数据库系统概论
An Introduction to Database System

第六章　关系数据理论

中国人民大学信息学院

第二篇　设计与应用开发篇

- 基于某个数据库管理系统设计数据库，如何基于数据库系统编程
  - 第6章 关系数据理论
  - 第7章 数据库设计
  - 第8章 数据库编程

第六章 关系数据理论

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

An Introduction to Database System

问题的提出（续）

- 关系模式由五部分组成，是一个五元组：

  R(U, D, DOM, F)

  - 关系名R是符号化的元组语义
  - U为一组属性
  - D为属性组U中的属性所来自的域
  - DOM为属性到域的映射
  - F为属性组U上的一组数据依赖

问题的提出（续）

- 由于D、DOM与模式设计关系不大，因此在本章中把关系模式看作一个三元组：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
- 作为二维表，关系要符合一个最基本的条件：每个分量必须是不可分开的数据项。满足了这个条件的关系模式就属于第一范式（1NF）

问题的提出（续）

- 数据依赖
  - 是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出来的数据间相互联系
  - 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

问题的提出（续）

- 数据依赖的主要类型
  - 函数依赖（Functional Dependency，简记为FD）
  - 多值依赖（Multi-Valued Dependency，简记为MVD）

问题的提出（续）

- 函数依赖普遍存在于现实生活中
  - 描述一个学生关系，可以有学号、姓名、系名等属性。
    - 一个学号只对应一个学生，一个学生只在一个系中学习
    - “学号”值确定后，学生的姓名及所在系的值就被唯一确定。
  - **Sname=f(Sno)，Sdept=f(Sno)**
    - **即Sno函数决定Sname**
    - **Sno函数决定Sdept**
    - **记作Sno→Sname，Sno→Sdept**

问题的提出（续）

- [例6.1] 建立一个描述学校教务的数据库。

  涉及的对象包括：

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

问题的提出（续）

- 假设学校教务的数据库模式用一个单一的关系模式Student来表示，则该关系模式的属性集合为：

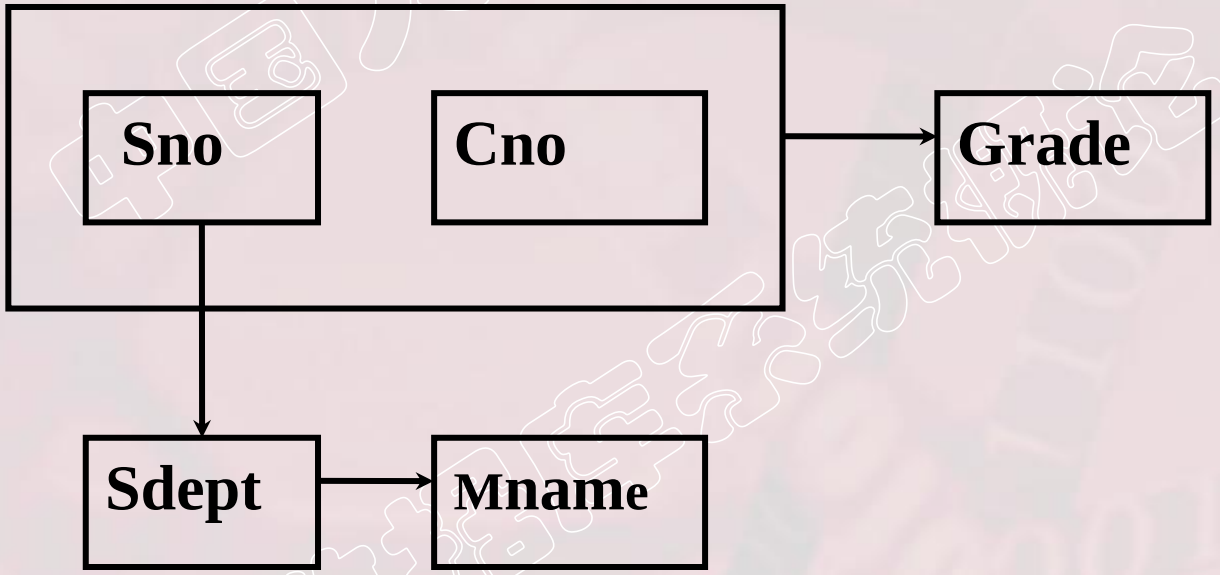
  U ＝{Sno, Sdept, Mname, Cno, Grade}

- 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。

问题的提出（续）

- 由此可得到属性组U上的一组函数依赖F：

F={Sno→Sdept, Sdept→ Mname, (Sno, Cno)→ Grade}



问题的提出（续）

关系模式Student<U, F>中存在的问题：

（1）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。

问题的提出（续）

（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。
  - 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。

问题的提出（续）

（3）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。

问题的提出（续）

（4）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。

问题的提出（续）

- 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
  - 由存在于模式中的某些数据依赖引起的。
- 解决方法
  - 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

问题的提出（续）

- 把这个单一的模式分成三个关系模式：
  - S(Sno,Sdept,Sno → Sdept);
  - SC(Sno,Cno,Grade,(Sno,Cno) → Grade);
  - DEPT(Sdept,Mname,Sdept → Mname);
- 这三个模式都不会发生插入异常、删除异常的问题，数据的冗余也得到了控制。

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

6.2.1 函数依赖

1.函数依赖

2.平凡函数依赖与非平凡函数依赖

3.完全函数依赖与部分函数依赖

4.传递函数依赖

1. 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*→*Y*。

函数依赖（续）

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

假设不允许重名，则有:

Sno → Ssex，　　Sno → Sage

Sno → Sdept，　　Sno ←→ Sname

Sname → Ssex，　Sname → Sage

Sname → Sdept

但Ssex ↛Sage, Ssex↛ Sdept

若X→Y，并且Y→X, 则记为X←→Y。
若Y不函数依赖于X, 则记为X↛Y。

函数依赖（续）

违背了Sno → Sname

Sno	Sname			Sdept
S1	张三		20	计算机系
S1	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

函数依赖（续）

❖ 由下面的关系表，能否得出**Sno → Sname**

Sno	Sname	Ssex	Sage	Sdept
S1	张三	男	20	计算机系
S2	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

函数依赖不是指关系模式R的某个或某些关系实例满足的约束条件，而是指R的所有关系实例均要满足的约束条件。

函数依赖（续）

- 函数依赖是语义范畴的概念，只能根据数据的语义来确定一个函数依赖。
  - 例如“姓名→年龄”这个函数依赖只有在不允许有同名人的条件下成立

2. 平凡函数依赖与非平凡函数依赖

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

对于任一关系模式，平凡函数依赖都是必然成立的，它不反映新的语义。
若不特别声明， 我们总是讨论非平凡函数依赖。

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因素（Determinant）。
- 若$X \rightarrow Y$，$Y \rightarrow X$，则记作$X \longleftrightarrow Y$。
- 若Y不函数依赖于X，则记作$X \nrightarrow Y$。

3. 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] 在关系SC(Sno, Cno, Grade)中，有：

- 由于：Sno $\not\rightarrow$ Grade，Cno $\not\rightarrow$ Grade，

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

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

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

4. 传递函数依赖

❖ 定义6.3 在$R(U)$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(transitive functional dependency)。记为：$X \xrightarrow{传递} Z$。

- 注: 如果$Y \rightarrow X$, 即$X \longleftrightarrow Y$，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，有：

  Sno → Sdept，Sdept → 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$R<U,F>$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（Surpkey）。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)。

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6.2]S(Sno, Sdept, Sage)，单个属性Sno是码

SC(Sno, Cno, Grade)中，(Sno, Cno)是码

[例6.3] 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

码（续）

- 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。
  - **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

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)

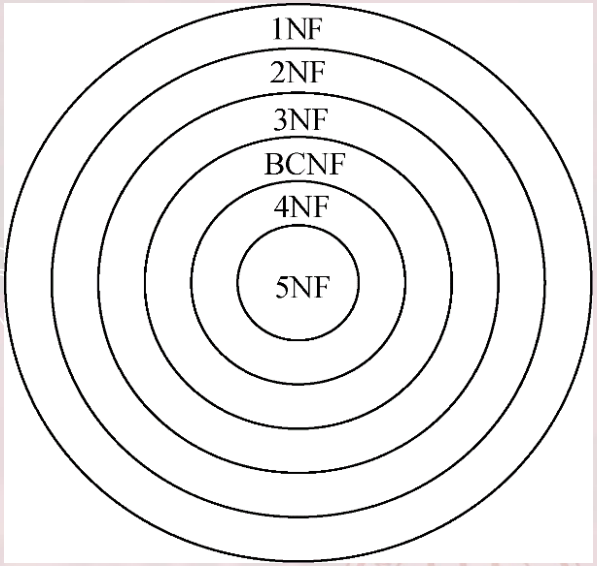
范式（续）

- 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（schema decomposition）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）。



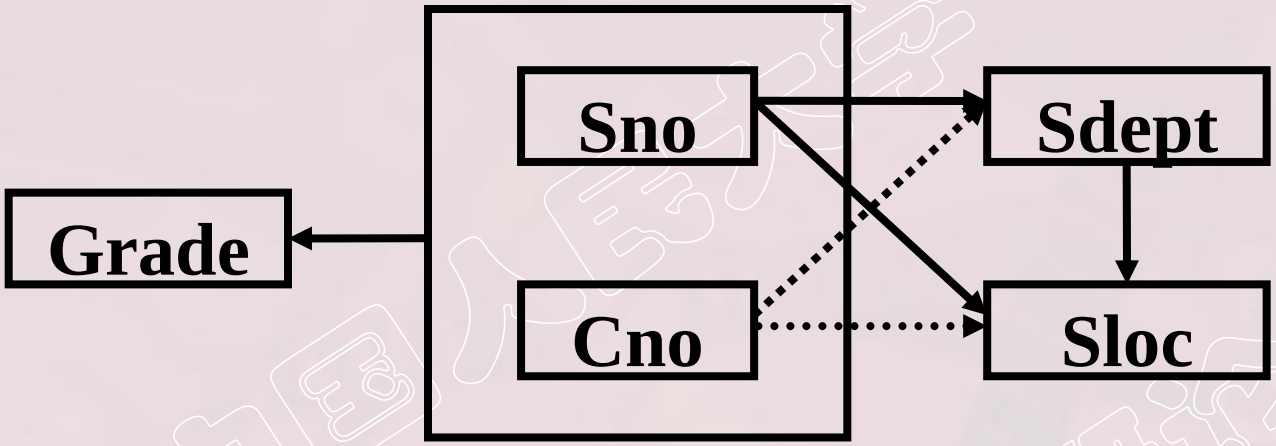
6.2 规范化

6.2.4 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
  - $(Sno,Cno) \xrightarrow{F} Grade$
  - $Sno \rightarrow Sdept$, $(Sno,Cno) \xrightarrow{P} Sdept$
  - $Sno \rightarrow Sloc$, $(Sno,Cno) \xrightarrow{P} Sloc$
  - $Sdept \rightarrow Sloc$

2NF（续）



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

2NF（续）

- 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
    - 如果一个学生选了多门课，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。

2NF（续）

- 出现这种问题的原因
  - 例子中有两类非主属性：
    - 一类如Grade，它对码完全函数依赖
    - 另一类如Sdept、Sloc，它们对码不是完全函数依赖
- 解决方法：
  - 用投影分解把关系模式S-L-C分解成两个关系模式
    - SC(Sno,Cno,Grade)
    - S-L(Sno,Sdept,Sloc)

2NF（续）

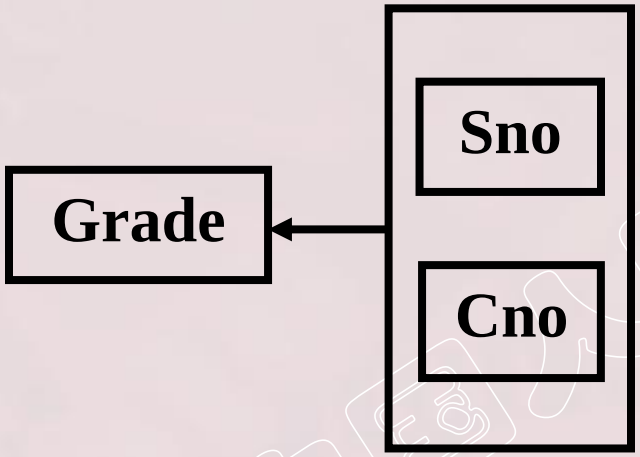


图6.4 SC中的函数依赖

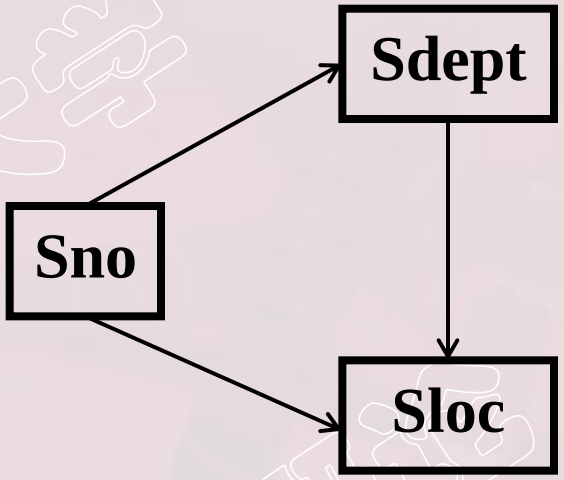


图6.5 S-L中的函数依赖

- SC的码为(Sno,Cno),SL的码为Sno，这样使得非主属性对码都是完全函数依赖了

6.2 规范化

6.2.5 3NF

- 定义6.7 设关系模式$R<U,F>\in$1NF,若R中不存在这样的码X、属性组Y及非主属性Z（$Z \supsetneq Y$），使得$X \rightarrow Y$，$Y \rightarrow Z$成立，$Y \nrightarrow X$不成立，则称$R<U,F> \in$ 3NF。
  - SC没有传递依赖，因此SC $\in$ 3NF
  - S-L中Sno $\rightarrow$Sdept(Sdept $\nrightarrow$ Sno), Sdept$\rightarrow$Sloc，可得Sno $\overset{传递}{\Rightarrow}$ Sloc。
  - 解决的办法是将S-L分解成
    - S-D(Sno,Sdept)$\in$ 3NF
    - D-L(Sdept,Sloc)$\in$ 3NF

6.2 规范化

6.2.6 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<U,F>\in$1NF，若$X\rightarrow Y$且$Y\subseteq X$时X必含有码，则$R<U,F>\in$BCNF。
- 换言之，在关系模式R<U,F>中，如果每一个决定属性集都包含候选码，则R∈BCNF。

BCNF（续）

- 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
- 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。

BCNF（续）

- [例6.5]考察关系模式C(Cno,Cname,Pcno)
  - 它只有一个码Cno，没有任何属性对Cno部分依赖或传递依赖，所以C∈3NF。
  - 同时C中Cno是唯一的决定因素，所以C∈BCNF。
  - 对于关系模式SC(Sno,Cno,Grade)可作同样分析。

BCNF（续）

- [例6.6] 关系模式S(Sno,Sname,Sdept,Sage)，
  - 假定Sname也具有唯一性，那么S就有两个码，这两个码都由单个属性组成，彼此不相交。
  - 其他属性不存在对码的传递依赖与部分依赖，所以S∈3NF。
  - 同时S中除Sno，Sname外没有其他决定因素，所以S也属于BCNF。

BCNF（续）

- [例6.7] 关系模式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)→P；(J,P)→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∈3NF。
  - 除(S,J)与(J,P)以外没有其他决定因素，所以SJP∈BCNF。

BCNF（续）

❖[例6.8] 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。

- 由语义可得到函数依赖：(S,J)→T；(S,T)→J；T→J
- 因为没有任何非主属性对码传递依赖或部分依赖，STJ ∈ 3NF。
- 因为T是决定因素，而T不包含码，所以STJ ∉ BCNF关系。

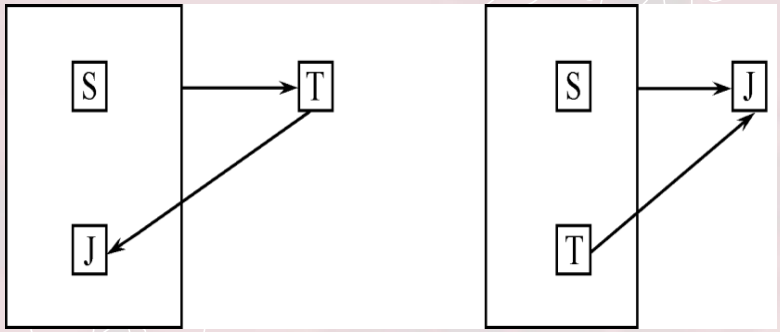


图6.6 STJ中的函数依赖

BCNF（续）

- 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。

BCNF（续）

- 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。

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[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

用关系模式Teaching(C,T,B)来表示课程C、教师T和参考书B之间的关系。

多值依赖（续）

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 …
…	…	…

多值依赖（续）

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
…	…	…

多值依赖（续）

- Teaching具有唯一候选码(C,T,B)， 即全码。
- Teaching∈BCNF

多值依赖（续）

课程 C	教员 T	参考书
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
…	…	…

(1)数据冗余度大：有多少名任课教师，参考书就要存储多少次。

多值依赖（续）

课程 C	教员 T	参考
物 理	李 勇	普通物
物 理	李 勇	光学
物 理	李 勇	物理习
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
…	…	…

(2)增加操作复杂：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。

多值依赖（续）

课程 C	教员 T	参考
物 理	李 勇	普通物
物 理	李 勇	光学
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
…	…	…

(3)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。

多值依赖（续）

课程 C	教员 T	参考
物 理	李 勇	普通
物 理	李 勇	光学
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

(4)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。

产生原因：存在多值依赖

多值依赖（续）

- 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*→→*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→→T。

多值依赖（续）

- 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$X \rightarrow\rightarrow Y$。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→→Y，而Z＝Φ，即Z为空，则称X→→Y为平凡的多值依赖。
  - 否则称X→→Y为非平凡的多值依赖。

多值依赖（续）

[例6.10]关系模式WSC(W,S,C)中，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

多值依赖（续）

- 按照语义对于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→→S。
- 如图6.7所示
  - 对应W的某一个值W_i的全部S值记作$\{S\}_{Wi}$（表示此仓库工作的全部保管员）
  - 全部C值记作$\{C\}_{Wi}$（表示在此仓库中存放的所有商品）
  - 应当有$\{S\}_{Wi}$中的每一个值和$\{C\}_{Wi}$中的每一个C值对应
  - 于是$\{S\}_{Wi}$与$\{C\}_{Wi}$之间正好形成一个完全二分图，因而W→→S。

多值依赖（续）

- 由于C与S的完全对称性，必然有W→→C成立。

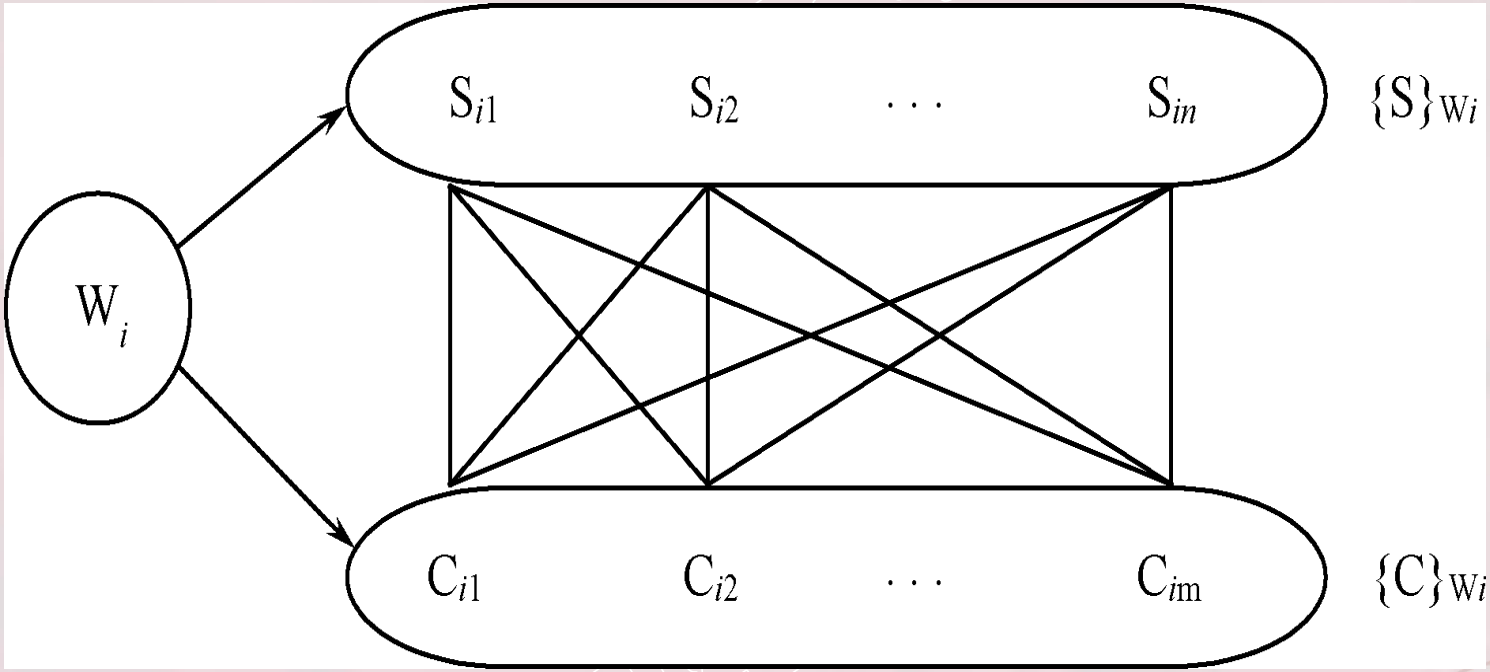


图6.7 W→→S且W→→C

多值依赖（续）

❖多值依赖的性质

（1）多值依赖具有对称性。

即若$X \rightarrow\rightarrow Y$，则$X \rightarrow\rightarrow Z$，其中$Z = U - X - Y$

- 多值依赖的对称性可以用完全二分图直观地表示出来。
- 从[例6.10] 容易看出，因为每个保管员保管所有商品，同时每种商品被所有保管员保管，显然若$W \rightarrow\rightarrow S$，必然有$W \rightarrow\rightarrow C$。

多值依赖（续）

（2）多值依赖具有传递性。即若$X \rightarrow\rightarrow Y$，$Y \rightarrow\rightarrow Z$，则$X \rightarrow\rightarrow Z - Y$。

（3）函数依赖是多值依赖的特殊情况。即若$X \rightarrow Y$，则$X \rightarrow\rightarrow Y$。

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

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

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

多值依赖（续）

- 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

  - 若$X \rightarrow\rightarrow Y$在$U$上成立，则在$W$（$XY \subseteq W \subseteq U$）上一定成立；反之则不然，即$X \rightarrow\rightarrow Y$在$W$（$W \subset U$）上成立，在$U$上并不一定成立。
  - 原因：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。

多值依赖（续）

- 多值依赖的有效性与属性集的范围有关（续）
  - 一般地，在$R(U)$上若有$X \rightarrow\rightarrow Y$在$W(W \subset U)$上成立，则称$X \rightarrow\rightarrow Y$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.I，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（续）

（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$X \rightarrow\rightarrow Y$若在$R(U)$上成立，不能断言对于任何$Y' \subset Y$有$X \rightarrow\rightarrow Y'$成立。

多值依赖（续）

例如，关系R(A,B,C,D)，A→→BC成立，当然也有A→→D成立。有R的一个关系实例，在此实例上A→→B是不成立的。

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

6.2 规范化

6.2.8 4NF

- 定义6.10 关系模式$R<U,F>\in$1NF，如果对于R的每个非平凡多值依赖$X\rightarrow\rightarrow Y$（$Y\not\subseteq X$），$X$都含有码，则$R<U,F>\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。4NF所允许的非平凡多值依赖实际上是函数依赖。

4NF（续）

- 如果一个关系模式是4NF， 则必为BCNF。
- 在[例6.10]的WSC中，W →→S, W→→C,他们都是非平凡多值依赖。而W不是码，关系模式WSC的码是(W,S,C)，即All-key，因此WSC ∉ 4NF。
- 可以把WSC分解成WS(W,S),WC(W,C)，WS∈4NF，WC∈4NF。

6.2 规范化

6.2.9 规范化小结

- 在关系数据库中，对关系模式的基本要求是满足第一范式。
- 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
  - 解决方法就是对其进行规范化，转换成高级范式。

规范化小结（续）

- 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。

规范化小结（续）

- 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”。
  - 即采用“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。
  - 因此 规范化实质上是概念的单一化。

规范化小结（续）

图6.8 规范化过程

规范化小结（续）

- 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。

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

❖ 定义6.11 对于满足一组函数依赖*F*的关系模式 *R* <*U*,*F*>，其任何一个关系*r*，若函数依赖*X*→*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* →*Y*。

数据依赖的公理系统（续）

- Armstrong公理系统
  - 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涵的函数依赖

数据依赖的公理系统（续）

- **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$R<U,F>$。对$R<U,F>$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。

数据依赖的公理系统（续）

- 定理6.1 Armstrong推理规则是正确的。
- 证明
  - A1 自反律

    设$Y \subseteq X \subseteq U$。

    对$R<U,F>$的任一关系r中的任意两个元组t、s：

    若$t[X]=s[X]$，由于$Y \subseteq X$，有$t[Y]=s[Y]$，

    所以$X \rightarrow Y$成立，

    自反律得证。

数据依赖的公理系统（续）

- **A2 增广律**

  设$X \rightarrow Y$为F所蕴涵，且$Z \subseteq U$。

  对$R<U,F>$ 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$t[YZ]=s[YZ]$，$XZ \rightarrow YZ$为F所蕴涵，

  增广律得证。

数据依赖的公理系统（续）

- **A3 传递律**

  设$X \rightarrow Y$及$Y \rightarrow Z$为F所蕴涵。

  对$R<U,F>$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所蕴涵，

  传递律得证。

数据依赖的公理系统（续）

- 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$X \rightarrow Z$。

数据依赖的公理系统（续）

❖根据合并规则和分解规则，可得引理6.1

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

数据依赖的公理系统（续）

- 定义6.12 在关系模式$R<U,F>$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^+。

- 定义6.13 设F为属性集U上的一组函数依赖，X、$Y \subseteq U$，$X_F^+=\{ A|X \rightarrow A$能由$F$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。

数据依赖的公理系统（续）

- 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$Y \subseteq X_F^+$。
  - 引理6.2的用途

    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^+的子集的问题。

数据依赖的公理系统（续）

- 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：

数据依赖的公

对$X^{(i)}$中的每个元素，依次检查相应的函数依赖,将依赖它的属性加入B

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。

数据依赖的公理系统（续）

[例6.11] 已知关系模式$R<U, F>$，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$(AB)_F^+$。

数据依赖的公理系统（续）

■ 解：由算法6.1，设$X^{(0)}=AB$。

计算$X^{(1)}$：逐一的扫描F集合中各个函数依赖，找左部为A、B或AB的函数依赖。得到两个：$AB\rightarrow C$，$B\rightarrow D$。于是$X^{(1)}=AB\cup CD=ABCD$。

因为$X^{(0)}\neq X^{(1)}$，所以再找出左部为$ABCD$子集的那些函数依赖，又得到$C\rightarrow E$，$AC\rightarrow B$，于是

$X^{(2)}=X^{(1)}\cup BE=ABCDE$。

因为$X^{(2)}$已等于全部属性集合，所以$(AB)_F^+=ABCDE$。

■ 参见爱课程网数据库系统概论6.3节动画《求闭包》

数据依赖的公理系统（续）

- 有效性与完备性的含义
  - 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

数据依赖的公理系统（续）

- 定理6.2 Armstrong公理系统是有效的、完备的。
- 证明：

1. 有效性

  - 有效性实际上是“正确性”
  - 可由定理6.1得证

数据依赖的公理系统（续）

2. 完备性

- 只需证明逆否命题：若函数依赖$X \rightarrow Y$不能由F从Armstrong公理导出，那么它必然不为F 所蕴涵
- 分三步证明：

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

证：因为 $V \subseteq X_F^+$，所以有$X \rightarrow V$成立；

因为$X \rightarrow V$，$V \rightarrow W$，于是$X \rightarrow W$ 成立；

所以$W \subseteq X_F^+$。

数据依赖的公理系统（续）

（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^+	$U-X_F^+$
11……1	00……0
11……1	11……1

若*r* 不是*R*<*U*,*F*> 的关系，则必由于*F*中有某一个函数依赖*V*→*W* 在*r*上 不成立所致。由*r* 的构成可知，*V* 必定是X_F^+ 的子集，而 *W* 不是 X_F^+ 的子集，可是由第（1）步，*W* ⊆ X_F^+，矛盾。所以*r* 必是*R*<*U*,*F*>的一个关系。

数据依赖的公理系统（续）

（3）若$X \rightarrow Y$不能由F从Armstrong公理导出，则Y不是X_F^+的子集。（引理6.2）

因此必有Y的子集Y' 满足$Y' \subseteq U - X_F^+$，

则$X \rightarrow Y$ 在r 中不成立，

即$X \rightarrow Y$ 必不为$R<U,F>$ 蕴涵。

数据依赖的公理系统（续）

- **Armstrong公理的完备性及有效性说明:**
  - “导出”与“蕴涵”是两个完全等价的概念
  - F^+：为F所逻辑蕴涵的函数依赖的全体（定义6.12）

  - F^+：可以说成由F出发借助Armstrong公理导出的函数依赖的集合

数据依赖的公理系统（续）

- 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。

两个函数依赖集等价是指它们的闭包等价

数据依赖的公理系统（续）

- 函数依赖集等价的充要条件
- 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。

引理6.3给出了判断两个函数依赖集等价的可行算法

（3）同理可证$G^+ \subseteq F^+$，所以$F^+ = G^+$。

如何判定$F \subseteq G^+$？

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

数据依赖的公理系统（续）

❖ 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[illegible]小依赖集或最小覆盖。

（1）*F*中任一函数依[illegible]右部仅含有一个属性。

（2）*F*中不存在这样的函数依赖*X*→*A*，使得*F*与*F*-{*X*→*A*}等价。

（3）*F*中不存在这样的函数依赖*X*→*A*，[illegible]有真子集*Z*使得*F*-{*X*→*A*}∪{*Z*→*A*}与*F*等价。

即*F*中的函数依赖均不能由*F*中其他函数依赖导出

*F*中各函数依赖左部均为最小属性集（不存在冗余属性）

数据依赖的公理系统（续）

- [例6.12] 考察6.1节中的关系模式*S*<*U*,*F*>，其中：

  U={Sno, Sdept, Mname, Cno, Grade}，

  F={Sno→Sdept, Sdept→Mname, (Sno,Cno)→Grade}

  *F*是最小覆盖

  F ' ={Sno→Sdept, Sno→Mname, Sdept→Mname, (Sno,Cno)→Grade, (Sno,Sdept)→Sdept}

  F '不是最小覆盖

  - 因为：*F* '- {Sno→Mname} 与 *F* '等价
  - *F* '- {(Sno,Sdept)→Sdept} 也与*F* '等价

- 参见爱课程网数据库系统概论6.2节动画《最小覆盖集难点解析》

数据依赖的公理系统（续）

❖ 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：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$Y=A_1A_2 \ldots A_k$，$k \geq 2$，

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

引理6.1保证了F变换前后的等价性。

数据依赖的公理系统（续）

（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。

由于F与G 等价的充要条件是$A \in X_G^+$

因此F变换前后是等价的。

数据依赖的公理系统（续）

（3）逐一取出F中各函数依赖FD_i：$X \rightarrow A$，

设$X=B_1B_2\ldots B_m$，$m \geq 2$，

逐一考查B_i（$i=1, 2, \ldots, m$），

若$A \in (X-B_i)_F^+$，则以$X-B_i$取代X。

由于F与$F-\{X \rightarrow A\} \cup \{Z \rightarrow A\}$等价的充要条件是$A \in Z_F^+$，其中$Z=X-B_i$，因此$F$变换前后是等价的。

最后剩下的F就一定是极小依赖集。

因为对F的每一次“改造”都保证了改造前后的两个函数依赖集等价，因此剩下的F与原来的F等价。

证毕

数据依赖的公理系统（续）

- 定理6.3的证明过程
  - 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

    若改造后的*F*与原来的*F*相同，说明*F*就是一个最小依赖集

数据依赖的公理系统（续）

❖[例6.13] $F=\{A\rightarrow B, B\rightarrow A, B\rightarrow C, A\rightarrow C, C\rightarrow A\}$

F的最小依赖集：

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

数据依赖的公理系统（续）

- 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。

数据依赖的公理系统（续）

[例6.13]（续）

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

F_{m1}、F_{m2}都是F的最小依赖集：

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

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

数据依赖的公理系统（续）

- 在$R<U,F>$中可以用与F等价的依赖集G来取代F
  - 原因：两个关系模式$R_1<U,F>$，$R_2<U,G>$，如果F与G等价，那么R_1的关系一定是R_2的关系。反过来，R_2的关系也一定是R_1的关系。

第六章 关系数据理论

6.1 问题的提出

6.2 规范化

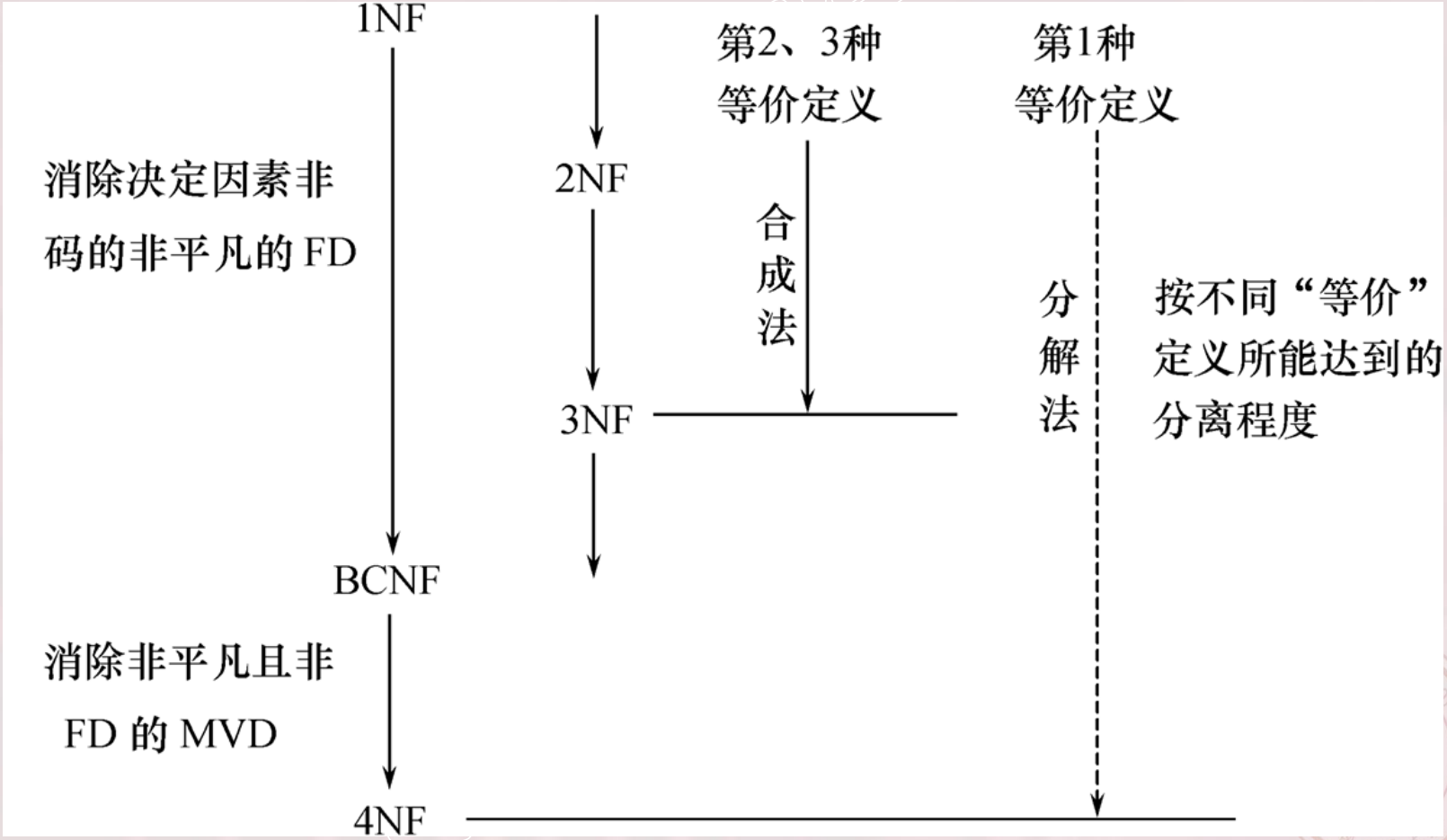
6.3 数据依赖的公理系统

*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