



2026年春季学期

# 数据库系统概论

An Introduction to Database Systems

## 第六章 关系数据理论

中国科学技术大学

人工智能与数据科学学院

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

- 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。
- $Sname = f(Sno)$ ,  $School = f(Sno)$ 
  - 即Sno函数决定Sname
  - Sno函数决定School
  - 记作 $Sno \rightarrow Sname$ ,  $Sno \rightarrow School$



## 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)、所在学院 (School)、所在学院院长姓名 (Mname)、选修课的课程号 (Cno)、成绩 (Grade)

### ➤ 语义

- 一个学院有若干学生，但一个学生只属于一个学院；
- 一个学院只有一名（正职）院长；
- 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**：  $\text{Student} \langle U, F \rangle$

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

思考 教务处数据库的语义？



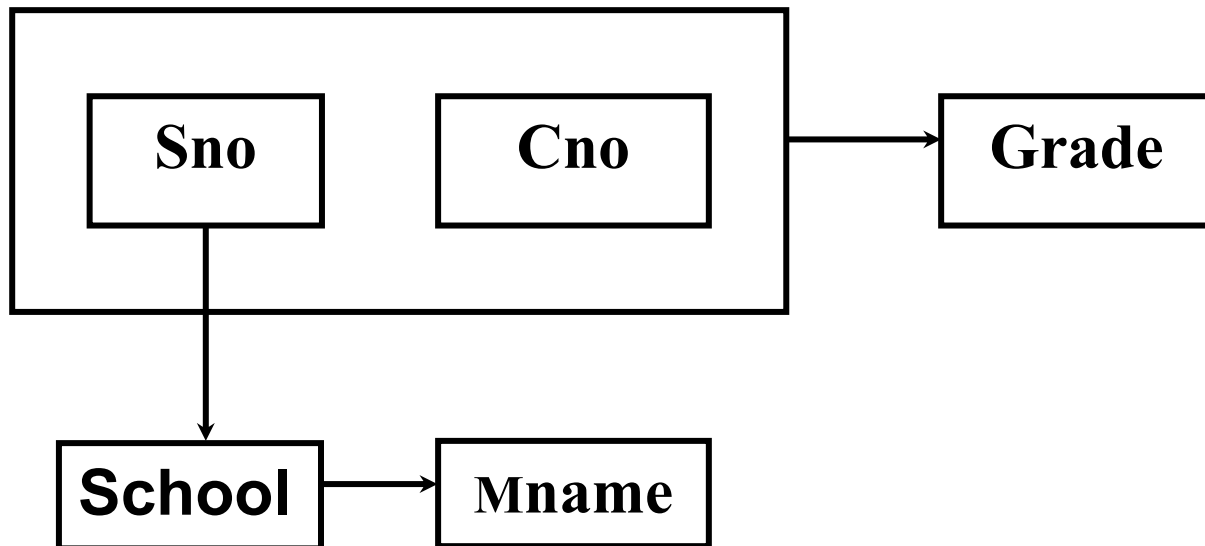
## 数据依赖对关系模式的影响（续）

18

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

- 现实世界的语义
- 属性组U上的一组函数依赖F:

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





# 数据依赖对关系模式的影响（续）

表6.1 Student表

Sno	School	Mname	Cno	Grade
S1	信息学院	张明	C1	95
S2	信息学院	张明	C1	90
S3	信息学院	张明	C1	88
S4	信息学院	张明	C1	70
S5	信息学院	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



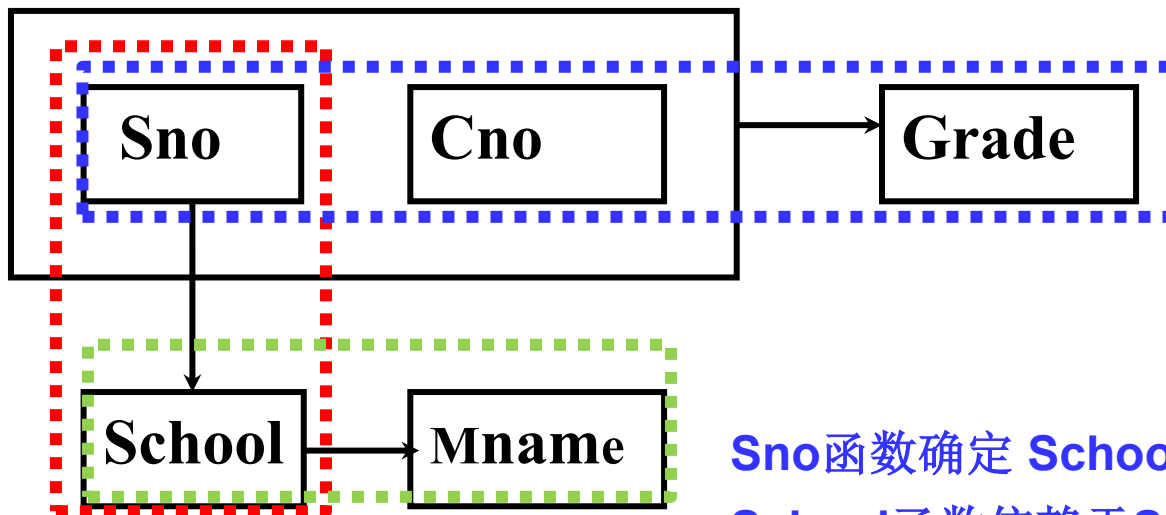
# 分解关系模式

□ 把这个单一模式分成3个关系模式:

**S (Sno, School, Sno → School) ;**

**SC (Sno, Cno, Grade, (Sno, Cno) → Grade) ;**

**SCH (School, Mname, School → Mname)**



Sno函数确定 School  
School函数依赖于Sno



# 第六章 关系数据理论

22

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

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 $R(U, F)$ 是一个属性集 $U$ 上的关系模式， $X$ 和 $Y$ 是 $U$ 的子集。

若对于 $R(U, F)$ 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 $X \not\rightarrow Y$ 。



## 函数依赖（续）

28

❖ [例] Student (Sno, Sname, Ssex, Sage, Sdept),

假设不允许重名，则有：

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

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

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

$Sname \rightarrow Sdept$

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

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

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



# 函数依赖（续）

❖ 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）

❖ 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

定义6.1补充:

在关系模式 $R(U, F)$ 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:  $(Sno, Cno) \rightarrow Sno$

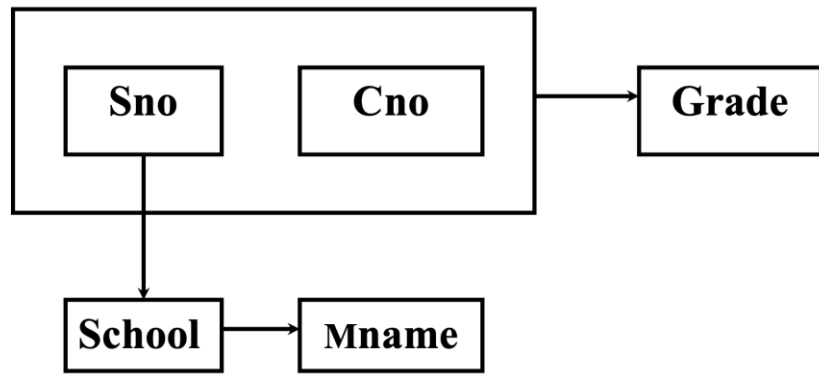
$(Sno, Cno) \rightarrow Cno$

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



定义6.2 在R(U, F)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]  $U = \{ Sno, School, Mname, Cno, Grade \}$

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

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

?  $\xrightarrow{F} U$

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



### 三、完全函数依赖与部分函数依赖

34

❖ [例] 在关系SC(Sno, Cno, Grade, Semester)中, 有:

- 由于: Sno  $\twoheadrightarrow$  Grade, Cno  $\twoheadrightarrow$  Grade, Sno  $\twoheadrightarrow$  Semester, Cno  $\twoheadrightarrow$  Semester,

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

(Sno, Cno)  $\xrightarrow{\mathbf{F}}$  Semester

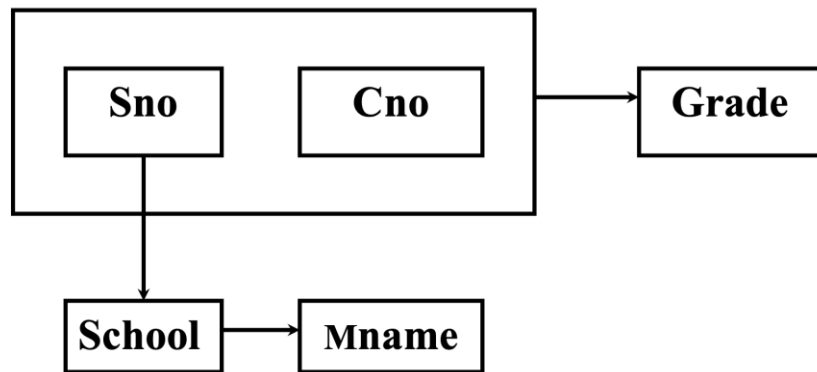
(Sno, Cno)  $\xrightarrow{\mathbf{P}}$  Sno

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

\*



# 四、传递函数依赖



定义6.3 在R(U, F)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:  $U = \{ Sno, School, Mname, Cno, Grade \}$ , 有:

$Sno \rightarrow School, School \rightarrow Mname,$

$Sno \xrightarrow{\text{传递}} Mname$



## 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 $R(U, F)$ 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, School, 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]

第二章的关系模式Student (Sno, Sname, Ssex, Sage, Sdept),  
单个属性Sno是码,

SC (Sno, Cno, Grade, Semester)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  $SC$  ( $Sno$ ,  $Cno$ , Grade, Semester) 中,  $Sno$  不是码。
- 但  $Sno$  是关系模式 Student ( $Sno$ , Sname, Ssex, Sage, Sdept) 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

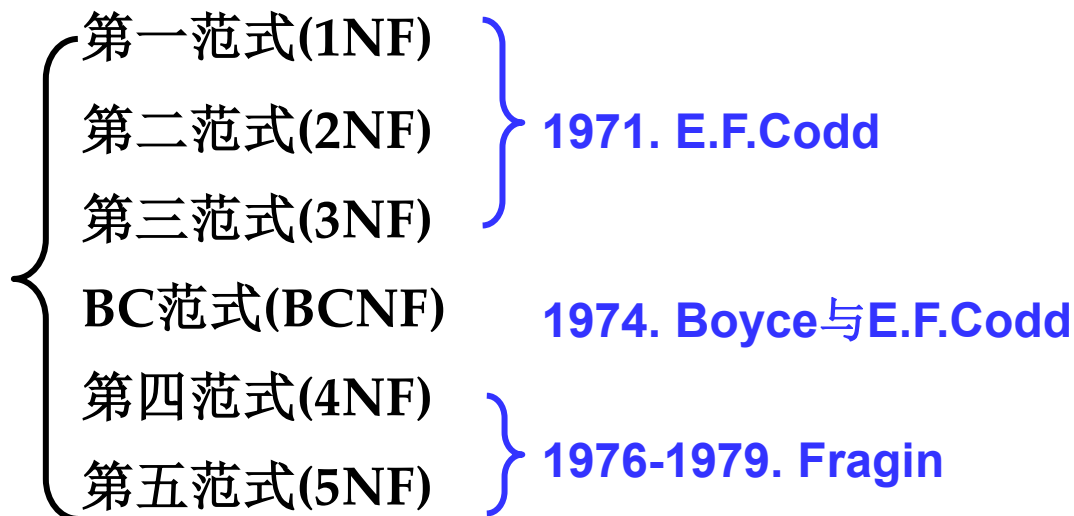
6.2.9 规范化小结



## 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：



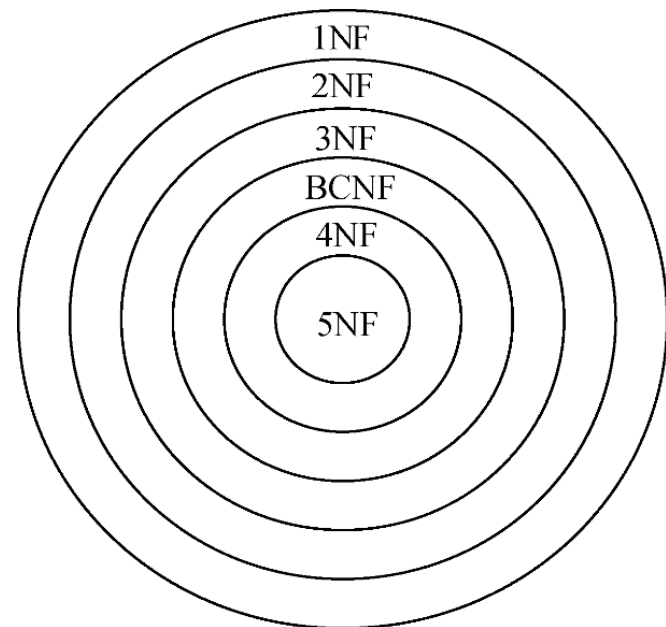


## 6.2.3 范式

- 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的所有属性都是**不可分的基本数据项**，  
则 $R \in 1NF$

- 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

### □ 1NF的例子

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，R是满足最低要求的模式，则 $R \in 1NF$

Sno	School	Mname	Cno	Grade
S1	信息学院	张明	C1	95
S2	信息学院	张明	C1	90
S3	信息学院	张明	C1	88
S4	信息学院	张明	C1	70
S5	信息学院	张明	C1	78
....	...	...	...	...



# 2NF—1NF的问题

[例6.4] 关系模式 S-L-C (Sno, School, Sloc, Cno, Grade)

Sloc为学生住处，假设每个学院的学生住在同一个地方

□ 函数依赖包括：

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

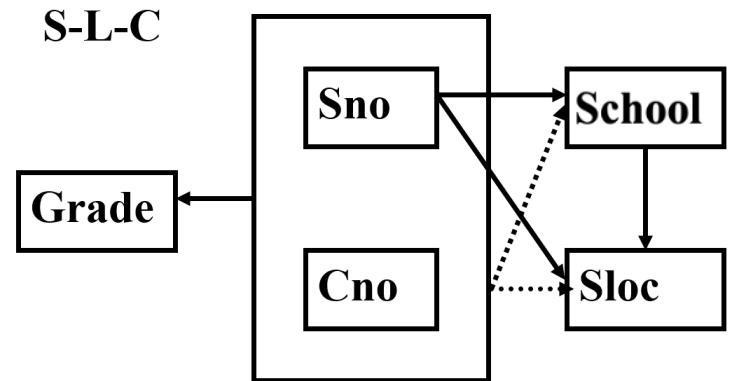
$$Sno \rightarrow School$$

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

$$Sno \rightarrow Sloc$$

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

$$School \rightarrow Sloc$$



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

关系模式 S-L-C(Sno, School, 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School Sloc等信息丢失。

## (3) 数据冗余度大

- Sdept, Sloc重复存储

Sno	School	Sloc	Cno	Grade
....	...	...	...	...

## (4) 修改复杂

- 修改学生系别School时需要同时修改Sloc。如果一个学生选了很多门课，则School, Sloc被存储了多次。如果该生转系，则需要修改所有相关的School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School、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

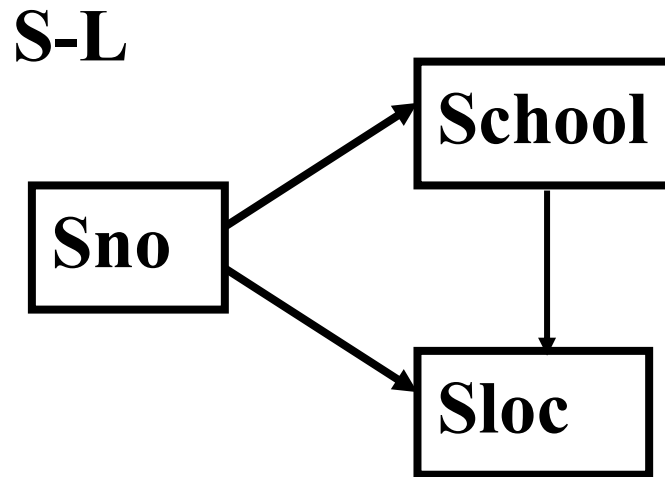
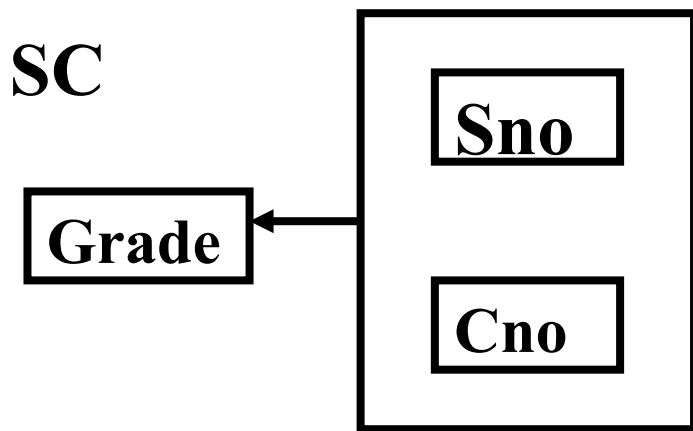
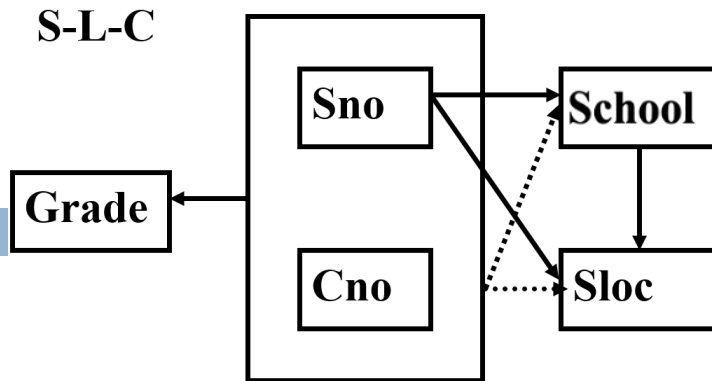
SC (Sno, Cno, Grade)

S-L (Sno, School, 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
- ❖ 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, School, Sloc, Cno, Grade)  $\in 1NF$

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

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

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



# 2NF (续)

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

## 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?

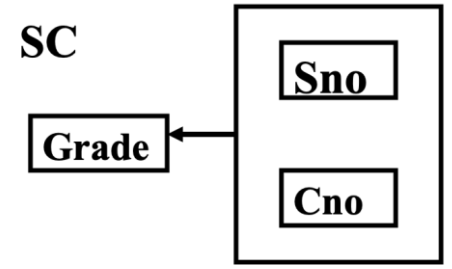
- 删除一个学生所有选课记录，只是SC关系没有记录。SL关系关于该学生不受影响

### 数据冗余度?

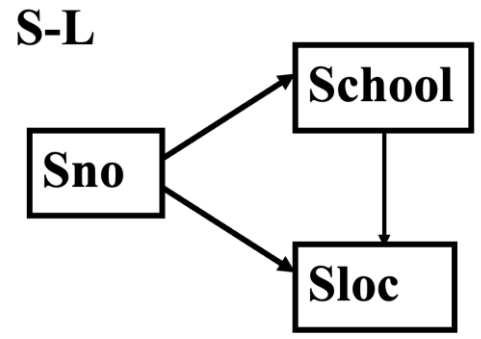
- 不论一个学生选多少课，School和Sloc值都只存储1次（SL关系中）

### 修改复杂?

- 学生转系：只修改SL关系即可。



Sno	Cno	Grade
...	...	...



Sno	School	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**