

# 数据库技术

Database System Technology

郭捷

([guojie@sjtu.edu.cn](mailto:guojie@sjtu.edu.cn))

饮水思源 · 爱国荣校

# 第三章 关系规范化基础



1

问题的提出

2

数据依赖

3

关系规范化

4

数据依赖的公理系统

## 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模式？

一个关系数据库模式由一个面向具体应用所涉及的若干个关系模式所组成，这些关系模式通过外码建立相互联系，形成一个结构化的数据整体。

- ◆ **数据库逻辑设计**的工具——关系数据库的规范化理论

如何构造适合于具体应用的数据库模式，即应该构造几个关系，每个关系由哪些属性组成等，是关系数据库的逻辑设计问题。

# 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：**用来定义关系。
- ▣ **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：

$$R(U, \underline{D}, \text{DOM}, F) \longrightarrow R(U, F)$$

- 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



## 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连（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；

# 三、什么是数据依赖



## 2. 数据依赖

- 是通过一个关系中属性间值的相等与否体现出来的数据间的相互关系；
- 是现实世界属性间相互联系的抽象，是数据内在的性质，是语义的体现；
- 常见的数据依赖：
  - ✓ 函数依赖
  - ✓ 多值依赖

### 三、什么是数据依赖



例：学生（学号，姓名，所在系，……）

一个学号对应一名学生，一名学生只能属于一个系，因而当“学号”的值确定后，姓名以及所在系的值也就被唯一确定了。

属性间的这种依赖关系类似于数学中的函数  $y=f(x)$ ，可写为 **Sname = f(Sno)**，**Sdept = f(Sno)**。

因此说“学号” **函数决定** “姓名”和“所在系”，或者说“姓名”和“所在系” **函数依赖**于“学号”。

记作：Sno  $\rightarrow$  Sname；Sno  $\rightarrow$  Sdept ；



## 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：

$$R(U, F)$$

- 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；

# 五、数据依赖对关系模式的影响



例：描述学校的数据库：

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 (U、F)

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

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
⋮	⋮	⋮	⋮	⋮

# 五、数据依赖对关系模式的影响



学校数据库的语义：

1. 一个系只有一名主任；
2. 一个系有若干学生；
3. 一个学生只属于一个系；
4. 一个学生可以选修多门课程， 每门课程有若干学生选修；
5. 每个学生所学的每门课程都有一个成绩；

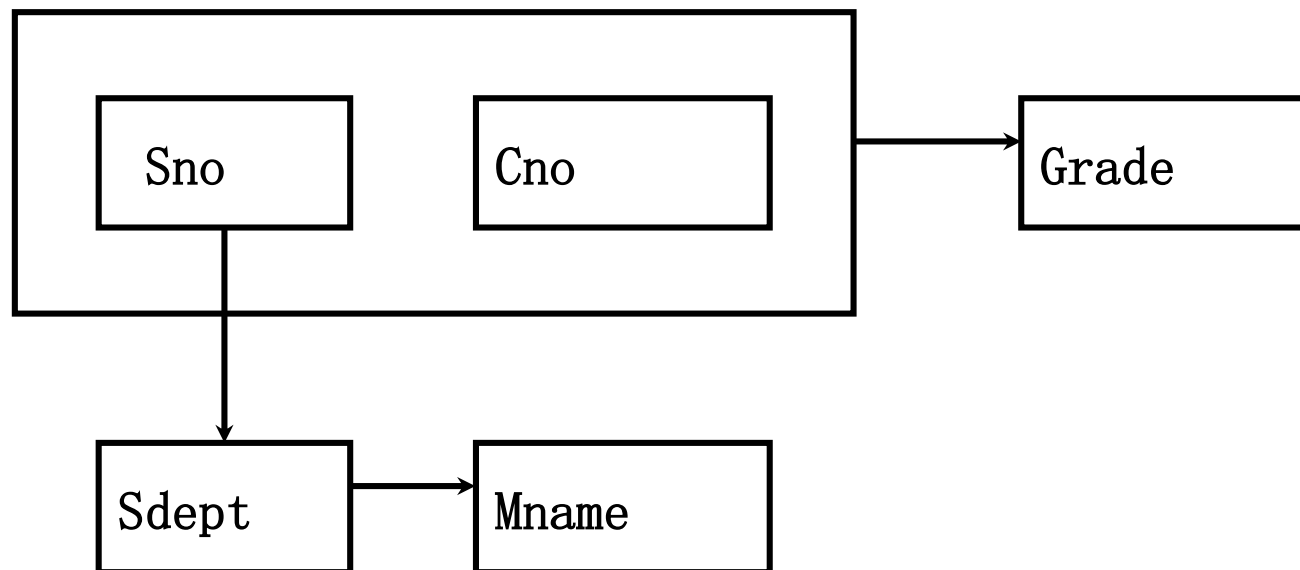
Sno	Sdept	Mname	Cno	Grade
S1	计算机系	张明	C1	95
S2	计算机系	张明	C1	90
S3	计算机系	张明	C1	88
S4	计算机系	张明	C1	70
S5	计算机系	张明	C1	78
⋮	⋮	⋮	⋮	⋮

# 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:

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



# 关系模式Student<U, F>中存在的问题



## 1. 数据冗余太大

- ◆ 浪费大量的存储空间

例：每一个系主任的姓名重复出现。

## 2. 更新异常 (Update Anomalies)

- ◆ 数据冗余，更新数据时，维护数据完整性代价大。

例：某系更换系主任后，系统必须修改与该系学生有关的每一个元组。

# 关系模式Student<U, F>中存在的问题



## 3. 插入异常 (Insertion Anomalies)

- ◆ 该插的数据插不进去

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:

把这个单一模式分解为3个关系模式

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

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

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



# 第三章 关系规范化基础



1

问题的提出

2

数据依赖

3

关系规范化

4

数据依赖的公理系统



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。

# 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 $X \rightarrow Y$ 。
- ◆  $X$ 称为这个函数依赖的决定属性组，又称决定因素 (Determinant)
- ◆ 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 $X \nrightarrow Y$

# 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 $\rightarrow$ 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 $\rightarrow$ 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。

## 函数依赖 (续)



例: Student (Sno, Sname, Ssex, Sage, Sdept)

假设不允许重名, 则有:

$Sno \rightarrow Ssex, Sno \rightarrow Sage, Sno \rightarrow Sdept,$

$Sno \leftrightarrow Sname, Sname \rightarrow Ssex, Sname \rightarrow Sage$

$Sname \rightarrow Sdept$

但  $Ssex \rightarrow Sage$

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  $X \nrightarrow Y$ 。

## 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;

平凡函数依赖都是必然成立的!

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:

$$(Sno, Cno) \rightarrow Sno \quad (Sno, Cno) \rightarrow Cno$$

# 平凡函数依赖与非平凡函数依赖（续）



- ◆ 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，因此若不特别声明，我们总是讨论非平凡函数依赖。

### 三、完全函数依赖与部分函数依赖



定义6.2 在关系模式 $R(U)$ 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

$Y$ 完全函数依赖于 $X$ , 记作 $X \xrightarrow{F} Y$ 。

✚ 如果 $X \rightarrow Y$ , 但 $Y$ 不完全函数依赖于 $X$ , 则称 $Y$ 部分函数依赖于 $X$ ,

记作 $X \xrightarrow{P} Y$ 。



# 完全函数依赖与部分函数依赖 (续)



例6.1 Student (Sno, Sdept, Mname, Cno, Grade)

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

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

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

## 四、传递函数依赖



定义6.3 在关系模式 $R(U)$ 中, 如果 $X \rightarrow Y$  ( $Y \subsetneq X$ ),  $Y \not\rightarrow X$ ,  $Y \rightarrow Z$  ( $Z \subsetneq Y$ ), 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 $X \xrightarrow{\text{直接}} Z$ 。

例: 在关系 $Std(Sno, Sdept, Mname)$ 中, 有:

$Sno \rightarrow Sdept, Sdept \rightarrow Mname$

$Sno \xrightarrow{\text{传递}} Mname$

定义6.4 设 $K$ 为关系模式 $R\langle U, F \rangle$ 中的属性或属性组合。若 $K \xrightarrow{F} U$ ，则 $K$ 称为 $R$ 的一个  
**候选码** (Candidate Key)。

- 若关系模式 $R$ 有多个候选码，则选定其中的一个做为**主码** (Primary key)。
- 若 $K \xrightarrow{P} U$ ，则 $K$ 为 $R$ 的**超码**。候选码是最小的超码。



- ✓ 包含在任何一个候选码的诸属性称为主属性。
- ✓ 不包含在任何侯选码中的属性称为非主属性。
- ✓ 在最极端的情况下，关系模式的所有属性组是这个关系模式的候选码，称为全码 (All-key)。

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。

例：SC(Sno, Cno, Grade)

- 主码和外部码一起提供了表示**关系间联系**的手段

# 第三章 关系规范化基础



1

问题的提出

2

数据依赖

3

关系规范化

4

数据依赖的公理系统

# 范式的概念



- 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：

第一范式 (1NF)

低

第二范式 (2NF)

第三范式 (3NF)

BC范式 (BCNF)

第四范式 (4NF)

第五范式 (5NF)

高





- 规范化程度过低的关系不一定能够很好地描述现实世界，可能会存在插入异常、删除异常、修改复杂、数据冗余等问题；
- 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**。





- 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 $R \in nNF$ 。

# 1NF (第一范式)



- 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。

# 1NF (第一范式)



例：借阅图书关系

借阅证号	姓名	性别	借阅图书登记					
			图书号1	书名1	图书号2	书名2	图书号3	书名3

非规范化的关系 → 第一范式的关系

✓ **方法一：** 在属性上展开

借阅证号	姓名	性别	图书号1	书名1	图书号2	书名2	图书号3	书名3
------	----	----	------	-----	------	-----	------	-----

# 1NF (第一范式)



✓ **方法二：** 在元组上展开

借阅证号	姓名	性别	图书号	书名
------	----	----	-----	----

✓ **方法三：** 模式分解

借阅证号	姓名	性别
------	----	----

借阅证号	图书号	书名	借阅日期	归还日期
------	-----	----	------	------

例：关系模式 SLC(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：

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

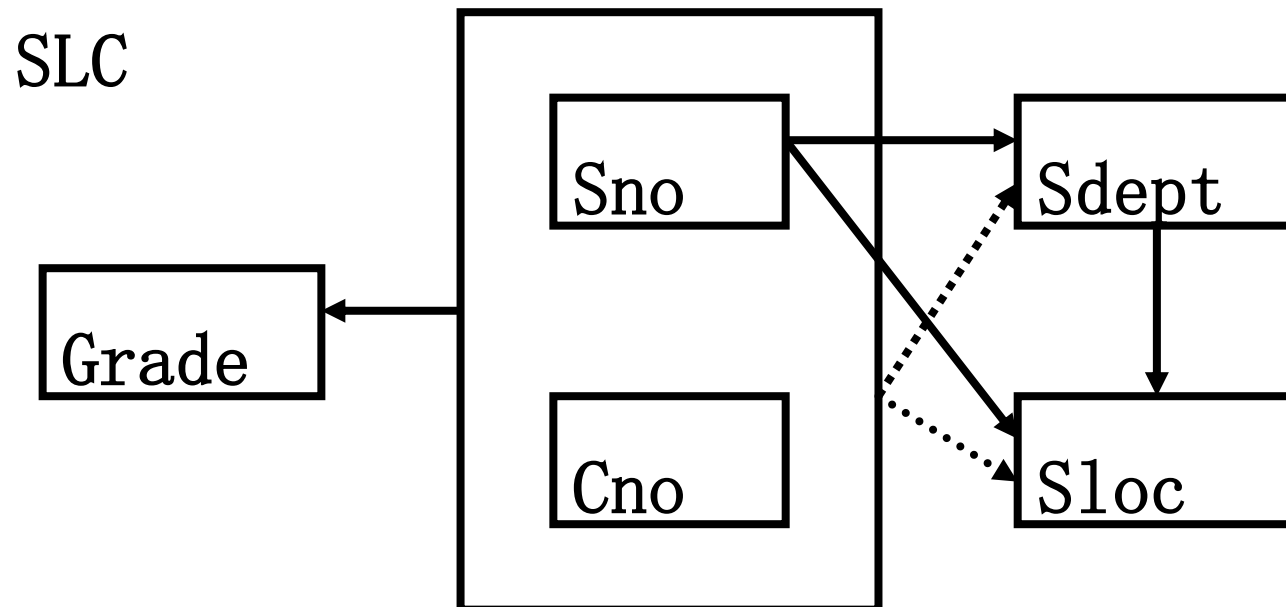
$\text{Sno} \rightarrow \text{Sdept}$

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

$\text{Sno} \rightarrow \text{Sloc}$

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

$\text{Sdept} \rightarrow \text{Sloc}$



- SLC的码为(Sno, Cno)
- SLC满足第一范式。
- 非主属性Sdept和Sloc部分函数依赖于码(Sno, Cno)

# SLC不是一个好的关系模式



SLC(Sno, Sdept, Sloc, Cno, Grade)

## (1) 插入异常

假设Sno=95102, Sdept=IS, Sloc=N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。

# SLC不是一个好的关系模式



SLC(Sno, Sdept, Sloc, Cno, Grade)

## (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。





## 原因

Sdept、Sloc部分函数依赖于码。

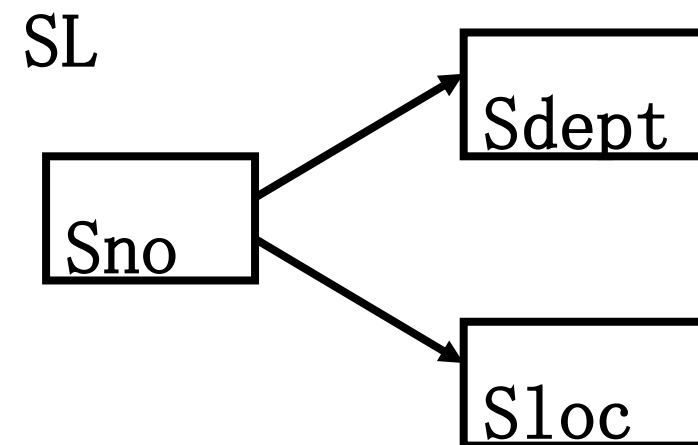
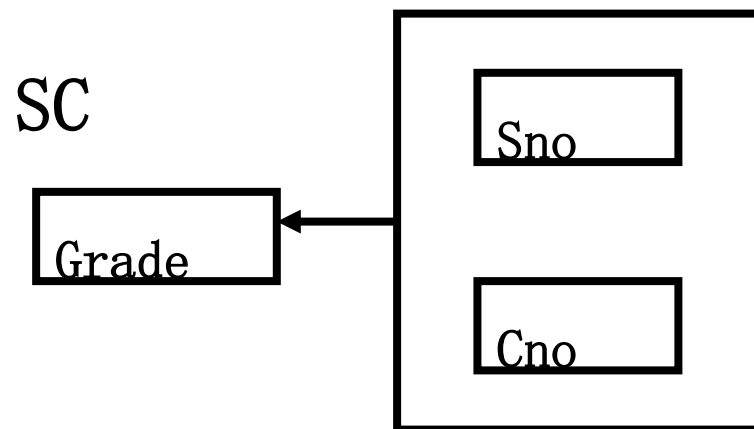
## 解决方法

SLC关系分解为两个关系模式，以消除这些部分函数依赖

SC (Sno, Cno, Grade)

SL (Sno, Sdept, Sloc)

## 函数依赖图:



# 2NF的定义



定义6.6 若关系模式 $R \in 1NF$ ，并且每一个非主属性都完全函数依赖于R的码，则 $R \in 2NF$ 。

例：SLC( Sno, Cno, Sdept, Sloc, Grade)  $\in 1NF$

SLC( Sno, Cno, Sdept, Sloc, Grade)  $\notin 2NF$

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

SL( Sno, Sdept, Sloc)  $\in 2NF$

## 第二范式 (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。
- 达到2NF的关系是不是就不存在问题？

例：2NF关系模式SL (Sno, Sdept, Sloc) 中

## ▣ 函数依赖：

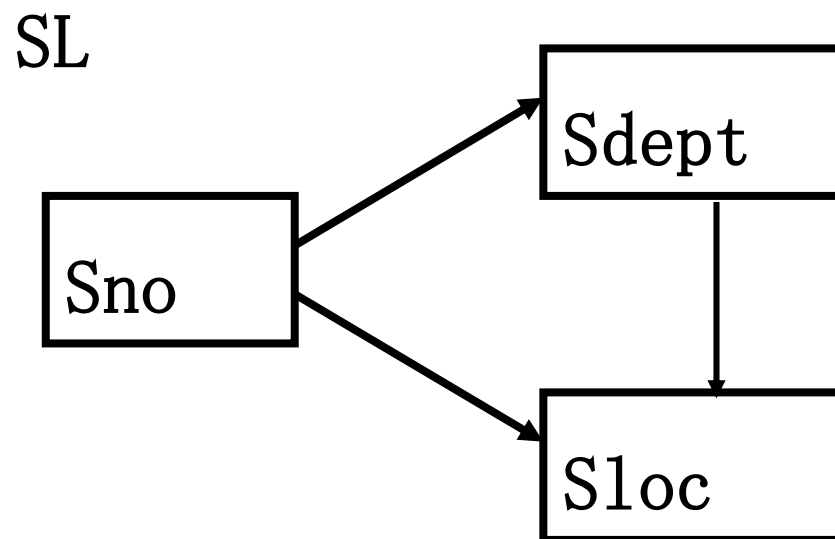
$Sno \rightarrow Sdept$

$Sdept \rightarrow Sloc$

$Sno \rightarrow Sloc$

Sloc传递函数依赖于Sno，即SL中存在**非主属性对码的传递函数依赖**。

函数依赖图：



会出现与2NF相类似的问题 ！

## ■ 解决方法:

采用关系分解法，将具有传递函数依赖关系的属性（组）逐层提取出来。

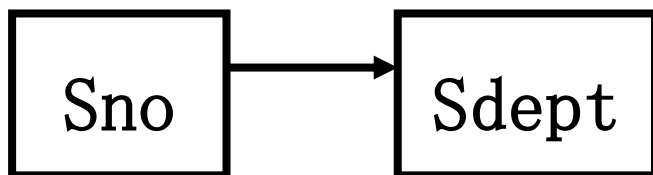
例如：把SL分解为两个关系模式，以消除传递函数依赖：

SD (Sno, Sdept)

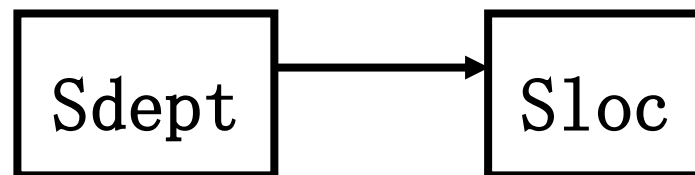
DL (Sdept, Sloc)

SD的码为Sno， DL的码为Sdept。

函数依赖图：SD的码为Sno， DL的码为Sdept。



SD



DL





## ▣ 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 $Z$  ( $Z \not\subseteq Y$ )，使得 $X \rightarrow Y$ ,  $Y \not\rightarrow X$ ,  $Y \rightarrow Z$  成立，则称 $R\langle U, F \rangle \in 3NF$ 。

例,  $SL(Sno, Sdept, Sloc) \in 2NF$

$SL(Sno, Sdept, Sloc) \notin 3NF$

$SD(Sno, Sdept) \in 3NF$

$DL(Sdept, Sloc) \in 3NF$

- 若 $R \in 3NF$ ，则R的每一个非主属性既不部分函数依赖于候选码也不传递函数依赖于候选码。
- 如果 $R \in 3NF$ ，则R也是2NF。
- 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，并不能完全消除关系模式中的各种异常情况和数据冗余。

# BC范式 (BCNF)



定义6.8 设关系模式 $R\langle U, F \rangle \in 1NF$ ，如果对于 $R$ 的每个函数依赖 $X \rightarrow Y$ ，若 $Y \not\subseteq X$ ，则 $X$ 必含有码，那么 $R \in BCNF$ 。

若 $R \in BCNF$ ：

- ▣ 每一个决定属性集（因素）都包含（候选）码；
- ▣  $R$ 中的所有属性（主，非主属性）都完全函数依赖于码；
- ▣ 没有任何属性完全函数依赖于非码的任何一组属性；
- ▣ 若 $R \in BCNF$ ，则  $R \in 3NF$ ；若 $R \in 3NF$ ，则  $R$ 不一定 $\in BCNF$ ；

例6.5 关系模式  $C(Cno, Cname, Pcnno)$  ,属于几级范式?

$$Cno \rightarrow Cname, Cno \rightarrow Pcnno$$
$$C \in BCNF$$

例6.6 关系模式  $S(Sno, Sname, Sdept, Sage)$  ,假定Sname具有唯一性, 该模式属于几级范式?

$$Sno \rightarrow Sname, Sno \rightarrow Sdept, Sno \rightarrow Sage$$
$$Sname \rightarrow Sno, Sname \rightarrow Sdept, Sname \rightarrow Sage$$
$$S \in BCNF$$

例6.7 关系模式 SJP (S, J, P) 中, S是学生, J是课程, P表示名次。每个学生选修每门课程的成绩有一定的名次, 每门课程中每一个名次只有一个学生 (即没有并列名次), 该模式属于几级范式?

$$(S, J) \rightarrow P;$$

$$(J, P) \rightarrow S;$$

$$SJP \in \text{BCNF}$$

## 举例6.8



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。

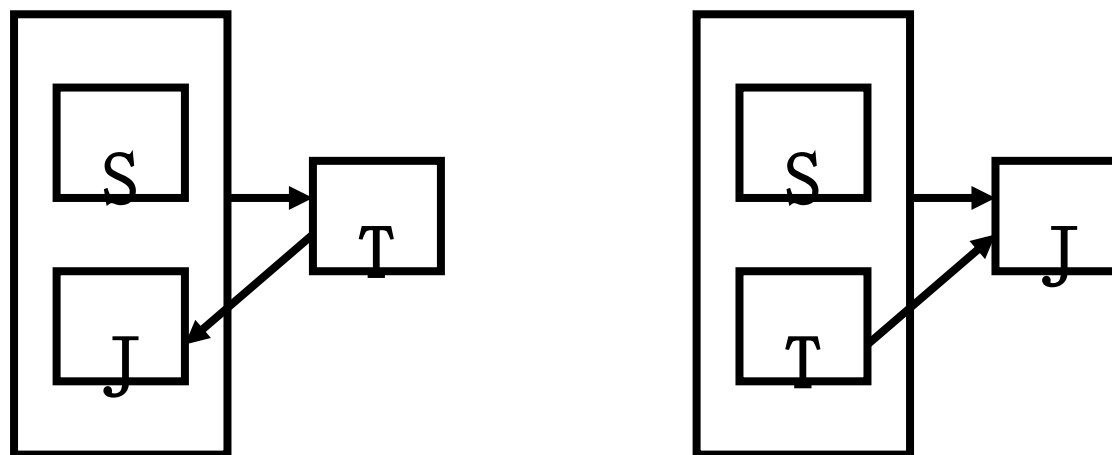
- 每一教师只教一门课。每门课由若干教师教, 某一学生选定某门课, 就确定了一个固定的教师。某个学生选修某个教师的课就确定了所选课的名称 :

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

## 举例6.8



函数依赖图：



STJ

## 举例6.8



关系STJ属于几级范式？ (  $T \rightarrow J$ ,  $(S, J) \rightarrow T$ ,  $(S, T) \rightarrow J$  )

1、 $STJ \in 3NF$  ?

- ◆  $(S, J)$  和  $(S, T)$  都可以作为候选码 ；
- ◆  **$S$ 、 $T$ 、 $J$ 都是主属性！**

2、 $STJ \in BCNF$  ?

- ◆  $T \rightarrow J$ ,  $T$ 是决定属性集,  $T$ 不是候选码

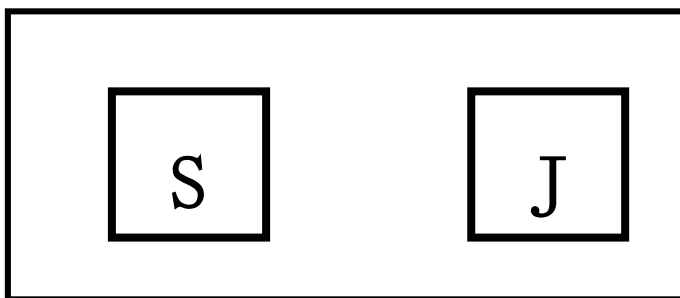


## 举例6.8

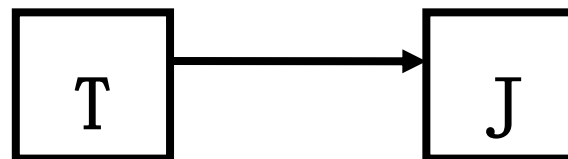


解决方法：将STJ分解为二个关系模式：

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



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



- 如果关系模式 $R \in \text{BCNF}$ ,

必定有 $R \in \text{3NF}$

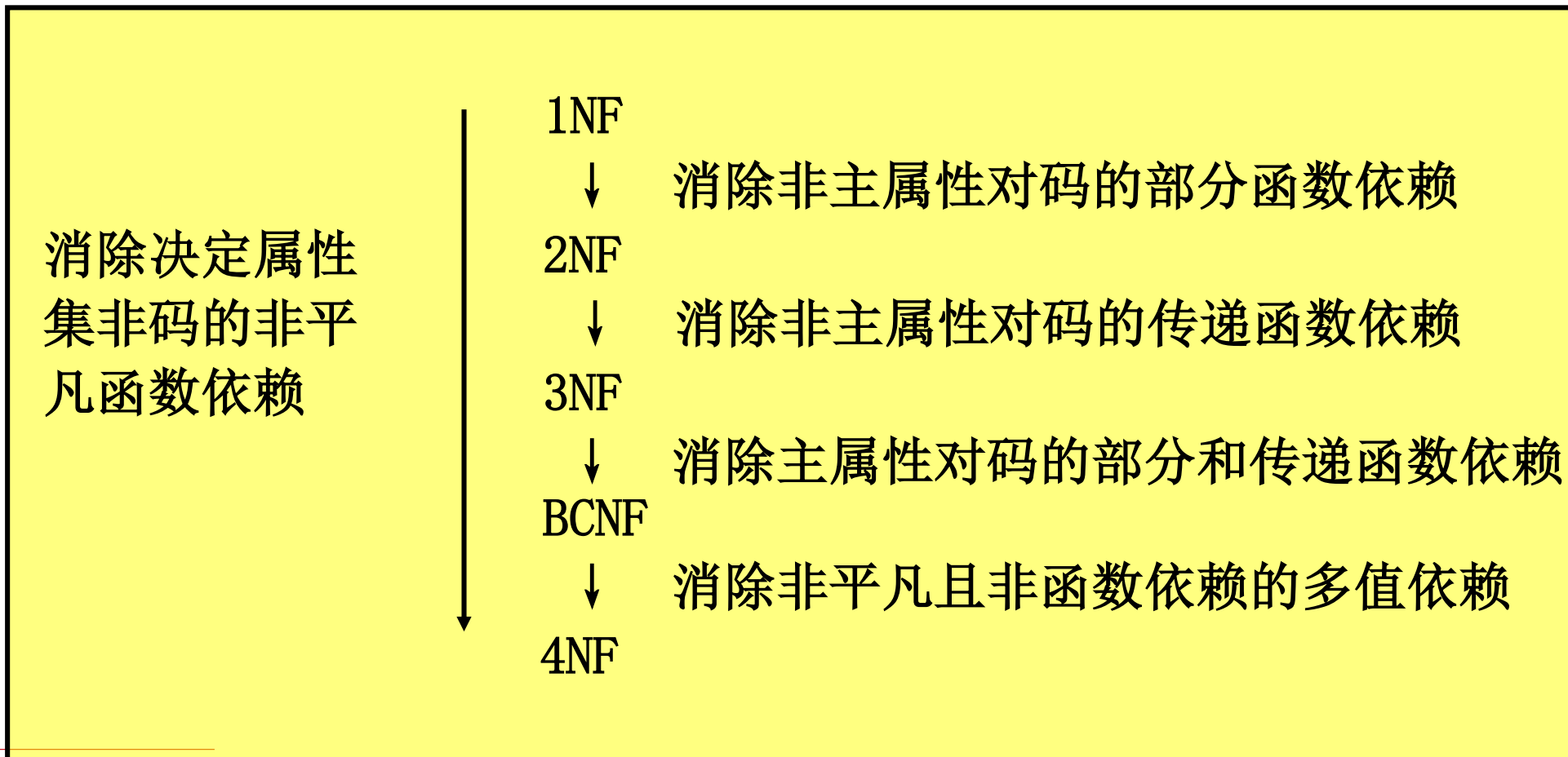
- 如果 $R \in \text{3NF}$ , 且 $R$ 只有一个候选码,

则 $R$ 必属于BCNF。

# 规范化小结



## 关系模式规范化的基本步骤





- ◆ 消除不合适的数据依赖，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；



- ❑ 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

- 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并不是唯一的；
- 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；



- ▣ 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

# 第三章 关系规范化基础



1

问题的提出

2

数据依赖

3

关系规范化

4

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, 则称

$F$  逻辑蕴含  $X \rightarrow Y$



- 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含的函数依赖



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

◆ A1. 自反律 (Reflexivity Rule) :

若 $Y \subseteq X \subseteq U$ , 则 $X \rightarrow Y$ 为 $F$ 所蕴含(平凡函数依赖)。

◆ A2. 增广律 (Augmentation Rule) :

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 Rule) :

若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 $F$ 所蕴含, 则 $X \rightarrow Z$ 为 $F$ 所蕴含。

# 定理 Armstrong推理规则是正确的



(1) 自反律: 若  $Y \subseteq X \subseteq U$ , 则  $X \rightarrow Y$  为  $F$  所蕴含

证: 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

# 定理 Armstrong推理规则是正确的



(2) 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：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  $t[Z] = s[Z]$ ；

由  $X \rightarrow Y$ ，于是有  $t[Y] = s[Y]$ ，

所以  $t[YZ] = s[YZ]$ ， $XZ \rightarrow YZ$  为  $F$  所蕴含。

增广律得证。

# 定理 Armstrong推理规则是正确的



(3) 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：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



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 (Union Rule) :

由 $X \rightarrow Y$ ,  $X \rightarrow Z$ , 有 $X \rightarrow YZ$ 。

◆ 伪传递规则 (Pseudo Transitivity Rule) :

由 $X \rightarrow Y$ ,  $WY \rightarrow Z$ , 有 $XW \rightarrow Z$ 。

◆ 分解规则 (Decomposition Rule) :

由 $X \rightarrow Y$  及  $Z \subseteq Y$ , 有 $X \rightarrow Z$ 。

# 最小依赖集



定义15 如果函数依赖集 $F$ 满足下列条件，则称 $F$ 为一个极小函数依赖集。亦称为**最小依赖集**或**最小覆盖**。

(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。



# 最小依赖集



例:对于关系模式 $S \langle U, F \rangle$ , 其中:

$$U = \{ \text{SNO}, \text{SDEPT}, \text{MN}, \text{CNAME}, \text{G} \},$$

$$F = \{ \text{SNO} \rightarrow \text{SDEPT}, \text{SDEPT} \rightarrow \text{MN}, \\ (\text{SNO}, \text{CNAME}) \rightarrow \text{G} \}$$

$$\text{设 } F' = \{ \text{SNO} \rightarrow \text{SDEPT}, \text{SNO} \rightarrow \text{MN}, \text{SDEPT} \rightarrow \text{MN}, \\ (\text{SNO}, \text{CNAME}) \rightarrow \text{G}, (\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \}$$

$F$ 是最小覆盖, 而 $F'$ 不是。

因为:  $F' - \{ \text{SNO} \rightarrow \text{MN} \}$  与  $F'$  等价

$F' - \{ (\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \}$  也与  $F'$  等价

$F' - \{ (\text{SNO}, \text{SDEPT}) \rightarrow \text{SDEPT} \} \cup \{ \text{SNO} \rightarrow \text{SDEPT} \}$  也与  $F'$  等价

# 极小化过程



[例]  $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C,$

$A \rightarrow C, C \rightarrow A\}$

$F_{m1} = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

$F_{m2} = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A\}$

$F_{m1}$ 、 $F_{m2}$ 都是 $F$ 的最小依赖集。

- $F$ 的最小依赖集 $F_m$ 不一定是唯一的，它与对各函数依赖 $FD_i$  及  $X \rightarrow A$  中 $X$  各属性的处置顺序有关

# 关系模式分解的标准



- ✓ **无损连接性：** 进行关系分解后得到的关系按照外码自然连接能够得到原来的关系。
- ✓ **函数依赖性：** 关系分解后每个关系的最小函数依赖集是原关系的最小函数依赖集的子集，并且所有子集的并等于原关系的最小函数依赖集。

## 三种模式分解的等价定义

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;

- 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



THANK YOU !

饮水思源 爱国荣校

---