第六章
关系的规范化设计

第六章 关系的规范化设计

教学内容：

函数依赖；规范化设计；数据依赖的公理系统；模式分解

要求掌握：

1、函数依赖和多值依赖；
2、各级范式的性质；
3、低级范式向高级范式的转化。

教学重点及难点：

函数依赖、范式

第一节 关系模式设计问题的提出

如何设计一个合理的关系数据库模式？

例：设计一个关系数据模式以存放学生各门课考试成绩。

有若干答案：

STUDENT关系：SNO,SNAME
COURSE关系：CNO,CNAME
GRADE关系：SNO,CNO,GRADE ①

GRADE关系：SNO,SNAME,CNO,GRADE
COURSE关系：CNO,CNAME ②

关系模式设计问题的提出

GRADE关系:SNO,CNO,CNAME,GRADE
STUDENT关系：SNO,SNAME ③

GRADE关系:SNO,SNAME,CNO,CNAME,
GRADE ④

我们称之为**泛关系模式**，它把现实问题的所有属性组成一个关系模式。

泛关系模式会出现哪些问题?

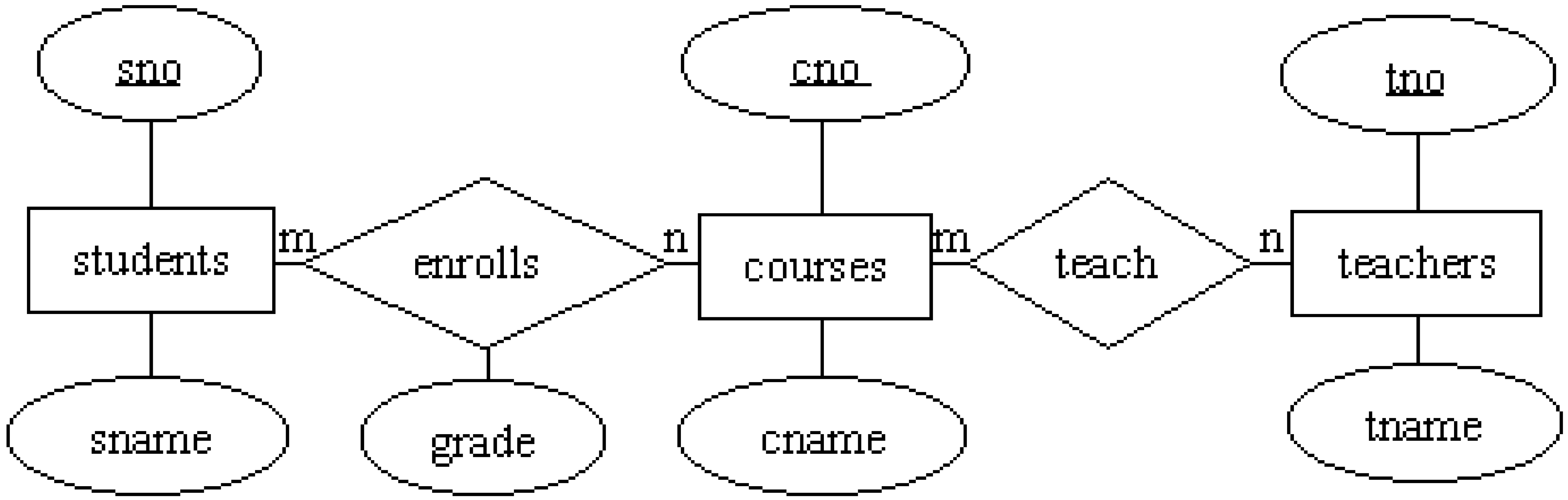
sno	sname	cno	cname	grade
s1	李立	c1	DB	78
s1	李立	c3	OS	87
s2	丁惠	c1	DB	90
s2	丁惠	c2	DS	83
s2	丁惠	c3	OS	77

(1) 数据冗余 (2) 更新异常 (3) 插入异常 (4) 删除异常

关系数据库设计的核心：关系模式设计

按照一定的原则从数量众多而又相互关联的数据中，构造出一组既能较好地反映现实世界，而又有良好的操作性能的关系模式。

解决问题的方法：分解关系



简单教学管理的实体联系模型 ER 图

解1：sct(sno, cno, tno, sname, grade, cname, tname)

解2：

students(<u>sno</u>, sname)

courses(<u>cno, tno</u>, cname)

teachers(<u>tno, cno</u>, tname)

sc(<u>sno, cno</u>, grade)

解3：

students(<u>sno</u>, sname)

courses(<u>cno</u>, cname)

teachers(<u>tno</u>, tname)

sc(<u>sno, cno</u>, grade)

teaching(<u>tno, cno</u>)

Students

Sno	Sname
S1	赵民
S2	李军
S3	陈江
S4	魏致
S5	乔远

Teachers

Tno	Tname
T1	彭
T2	杨
T3	刘
T4	张

SC

Sno	Cno	Grade
S1	C1	90
S1	C2	90
S1	C3	85
S1	C4	87
S2	C1	90
S3	C1	75
S3	C2	70
S3	C4	56
S4	C1	90
S4	C2	85
S5	C1	95
S5	C4	80

Courses

Cno	Cname
C1	OS
C2	DS
C3	C++
C4	DB

Teaching

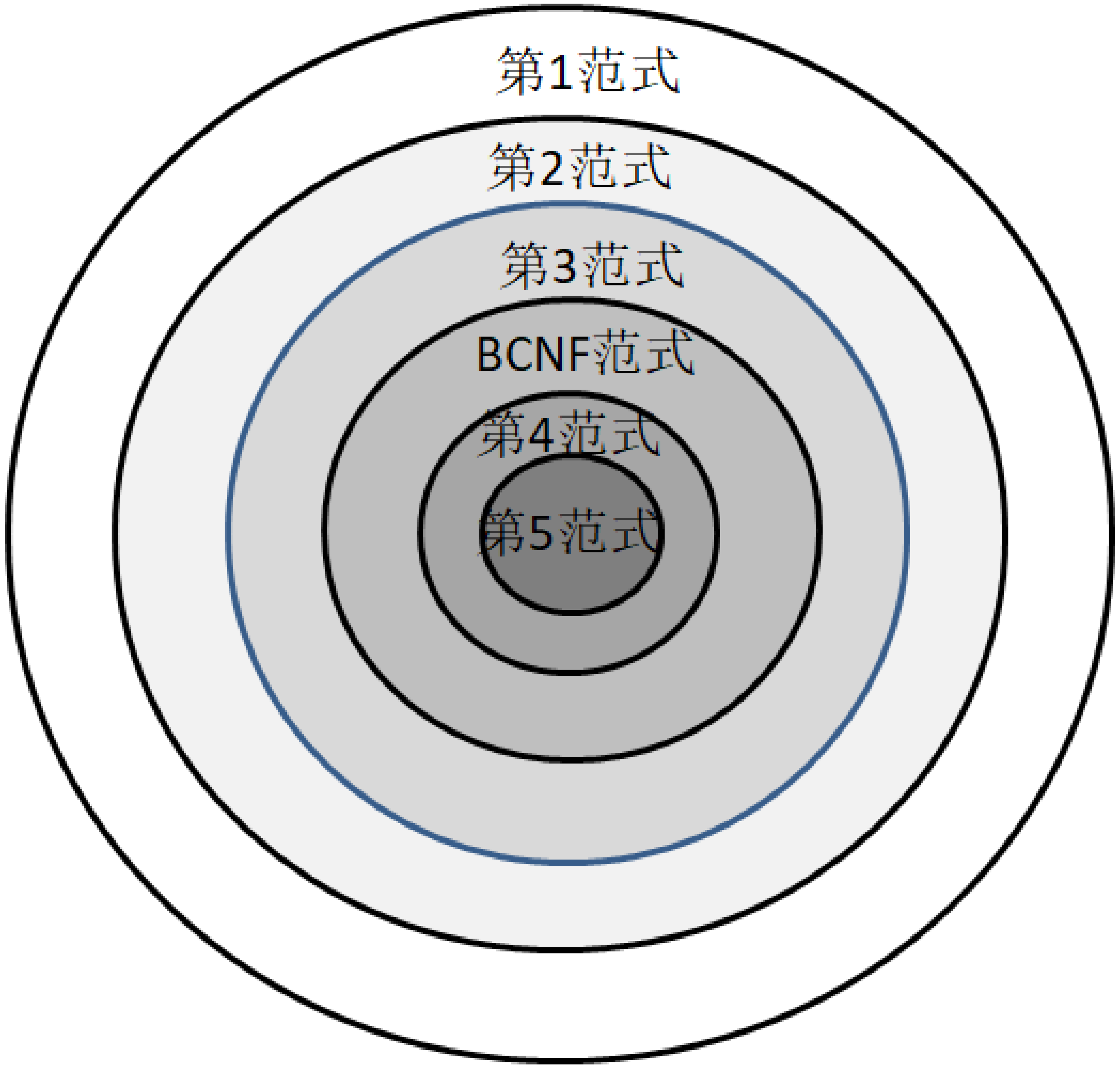
Cno	Tno
C1	T1
C1	T4
C2	T2
C3	T3
C4	T4

本章要解决的主要问题：

1. 怎样**评价**关系模式的优劣
2. 怎样将关系模式**分解**为一组较**理想**的关系模式

第二节　规范化

- 函数依赖
- 范式
- 2NF
- 3NF
- BCNF
- 多值依赖
- 4NF



函数依赖

数据依赖是通过一个关系中**属性之间的相等与否**体现出来的数据间的相互关系，分函数依赖和多值依赖。

函数依赖(Functional Dependency，FD)是数据依赖的一种，它反映**属性或属性组之间相互依存、互相制约的关系**，即反映现实世界的约束关系。

（1） 函数依赖的定义

设R(U)是属性U上的一个关系模式，X和Y均为U={A_1，A_2，…，A_n}的子集，r为R的任一关系，如果对于r中的任意两个元组u和v，只要有u[X]=v[X]，就有u[Y]=v[Y]，则称**X函数决定Y**，或称**Y函数依赖于X**，记为**X→Y**。

例： 1.sno→sname, ✓

cno→cname,

(sno,cno)→grade

2.tno→cno, ✗

sno→tname

可以根据语义来确定函数依赖的存在与否。

(2）属性间的联系决定函数依赖关系

设X、Y均是U的子集

1. X和Y间联系是1:1，则X→Y、Y→X。

2. X和Y间联系是M:1(M>1)，则X→Y。

3.X和Y间联系是M:N(M,N>1)，则X、Y间不存在函数依赖。

例：

SCX(SNO,SNAME,AGE,SEX,CNO,CNAME,GRADE)

$$SNO \rightarrow SNAME$$

$$SNO \rightarrow AGE$$

$$SNO \rightarrow SEX$$

$$(SNO,CNO) \rightarrow GRADE$$

$$CNO \rightarrow CNAME$$

$$SNO \nrightarrow GRADE$$

(3) 自反性、增广性和传递性

设U是关系模式R的属性集，F是R上成立的只涉及到U中属性的函数依赖集，则：

自反性：若$Y \subseteq X \subseteq U$，则$X \rightarrow Y$在R上成立

增广性：若$X \rightarrow Y$在R上成立，且$Z \subseteq U$，则$XZ \rightarrow YZ$在R上成立

传递性：若$X \rightarrow Y$和$Y \rightarrow Z$在R上成立，则$X \rightarrow Z$在R上成立

（4）完全函数依赖和部分函数依赖

在R(U)中，如果X→Y，并且对于X的任何真子集X'都不存在X'→Y，则称**Y完全函数依赖于X**，记作$X \xrightarrow{F} Y$；

否则，如果X→Y，且X中存在一个真子集X'，使得X'→Y成立，则**Y部分函数依赖于X**，记作$X \xrightarrow{P} Y$。

例：

$$(sno,cno) \xrightarrow{P} cname$$

$$(sno,cno) \xrightarrow{F} tname$$

$$cno \xrightarrow{F} cname$$

$$(sno,cno) \xrightarrow{F} g$$

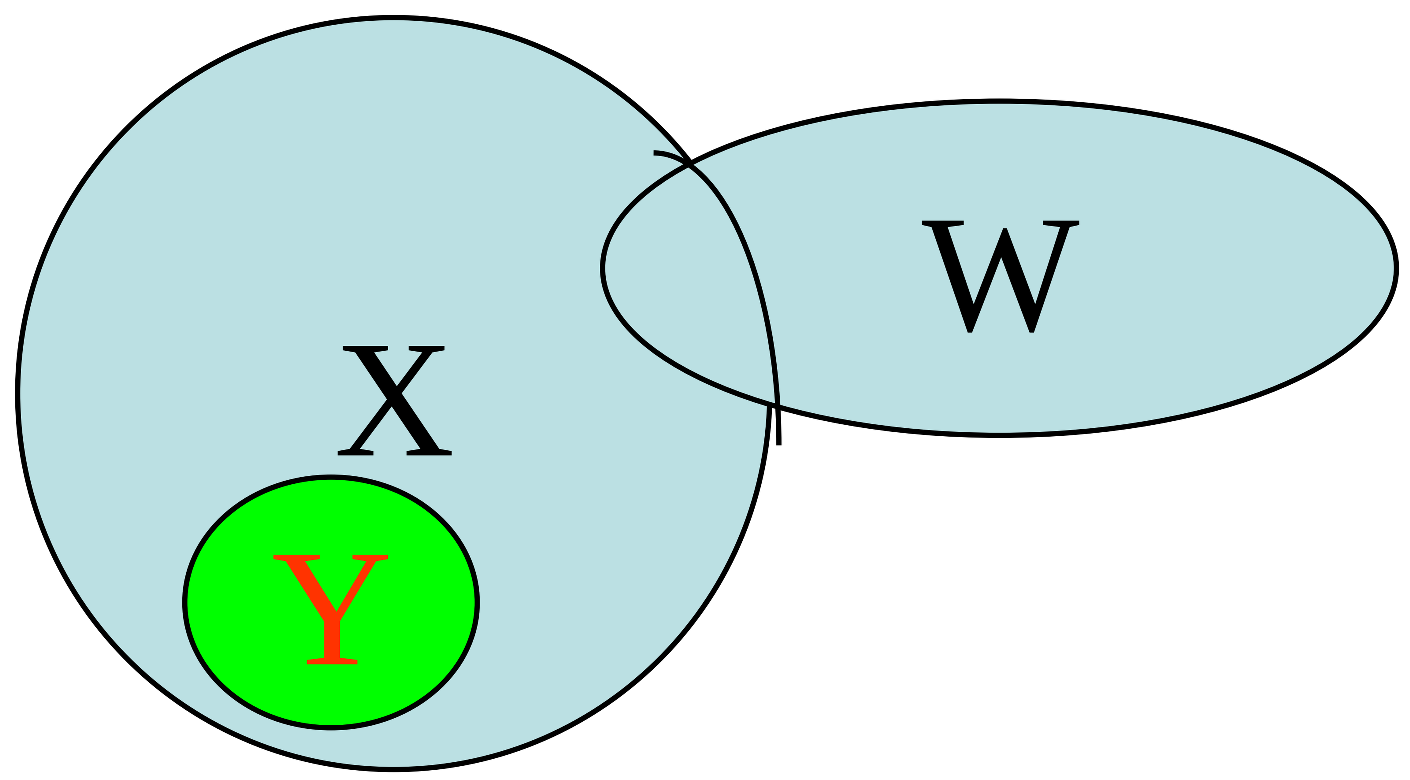
5）平凡函数依赖和非平凡函数依赖

设X、Y均为某关系上的属性集，且X→Y

1）若Y是X的子集，则称X→Y为**平凡函数依赖**；

2）若Y不是X的子集，则称X→Y为**非平凡函数依赖**。(一般都指的是非平凡函数依赖)

例：设X、Y、W为关系R中的三个属性组，属性关系如下图所示，问X→Y、X→W、W→Y各属于上述何种函数依赖。



X→Y为平凡函数依赖

X→W、W→Y为非平凡函数依赖

范式

关系模式满足的确定约束条件称为**范式**，根据满足约束条件的级别不同，范式由低到高分为1NF、2NF、3NF、BCNF、4NF、5NF等。

关系模式的规范化：把一个低一级范式的关系模式分解为高一级范式的关系模式的过程。

第一范式（1NF)

关系模式的所有域为简单域，其元素不可再分，即**属性不能再分**。

例1： $A_1,A_2,A_3,\ldots,\ A_k,\ldots,A_n$

$A_{k1}\quad A_{k2}$

例2： 工资(工号,姓名,工资(基本工资,年绩津贴,供暖补贴))

第一范式（1NF）

★不满足1NF的关系称为非规范化关系

★关系数据模型不能存储上述例子（非规范化关系）

★转化方法：

1) $A_1,A_2,A_3,\cdots,\ A_{k1},\ A_{k2},\ \cdots,A_n$

2)工资(工号,姓名,基本工资,年绩津贴,供暖补贴)

第一范式（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

第一范式（1NF）

例:关系模式 S-L-C(Sno, Sdept, Sloc, Cno, Grade)，Sloc为学生住处，假设每个系的学生住在同一个地方

- 函数依赖包括：Sno → Sdept

  Sno → Sloc

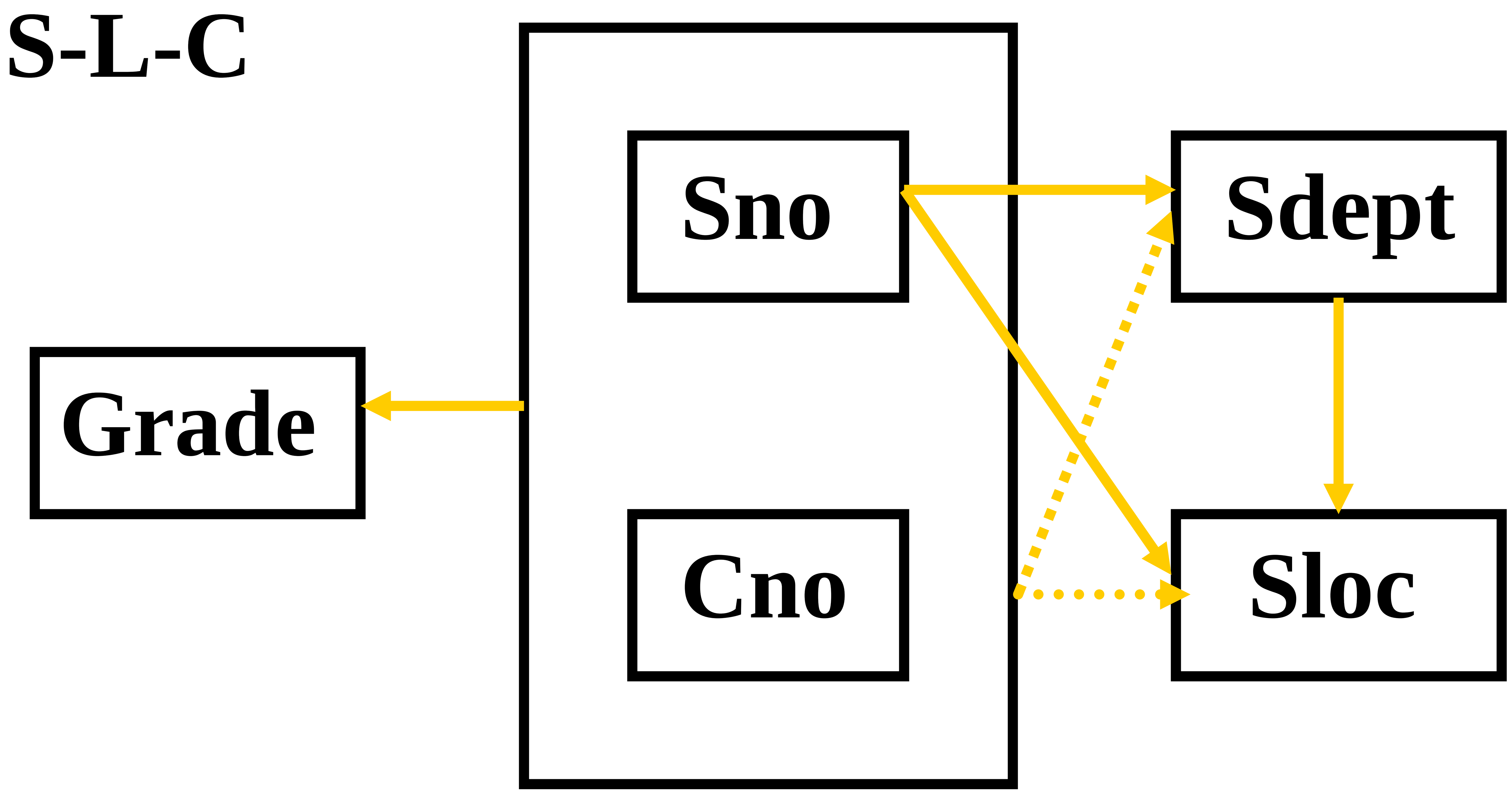
  Sdept → Sloc

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

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

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

函数依赖图



- S-L-C的码为(Sno, Cno)
- S-L-C满足第一范式
- 非主属性Sdept和Sloc部分函数依赖于码(Sno, Cno)

S-L-C不是一个好的关系模式

S-L-C(Sno, Sdept, Sloc, Cno, Grade)

(1) 插入异常

(2) 删除异常

(3) 数据冗余度大

(4) 修改复杂

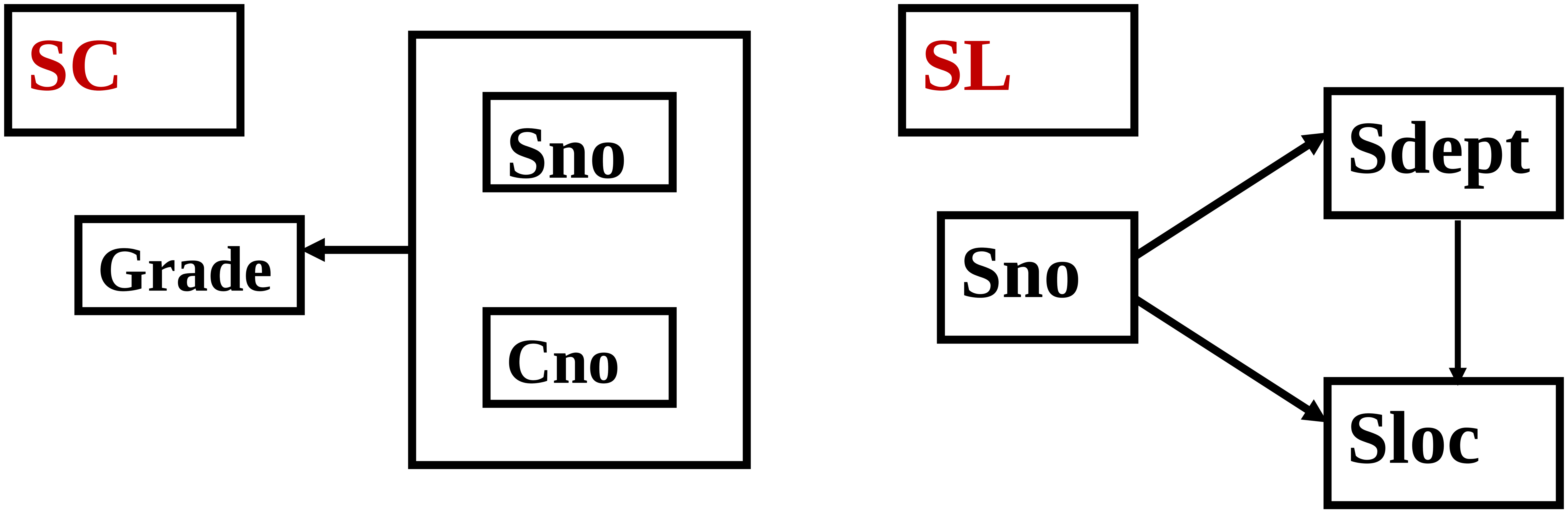
S-L-C不是一个好的关系模式

- 原因：Sdept、 Sloc部分函数依赖于码
- 解决方法：S-L-C 分解为两个关系模式，以消除这些部分函数依赖

SC(<u>Sno, Cno</u>, Grade)

SL(<u>Sno</u>, Sdept, Sloc)

函数依赖图



- 关系模式SC的码为（Sno，Cno）
- 关系模式SL的码为Sno
- 这样**非主属性对码都是完全函数依赖**

第二范式（2NF）

给定关系模式R及其上的函数依赖集F，如果R的任何一个**非主属性**都**完全函数依赖**于码，则称R满足第二范式，简记为$R \in 2NF$。

示例

例1：

S-L-C(Sno, Sdept, Sloc, Cno, Grade) $\in$1NF

S-L-C(Sno, Sdept, Sloc, Cno, Grade) $\notin$2NF

SC(Sno, Cno, Grade) $\in$ 2NF

SL(Sno, Sdept, Sloc) $\in$ 2NF

示例

SCT(SNO,CNO,CNAME,GRADE,TNAME,SALARY)
//一门课只能由一个老师教；教师名唯一
是否满足2NF？不满足的话如何分解？

候选码:(SNO,CNO)
F={(SNO,CNO)→GRADE，CNO→CNAME,
CNO→TNAME, TNAME→SALARY}

是否存在非主属性对码的不完全依赖?
存在：$(SNO,CNO) \xrightarrow{P} CNAME$
$(SNO,CNO) \xrightarrow{P} TNAME$

非2NF!

解决方法

解决的方法：是用投影分解把关系模式SCT分解为两个关系模式

SCT(<u>SNO,CNO</u>,CNAME,GRADE,TNAME,SALARY)

分解：SC(<u>SNO, CNO</u>, GRADE),

CT(<u>CNO</u>, CNAME, TNAME, SALARY)

分解后得到的两个关系模式均为2NF

分解为2NF后，是否都不存在前述的四种问题？

回顾例1 S-L-C(Sno, Sdept, Sloc, Cno, Grade)

分解为：

SC(<u>Sno, Cno</u>, Grade) $\in$ 2NF

SL(<u>Sno</u>, Sdept, Sloc) $\in$ 2NF

在SC中，不存在前述的四种问题；

但在SL中，仍然存在着前述的四种问题。

说明2NF还不够好。

回顾例2

SC(<u>SNO,CNO,</u>CN,GRADE,TNAME,SALARY)

分解为：

SC(<u>SNO,CNO</u>,GRADE),

CT(<u>CNO</u>,CN,TNAME,SALARY)

在SC中，不存在前述的四种问题；
但在CT中，仍然存在着前述的四种问题。
说明2NF还不够好。

传递函数依赖

在R(U)中，如果X→Y，Y→Z，并且Y↛X，那么称X→Z是**传递函数依赖**，即**Z传递函数依赖于X**。

SL(<u>Sno</u>, Sdept, Sloc)

Sno → Sdept，Sdept → Sloc

CT(<u>CNO</u>, CN, TNAME, SALARY)

CNO→TNAME, TNAME→SALARY

第三范式（3NF）

给定关系模式R及其上的函数依赖集F，如果R的任何一个非主属性都不传递函数依赖于它的任何一个候选码，则称R满足第三范式，简记为R$\in$ 3NF。

第三范式（3NF）

△**定理：一个3NF的关系(模式)必定是2NF的(3NF$\subseteq$2NF)。**

证明：

反证法。

如果一个关系满足3NF，不满足2NF，说明存在非主属性A_k部分函数依赖于一个候选码(A_1,A_2)，即$A_1 \to A_k$；则存在非主属性A_k传递函数依赖于(A_1,A_2)，即$(A_1,A_2) \to A_1$，$A_1 \to A_k$，则该关系不满足3NF，与假设矛盾。

例：CT(CNO, CNAME, TNAME, SALARY)

F={CNO→TNAME, CNO→CNAME, TNAME→SALARY}

问题：插入和删除异常

原因：存在非主属性SALARY传递函数依赖于码，故CT∈2NF，不是3NF

分解：C(CNO,CNAME,TNAME),

T(TNAME, SALARY)均是3NF

例2 SP={<u>SNO</u>,SNAME,CITY,PROVINCE}

F={SNO→SNAME,SNO→CITY,

CITY→ PROVINCE}

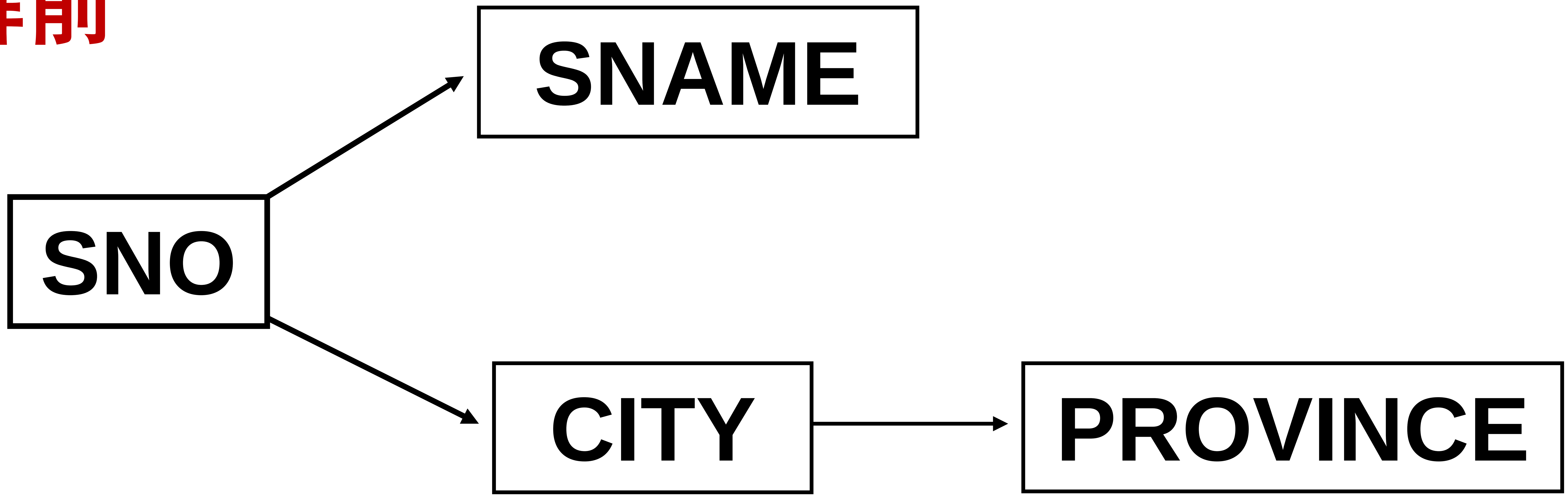
存在非主属性PROVINCE对候选码的传递函数依赖，故SP为2NF,不是3NF。

分解：

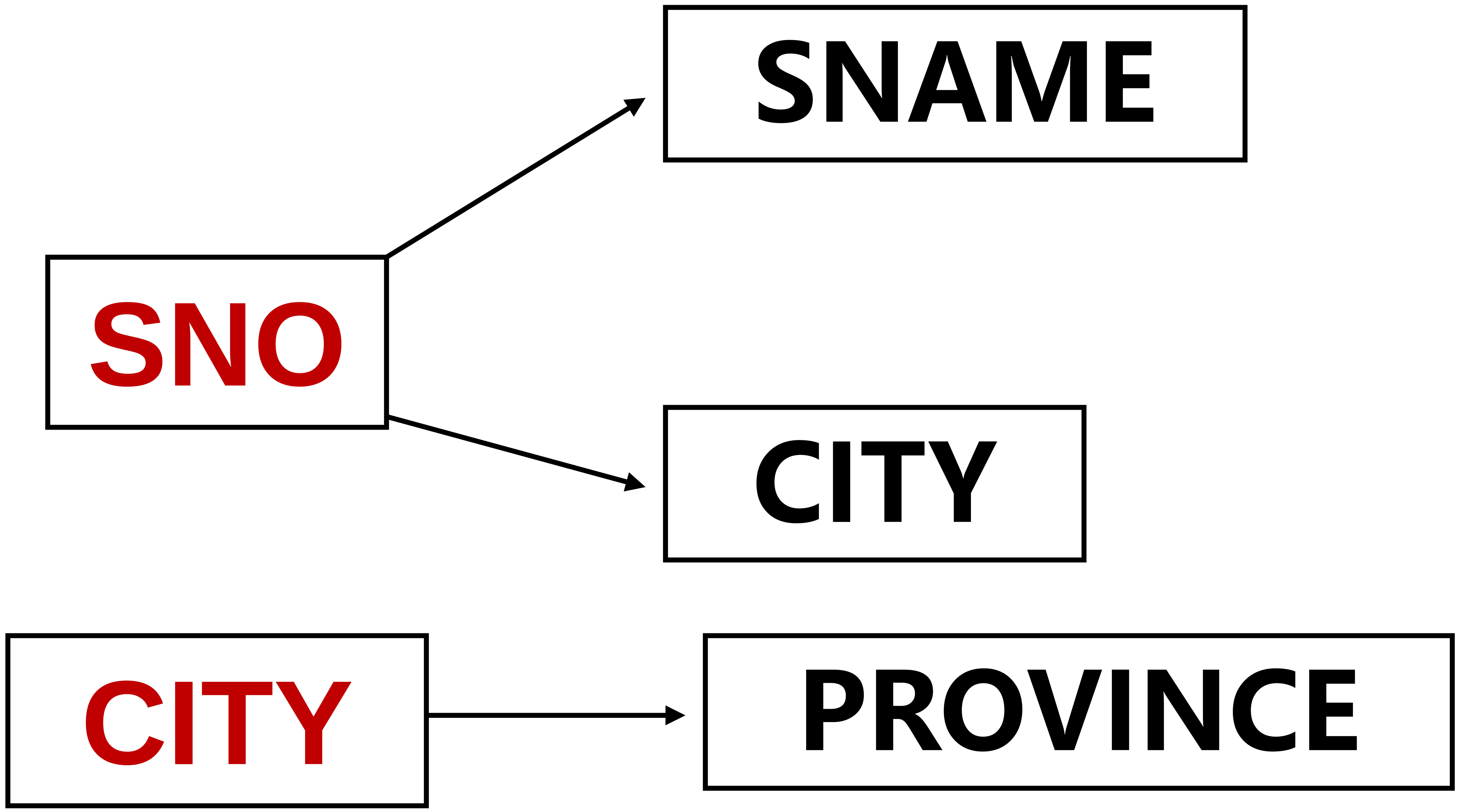
SP1={<u>SNO</u>, SNAME, CITY}

SP2={<u>CITY</u>, PROVINCE}均为3NF。

分解前



分解后



示例

例3： R={SNO,TNO,CNO}

一名教师只能上一门课，一门课可由多个教师上

F={TNO→CNO,(SNO,TNO)→CNO,

(SNO,CNO)→TNO}

候选码为（SNO,TNO）和（SNO,CNO）

R中没有非主属性，故R为3NF

3NF是否就一定不存在以前的四种问题呢？

回顾例3：

R={SNO,TNO,CNO}

F={TNO→CNO, (SNO, CNO)→TNO, (SNO,TNO)→ CNO}

候选码为（SNO,TNO）和（SNO,CNO）

仍然存在问题：例如要插入某个教师能讲授某门课的信息，而又没有学生选修该课时，会有插入异常。

BCNF（改进的3NF）

给定关系模式R及其上的函数依赖集F，如果F中每个**非平凡函数依赖**$X \rightarrow Y$的左部（决定因素）X中必含有候选键（码），则称R满足Boyce/Codd范式，简记为$R \in BCNF$

$R \in 1NF$，若$X \rightarrow Y$且$Y \not\subseteq X$，X中必含有候选键（码），则$R \in BCNF$

BCNF的内涵

1. 非主属性对码完全函数依赖
2. 非主属性不传递函数依赖于任何一个码
3. 主属性对不含它的码完全函数依赖
4. 主属性不传递函数依赖于任何一个码
5. 没有属性完全函数依赖于一组非主属性

BCNF

定理： BCNF满足3NF

(BCNF⊆3NF⊆2NF⊆1NF)

证明：

反证法。如果一个关系满足BCNF，不满足3NF，说明存在非主属性A_k传递函数依赖于一个候选码(A_1,A_2)，即$(A_1,A_2)\rightarrow A_3$，$A_3\rightarrow A_k$，且$A_3 \not\rightarrow (A_1,A_2)$；由于$A_3$一定不是（包含）码，因此该关系不满足BCNF，与假设矛盾。

关系模式规范化的基本步骤

1NF

↓ 消除非主属性对码的部分函数依赖

2NF

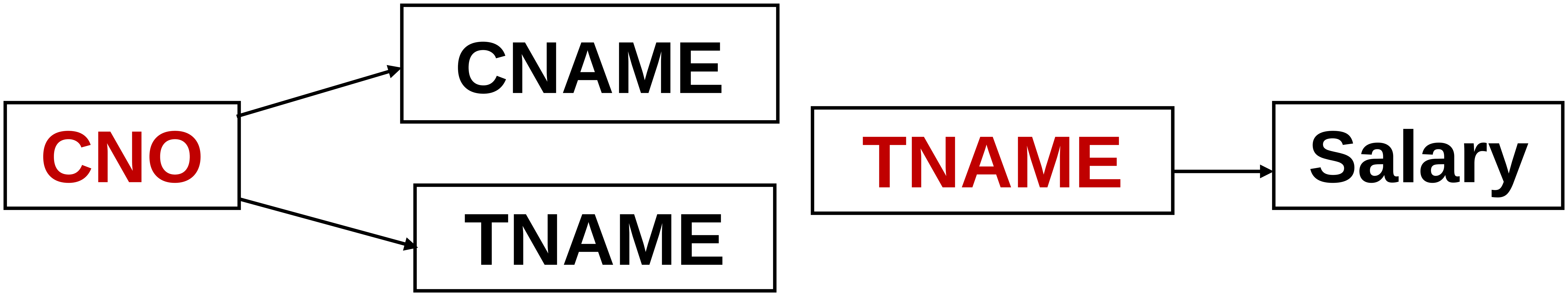
↓ 消除非主属性对码的传递函数依赖

3NF

↓ 消除主属性对码的部分和传递函数依赖

BCNF

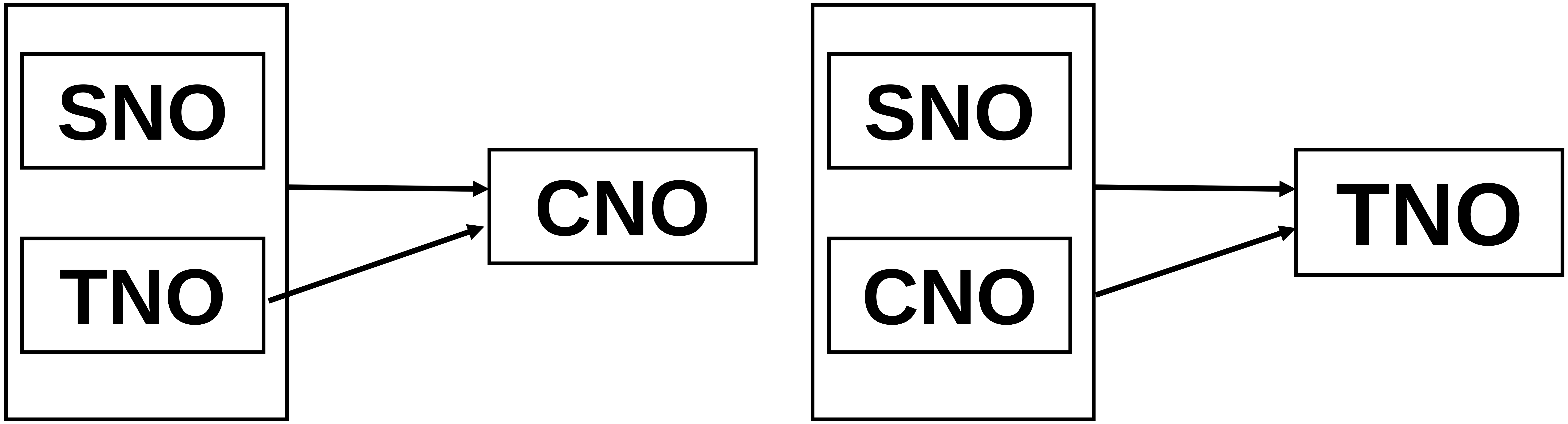
例1：C(CNO,CNAME,TNAME)
T(TNAME, SALARY)
以上两个关系是否满足BCNF？



可以由每个函数依赖的左部均包含码，判断出C和T均属于BCNF。

例2：R={SNO,TNO,CNO}是否满足BCNF？

F={TNO→CNO,(SNO,TNO)→CNO,
(SNO,CNO)→TNO}



由TNO→CNO知，R不是BCNF。

分解：SC(SNO,CNO), TC(TNO,CNO)

多值依赖

函数依赖有效地表达了属性之间的多对一联系，但不能表达属性之间的一对多联系，更不能表达属性之间的多对多联系。

课程C	教师T	参考书B
物理	李勇	普通物理
物理	李勇	微分方程
物理	王军	普通物理
物理	王军	微分方程
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	微分方程
数学	张平	高等代数

该模式的候选码是 (C,T, B)，即全码是BCNF，但依然存在各种异常。

1）多值依赖的定义

设R(U)是属性U上的一个关系模式，X、Y、Z均为U={A_1，A_2，…，A_n}的子集，并且Z=U-X-Y，用小写字母x、y、z分别代表属性集X、Y、Z 的值，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**

课程C	教师T	参考书B
物理	李勇	普通物理
物理	李勇	微分方程
物理	王军	普通物理
物理	王军	微分方程
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	微分方程
数学	张平	高等代数

存在如下多值依赖：C→→T

例: 存储(仓库W，保管员S，货物C)

仓库和保管员　1:n

仓库和货物　　m:n

保管员和货物　m:n

其候选码为(W,S,C)

W→→S，W→→C

W	S	C
w1	s1	c1
w1	s1	c2
w1	s2	c1
w1	s2	c2
w2	s3	c1
w2	s3	c3
w2	s4	c1
w2	s4	c3

多值依赖的性质

① 多值依赖具有**对称性**，即若X→→Y，则X→→Z，其中Z=U-X-Y

② 多值依赖的**传递性**，即若X→→Y，Y→→Z，则X→→Z-Y

③ 函数依赖可以看作是多值依赖的特殊情况，即若X→Y，则X→→Y

④ 若X→→Y，X→→Z，则X→→YZ

⑤ 若X→→Y，X→→Z，则X→→Y∩Z

⑥ 若X→→Y，X→→Z，则X→→Y-Z，X→→Z-Y

2）函数依赖和多值依赖的区别

(1) 多值依赖的有效性与属性集的范围有关：如果X→→Y在W（W⊂U）上成立，在U上不一定成立。

函数依赖X→Y的有效性仅取决于X、Y这两个属性集的值。

2）函数依赖和多值依赖的区别

(2) 对于多值依赖X→→Y，若在R(U)上成立，不能断言对于任何Y的真子集Y’均有X→→Y’也成立。

若函数依赖X→Y在R(U)上成立，则对于任何Y的真子集Y’均有X→Y’也成立；

2）平凡的多值依赖和非平凡的多值依赖

对于属性集U上的一个多值依赖X→→Y（X、Y均为U的子集），如果X包含Y或者 Z=U-X-Y= 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。

第四范式

设R是一个关系模式，D是R上的多值依赖集合。如果D中每个**非平凡多值依赖**X→→Y，X必包含R的码，那么称R是**第四范式的模式**，简记为R∈4NF。

4NF限制属性之间不允许有非平凡且非函数依赖的多值依赖

如果关系模式是**4NF**，则必为**BCNF**

课程C	教师T	参考书B
物理	李勇	普通物理
物理	李勇	微分方程
物理	王军	普通物理
物理	王军	微分方程
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	微分方程
数学	张平	高等代数

候选码是(C,T,B)，存在多值依赖C→→T，C→→B

不是4NF

一个关系模式达到BCNF，但不是4NF，仍然有不好的性质：数据冗余度太大，插入删除复杂。

采用**投影分解**的方法消去非平凡且非函数依赖的多值依赖。

将CTB分解为CT(C, T)，CB(C, B)，这都是平凡的多值依赖，所以CT、CB是4NF。

规范化小结

- 关系数据库的规范化理论是数据库逻辑设计的工具
- 目的：尽量消除插入、删除异常，修改复杂，数据冗余
- 基本思想：逐步消除数据依赖中不合适的部分
- 实质：概念的单一化

关系模式规范化的基本步骤

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

规范化小结(续)

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

例1：设有一个关系模式R(A,B,C,D)，F是R上的函数依赖，F={A→B,A→C}。试写出关系模式R的候选键，分析其满足的范式并将其分解为满足BCNF的关系模式。

解： R的候选码为(A,D)

R为1NF，因为存在着非主属性B、C对码的部分函数依赖（不是2NF）

可分解为: (<u>A,D</u>)和(<u>A</u>,B,C)，均为BCNF

例2：已知关系模式学生(学号,姓名,系名,系主任,课程,成绩),

其上成立的函数依赖有：{学号→姓名，学号→系名, 系名→系主任，(学号,课程)→成绩},

试确定该关系模式的候选键及满足第几范式，如何分解为满足更高级范式的关系模式。

其候选码为 (学号,课程)

该关系模式为1NF，因为存在着非主属性对码的部分函数依赖，所以不是2NF 。

将其分解为2NF

R1(<u>学号</u>,姓名,系名,系主任)

学号→姓名，学号→系名,系名→系主任,

R2(<u>学号,课程</u>,成绩)

(学号,课程)→成绩

将其分解为BCNF

R1(<u>学号</u>,姓名,系名)

学号→姓名，学号→系名

R2(<u>系名</u>,系主任)

系名→系主任

R3(<u>学号,课程</u>,成绩)

(学号,课程)→成绩

第三节　数据依赖的公理系统

1、函数依赖的逻辑蕴涵

定义1：F逻辑蕴含X→Y

对于关系模式R<U，F>，其任何一个具体关系r，若函数依赖X→Y都成立，则称<u>函数依赖集F逻辑蕴含X→Y</u>

定义2：函数依赖集F的闭包

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做<u>F的闭包</u>，记作F^+

例1：R= (A,B,C),F={A→B, B→C}, 求F^+

F^+ =

{A→Φ,AB→Φ,AC→Φ,ABC→Φ,B→Φ,C→Φ

A→A, AB→A, AC→A, ABC→A, B→B, C→C,
A→B, AB→B, AC→B, ABC→B, B→C,
A→C, AB→C, AC→C, ABC→C, B→BC,

A→AB,AB→AB,AC→AB,ABC→AB,BC→Φ,

A→AC,AB→AC,AC→AC,ABC→AC,BC→B,

A→BC,AB→BC,AC→BC,ABC→BC,BC→C,Φ→Φ

A→ABC,AB→ABC,AC→ABC,ABC→ABC,BC→BC }

2、ARMSTRONG公理

定理：若U为关系模式R的属性全集，F为U上的一组函数依赖，设X、Y、Z、W 均为U的子集，对R<U,F>有：

F1(自反律): 若$Y \subseteq X$（表示X包含Y），则$X \rightarrow Y$为F所蕴涵；

F2(增广律): 若$X \rightarrow Y$为F所蕴涵，则$XZ \rightarrow YZ$为F所蕴涵；

F3(传递律): 若$X \rightarrow Y$,$Y \rightarrow Z$为F所蕴涵，则$X \rightarrow Z$为F所蕴涵

定理6.1 ARMSTRONG推理规则是正确的

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6.1 ARMSTRONG推理规则是正确的

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

证：设X→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→Y，于是有t[Y]=s[Y]，所以t[YZ]=s[YZ]，

所以XZ→YZ为F所蕴含，增广律得证。

定理6.1 ARMSTRONG推理规则是正确的

(3) 传递律：若X→Y及Y→Z为F所蕴含，则X→Z为 F所蕴含。

证：设X→Y及Y→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→Y，有 t[Y]=s[Y]；

再由Y→Z，有t[Z]=s[Z]，所以X→Z为F所蕴含，传递律得证。

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→Y，X→Z，有X→YZ。（A2,A3）
- 伪传递规则：由X→Y，WY→Z，有XW→Z。（A2,A3）
- 分解规则：由X→Y及 Z⊆Y，有X→Z。（A1,A3）

2.根据合并规则和分解规则，可得引理6.1：

X→A1 A2…Ak成立的充分必要条件是X→Ai成立（i=l，2，…，*k*）

ARMSTRONG公理系统

- Armstrong公理系统是有效的、完备的
  - 有效性：由F出发根据Armstrong公理推导出来的每一个函数依赖一定在F+中；
  - 完备性：F+中的每一个函数依赖，必定可以由F出发根据Armstrong公理推导出来
- 有效性的证明可通过Armstrong推理规则的正确性得到；
- 要证明完备性，首先要解决如何判定一个函数依赖是否属于由F根据Armstrong公理推导出来的函数依赖的集合

3、属性集闭包

定义：若F为关系模式R(U)的函数依赖集，X是U的子集，则由Armstrong公理推导出的所有$X \rightarrow A_i$所形成的属性集$\{A_i | i=1,2,\ldots\}$称为**X关于F的闭包**，记为X_F^+

例：设R=(A,B,C)，F={A→B, B→C}，当X分别为A、B、C时，求X_F^+。

解：1)当X=A时，$X_F^+ = \{A,B,C\}$

2)当X=B时，$X_F^+ = \{B,C\}$

3)当X=C时，$X_F^+ = \{C\}$

关于闭包的引理

- **引理6.2**

  设F为属性集U上的一组函数依赖，X，Y $\subseteq$ U，X→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→Y是否能由F根据Armstrong公理导出的问题，转化为求出X_F^+、判定Y是否为X_F^+的子集的问题

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^+

（1）令$X^{(0)}$=X，i=0

（2）求B，这里B=

$\{A|(\exists V)(\exists W)(\underline{V \rightarrow W \in F} \wedge \underline{V \subset X^{(i)}} \wedge \underline{A \in W})\}$;

（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

求闭包的算法

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] 已知关系模式R<U,F>，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) $X^{(1)}$= AB∪CD=ABCD

(2) $X^{(0)} \neq X^{(1)}$

$X^{(2)} = X^{(1)}$ ∪BE=ABCDE

(3) $X^{(2)}$ =U，算法终止

$(AB)_F^+$ =ABCDE

ARMSTRONG公理系统的有效性和完备性

定理6.2 Armstrong公理系统是有效的、完备的

证明:

1. 有效性（可由定理6.1得证）

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;

2. 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

只需证明逆否命题: 若函数依赖X→Y不能由F从Armstrong公理导出，那么它必然不为F所蕴含

4、函数依赖的等价与覆盖

定义

设F和G是关系模式R(U)上的两个函数依赖集，如果$F^+ = G^+$，则称**F等价于G**，亦称F覆盖G，或G覆盖F，记为F≡G。

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$F^+ = G^+$

5、最小函数依赖集

设F为函数依赖集，如果F满足：

(1) F中每个函数依赖$X \rightarrow Y$的右边Y均为单个属性；

(2) F中任何一个函数依赖$X \rightarrow A$，其$F-\{X \rightarrow A\}$与F都不等价；

(3) F中任何一个函数依赖$X \rightarrow A$，Z为X的真子集，$(F-\{X \rightarrow A\}) \cup \{Z \rightarrow A\}$与F都不等价；

则称F为极小函数依赖集（最小依赖集或最小覆盖），记作F_{min}。

[例] 关系模式R<U，F>，其中：

U = { Sno，Sdept，Mname，Cno，Grade }，

F = { Sno→Sdept，Sdept→Mname，

(Sno，Cno)→Grade }

F'={Sno→Sdept，Sno→Mname，Sdept→Mname，

(Sno，Cno)→Grade，(Sno，Sdept)→Sdept}

F和F'是否是最小依赖集？

F是最小依赖集，而F'不是。

因为：F' - {Sno→Mname}与F'等价

F' - {(Sno，Sdept)→Sdept}也与F'等价

6、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。

证明: 构造性证明，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 $\{X \rightarrow A_i | i=1，2，\ldots，k\}$ 来取代$X \rightarrow Y$

(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

(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 $(i=1,2,\ldots,m)$，

若$A \in (X-B_i)_F^+$，

则以$X-B_i$ 取代X

[例] $F = \{A\rightarrow B，B\rightarrow A，B\rightarrow C，A\rightarrow C, C\rightarrow A\}$ 求最小依赖集。

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不唯一
- 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

第四节 模式的分解

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$U_i \subseteq U_j$（$i\neq j$），

F_i是F在U_i上的投影，即

$$F_i=\{X\rightarrow Y \mid X\rightarrow Y\in F^+ \wedge XY\subseteq U_i\}$$

一个关系分解为多个关系，原来存储在一张二维表内的数据就要分散存储到多张二维表中，后者不能“丢失”前者的信息

举例

Sno	School	Mname
S1	D1	张五
S2	D1	张五
S3	D2	李四
S4	D3	王一

U={Sno, School, Mname}，F={Sno→School，School→Mname}

一个学生只在一个学院学习，一个学院只有一名院长。

存在传递函数依赖，会发生插入和删除异常。

举例

Sno	School	Mname
S1	D1	张五
S2	D1	张五
S3	D2	李四
S4	D3	王一

现考虑如下分解：

R_1=({Sno,School},{Sno→School})

R_2=({Sno,Mname},{Sno→Mname})

是否还存在异常？存在！

其原因在于School→Mname这个函数依赖被丢掉了

保持函数依赖的分解

ρ={$R_1(U_1,F_1)$, $R_2(U_2,F_2)$, …, $R_k(U_k,F_k)$}是关系模式R(U,F)的一个保持函数依赖的分解，如果

$$F^+ = \left(\bigcup_{i=1}^{k} F_i\right)^+$$

转换为3NF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

② 找出所有不在F中出现的属性，记为U_0，构成关系模式$R_0(U_0,F_0)$，把这些属性从U中去掉

③ 若有$X\rightarrow A\in F$，且$XA=U$，则$\rho=\{R\}$，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形成一个属性集U_i。若$U_i\subseteq U_j$就去掉U_i，此时，$\rho=\{R_1(U_1,F_1), \ldots, R_k(U_k,F_k)\}\cup R_0(U_0,F_0)$构成R(U,F)的一个保持函数依赖的分解，且$R_i(U_i,F_i)$均属于3NF

无损连接的模式分解

- 无损连接(lossless join)，即分解后的关系通过自然连接可以“恢复”原始关系
- 保持函数依赖和无损连接是两个不同的概念
  - 具有无损连接的分解方式不一定能保持函数依赖
  - 保持函数依赖的分解不一定满足无损连接性
- 既无损连接又保持函数依赖的模式分解算法

例1：设有一个记录各个球队球员每场比赛进球的关系模式：

R(球员编号, 比赛场次, 进球数, 球队名, 队长名)

如果规定，每个队员只能属于一个球队，每个球队只有一个队长。

(1)试写出该关系模式的基本函数依赖和候选键;

(2)说明R不是2NF模式的理由，并把R分解为2NF模式集;

(3)进而把R分解为3NF模式集，并说明理由。

R(球员编号,比赛场次,进球数,球队名,队长名),如果规定，每个队员只能属于一个球队，每个球队只有一个队长。

解：其上成立的函数依赖有：

球员编号→球队名，球队名→队长名, (球员编号，比赛场次)→进球数,

其候选码为(球员编号，比赛场次)，该关系模式为1NF,

因为存在着非主属性对码的部分函数依赖，所以不是2NF。

将R(球员编号，比赛场次,进球数,球队名,队长名)分解为2NF

R1(球员编号,球队名,队长名)为2NF

球员编号→球队名，球队名→队长名

R2(球员编号,比赛场次,进球数)为3NF

(球员编号，比赛场次)→进球数，

将R1分解为3NF

R11(球员编号,球队名)为3NF

球员编号→球队名

R12(球队名,队长名)为3NF

球队名→队长名

例2：设有关系模式

R(职工号, 项目名, 工资, 部门名, 部门经理),

如果规定，每个职工可以参加多个项目，各领一份工资，每个项目只属于一个部门管理，每个部门只有一个经理。

(1)试写出该关系模式的基本函数依赖和候选键;

(2)说明R不是2NF模式的理由，并把R分解为2NF模式集;

(3)进而把R分解为3NF模式集，并说明理由。

R(职工号, 项目名, 工资, 部门名, 部门经理),
每个职工可以参加多个项目，各领一份工资，每个项目只属于一个部门管理，每个部门只有一个经理。

解：其上成立的函数依赖有：项目名→部门名，部门名→部门经理，(职工号，项目名)→工资

其候选码为 (职工号，项目名)

该关系模式为1NF

因为存在着非主属性对码的部分函数依赖，所以不是2NF 。

将R(职工号, 项目名, 工资, 部门名, 部门经理)分解为2NF:

R1(项目名，部门名，部门经理)为2NF

项目名→部门名，部门名→部门经理

R2(职工号，项目名，工资)为3NF

(职工号，项目名)→工资

将R1分解为3NF:

R11(项目名，部门名)为3NF

项目名→部门名

R12(部门名，部门经理)为3NF

部门名→部门经理

本章作业

第六版：

P.197

2,3,6,7

第五版：

P.202

2,3,6,7