数据库系统原理

李瑞轩
华中科技大学计算机学院

第六章 关系数据理论

本章要点

- 问题的提出
- 函数依赖
- 范式与规范化
- 函数依赖的推理规则
- 模式分解

6.1 问题的提出

- 如何设计一个适合的关系数据库系统，关键是关系数据库模式的设计
  - 一个好的关系数据库模式应该包括多少关系模式
  - 每一个关系模式又应该包括哪些属性
  - 如何将这些相互关联的关系模式组建一个适合的关系模型
- 这些工作决定了整个系统运行的效率，也是系统成败的关键所在，所以必须在关系数据库的规范化理论的指导下逐步完成

问题的提出

- 规范化理论主要包括三个方面的内容：
  - 函数依赖（Functional Dependency）
  - 范式（Normal Form）
  - 模式设计
  - 其中，函数依赖起着核心的作用，是模式分解和模式设计的基础，范式是模式分解的标准。

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联（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] 建立一个描述学校教务的数据库：

学生学号（Sno），学生姓名（Sname），
学生年龄（Age），学生所在的系别（Dept），
系主任姓名（Mn），课程号（Cno），
成绩（Score）

单一的关系模式： Student <U, F>

U ＝｛ Sno, Sname, Age, Dept, Mn, Cno, Score｝

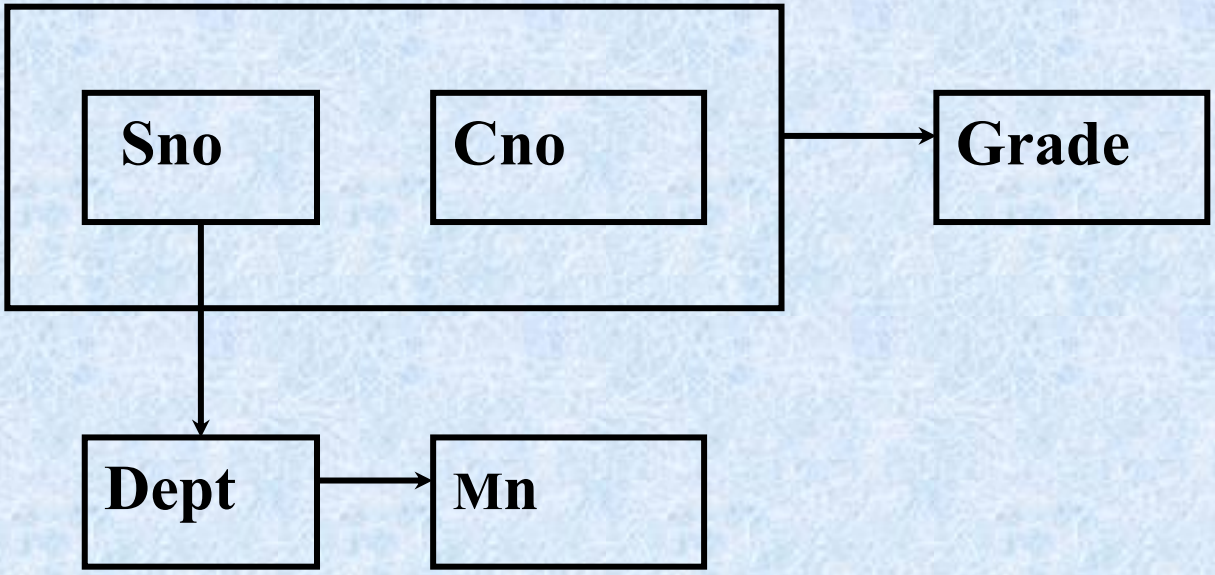
关系Student

学号 Sno	姓名 Sname	年龄 Age	系别 Dept	系主任 Mn	课程号 Cno	成绩 Score
S1	李勇	18	计算机	王平	C1	85
S1	李勇	18	计算机	王平	C2	82
S1	李勇	18	计算机	王平	C3	80
S2	刘晨	17	计算机	王平	C1	81
S2	刘晨	17	计算机	王平	C2	80
S2	刘晨	17	计算机	王平	C4	78
S3	王敏	18	自控系	刘伟	C2	87
S3	王敏	18	自控系	刘伟	C3	76
S4	张立	19	机械学院	李刚	C3	80
S4	张立	19	机械学院	李刚	C4	84

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 ＝ { Sno → Dept, Dept → Mn,
(Sno, Cno) → Grade }



关系Student

学号 Sno	姓名 Sname	年龄 Age	系别 Dept	系主任 Mn	课程号 Cno	成绩 Score
S1	李勇	18	计算机	王平	C1	85
S1	李勇	18	计算机	王平	C2	82
S1	李勇	18	计算机	王平	C3	80
S2	刘晨	17	计算机	王平	C1	81
S2	刘晨	17	计算机	王平	C2	80
S2	刘晨	17	计算机	王平	C4	78
S3	王敏	18	自控系	刘伟	C2	87
S3	王敏	18	自控系	刘伟	C3	76
S4	张立	19	机械学院	李刚	C3	80
S4	张立	19	机械学院	李刚	C4	84

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 → Sdept）；

  SC（Sno，Cno，Grade，（Sno，Cno）→ Grade）；

  DEPT（Sdept，Mname，Sdept→ Mname）

分解后的关系模式

学号 Sno	姓名 Sname	年龄 Age	系别 Dept
S1	李勇	18	计算机
S2	刘晨	17	计算机
S3	王敏	18	自控系
S4	张立	19	机械学院

系别 Dept	系主任 Mn
计算机	王平
自控系	刘伟
机械学院	李刚

学号 Sno	课程号 Cno	成绩 Score
S1	C1	85
S1	C2	82
S1	C3	80
S2	C1	81
S2	C2	80
S2	C4	78
S3	C2	87
S3	C3	76
S4	C3	80
S4	C4	84

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。

6.2.1 函数依赖

- **函数依赖**
  - 设R(U)是属性集U上的关系模式，X，Y ⊆ U， r是R(U) 上的任意一个关系，如果

对∀t，s ∈ r，若t[X] = s[X]，则t[Y] = s[Y] 成立，

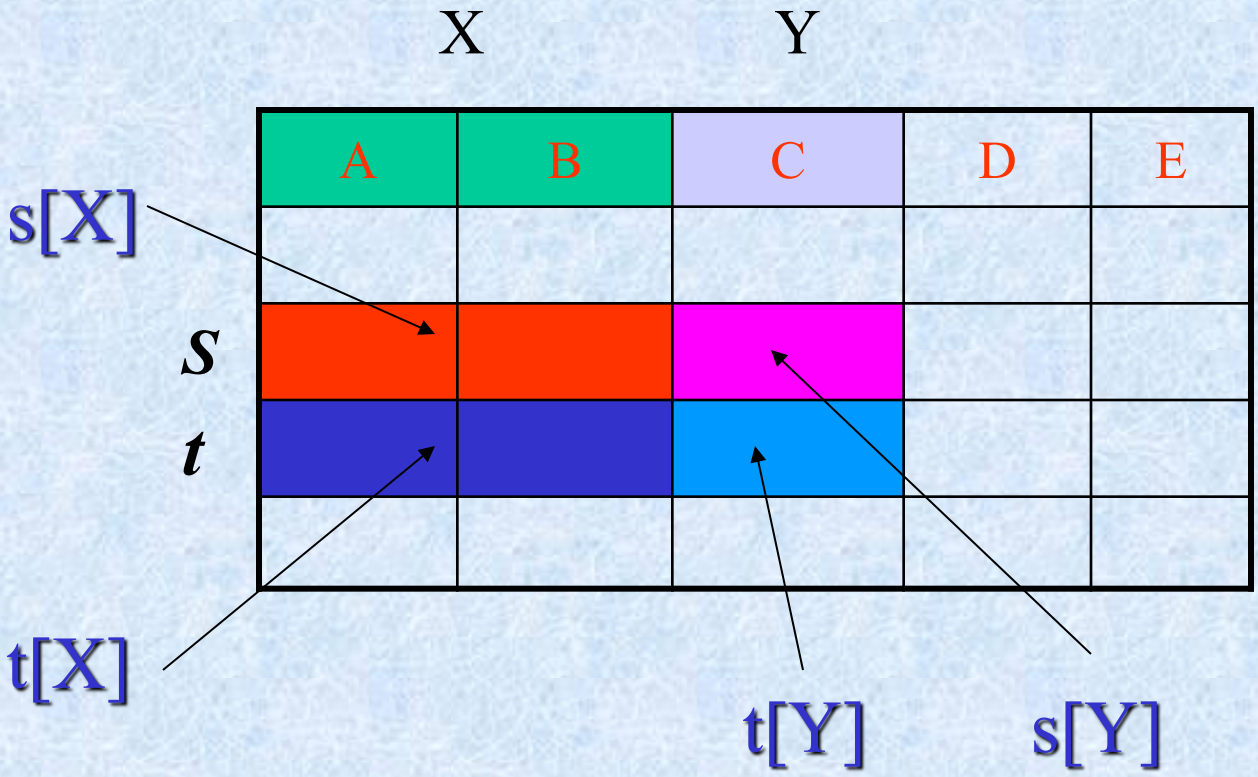
那么称“X函数决定Y”，或“Y函数依赖于X”，记作**X→Y**。

称X为决定因素。

如Sno → Sname，（Sno，Cno）→ Score

6.2.1 函数依赖

R(A,B,C,D,E):



只要s[X] = t[X] , 就有 s[Y] = t[Y]:
则 X→Y

6.2.1 函数依赖

学号 Sno	姓名 Sname	年龄 Age	系别 Dept	系主任 Mn	课程号 Cno	成绩 Score
S1	李勇	17	计算机	王平	C1	85
S1	李勇	17	计算机	王平	C2	82
S1	李勇	17	计算机	王平	C3	80
S2	刘晨	18	计算机	王平	C1	81
S2	刘晨	18	计算机	王平	C2	80
S2	刘晨	18	计算机	王平	C4	78
S3	王敏	17	自控系	刘伟	C2	87
S3	王敏	17	自控系	刘伟	C3	76
S4	张立	19	机械学院	李刚	C3	80
S4	张立	19	机械学院	李刚	C4	84

6.2.1 函数依赖

上述关系中的函数依赖：

学号→姓名
学号→年龄
学号→系别
学号→系主任
学号→(姓名,年龄)
学号→(姓名,系别)
学号→(姓名,系主任)
学号→(年龄,系别)
学号→(年龄,系主任)
学号→(系别,系主任)

学号→(姓名,年龄,系别)
学号→(姓名,年龄,系主任)
学号→ (姓名,系别,系主任)
学号→ (年龄,系别,系主任)
学号→ (姓名,年龄,系别,系主任)

系别→系主任

(学号,课程号)→成绩

6.2.1 函数依赖

- **平凡函数依赖**

  如果$X \rightarrow Y$，且$Y \subset X$，则称其为平凡的函数依赖。

  如（Sno，Sname）$\rightarrow$ Sname是平凡的函数依赖

- **非平凡函数依赖**

  如果Y不是X的子集，则称X→Y为非平凡的函数依赖。

- **若不特别声明，我们讨论的都是非平凡的函数依赖。**

6.2.1 函数依赖

有关函数依赖的几点说明：

1. 函数依赖是语义范畴的概念。

- 我们只能根据语义来确定一个函数依赖，而不能按照其形式化定义来证明一个函数依赖是否成立。
- 例如，对于关系模式S，当学生不存在重名的情况下，可以得到：

  SNAME→AGE

  SNAME→DEPT

- 这种函数依赖关系，必须是在没有重名的学生条件下才成立的，否则就不存在函数依赖了。
- 所以函数依赖反映了一种语义完整性约束。

6.2.1 函数依赖

2．函数依赖与属性之间的联系类型有关。

- （1）在一个关系模式中，如果属性X与Y有1:1联系时，则存在函数依赖X→Y，Y→X，即X←→Y。

  例如，当学生无重名时，SNO←→SN。

- （2）如果属性X与Y有m:1的联系时，则只存在函数依赖X→Y。

  例如，SNO与AGE、DEPT之间均为m:1联系，所以有SNO→AGE，SNO→DEPT。

- （3）如果属性X与Y有m: n的联系时，则X与Y之间不存在任何函数依赖关系。

  例如，一个学生可以选修多门课程，一门课程又可以为多个学生选修，所以SNO与CNO之间不存在函数依赖关系。

- 由于函数依赖与属性之间的联系类型有关，所以在确定属性间的函数依赖关系时，可以从分析属性间的联系类型入手，便可确定属性间的函数依赖。

6.2.1 函数依赖

3. 函数依赖关系的存在与时间无关。

- 因为函数依赖是指关系中的所有元组应该满足的约束条件，而不是指关系中某个或某些元组所满足的约束条件。
- 当关系中的元组增加、删除或更新后都不能破坏这种函数依赖。
- 因此，必须根据语义来确定属性之间的函数依赖，而不能单凭某一时刻关系中的实际数据值来判断。
- 函数依赖关系的存在与时间无关，而只与数据之间的语义规定有关。

6.2.1 函数依赖

- **完全函数依赖与部分函数依赖**

在R(U)中，如果$X \rightarrow Y$，且对于任意X的真子集X′，都有$X' \not\rightarrow Y$，则称Y对X完全函数依赖，记作

$$X \xrightarrow{f} Y$$

否则称为Y对X部分函数依赖，记作

$$X \xrightarrow{p} Y$$

例如：

$$(Sno, Cno) \xrightarrow{f} Score$$

$$(Sno, Cno) \xrightarrow{p} Sname$$

6.2.1 函数依赖

- **传递函数依赖**

在R(U)中，如果

$$X \rightarrow Y,\ Y \rightarrow Z,\ Y \nrightarrow X,\ \text{且} Y \not\subset X$$

则称Z对X传递函数依赖

例：Sno → Dept，Dept → Mn

存在传递依赖 Sno ⟶ Mn

6.2.1 函数依赖

- 由定义可知：
  - 只有当决定因素是组合属性时，讨论部分函数依赖才有意义。
  - 当决定因素是单属性时，只能是完全函数依赖。

例如，在关系模式S（SNO，SN，AGE，DEPT），决定因素为单属性SNO，有SNO $\xrightarrow{f}$ (SN，AGE，DEPT），不存在部分函数依赖。

6.2.1 函数依赖

关系模式*S(Sno, Sname, Dept, Mn , Cno , Score)*

主码：（Sno，Cno）

函数依赖：

(Sno，Cno) $\xrightarrow{f}$ Score

Sno → Sname，（Sno,Cno）$\xrightarrow{p}$ Sname

Sno → Dept，（Sno，Cno）$\xrightarrow{p}$ Dept

Dept → Mn （Sno，Cno）$\xrightarrow{\text{传递依赖}}$ Mn

6.2.2 码

- 候选码

  设K为R<U , F>的属性或属性组合，若K $\xrightarrow{F}$ U，则称K为R的候选码

- 主码

  若R<U , F>有多个候选码，则可以从中选定一个作为R的主码

- 主属性/非主属性

  包含在任何一个候选码中的属性，称作主属性

  不包含在任何一个候选码中的属性，称作非主属性

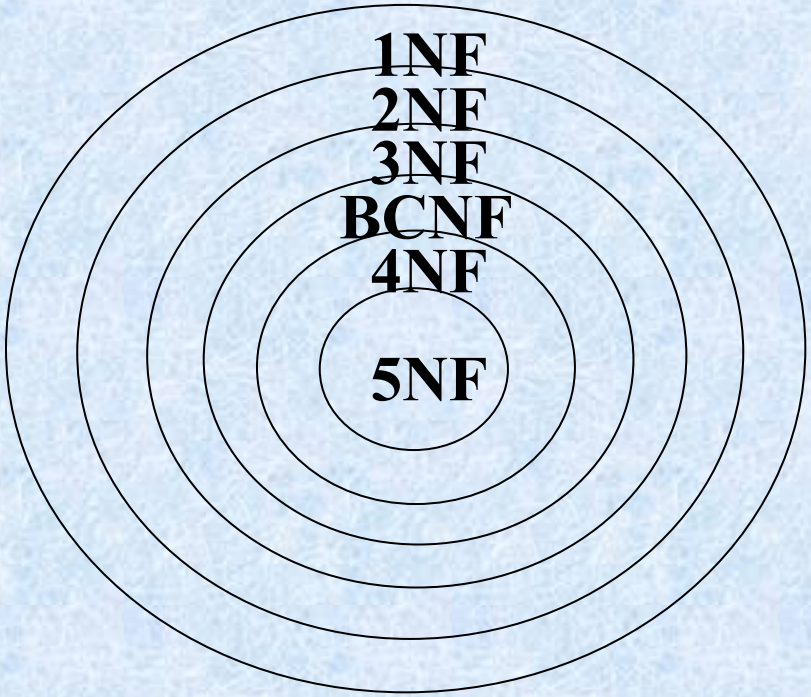
- 全码

  整个属性组是码

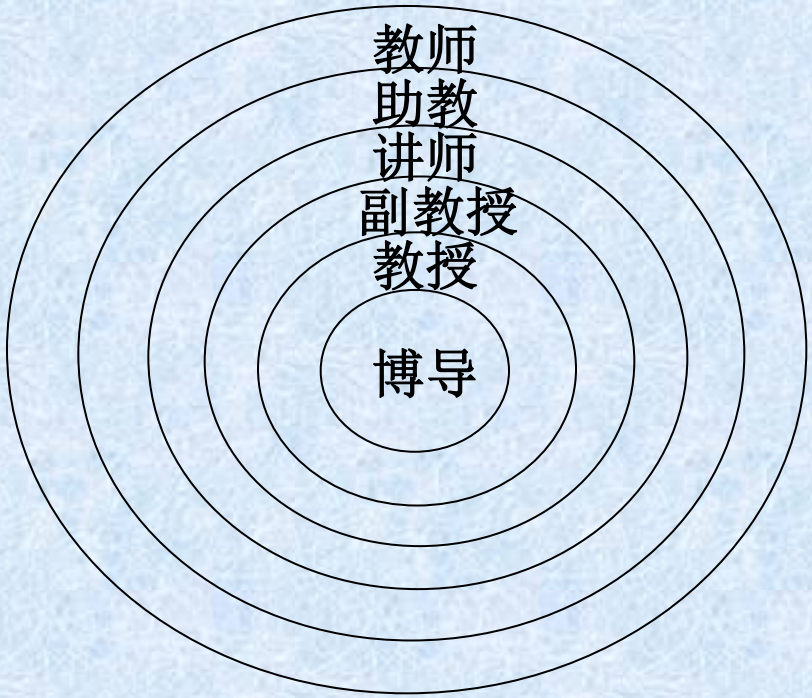
6.2.3 范 式

- 定义
  - 范式是对关系的不同数据依赖程度的要求。
  - 通过模式分解将一个低级范式转换为若干个高级范式的过程称作规范化（概念的纯粹化）。

范式关系图



现实世界



6.2.3 范式 1NF

- 定义：1NF

  关系中每一分量不可再分，即不能以集合、序列等作为属性值。

S#	C#
S1	{C1，C2，C3}

S#	C#
S1	C1
S1	C2
S1	C3

6.2.3 范式 1NF

关系模式SCD(*Sno , Sname , Dept , Mn, Cno , Score*)

- 不良特性
  - 插入异常：如果学生没有选课，关于他的个人信息及所在系的信息就无法插入
  - 删除异常：如果删除学生的选课信息，则有关他的个人信息及所在系的信息也随之删除了
  - 修改复杂：如果学生转系，若他选修了k门课，则需要修改k次
  - 数据冗余：如果一个学生选修了k门课，则有关他的所在系的信息重复k次

6.2.4 2NF

- 定义
  - 若R∈1NF，且每个非主属性完全依赖于码，则称R∈2NF。
  - 消除非主属性对码的部分依赖。

  如SCD∉2NF，因为

  (Sno，Cno) $\xrightarrow{p}$ Sname

  (Sno，Cno) $\xrightarrow{p}$ Dept

2NF (续)

- 改造

  非主属性有两种，一种完全依赖于码，一种部分依赖于码。

  将SCD分解为：

  SC(Sno , Cno , Score)　　完全依赖于码的部分

  S_SD(Sno , Sname , Dept , Mn)　部分依赖于码的部分

  即达到2NF。

2NF (续)

S_SD(Sno , Sname , Dept , Mn)

- 不良特性
  - 插入异常：如果系中没有学生，则有关“系”的信息就无法插入
  - 删除异常：如果学生全部毕业了，则在删除学生信息的同时有关“系”的信息也随之删除了
  - 修改复杂：如果学生转系，不但要修改Dept，还要修改Mn，如果换系主任，则该系每个学生元组都要做相应修改
  - 数据冗余：每个学生都存储了系主任的信息

6.2.5 3NF

- 定义
  - 关系模式R〈U , F〉中，若不存在这样的码X，属性组Y及非主属性Z(Z ⊄ Y)，使得下式成立，

$$X \rightarrow Y , Y \rightarrow Z , Y \not\rightarrow X$$

则称R∈3NF。

  - 消除非主属性对码的传递依赖。

如S_SD ∉3NF，因为有Sno→Dept，Dept→Mn

3NF(续)

- 改造

  将S_SD分解为：

  Student(Sno, Sname, Dept) 非传递依赖部分

  Dept(Dept, Mn) 传递依赖部分

  即达到3NF。

3NF(续)

- 示例

  STC(S# , T# , C#)

  T# → C#，每位老师只教授一门课

  （S#，T#）→ C#，某学生选定一位老师，就对应一门课

  （S#，C#）→ T#，某学生选定一门课，就对应一位老师

  （S#，C#）和 （S#，T#）为候选码

- 思考

  STC ∈ 3NF ？

3NF(续)

STC(S# , T# , C#)

- 不良特性
  - 插入异常：如果没有学生选修某位老师的任课，则该老师担任课程的信息就无法插入
  - 删除异常：删除学生选课信息，会删除掉老师的任课信息
  - 修改复杂：如果老师所教授的课程有所改动，则所有选修该老师课程的学生元组都要做改动
  - 数据冗余：每位学生都存储了有关老师所教授的课程的信息
- 症由：

  主属性对码的不良依赖

6.2.6 BCNF

- 定义
  - 关系模式R< U , F >中，对于属性组X，Y，若X→Y且Y $\not\subset$ X时X必含有码，则R< U , F > $\in$ BCNF
  - 等价于：每一个决定属性因素都包含码

    如STC $\notin$ BCNF，因为T# → C#，而T不含有码
- 改造

  将S分解为（S#，C#），（T#，C#）

BCNF(续)

- 若R∈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

BCNF(续)

- 思考

  SCO(S# , C# , ORDER)，表示学生选修课程的名次，有函数依赖(S#，C#)→ ORDER，(C#，ORDER) → S#，它属于BCNF吗？

解答:

- (S#, C#) 或 (C#, ORDER)都可以作为候选码
- 不存在非主属性对码的传递依赖或部分依赖，SCO∈3NF
- 没有其他决定因素，SCO ∈BCNF

BCNF与3NF的关系

- R ∈BCNF ⇄ R ∈3NF（充分；不必要）

- 如果R∈3NF，且R只有一个候选码

R ∈BCNF ⇄ R ∈3NF（充分；必要）

关于BCNF与3NF的定理

定理：如果R∈BCNF，则R∈3NF

证明：（反证法）设R∈BCNF，但R∉3NF,则总可找到属性集X，Y，Z，其中X为候选码，Y，Z为非主属性（R∉3NF，则存在多个非主属性，它们存在传递依赖），使得X→Z成立，即X→Y，而Y不包含R的候选码X，但Y→Z成立（Y是非主属性，这样决定因素不含候选码，违反BCNF定义）。

根据BCNF定义，R∉BCNF，与假设矛盾。

定理得证。

关系模式R<U，F>中，若不存在这样的码X，属性组Y及非主属性Z(Z⊄Y)，使得下式成立，X→Y，Y→Z，Y↛X，则称R∈3NF。

6.2.7 多值依赖

先看一个例子：

例 Teaching(C, T, B)(其中C：课程名，T:教师名，B:参考书名)

非规范化的关系：

C	T	B
数学	邓军	数学分析
	陈斯	高等代数
		微分方程
物理	李平	普通物理学
	王强	光学原理
	刘明	

C	T	B
数学	邓军	数学分析
数学	邓军	高等代数
数学	邓军	微分方程
数学	陈斯	数学分析
数学	陈斯	高等代数
数学	陈斯	微分方程
物理	李平	普通物理学
物理	李平	光学原理
物理	王强	普通物理学
物理	王强	光学原理
物理	刘明	普通物理学
物理	刘明	光学原理
……	……	……

⇦ BCNF? 问题？

(C, T, B)是码
全码→BCNF

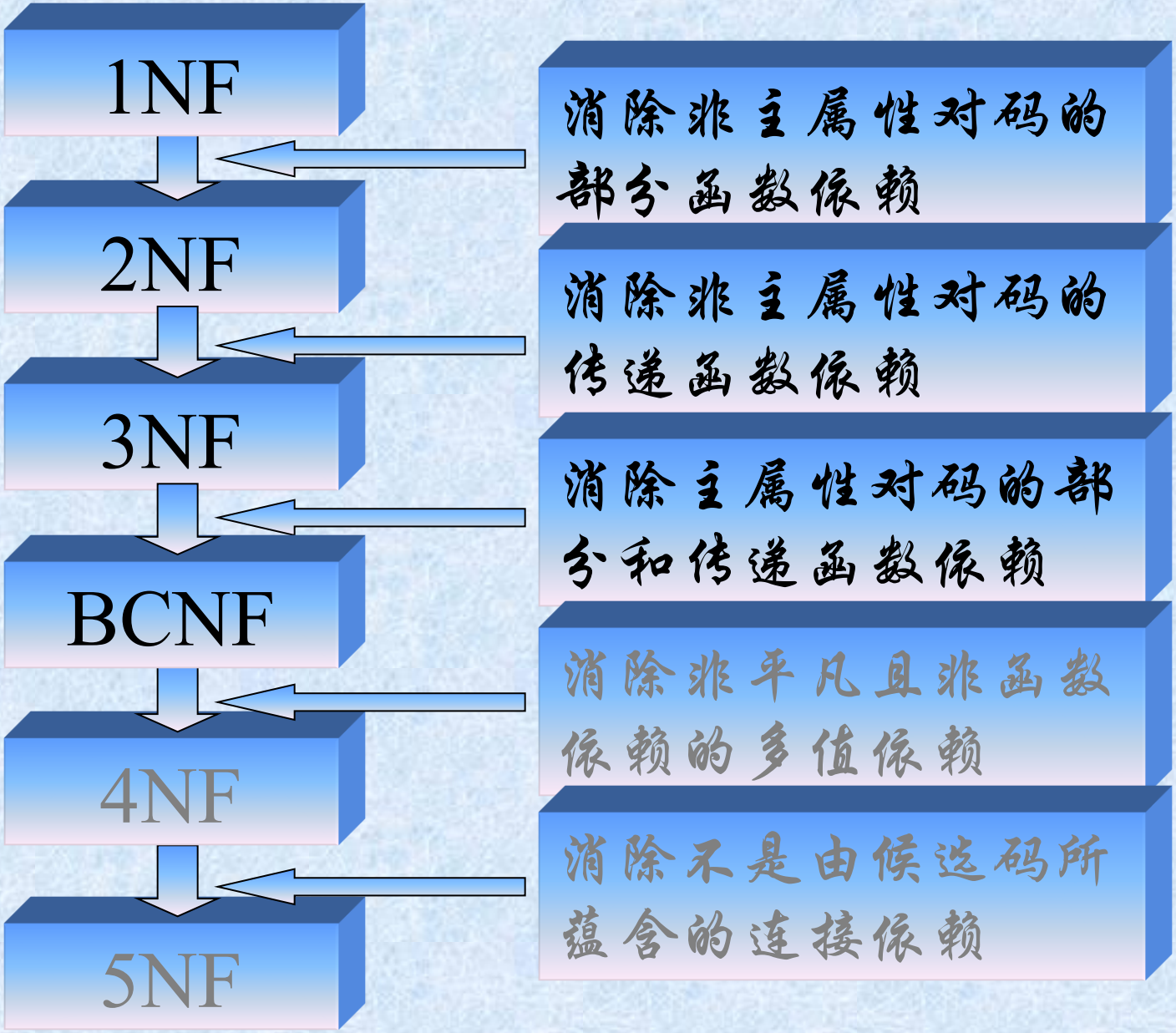
4NF & 5NF

- BCNF可能存在多值依赖
  - 4NF：消除非平凡且非函数依赖的多值依赖
- 4NF可能存在连接依赖
  - 5NF：消除不是由候选码所蕴含的连接依赖
- 范式之间的关系
  - 5NF $\subset$ 4NF $\subset$ BCNF $\subset$ 3NF $\subset$ 2NF $\subset$ 1NF

关系规范化小结

- 关系模式规范化的思想
  - 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
  - 概念的逐步单一化，采用投影的方法
  - 关系规范化在实际应用中应适可而止

关系规范化的步骤


1NF
2NF
3NF
BCNF
4NF
5NF
消除非主属性对码的部分函数依赖
消除非主属性对码的传递函数依赖
消除主属性对码的部分和传递函数依赖
消除非平凡且非函数依赖的多值依赖
消除不是由候选码所蕴含的连接依赖

关系规范化结论

1）3NF必定为2NF和1NF，反之不一定；

2）BCNF必为3NF，反之不一定；

3）3NF已在很大程度上控制了数据冗余；

4）3NF已在很大程度上消去了插入和删除操作异常；

5）3NF分解仍不够彻底（可能存在主属性对候选码的部分依赖和传递依赖）；

6）在函数依赖范围内，BCNF已完全消去了插入和删除异常；

7）范式并非越高越好；范式越高，异常越少，但查询操作越麻烦；

8）关系规范化应适可而止：

理论上：一般到3NF

应用上：连接开销。

9）分解不唯一。

练　习

- Ex 1: 关系模式中，满足2NF的模式，_____。

  A. 可能是1NF　　C. 必定是1NF

  B. 必定是3NF　　D. 必定是BCNF

  解答： C

- Ex 2: 设有如图所示的关系R，它是____。

  A. 1NF

  B. 2NF

  C. 3NF

  D. BCNF

  解答： B

材料号	材料名	生产厂
M1	线材	武汉
M2	型材	武汉
M3	板材	广东
M4	型材	武汉

练　习

Ex3：指出下列关系模式是第几范式？

(1) R(X,Y,Z)　F={XY⟶Z}　　BCNF

(2) R(X,Y,Z)　F={Y⟶Z, XZ⟶Y}　　3NF

(3) R(X,Y,Z)　F={Y⟶Z, Y⟶X, X⟶YZ}　　BCNF

(4) R(X,Y,Z)　F={X⟶Y, X⟶Z}　　BCNF

(5) R(W,X,Y,Z)　F={X⟶Z, WX⟶Y}　　1NF

6.3 数据依赖的公理系统

- 逻辑蕴涵
- Armstrong公理系统
- 闭包的计算
- 函数依赖的等价和覆盖

逻辑蕴涵

- 定义
  - 关系模式R，F是其函数依赖，X，Y是其属性子集，如果从F的函数依赖能够推出X→Y，则称F **逻辑蕴涵**X→Y，记作$F \vdash X \rightarrow Y$
  - 被F所逻辑蕴涵的函数依赖的全体所构成的集合称作F的**闭包**，记作$F^+ = \{X \rightarrow Y \mid F \vdash X \rightarrow Y\}$

Armstrong公理系统

- Armstrong公理

  U为属性集，F是U上的一组函数依赖，对关系模式R<U, F>, X，Y，Z是属性集

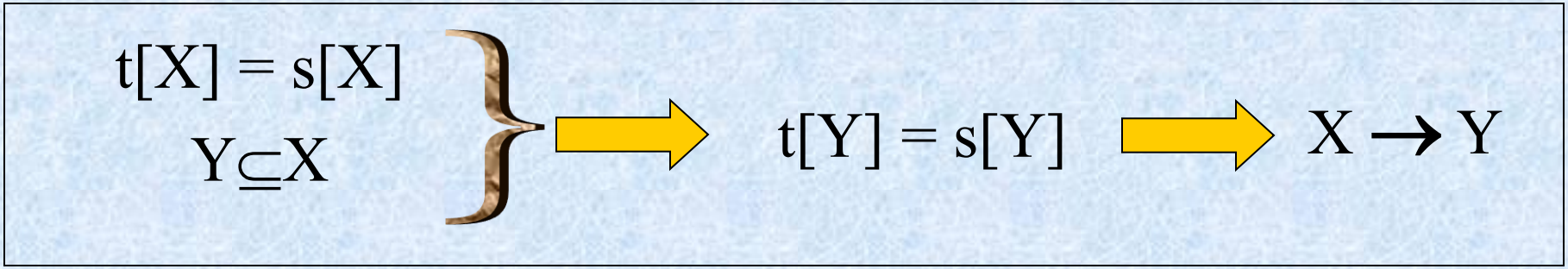
  - 自反律(reflexivity)：若$Y \subseteq X$， 则$X \rightarrow Y$
  - 增广律(augmentation)：若$X \rightarrow Y$，则$XZ \rightarrow YZ$
  - 传递律(transitivity)：若$X \rightarrow Y$，$Y \rightarrow Z$，则$X \rightarrow Z$

Armstrong公理系统

- 关于Armstrong公理正确性

按函数依赖定义，r是R<U, F>上的任一关系，t，s ∈ r

自反律



Armstrong公理系统

增广律

$t[XZ] = s[XZ] \Rightarrow t[X] = s[X]$

$X \rightarrow Y$

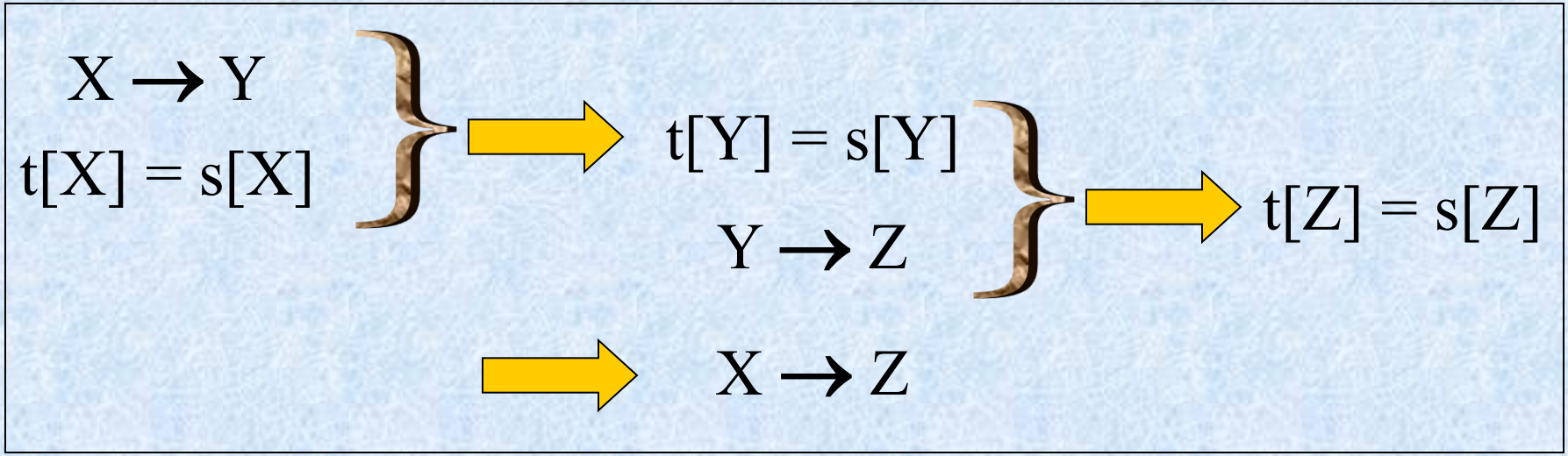
$\Rightarrow t[Y] = s[Y]$

$t[XZ] = s[XZ] \Rightarrow t[Z] = s[Z]$

$\Rightarrow t[YZ] = s[YZ] \Rightarrow XZ \rightarrow YZ$

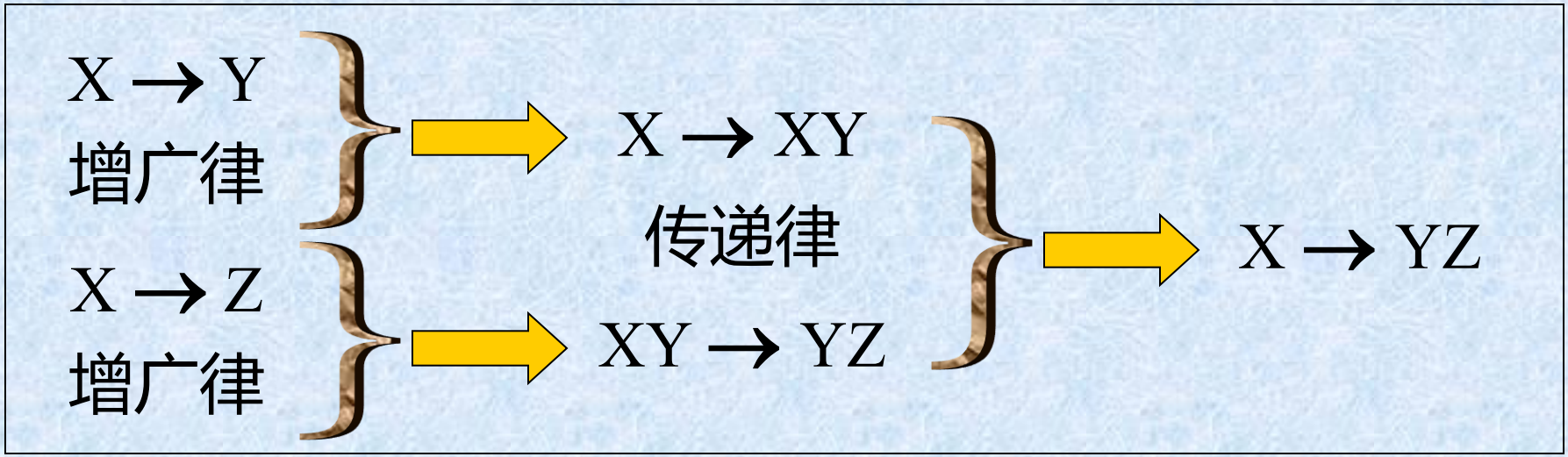
Armstrong公理系统

传递律



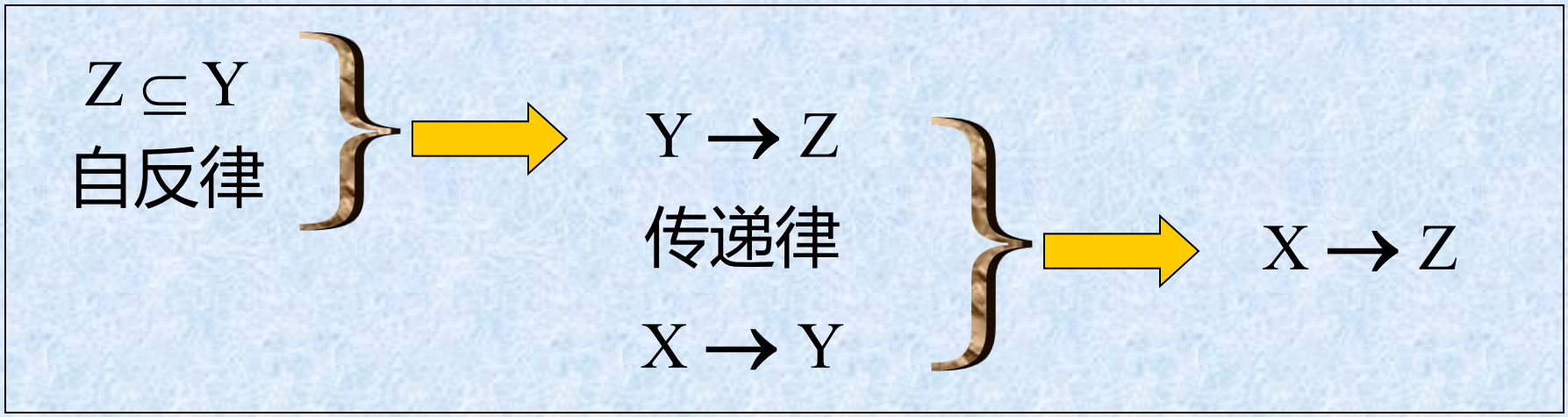
Armstrong公理系统

- 由Armstrong公理导出的推理规则
  - **合并律(union rule)**
    - 若$X \to Y$，$X \to Z$，则$X \to YZ$



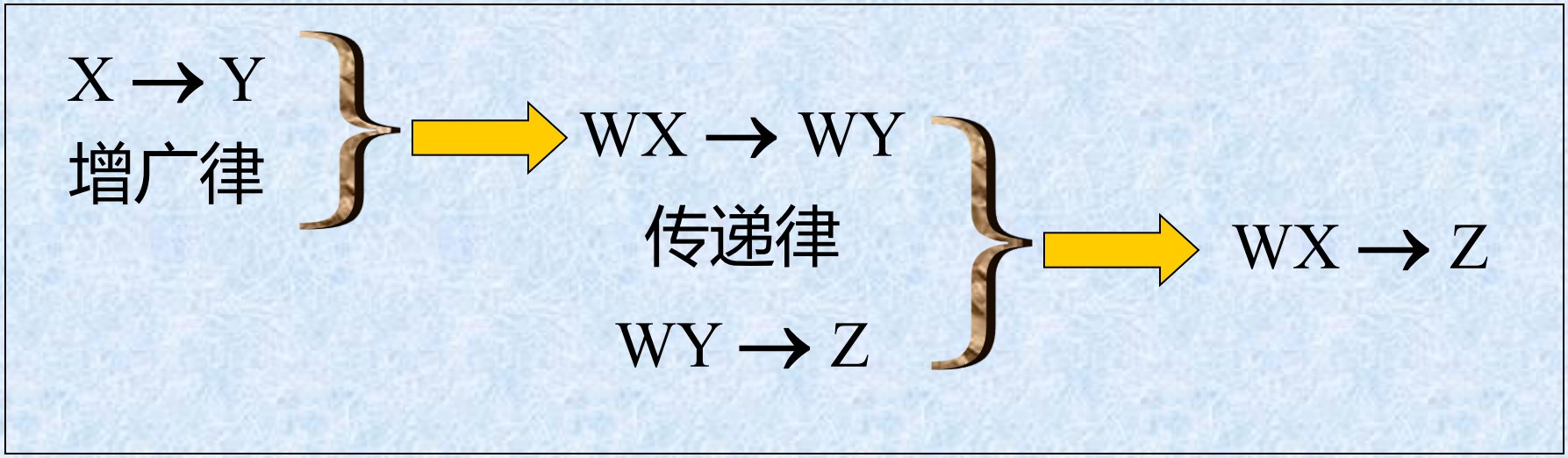
Armstrong公理系统

- 由Armstrong公理导出的推理规则
  - **分解律(decomposition rule)**
    - 若$X \to Y$，及$Z \subseteq Y$ 则 $X \to Z$



Armstrong公理系统

- 由Armstrong公理导出的推理规则
  - **伪传递律(pseudo-transitivity rule)**
    - 若$X \rightarrow Y$，$WY \rightarrow Z$，则$WX \rightarrow Z$



Armstrong公理系统

- 示例

R< U, F >, U = (A, B, C, G, H, I),

F = {A→B, A→C, CG→H, CG→I, B→H}

  - A→ H?
  - CG → HI?
  - AG → I?

Armstrong公理系统

- Armstrong公理是有效的、完备的。

  证明: (略)

  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。

6.3 数据依赖的公理系统

- 属性集的闭包
  - 设F为属性集U上的一组函数依赖，$X \subseteq U$，

    X_F^+= {A | X→A能由F根据Armstrong公理导出}

    称 X_F^+ 为属性集X关于函数依赖集F的闭包

6.3 数据依赖的公理系统

- 引理6.1

$$X\rightarrow A_1 A_2 \ldots A_k\text{成立} \Leftrightarrow X\rightarrow A_i\text{成立}(i=1, 2, \ldots, k)$$

- 引理6.2

$$X\rightarrow Y\text{能由Armstrong公理导出} \Leftrightarrow Y \subseteq X_F^+$$

闭包的计算

问题：有没有一般性的算法判定X→Y是否能由F根据Armstrong公理导出？

如果计算出F^+，再判断X→Y是否属于F^+，则由于F^+的计算非常复杂（NP完全问题），实际上是不可行的

由引理6.2，判定X→Y是否能由F根据Armstrong公理导出，可转化为求X_F^+，判定Y⊆ X_F^+是否成立

闭包的计算

- 算法（求属性集的闭包）

Input：X，F　　Output：X_F^+

方法：计算$X^{(i)}$ (i = 1,2,…n)

(1) $X^{(0)} = X$, i = 0

(2) $X^{(i+1)} = X^{(i)} \cup A$，其中，A是这样的属性：

在F中寻找尚未用过的左边是$X^{(i)}$的子集的函数依赖 $Y_j \rightarrow Z_j$ (j= 1,2,…,k)，$Y_j \subseteq X^{(i)}$，

在Z_j中寻找$X^{(i)}$中未出现过的属性集合A；若无这样的集合则转（4）

(3) 判断是否有$X^{(i+1)} = X^{(i)}$，若是则转（4），否则转（2）

(4) 输出$X^{(i)}$，即为X_F^+

闭包的计算

- 示例1

R< U, F >, U = (A, B, C, G, H, I), F = {A→B, A→C, CG→H, CG→I, B→H},计算$(AG)_F^+$

所用依赖	$(AG)_F^+$
$X^{(0)}$	AG
A→B	AGB
A→C	AGBC
CG→H	AGBCH
CG→I	AGBCH I
$(AG)_F^+$	AGBCH I

闭包的计算

- 示例2

  R< U, F >, U = (A, B, C, D, E), F = {AB→C, B→D, C→E, CE→B, AC→B},计算 $(AB)_F^+$

闭包的计算

- 示例2

R< U, F >, U = (A, B, C, D, E), F = {AB→C, B→D, C→E, CE→B, AC→B},计算 $(AB)_F^+$

所用依赖	$(AB)_F^+$
$X^{(0)}$	AB
AB→C	ABC
B→D	ABCD
C→E	ABCDE
$(AB)_F^+$	ABCDE

闭包的计算

- 示例3

  R< U, F >, U = (A, B, C, D, E, G), F = {A→E, BE→AG, CE→A, G→D},计算$(AB)_F^+$

闭包的计算

- 示例3

  R< U, F >, U = (A, B, C, D, E, G), F = {A→E, BE→AG, CE→A, G→D},计算$(AB)_F^+$

所用依赖	$(AB)_F^+$
A→E	ABE
BE→AG	ABEG
G→D	ABEGD
$(AB)_F^+$	ABEGD

Armstrong公理系统

- 示例

  属性集U={A，B，C}，

  函数依赖集F={$A \rightarrow B$，$B \rightarrow C$}

  求A^+，B^+，C^+

  $A^+ = ABC$

  $B^+ = BC$

  $C^+ = C$

函数依赖的等价和覆盖

- **函数依赖集的等价性**
  - 函数依赖集F，G，若$F^+ = G^+$，则称F与G等价。
  - $F^+ = G^+ \Leftrightarrow F \subseteq G^+$，$G \subseteq F^+$
- **最小覆盖（最小依赖集）**

  满足下列条件的函数依赖集F称为最小覆盖，记作F_{min}：
  - **单属性化**：F中任一函数依赖$X \rightarrow A$，A必是单属性
  - **无冗余化**：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
  - **既约化**：F中不存在这样的函数依赖$X \rightarrow A$，在X中有真子集Z，使得F与$F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$等价

函数依赖的等价和覆盖

- 算法—求解函数依赖集F的最小覆盖F_{min}
  - **单属性化**：逐个检查F中各函数依赖FD_i：$X\rightarrow Y$，

    若$Y=A_1 A_2 \ldots A_k$，$k\geq 2$，则用诸$X\rightarrow A_i$代替Y
  - **无冗余化**：逐个检查F中各函数依赖$X\rightarrow A$，

    令$G = F-\{X\rightarrow A\}$，若$A\in (X)_G^+$，则从F中去掉该函数依赖
  - **既约化**：逐个检查F中各函数依赖$X\rightarrow A$，

    设$X = B_1\ldots B_m$，逐个考查B_i，

    若$A \in (X - B_i)_F^+$，则以（$X - B_i$）取代X

函数依赖的等价和覆盖

- 示例一

  F = {A→B，B→A，A→C，B→C}，求F_{min}。

  - 检查A→B，G=F−{A→B}={B→A，A→C，B→C}

    $(A)_G^+$ ={A，C}，B∉{A，C}

  - 检查A→C，G=F−{A→C}={A→B，B→A，B→C}

    $(A)_G^+$ ={A，B，C}，C∈{A，B，C}

  所以从F中删除A→C，

  F_{min} = {A→B，B→A，B→C}

  或者

  F_{min} = {A→B，B→A，A→C}

  （主要是冗余检查）

函数依赖的等价和覆盖

- 示例二

F = {C→A，A→G，CG→B，B→A}，求F_{min}。

解答：F是无冗余的。

判断CG→B，

$(CG-C)_F^+ = (G)_F^+ = \{G\}$

$B \notin (CG-C)_F^+$

$(CG-G)_F^+ = (C)_F^+ = \{C，A，G，B\}$

$B \in (CG-G)_F^+$，以C代替CG

得出，F_{min} = {C→A，A→G，C→B，B→A}

进一步检查是否有冗余，去除C→A，最后

F_{min} = {A→G，C→B，B→A}

练习：将下列函数依赖集F化为最小依赖集

$$F = \left\{ \begin{array}{ll} AB \rightarrow C & D \rightarrow EG \\ C \rightarrow A & BE \rightarrow C \\ BC \rightarrow D & CG \rightarrow BD \\ ACD \rightarrow B & CE \rightarrow AG \end{array} \right)$$

6.4 模式分解

- 模式分解的定义
- 模式分解中的问题
- 无损连接分解
- 保持函数依赖的分解
- 关系模式的分解算法

模式分解的定义

- 模式分解
  - 函数依赖集合$F_i = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq U_i\}$称为F在$U_i$上的投影
  - 关系模式$R<U, F>$的一个分解是指

    $\rho = \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_n<U_n, F_n>\}$

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

模式分解的定义

- 分解的基本代数运算
  - 投影
  - 自然连接
- 分解的目标
  - 无损连接分解
  - 保持函数依赖
  - 达到更高级范式

模式分解中存在的问题

R(A, B, C)

A	B	C
1	1	2
2	2	1

$\prod_{AB}(R)$

A	B
1	1
2	2

$\prod_{BC}(R)$

B	C
1	2
2	1

$\prod_{AB}(R) \bowtie \prod_{BC}(R)$

A	B	C
1	1	2
2	2	1

无损分解

R(A, B, C)

A	B	C
1	1	1
2	1	2

$\prod_{AB}(R)$

A	B
1	1
2	1

$\prod_{BC}(R)$

B	C
1	1
1	2

$\prod_{AB}(R) \bowtie \prod_{BC}(R)$

A	B	C
1	1	1
1	1	2
2	1	1
2	1	2

有损分解

模式分解中存在的问题

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

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

A
a1
a2
a3
a4

B
b1
b2
b3

C
c1
c2

A	B
a1	b1
a2	b1
a3	b2
a4	b3
a5	b3

插入

⋈

A	C
a1	c1
a2	c1
a3	c2
a4	c1
a5	c3

=

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1
a5	b3	c3

违反 $B \rightarrow C$

模式分解中存在的问题

A	C
a1	c1
a2	c1
a3	c2
a4	c1

⋈

B	C
b1	c1
b2	c2
b3	c1

=

A	B	C
a1	b1	c1
a1	b3	c1
a2	b1	c1
a2	b3	c1
a3	b2	c2
a4	b1	c1
a4	b3	c1

A	B
a1	b1
a2	b1
a3	b2
a4	b3

⋈

B	C
b1	c1
b2	c2
b3	c1

=

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

无损连接分解

- 保持无损的模式分解
  - 有损分解的例子

R(A, B, C)

A	B	C
1	1	1
2	1	2

$\prod_{AB}(R)$

A	B
1	1
2	1

$\prod_{BC}(R)$

B	C
1	1
1	2

$\prod_{AB}(R) \bowtie \prod_{BC}(R)$

A	B	C
1	1	1
1	1	2
2	1	1
2	1	2

6.4 模式分解

- 1. $m_\rho(r)$
  - 设$\rho=\{R_1,R_2,\ldots R_n\}$是关系模式R的一个分解，r是R的一个关系，定义

$$m_\rho(r) = \Pi_{R1}(r) |\times| \Pi_{R2}(r) |\times| \ldots |\times| \Pi_{Rn}(r)$$

即$m_\rho(r)$ 是r在ρ中各关系模式上投影的连接

- 2. 引理（5.4）

设R<U,F>是一个关系模式，$\rho=\{R_1,R_2,\ldots R_n\}$是关系模式R的一个分解，r是R的一个关系实例，则：

(1) $r \subseteq m_\rho(r)$

(2) 若$s = m_\rho(r)$,则$\Pi_{Ri}(r) = \Pi_{Ri}(s)$

(3) $m_\rho(m_\rho(r)) = m_\rho(r)$

无损连接分解

- 定义

  关系模式R<U , F> ， $U = \bigcup_{i=1}^{n} U_i$ ，

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

  是R<U , F>的一个分解，r是R<U , F>的一个关系，定义$m_\rho(r) = \bowtie_{i=1}^{n} \prod R_i(r)$ ，若对于R<U , F>的任一个关系r，都有$r = m_\rho(r)$，则称ρ是R<U , F>的一个无损连接分解。

无损连接分解的通用判别算法

- 通用算法：判别一个分解是否具有无损连接性

输入: $R(A_1,A_2,\ldots A_n)$，R的函数依赖集F，R的分解 $\rho=\{R_1,R_2,\ldots R_k\}$

输出：分解ρ是否具有无损连接性

无损连接分解判别算法

方法：

- (1) 建立矩阵S, 列j 对应属性A_j, 行i对应R_i;

- (2) FOR i = 1 TO k DO
  FOR j =1 TO n DO
  IF R_i 包含属性A_j THEN
  S[i,j] := a_j;
  ELSE
  S[i,j] : = b_{ij};
  END FOR;
  END FOR;

无损连接分解判别算法 示例1-2步

示例 第1, 2步

- U={A,B,C,D,E}, F={AB→C, C→D,D→E}
  ρ ={(A, B, C), (C, D), (D, E)}

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

无损连接分解判别算法

- (3) DO UNTIL S 无变化

  FOR 每个X－>Y DO

  FOR S中所有在X对应的列上具有相同符号的行 $i_1,i_2,\ldots i_k$ DO

  按照下列规则修改Y所对应列的符号：

  a. FOR 每个具有“a”类符号的Y对应列 DO

  把该列$i_1,i_2,\ldots i_k$行的符号改为相同的a类符号；

  END FOR;

  b. FOR 每个不具有“a”类符号的Y对应列 DO

  在该列上选择一个“b”类符号

  把该列$i_1,i_2,\ldots i_k$行的符号改为相同的b类符号；

  END FOR;

  END FOR;

  END FOR;

  END UNTIL ;

无损连接分解判别算法 示例第3步

- U={A,B,C,D,E}, F={AB→C, C→D,D→E}
  ρ ={(A, B, C), (C, D), (D, E)}

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

AB→C

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

C→D

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

无损连接分解判别算法

- (4) 如果S中存在一行全为“a”类符号，则ρ具有无损连接性，否则ρ不具有无损连接性

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

D→E

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	a_5
CD	b_{21}	b_{22}	a_3	a_4	a_5
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

无损连接分解判别算法示例

- 示例一： U={A,B,C,D,E}, F={AB→C, C→D,D→E}
  ρ ={(A, B, C), (C, D), (D, E)}

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

AB→C

	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

C→D

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	b_{15}
CD	b_{21}	b_{22}	a_3	a_4	b_{25}
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

D→E

	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	a_5
CD	b_{21}	b_{22}	a_3	a_4	a_5
DE	b_{31}	b_{32}	b_{33}	a_4	a_5

无损连接分解判别算法示例

- 示例二：U={A,B,C,D,E},
  F={A→C, B→C, C→D,DE→C ,CE→A}
  ρ ={(A, D), (A, B), (B, E), (C, D, E), (A, E)}

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{53}	b_{54}	a_5

A→C

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	b_{13}	b_{54}	a_5

无损连接分解判别算法示例

B→C

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{13}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{13}	b_{54}	a_5

C→D

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	a_4	b_{25}
BE	b_{31}	a_2	b_{13}	a_4	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{13}	a_4	a_5

无损连接分解简单判别定理

简单算法：

- 定理5.5

关系模式R(U)的分解ρ={R_1, R_2}，则ρ是一个无损连接分解的充要条件是 $R_1 \cap R_2 \rightarrow R_1 - R_2$（或$R_1 \cap R_2 \rightarrow R_2 - R_1$）成立

	$R_1 \cap R_2$	$R_1 - R_2$	$R_2 - R_1$
R_1	a…a	a…a	b…b
R_2	a…a	b…b	a…a

$R_1 \cap R_2 \rightarrow R_1 - R_2$

	$R_1 \cap R_2$	$R_1 - R_2$	$R_2 - R_1$
R_1	a…a	a…a	b…b
R_2	a…a	a…a	a…a

无损连接分解简单判别示例

R=ABC, F={A → B},

ρ_1={R_1(AB), R_2(AC)}

$R_1 \cap R_2$ = A, $R_1 - R_2$ = B

由A → B，得到ρ_1是无损连接分解

ρ_2={R_1(AB), R_2(BC)}

$R_1 \cap R_2$ = B, $R_1 - R_2$ = A, $R_2 - R_1$ = C

B→A, B→C均不成立，所以ρ_2不是无损连接分解

保持函数依赖的分解

- 保持函数依赖的模式分解
  - Z是U的子集，函数依赖集合F在Z上的投影定义为

$$\prod_Z(F) = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq Z\}$$

  - 设$\rho = \{R_1, R_2, \ldots, R_n\}$是关系模式R<U , F>的一个分解，如果$F^+ = (\bigcup_{i=1}^{n} \prod_{R_i}(F))^+$，则称ρ是保持函数依赖的分解

保持函数依赖的分解

关系模式R<U, F>

U = {CITY, ST, ZIP},

F = {(CITY, ST) → ZIP, ZIP → CITY}

ρ = {R_1 = {ST, ZIP}, R_2 = {CITY, ZIP}}

$R_1 \cap R_2$ ={ZIP}, $R_2 - R_1$ ={CITY}

∵ $R_1 \cap R_2 \rightarrow R_2 - R_1$　　∴分解是无损的

$\prod_{R_1}(F) = \{\}$，　$\prod_{R_2}(F) = \{ZIP \rightarrow CITY\}$

$\prod_{R_1}(F) \cup \prod_{R_2}(F) = \{ZIP \rightarrow CITY\}$

丢失了函数依赖(CITY, ST) → ZIP

保持函数依赖的分解

ST	ZIP
Haidian	100862
Haidian	100971

ZIP	CITY
100862	Beijing
100971	Beijing

CITY	ST	ZIP
Beijing	Haidian	100862
Beijing	Haidian	100971

违反了函数依赖(CITY, ST) → ZIP

模式分解实例

- 实例2：表（姓名，级别，工资)，
  其中函数依赖为{姓名—>级别，级别—>工资}
  可以有两种分解途径，
  分解一：（姓名，工资），（工资，级别）
  分解二：（姓名，级别），（工资，级别）

姓名	级别	工资
ZHAO	4	500
QIAN	5	600
SUN	6	700
LI	7	600

姓名	工资
ZHAO	500
QIAN	600
SUN	700
LI	600

级别	工资
4	500
5	600
6	700
7	500

丢失函数依赖

不同行业机构的不同工资级别会有相同工资数额，因此按分解一，有可能导致同一职工对应不同的工资级别，从而丢失了有关职工工资级别的信息（丢失了函数依赖：姓名→级别）

关系模式的分解算法

- 三个重要事实：
  - （1）若要求分解保持函数依赖，那么模式分解总可以达到3NF，但不一定能达到BCNF；
  - （2）若要求分解既保持函数依赖，又具有无损连接性，可以达到3NF，但不一定能达到BCNF；
  - （3）若要求分解具有无损连接性，则一定可以达到4NF。
  - 相关算法参见教材P_{196}～P_{201}

练习题

观察关系R(A, B, C, D), 具有FD:

C→D; C→A; B→C

请回答:

1)找出R的候选码;

2)指出R符合的最高范式;

3)给出一个保持依赖的分解。

练习题解答

1)B^+=BCDA, C^+=CDA, 因此B是候选码

2)主属性是B，其余均为非主属性，它们均完全依赖于码；D、A均通过C传递依赖于码(B)；C→D; C→A函数依赖于非码属性，不是BCNF，3NF; 仅满足2NF

3)∵C→D;C→A ∴C^+=CDA

∴分解为 R1(CDA), R2(CB)

∵$F_1 \cup F_2 = F$ ∴依赖保持

下课了。。。

研究

休息一会儿。。。