

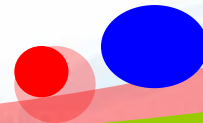
数据库原理与技术



An Introduction to Database System

第六章 关系数据理论

安徽农业大学信息与计算机学院



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

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
- 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



- 一、概念回顾
- 二、关系模式的形式化定义
- 三、什么是数据依赖
- 四、关系模式的简化定义
- 五、数据依赖对关系模式影响

一、概念回顾

❖ 关系

❖ 关系模式

❖ 关系数据库

❖ 关系数据库的模式



二、关系模式的形式化定义

关系模式由五部分组成，即它是一个五元组：

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

R: 关系名

U: 组成该关系的属性名集合

D: 属性组U中属性所来自的域

DOM: 属性向域的映象集合

F: 属性间数据的依赖关系集合



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

什么是数据依赖（续）

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）



3. 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

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>

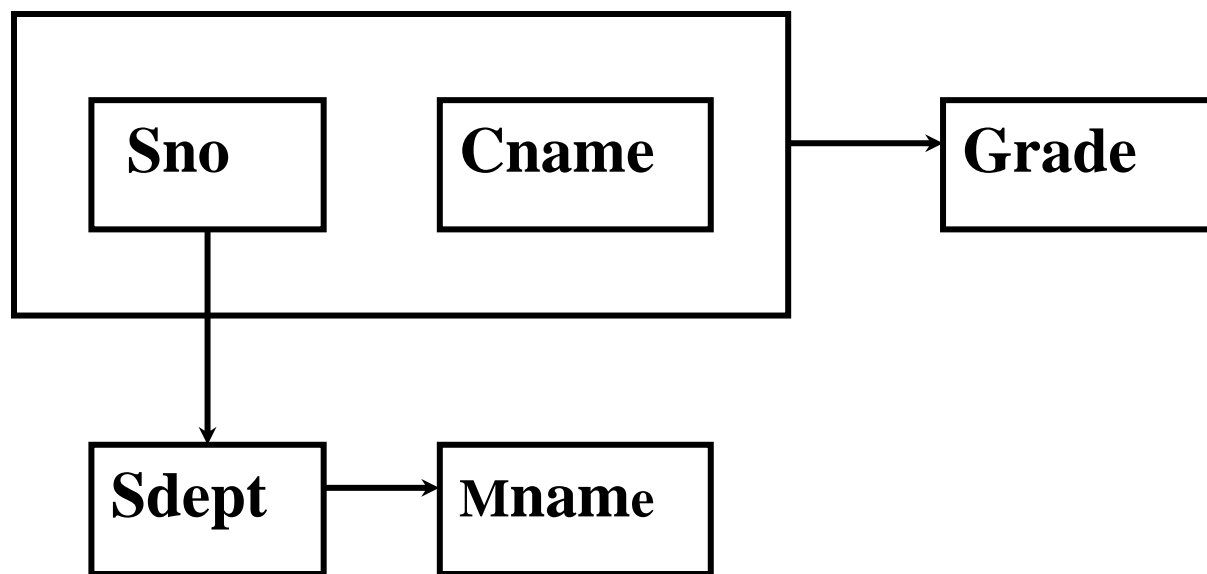
$U = \{ Sno, Sdept, Mname, Cname, Grade \}$



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)
4. 删除异常 (Deletion Anomalies)



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：

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

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

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





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 4NF

6.2.9 规范化小结



6.2.1 函数依赖



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

一、函数依赖

定义6.1 设 $R(U)$ 是一个属性集 U 上的关系模式， X 和 Y 是 U 的子集。

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

说明



1. 所有关系实例均要满足
2. 语义范畴的概念
3. 数据库设计者可以对现实世界作强制的规定

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 $Y \subsetneq X$ ，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 $X \nrightarrow Y$ 。



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 $X \xrightarrow{P} Y$ 。

完全函数依赖与部分函数依赖（续）



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

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

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

四、传递函数依赖

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 $Std(Sno, Sdept, Mname)$ 中, 有:

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

$Mname$ 传递函数依赖于 Sno



6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结



6.2.2 码

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



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）或非码属性（Non-key attribute）

❖ 全码

- 整个属性组是码，称为全码（All-key）

码（续）



[例2]

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

[例3]

关系模式 $R(P, W, A)$

P : 演奏者 W : 作品 A : 听众

一个演奏者可以演奏多个作品

某一作品可被多个演奏者演奏

听众可以欣赏不同演奏者的不同作品

码为 (P, W, A) ，即All-Key

外部码

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

- ❖ 如在 $SC(\underline{Sno}, \underline{Cno}, Grade)$ 中， Sno 不是码，但 Sno 是关系模式 $S(\underline{Sno}, Sdept, Sage)$ 的码，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 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.3 范式



- ❖ 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)

第二范式(2NF)

第三范式(3NF)

BC范式(BCNF)

第四范式(4NF)

第五范式(5NF)

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



6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.4 2NF



❖ 1NF的定义

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

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

2NF (续)

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



❖ 函数依赖包括：

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

$Sno \rightarrow Sdept$

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

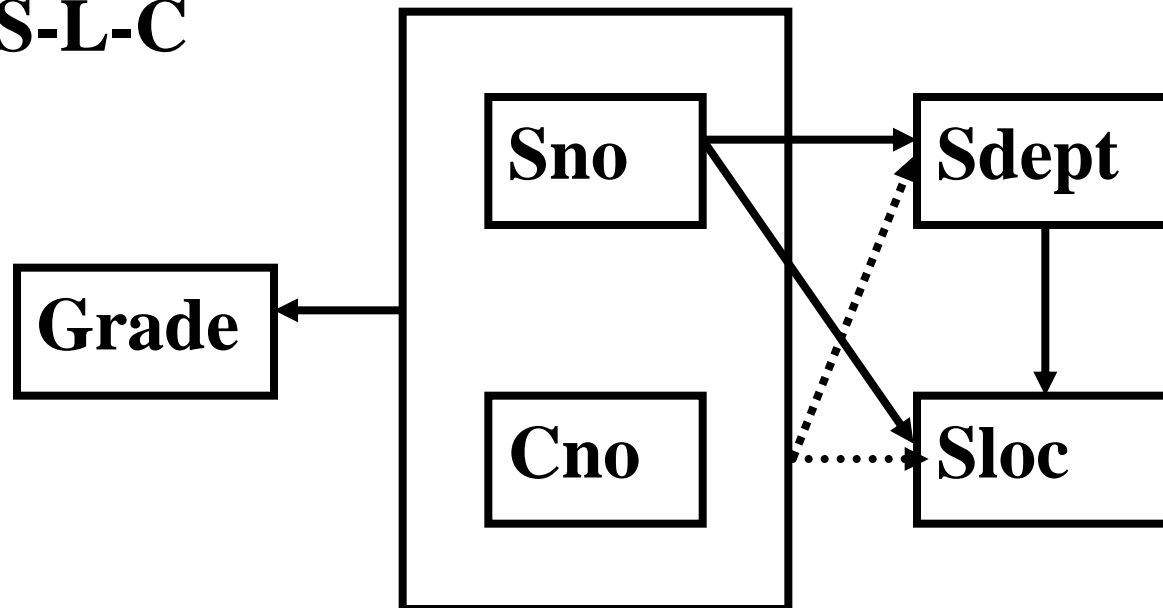
$Sno \rightarrow Sloc$

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

$Sdept \rightarrow Sloc$

2NF (续)

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）



- (1) 插入异常
- (2) 删除异常
- (3) 数据冗余度大
- (4) 修改复杂

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

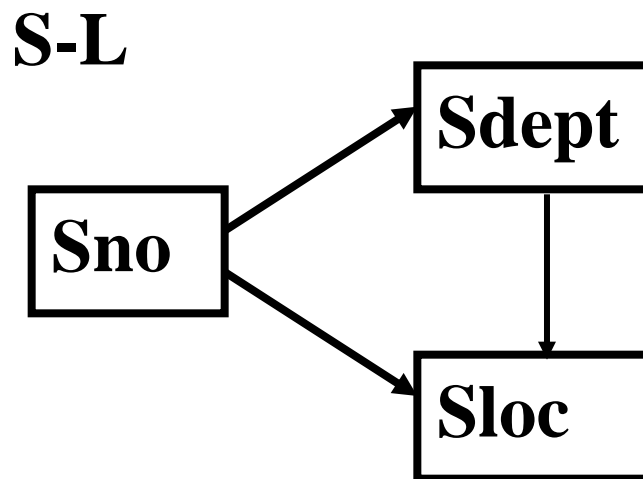
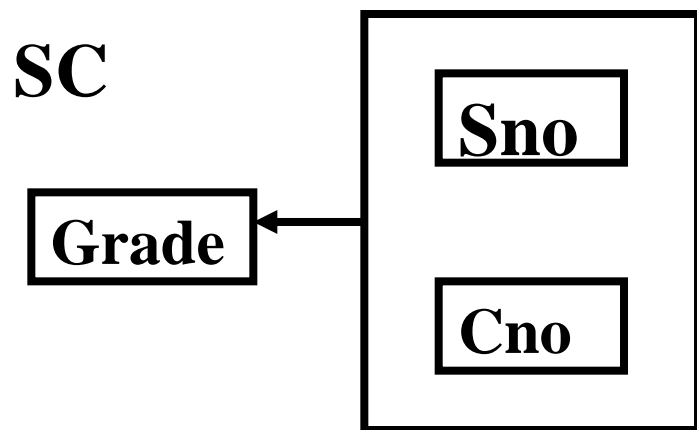
SC (Sno, Cno, Grade)

S-L (Sno, Sdept, Sloc)



2NF (续)

函数依赖图:



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

2NF (续)

❖ 2NF的定义

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

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

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

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

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



2NF（续）

- ❖ 采用投影分解法将一个1NF的关系分解为多个2NF的关系，可以在一定程度上减轻原1NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- ❖ 将一个1NF关系分解为多个2NF的关系，并不能完全消除关系模式中的各种异常情况和数据冗余。



6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结



6.2.5 3NF

❖ 3NF的定义



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

- 若 $R \in 3NF$, 则每一个非主属性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：

$Sno \rightarrow Sdept$

$Sdept \nrightarrow Sno$

$Sdept \rightarrow Sloc$

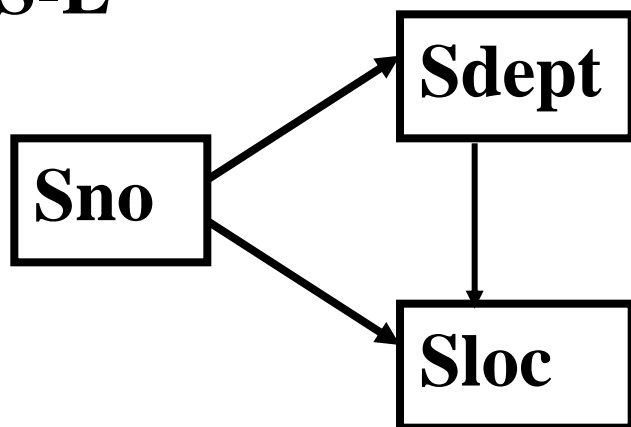
可得：

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

3NF (续)

函数依赖图:

S-L



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)

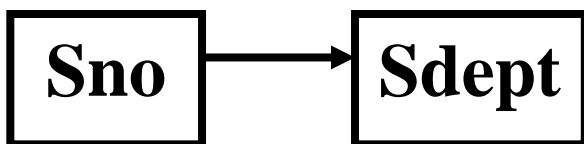
D-L (Sdept, Sloc)

S-D的码为Sno， D-L的码为Sdept。

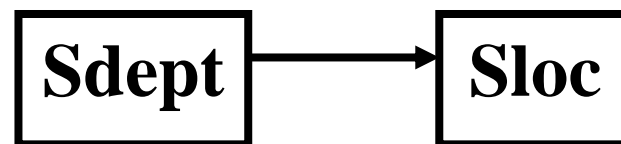
- 分解后的关系模式S-D与D-L中不再存在传递依赖

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。



6.2 规范化



6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.6 BC范式 (BCNF)

❖ 定义6.8 关系模式 $R\langle U, F \rangle \in 1NF$, 若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ 时 X 必含有码, 则 $R\langle U, F \rangle \in BCNF$ 。

❖ 等价于: 每一个决定属性因素都包含码

BCNF (续)



❖ 若 $R \in \text{BCNF}$

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

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

BCNF (续)



[例5] 关系模式C (Cno, Cname, Pcn)

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

[例6] 关系模式S (Sno, Sname, Sdept, Sage)

- 假定S有两个码Sno, Sname
- $S \in 3NF$ 。
- $S \in BCNF$

BCNF (续)

[例7] 关系模式SJP (S, J, P)



- 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)



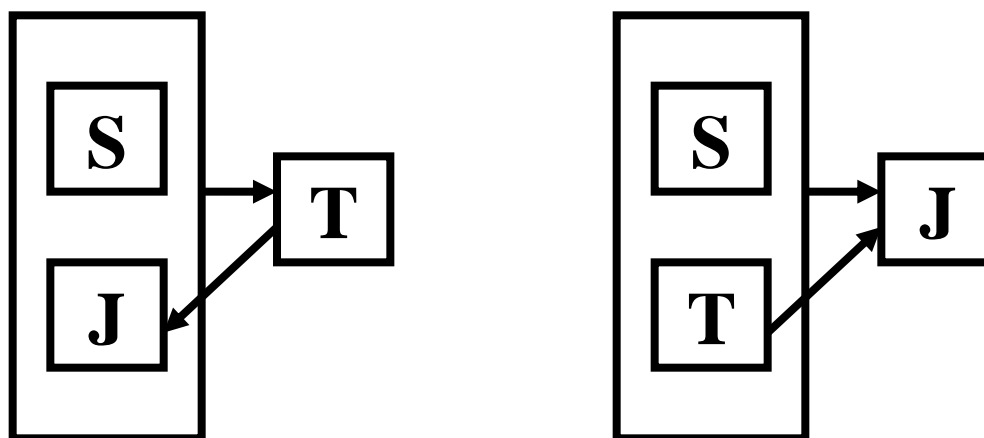
[例8]在关系模式STJ (S, T, J) 中, S表示学生, T表示教师, J表示课程。

- 函数依赖:

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

- (S, J)和(S, T)都是候选码

BCNF (续)



STJ中的函数依赖

BCNF (续)

❖ $STJ \in 3NF$

- 没有任何非主属性对码传递依赖或部分依赖

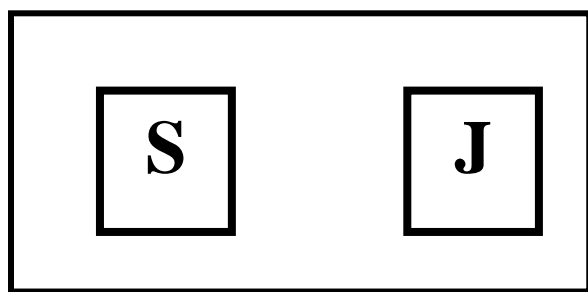
❖ $STJ \notin BCNF$

- T是决定因素，T不包含码

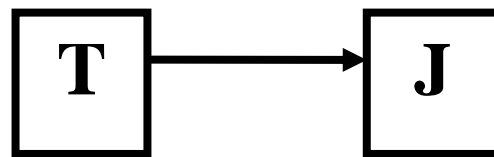


BCNF (续)

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



ST



TJ

没有任何属性对码的部分函数依赖和传递函数依赖



3NF与BCNF的关系



$$\diamond R \in \text{BCNF} \begin{array}{c} \xrightarrow{\text{充分}} \\ \xleftarrow{\text{不必要}} \end{array} R \in \text{3NF}$$

❖ 如果 $R \in \text{3NF}$ ，且 R 只有一个候选码

$$R \in \text{BCNF} \begin{array}{c} \xrightarrow{\text{充分}} \\ \xleftarrow{\text{必要}} \end{array} R \in \text{3NF}$$

6.2 规范化



6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.7 多值依赖

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



多值依赖（续）

❖ 非规范化关系



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

多值依赖（续）

❖ 用二维表表示Teaching



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

多值依赖（续）

❖ Teaching \in BCNF

❖ Teaching具有唯一候选码(C, T, B), 即全码



多值依赖（续）

Teaching模式中存在的问题

- (1) 数据冗余度大
- (2) 插入操作复杂
- (3) 删除操作复杂
- (4) 修改操作复杂

存在
多值依赖



多值依赖（续）

❖ 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)

多值依赖（续）

❖ 多值依赖的另一个等价的形式化的定义：



在 $R(U)$ 的任一关系 r 中，如果存在元组 t, s 使得 $t[X]=s[X]$ ，那么就必然存在元组 $w, v \in r$ ，（ w, v 可以与 s, t 相同），使得 $w[X]=v[X]=t[X]$ ，而 $w[Y]=t[Y]$ ， $w[Z]=s[Z]$ ， $v[Y]=s[Y]$ ， $v[Z]=t[Z]$ （即交换 s, t 元组的 Y 值所得的两个新元组必在 r 中），则 Y 多值依赖于 X ，记为 $X \twoheadrightarrow Y$ 。这里， X, Y 是 U 的子集， $Z=U-X-Y$ 。

多值依赖（续）



❖ 平凡多值依赖和非平凡的多值依赖

- 若 $X \twoheadrightarrow Y$ ，而 $Z = \varphi$ ，则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖
- 否则称 $X \twoheadrightarrow Y$ 为非平凡的多值依赖

多值依赖（续）



[例10] 关系模式WSC (W, S, C)

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

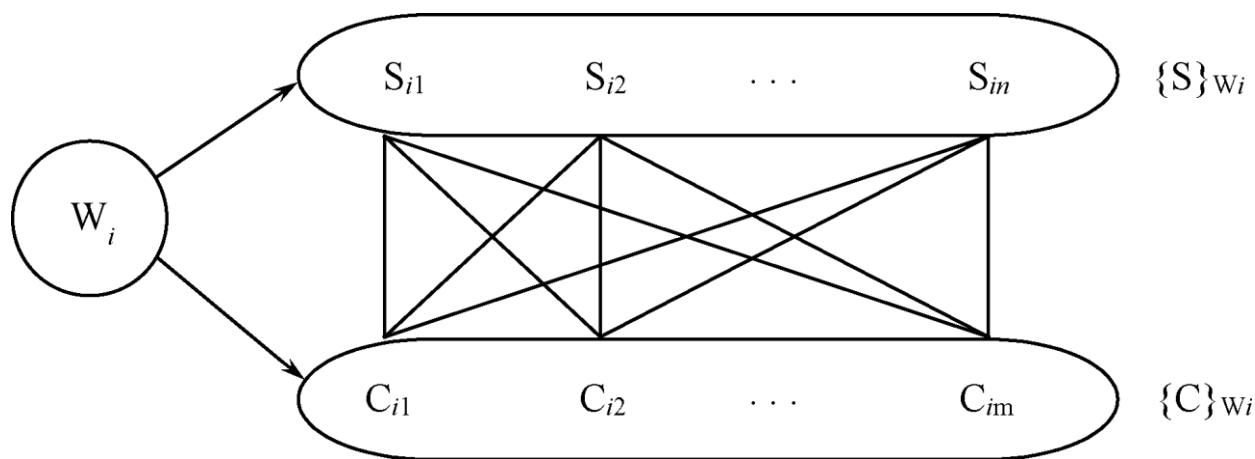
多值依赖（续）

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）

用下图表示这种对应



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

多值依赖的性质



(1) 多值依赖具有对称性

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

(2) 多值依赖具有传递性

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

(3) 函数依赖是多值依赖的特殊情况。

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

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

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

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

多值依赖与函数依赖的区别



(1) 多值依赖的有效性与属性集的范围有关

(2)

- 若函数依赖 $X \rightarrow Y$ 在 $R(U)$ 上成立，则对于任何 $Y' \subset Y$ 均有 $X \rightarrow Y'$ 成立
- 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

6.2 规范化



6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.8 4NF

❖ **定义6.10** 关系模式 $R\langle U, F \rangle \in 1NF$ ，如果对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y$ ($Y \not\subseteq X$)， X 都含有码，则 $R \in 4NF$ 。

❖ 如果 $R \in 4NF$ ， 则 $R \in BCNF$

- 不允许有非平凡且非函数依赖的多值依赖
- 允许的非平凡多值依赖是函数依赖



4NF (续)

例: Teaching(C,T,B) \notin 4NF



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

■ 用投影分解法把Teaching分解为如下两个关系模式:

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

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

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

6.2 规范化



6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

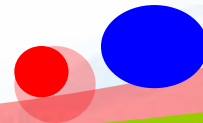
6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结



6.2.9 规范化小结



- ❖ 关系数据库的规范化理论是数据库逻辑设计的工具
- ❖ 目的：尽量消除插入、删除异常，修改复杂，数据冗余
- ❖ 基本思想：逐步消除数据依赖中不合适的部分
 - 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

消除决定属性
集非码的非平
凡函数依赖

1NF

↓ 消除非主属性对码的部分函数依赖

2NF

↓ 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



规范化小结（续）



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



第六章 关系数据理论



6.1 问题的提出

6.2 规范化

6.3 数据依赖的公理系统

***6.4 模式的分解**

6.5 小结

6.3 数据依赖的公理系统

❖ 逻辑蕴含

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



1. Armstrong公理系统

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



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

定理 6.1 Armstrong推理规则是正确的

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

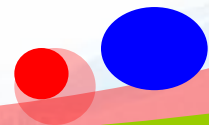


证: 设 $Y \subseteq X \subseteq U$

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 6.1 Armstrong推理规则是正确的（续）

(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 $R \subseteq U$, F 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 $t[YZ] = s[YZ]$, 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）

(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 $R \leq U, F$ 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条推理规则:



- **合并规则**: 由 $X \rightarrow Y$, $X \rightarrow Z$, 有 $X \rightarrow YZ$.
(A2, A3)
- **伪传递规则**: 由 $X \rightarrow Y$, $WY \rightarrow Z$, 有 $XW \rightarrow Z$.
(A2, A3)
- **分解规则**: 由 $X \rightarrow Y$ 及 $Z \subseteq Y$, 有 $X \rightarrow Z$.
(A1, A3)

导出规则

2. 根据合并规则和分解规则，可得引理6.1



引理6.1 $X \rightarrow A_1 A_2 \dots A_k$ 成立的充分必要条件是
 $X \rightarrow A_i$ 成立 ($i=1, 2, \dots, k$)

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



❖ Armstrong公理系统是有效的、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

3. 函数依赖闭包



定义6.13 设 F 为属性集 U 上的一组函数依赖, $X \subseteq U$
， $X_F^+ = \{ A/X \rightarrow A \text{ 能由 } F \text{ 根据Armstrong公理导出} \}$
， X_F^+ 称为**属性集 X 关于函数依赖集 F 的闭包**。

F的闭包

$$F=\{X\rightarrow Y, Y\rightarrow Z\}$$

$$F^+=\{$$

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

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



关于闭包的引理

❖ 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 $Y \subseteq X_F^+$

❖ 用途

将判定 $X \rightarrow Y$ 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



求闭包的算法

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^+



输入: X, F

输出: X_F^+

步骤:

(1) 令 $X^{(0)} = X, i=0$

(2) 求 B , 这里 $B = \{ A | (\exists V)(\exists W)(V \rightarrow W \in F \wedge V \subseteq X^{(i)} \wedge A \in W) \}$;

(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



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



[例1] 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

$\rightarrow (AB)_F^+=ABCDE。$

4. Armstrong公理系统的有效性与完备性

❖ 定理6.2 Armstrong公理系统是有效的、完备的

❖ 证明：

1. 有效性

可由定理6.1得证

2. 完备性

只需证明**逆否命题**：若函数依赖 $X \twoheadrightarrow Y$ 不能由 F 从Armstrong公理导出，那么它必然不为 F 所蕴含



Armstrong公理系统完备性证明



(1) 引理: 若 $V \rightarrow W$ 成立, 且 $V \subseteq X_F^+$, 则 $W \subseteq X_F^+$

(2) 构造一张二维表 r , 它由下列两个元组构成, 可以证明 r 必是 $R(U, F)$ 的一个关系, 即 F^+ 中的全部函数依赖在 r 上成立。

X_F^+	$U - X_F^+$
$\overbrace{11\dots\dots 1}$	$\overbrace{00\dots\dots 0}$
11.....1	11.....1

(3) 若 $X \rightarrow Y$ 不能由 F 从 Armstrong 公理导出, 则 Y 不是 X_F^+ 的子集。

5. 函数依赖集等价

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。

引理6.3 $F^+ = G^+$ 的充分必要条件是 $F \subseteq G^+$ ，和 $G \subseteq F^+$

证：必要性显然，只证充分性。

(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 $F^+ = G^+$ 。

6. 最小依赖集

定义6.15 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2] 关系模式 $S\langle U, F\rangle$, 其中:

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

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

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

F 是最小覆盖, 而 F' 不是。

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

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

7. 极小化过程

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（续）



(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 $\{ X \rightarrow A_j | j = 1, 2, \dots, k \}$ 来取代 $X \rightarrow Y$ 。

(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。

(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 。

极小化过程（续）



[例3] $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A\}$

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

❖ 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



第六章 关系数据理论



6.1 问题的提出

6.2 规范化

6.3 数据依赖的公理系统

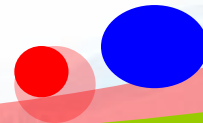
***6.4 模式的分解**

6.5 小结

6.4 模式的分解

- ❖ 把低一级的关系模式分解为若干个高一级的关系模式的方法不是唯一的
- ❖ 只有能够保证分解后的关系模式与原关系模式等价，分解方法才有意义





关系模式分解的标准



三种模式分解等价的定义：

1. 分解具有无损连接性
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

模式的分解（续）

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

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

$$U = \bigcup_{i=1}^n U_i, \text{ 且不存在 } U_i \subseteq U_j, F_i \text{ 为 } F \text{ 在 } U_i \text{ 上的投影}$$

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



模式的分解（续）

例：S-L (Sno, Sdept, Sloc)

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

$S-L \in 2NF$



分解方法可以有多种：

1. S-L分解为三个关系模式：
SN(Sno)
SD(Sdept)
SO(Sloc)
2. SL分解为下面二个关系模式：
NL(Sno, Sloc)
DL(Sdept, Sloc)
3. 将SL分解为下面二个关系模式：
ND(Sno, Sdept)
NL(Sno, Sloc)

具有无损连接性的模式分解

- ❖ 关系模式 $R\langle U, F \rangle$ 的一个分解 $\rho = \{ R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle \}$

若 R 与 R_1 、 R_2 、...、 R_n 自然连接的结果相等，则称关系模式 R 的这个分解 ρ 具有无损连接性（Lossless join）

- ❖ 具有无损连接性的分解保证不丢失信息
- ❖ 无损连接性不一定能解决插入异常、删除异常、修改复杂、数据冗余等问题





模式的分解（续）



第3种分解方法具有无损连接性

问题:这种分解方法没有保持原关系中的函数依赖

- SL中的函数依赖 $S_{dept} \rightarrow S_{loc}$ 没有投影到关系模式ND、NL上

保持函数依赖的模式分解

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

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

（其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$ ，且不存在 $U_i \subseteq U_j$ ， F_i 为 F 在 U_i 上的投影），若 F 所逻辑蕴含的函数依赖一定也由分解得到的某个关系模式中的函数依赖 F_i 所逻辑蕴含，则称关系模式 R 的这个分解是保持函数依赖的（Preserve dependency）





模式的分解（续）

4. 将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法就保持了函数依赖



模式的分解（续）



- ❖ 如果一个分解具有无损连接性，则它能够保证不丢失信息
- ❖ 如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况
- ❖ 分解具有无损连接性和分解保持函数依赖是两个互相独立的标准。具有无损连接性的分解不一定能够保持函数依赖；同样，保持函数依赖的分解也不一定具有无损连接性。

模式的分解（续）



第1种分解方法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种分解方法保持了函数依赖，但不具有无损连接性

第3种分解方法具有无损连接性，但未持函数依赖

第4种分解方法既具有无损连接性，又保持了函数依赖

分解算法



- ❖ 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



第六章 关系数据理论



6.1 问题的提出

6.2 规范化

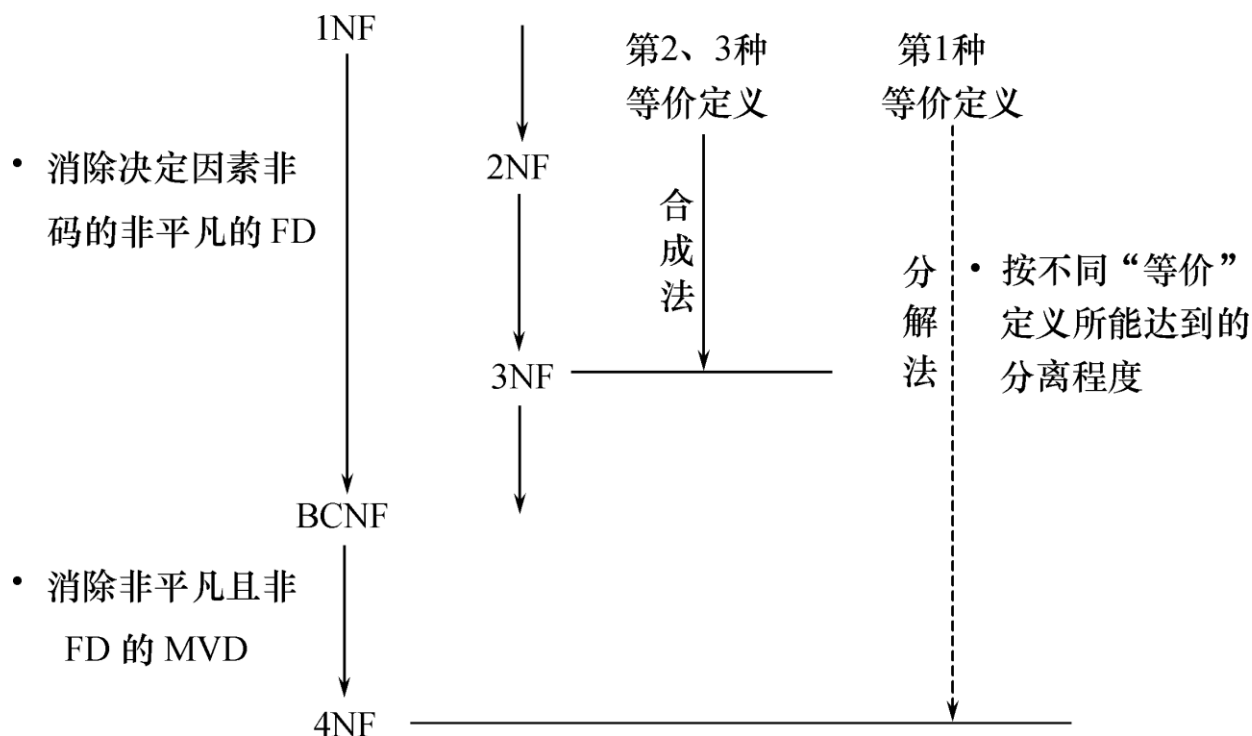
6.3 数据依赖的公理系统

***6.4 模式的分解**

6.5 小结

6.5 小结

关系模式的规范化，其基本思想：



小结(续)



- ❖ 若要求分解具有无损连接性，那么模式分解一定能够达到4NF
- ❖ 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到BCNF
- ❖ 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF

小结(续)

- ❖ 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

