

第七章 关系数据理论

第七章 关系数据理论

7.1 问题的提出

7.2 规范化

7.3 数据依赖的公理系统

第七章 关系数据理论

7.1 问题的提出

7.2 规范化

7.3 数据依赖的公理系统

7.1 问题的提出

[例] 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)

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. 数据冗余太大
2. 更新异常 (Update Anomalies)
3. 插入异常 (Insertion Anomalies)
4. 删除异常 (Deletion Anomalies)

7.1 问题的提出

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

7.1 问题的提出

原因：

- 把不同的概念放在了一起
 - 概念单一化
 - 采用“一事一地”的原则，让一个关系描述一个概念、一个实体或实体间的联系
- 模式中的某些数据依赖引起的

7.1 问题的提出-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

7.1 问题的提出-什么是数据依赖

■ 数据依赖

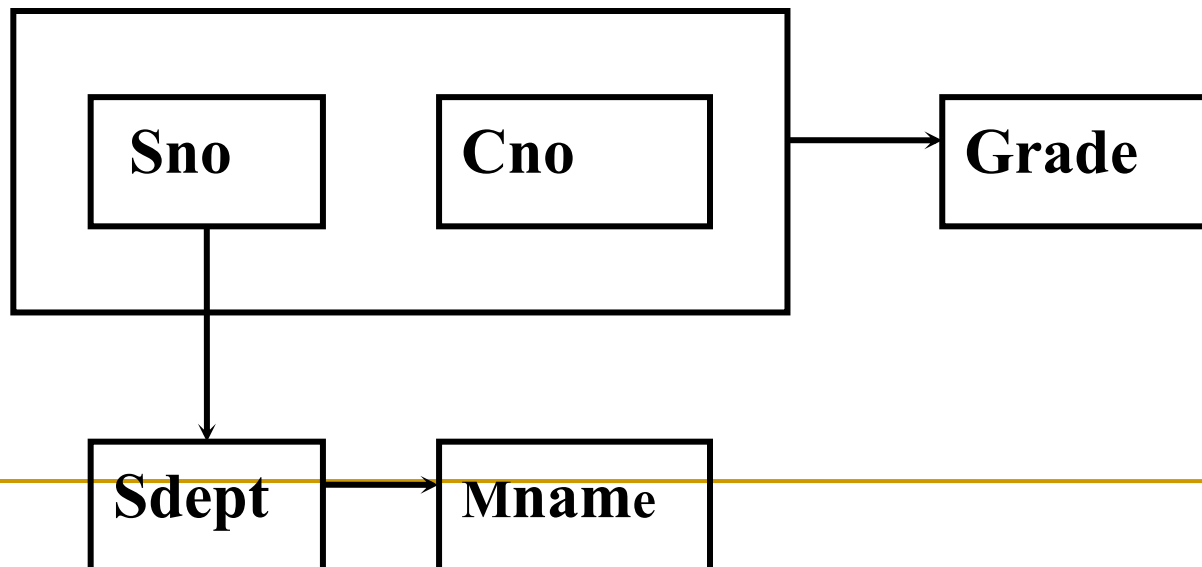
- 一个关系内部属性与属性之间的约束关系
- 现实世界属性间相互联系的抽象
- 数据内在的性质
- 语义的体现

7.1 问题的提出-什么是数据依赖

关系模式 Student

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

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



7.1 问题的提出-什么是数据依赖

■ 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

7.1 问题的提出-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：

$$R\langle U, F \rangle$$

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

7.1 问题的提出-数据依赖对关系模式的影响

- 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

7.1 问题的提出-数据依赖对关系模式的影响

■ 分解关系模式

□ 把这个单一模式分成3个关系模式：

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

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

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

7.1 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*

第七章 关系数据理论

7.1 问题的提出

7.2 规范化

7.3 数据依赖的公理系统

7.2 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。

7.2 规范化-函数依赖

■ 函数依赖

- 定义7.1 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 $X \rightarrow Y$ 。

7.2 规范化-函数依赖

- **学生表**

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

- **课程表**

Course(Cno,Cname,Cpno,Ccredit)

- **学生选课**

SC(Sno,Cno,Grade)

7.2 规范化-函数依赖

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

7.2 规范化-函数依赖

■ 思考

- 码是否一定函数决定非码属性？非码属性间是否存在函数依赖
- $Sno \rightarrow Sdept$ 成立， $(Sno, Cno) \rightarrow Sdept$ 是否也成立？

7.2 规范化-函数依赖

■ 函数依赖

- 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 $X \nrightarrow Y$ 。

7.2 规范化-函数依赖

■ 函数依赖

- 例，在关系S(SNO,SNAME,SEX,AGE,DEPT)中，假设无人重名

函数依赖有： $SNO \rightarrow SEX$ ， $SNO \rightarrow AGE$ ， $SNO \rightarrow dept$ ， $SNO \longleftrightarrow SNAME$

7.2 规范化-函数依赖

■ 函数依赖

- 在一个关系模式 $R(U)$ 中，对于 U 的子集 X 和 Y
 - 若 X 和 Y 有1 : 1关系，则 $X \rightarrow Y, Y \rightarrow X$ ，记作 $X \leftrightarrow Y$
 - 若 X 和 Y 有1 : m 关系，则记作 $Y \rightarrow X$ ，但 $X \nrightarrow Y$ 。
 - 若 X 和 Y 有 n : m 关系，则 X 和 Y 不存在任何函数依赖

7.2 规范化-函数依赖

■ 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 $Y \subseteq X$ ，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*

7.2 规范化-函数依赖

■ 平凡函数依赖与非平凡函数依赖

□ 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： $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$

7.2 规范化-函数依赖

■ 完全函数依赖与部分函数依赖

□ 定义7.2 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 $X \xrightarrow{P} Y$ 。

7.2 规范化-函数依赖

■ 完全函数依赖与部分函数依赖

- 由定义可知，若 X 为单属性，由于 X 不存在任何真子集，若 $X \rightarrow Y$ ，则 $X \xrightarrow{E} Y$ 。

7.2 规范化-函数依赖

■ 完全函数依赖与部分函数依赖

[例] student (*Sno*, *Sdept*, *Mname*, *Cno*, *Grade*) 中

$(Sno, Cno) \rightarrow Grade$

是完全函数依赖 ,

$(Sno, Cno) \rightarrow Sdept$

是部分函数依赖

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

7.2 规范化-函数依赖

■ 完全函数依赖与部分函数依赖

[例] 在关系SC1(Sno, Cno, Grade, Credit)中

$(Sno, Cno) \rightarrow Grade$

是完全函数依赖 ,

$Cno \rightarrow Credit$

$(Sno, Cno) \rightarrow Credit$

是部分函数依赖

7.2 规范化-函数依赖

■ 传递函数依赖

- **定义7.3** 在 $R(U)$ 中，如果 $X \rightarrow Y$ ， $(Y \not\subseteq X)$ ， $Y \nrightarrow X$ ， $Y \rightarrow Z$ ， $Z \not\subseteq Y$ 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 $X \longleftrightarrow Y$ ，则 **Z 直接依赖于 X** 。

7.2 规范化-函数依赖

■ 传递函数依赖

例: 在关系student (*Sno*, *Sdept*, *Mname*, *Cno*, *Grade*) 中, 有:

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

*Mname*传递函数依赖于*Sno*

7.2 规范化-函数依赖

■ 传递函数依赖

例: 在关系s1 (*Sno*, *Sname*, *Dno*, *Dname*, *location*) 中 , 有 :

$Sno \rightarrow Dno$, $Dno \nrightarrow Sno$, $Dno \rightarrow location$

*location*传递函数依赖于*Sno*

7.2 规范化-码

- 定义7.4 设K为R<U,F>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）。
- 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）

7.2 规范化-码

- 全码

- 整个属性组是码，称为 **全码** (All-key)

- 定义7.5

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

7.2 规范化-范式

- 关系模型要求关系必须是**规范化的**（normalization），即要求关系必须满足一定的规范条件（满足一定的要求）。
- 规范化的关系称为**范式**（Normal Form）
- 满足不同程度要求的为不同范式

7.2 规范化-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)

7.2 规范化-范式

- 1971-1972年,E.F.Codd提出了1NF,2NF,3NF
- 1974年 , Codd和Boyce又共同提出了一个新范式 , 即BCNF
- 1976年 , Fagin又提出了4NF
- 后来又有人提出了5NF

7.2 规范化-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**

7.2 规范化- 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

7.2 规范化- 2NF

[例]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

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

S-L-C不是一个好的关系模式

7.2 规范化- 2NF

[例] 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：

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

$Sno \rightarrow Sdept$

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

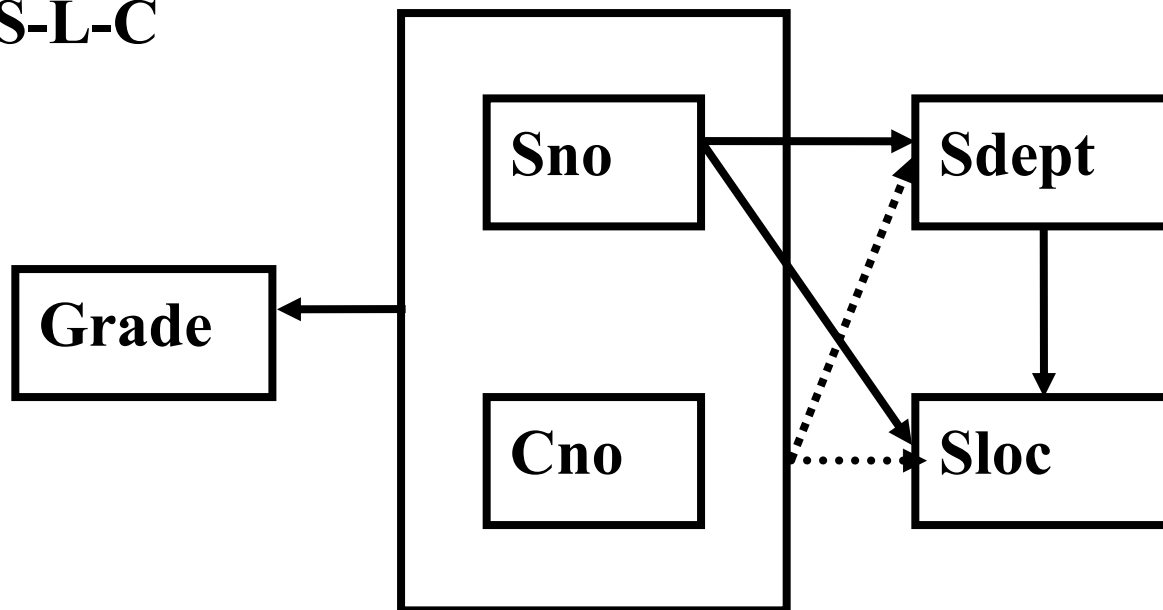
$Sdept \rightarrow Sloc$

$Sno \rightarrow Sloc$

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

7.2 规范化- 2NF

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)

7.2 规范化- 2NF

- **原因**

Sdept、 Sloc部分函数依赖于码。

- **解决方法**

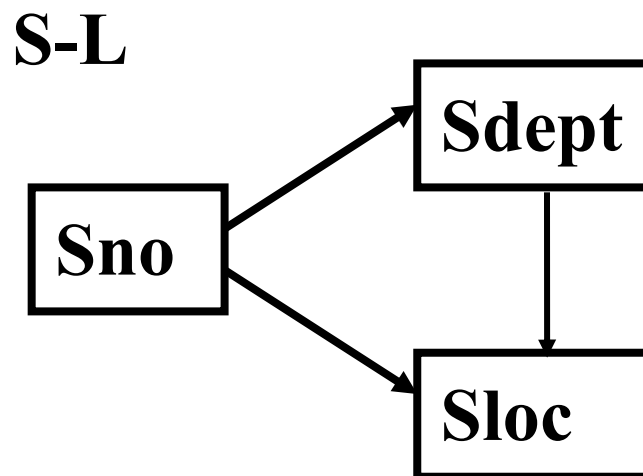
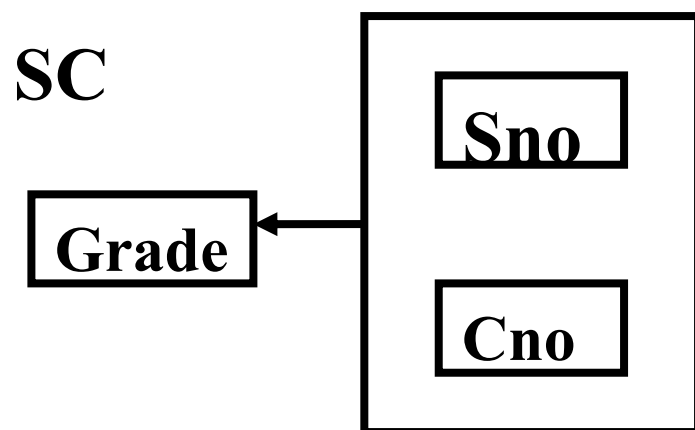
S-L-C分解为两个关系模式，以消除这些部分函数依赖

SC (Sno , Cno , Grade)

S-L (Sno , Sdept , Sloc)

7.2 规范化- 2NF

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

7.2 规范化- 2NF

■ 2NF的定义

定义7.6 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) $\in 2NF$

7.2 规范化- 2NF

- 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。

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

7.2 规范化- 3NF

■ 3NF的定义

定义7.7 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 $X \rightarrow Y$, $Y \rightarrow Z$ 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。

7.2 规范化- 3NF

例：2NF关系模式S-L(Sno, Sdept, Sloc)中

□ 函数依赖：

$Sno \rightarrow Sdept$

$Sdept \twoheadrightarrow Sno$

$Sdept \rightarrow Sloc$

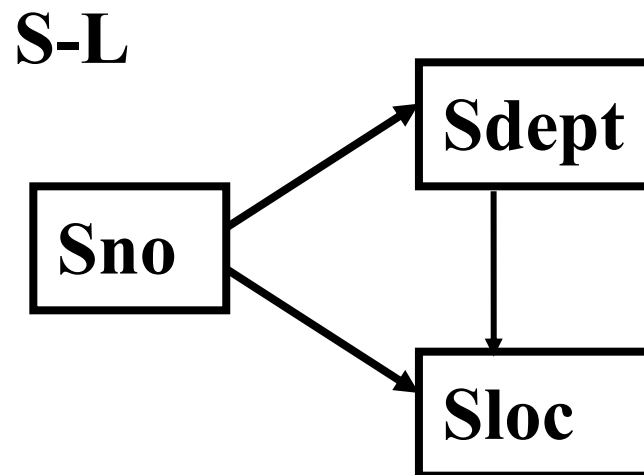
可得：

$Sno \xrightarrow{\text{传递}} Sloc$ ，即S-L中存在非主属性对码的传递

函数依赖， $S-L \notin 3NF$

7.2 规范化- 3NF

函数依赖图：



7.2 规范化- 3NF

■ 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)

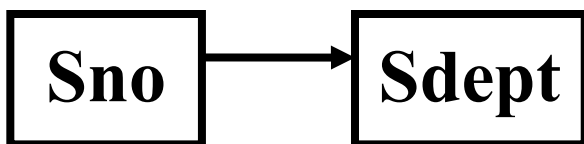
D-L (Sdept , Sloc)

S-D的码为Sno，D-L的码为Sdept。

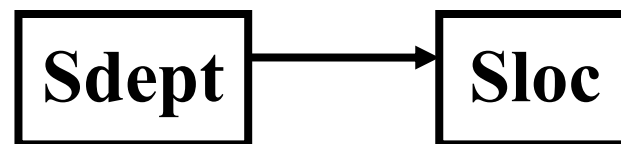
■ **分解后的关系模式S-D与D-L中不再存在传递依赖**

7.2 规范化- 3NF

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

7.2 规范化- 3NF

- 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。

7.2 规范化- 3NF

- 例如SC2(SNO,SNAME,CNO,Grade) , 设姓名是唯一的
 - 候选码(SNO,CNO)和(SNAME,CNO)
 - 非主属性Grade
 - $SC2 \in 3NF$
 - 如果学生退选课, 元组被删除也就失去了学号和姓名的对应关系
 - 由于学生选课很多, 姓名也将重复存储

7.2 规范化- 3NF

- 例如仓库保管WPE(W#,P#,E#,QNT) , 其中各属性分别表示仓库号、零件号、职工号、数量

W#	P#	E#	QNT
w1	p1	e1	20
w1	p2	e3	15
w1	p3	e2	18
w1	p4	e1	10
w1	p5	e3	20
w1	p7	e3	18
w1	p6	e2	15
w2	p4	e4	12
w2	p1	e5	20
w2	p3	e4	18
w2	p6	e5	15
w3	p2	e6	20
w3	p3	e6	15
w3	p4	e7	18

它所反映的语义是：

一个仓库有多个职工

一个职工仅在一个仓库工作

每个仓库里一种器件由专人管理

存在函数依赖：

$E\# \rightarrow W\#$

$(W\#, P\#) \rightarrow E\#$

$(W\#, P\#) \rightarrow QNT$

$(E\#, P\#) \rightarrow QNT$

7.2 规范化- 3NF

■存在函数依赖：

$E\# \rightarrow W\#$, $(W\#,P\#) \rightarrow E\#$

$(W\#,P\#) \rightarrow QNT$, $(E\#,P\#) \rightarrow QNT$

候选关键字是： $(W\#,P\#)$, $(E\#,P\#)$

只有一个非主属性QNT，它对任何候选关键字都是完全依赖，并且是直接依赖，因此属于3NF

7.2 规范化- 3NF

- 例如仓库保管WPE(W#,P#,E#,QNT) , 其中各属性分别表示仓库号、零件号、职工号、数量

候选关键字是：(W#,P#) , (E#,P#)

如果一个新职工被分配到仓库工作，但暂时处于实习阶段，没有独立承担任务，由于缺少关键字的一部分p#，而无法插入到关系中，要消除这些异常需要提出更高的要求。

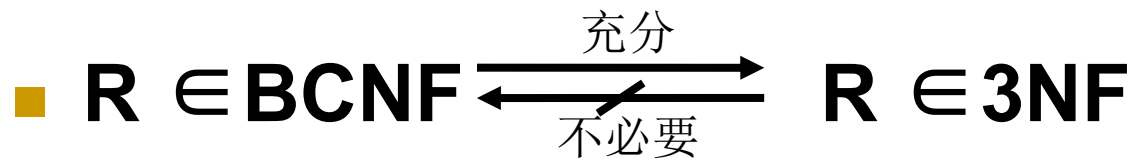
7.2 规范化- BCNF

- **定义7.8** 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**

7.2 规范化- BCNF

- 若 $R \in BCNF$
 - 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
- $R \in BCNF$ ，排除了任何属性对码的传递依赖与部分依赖

7.2 规范化- BCNF



7.2 规范化- BCNF

- 仓库保管WPE(W#,P#,E#,QNT) , 其中各属性分别表示仓库号、零件号、职工号、数量。存在函数依赖：

$E\# \rightarrow W\#$, $(W\#,P\#) \rightarrow E\#$

$(W\#,P\#) \rightarrow QNT$, $(E\#,P\#) \rightarrow QNT$

候选关键字是： $(W\#,P\#)$, $(E\#,P\#)$

决定因素E#不包含关键字

这就造成了主属性W#对候选关键字(E#,P#)的部分依赖

7.2 规范化- BCNF

[例] 关系模式C (Cno , Cname , Pcnno)

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

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

假定S有两个码Sno , Sname

$S \in 3NF$

$S \in BCNF$

7.2 规范化- BCNF

[例] 关系模式SJP (S , J , P) 中 , S表示学生 , J表示课程 , P表示名次 。 (假设不存在并列名次)

- **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**

7.2 规范化- BCNF

[例]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 :**

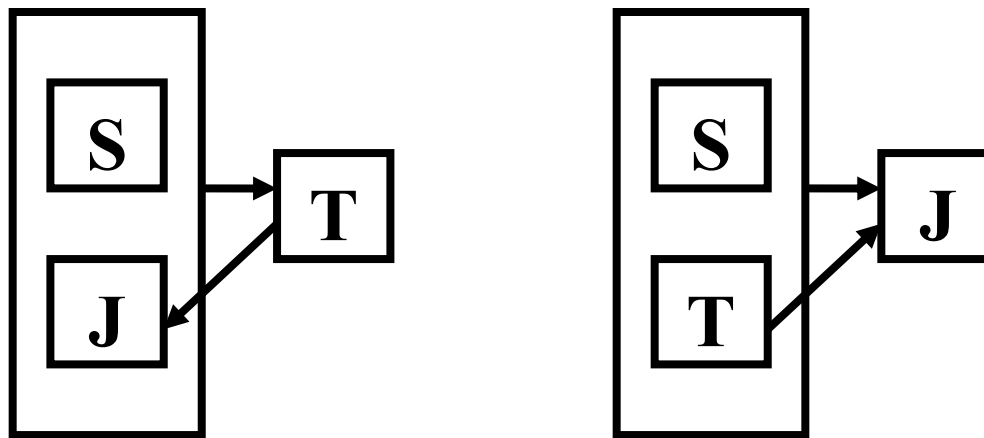
$T \rightarrow J$

$(S, J) \rightarrow T$

$(S, T) \rightarrow J$

(S, J) 和 (S, T) 都是候选码

7.2 规范化- BCNF



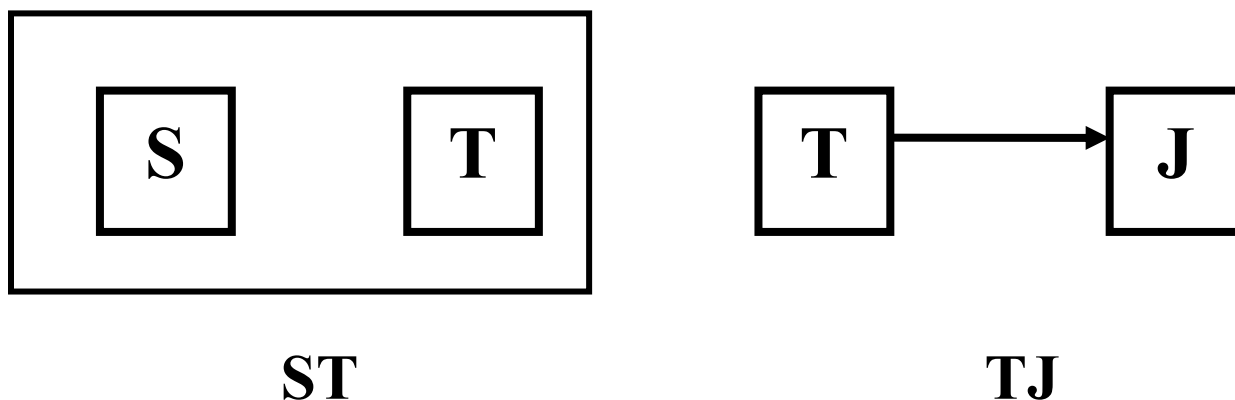
STJ中的函数依赖

7.2 规范化- BCNF

- $STJ \in 3NF$
 - 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

7.2 规范化- BCNF

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



没有任何属性对码的部分函数依赖和传递函数依赖

7.2 规范化- BCNF

- 3NF和BCNF是在函数依赖的条件下的关系模式规范化程度的测度。
- 一个模式中的关系模式如果都属于BCNF，那么在函数依赖范畴内，它已经实现了彻底的分离，已消除了插入和删除异常。
- 3NF的“不彻底”性表现在可能存在主属性对码的部分依赖和传递依赖。
- 在信息系统的设计中，普遍采用的是“基于3NF的系统设计”方法。

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

7.2 规范化- 多值依赖

[例] 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。

7.2 规范化- 多值依赖

❖ 非规范化关系

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

7.2 规范化- 多值依赖

❖ 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	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
...

7.2 规范化- 多值依赖

- Teaching具有唯一候选码(C , T , B) , 即全码
- Teaching \in BCNF
- 对数据的增删改很不方便 , 数据冗余也十分明显

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
...

Teaching模式中存在的问题

(1) 数据冗余度大

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
...

Teaching模式中存在的问题

(2) 插入操作复杂

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
...

Teaching模式中存在的问题

(3) 删除操作复杂

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
...

Teaching模式中存在的问题

(4) 修改操作复杂

7.2 规范化- 多值依赖

■ 定义7.9

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

7.2 规范化- 多值依赖

例 Teaching (C, T, B)

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
...

7.2 规范化- 多值依赖

■ 平凡多值依赖和非平凡的多值依赖

- 若 $X \twoheadrightarrow Y$ ，而 $Z = \phi$ ，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**

7.2 规范化- 多值依赖

例：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

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

7.2 规范化- 多值依赖

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) 函数依赖是多值依赖的特殊情况。

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

(3) 多值依赖具有传递性

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

7.2 规范化- 多值依赖

多值依赖的性质

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

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

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

7.2 规范化- 多值依赖

- **多值依赖与函数依赖的区别**
 - **多值依赖的有效性与属性集的范围有关**
 - **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**

7.2 规范化- 4NF

- 定义7.10 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 $R \in 4NF$ 。
- 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

7.2 规范化- 4NF

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：

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

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

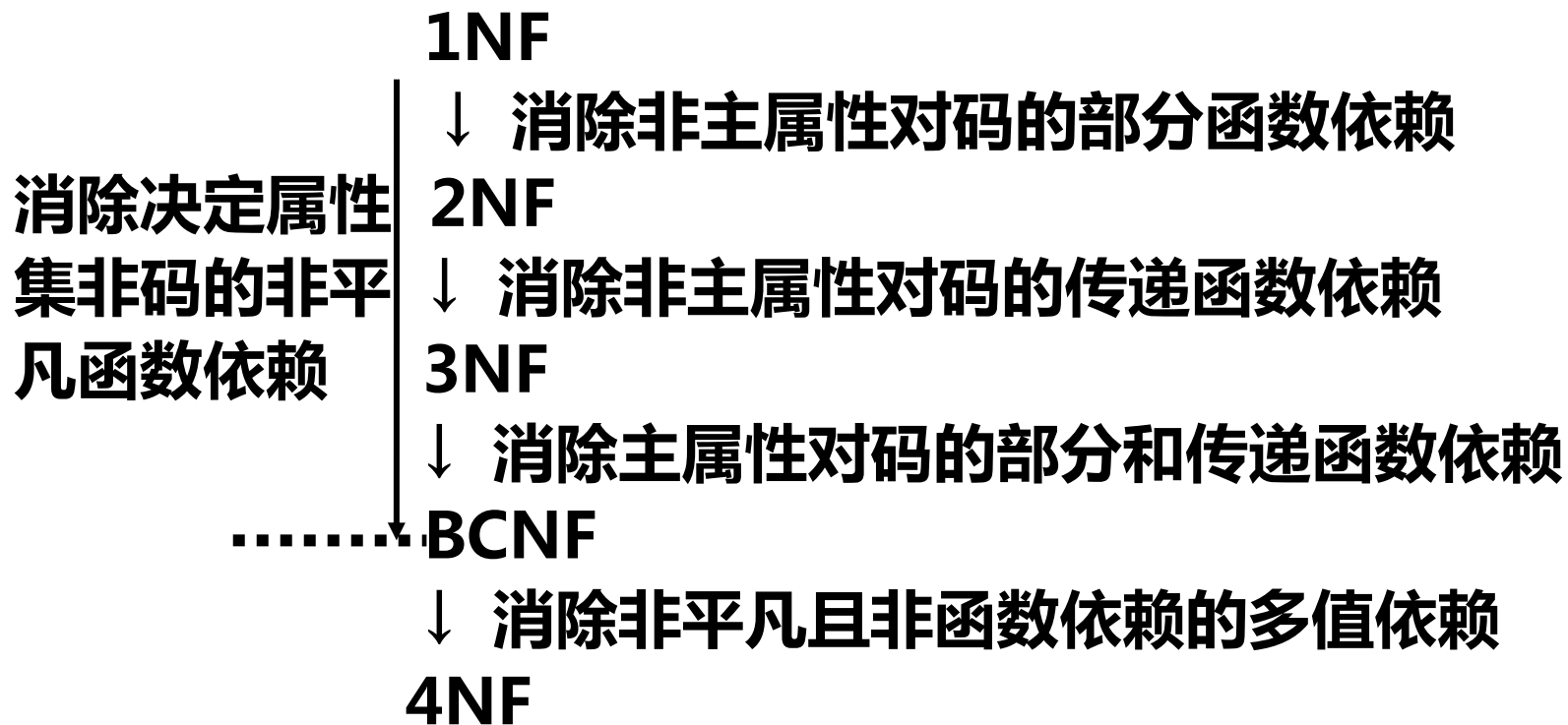
$C \twoheadrightarrow T$ ， $C \twoheadrightarrow B$ 是平凡多值依赖

7.2 规范化-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

7.2 规范化-规范化小结

■ 关系模式规范化的基本步骤



7.2 规范化-规范化小结

- 一个关系只要其分量都是不可分的数据项，他就是规范化的关系，但这只是最基本的规范化。规范化程度可以有6个不同的级别，即6个范式。
- 一个低一级范式的关系模式通过 **模式分解** 可以转换为若干高一级范式的关系模式集合，这个过程叫关系模式的 **规范化**。

7.2 规范化-规范化小结

- **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**

7.2 规范化-规范化小结

- **关系模式规范化时应遵循以下原则：**
 - **关系模式进行无损连接分解。避免信息丢失。**
简单地说就是模式R分解为 R_1, R_2, \dots, R_n 后，将 R_1, R_2, \dots, R_n 自然连接还应该等于R。

7.2 规范化-规范化小结

A	B	C
a1	b1	c1
a2	b2	c1
a3	b2	c1

R1

A	C
a1	c1
a2	c1
a3	c1

R2

B	C
b1	c1
b2	c1

$R1 \bowtie R2$

A	B	C
a1	b1	c1
a1	b2	c1
a2	b1	c1
a2	b2	c1
a3	b1	c1
a3	b2	c1

7.2 规范化-规范化小结

例 : S-L (Sno , Sdept , Mname)

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

$S-L \in 2NF$

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(Mname)

7.2 规范化-规范化小结

例 : S-L (Sno , Sdept , Mname)

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

$S-L \in 2NF$

分解方法可以有多种 :

2. 将SL分解为下面二个关系模式 :

ND(Sno, Sdept)

NL(Sno, Mname)

7.2 规范化-规范化小结

- **关系模式规范化时应遵循以下原则：**
 - **保持原来模型的函数依赖。因为这些函数依赖关系是数据模型反映的客观事物的固有属性，一般是不能舍弃的。简单地说就是模式R分解为 R_1, R_2, \dots, R_n 后，函数依赖集合F也被对应分解为 F_1, F_2, \dots, F_n ，应满足F与各 F_i （ $i=1, 2, \dots, n$ ）的并集等价。**

7.2 规范化-规范化小结

例：S-L (Sno , Sdept , Mname)

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

$S-L \in 2NF$

分解方法可以有多种：

3. SL分解为下面二个关系模式：

NL(Sno, Sdept)

DL(Sdept, Mname)

7.2 规范化-规范化小结

- 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

7.2 规范化-规范化小结

- **关系模式规范化时应遵循以下原则：**
 - **合理选择规范化程度。考虑到存取效率，低级模式造成的冗余度很大，既浪费了存储空间，又影响了数据的一致性。因此希望一个子模式的属性越少越好，即取高级范式；若考虑到查询效率，低级范式又比高级范式好，此时连接运算的代价较小，这是一对矛盾，应根据情况合理选择规范化程序。**

7.2 规范化-规范化小结

例：在关系模式

学生成绩单(学号,英语,数学,语文,平均成绩)

中存在下列函数依赖：

学号→英语

学号→数学

学号→语文

学号→平均成绩

(英语, 数学, 语文)→平均成绩

7.2 规范化-规范化小结

显然有：

学号 \rightarrow (英语,数学,语文)

因此该关系模式中存在传递函数依赖，是2NF关系

虽然平均成绩可以由其他属性推算出来，但如果应用中需要经常查询学生的平均成绩，为提高效率，仍然可保留该冗余数据，对关系模式不再做进一步分解

第七章 关系数据理论

7.1 问题的提出

7.2 规范化

7.3 数据依赖的公理系统

7.3 数据依赖的公理系统

■ 逻辑蕴含

定义7.11 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 $X \rightarrow Y$

7.3 数据依赖的公理系统

■ 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。

7.3 数据依赖的公理系统

■ 导出规则

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**：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 $X \rightarrow Z$ 。

7.3 数据依赖的公理系统

■ 导出规则

根据合并规则和分解规则，可得引理7.1

引理7.1 $X \rightarrow A_1 A_2 \dots A_k$ 成立的充分必要条件是
 $X \rightarrow A_i$ 成立 ($i=1, 2, \dots, k$)

7.3 数据依赖的公理系统

■ 函数依赖闭包

定义7.12 关系模式 $R\langle U, F\rangle$ 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^+ 。

7.3 数据依赖的公理系统

例: 已知 $R\langle U, F \rangle$, 其中 $U = \{A, B, C\}$, $F = \{A \rightarrow B, B \rightarrow C\}$, 求关系模式 R 上的函数依赖集 F 的闭包 F^+ 。

解：

$F^+ = \{ \Phi \rightarrow \Phi, A \rightarrow \Phi, A \rightarrow A, \dots, AB \rightarrow A, \dots \quad //A1$

$A \rightarrow B, A \rightarrow AB, AB \rightarrow B, \dots, ABC \rightarrow BC, \dots \quad //A2$

$B \rightarrow C, AB \rightarrow AC, \dots \quad //A2$

$A \rightarrow C \quad //A3$

注：本题 F^+ 共计43个不重复的FD。

7.3 数据依赖的公理系统

- **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

7.3 数据依赖的公理系统

■ 函数依赖闭包

定义7.13 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

7.3 数据依赖的公理系统

■ 关于闭包的引理

引理7.2 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

7.3 数据依赖的公理系统

■ 求闭包的算法

算法7.1 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^+

7.3 数据依赖的公理系统

步骤:

(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) 若相等或 $X^{(i+1)} = U$, 则 $X^{(i+1)}$ 就是 X_F^+ , 算法终止。

(6) 若否, 则 $i=i+1$, 返回第 (2) 步。

[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

$(AB)_F^+ = ABCDE$ 。

7.3 数据依赖的公理系统

■ 函数依赖集等价

定义7.14 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7.3 $F^+ = G^+$ 的充分必要条件是 $F \subseteq G^+$, 和 $G \subseteq F^+$

7.3 数据依赖的公理系统

■ 最小依赖集

定义7.15 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**。

7.3 数据依赖的公理系统

■ 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。**

7.3 数据依赖的公理系统

[例] 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 $F' = \{ \text{Sno} \rightarrow \text{Sdept} , \text{Sno} \rightarrow \text{Mname} , \text{Sdept} \rightarrow \text{Mname} ,$

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

7.3 数据依赖的公理系统

[例] 关系模式 $S\langle U, F\rangle$, 其中 :

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

7.3 数据依赖的公理系统

■ 最小依赖集

定理7.3 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。

7.3 数据依赖的公理系统

■ 求F最小依赖集算法

(1) 逐一检查 F 中各函数依赖 $FD_i: X \rightarrow Y$, 若

$$Y = A_1 A_2 \dots A_k, \quad k > 2,$$

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

(利用分解规则, 把右端多属性 (n 个) 的函数依赖变成左端不变 n 个函数依赖)

7.3 数据依赖的公理系统

■ 求F最小依赖集算法

(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。否则保留

7.3 数据依赖的公理系统

■ 求F最小依赖集算法

(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 。

7.3 数据依赖的公理系统

[例] $F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow A\}$, 求 F_{\min}

解：①先分解右端， $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow A, B \rightarrow C, C \rightarrow A\}$

②判断每个函数依赖：

设 $G = F - \{A \rightarrow B\}$,

$A_G^+ = AC$, 不包含 B , 所以 $A \rightarrow B$ 保留

设 $G = F - \{A \rightarrow C\}$,

$A_G^+ = ABC$, 包含 C , 所以 $A \rightarrow C$ 删除, 令 $F = G$,

~~$F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow A\}$~~

7.3 数据依赖的公理系统

[例] $F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow A\}$, 求 F_{\min}

解: $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow A\}$

②判断每个函数依赖:

设 $G = F - \{B \rightarrow A\}$,

$B_G^+ = BAC$, 包含 A , 所以 $B \rightarrow A$ 删除, 令 $F = G$,
 $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

设 $G = F - \{B \rightarrow C\}$,

$B_G^+ = B$, 不包含 C , 所以 $B \rightarrow C$ 保留

7.3 数据依赖的公理系统

[例] $F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow A\}$, 求Fmin

解： $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

②判断每个函数依赖：

设 $G = F - \{C \rightarrow A\}$,

$C_G^+ = C$, 不包含A , 所以 $C \rightarrow A$ 保留

③F中函数依赖的左端都是单属性 , 算法结束。

$Fmin = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

7.3 数据依赖的公理系统

[例] $F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow A\}$, 求Fmin

解：应该注意的是Fmin不是唯一的，如果改变考察次序（从右到左），将得到

$$Fmin = \{A \rightarrow B, A \rightarrow C, B \rightarrow A, C \rightarrow A\}$$

7.3 数据依赖的公理系统

[例] $F = \{BE \rightarrow G, BD \rightarrow G, CDE \rightarrow AB, CD \rightarrow A, CE \rightarrow G, BC \rightarrow A, B \rightarrow D, C \rightarrow D\}$, 求 F_{\min}

解：①先分解右端

$F = \{BE \rightarrow G, BD \rightarrow G, CDE \rightarrow A, CDE \rightarrow B, CD \rightarrow A, CE \rightarrow G, BC \rightarrow A, B \rightarrow D, C \rightarrow D\}$

7.3 数据依赖的公理系统

[例] $F = \{BE \rightarrow G, BD \rightarrow G, CDE \rightarrow AB, CD \rightarrow A, CE \rightarrow G, BC \rightarrow A, B \rightarrow D, C \rightarrow D\}$, 求Fmin

解：②判断每个函数依赖

$F = \{BD \rightarrow G, CDE \rightarrow B, CD \rightarrow A, CE \rightarrow G, B \rightarrow D, C \rightarrow D\}$

7.3 数据依赖的公理系统

[例] $F = \{BE \rightarrow G, BD \rightarrow G, CDE \rightarrow AB, CD \rightarrow A, CE \rightarrow G, BC \rightarrow A, B \rightarrow D, C \rightarrow D\}$, 求 F_{\min}

解: $F = \{BD \rightarrow G, CDE \rightarrow B, CD \rightarrow A, CE \rightarrow G, B \rightarrow D, C \rightarrow D\}$

③ $B_F^+ = BDG$, 包含 G , 用 $B \rightarrow G$ 代替 $BD \rightarrow G$

$(CE)_F^+ = CEBDGA$, 包含 B , 用 $CE \rightarrow B$ 代替 $CDE \rightarrow B$

$C_F^+ = CDA$, 包含 A , 用 $C \rightarrow A$ 代替 $CD \rightarrow A$

$F = \{B \rightarrow G, CE \rightarrow B, C \rightarrow A, CE \rightarrow G, B \rightarrow D, C \rightarrow D\}$

7.3 数据依赖的公理系统

- 关系模型的分析
 - 候选码的确定
 - 对关系模式规范化程度判定

7.3 数据依赖的公理系统

■ 候选码的确定

□ 关系 $R\langle U, F \rangle$ 中， F 是最小函数依赖集

- 准则1：如果属性 A **只在** F 中各个函数依赖的左端出现，则 A 必是码中的属性
- 准则2：如果属性 A **只在** F 中各个函数依赖的右端出现，则 A 必不是码中的属性
- 准则3：如果 A 不在 F 的各个函数依赖中出现，则 A 必不是码中的属性

7.3 数据依赖的公理系统

■ 候选码的确定

□ 根据这些准则，确定候选码的步骤是

- (1) 根据准则3，把不在F中出现的各个函数依赖中出现的属性去掉，因为这些属性一般对模型没什么意义。
- (2) 根据准则1，确定码中必须有的属性集（设为M）
- (3) 根据准则2，去掉码中肯定没有的属性

7.3 数据依赖的公理系统

■ 候选码的确定

□ 根据这些准则，确定候选码的步骤是

(4) 确定余下的属性集（设为W）

(5) 从属性M开始，令 $K=M$ ，如果 $K_F^+=U$ ，K就是候选码。否则从W选择属性加入到K中，直到 $K_F^+=U$ ，K就是候选码。

(6) 注意可能有多个候选码

- 例，设有关系模式 $R(C, T, S, N, G)$, 其函数依赖集 $F = \{C \rightarrow T, CS \rightarrow G, S \rightarrow N\}$ ，求：

(1) R 的候选码

(2) R 最高属于哪级范式

解：(1)

只在左端出现的属性 $M = CS$

只在右端出现的属性 TGN

余下的属性为空

$(CS)_F^+ = CTSNG$

CS 是候选码

- 例，设有关系模式 $R(C, T, S, N, G)$, 其函数依赖集 $F = \{C \rightarrow T, CS \rightarrow G, S \rightarrow N\}$ ，求：

(1) R 的候选码

(2) R 最高属于哪级范式

解：(2)

CS 为码，存在非主属性对码的部分依赖

$CS \rightarrow T (C \rightarrow T), CS \rightarrow N (S \rightarrow N)$

为1NF

- 例：设关系模式 $R(A,B,C,D,E,F)$ ，函数依赖集 $F=\{AB\rightarrow E, AC\rightarrow F, AD\rightarrow B, B\rightarrow C, C\rightarrow D\}$

证明：

(1) AB, AC, AD 均是码

(2) 主属性 C 传递依赖于码 AB

- 例：设关系模式 $R(A,B,C,D,E,F)$ ，函数依赖集 $F = \{AB \rightarrow E, AC \rightarrow F, AD \rightarrow B, B \rightarrow C, C \rightarrow D\}$

证明：

$$(1) (AB)_F^+ = ABCDEF$$

$$A_F^+ = A \neq ABCDEF$$

$$B_F^+ = BCD \neq ABCDEF$$

$$(AC)_F^+ = ABCDEF$$

$$C_F^+ = CD \neq ABCDEF$$

$$(AD)_F^+ = ABCDEF$$

$$D_F^+ = D \neq ABCDEF$$

- 例：设关系模式 $R(A,B,C,D,E,F)$ ，函数依赖集 $F=\{AB\rightarrow E, AC\rightarrow F, AD\rightarrow B, B\rightarrow C, C\rightarrow D\}$

证明：(2)

由 $AB\rightarrow E, AB\rightarrow B$ 得出 $AB\rightarrow BE$

$BE\rightarrow B, B\rightarrow C$ 得出 $BE\rightarrow C$

由此， $AB\rightarrow BE$ ， $BE\not\rightarrow AB$ ， $BE\rightarrow C$

因此主属性C传递依赖于码AB