 没有任何属性对Cno部分依赖或传递依赖
- $C \in BCNF$ : 同时C中Cno是唯一的决定因素, 所以 $C \in BCNF$

[例6.6] 关系模式S (Sno, Sname, Sdept, Sage)

- 假定S有两个码Sno, Sname, 即Sname也是唯一的
- $S \in 3NF$
- $S \in BCNF$





## BCNF (续)

74

[例6.7] 关系模式SJP (S, J, P), S是学生, J是课程, P表示名次。  
每个学生选修每门课的成绩有一定的名次, 每门课中每一名只有一个学生 (无并列)。

- 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
- $SJP \in 3NF$ ,
- $SJP \in BCNF$



# BCNF (续)

75

[例6.8]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。每个教师只教一门课, 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。

□ 函数依赖:

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

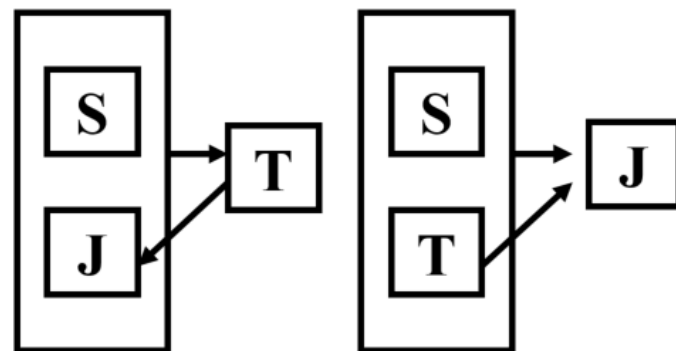
□ (S, J)和(S, T)都是候选码

□  $STJ \in 3NF$

- 没有任何非主属性对码传递依赖或部分依赖
- S,T,J都是主属性

□  $STJ \notin BCNF$  (BCNF每一个决定属性因素都包含码)

- T是决定因素, T不包含码



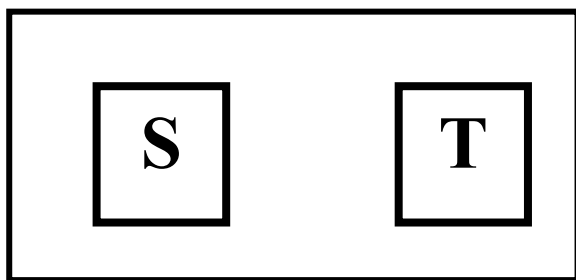


## BCNF (续)

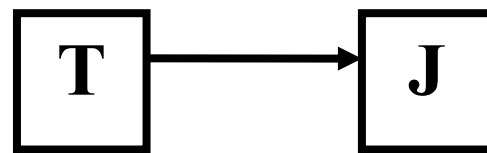
76

- 解决方法：将STJ分解为二个关系模式：

$SJ(S, J) \in \text{BCNF}, TJ(T, J) \in \text{BCNF}$



ST



TJ

没有任何属性（包括主属性）对码的部分函数依赖和传递函数依赖



## BCNF (续)

77

- 若  $R \in \text{BCNF}$ , 则含有性质
  - 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
- 如果关系模式  $R \in \text{BCNF}$ , 则  $R \in 3\text{NF}$
- 如果  $R \in 3\text{NF}$ , 且  $R$  中只有一个候选码, 则  $R$  必为  $\text{BCNF}$ 
  - Recall: 若  $R \in 3\text{NF}$ , 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与BCNF的关系

78

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

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

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

□ 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

□ 如果一个关系数据库中的所有关系模式都属于BCNF，那么在函数依赖范畴内，它已实现了模式的彻底分解，达到了最高的规范化程度，消除了插入异常和删除异常

□ 3NF的“不彻底”性表现在可能存在主属性对码的部分依赖或传递依赖。



# 关系模式规范化的基本步骤

79

## □ 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**

范式越高越好？



## 6.2 规范化

92

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

93

[例6.9] 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。每个教师可以讲授多门课程，每种参考书可以供多门课程使用。

□ 关系模式Teaching(C,T,B)

- 课程C
- 教师T
- 参考书B





# 6.2.7 多值依赖

[例6.9] 学校中某一门课程C由多个教师T讲授，他们使用相同的一套参考书B。每个教师可以讲授多门课程，每种参考书可以供多门课程使用。

❖ 表6.3. 非规范化关系

课程 C	教师 T	参考书 B
物理	{李 勇 王 军}	{普通物理学 光学原理 物理习题集}
数学	{李 勇 张 平}	{数学分析 微分方程 高等代数}
计算数学	{张 平 周 峰}	{数学分析 ...}
...	...	...



# 多值依赖（续）

❖ 表6.4 用二维表表示Teaching

课程C	教师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）

96

[例6.9] 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。每个教师可以讲授多门课程，每种参考书可以供多门课程使用。

- Teaching具有**唯一候选码**(C, T, B)，即全码all-key
- $\text{Teaching} \in \text{BCNF}$



# 多值依赖（续）

存在  
多值依赖

Teaching模式中存在的问题

- (1) 数据冗余度大
  - 存储教师与参考书？
- (2) 插入操作复杂
  - 课程增加一个教师，参考书？
- (3) 删除操作复杂
  - 课程删除一本参考书，教师？
- (4) 修改操作复杂
  - 课程修改一本参考书，教师？

原因：参考书B和教师T之间的关系是笛卡尔积，则彼此独立，毫无关系，都只取决于课程C

课程C	教师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）

98

□ 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

□ 例 Teaching ( $C, T, B$ )

□ 对于 $C$ 的每一个值， $T$ 有一组值与之对应，而不论 $B$ 取何值。因此， $T$ 多值依赖于 $C$ ，即 $C \twoheadrightarrow T$

■ 分析：对于一个 $(x_{\text{物理}}, z_{\text{光学原理}})$ 有一组 $T \{ \text{李勇, 王军} \}$ ，这组 $T$ 仅取决于 $C$ 的值。即，对于另一个 $(\text{物理}, \text{普通物理学})$ ，它对应的一组 $T$ 仍然是 $\{ \text{李勇, 王军} \}$ ，(尽管此时参考书 $B$ 已经改变)。所以， $T$ 多值依赖于 $C$ ，即 $C \twoheadrightarrow T$

■ 同理：对于一个 $(x_{\text{物理}}, z_{\text{李勇}})$ 也是如此，即 $C \twoheadrightarrow B$ 也成立



## 多值依赖（续）

99

□ 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(1)  $w[X]=v[X]=t[X]=s[X]$ ，而(2)  $w[Y]=t[Y]$ ， $w[Z]=s[Z]$ ，(3)  $v[Y]=s[Y]$ ， $v[Z]=t[Z]$ 均成立（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）

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

(1)  $w[X]=v[X]=t[X]=s[X]$ ,

(2)  $w[Y]=t[Y], w[Z]=s[Z]$ ,

(3)  $v[Y]=s[Y], v[Z]=t[Z]$

(即交换 $s, t$ 元组的 $Y$ 值所得的两个新元组必在 $r$ 中)均成立。

则 $Y$ 多值依赖于 $X$ , 记为 $X \twoheadrightarrow Y$ 。

➤ 这里,  $X, Y$ 是 $U$ 的子集,  
 $Z=U-X-Y$ 。

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

101

- 平凡多值依赖和非平凡的多值依赖
  - 若 $X \twoheadrightarrow Y$ ，而 $Z = \emptyset$ （空），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）

102

## [例6.10] 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
- 每种商品被所有保管员保管

W	S	C
W1	S1	C1
W1	S1	C2
W1	S1	C3
W1	S2	C1
W1	S2	C2
W1	S2	C3
W2	S3	C4
W2	S3	C5
W2	S4	C4
W2	S4	C5

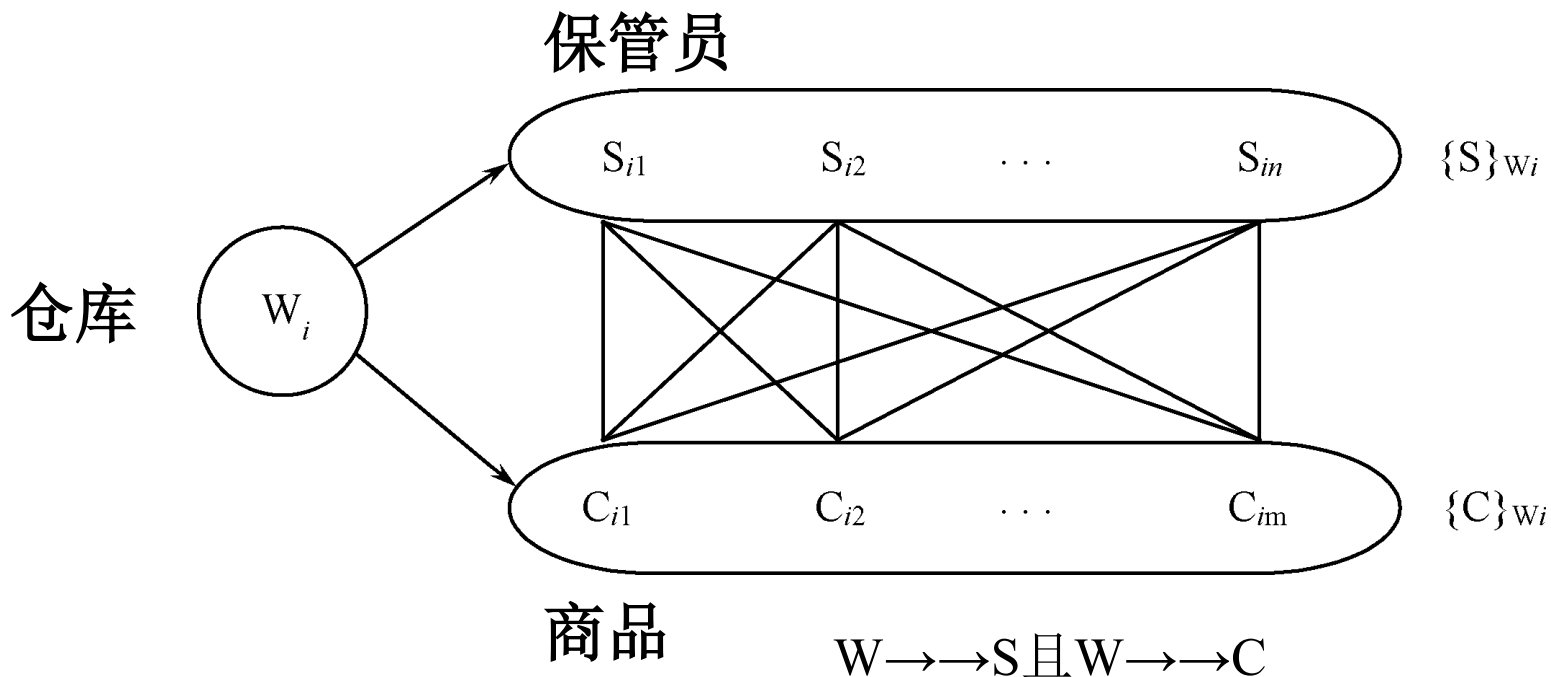


# 多值依赖（续）

103

## [例6.10] 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
- 每种商品被所有保管员保管





# 多值依赖的性质

104

## (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例

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

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

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

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

课后思考：举出实际的例子验证或证明上述性质

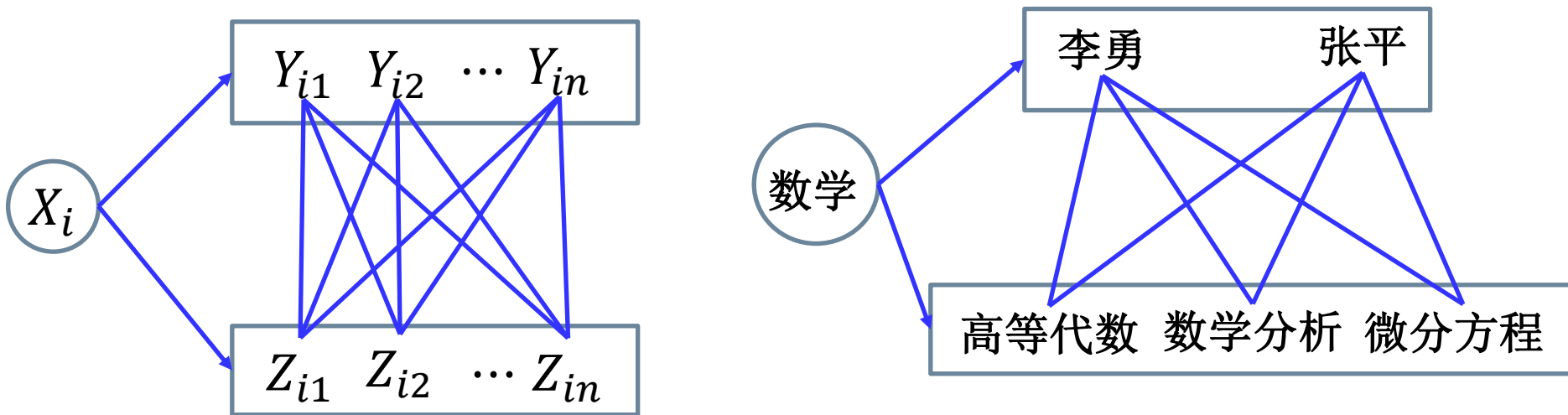


# 多值依赖的性质

105

(1) 多值依赖具有对称性

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





# 多值依赖与函数依赖的区别

106

## (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 $W$  ( $XY \subseteq W \subseteq U$ ) 上一定成立。反之则不然, 即 $X \twoheadrightarrow Y$ 在 $W$  ( $W \subset U$ ) 上成立, 在 $U$ 上并不一定成立
  - 原因: 多值依赖不仅涉及属性组 $X$ 和 $Y$ , 而且涉及 $U$ 中其余属性 $Z$
- 一般地, 在 $R(U)$ 上若有 $X \twoheadrightarrow Y$ 在 $W$  ( $W \subset U$ ) 上成立, 则称 $X \twoheadrightarrow Y$ 为 $R(U)$ 的嵌入型多值依赖
- 函数依赖 $X \rightarrow Y$ 的有效性仅决定于 $X$ 、 $Y$ 这两个属性集的值
- 只要在 $R(U)$ 的任何一个关系 $r$ 中, 元组在 $X$ 和 $Y$ 上的值满足定义6.1, 则函数依赖 $X \rightarrow Y$ 在任何属性集 $W$  ( $XY \subseteq W \subseteq U$ ) 上成立

定义6.1 函数依赖



# 多值依赖与函数依赖的区别

107

(2) 若函数依赖 $X \rightarrow Y$ 在 $R(U)$ 上成立, 则任何 $Y' \subset Y$ 均有 $X \rightarrow Y'$  成立

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

➤ 例如, 关系 $R(A,B,C,D)$ ,  $A \twoheadrightarrow BC$ 成立, (也有 $A \twoheadrightarrow D$ 成立)。  $A \twoheadrightarrow B$ 不一定成立, 因为 $B$ 的取值不仅取决于属性 $A$ , 还取决于属性 $C,D$ 。

➤ 表6.6: 有 $R$ 的一个关系实例, 在此实例上 $A \twoheadrightarrow B$ 是不成立的

表6.6  $R$ 的一个实例

	A	B	C	D
$t$	$a_1$	$b_1$	$c_1$	$d_1$
$w$	$a_1$	$b_1$	$c_1$	$d_2$
$v$	$a_1$	$b_2$	$c_2$	$d_1$
$s$	$a_1$	$b_2$	$c_2$	$d_2$



# 多值依赖与函数依赖的区别

108

## (2) 多值依赖的有效性与属性集的范围有关

- $X \twoheadrightarrow Y$  若在  $R(U)$  上成立, 不能断言对于任何  $Y' \subset Y$  有  $X \twoheadrightarrow Y'$  成立
  - 例如, 表6.6,  $A \twoheadrightarrow BC$  成立的时候,  $A \twoheadrightarrow B$  不一定成立, 因为  $B$  的取值不仅取决于属性  $A$ , 还取决于属性  $C, D$ 。

	A	B	C	D
$t$	a1	b1	c1	d1
$w$	a1	b1	c1	d2
$v$	a1	b2	c2	d1
$s$	a1	b2	c2	d2

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  $w$   
 $w[X]=v[X]=t[X]$ , 而  
 $w[Y]=t[Y]$ ,  
 $w[Z]=s[Z]$ ,  
 $v[Y]=s[Y]$ ,  $v[Z]=t[Z]$



# 多值依赖与函数依赖的区别

(2)  $A \twoheadrightarrow B$ 成立的情况

		A	B	C	D
t w v s		a1	b1	c1	d1
		a1	b1	c1	d2
		a1	b2	c2	d1
		a1	b2	c2	d2
		a1	b1	c2	d1
		a1	b1	c2	d2
		?	?	?	?
		?	?	?	?





# 练习

110

关系U中有属性ABCD，包含如下记录，请问最后两个元组中的A和D的值分别为多少时，多值依赖 $BC \twoheadrightarrow A$ 才有可能成立？

		Y	X		Z
		A	B	C	D
t		2	4	3	3
		2	3	5	3
s		1	4	3	2
		3	1	2	2
w		1	4	3	3
v		2	4	3	2



## 6.2 规范化

111

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

112

- 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 $X \twoheadrightarrow Y$  ( $Y \subsetneq X$ )， $X$ 都含有码，则 $R \in 4NF$ 。
- 4NF：限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。
- 4NF所允许的非平凡多值依赖实际上是函数依赖
  - $X \twoheadrightarrow Y$ ， $Z$ 不为空， $X$ 含有码
- 4NF消除了非主属性对候选码以外属性的多值依赖



## 6.2.8 4NF

113

- 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（非1:1）的多值依赖（1:n）
    - 不允许像Teaching（教员，课程，参考书）关系中的多值依赖存在
  - 即，允许的非平凡多值依赖是函数依赖（1:1）
    - 不会有冗余

**NOTE:** 若 $X \twoheadrightarrow Y$ ，而 $Z = \varnothing$ （空），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

不用管平凡多值依赖，因为它是合理的（平凡多值依赖是4NF）



# 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 (C只是码的一部分), 这是存在不好性质的根源

- 数据冗余
- 插入复杂
- 删除操作复杂
- 修改操作复杂

课程C	教师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



# 4NF

115

- 用投影分解法把Teaching分解为如下两个关系模式：

$$CT(C, T) \in 4NF, \quad CB(C, B) \in 4NF$$

➤  $C \twoheadrightarrow T$ ,  $C \twoheadrightarrow B$  是平凡多值依赖

解决上述问题

- 数据冗余：参考书只需要CB中存储一次
- 插入操作：课程增加一名教师时，只需CT插入
- 删除操作：某课程去掉一本参考书，只需CB中删除一元组
- 修改操作：某课程修改一本参考书，只需修改CB



## 4NF (续)

116

- ❖ [例6.10]的WSC中,
  - ❖ 关系模式WSC的码是(W,S,C), 即All-key
  - ❖ 存在 $W \twoheadrightarrow S$ ,  $W \twoheadrightarrow C$ ,他们都是非平凡多值依赖。而W不是码,
  - ❖ 因此 $WSC \notin 4NF$ 。
- ❖ 可以把WSC分解成 $WS(W,S), WC(W,C)$ 
  - ❖  $WS \in 4NF$
  - ❖  $WC \in 4NF$



## 4NF (续)

117

确定一个关系模式是否为4NF（一般可以分为三个步骤）：

- 1.确定该关系模式的码
- 2.判断关系模式里的多值依赖是否是平凡多值依赖，如果是平凡多值依赖，则该范式是4NF
- 3.如果多值依赖是非平凡多值依赖，判断该多值依赖的决定因素（左半部分）是否包含码。如果包含码，则该范式是4NF，否则，该范式不是4NF





## 6.2 规范化

118

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

119

- 在关系数据库中
  - 对关系模式的基本要求是满足第一范式。
- 规范化程度过低的关系不一定能够很好地描述现实世界
  - 可能存在插入异常、删除异常、修改复杂、数据冗余等问题
  - 解决方法就是对其进行规范化，转换成高级范式。
- 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集合，这种过程就叫关系模式的规范化。



## 6.2.9 规范化小结

120

- 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
- 规范化的基本思想
  - 逐步消除数据依赖中不合适的部分，使模式中的各关系模式达到某种程度的“分离”
  - 即采用“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。
  - 实质：概念的单一化



# 规范化小结（续）

121

## □ 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



# 规范化小结（续）

122

- ❑ 不能说规范化程度越高的关系模式就一定越好
- ❑ 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

133

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

134

### □ 回顾规范化理论

- 函数依赖，部分函数依赖，传递函数依赖，成为判断关系模式是好是坏的因素
- 问题：如何找到全部的函数依赖？
  - 现实世界的语义约束
  - 有没有别的办法？



# 1. Armstrong公理系统

135

- 从已知的一些函数依赖**F**，可以推导出其蕴含的另外一些函数依赖 $\{X \rightarrow Y\}$ ，需要一系列推理规则，这些规则常被称作“**Armstrong 公理**”。
- Armstrong公理系统
  - 一套推理规则，是模式分解算法的理论基础
  - 用途
    - 求给定关系模式的码
    - 从一组函数依赖求得蕴涵的函数依赖

已知:  $R\langle U, F \rangle$ ,  $U=\{X, C, W, Y, Z\}$ ,  $F=\{X \rightarrow YZ, Z \rightarrow CW\}$   
问:  $X \rightarrow CWYZ$ 是否为F逻辑蕴含? R的码?





## 6.3 数据依赖的公理系统

136

### □ 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

137

## □ Armstrong公理系统（推理系统）

设 $U$ 为属性集总体， $F$ 是 $U$ 上的一组函数依赖，于是有关系模式 $R \langle U, F \rangle$ 。对 $R \langle U, F \rangle$ 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(并集)
- 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



## 定理 6.1 Armstrong推理规则是正确的

138

**A1. 自反律:** 若  $Y \subseteq X \subseteq U$ , 则  $X \rightarrow Y$  为  $F$  所蕴含

证明:

设  $Y \subseteq X \subseteq U$

对  $R \langle U, F \rangle$  的任一关系  $r$  中的任意两个元组  $t, s$ :

若  $t[X]=s[X]$ , 由于  $Y \subseteq X$ , 有  $t[Y]=s[Y]$ ,

所以  $X \rightarrow Y$  成立,

自反律得证

**注意:** 由自反律所得到的函数依赖均是平凡的函数依赖,  
自反律的使用并不依赖于  $F$ 。



## 定理 6.1 Armstrong推理规则是正确的（续）

139

**A2. 增广律:** 若 $X \rightarrow Y$ 为 $F$ 所蕴含, 且 $Z \subseteq U$ , 则 $XZ \rightarrow YZ$  为 $F$ 所蕴含

证明: 设 $X \rightarrow Y$ 为 $F$ 所蕴含, 且 $Z \subseteq U$ 。

设 $R \langle U, F \rangle$  的任一关系 $r$ 中任意的两个元组 $t, s$ :

若 $t[XZ]=s[XZ]$ , 则有 $t[X]=s[X]$ 和 $t[Z]=s[Z]$ ;

因为 $X \rightarrow Y$ , 则 $t[Y]=s[Y]$ , 所以 $t[YZ]=s[YZ]$ ,

所以:  $XZ \rightarrow YZ$ 为 $F$ 所蕴含

增广律得证。



## 定理 6.1 Armstrong推理规则是正确的（续）

140

**A3. 传递律：** 若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 $F$ 所蕴含，则 $X \rightarrow Z$ 为 $F$ 所蕴含。

证明： 设 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 $F$ 所蕴含。

对 $R \langle U, F \rangle$  的任一关系  $r$  中的任意两个元组  $t, s$ ：

若 $t[X]=s[X]$ ，由于 $X \rightarrow Y$ ，有 $t[Y]=s[Y]$ ；

再由 $Y \rightarrow Z$ ，有 $t[Z]=s[Z]$ ，所以 $X \rightarrow Z$ 为 $F$ 所蕴含，

传递律得证。



## 2. 导出规则

141

□ 1. 根据A1, A2, A3可以得到下面三条推理规则:

- 合并规则: 由 $X \rightarrow Y$ ,  $X \rightarrow Z$ , 有 $X \rightarrow YZ$
- 伪传递规则: 由 $X \rightarrow Y$ ,  $WY \rightarrow Z$ , 有 $XW \rightarrow Z$
- 分解规则: 由 $X \rightarrow Y$ 及  $Z \subseteq Y$ , 有 $X \rightarrow Z$

□ 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  
( $i=1, 2, \dots, k$ )

证明课后思考



## 2. 导出规则

142

- 1. 根据A1, A2, A3可以得到下面三条推理规则:

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

- $(A2, A3) X \rightarrow YX, XY \rightarrow ZY$

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

- $(A2, A3) XW \rightarrow YW$

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

- $(A1, A3) Z \subseteq Y, Y \rightarrow Z$

- 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

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



# F的闭包

143

- **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 $F^+$ 。
- Armstrong公理系统（A1,A2,A3）是有效的、完备的
  - 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；
  - 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





# Armstrong公理系统

144

- 要证明完备性，就首先要解决如何判定一个函数依赖是否属于由F根据Armstrong公理系统推导出来的函数依赖的集合
- 如果能够求出这个集合，问题就解决了
- 但是，这个是一个NP完全问题



## F的闭包

145

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

$F^+=\{$

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

$\}$

$F=\{X \rightarrow A_1, \dots, X \rightarrow A_n\}$ , 求F的闭包 $F^+$  计算是一个NP完全问题



### 3. 函数依赖闭包

146

**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 $F^+$ 。



**定义6.13** 设 $F$ 为属性集 $U$ 上的一组函数依赖， $X \subseteq U$ ，

**$X_F^+ = \{ A \mid X \rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出} \}$** ， $X_F^+$

称为属性集 **$X$ 关于函数依赖集 $F$ 的闭包**



## 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。



## 算法6.1

149

- 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。



# 函数依赖闭包

150

(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 $X^{(2)} = X^{(1)} \cup BE = ABCDE$



## 练习

151

- 例：设有关系模式 $R(U, F)$ ，其中
- $U = \{A, B, C, D, E, I\}$
- $F = \{A \rightarrow D, AB \rightarrow E, BI \rightarrow E, CD \rightarrow I, E \rightarrow C\}$
- 请计算  $(AE)_F^+$





## 4. Armstrong公理系统的有效性与完备性

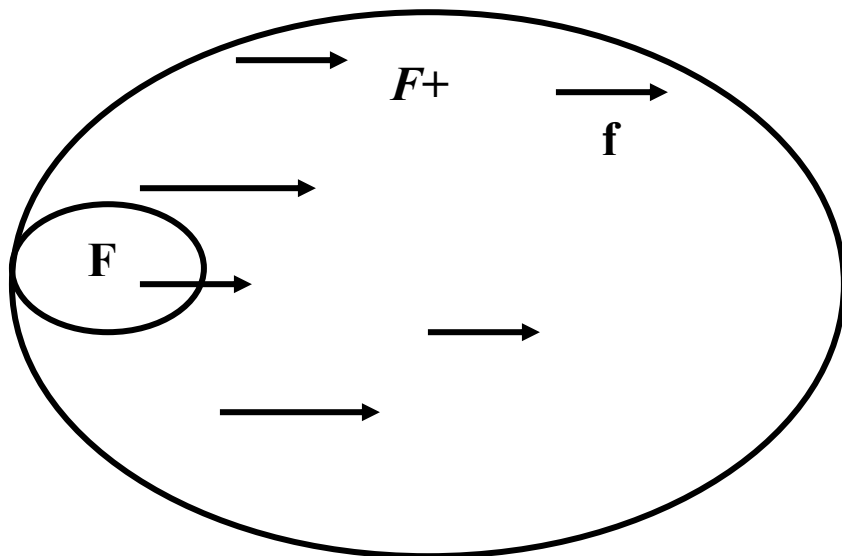
152

□ 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?

□



$f$  不能由  $F$  导出,  $f \notin F$





## 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 $Y \subseteq X_F^+$



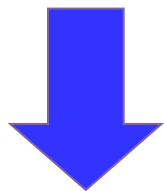
# Armstrong公理系统完备性证明

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:

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

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

设 $F' = \{ Sno \rightarrow Sdept, Sno \rightarrow Mname, Sdept \rightarrow Mname,$

$(Sno, Cno) \rightarrow Grade, (Sno, Sdept) \rightarrow Sdept \}$

$F$ 是最小覆盖, 而 $F'$ 不是。

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

$F' - \{ (Sno, Sdept) \rightarrow Sdept \}$ 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

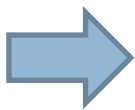
178

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

□ 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 $\rho$ 中各关系模式上投影的自然连接成立, 则称分解 $\rho$ 具有无损连接性。简称 $\rho$ 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6.2 判别一个分解的无损连接性。 p.197



## 2. 保持函数依赖的模式分解

187

[定义6.19]:  $\rho=\{R_1, \dots, R_n\}$ 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.193



### 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

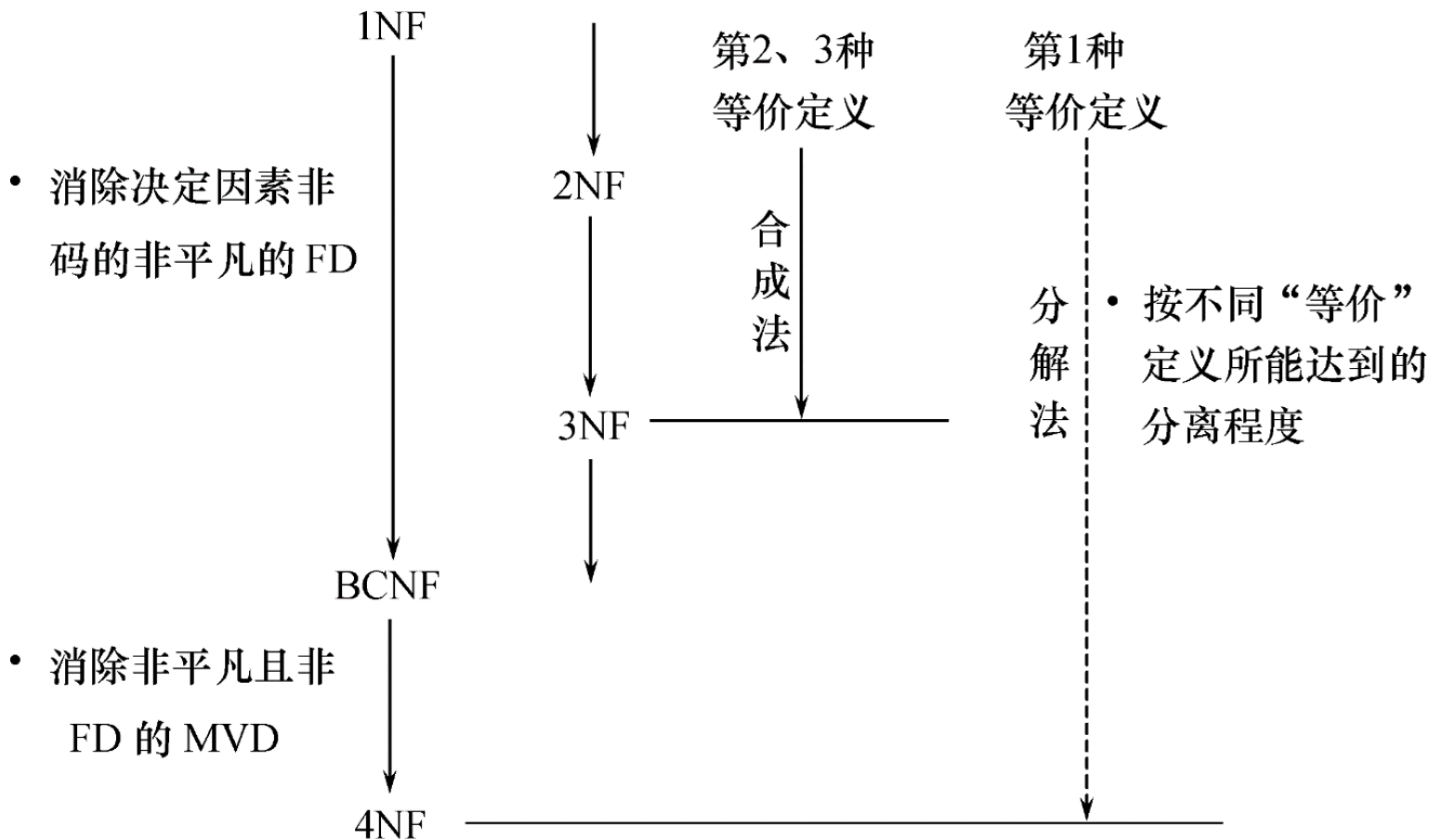
6.5 小结



## 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



## 小结(续)

195

- 规范化理论为数据库设计提供了理论的指南和工具
  - 也仅仅是指南和工具
  
- 并不是规范化程度越高，模式就越好
  - 必须结合应用环境和现实世界的具体情况合理地选择数据库模式