厦门大学计算机科学系本科生课程

《数据库系统原理》

第6章 关系数据理论
（2016版）

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

主页：http://www.cs.xmu.edu.cn/linziyu

第6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

6.1 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

6.1 问题的提出

- 概念回顾

– 关系：描述实体、属性、实体间的联系。
  – 从形式上看，它是一张二维表，是所涉及属性的笛卡尔积的一个子集。
– 关系模式：用来定义关系。
– 关系数据库：基于关系模型的数据库，利用关系来描述现实世界。
  – 从形式上看，它由一组关系组成。
– 关系数据库的模式：定义这组关系的关系模式的全体。

6.1 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

6.1 问题的提出

- 什么是数据依赖

1. 完整性约束的表现形式

– 限定属性取值范围：例如学生成绩必须在**0-100**之间

– 定义属性值间的相互关连（主要体现于值的相等与否） 这就是数据依赖，它是数据库模式设计的关键

2. 数据依赖

– 是通过一个关系中属性间值的相等与否体现出来的数据间的相互关系

– 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

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

6.1 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（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

6.1 问题的提出

- 数据依赖对关系模式的影响

例：描述学校的数据库：

学生的学号（Sno）、所在系（Sdept）

系主任姓名（Mname）、课程名（Cname）

成绩（Grade）

单一的关系模式： **Student <U、F>**

***U* ＝ {*Sno, Sdept, Mname, Cname, Grade*}**

6.1 问题的提出

- 数据依赖对关系模式的影响

学校数据库的语义：

1. 一个系有若干学生， 一个学生只属于一个系；
2. 一个系只有一名主任；
3. 一个学生可以选修多门课程， 每门课程有若干学生选修；
4. 每个学生所学的每门课程都有一个成绩。

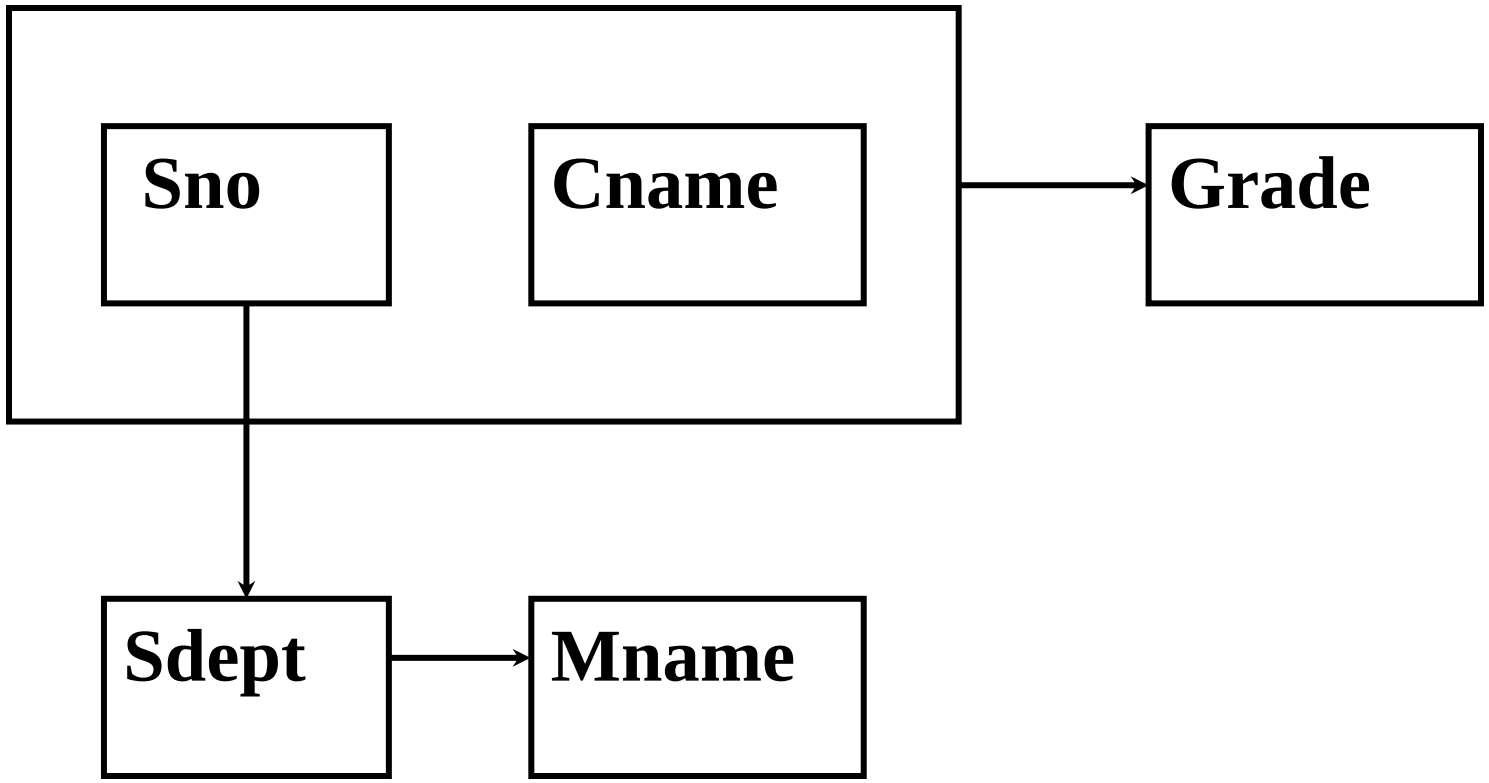
***U* = { *Sno, Sdept, Mname, Cname, Grade* }**

属性组**U**上的一组函数依赖**F**：

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

6.1 问题的提出

- 数据依赖对关系模式的影响



6.1 问题的提出

- 关系模式**Student<U, F>**中存在的问题

学 号 Sno	所 在 系 Sdept	系主任 Mname	课程名 Cname	成绩 Grade
95001	IS	李勇	高数	80
95002	IS	李勇	高数	73
95003	MA	王敏	高数	91
95004	IS	李勇	外语	67

6.1 问题的提出

- 关系模式Student<U, F>中存在的问题

1. 数据冗余太大

浪费大量的存储空间

例：每一个系主任的姓名重复出现

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。

例：某系更换系主任后，系统必须修改与该系学生有关的每一个元组

3. 插入异常（Insertion Anomalies）

该插的数据插不进去

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

4. 删除异常（Deletion Anomalies）

不该删除的数据不得不删

例，如果某个系的学生全部毕业了，我们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。

6.1 问题的提出

- 数据依赖对关系模式的影响

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。

第6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

6.2 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

一、函数依赖

二、平凡函数依赖与非平凡函数依赖

三、完全函数依赖与部分函数依赖

四、传递函数依赖

6.2.1 函数依赖

一、函数依赖

定义6.1　设R(U)是一个属性集U上的关系模式，X和Y是U的子集。

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

X称为这个函数依赖的决定属性集(Determinant)。

Y=f(x)

6.2.1 函数依赖

说明：

1. 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。
2. 函数依赖是语义范畴的概念。只能根据数据的语义来确定函数依赖。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
3. 数据库设计者可以对现实世界作强制的规定。例如规定不允许同名人出现，函数依赖“姓名→年龄”成立。所插入的元组必须满足规定的函数依赖，若发现有同名人存在， 则拒绝装入该元组。

6.2.1 函数依赖

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

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。

二、平凡函数依赖与非平凡函数依赖

在关系模式R(U)中，对于U的子集X和Y，

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

若X→Y，但Y $\subseteq$ X，则称X→Y是平凡的函数依赖

例：在关系SC(Sno, Cno, Grade)中，

非平凡函数依赖：(Sno, Cno) → Grade

平凡函数依赖：(Sno, Cno) → Sno

(Sno, Cno) → Cno

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，因此若不特别声明，我们总是讨论非平凡函数依赖。

6.2.1 函数依赖

三、完全函数依赖与部分函数依赖

定义6.2 在关系模式R(U)中，如果X→Y，并且对于X的任何一个真子集X'，都有 $X' \not\rightarrow Y$，则称Y完全函数依赖于X，记作 $X \xrightarrow{f} Y$。

若X→Y，但Y不完全函数依赖于X，则称Y部分函数依赖于X，记作 $X \xrightarrow{P} Y$。

例: 在关系SC(Sno, Cno, Grade)中，

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

因此：(Sno, Cno) $\xrightarrow{f}$ Grade

6.2.2 码

定义6.4 设K为关系模式R<U,F>中的属性或属性组合。若K $\xrightarrow{f}$U，则K称为R的一个侯选码（Candidate Key）。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）。

- 主属性与非主属性
- **ALL KEY**

6.2.2 码

- **外部码**

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

– 主码又和外部码一起提供了表示关系间联系的手段。

6.2.3 范式

- 范式是符合某一种级别的关系模式的集合。
- 关系数据库中的关系必须满足一定的要求。满足不同程度要求的为不同范式。
- 范式的种类：

第一范式(1NF)
第二范式(2NF)
第三范式(3NF)
BC范式(BCNF)
第四范式(4NF)
第五范式(5NF)

6.2.3 范式

- 各种范式之间存在联系：

$$1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$$

- 某一关系模式**R**为第**n**范式，可简记为**R∈nNF**。

6.2.4 2NF

- **1NF的定义**
  如果一个关系模式**R**的所有属性都是不可分的基本数据项，则**R∈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。

6.2.4 2NF

例: 关系模式 **SLC(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 → Sdept

(Sno, Cno) $\xrightarrow{p}$ Sdept (?)

Sno → Sloc

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

Sdept → Sloc

6.2.4 2NF

例: 关系模式 SLC(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

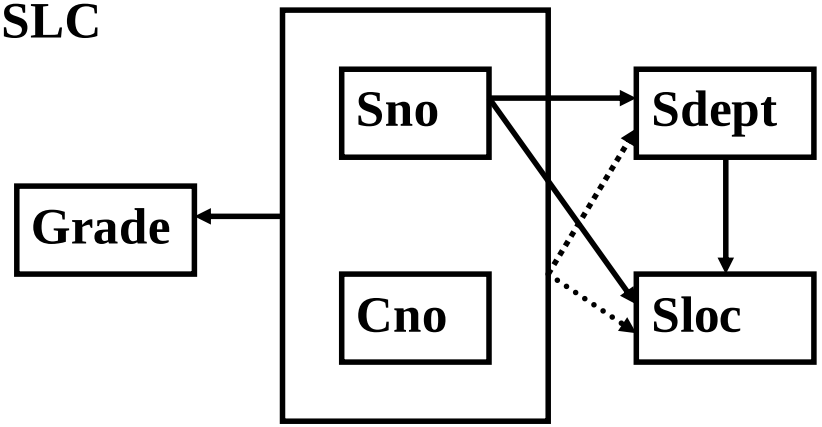
Sno → Sdept

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

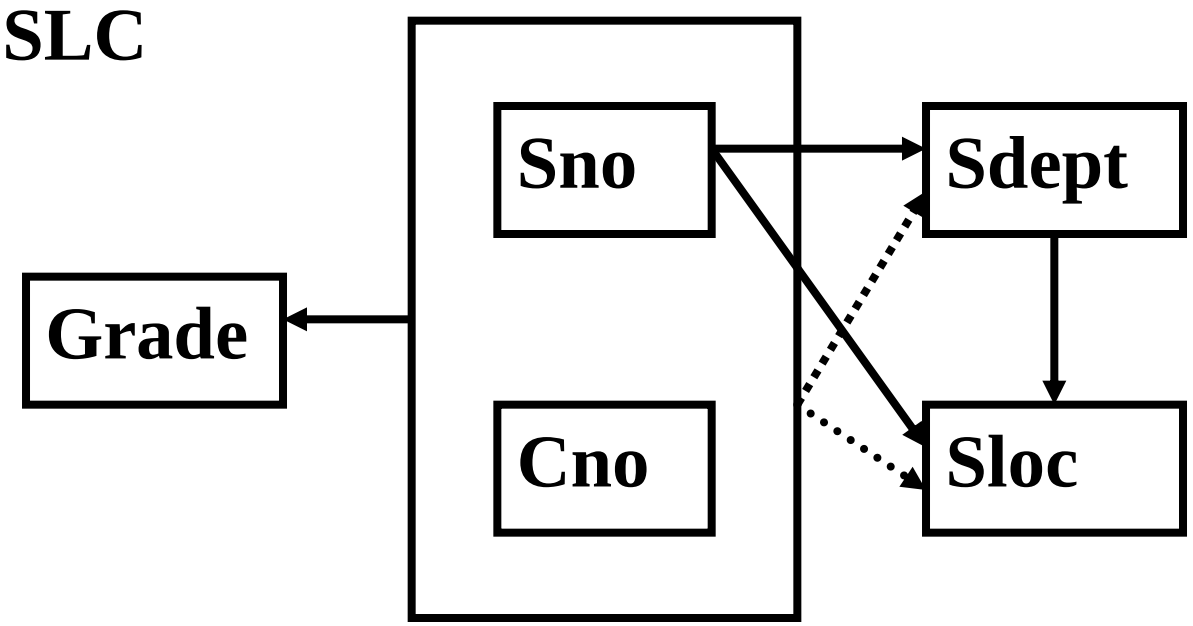
Sno → Sloc

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

Sdept → Sloc



6.2.4 2NF



SLC的码为(Sno, Cno)

- **SLC满足第一范式**
- **非主属性Sdept和Sloc部分函数依赖于码(Sno, Cno)**

课堂练习

- 已知学生关系模式

S(Sno,Sname,SD,Sdname,Course,Grade)

其中，Sno是学号，Sname是姓名，SD是系名，Sdname是系主任名，Course是课程，Grade是成绩

（1）写出关系模式S的基本函数依赖和主码；（做本题）

（2）原关系模式是第几范式？如何分解成高一级范式？

（3）将关系模式分解成3NF，并说明为什么？

6.2.4 2NF

- SLC不是一个好的关系模式

(1) 插入异常　　**SLC(Sno, Sdept, Sloc, Cno, Grade)**

假设Sno＝95102，Sdept＝IS，Sloc＝N的学生还未选课，因课程号是主属性，因此该学生的信息无法插入SLC。

(2) 删除异常

假定某个学生本来只选修了3号课程这一门课。现在因身体不适，他连3号课程也不选修了。因课程号是主属性，此操作将导致该学生信息的整个元组都要删除。

6.2.4 2NF

(3) 数据冗余度大 **SLC(Sno, Sdept, Sloc, Cno, Grade)**

如果一个学生选修了10门课程，那么他的Sdept和Sloc值就要重复存储了10次。

(4) 修改复杂

例如学生转系，在修改此学生元组的Sdept值的同时，还可能需要修改住处（Sloc）。如果这个学生选修了K门课，则必须无遗漏地修改K个元组中全部Sdept、Sloc信息。

6.2.4 2NF

- 原因

  Sdept、 Sloc部分函数依赖于码。

- 解决方法

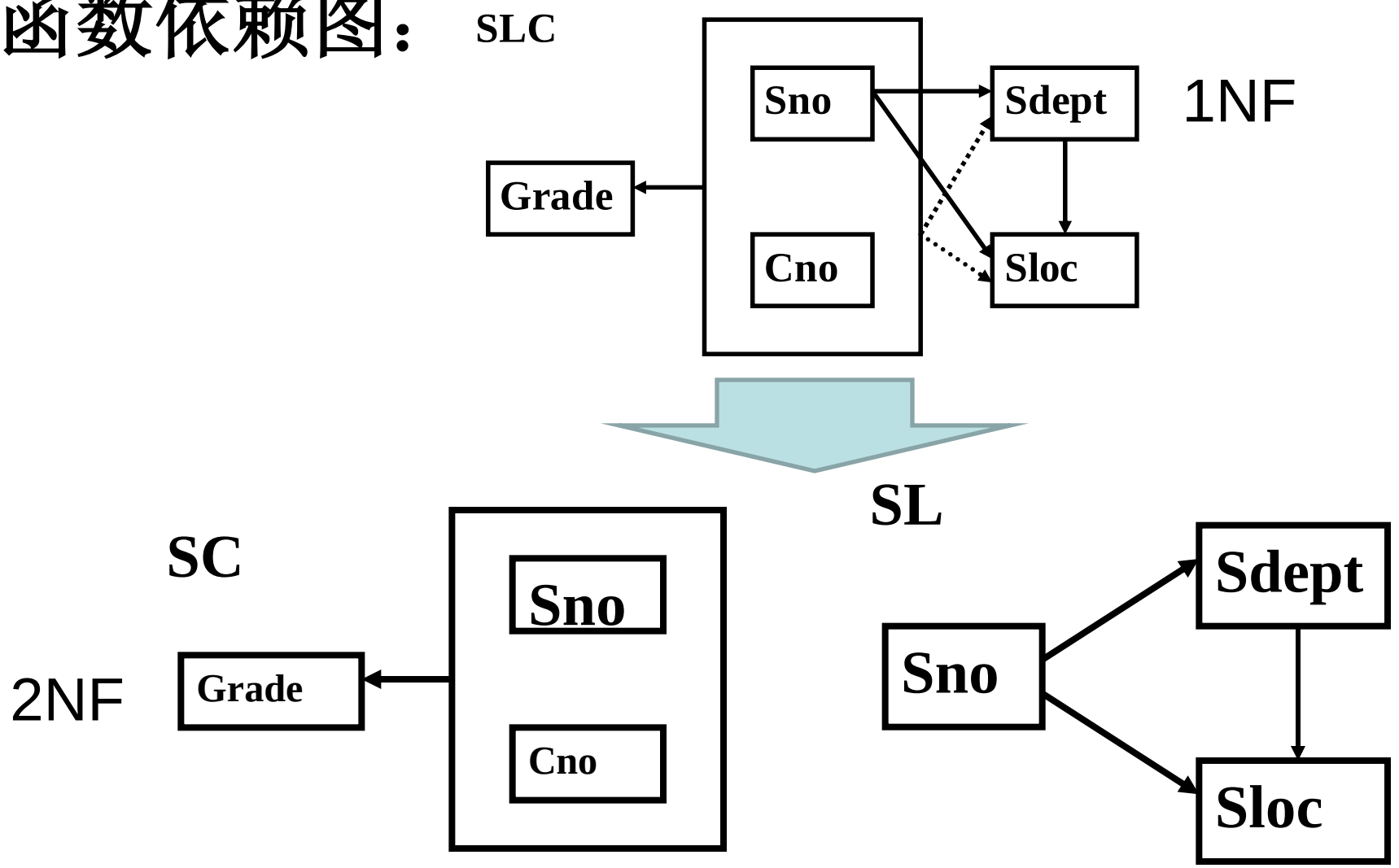
  SLC分解为两个关系模式，以消除这些部分函数依赖

  SC（Sno， Cno， Grade）

  SL（Sno， Sdept， Sloc）

6.2.4 2NF

函数依赖图：



6.2.4 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。

6.2.4 2NF

- **2NF的定义**

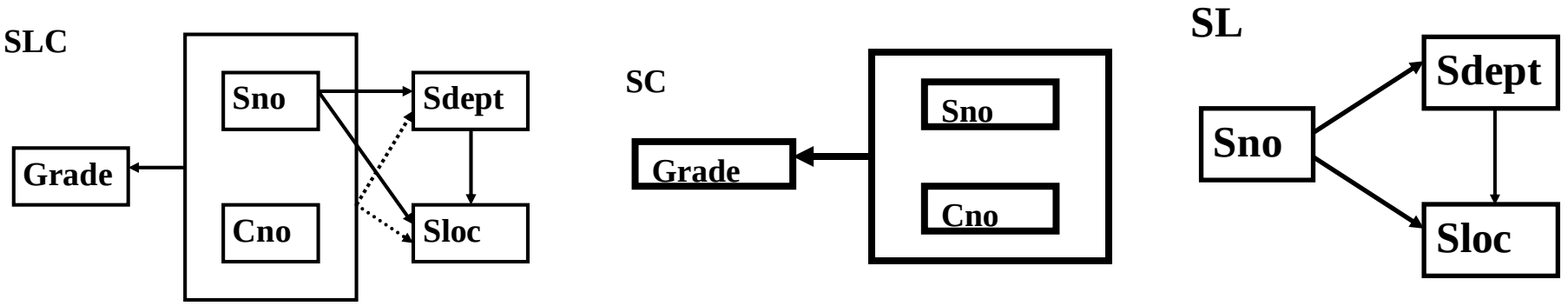
  定义6.6 若关系模式R∈1NF，并且每一个非主属性都完全函数依赖于R的码，则R∈2NF。

  例：SLC(Sno, Sdept, Sloc, Cno, Grade) ∈1NF

  SLC(Sno, Sdept, Sloc, Cno, Grade) ∉2NF

  SC（Sno， Cno， Grade） ∈ 2NF

  SL（Sno， Sdept， Sloc） ∈ 2NF



课堂练习

- 已知学生关系模式

S(Sno,Sname,SD,Sdname,Course,Grade)

其中，Sno是学号，Sname是姓名，SD是系名，Sdname是系主任名，Course是课程，Grade是成绩

（1）写出关系模式S的基本函数依赖和主码；

（2）原关系模式是第几范式？如何分解成高一级范式？（做本题）

（3）将关系模式分解成3NF，并说明为什么？

四、传递函数依赖

定义6.3 在关系模式R(U)中，如果X→Y，Y→Z，且 $Y \subsetneq X$，$Y \not\rightarrow X$，则称Z传递函数依赖于X。

注: 如果Y→X， 即X←→Y，则Z直接依赖于X。

例: 在关系Std(Sno, Sdept, Mname)中，有：

Sno → Sdept，Sdept → Mname

Mname传递函数依赖于Sno

6.2.5 3NF

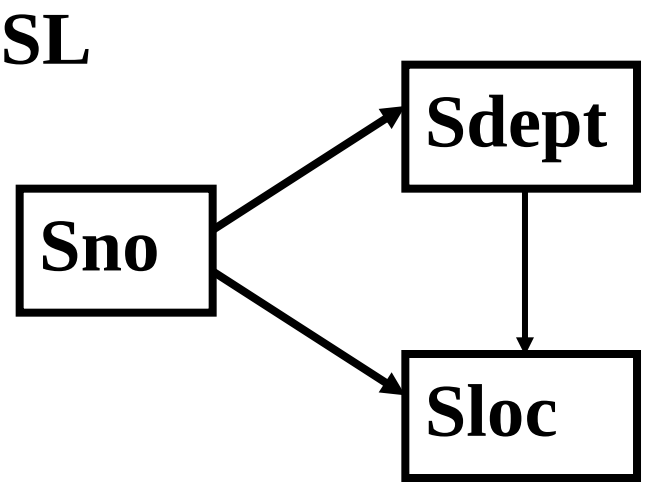
例：2NF关系模式SL(Sno, Sdept, Sloc)中

- 函数依赖：

  Sno→Sdept

  Sdept→Sloc

  Sno→Sloc



Sloc传递函数依赖于Sno，即SL中存在非主属性对码的传递函数依赖。

6.2.5 3NF

- 解决方法

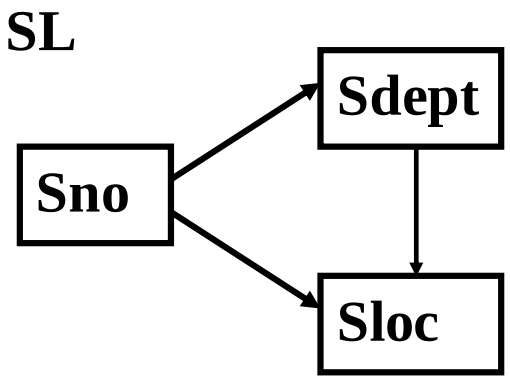
  采用投影分解法，把**SL**分解为两个关系模式，以消除传递函数依赖：

  SD（Sno， Sdept）

  DL（Sdept， Sloc）

SD的码为Sno， DL的码为Sdept。

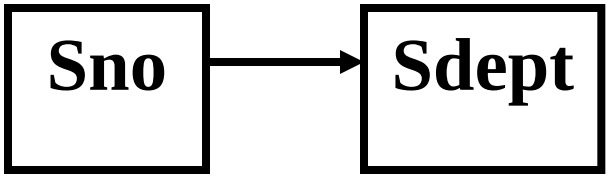
6.2.5 3NF

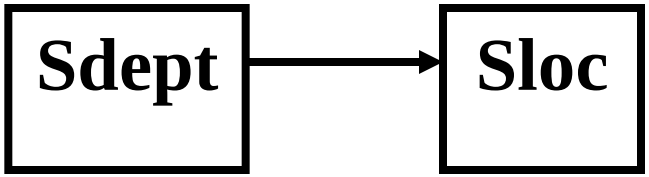


2NF

SD的码为Sno， DL的码为Sdept。

3NF





SD **DL**

6.2.5 3NF

- **3NF的定义**

  定义6.8 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（$Z \subsetneq Y$），使得$X \rightarrow Y$，$Y \not\rightarrow X$，$Y \rightarrow Z$，成立，则称$R<U，F> \in 3NF$。

例， SL(Sno, Sdept, Sloc) ∈ 2NF

SL(Sno, Sdept, Sloc) ∉ 3NF

SD（Sno， Sdept） ∈ 3NF

DL（Sdept， Sloc）∈ 3NF

6.2.5 3NF

- 若R∈3NF，则R的每一个非主属性既不部分函数依赖于候选码也不传递函数依赖于候选码。
- 如果R∈3NF，则R也是2NF。
- 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，并不能完全消除关系模式中的各种异常情况和数据冗余。

课堂练习

- 已知学生关系模式

S(Sno,Sname,SD,Sdname,Course,Grade)

其中，Sno是学号，Sname是姓名，SD是系名，Sdname是系主任名，Course是课程，Grade是成绩

（1）写出关系模式S的基本函数依赖和主码；

（2）原关系模式是第几范式？如何分解成高一级范式？

（3）将关系模式分解成3NF，并说明为什么？（做本题）

6.2.6 BCNF

- 定义6.9 设关系模式R<U，F>∈1NF，如果对于R的每个函数依赖X→Y，若Y不属于X，则X必含有码，那么R∈BCNF。

若R∈BCNF

- 每一个决定属性集（因素）都包含（候选）码
- R中的所有属性（主，非 主属性）都完全函数依赖于码
- 没有任何属性完全函数依赖于非码的任何一组属性
- R∈3NF(?) 若R∈3NF 则 R不一定∈BCNF
- 即在第三范式的基础上，数据库表中不存在任何属性对任一候选码的传递函数依赖和部分函数依赖

6.2.6 BCNF

- **证明题：若关系模式R∈BCNF，则R∈2NF。**

反证法：

假设 R∉2NF，

则存在$X \xrightarrow{P} Y$（X 码，Y 非主属性），

即∃子集$X' \xrightarrow{F} Y$，X'不包含码。

∵R∈BCNF 任给 V->W 则 V 必然包含码

∴R∈2NF。

6.2.6 BCNF

- 例子：关系模式C(Cno,Cname,Pcno)，它只有一个码Cno，这里没有任何属性对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，C同时又是码，根据定义，C ∈ BCNF

6.2.6 BCNF

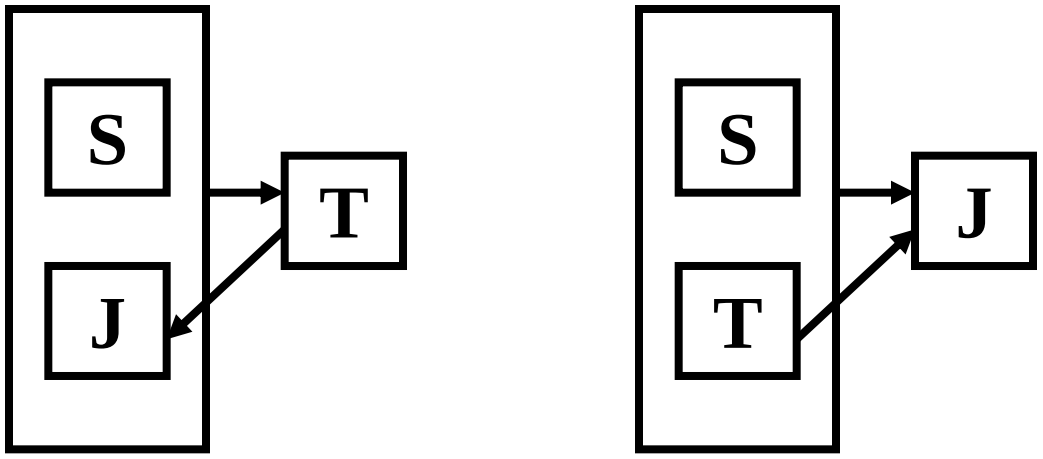
- 例子：关系模式SJP（S,J,P）中，S是学生，J表示课程，P表示名次，每一个学生选修每门课程的成绩有一定的名次，每门课程中每一名次只有一个学生（即没有并列名次），可以得到以下依赖：
- （S，J）→ P;（J, P) → S
- 所以(S,J)和(J,P)都可以作为候选码，这个关系模式中没有属性对码传递依赖或部分依赖，所以，SJP ∈ 3NF，而且除(S,J)和(J,P)以外没有其他决定因素，所以SJP ∈ BCNF

6.2.6 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。每门课由若干教师教，某一学生选定某门课，就确定了一个固定的教师。某个学生选修某个教师的课就确定了所选课的名称：
- **T→J，(S，J)→T，(S，T)→J**

6.2.6 BCNF



STJ

6.2.6 BCNF

STJ∈3NF

- **(S，J)和(S，T)都可以作为候选码**
- **S、T、J都是主属性**

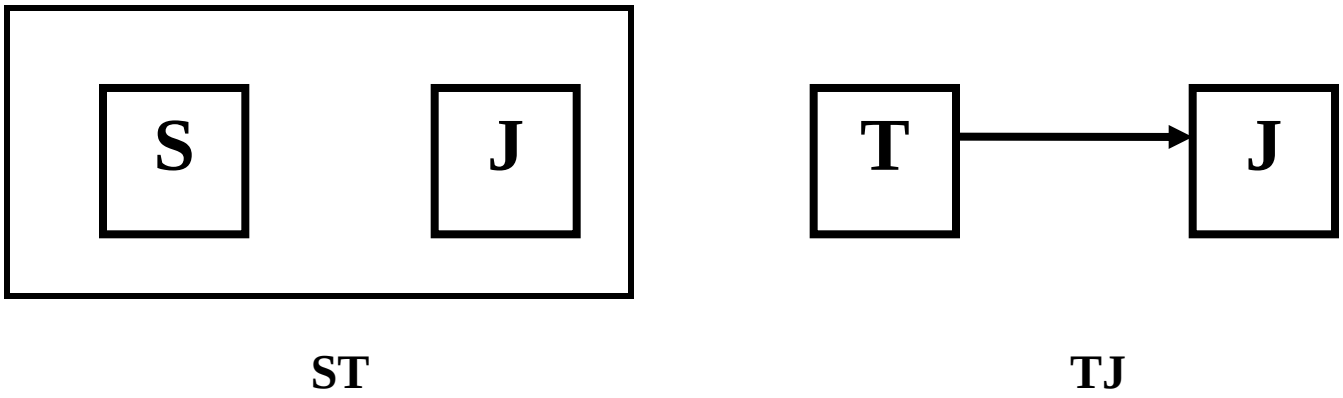
STJ∉BCNF

- **T→J，T是决定属性集，T不是候选码**

6.2.6 BCNF

解决方法：将STJ分解为二个关系模式：

SJ(S，J) ∈ BCNF， TJ(T，J)∈ BCNF



没有任何属性对码的部分函数依赖和传递函数依赖

课堂作业

有一个配件管理表WPE(WNO,PNO,ENO,QNT)，其中，WNO表示仓库号，PNO表示配件号，ENO表示职工号，QNT表示数量。有以下约束要求：

（1）一个仓库有多名职工

（2）一个职工仅在一个仓库工作

（3）每个仓库里一种型号的配件由一个职工负责，但一个人可以管理几种配件；

（4）同一个型号的配件可以分别放在几个仓库中

（5）一个仓库存储某种配件的数量是一定的

（6）一个职工管理某种配件的数量是一定的

问题：

（1）请写出表中的函数依赖关系

（2）判断该表是否是3NF？

（3）判断该表是否是BCNF？

课堂作业答案

- 函数依赖关系：
- ENO →**WNO**
- **(WNO,PNO) →QNT**
- **(WNO,PNO) →ENO**
- **(ENO,PNO) →QNT**

课堂作业答案

- 候选码包括： **(WNO,PNO)和(ENO,PNO)**
- **ENO,PNO,WNO都是主属性，QNT是非主属性**
- **所有非主属性都是直接依赖于候选码，因此是3NF**
- **关于主属性：**
- **(WNO,PNO) →ENO；**ENO **→WNO，得到传递依赖(WNO,PNO) →WNO，所以不是BCNF**

课堂作业答案

- 可以继续分拆成两个表
- 管理表EP（ENO,PNO,QNT）
- 工作表EW（ENO,WNO）

两个表属于BCNF

多值依赖与第四范式（4NF）

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。每个教师可以讲多门课程，每种参考书可供多门课使用。

关系模式**Teaching(C, T, B)**

课程**C**、教师**T** 和 参考书**B**

表6.1

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
⋮	⋮	⋮

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
…	…	…

多值依赖与第四范式（续）

- **Teaching∈BCNF:**
- **Teach具有唯一候选码(C，T，B)， 即全码**
- **Teaching模式中存在的问题**

(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

(2)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

例如物理课增加一名教师刘关，需要插入两个元组：

（物理，刘关，普通物理学）

（物理，刘关，光学原理）

多值依赖与第四范式（续）

(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

(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

6.2.7 多值依赖

- **定义6.10**

  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，多值依赖 X→→Y成立当且仅当对R的任一关系r，r在（X，Z）上的每个值对应一组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，B有一组值与之对应，而不论T取何值

6.2.7 多值依赖

- 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*** ∈ ***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*→→*Y***。 这里，***X***，***Y***是***U***的子集，***Z*=*U*-*X*-*Y***。

6.2.7 多值依赖

t	***x***	***y1***	***z2***
s	***x***	***y2***	***z1***
w	***x***	***y1***	***z1***
v	***x***	***y2***	***z2***

6.2.7 多值依赖

- 平凡多值依赖和非平凡的多值依赖
  - 若**X→→Y**，而**Z＝φ**，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

6.2.8 4NF

- 定义6.10　关系模式R<U，F>∈1NF，如果对于R的每个非平凡多值依赖X→→Y（$Y \not\subseteq X$），X都含有候选码，则R∈4NF。
- 如果R ∈ 4NF， 则R ∈ BCNF

  不允许有非平凡且非函数依赖的多值依赖

  允许的是函数依赖（是非平凡多值依赖）

6.2.8 4NF

例： Teach(C,T,B) ∉ 4NF

存在非平凡的多值依赖C→→T，且C不是候选码

- 用投影分解法把Teach分解为如下两个关系模式：

CT(C, T) ∈ 4NF

CB(C, B) ∈ 4NF

C→→T， C→→B是平凡多值依赖

6.2.9 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。
- 一个关系只要其分量都是不可分的数据项，它就是规范化的关系，但这只是最基本的规范化。
- 规范化程度可以有多个不同的级别

6.2.9 规范化

- 规范化程度过低的关系不一定能够很好地描述现实世界，可能会存在插入异常、删除异常、修改复杂、数据冗余等问题
- 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

6.2.9 规范化

- 关系模式规范化的基本步骤

消除决定属性集非码的非平凡函数依赖	1NF ↓ 消除非主属性对码的部分函数依赖 2NF ↓ 消除非主属性对码的传递函数依赖 3NF ↓ 消除主属性对码的部分和传递函数依赖 BCNF ↓ 消除非平凡且非函数依赖的多值依赖 4NF

6.2.9 规范化

- **规范化的基本思想**
  - 消除不合适的数据依赖
  - 将各关系模式达到某种程度的“分离”
  - 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
  - 所谓规范化实质上是概念的单一化
  - 不能一味追求规范化程度高

6.2.9 规范化

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

第六章 关系数据理论

6.1 数据依赖

6.2 规范化

6.3 数据依赖的公理系统

6.4 模式的分解

6.3 数据依赖的公理系统

- 逻辑蕴含

  定义6.11 对于满足一组函数依赖 F 的关系模式$R<U，F>$，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$X \rightarrow Y$

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含的函数依赖

1. Armstrong公理系统

关系模式$R<U, F>$来说有以下的推理规则：

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

定理 6.l Armstrong推理规则是正确的

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

证: 设$Y \subseteq X \subseteq U$

对$R<U, F>$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

定理6.l

(2)增广律： 若$X \rightarrow Y$为F所蕴含，且$Z \subseteq U$，则$XZ \rightarrow YZ$ 为F所蕴含。

证：设$X \rightarrow Y$为F所蕴含，且$Z \subseteq U$。

设$R<U, F>$ 的任一关系r中任意的两个元组t，s；

若$t[XZ]=s[XZ]$，则有$t[X]=s[X]$和$t[Z]=s[Z]$；

由$X \rightarrow Y$，于是有$t[Y]=s[Y]$，所以

$t[YZ]=s[YZ]$，所以$XZ \rightarrow YZ$为F所蕴含.

增广律得证。

定理6.l

(3) 传递律：若$X \rightarrow Y$及$Y \rightarrow Z$为F所蕴含，则$X \rightarrow Z$为 F所蕴含。

证：设$X \rightarrow Y$及$Y \rightarrow Z$为F所蕴含。

对$R<U, F>$ 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$Y \rightarrow Z$，当$t[Y]=s[Y]$时，一定有$t[Z]=s[Z]$

所以$X \rightarrow Z$为F所蕴含.

传递律得证。

2. 导出规则

1.根据A1，A2，A3这三条推理规则可以得到下面三条推理规则：

- 合并规则：由$X \rightarrow Y$，$X \rightarrow Z$，有$X \rightarrow YZ$。

  （A2， A3）

- 伪传递规则：由$X \rightarrow Y$，$WY \rightarrow Z$，有$XW \rightarrow Z$。

  （A2， A3）

- 分解规则：由$X \rightarrow Y$及 $Z \subseteq Y$，有$X \rightarrow Z$。

  （A1， A3）

导出规则

2.根据合并规则和分解规则，可得引理6.1

引理6.l $X \rightarrow A_1 A_2 \ldots A_k$成立的充分必要条件是$X \rightarrow A_i$成立（$i$=l，2，…，$k$）。

3. 函数依赖闭包

定义6.12　在关系模式*R*<*U*，*F*>中为*F*所逻辑蕴含的函数依赖的全体叫作*F*的闭包，记为F^+。

3. 函数依赖闭包

- 人们把自反律、传递律和增广律称为Armstrong公理系统

Armstrong公理系统

- 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

  /* Armstrong正确

- 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

  /* Armstrong公理够用，完全

Armstrong公理系统

- 要证明完备性，就首先要解决如何判定一个函数依赖是否属于由F根据Armstrong公理系统推导出来的函数依赖的集合
- 如果能够求出这个集合，问题就解决了
- 但是，这个是一个NP完全问题

F的闭包

$F=\{X\rightarrow Y, Y\rightarrow Z\}$　　算是NP完全问题，$X\rightarrow A1A2...An$

$F^+=\{$

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

$X\rightarrow X$, $Y\rightarrow Y$, $Z\rightarrow Z$, $XY\rightarrow X$, $XZ\rightarrow X$, $YZ\rightarrow Y$, $XYZ\rightarrow X$,

$X\rightarrow Y$, $Y\rightarrow Z$, $XY\rightarrow Y$, $XZ\rightarrow Y$, $YZ\rightarrow Z$, $XYZ\rightarrow Y$,

$X\rightarrow Z$, $Y\rightarrow YZ$, $XY\rightarrow Z$, $XZ\rightarrow Z$, $YZ\rightarrow YZ$, $XYZ\rightarrow Z$,

$X\rightarrow XY$, $XY\rightarrow XY$, $XZ\rightarrow XY$, $XYZ\rightarrow XY$,

$X\rightarrow XZ$, $XY\rightarrow YZ$, $XZ\rightarrow XZ$, $XYZ\rightarrow YZ$

$X\rightarrow YZ$, $XY\rightarrow XZ$, $XZ\rightarrow XY$, $XYZ\rightarrow XZ$,

$X\rightarrow ZYZ$, $XY\rightarrow XYZ$, $XZ\rightarrow XYZ$, $XYZ\rightarrow XYZ\}$

3. 函数依赖闭包

为了证明Armstrong公理系统完备性，需要引入以下概念：

定义6.13　设*F*为属性集*U*上的一组函数依赖，*X* ⊆*U*，　X_F^+ ={ *A*\|*X*→*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

关于闭包的引理

根据引理6.1可以进一步得到：

- 引理6.2
  设F为属性集U上的一组函数依赖，X，$Y \subseteq U$，$X \rightarrow Y$能由F 根据Armstrong公理导出的充分必要条件是$Y \subseteq X_F^+$
- 用途
  将判定$X \rightarrow Y$是否能由F根据Armstrong公理导出的问题，就转化为求出X_F^+，判定Y是否为X_F^+的子集的问题（不再是NP完全问题）

实例

U={A, B, C, D}; F={A → B, BC → D};

- A^+ = AB.
- C^+ = C.
- $(AC)^+$ = ABCD

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：

算法6.l

（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$X^{(i)}=U$，则$X^{(i)}$就是X_F^+，算法终止。

（6）若否，则$i=i+1$，返回第（2）步。

算法6.1

对于算法6.1，令$a_i = |X^{(i)}|$，$\{a_i\}$形成一个步长大于1的严格递增的序列，序列的上界是$|U|$，因此该算法最多$|U| - |X|$次循环就会终止。

函数依赖闭包

[例1] 已知关系模式$R<U，F>$，其中

$U=\{A，B，C，D，E\}$;

$F=\{AB→C，B→D，C→E，EC→B，AC→B\}$。

求$(AB)_F^+$。

解 设$X^{(0)}=AB$;

(1)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：$AB→C$，$B→D$。

于是$X^{(1)}=AB∪CD=ABCD$。

函数依赖闭包

(2)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$AB \rightarrow C$，$B \rightarrow D$，$C \rightarrow E$，$AC \rightarrow B$，

于是$X^{(2)}=X^{(1)} \cup BCDE=ABCDE$。

(3)因为$X^{(2)}=U$，算法终止

所以$(AB)_F^+ =ABCDE$。

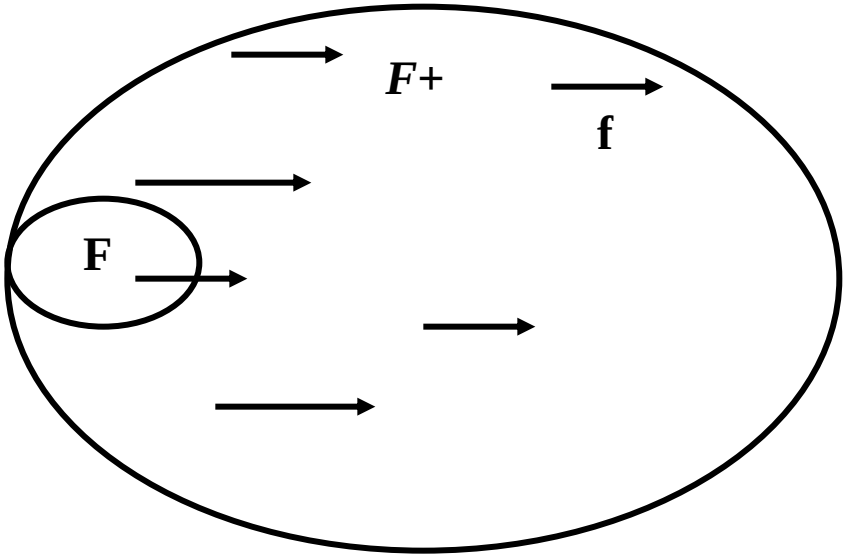
课堂练习

- 例：设有关系模式R(U,F)，其中
- U={A,B,C,D,E,I}
- F={A → D,AB → E,BI → E,CD → I,E → C}
- 请计算（***AE***）$_F^+$

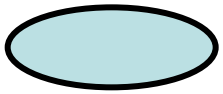
4. Armstrong公理系统的有效性与完备性

- 建立公理系统体系目的：从已知的 f 推导出未知的f
- 明确：1.公理系统推导出来的 f 正确？

  2. F^+中的每一个 f 都能推导出来？



f 不能由F 导出, $f \notin F^+$



4. Armstrong公理系统的有效性与完备性

- 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

  /* Armstrong正确

- 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

  /* Armstrong公理够用，完全

完备性：所有不能用**Armstrong**公理推导出来**f,** 都不为真

若 **f** 不能用**Armstrong**公理推导出来， $f \notin F^+$

有效性与完备性的证明

证明：

1. 有效性

根据定理6.1可以得证

定理 6.l Armstrong推理规则是正确的

Armstrong公理系统推理规则

关系模式$R<U, F>$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→Y为F所蕴含，且$Z \subseteq U$，则XZ→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。

有效性与完备性的证明

证明：

2. 完备性

要证明的题目： 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

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所蕴含

分三步证明：

有效性与完备性的证明

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

证 因为 $V \subseteq X_F^+$，所以有$X \to V$成立；（？）

因为$X \to V$，$V \to W$，于是$X \to W$成立

所以$W \subseteq X_F^+$

(2) 构造一张二维表r，它由下列两个元组构成

可以证明r必是R（U，F）的一个关系，即F^+中的全部函数依赖在 r上成立。

- 引理6.2
  设F为属性集U上的一组函数依赖，X，$Y \subseteq U$，$X \to Y$能由F 根据Armstrong公理导出的充分必要条件是$Y \subseteq X_F^+$

Armstrong公理系统的有效性与完备性(续)

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，$W \subseteq X_F^+$，矛盾。所以*r*必是*R<U，F>*的一个关系。

Armstrong公理系统的有效性与完备性(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 $R<U, F>$ 蕴含

/* 因为 F^+中的全部函数依赖在 r上成立。

- 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$Y \subseteq X_F^+$

5. 函数依赖集等价

定义6.14 如果$G^+=F^+$，就说函数依赖集*F*覆盖*G*（*F*是*G*的覆盖，或*G*是*F*的覆盖），或*F*与*G*等价。

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。

（1）若 $F \subseteq G^+$，则 $X_F^+ \subseteq X_{G+}^+$。

（2）任取 $X \rightarrow Y \in F^+$ 则有 $Y \subseteq X_F^+ \subseteq X_{G+}^+$。(?)

所以 $X \rightarrow Y \in (G^+)^+ = G^+$。即 $F^+ \subseteq G^+$。

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

- 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$Y \subseteq X_F^+$

函数依赖集等价

- 要判定$F \subseteq G^+$，只须逐一对F中的函数依赖$X \rightarrow Y$，考察 Y 是否属于X_{G+} 就行了。因此引理6.3 给出了判断两个函数依赖集等价的可行算法。

6. 最小依赖集

定义6.15　如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集。亦称为最小依赖集或最小覆盖。

(1) F中任一函数依赖的右部仅含有一个属性。

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

最小依赖集

[例2] 对于6.1节中的关系模式*S*<*U*，*F*>，其中：

U={ SNO，SDEPT，MN，CNAME，G },

F={ SNO→SDEPT，SDEPT→MN，
（SNO，CNAME）→G }

设F'={SNO→SDEPT，SNO→MN，
SDEPT→MN，(SNO，CNAME)→G，
(SNO，SDEPT)→SDEPT}

*F*是最小覆盖，而*F*'不是。

因为：*F*'-{SNO→MN}与*F*'等价

F'-{(SNO，SDEPT)→SDEPT}也与*F*'等价

F'-{(SNO，SDEPT)→SDEPT}
∪{SNO→SDEPT}也与*F*'等价

7. 极小化过程

定理6.3　每一个函数依赖集F均等价于一个极小函数依赖集F_m。此F_m称为F的最小依赖集

证:构造性证明，依据定义分三步对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$Y=A_1A_2 \ldots A_k$，$k > 2$，

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。

引理6.I　$X \rightarrow A_1 A_2 \ldots A_k$成立的充分必要条件是$X \rightarrow A_i$成立（$i$=I，2，…，$k$）。

极小化过程

(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$G=F-\{X \rightarrow A\}$等价的充要条件是$A \in X_G^+$

因此F变换前后是等价的。

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$Y \subseteq X_F^+$

极小化过程

(3)逐一取出F中各函数依赖FD_i：$X \rightarrow A$，

设$X=B_1B_2\ldots B_m$，

逐一考查B_i（i=l，2，…，m），

若$A \in (X-B_i)_F^+$，

则以$X-B_i$取代X。

定义6.15 (3) F中不存在这样的函数依赖X→A，X有真子集Z使得F-{X→A}∪{Z→A}与F等价。

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。

极小化过程

由定义，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

- 定理6.3的证明过程 也是求*F*极小依赖集的过程

极小化过程

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

F_{m1}、F_{m2}都是F的最小依赖集：

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

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

- F的最小依赖集F_m不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
- F_{m2}是先验证$B\rightarrow C$得到的结果

例子：计算F的最小函数依赖集

- **例1：**关系模式R<U,F>，其中U={C,T,H,R,S,G},
- F={CS→G,C→T,TH→R,HR→C,HS→R}

请**计算F的最小函数依赖集**

例子：计算F的最小函数依赖集

- ① 利用分解规则，将所有的函数依赖变成右边都是单个属性的函数依赖。由于F的所有函数依赖的右边都是单个属性，故不用分解。

例子：计算F的最小函数依赖集

- ② 去掉F中多余的函数依赖
- A．设CS→G为冗余的函数依赖，则去掉CS→G，得：
- F1={C→T,TH→R,HR→C,HS→R}
- 计算(CS)F1+：
- 设X(0)=CS
- 计算X(1)：扫描F1中各个函数依赖，找到左部为CS或CS子集的函数依赖，找到一个C→T函数依赖。故有X(1)=X(0)∪T=CST。
- 计算X(2)：扫描F1中的各个函数依赖，找到左部为CST或CST子集的函数依赖，没有找到任何函数依赖。故有X(2)=X(1)。算法终止。
- (CS)F1+= CST不包含G，故CS→G不是冗余的函数依赖，不能从F1中去掉。

例子：计算F的最小函数依赖集

- B．设C→T为冗余的函数依赖，则去掉C→T，得：
- F2={CS→G,TH→R,HR→C,HS→R}
- 计算(C)F2+：
- 设X(0)=C
- 计算X(1)：扫描F2中的各个函数依赖，没有找到左部为C的函数依赖。故有X(1)=X(0)。算法终止。故C→T不是冗余的函数依赖，不能从F2中去掉。

例子：计算F的最小函数依赖集

- C．设TH→R为冗余的函数依赖，则去掉TH→R，得：
- F3={CS→G,C→T,HR→C,HS→R}
- 计算(TH)F3+：
- 设X(0)=TH
- 计算X(1)：扫描F3中的各个函数依赖，没有找到左部为TH或TH子集的函数依赖。故有X(1)=X(0)。算法终止。故TH→R不是冗余的函数依赖，不能从F3中去掉。

例子：计算F的最小函数依赖集

- D．设HR→C为冗余的函数依赖，则去掉HR→C，得：
- F4={CS→G,C→T,TH→R,HS→R}
- 计算(HR)F4+：
- 设X(0)=HR
- 计算X(1)：扫描F4中的各个函数依赖，没有找到左部为HR或HR子集的函数依赖。故有X(1)=X(0)。算法终止。故HR→C不是冗余的函数依赖，不能从F4中去掉。

例子：计算F的最小函数依赖集

- E．设HS→R为冗余的函数依赖，则去掉HS→R，得：
- F5={CS→G,C→T,TH→R,HR→C}
- 计算(HS)F5+：
- 设X(0)=HS
- 计算X(1)：扫描F5中的各个函数依赖，没有找到左部为HS或HS子集的函数依赖。故有X(1)=X(0)。算法终止。故HS→R不是冗余的函数依赖，不能从F5中去掉。即：F5={CS→G,C→T,TH→R,HR→C,HS→R}

例子：计算F的最小函数依赖集

- ③ 去掉F5中各函数依赖左边多余的属性（只检查左部不是单个属性的函数依赖）,没有发现左边有多余属性的函数依赖。
- **故最小函数依赖集为：**
  F={CS→G,C→T,TH→R,HR→C,HS→R}

极小化过程

- 极小化过程(定理6.3的证明)也是检验*F*是否为极小依赖集的一个算法
  - 若改造后的*F*与原来的*F*相同，说明*F*本身就是一个最小依赖集

极小化过程

- 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。

候选码的求解

- 对于给定的关系R(A1,A2,…,An)和函数依赖集F，可以将其属性分为4类：
- L类：仅出现在F的函数依赖左边的属性
- R类：仅出现在F的函数依赖右边的属性
- N类：在F的函数依赖左右两边均未出现的属性
- LR类：在F的函数依赖左右两边均出现的属性

候选码的求解

- （1）快速求解候选码的一个充分条件
- 定理：对于给定的关系模式R及其函数依赖集F，若X（$X \in R$）是L类属性，则X必为R的任一候选码的成员

候选码的求解

- 例子：设有关系模式R(A,B,C,D)，其函数依赖集F={***D→B***，　***B→D***，　***AD→B***，***AC→D***}，求R的所有候选码
- 解：可以发现，A和C属性都是L类属性，由前面定理可知，AC必是R的一候选码的成员
- 又因为$(AC)^+$=ACBD，所以，AC是R的候选码

候选码的求解

- 推论：对于给定的关系模式R及其函数依赖集F，若 X_F^+ 包含了R的全部属性，则X必为R的唯一候选码
- 定理：对于给定的关系模式R及其函数依赖集F，如果X（$X \in R$）是R类属性，则X不在任何候选码中
- 定理：设有关系模式R及其函数依赖集F，X（$X \in R$）是N类属性，则X必包含在R的任一候选码中

候选码的求解

推论：对于给定的关系模式R及其函数依赖集F，如果X是R的N类和L类组成的属性集，且 X_F^+包含了R的全部属性，则X是R的唯一候选码

候选码的求解

例：设有关系模式**R(A,B,C,D,E,P)**，**R**的函数依赖集为**F={A→D, E→D, D→B, BC→D, DC→A}**,求**R**的所有候选码。

解：考察**F**发现，**C**、**E**两属性是**L**类属性，故**C**、**E**必在**R**的任何候选码中；

∵P是N类属性，∴P也必在R的任何候选码中。

∵ $(CEP)^+$=ABCDEP，∴CEP是R的惟一候选码。

练习

已知关系模式*R<U，F>*，其中
U={A，B，C，D，E};
F={A→BC，B→D，C D →E，D→C，AC→B}。

1.列出R的码

2.计算B_F+

3.求R的最小覆盖Fm

第六章 关系数据理论

6.1 数据依赖

6.2 规范化

6.3 数据依赖的公理系统

6.4 模式的分解

6.4 模式的分解

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

模式的分解（续）

定义6.16 关系模式R<U,F>的一个分解：

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

$U=U_1 \cup U_2 \cup \ldots \cup U_n$，且不存在 $U_i \subseteq U_j$，F_i 为 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 叫作 F 在属性 U_i 上的投影m

模式的分解（续）

例：SL（Sno， Sdept， Sloc）

F={ Sno→Sdept,Sdept→Sloc,Sno→Sloc}

SL∈2NF

存在插入异常、删除异常、冗余度大和修改复杂等问题

分解方法可以有多种

模式的分解（续）

SL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B

模式的分解（续）

1. SL分解为下面三个关系模式：

SN(Sno)

SD(Sdept)

SO(Sloc)

分解后的关系为：

SN
Sno
95001
95002
95003
95004
95005

SD
Sdept
CS
IS
MA
PH

SO
Sloc
A
B
C

模式的分解（续）

分解后的数据库丢失了许多信息

例如无法查询95001学生所在系或所在宿舍。

如果分解后的关系可以通过自然连接恢复为原来的关系，那么这种分解就没有丢失信息

模式的分解（续）

2. SL分解为下面二个关系模式：

NL(Sno, Sloc)

DL(Sdept, Sloc)

分解后的关系为：

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

DL

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

模式的分解（续）

NL ⋈ DL

Sno	Sloc	Sdept
95001	A	CS
95002	B	IS
95002	B	PH
95003	C	MA
95004	B	IS
95004	B	PH
95005	B	IS
95005	B	PH

模式的分解（续）

NL⋈DL比原来的SL关系多了3个元组

无法知道95002、95004、95005

究竟是哪个系的学生

元组增加了，信息丢失了

第三种分解方法

3. 将SL分解为下面二个关系模式：

ND(Sno, Sdept)

NL(Sno, Sloc)

分解后的关系为：

模式的分解（续）

ND

Sno	Sdept
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

模式的分解（续）

ND ⋈ NL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	CS	A
95005	PH	B

与SL关系一样，因此没有丢失信息

三种模式分解的等价定义

1. 分解具有无损连接性

2. 分解要保持函数依赖

3. 分解既要保持函数依赖，又要具有无损连接性

具有无损连接性的模式分解

- 关系模式R<U,F>的一个分解 ρ={ R_1<U_1,F_1>，R_2<U_2,F_2>， …，R_n<U_n,F_n>}

若R与R1、R2、…、Rn自然连接的结果相等，则称关系模式R的这个分解ρ具有无损连接性（Lossless join）

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

模式的分解（续）

第三种分解方法具有无损连接性

问题:

这种分解方法没有保持原关系中的函数依赖

SL中的函数依赖Sdept→Sloc

没有投影到关系模式ND、NL上

保持函数依赖的模式分解

设关系模式R<U,F>被分解为若干个关系模式

$R_1<U_1,F_1>$，$R_2<U_2,F_2>$，…，$R_n<U_n,F_n>$

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

第四种分解方法

将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法就保持了函数依赖。

模式的分解（续）

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

模式的分解（续）

1. SL分解为下面三个关系模式：

SN(Sno)　SD(Sdept)　SO(Sloc)

2.　SL分解为下面二个关系模式：

NL(Sno, Sloc)　DL(Sdept, Sloc)

3. 将SL分解为下面二个关系模式：

ND(Sno, Sdept)　NL(Sno, Sloc)

4.将SL分解为下面二个关系模式：

ND(Sno, Sdept)　DL(Sdept, Sloc)

模式的分解（续）

第一种分解方法既不具有无损连接性，也未保持函数依赖，它不是原关系模式的一个等价分解

第二种分解方法未保持函数依赖，不具有无损连接性

第三种分解方法具有无损连接性，但未持函数依赖

第四种分解方法既具有无损连接性，又保持了函数依赖

模式分解的定义

- 设$p=\{R_1<U_1,F_1>,\ldots,R_K<U_K,F_K>\}$是$R<U,F>$的一个分解，r是$R<U,F>$的一个关系。

  定义$m_p(r)=\bowtie_{i=1}^{k}\pi_{Ri}(r)$

$M_p(r)$是r在p中各关系模式上投影的连接

$r_i=\pi_{Ri}(r)=\{t.Ui|t \in r\}$

模式的分解（续）

引理6.4 设R<U,F>是一个关系模式，

ρ={ R_1<U_1,F_1>，R_2<U_2,F_2>，…，R_k<U_k,F_k>}是R的一个分解，r是R的一个关系，则

（1）$r \subseteq m_\rho(r)$

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

（3） $m_\rho(m_\rho(r))= m_\rho(r)$

无损分解的定义

定义6.18 $p=\{R_1<U_1,F_1>, R_2<U_2,F_2>, \ldots, R_k<U_k,F_k>\}$是$R<U,F>$的一个分解

若对R中的任何一个关系r均有$r= m_p(r)$成立，则称分解p具有无损连接性。

无损分解的判定算法

算法6.2 判断一个分解的无损连接性

设$\rho=\{R_1<U_1,F_1>, R_2<U_2,F_2>, \ldots, R_k<U_k,F_k>\}$是$R<U,F>$的一个分解，$U=\{A_1, A_2, \ldots, A_n\}$，$F=\{FD_1, FD_1, \ldots, FD_p\}$

1.建立一个n列k行的表，每列对应一个属性，每行对应分解中的一个关系模式。若属性A_j属于U_i，则在j列i行的交叉处天上a_j,否则填上b_{ij};

无损分解的判定算法

2.对应每个FD_i (FD_i为$X_i \rightarrow A_{li}$)做下列操作：

找到X_i所对应的列中具有相同符号的那些行。考察这些行的l_i列，若其中有a_{li}则全部改为a_{li}；否则全部改为b_{mli};m是这些行的行号最小值

如在某次更改之后，有一行成为$a_1, a_2,\ldots, a_n$,则算法终止,P具有无损连接性,否则P不具有无损连接性

3.比较扫描前后，表有无变化，如有变化，则返回第2步，否则算法终止

分解示例

例： 已知关系模式*R<U，F>*，其中

U={*A*，*B*，*C*，*D*，*E*}；

F={*AB*→*C*，*C*→*D*，*D* →*E*}。

R的一个分解为R1(A, B, C), R2(C, D), R3(D, E). 判断分解是否是无损连接。

课堂练习

例4.4.3：关系模式R(SAIP)，F={S→A, SI→P}，ρ={R_1(SA)，R_2(SIP)}，检验分解是否为无损联接。

	S	A	I	P
R_1	a_1	a_2	b_{13}	b_{14}
R_2	a_1	b_{22}	a_3	a_4

⇨

	S	A	I	P
R_1	a_1	a_2	b_{13}	b_{14}
R_2	a_1	a_2	a_3	a_4

通过修改发现表中第二行元素变为$a_1, a_2, \cdots, a_n$，分解是无损联接。

无损连接例子（1）

例4.4.4：已知关系模式R(ABCDE)及函数依赖集F={A→C, B→C, C→D, DE→C, CE→A}
验证分解ρ={R_1(AD), R_2(AB), R_3(BE), R_4(CDE), R_5(AE)}是否为无损联接。

无损连接例子（2）

例4.4.4：已知关系模式R(ABCDE)及函数依赖集F={A→C, B→C, C→D, DE→C, CE→A}
验证分解ρ={R_1(AD), R_2(AB), R_3(BE), R_4(CDE), R_5(AE)}是否为无损联接。

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5

⇒

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	b_{54}	a_5

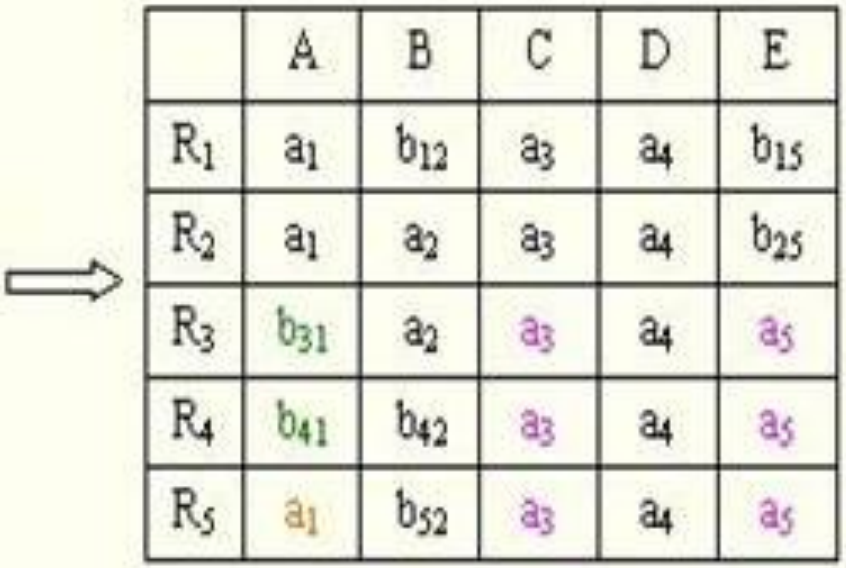
无损连接例子（3）

例4.4.4：已知关系模式R(ABCDE)及函数依赖集F={A→C, B→C, C→D, DE→C, CE→A}
验证分解ρ={R_1(AD), R_2(AB), R_3(BE), R_4(CDE), R_5(AE)}是否为无损联接。

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{13}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	b_{54}	a_5

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	a_4	a_5

无损连接例子（4）



	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	b_{31}	a_2	a_3	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	a_1	a_2	a_3	a_4	a_5
R_4	a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

通过修改发现表中第三行元素变为$a_1, a_2, \cdots, a_n$，分解是无损联接。

定理6.5

对于分解为两个关系模式的情况，有如下的定理：

设$\rho=\{R_1, R_2\}$是关系模式R的一个分解，F是R的函数依赖集，那么ρ是R（关于F）的无损分解的充分必要条件是：

$(R_1 \cap R_2) \rightarrow R_1 - R_2 \in F^+$ 或 $(R_1 \cap R_2) \rightarrow R_2 - R_1 \in F^+$

例4.4.5：关系模式$R(SAIP)$，$F=\{S \rightarrow A, SI \rightarrow P\}$，$\rho=\{R_1(SA), R_2(SIP)\}$

检验分解是否为无损联接？

解：$R_1 \cap R_2 = SA \cap SIP = S$ $R_1 - R_2 = SA - SIP = A$，$S \rightarrow A \in F$，所以ρ是无损分解。

保持函数依赖分解的定义

定义6.19 $p=\{R_1<U_1,F_1>, R_2<U_2,F_2>, \ldots, R_k<U_k,F_k>\}$是$R<U,F>$的一个分解

若$F^+=(\bigcup_{i=1}^{i=k} F_i)^+$则分解p保持函数依赖

模式分解的事实

- 若要求分解保持函数依赖，则分解可以达到**3NF,**但不一定达到**BCNF;**
- 若要求分解既要保持函数依赖，又要保持无损连接则分解可以达到**3NF,**但不一定达到**BCNF;**
- 若要求分解保持无损连接，那一定能达到**4NF;**

模式分解算法

[算法6.3]（合成法）转换为3NF的保持函数依赖的分解。

- （1）对R〈U，F〉中的函数依赖集F进行“极小化处理”（处理后得到的依赖集仍记为F）
- （2）找出不在F中出现的属性，把这样的的属性构成一个关系模式。把这些属性从U中去掉，剩余的属性仍记为U。
- （3）若有$X \rightarrow A \in F$，且XA=U，则ρ={R}，算法终止。
- （4）否则，对F按具有相同左部的原则分组（假定分为k组），每一组函数依赖，所涉及的全部属性集U_i。

  若$U_i \leq U_j$ (i≠j)就去掉Ui。

例子：转换为3NF的保持函数依赖的分解

- **例1：**关系模式R<U,F>，其中U={C,T,H,R,S,G},
- F={CS→G,C→T,TH→R,HR→C,HS→R}，将其分解成3NF并保持函数依赖。

例子：转换为3NF的保持函数依赖的分解

- **(一)计算F的最小函数依赖集**
- 最小函数依赖集为：
  F={CS→G,C→T,TH→R,HR→C,HS→R}
- **(二)由于R中的所有属性均在F中都出现，所以转下一步。**
- **(三)对F按具有相同左部的原则分为：**
- R1=CSG，R2=CT，R3=THR，R4=HRC，R5=HSR。
- 所以ρ={R1(CSG),R2(CT),R3(THR),R4(HRC),R5(HSR)}。

模式分解算法

[算法6.4] 转换为3NF既有无损连接性又保持函数依赖的分解。

- **（1）设X是R〈U，F〉的码。R〈U，F〉已由算法6.3分解为ρ={R1〈U1，F1〉，R2〈U2，F2〉，…，Rk〈Uk，Fk〉}，**
- **令T＝ ρ∪{R*〈X，Fx〉}。**
- **（2）若有某个Ui，X⊆Ui，将R*〈X，Fx〉从中T去掉，**
- **（3）T就是所求的分解。**

例子：转换为3NF既有无损连接性又保持函数依赖的分解

- **例2：**关系模式R<U,F>，其中：U={C,T,H,R,S,G}，
- F={CS→G,C→T,TH→R,HR→C,HS→R}，分解成3NF并保持无损连接和函数依赖。
-
- **解：**(1) 根据上**例例1**，得到3NF并保持函数依赖的分解如下：
- σ={ R1(CSG),R2(CT),R3(THR),R4(HRC),R5(HSR) }。
- (2) 而HS是原模式的码，所以τ={CT，CSG，CHR，HSR，HRT，HS}。由于HS是模式HSR的一个子集，所以消去HS后的分解{CT，CSG，CHR，HSR，HRT}就是具有无损联接性和保持函数依赖性的分解，且其中每一个模式均为3NF。

模式分解算法

- [算法6.5]转换为BCNF的无损连接分解（分解法）。
- （1）令ρ={R〈U，F〉}
- （2）检查ρ中各关系模式是否均属于BCNF。若是，则算法终止。
- （3）设ρ中R_i〈U_i，F_i〉不属于BCNF，那么必有$X \rightarrow A \in F_i^+$（$A \notin X$），且X非Ri的码。因此，XA是$U_i$的真子集。对$R_i$进行分解：$\sigma$={S1，S2}，$US_1$=XA，$US_2=U_i-\{A\}$，以$\sigma$代替Ri〈$U_i$，$F_i$〉返回第（2）步
- 由于U中属性有限，因而有限次循环后算法6.5一定会终止。

作业

设有关系模式R(C，T，S，N，G)，其中C代表课程，T代表教师的职工号，S代表学生号，N代表学生的姓名，G代表分数（成绩）。其函数依赖集F=｛C→ T，CS→G ，S→ N｝，即每一门课由一名教师讲授，每个学生每门课只有一个成绩，学生的学号决定学生的姓名。

1. 将该模式分解成既符合BCNF，又具有无损连接的若干关系模式
2. 将R分解成R_1（C，T，S，G）和R2（C，S，N，G），试判断它们各符合第几范式。
3. 判断2的分解是否是无损连接

注：需给出详细的求解过程

附件：主讲教师和助教

主讲教师：林子雨

单位：厦门大学计算机科学系
E-mail: ziyulin@xmu.edu.cn
个人网页：http://www.cs.xmu.edu.cn/linziyu
数据库实验室网站：http://dblab.xmu.edu.cn

助教：谢荣东

单位：厦门大学计算机科学系数据库实验室2014级硕士研究生
E-mail: xrdxmu@sina.com

助教：薛倩

单位：厦门大学计算机科学系数据库实验室2015级硕士研究生
E-mail: xueqian_victoria@163.com


Thank You!