

第6章 关系数据理论

本章目标

- 完成本章的学习，你应该能够
 - 理解规范化的目的
 - 理解“不好”数据库模式特点
 - 深刻理解并掌握函数依赖和多值依赖的概念，范式的概念
 - 能够根据应用语义完整地写出关系模式的数据依赖集合
 - 熟练掌握如何求关系模式的所有候选码
 - 熟练掌握如何分析某一个关系模式属于第几范式
 - 熟练掌握基本的模式分解方法
 - 熟练掌握多值依赖的判断

大纲

- **问题的提出**
- 规范化
- 数据依赖的公理系统
- 模式分解
- 本章小结

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。

- 即：数据库需要构造多少个关系模式，每个关系由哪些属性组成？

- 数据库逻辑设计的工具——关系数据库的规范化理论。

- 关系模式用一个五元组描述：

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

R：关系名； **U**：全体属性组； **D**：所有属性取值的域； **DOM**：属性到域的映射； **F**：U上的一组数据依赖

■ 第一范式(1NF, Normal Form)

- 对一个二维表R, 如果它所有关系的分量都是不可分的, 则称关系模式R属于第一范式, 简称为 $R \in 1NF$ 。

■ 数据依赖

- 一个关系内部属性与属性之间的一种约束关系: 通过属性间值的相等与否体现数据间的相互联系
- 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

■ 数据依赖的类型

- 函数依赖(Function dependency, FD)
- 多值依赖(Multi-valued dependency, MVD)

■ 函数依赖普遍存在于现实生活中

– 例：学生(学号, 姓名, 系名)

- 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
- 姓名= f (学号)，系名= f (学号)

– 学号函数决定姓名，学号函数决定系名

– 记为：学号 \rightarrow 姓名，学号 \rightarrow 系名

[例6.1]建立一个描述学校教务的数据库，该数据库涉及的对象包括：

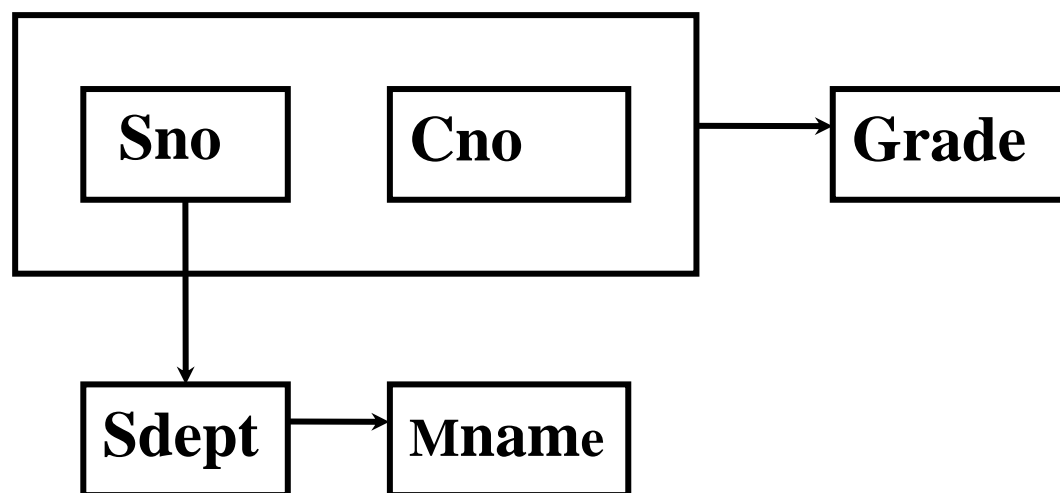
- 学生学号(Sno)
- 所在系(Sdept)
- 系主任姓名(Mname)
- 课程号(Cno)
- 成绩(Grade)

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

■ 现实世界的已知事实(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。

- 假设单一关系模式为：Student<U, F> , 如何确定F?
 - 根据上下文给定的语义, 检查不同属性之间**所有可能的**函数依赖!
 - $F = \{Sno \rightarrow Sdept, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade\}$



问题：确定F的方法能用于甄别候选码吗？

- 对 $R \langle U, F \rangle$ 而言, R 上的所有关系(实例)均满足 F 上的约束; 反之, 如果 R 上给出的部分实例使得 U 上属性具有某种“依赖”关系,
- 问: 能否推断出 R 的所有关系均满足该“依赖”关系?

| Student | Course | Room |
|---------|--------|------|
| Mary | CS145 | B01 |
| Joe | CS145 | B01 |
| Sam | CS145 | B01 |
| .. | .. | .. |



FD $\{\text{Course}\} \rightarrow \{\text{Room}\}$?

- 事实上, 上例只能说明FD可能成立, 却无法证明一定成立!

| EmpID | Name | Phone | Position |
|-------|-------|-------|----------|
| E0045 | Smith | 1234 | Clerk |
| E3542 | Mike | 9876 | Salesrep |
| E1111 | Smith | 9876 | Salesrep |
| E9999 | Mary | 1234 | Lawyer |

| EmpID | Name | Phone | Position |
|-------|-------|--------|----------|
| E0045 | Smith | 1234 | Clerk |
| E3542 | Mike | 9876 ← | Salesrep |
| E1111 | Smith | 9876 ← | Salesrep |
| E9999 | Mary | 1234 | Lawyer |

$\{\text{Position}\} \rightarrow \{\text{Phone}\}$

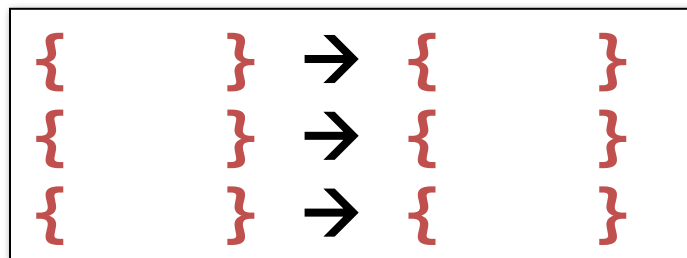
| EmpID | Name | Phone | Position |
|-------|-------|--------|----------|
| E0045 | Smith | 1234 → | Clerk |
| E3542 | Mike | 9876 | Salesrep |
| E1111 | Smith | 9876 | Salesrep |
| E9999 | Mary | 1234 → | Lawyer |

but *not* $\{\text{Phone}\} \rightarrow \{\text{Position}\}$

课堂练习一

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | 2 | 4 | 3 | 6 |
| 3 | 2 | 5 | 1 | 8 |
| 1 | 4 | 4 | 5 | 7 |
| 1 | 2 | 4 | 3 | 6 |
| 3 | 2 | 5 | 1 | 8 |

找到至少3个与此实例相冲突的FDs:



- 由下面的关系表, 能否得出 $Sno \rightarrow Sname$? $Sno \rightarrow Sdept$?

| 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 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

■ Student<U, F>存在的问题:

– 数据冗余

- 浪费大量的存储空间
- 每一个系主任的姓名重复出现, 重复次数与该系所有学生的所有课程成绩出现次数相同

– 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
- 某系更换系主任后, 必须修改与该系学生有关的每一个元组

– 插入异常(Insertion Anomalies)

- 如果一个系刚成立, 尚无学生, 则无法把这个系及其系主任的信息存入数据库

– 删除异常(Deletion Anomalies)

- 如果某个系的学生全部毕业了, 则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

■ 结论:

- Student关系模式不是一个好的模式
- 一个“好”的模式应当不会发生插入异常、删除异常和更新异常，数据冗余应尽可能少

■ 原因:

- 由模式中的某些数据依赖引起的

■ 解决方案:

- 用规范化理论改造关系模式来消除其中不合适的数据依赖


不会发生插入异常、删除异常，冗余得到控制



Student $\langle U, F \rangle$ 

S(Sno, Sdept, Sno \rightarrow Sdept);
SC(Sno, Cno, Grade, (Sno,Cno) \rightarrow Grade);
DEPT(Sdept, Mname, Sdept \rightarrow Mname);

大纲

- 问题的提出
- 规范化  根据函数依赖分解一个不好的关系模式以达到消除更新异常、减少冗余的过程
- 数据依赖的公理系统
- 模式分解
- 本章小结

规范化

- 函数依赖
- 码
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF
- 规范化小结

函数依赖

■ 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 $X \rightarrow Y$ 。
- 函数依赖属于语义范畴，只能根据语义来确定一个函数依赖。如， $姓名 \rightarrow 年龄$ 只有在该部门没有同名人的条件下成立。如果允许有同名名人，则年龄不在函数依赖于姓名
- 函数依赖不是指关系模式 R 的某个或某些关系满足的约束条件，而是指 R 的一切关系均要满足的约束条件。

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

- 假设不允许重名，则所有的函数依赖如下：

Sno \rightarrow Ssex, Sno \rightarrow Sage, Sno \rightarrow Sdept, Sno \rightarrow Sname, Sname \rightarrow Sno,
Sname \rightarrow Ssex, Sname \rightarrow Sage, Sname \rightarrow Sdept

- 思考：

- Ssex与Sage, Ssex与Sdept, Sage与Sdept之间是否存在函数依赖？
 - 假设允许重名，请列出Student上所有的函数依赖

- 非平凡的函数依赖

- $X \rightarrow Y$, 但 $Y \not\subseteq X$, 则称 $X \rightarrow Y$ 是非平凡的函数依赖

- 平凡的函数依赖

- $X \rightarrow Y$, 但 $Y \subseteq X$, 则称 $X \rightarrow Y$ 是平凡的函数依赖

- 平凡函数依赖对任何关系都成立, 它不反映新的语义。若无特别声明, 总讨论非平凡的函数依赖

- 若 $X \rightarrow Y$, 则 X 称为这个函数依赖的决定因素(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 $X \nrightarrow Y$

■ 完全函数依赖

- 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 $X' \nrightarrow Y$, 则称Y 对X **完全函数依赖**, 记作 $X \xrightarrow{F} Y$

■ 部分函数依赖

- 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: 在关系SC(Sno, Cno, Grade)中,

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

■ 传递函数依赖

- 定义6.3 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, 记作 $X \xrightarrow{\text{传递}} Z$ 。

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: 在关系 $Std(Sno, Sdept, Mname)$ 中,
 - $Sno \rightarrow Sdept$, $Sdept \rightarrow Mname$, $Mname$ 传递函数依赖于 Sno

规范化

- 函数依赖
- **码**
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF
- 规范化小结

码

- 码(key)也称键或键码，是关系模式中的一个重要概念。
 - 用函数依赖的概念来定义码。
- 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 的一个候选码(Candidate Key)
 - 如果 U 部分函数依赖于 K ，即 $K \xrightarrow{P} U$ ，则 K 称为超码(Superkey)。
 - 候选码是最小的超码，即 K 的任意一个真子集都不是候选码。
- 若候选码多于一个，则选定其中的一个为主码。
- 必备技能：求一个关系模式候选码的方法(Armstrong公理)

■ 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(all-key)

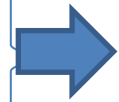
- 整个属性组是码

该定义是否等价于：所有属性都是主属性？

[例6.2] S(Sno, Sdept, Sage)中, Sno是单属性码, SC(Sno, Cno, Grade)中, (Sno, Cno)是多属性码, 请指出该关系模式中所有的主属性和非主属性

[例6.3] $R(P,W,A)$ 中, 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) , 即全码

▪ 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**也称**外码**。

- $SC(Sno, Cno, Grade)$ 中, Sno 不是码。
 - Sno 是 $S(Sno, Sdept, Sage)$ 的码, 则 Sno 是 SC 的外码。
- 主码与外部码一起提供了表示关系间联系的手段。
- 如, Sno

规范化

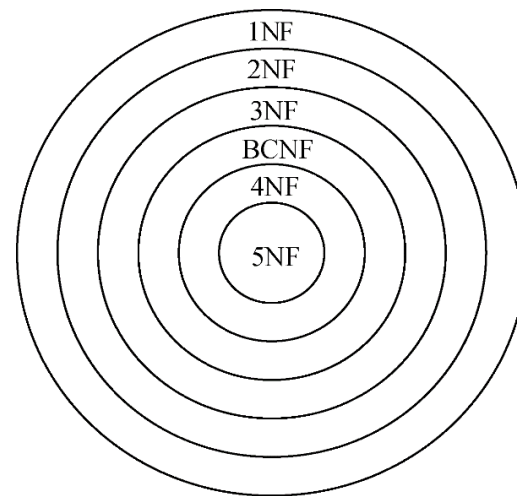
- 函数依赖
- 码
- **范式**
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF
- 规范化小结

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。

- **范式的种类及关系**

- $1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$
- 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(normalization)**。

规范化

- 函数依赖
- 码
- 范式
- **2NF**
- 3NF
- BCNF
- 多值依赖
- 4NF
- 规范化小结

2NF

■ 2NF 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任何一个候选码，则 $R \in 2NF$ 。

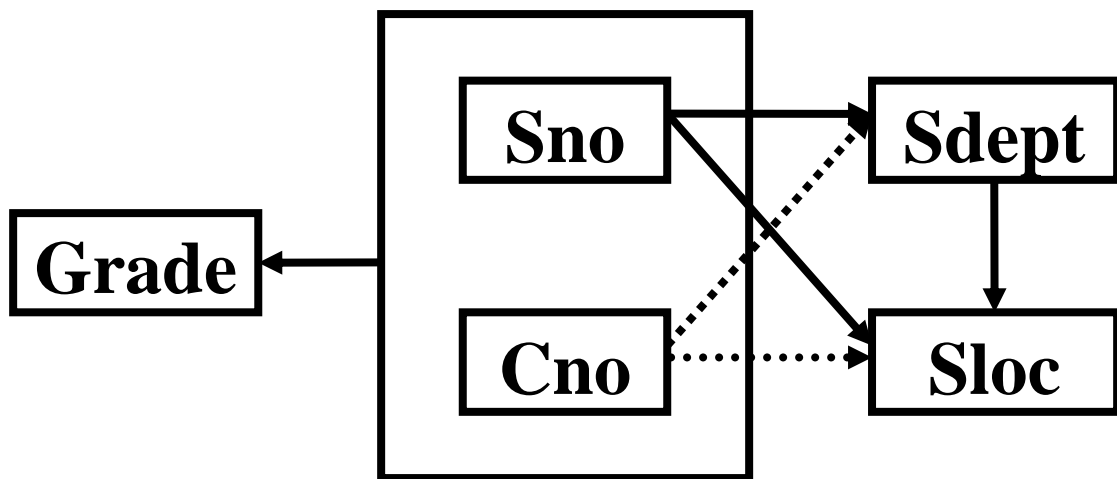
- [例6.4] S-L-C(Sno, Sdept, Sloc, Cno, Grade), Sloc为学生的住处，并且每个系的学生住在同一个地方。

- S-L-C的码为(Sno, Cno)

- 函数依赖有：(Sno, Cno) \xrightarrow{F} Grade, Sdept \rightarrow Sloc

- Sno \rightarrow Sdept, (Sno, Cno) \xrightarrow{P} Sdept

- Sno \rightarrow Sloc, (Sno, Cno) \xrightarrow{P} Sloc



- 非主属性Sdept、Sloc并不完全依赖于码
- 关系模式S-L-C \notin 2NF

观察:

如果关系模式R的主码由单属性构成, 则R一定属于2NF

一个关系模式不属于2NF, 会产生以下问题:

- 插入异常

如果插入一个新学生, 但该生未选课, 即该生无Cno, 由于插入元组时, 必须给定码值, 因此插入失败。

- 删除异常

如果S4只选了一门课C3, 现在他不再选这门课, 则删除C3后, 整个元组的其他信息也被删除了。

- 修改复杂

如果一个学生选多门课, 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。

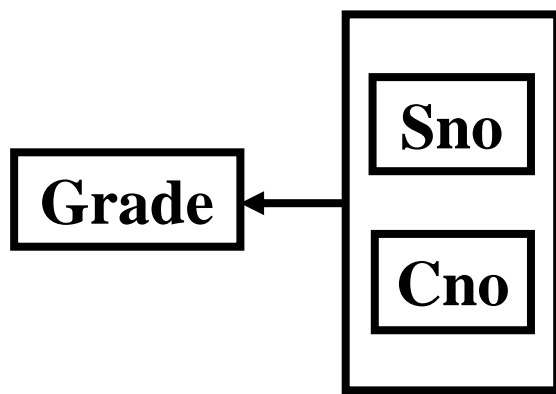
原因:

有两类非主属性, 一类对码完全函数依赖(如Grade); 一类对码不完全函数依赖(如Sdept, Sloc)

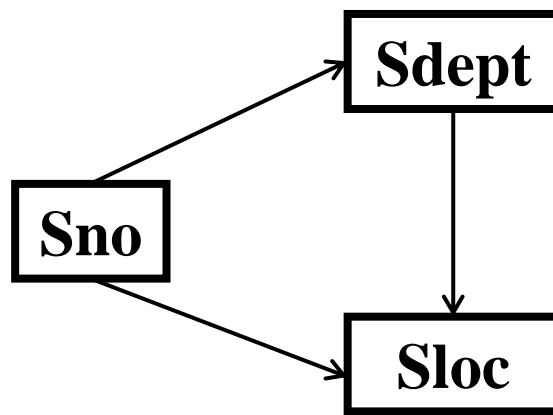
■ 解决办法:

— 投影分解

- 将S-L-C分解成两个关系模式: SC(Sno, Cno, Grade)和S-L(Sno, Sdept, Sloc)



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



$Sno \xrightarrow{F} Sdept, Sno \xrightarrow{F} Sloc$

结论: SC和S-L均不存在部分函数依赖, 都是完全函数依赖

规范化

- 函数依赖
- 码
- 范式
- 2NF
- **3NF**
- BCNF
- 多值依赖
- 4NF
- 规范化小结

3NF

▪ 3NF 定义6.7

– 设关系模式 $R\langle U, F \rangle \in 1NF$, 若 R 中不存在这样的码 X 、属性组 Y 及非主属性 $Z (Y \not\supset Z)$, 使得 $X \rightarrow Y (Y \not\rightarrow X)$, $Y \rightarrow Z$ 成立, 则称 $R\langle U, F \rangle \in 3NF$ 。

▪ 可以证明, 如果 $R \in 3NF$, 则 $R \in 2NF$. (第6章习题8)

▪ 课堂练习:

– 请指出关系模式 $SC(Sno, Cno, Grade)$ 和 $S-L(Sno, Sdept, Sloc)$ 是否属于 3NF? 根据是什么? 如果不是 3NF, 请将其分解到 3NF。

• $SC \in 3NF$

• $S-L \in 2NF, S-L \notin 3NF$

• 将 $S-L$ 分解为: $S-D(Sno, Sdept)$, $D-L(Sdept, Sloc)$, 则 $S-D \in 3NF, D-L \in 3NF$

■ 保持无损连接性的一种分解算法-Heath定理

– 无损连接性(non-loss join)

If R is a relation and P_1, P_2, \dots, P_n are projections on R such that

$$P_1 \text{ JOIN } P_2 \text{ JOIN } \dots \text{ JOIN } P_n = R,$$

then P_1, P_2, \dots, P_n constitutes a non-loss decomposition of R .

– Heath定理(Heath's Theorem)

If we have a relation $R\{A, B, C, \dots\}$ and if $A \rightarrow B$ and $B \rightarrow C$, then projections $P_1\{A, B\}$ and $P_2\{B, C\}$ constitute a non-loss decomposition of R .

- How do we find non-loss decompositions?
- When should we replace a relation by a non-loss decomposition?

■ 范式判断示例：

R0{Suppl#, SupplName, Item#, ItemName, Quantity, SupplStatus, Location} PK [Suppl#, Item#]

[Suppl#, Item#] \rightarrow {Quantity, SupplName, SupplStatus, Location, ItemName}

Suppl# \rightarrow {SupplName, SupplStatus, Location}

Item# \rightarrow ItemName



R1{Suppl#, SupplName, Location, SupplStatus} PK[Suppl#]

R2{Item#, ItemName} PK [Item#]

R3{Suppl#, Item#, Quantity} PK[Suppl#, Item#]

经2NF定义分析，R1，R2，R3都属于2NF。

规范化

- 函数依赖
- 码
- 范式
- 2NF
- 3NF
- **BCNF**
- 多值依赖
- 4NF
- 规范化小结

BCNF

- BCNF(Boyce Codd Normal Form)

- 由Boyce和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。

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

- 换言之，在关系模式 $R\langle U, F \rangle$ 中，如果每一个决定属性集都包含候选码，则 $R \in BCNF$ 。

- BCNF的关系模式所具有的性质：

- 所有非主属性都完全函数依赖于每个候选码；
- 所有主属性都完全函数依赖于每个不包含它的候选码：
- 没有任何属性完全函数依赖于非码的任何一组属性。

课堂练习：证明之

- 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。
- 关于BCNF, 3NF, 2NF之间关系的结论：
 - $R \in \text{BCNF} \Rightarrow R \in 3\text{NF}$ (第6章习题8)
 - $R \in \text{BCNF} \not\Leftrightarrow R \in 3\text{NF}$ (请给出例子, 后面例子6.8)

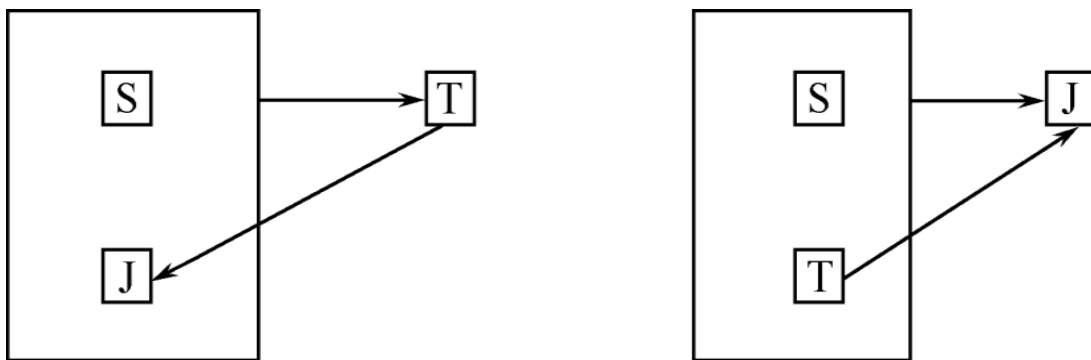
[例6.5] C(Cno, Cname, Pcno)

- 只有唯一的码: Cno
- 没有任何属性对Cno部分依赖或传递依赖, 所以 $C \in 3\text{NF}$
- C中Cno是唯一的决定因素, 所以 $C \in \text{BCNF}$
- 课堂练习: 分析SC(Sno, Cno, Grade)的最高范式

[例6.7] SJP(S,J,P)中, S是学生, J表示课程, P表示名次。每一个学生选修每门课程的成绩有一定的名次, 每门课程中每一名次只有一个学生(即没有并列名次)

- 由语义可得到函数依赖: $(S, J) \rightarrow P, (J, P) \rightarrow S$
- (S, J)与(J, P)都可以作为候选码
- 关系模式中没有属性对码传递依赖或部分依赖, 所以 $SJP \in 3NF$
- 除(S,J)与(J,P)以外没有其他决定因素, 所以 $SJP \in BCNF$

- [例6.8] STJ(S,T,J)中, S是学生, 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。
 - 由语义可得到函数依赖: $(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$
 - 因为没有任何非主属性对码的传递依赖或部分依赖, 所以 $STJ \in 3NF$.
 - $STJ \notin BCNF$, 因为T是决定因素, 而T不包含码



- 对于不是BCNF的关系模式，仍然存在不合适的地方。
- 非BCNF的关系模式也可以通过分解成为BCNF。例如STJ可分解为ST(S,T)与TJ(T,J)，它们都是BCNF。
- 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。
- 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，它已实现了彻底的分离，已消除了插入和删除的异常。
- 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。

■ 分解成BCNF的一种算法:

– 假设关系模式 $R(U)$, 分解算法BCNFDecomp(R):

- 找到属性(组) X , 满足 $X^+ \neq X$, $X^+ \neq U$;
- 如果没找到, 则返回 R 。
- 否则, 令 $Y = X^+ - X$, $Z = (X^+)^c$, 则 R 分解为 $R1(XUY)$, $R2(XUZ)$

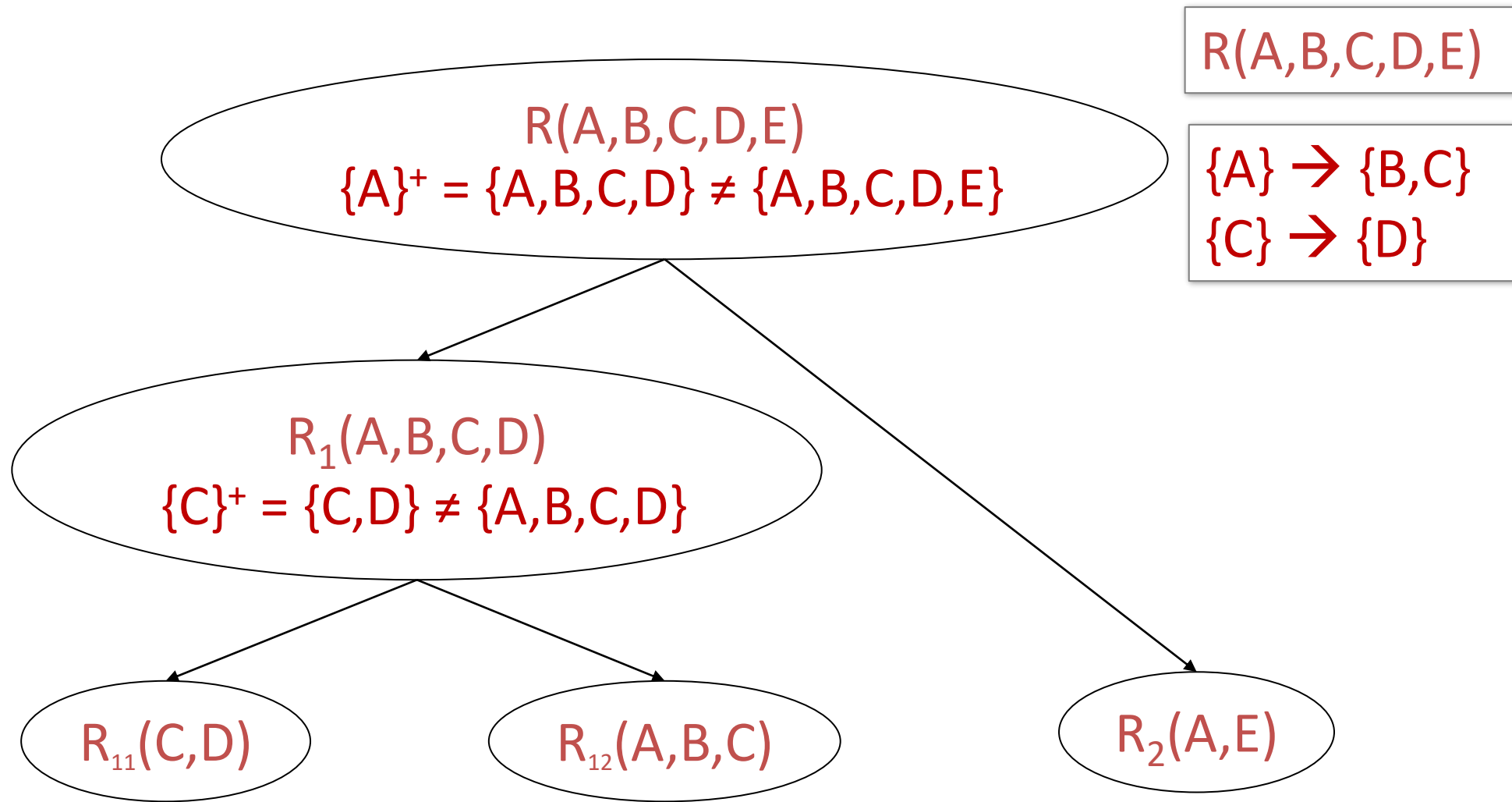
■ 此算法能够保持无损连接性

– 教材中的关系模式STJ分解成ST(S,T)与TJ(T,J)就是用此算法

- **[例题]:** 设有关系模式 $R(A, B, C, D, E)$, $F=\{A \rightarrow (B, C), C \rightarrow D\}$, 判定R的最高范式。如果不是BCNF, 则分解到BCNF。

[解]:

- 通过分析可知, AE为R的主码(只有一个候选码), 且存在非主属性 (B,C) 对码的部分依赖, 所以R属于1NF。
- 利用BCNF分解算法, 可得到R如下的一种分解结果。



函数依赖范畴内关系模式范式的确定

■ 传统方法(Bottom-top)

1. 找出F;
2. 验证R是否符合2NF定义。若否, $R \in 1NF$; 若是, 执行步骤3;
3. 验证R是否符合3NF定义。若否, $R \in 2NF$; 若是, 转向执行步骤4;
4. 验证R是否符合BCNF定义。若否, $R \in 3NF$; 若是, $R \in BCNF$

■ 新方法(Top-bottom)

- 依据: $1NF \supset 2NF \supset 3NF \supset BCNF$
- 找出F; 先验证R是否符合BCNF定义。若是, 则 $R \in BCNF$; 若否, 再验证R是否符合3NF定义, 若是, 则 $R \in 3NF$; 若否, 则验证R是否符合2NF定义, 若是, 则 $R \in 2NF$; 若否, 则 $R \in 1NF$ 。

- 两种方法的前提是需要先确定R的候选码
 - 求出R的所有候选码方法: **Armstrong公理**(后续章节)
 - **原因**: 只单纯从语义出发得出的F未必是完全没有遗漏
- **BCNF的确定**
 - 求出R的所有候选码;
 - 列出F的所有函数依赖;
 - 如果F中所有的函数依赖中的决定因素都包含码, 则 $R \in \text{BCNF}$; 否则, $R \notin \text{BCNF}$, R最高只能是3NF。
- [例] **TEACH**(STUDENT,COURSE,INSTRUCTOR)
 - $F = \{(STUDENT, COURSE) \rightarrow INSTRUCTOR, INSTRUCTOR \rightarrow COURSE\}$
 - **TEACH** \in 3NF (为什么不是**TEACH** \in 2NF?)

规范化

- 函数依赖
- 码
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF
- 规范化小结

多值依赖

例[6.9] 设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。用Teaching(C, T, B)表示上述关系模式。

表6.3 非规范化关系示例

| 课程 C | 教员 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 ...} |
| ... | ... | ... |

表6.4 规范化的二维表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 | 普通物理学 |
| 数学 | 李勇 | 光学原理 |
| 数学 | 李勇 | 物理习题集 |
| 数学 | 张平 | 普通物理学 |
| 数学 | 张平 | 光学原理 |
| 数学 | 张平 | 物理习题集 |
| ... | ... | ... |

■ Teaching模式分析

– 候选码：唯一， $(C,T,B) \Rightarrow$ 全码

– Teaching \in BCNF

– 但Teaching仍然存在冗余度大，更新异常的问题

原因：存在多值依赖



| 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 | 普通物理学 |
| 数学 | 李 勇 | 光学原理 |
| 数学 | 李 勇 | 物理习题集 |
| 数学 | 张 平 | 普通物理学 |
| 数学 | 张 平 | 光学原理 |
| 数学 | 张 平 | 物理习题集 |
| ... | ... | ... |

- 冗余度大：有多少名任课教师，参考书就要存储多少次
- 插入操作复杂：当某一课程增加一名任课教师时，该课程有多少本参照书，就必须插入多少个元组
- 删除操作复杂：某一门课要去掉一本参考书，该课程有多少名教师，就必须删除多少个元组
- 修改操作复杂：某一门课要修改一本参考书，该课程有多少名教师，就必须修改多少个元组

■ 多值依赖(Multi-valued dependency, MVD) 定义6.9

- 设 $R(U)$ 是属性集 U 上的一个关系模式。 X, Y, 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 $C \twoheadrightarrow T$ 。

■ 平凡的多值依赖

- 若 $X \twoheadrightarrow Y$, 而 $Z = \Phi$, 即 Z 为空, 则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖。
- 即, 对于 $R(X, Y)$, 如果有 $X \twoheadrightarrow Y$ 成立, 则 $X \twoheadrightarrow Y$ 为平凡的多值依赖。

■ 非平凡的多值依赖

■ 多值依赖的另一个等价的定义：

- 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， $Z=U-X-Y$ 。

[例6.10] $WSC(W, S, C)$ ： W -仓库， S -保管员， C -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 |

[例6.10]的多值依赖分析

- 对于W的每一个值 W_i ，S有一个完整的集合与之对应而不问C取何值。所以 $W \twoheadrightarrow S$ 。
- C与S的完全对称性，必然有 $W \twoheadrightarrow C$ 成立。

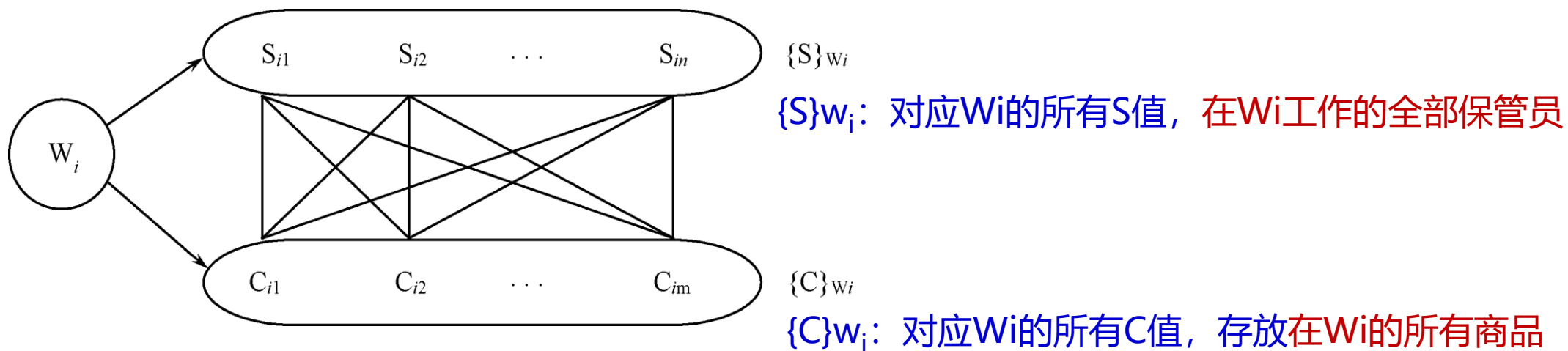


图6.7 $W \twoheadrightarrow S$ 且 $W \twoheadrightarrow C$

■ 多值依赖的性质:

- 对称性: 即若 $X \twoheadrightarrow Y$, 则 $X \twoheadrightarrow Z$, 其中 $Z = U - X - Y$
- 可用完全二分图直观地表示出来
- 传递性: 即若 $X \twoheadrightarrow Y$, $Y \twoheadrightarrow Z$, 则 $X \twoheadrightarrow Z - Y$
- 函数依赖是多值依赖的特殊情况。即若 $X \rightarrow Y$, 则 $X \twoheadrightarrow Y$ 。
- 若 $X \twoheadrightarrow Y$, $X \twoheadrightarrow Z$, 则 $X \twoheadrightarrow YZ$
- 若 $X \twoheadrightarrow Y$, $X \twoheadrightarrow Z$, 则 $X \twoheadrightarrow Y \cap Z$
- 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

– 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。
- 这是因为多值依赖的定义中不仅涉及属性组 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。

■ 多值依赖与函数依赖的区别(cont'd)

- 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。

[例] 关系 $R(A, B, C, D)$, $A \twoheadrightarrow BC$ 成立, 当然也有 $A \twoheadrightarrow D$ 成立。但表6.6说明, 有 R 的一个关系实例, 在此实例上 $A \twoheadrightarrow B$ 不成立。

表6.6 R 的一个实例

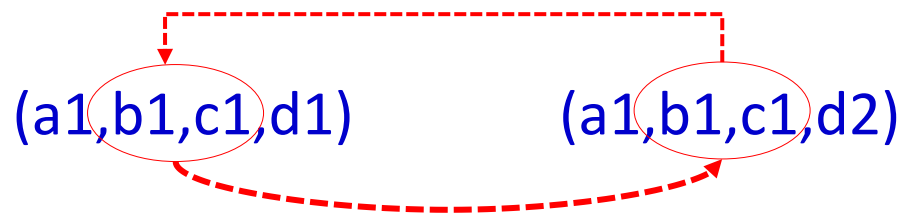
| A | B | C | D |
|-------|-------|-------|-------|
| a_1 | b_1 | c_1 | d_1 |
| a_1 | b_1 | c_1 | d_2 |
| a_1 | b_2 | c_2 | d_1 |
| a_1 | b_2 | c_2 | d_2 |

■ 多值依赖的判定方法

– 以上例为例进行说明：

1. $A \twoheadrightarrow BC$ 成立：

- 先选取第一条元组 r ，得出 $r(A)$ 的值，再往下从第二条元组开始，依次检查后续元组 r' 在 A 上的分量是否等于 $r(A)$ 。若相等，则调换两条元组在 BC 分量并检查调换后得到新的两个元组是否还是 R 中，若在，则继续上述比较过程，若不在，则表明 $A \nrightarrow BC$ ，即不满足多值依赖。
- 示例操作：因为第1，2条元组在属性 A 上的值相同，交换 $(a1, b1, c1, d1)$ 与 $(a1, b1, c1, d2)$ 在 BC 上的值后仍然在 R 中。穷尽实例上的所有元组，可得最后结果。



2. $A \twoheadrightarrow B$ 不成立

- 因为交换 $(a1, b1, c1, d1)$ 与 $(a1, b2, c2, d1)$ 在 B 上的分量后变为： $(a1, b2, c1, d1)$ 和 $(a1, b1, c2, d1)$ ，这两条新元组都不在 R 中，所以 $A \nrightarrow B$ 。

课堂练习一

1. 右边为关系模式R的一个实例，问：该实例满足以下哪个多值依赖？

- $B \twoheadrightarrow A$
- $D \twoheadrightarrow BC$
- $A \twoheadrightarrow A$
- $D \twoheadrightarrow A$

| A | B | C | D |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 1 | 3 | 3 | 3 |
| 1 | 3 | 3 | 4 |
| 1 | 2 | 3 | 3 |
| 2 | 2 | 4 | 4 |
| 2 | 4 | 2 | 4 |
| 2 | 4 | 4 | 4 |
| 2 | 2 | 2 | 4 |

2. 假设关系模式R(A, B, C, D, E)有如下多值依赖： $A \twoheadrightarrow B$, $B \twoheadrightarrow D$
现在R包含两个元组(0,1,2,3,4)和(0,5,6,7,8)，请问下列哪些元组必须也包含在R中？

- (0,5,2,7,4)
- (0,5,6,7,4)
- (0,1,2,3,8)
- (0,5,2,7,8)

规范化

- 函数依赖
- 码
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- **4NF**
- 规范化小结

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 $R\langle U, F \rangle \in 4NF$ 。
- 4NF就是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**。
- 对于每一个非平凡的多值依赖 $X \twoheadrightarrow Y$ ，X都含有候选码，于是 $X \rightarrow Y$ ，所以 4NF 所允许的非平凡多值依赖实际上是函数依赖。
- **$R \in 4NF \Rightarrow R \in BCNF$**
- 分析[例6.10]中的 **$WSC \in BCNF$** ，但 **$WSC \notin 4NF$**
 - 把WSC分解成WS(W,S), WC(W,C)，则 $WS \in 4NF$ ， $WC \in 4NF$ 。

■ 将BCNF的R分解成4NF的方法

- 假设 $R(A,B,C) \in \text{BCNF}$ 满足 $A \twoheadrightarrow B$, $A \twoheadrightarrow C$, 则分解R可分解为 $R_1(A, B)$ 和 $R_2(A, C)$, 要求 $R_1(A, B)$ 和 $R_2(A, C)$ 都是平凡的多值依赖。
- 其它形式的分解类似, 原则是分解后的所有关系模式必须都是平凡的多值依赖。

■ 将[例6.10]中的WSC分解成4NF

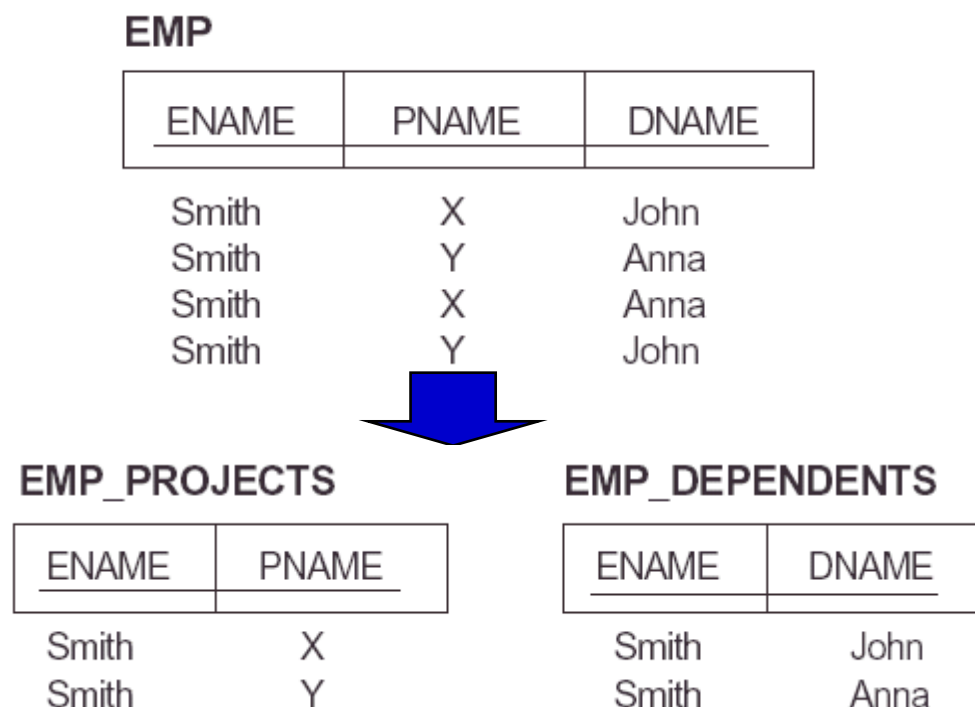
- 经分析, $\text{WSC} \in \text{BCNF}$, 且 $\text{WSC} \notin 4\text{NF}$ 。
- 因 $W \twoheadrightarrow S$, $W \twoheadrightarrow C$, 可分解WSC为 $WS(W,S)$ 和 $WC(W,C)$, $WS(W,S)$ 和 $WC(W,C)$ 都是平凡的多值依赖, 且 $WS \in 4\text{NF}$, $WC \in 4\text{NF}$ 。

- [例] 假设EMP (Ename, Pname, Dname)(注：全码)具有多值依赖：

$ENAME \twoheadrightarrow PNAME, ENAME \twoheadrightarrow DNAME$

问题：请判断EMP \in 4NF是否成立？若不成立，请将EMP分解成4NF范式。

- 全码，不成立



课堂练习二

- 对于下列各个关系模式和依赖，判定其是否为4NF，若不是，则将关系模式分解成满足4NF的关系：
 - ① $R(A, B, C)$ ，存在如下依赖： $A \twoheadrightarrow B$ 和 $A \twoheadrightarrow C$ ；
 - ② $R(A, B, C, D)$ ，存在如下依赖： $A \twoheadrightarrow C$ 和 $C \twoheadrightarrow BD$ 。

提示：

- 要判定一个关系模式是否为4NF，首先要确定码；然后验证是否符合4NF的定义
- 分解方法见前面的“[将BCNF的R分解成4NF的方法](#)”

- 函数依赖和多值依赖是两种最重要的数据依赖。
 - BCNF为规范化程度最高的函数依赖
 - 4NF为规范化程度最高的多值依赖
- 连接依赖：一种除函数依赖和多值依赖外的数据依赖。
 - 5NF
- 函数依赖是多值依赖的一种特殊情况，多值依赖又是连接依赖的一种特殊情况

课堂练习三

- 下列关于函数依赖的叙述中，哪一条是不正确的？
 - A. 若 $X \rightarrow Y$, $Y \rightarrow Z$, 则 $X \rightarrow Z$
 - B. 若 $X \rightarrow Y$, $Y \rightarrow Y'$, 则 $X \rightarrow Y'$
 - C. 若 $X \rightarrow Y$, $X' \rightarrow X$, 则 $X' \rightarrow Y$
 - D. 若 $X' \rightarrow X$, 则 $X \rightarrow X'$
- 关系数据库规范化是为解决关系数据库中的_____问题而引入的
 - A. 操作异常和数据冗余
 - B. 提高查询速度
 - C. 减少数据操作的复杂性
 - D. 保证数据的安全性和完整性
- 假设关系模式属于R(A,B)属于3NF，下列说法中_____是正确的
 - A. 它一定消除了插入和删除异常
 - B. 仍存在一定的插入或删除异常
 - C. 一定属于BCNF
 - D. B和C均是正确的

规范化

- 函数依赖
- 码
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF
- **规范化小结**

规范化小结

- 关系数据库中的所有关系模式必须满足第一范式。
- 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
 - 解决方法就是对其投影分解，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。
- 关系数据库的规范化理论是数据库逻辑设计的工具。

■ 规范化的基本思想

- 是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”。
- “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
- 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。

- 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

■ 规范化过程：

关系模式规范化的基本步骤



大纲

- 问题的提出
- 规范化
- **数据依赖的公理系统**
- 模式分解
- 本章小结

数据依赖的公理系统

- **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逻辑蕴涵 $X \rightarrow Y$ 。
- **Armstrong公理系统**
 - 一套推理规则，是模式分解算法的理论基础。
 - 1974年由Armstrong提出。
 - **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

■ Armstrong公理系统:

- 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 rule)

- 若 $Y \subseteq X \subseteq U$, 则 $X \rightarrow Y$ 为 F 所蕴涵.

A2 增广律(augmentation rule)

- 若 $X \rightarrow Y$ 为 F 所蕴涵, 且 $Z \subseteq U$, 则 $XZ \rightarrow YZ$ 为 F 所蕴涵.

A3 传递律(transitivity rule)

- 若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 F 所蕴涵, 则 $X \rightarrow Z$ 为 F 所蕴涵

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。
 - 证明请自行看书
- 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 $X \rightarrow Z$
- 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 $R\langle U, F \rangle$ 中为 F 所逻辑蕴涵的函数依赖的全体叫作 **F 的闭包**，记为 F^+ 。
- **Armstrong是有效的**
 - 指 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
- **Armstrong是完备的**
 - 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
- **定义6.13** 设 F 为属性集 U 上的一组函数依赖， $X, Y \subseteq U$ ， $X_F^+ = \{ A | X \rightarrow A \text{ 能由 } F \text{ 根据Armstrong公理导出} \}$ ， X_F^+ 称为属性集 X 关于函数依赖集 F 的**闭包**
- **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 **$Y \subseteq X_F^+$**
 - **作用：**判定 $X \rightarrow Y$ 是否能由 F 根据Armstrong公理导出的问题，就转化为求出 X_F^+ ，判定 Y 是否为 X_F^+ 的子集问题

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^+

计算步骤:

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 $X^{(i+1)} = X^{(i)}$

5. 若 $X^{(i+1)}$ 与 $X^{(i)}$ 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

- **[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 $(AB)_F^+$ 。

课堂练习(四)

- 设有关系模式 $R(A, B, C, D)$ 及其函数依赖集 $F=\{D\rightarrow B, B\rightarrow D, AD\rightarrow B, AC\rightarrow D\}$ ，求 $(AC)_F^+$ 和 $(AB)_F^+$ 。

- 利用引理6.2求给定关系模式的所有候选码

- 结论：在关系模式 $R\langle U, F \rangle$ 中，若 $A_F^+ = U$ ，则A为R的候选码

- [例] 设 $U = \{A, B, C, D\}$, $F = \{A \rightarrow B, C \rightarrow D\}$ ，试求此关系的候选码。

- 求给定关系模式所有候选码的一般思路：

- 只出现在**箭头右侧**的属性一定不会是码的组成部分，因为它不可能是决定因素；
 - 只出现在**箭头左侧**的属性一定是码的组成部分，因为它只能是决定因素；
 - 既出现在**箭头左侧**又出现在**箭头右侧**的属性需要进行验证是否为码的组成部分。

- 本题分析

- A, C一定是码的组成部分；B, D一定不是码的组成部分。因为 $(AC)_F^+ = U$ ，所以R只有一个候选码AC

课堂练习(五)

- 设有关系模式 $R(A, B, C, D, E, P)$ 及其函数依赖集 $F=\{A\rightarrow D, E\rightarrow D, D\rightarrow B, BC\rightarrow D, DC\rightarrow A\}$, 求 R 的所有候选码。
- 关系模式 $R\langle U, F \rangle$, $U=\{A, B, C, D, E, P\}$, $F=\{A\rightarrow B, C\rightarrow D, E\rightarrow A, CE\rightarrow D\}$, 试求此关系的候选码。

■ 关系模式R规范化程度的判断(BCNF内)

[例] 关系模式 $R\langle U, F\rangle$ ，请针对以下不同情形的U和F，确定R的最高范式。

1. $U=\{A, B, C, D\}$, $F=\{B\rightarrow D, AB\rightarrow C\}$;
2. $U=\{A, B, C, D, E\}$, $F=\{AB\rightarrow CE, E\rightarrow AB, C\rightarrow D\}$;
3. $U=\{A, B, C\}$, $F=\{A\rightarrow B, B\rightarrow A, A\rightarrow C\}$

[解]:

1. 首先确定R的候选码：AB，因此AB也是主码。接着分析所有非主属性对AB的函数依赖情况：D部分函数依赖于AB，所以 $R\in 1NF$ 。
2. 易得R有两个候选码：AB、E。因为 $R\notin BCNF$ ，所以R最高为3NF。由于 $AB\rightarrow C$, $AB\rightarrow E$, $C\rightarrow D$ ，存在非主属性对码的传递依赖，所以 $R\notin 3NF$ 。又F中不存在部分依赖，所以R最高范式为2NF。
3. 作为课堂练习，请自行完成

- 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 $G \subseteq F^+$

- 判定 $F \subseteq G^+$ 的方法:

- 根据引理6.3, 只需逐一对F中的函数依赖 $X \rightarrow Y$ 考察Y是否属于 $X_{G^+}^+$

- 例题:

- $F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$, $G = \{A \rightarrow CD, E \rightarrow AH\}$, 证明F与G等价.

▪ 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, 或称**最小依赖集、最小覆盖**(minimal cover)。

- F中任一函数依赖的右部仅含有一个属性;
- 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;
- 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6.12]

- 考察6.1节中的关系模式 $S \langle U, F \rangle$, 其中, $U = \{Sno, Sdept, Mname, Cno, Grade\}$, $F = \{Sno \rightarrow Sdept, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade\}$ 。
- 设 $F' = \{Sno \rightarrow Sdept, Sno \rightarrow Mname, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade, (Sno, Sdept) \rightarrow Sdept\}$
- 可以验证, F为最小覆盖; 但F'不是, 因为 $F' - \{Sno \rightarrow Mname\}$ 与F'等价

■ 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。

■ 构造性证明:

- 分三步对F进行“极小化处理”，找出F的一个最小依赖集。
 1. 逐一检查F中各函数依赖 $FD_i: X \rightarrow Y$ ，若 $Y = A_1 A_2 \dots A_k$ ， $k \geq 2$ ，则用 $\{X \rightarrow A_j \mid j = 1, 2, \dots, k\}$ 来取代 $X \rightarrow Y$ 。（注：引理6.1保证了变换的等价性）
 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（因为F与G等价的充要条件是 $A \in X_G^+$ ，保证了变换的等价性）
 3. 逐一取出F中各函数依赖 $FD_i: X \rightarrow A$ ，设 $X = B_1 B_2 \dots B_m$ ， $m \geq 2$ ，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（因为F与 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 等价的充要条件是 $A \in Z_F^+$ ，其中 $Z = X - B_i$ ，这保证了变换的等价性）

- 定理6.3的证明过程就是求F极小依赖集的过程，也是检验F是否为极小依赖集的一个算法。
- 但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。

[例6.13] $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A\}$, $F_{m1} = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$,
 $F_{m2} = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A\}$

- 可以验证, F_{m1} 和 F_{m2} 是F的两个不同的最小依赖集。

■ 最小依赖集的求法示例:

- 设有依赖集: $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C, CG \rightarrow BD, CE \rightarrow AG\}$, 计算与其等价的最小依赖集。

解:

1. 将依赖右边属性**单一化**

$$F1 = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow B, CG \rightarrow D, CE \rightarrow A, CE \rightarrow G\}$$

2. 在F1中去掉依赖**左部多余**的属性。对于 $CE \rightarrow A$, 由于 $C \rightarrow A$, 故E是多余的; 对于 $ACD \rightarrow B$, 由于 $(CD)^+ = ABCDEG$, 故A多余。删除依赖左部多余的依赖后:

$$F2 = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, CD \rightarrow B, D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow B, CG \rightarrow D, CE \rightarrow G\}$$

3. 在F2中去掉**多余的依赖**。对 $CG \rightarrow B$, 由于 $(CG)^+ = ABCDEG$, 故 $CG \rightarrow B$ 是多余的。删除依赖左部多余的依赖后:

$$F3 = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, CD \rightarrow B, D \rightarrow E, D \rightarrow G, BE \rightarrow C, CG \rightarrow D, CE \rightarrow G\}.$$

$CG \rightarrow B$ 与 $CD \rightarrow B$ 不能同时存在, 但去掉任何一个都可以, 说明最小依赖集不唯一。

课堂练习(六)

- 已知 $R\langle U, F \rangle$, $U=\{A, B, C, D, E, F, G, H, I, J\}$, $F=\{ABC \rightarrow E, AB \rightarrow G, B \rightarrow F, C \rightarrow J, CJ \rightarrow I, G \rightarrow H\}$, 求 $R\langle U, F \rangle$ 的最小函数依赖集。

大纲

- 问题的提出
- 规范化
- 数据依赖的公理系统
- **模式分解**
- 本章小结

模式分解

- 定义6.16 (模式分解的定义)

- 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 $U = \bigcup_{i=1}^n U_i$ ，且没有 $U_i \subseteq U_j$ ， $1 \leq i, j \leq n$ ， F_i 是 F 在 U_i 上的投影。

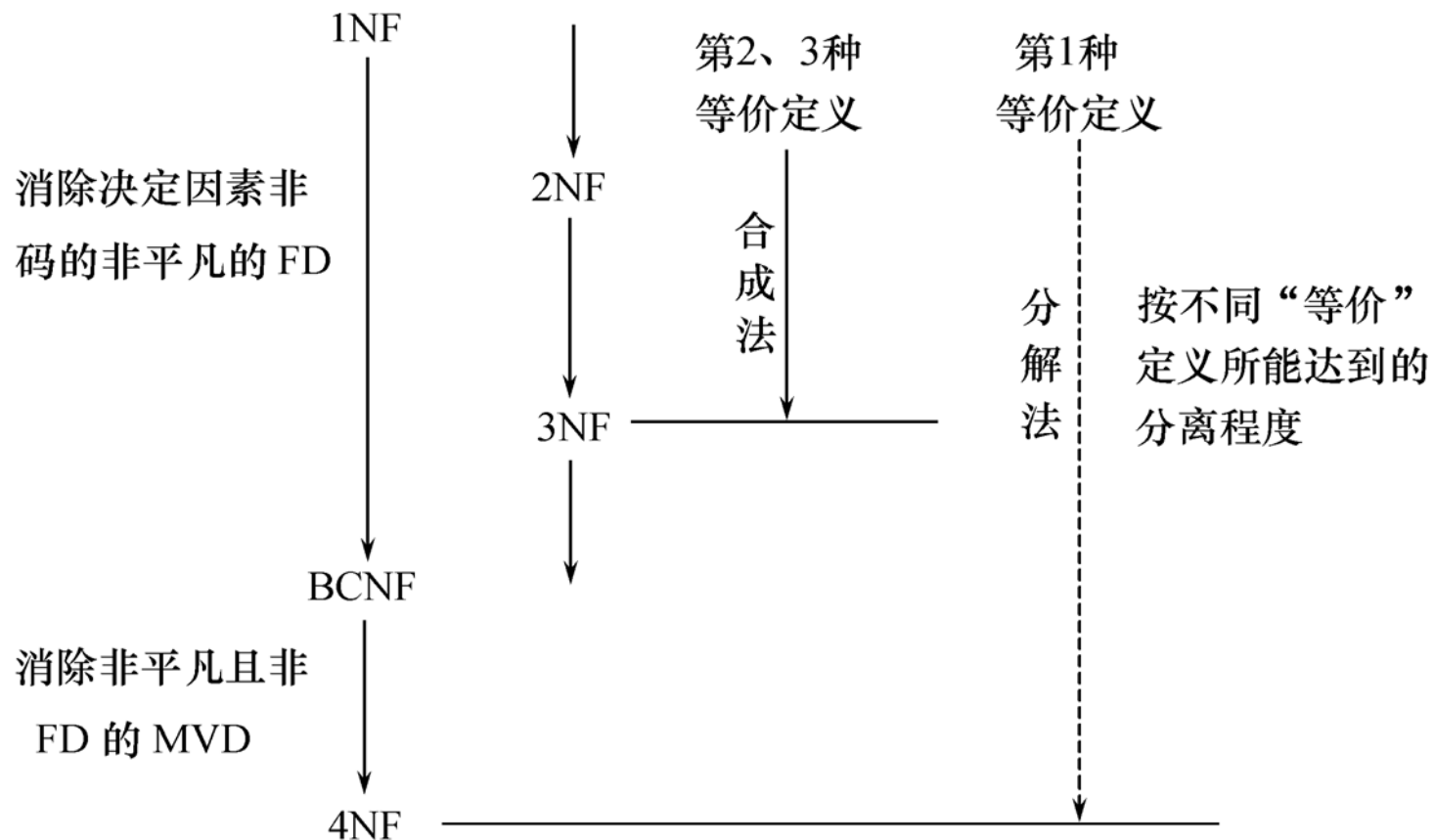
- 定义6.17

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 叫做 F 在 U_i 上的投影。

- 实行分解的三条不同准则：
 - 无损连接性(lossless join)
 - 保持函数依赖(preserve functional dependency)
 - 既要保持函数依赖，又要具有无损连接性
- 按照不同的分解准则，模式所能达到的分离程度各不相同，各种范式就是对分离程度的测度。
- 对于一个模式的分解是多种多样的，但分解后产生的模式应该与原模式等价。

本章小结

■ 关系模式的规范化，其基本思想



- 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。
- 规范化理论为数据库设计提供理论的指南和工具。
 - 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

本章作业

- 教材第六章习题：1, 2, 6.