

Chapter 8: Relational Database Design (2)

Dr. CHEN Jian
Professor
ellachen@scut.edu.cn

SCUT

不良的模式设计

各种异常 现象

不良的数据依赖

部分函数依赖、传递函数依赖

症由

函数依赖的运算

Armstrong公理、属性集的闭包、
函数依赖的推导、候选码的计算
、函数依赖集的等价和最小覆盖

工具

模式分解

模式分解的定义、保持函数依赖
分解和保持无损连接分解的判定
和分解算法

方法

良好的模式设计

1NF、2NF、3NF、BCNF、4NF

规范

关系模式的分解和问题

- SCD(SNO,SN,AGE,DEPT,MN,CNO,SCORE)
 - S(SNO,SN,AGE,DEPT)
 - SC(SNO,CNO,SCORE)
 - D(DEPT,MN)
- 我们可以通过把一个关系模式的分解成多个关系模式，以解决它的插入、删除和更新操作所带来的一些问题。
- 为了在分解要保证原来模式所满足的特性，要求分解处理具有无损连接和保持函数依赖。

模式的分解

定义 关系模式 $R<U,F>$ 的一个分解：

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

$U = U_1 \cup U_2 \cup \dots \cup U_n$, 且不存在 $U_i \subseteq U_j$, F_i

为 F 在 U_i 上的投影

模式的分解

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

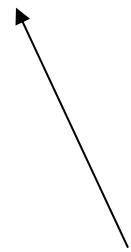
模式的分解

1. SL分解为下面三个关系模式：

SN(Sno)

SD(Sdept)

SO(Sloc)



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

分解后的关系为：

Sno
95001
95002
95003
95004
95005

Sdept
CS
IS
MA
PH

Sloc
A
B
C

分解后的数据库丢失了许多信息。例如无法查询95001学生所在学院或所在宿舍

模式的分解

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：

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

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

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

模式的分解

NL \bowtie 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

NL \bowtie DL 比原来的 SL 关系多了 3 个元组

无法知道 95002、95004、95005 究竟是哪个学院的学生

元组增加了，信息丢失了

模式的分解

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：

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



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



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

信息没有丢失

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

关系模式分解的标准

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

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

若 R 与 R_1 、 R_2 、 \dots 、 R_n 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

模式分解中存在的问题

无损分解

$R(A, B, C)$

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

$R(A, B, C)$

$\prod_{AB}(R)$

A	B	C
1	1	1
2	1	2

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

定理

- 如果 R 的分解为 $\rho = \{R_1, R_2\}$, F 为 R 上的函数依赖集合, 分解 ρ 具有无损连接性的充分必要条件为:

$$R_1 \cap R_2 \rightarrow (R_1 - R_2)$$

或 $R_1 \cap R_2 \rightarrow (R_2 - R_1)$

两个模式的公共属性可以函数确定其中一个模式

举例：

设 $R = \langle U, F \rangle$, $U = \{ABC\}$, $F = \{A \rightarrow B\}$, 证明 $\rho_1 = \{R_1(AB), R_2(AC)\}$ 是无损连接, $\rho_2 = \{R_1(AB), R_3(BC)\}$ 不是无损连接。

解:

$$\rho_1 = \{R_1(AB), R_2(AC)\}$$

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

由 $A \rightarrow B$, 得到 ρ_1 是无损连接分解

$$\rho_2 = \{R_1(AB), R_3(BC)\}$$

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

$B \rightarrow A, B \rightarrow C$ 均不成立, 所以 ρ_1 不是无损连接分解

保持函数依赖的分解

- 保持关系模式分解等价的另一个重要条件是关系模式的函数依赖集在分解后仍在数据库模式中保持不变。
- 即关系模式 R 到 $\rho = \{R_1, R_2, \dots, R_k\}$ 的分解，应使函数依赖集 F ，被 F 在这些 R_i 上的投影蕴涵

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

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

(其中 $U=U_1 \cup U_2 \cup \dots \cup U_n$, 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)。

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

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

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

保持函数依赖的分解

■ 如何判断分解保持函数依赖？

$$F^+ = (\bigcup_{i=1}^n F_i)^+ \Leftrightarrow F^+ \subseteq (\bigcup_{i=1}^n F_i)^+, (\bigcup_{i=1}^n F_i)^+ \subseteq F^+$$

如对于{ABC, {A→B, B→C}}的分解

{<AB, {A→B}>, <AC, {A→C}>}

思考: B→C ∈ {A→B, A→C}+ ?

检验: C ∈ B+_{A→B, A→C} ?

模式的分解

- 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。

保持函数依赖的分解

关系模式 $R < U, F >$

$U = \{CITY, ST, ZIP\}$,

$F = \{(CITY, ST) \rightarrow ZIP, ZIP \rightarrow CITY\}$

$\rho = \{R_1 = \{ST, ZIP\}, R_2 = \{CITY, ZIP\}\}$

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

$\therefore R_1 \cap R_2 \rightarrow R_2 - R_1 \quad \therefore$ 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 $(CITY, ST) \rightarrow ZIP$

练习

分析下列分解是否具有无损连接和保持FD的特点。

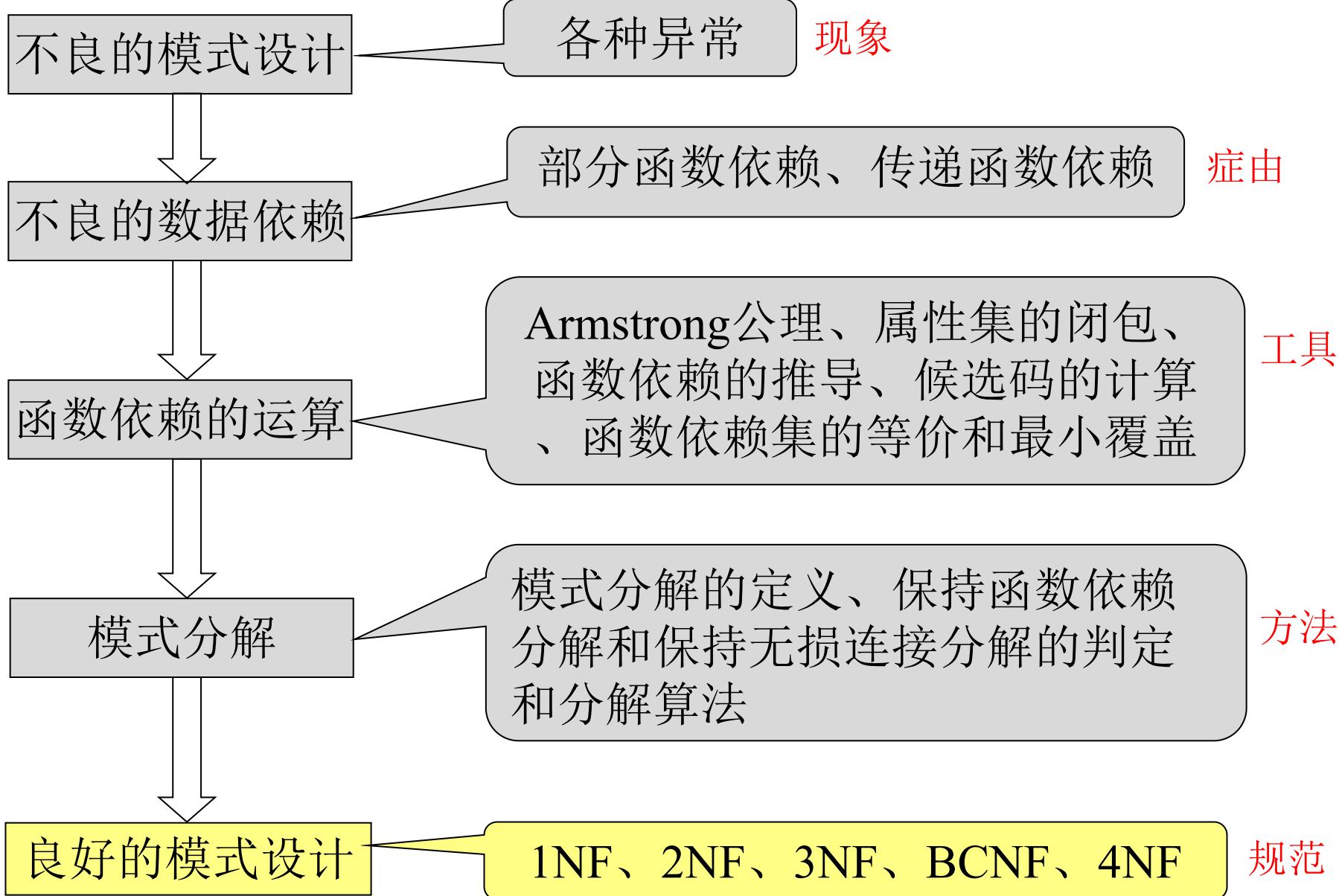
- 1、 $R(ABC), F=\{C \rightarrow A\}, \rho=\{AC, BC\}$
- 2、 $R(ABC), F=\{C \rightarrow B, A \rightarrow B\}, \rho=\{AC, BC\}$
- 3、 $R(ABC), F=\{C \rightarrow A\}, \rho=\{AB, AC\}$
- 4、 $R(ABC), F=\{C \rightarrow A, B \rightarrow C\}, \rho=\{AB, AC\}$

- 1、 无损， 保持FD
- 2、 无损， 不保持FD,丢失 $A \rightarrow B$
- 3、 有损分解， 保持FD
- 4、 有损， 不保持FD(丢失 $B \rightarrow C$)

题目：设关系模式 $R(A,B,C,D)$
 F 是 R 上成立的
FD集， $F=\{A\rightarrow B, B\rightarrow C, A\rightarrow D, D\rightarrow C\}$ ，
 $\rho=\{AB, AC, BD, CD\}$ 是 R 的一个分解。

- 1、试求 F 在 ρ 的每个模式上的投影？
- 2、 ρ 保持函数依赖吗？为什么？

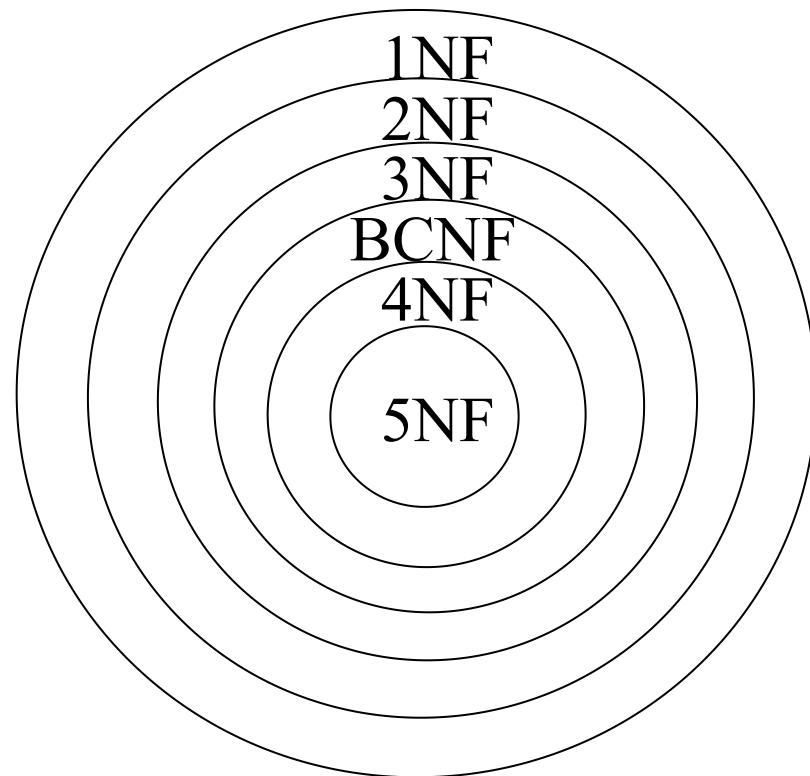
- F 在 AB, AC, BD, CD 上的投影分别为
 $\{A\rightarrow B\}, \{A\rightarrow C\}, \emptyset, \{D\rightarrow C\}$ 丢失了 $B\rightarrow C, A\rightarrow D$ ，
- 所以此分解不保持函数依赖



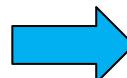
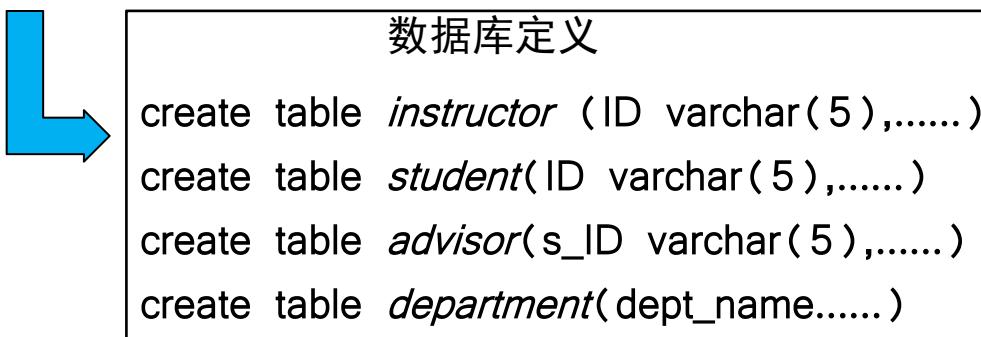
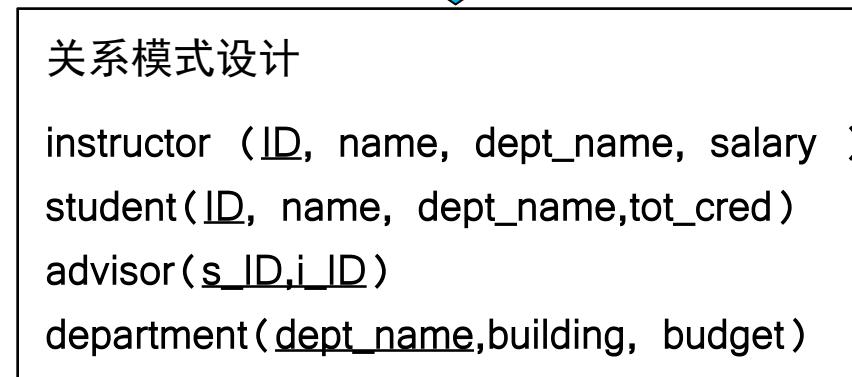
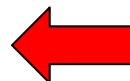
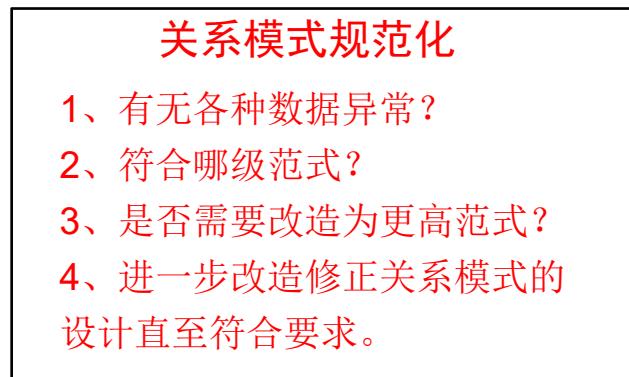
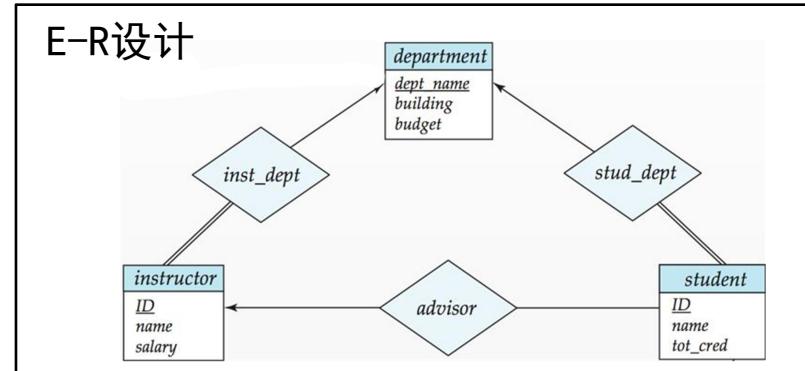
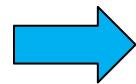
范式

■ 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）



范式的作用



主属性与非主属性

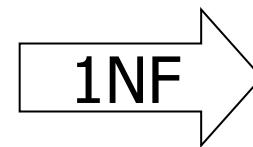
- 主属性：候选码中的属性
- 非主属性：不包含在任何一个候选码中的属性称为非主属性
- 例子：
 - 学生（学号、姓名、性别、年龄）
 - 选修（学号、课程号、成绩）
 - 使用（职工编号、项目编号、机器编号）
 - 设关系R (A,B,C,D),同时满足 $AB \rightarrow C$, $C \rightarrow D$, $D \rightarrow A$,请问其主属性和非主属性分别是什么?
 - ▶ 由于其候选码是AB,BC和BD，因此A,B,C,D均为主属性。

1NF

- 定义：一个关系模式 R 满足第一范式，当且仅当 R 中每个属性的域都是原子的（不可再分的），即所有属性的值都是标量。
- 核心要求：

1. 表中无重复组。
2. 每一列都是单一数据类型的最小数据单元。
3. 通常需要有一个主码来确保行的唯一性。

SNO	CNO
S1	{C1, C2, C3}



SNO	CNO
S1	C1
S1	C2
S1	C3

对象—关系数据库
嵌套关系

传统数据库
平面关系

1NF

适合于查询每个学生的选修课程

适合于查询每门课程的选修学生

SNO	CNO
S1	{C1, C2, C3}

SNO	CNO
{S1, S2, S3}	C1

- 两个都保存，数据冗余，更新困难
- 只保存一个，某些查询困难

■ 原子粒度的选择

- 分量是否需要再分，与具体应用有关。如果用到值的一部分，则需要进一步分割
- 否则需要应用编码解析

只是查询
出生日期

姓名	生日
王军	68.7.10
张立	69.7.10
李明	80.3.28

比较两人生
肖是否相同

姓名	年	月日
王军	68	7.10
张立	69	7.10
李明	80	3.28

比较两人生
日是否相同

姓名	年	月	日
王军	68	7	10
张立	69	7	10
李明	80	3	28

- 较细的原子粒度有助于标准化，施加约束，避免输入错误，从而提高数据质量

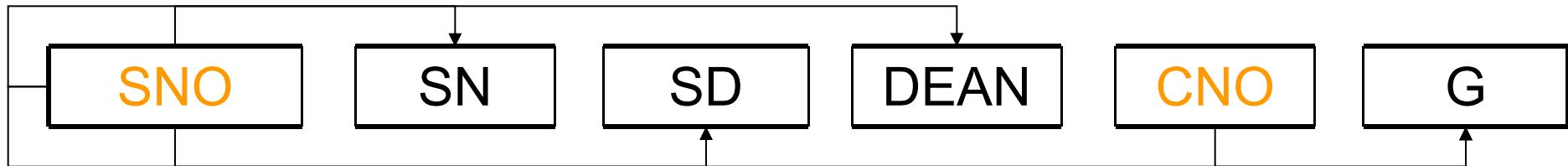
北京大学，北京，中国，100871，11/25/2006

中国，北京，北京大学，100871，25/11/2006

国家	城市	单位	邮编	年	月	日
中国	北京	北京大学	100871	2006	11	25

2NF

SNO	SN	SD	DEAN	CNO	G
S01	杨明	D01	思齐	C01	90
S02	李婉	D01	思齐	C01	87
S01	杨明	D01	思齐	C02	92
S03	刘海	D02	述圣	C01	95
S04	安然	D02	述圣	C02	78
S05	乐天	D03	省身	C01	82



S(SNO, SN, SD, DEAN, CNO, G)

■ 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
- 更新异常：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

2NF

■ 定义：一个关系模式 R 满足第二范式，当且仅当：

1. R 满足第一范式。
2. 每一个非主属性都完全函数依赖于 R 的每一个候选码。

■ 核心要求：

1. 消除非主属性对任何候选码的部分函数依赖。
2. 此范式主要针对含有复合候选码的关系。

上述 $S \notin 2NF$, 因为

$$(SNO, CNO) \xrightarrow{p} SN$$

$$(SNO, CNO) \xrightarrow{p} SD$$

■ 改造

非主属性有两种，一种完全依赖于候选码，一种部分依赖于候选码

将S分解为：

$SC(\underline{SNO}, \underline{CNO}, G)$

$S_SD(\underline{SNO}, SN, SD, DEAN)$

■ 快速热身

关系模式R (A, B, C, D)，唯一候选码为AB，给出它的一个函数依赖集，使得R属于1NF而不属于2NF

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

或 $F = \{ \dots, A \rightarrow D, \dots \}$ 或 $F = \{ \dots, B \rightarrow D, \dots \}$

分解说明

- 一个1NF，但非2NF的关系总是可以被分解成为一组2NF的关系
- 规范化过程中通过一组投影运算消除部分依赖，建议作如下分解(**第一步分解**)
 - 已知关系R(A,B,C,D), (A,B)为唯一候选码，即 $(A,B) \rightarrow C, (A,B) \rightarrow D$, 且 $A \rightarrow D$, 则将R分解成为两个投影:
 $R1(\underline{A},D)$, A为唯一候选码
 $R2(\underline{A},B,C)$, (A,B)为唯一候选码, A为外码
 - 这样，R可以通过R1和R2的自然连接运算得以恢复

习题

设有一个记录各个球队队员每场比赛进球的关系模式如下所示：

$R(\text{队员ID}, \text{比赛场次}, \text{进球数}, \text{球队名}, \text{队长ID})$

若规定每个队员只能属于一个球队，每个球队只有一个队长，要求：

- 1) 写出关系模式R的基本函数依赖和候选码。
- 2) 说明R不是2NF模式的理由，并把R分解成2NF模式集。

习题

- 每个队员只能属于一个球队
 - 队员ID→球队名
- 每个球队只有一名队长
 - 球队名→队长ID
- 每个队员每场比赛只有一个进球数
 - (队员ID, 比赛场次) →进球数
- 因此, 唯一候选码为 (队员ID, 比赛场次)

习题

■ 因此，R中F的正则覆盖为：

- 队员ID → (球队名, 队长ID)
- (队员ID, 比赛场次) → 进球数

■ 第一个FD是部分函数依赖，因此不是2NF

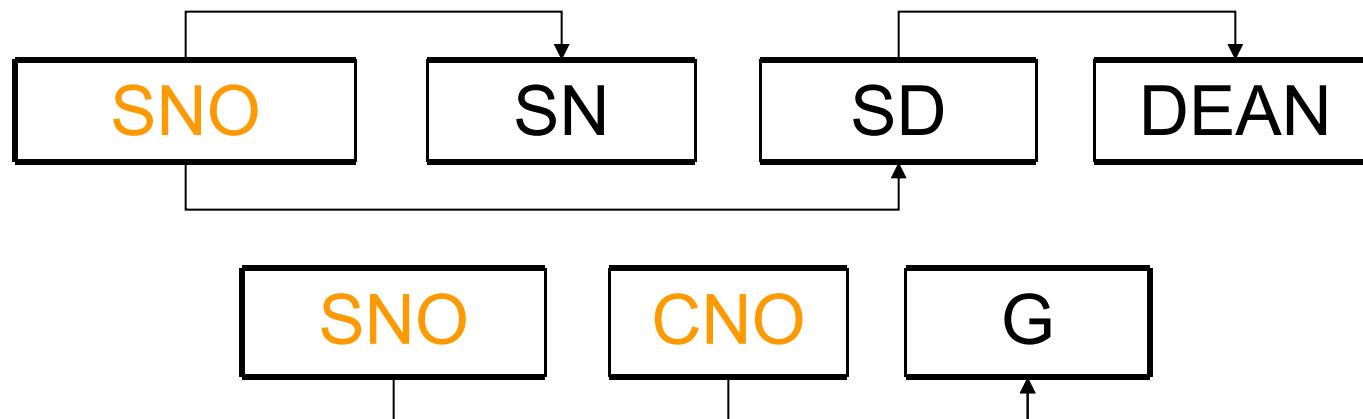
■ 分解为：

- R1 (队员ID, 球队名, 队长ID)
- R2 (队员ID, 比赛场次, 进球数)

3NF

SNO	SN	SD	DEAN
S01	杨明	D01	思齐
S02	李婉	D01	思齐
S03	刘海	D02	述圣
S04	安然	D02	述圣
S05	乐天	D03	省身

SNO	CNO	G
S01	C01	90
S02	C01	87
S01	C02	92
S03	C01	95
S04	C02	78
S05	C01	82



S_SD(SNO , SN , SD , DEAN)

■ 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
- 更新异常：如果学生转系，不但要修改SD，还要修改DEAN，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：每个学生都存储所在系的系主任的信息

■ 定义：一个关系模式 R 满足第三范式，当且仅当：

1. R 满足第二范式。
2. 每一个非主属性都不传递函数依赖于 R 的每一个候选码。

（即，不存在非主属性 A ，使得候选码 $\rightarrow B \rightarrow A$ 成立，且 $B \not\rightarrow$ 候选码）

■ 核心要求：

1. 消除非主属性对任何候选码的传递函数依赖。

如 $S_SD \notin 3NF$ ，因为有 $SNO \rightarrow SD$, $SD \rightarrow DEAN$

■ 改造

将S分解为

STUDENT(SNO , SN , SD)
DEPT(SD , DEAN)

■ 快速热身

关系模式R (A, B, C, D) , 唯一候选码为AB
 , 给出它的一个函数依赖集, 使得R属于2NF而不
 属于3NF $F = \{ \dots, C \rightarrow D, \dots \}$ 或 $F = \{ \dots, D \rightarrow C, \dots \}$

SNO	SN	SD
S01	杨明	D01
S02	李婉	D01
S03	刘海	D02
S04	安然	D02
S05	乐天	D03

SD	DEAN
D01	思齐
D02	述圣
D03	省身

分解说明

- 一个2NF，但非3NF的关系总是可以被分解成为一组3NF的关系
- 规范化过程中通过一组投影运算消除传递依赖，建议作如下分解(**第二步分解**)
 - 已知关系R(A,B,C), A为唯一候选码($A \rightarrow B, A \rightarrow C$), 且 $B \rightarrow C$,
则将R分解成为两个投影:
 $R1(\underline{B},C)$, B为唯一候选码
 $R2(\underline{A},B)$, A为唯一候选码, B为外码
 - 这样, R可以通过R1和R2的自然连接运算得以恢复, 分解满足

习题

[例]设有如下所示的关系R

职工号	职工名	年龄	性别	部门号	部门名
E1	万千里	20	女	D3	开发部
E2	于得水	25	男	D1	财务部
E3	余乐	38	男	D3	开发部
E4	高明亮	25	女	D3	开发部

- (1) 它为第几范式？为什么？
- (2) 是否存在删除异常？若存在，请说明在什么情况下发生？
- (3) 将它分解为高一级范式。
- (4) 分解后的关系如何解决分解前可能存在 的删除操作异常问题？

习题

(1) 关系R是2NF

(2) 存在插入，删除异常：

当插入一个新部门，而该部门还没有员工时，
不能加入，从而发生插入异常；

当某部门只有一名员工，删除该员工时会把
该部门的信息也删除，从而发生删除异常。

原因在于非主属性“部门名称”对码“职工
号”存在传递函数依赖。

(3) 分解为两个关系，使之达到3NF；

职工（职工号，职工名，年龄，性别，部门号）

部门（部门号，部门名称）

(4) 分解后的关系可以避免上述操作异常问题。

当增加一个新部门，而该部门还没有员工时，仍然
可以向部门关系中加入该部门信息，从而解决了
插入异常问题。

当某部门只有一名员工，删除该员工时，直接删除
职工关系中的该行信息即可。不再会把该部门的
信息也删除，从而解决了删除异常的问题。

习题

假设某商业集团数据库中有一关系模式R如下：

R (商店编号, 商品编号, 数量, 部门编号, 部门负责人)

如果规定：

- (1) 每个商店的每种商品只在一个部门销售；
- (2) 每个商店的每个部门只有一个部门负责人；
- (3) 每个商店的每种商品只有一个库存数量。

试回答下列问题：

- (1) 根据上述规定，写出关系模式R的基本函数依赖；
- (2) 找出关系模式R的候选码；
- (3) 试问关系模式R最高已经达到第几范式？为什么？
- (4) 如果R不属于3NF，请将R分解成3NF模式集。

习题

(1) 有三个函数依赖:

(商店编号, 商品编号) → 部门编号

(商店编号, 商品编号) → (商店编号, 部门编号) → 部门负责人

(商店编号, 商品编号) → 数量

(2) R的候选码是 (商店编号, 商品编号)

(3) 因为R中存在着非主属性"部门负责人"对候选码 (商店编号、商品编号) 的传递函数依赖, 所以R属于2NF, R不属于3NF。

(4) 将R分解成:

R1 (商店编号, 商品编号, 数量, 部门编号)

R2 (商店编号, 部门编号, 部门负责人)

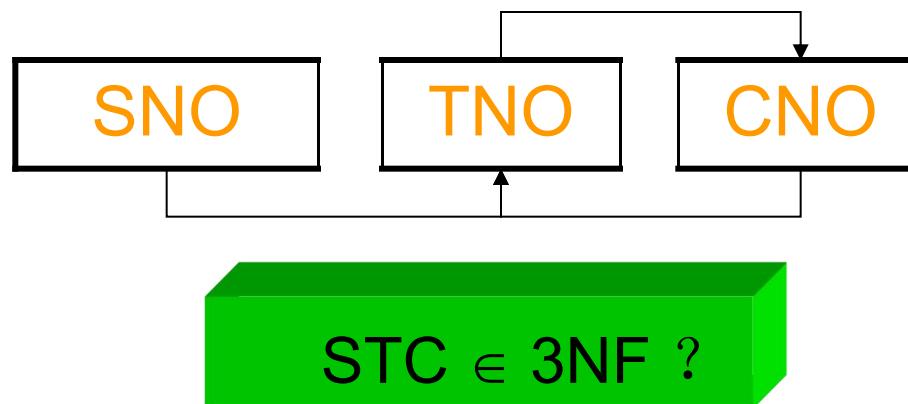
BCNF

STC(SNO , TNO , CNO)

每位老师只教授一门课 \rightarrow $TNO \rightarrow CNO$

某学生选定一门课，就对应一位老师 \rightarrow $(SNO, CNO) \rightarrow TNO$

候选码 \rightarrow (SNO, TNO) and (SNO, CNO)



BCNF

SNO	TNO	CNO
s1	t1	c1
s2	t2	c2
s3	t3	c2
s3	t1	c1

■ 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
- 更新异常：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： $TNO \rightarrow CNO$, 属性对码的不良依赖

BCNF

■ 定义：一个关系模式 R 满足第 BC 范式，当且仅当：

1. R 满足第一范式。
2. 对于 R 中的每一个非平凡的函数依赖 $X \rightarrow Y$ ，其决定因素 X 必须是一个超码（即 X 包含 R 的某个候选码）。

■ 核心要求：

1. 比 3NF 更严格。在所有情况下， \rightarrow 的左侧都必须是超码。
◦

如 $STC \notin BCNF$ ，因为 $TNO \rightarrow CNO$ ，而 TNO 不含有任何完整候选码，如下改造：

SNO	TNO
s1	t1
s2	t2
s3	t3
s3	t1

TNO	CNO
t1	c1
t2	c2
t3	c2

习题

■ 学生选课(学号, 课程号, 身份证号, 成绩) 问题: 满足2NF? 3NF? BCNF?

1. 列出所有的函数依赖关系

(学号, 课程号) \rightarrow 成绩

(学号, 课程号) \rightarrow 身份证号

(身份证号, 课程号) \rightarrow 成绩

(身份证号, 课程号) \rightarrow 学号

学号 \rightarrow 身份证号

身份证号 \rightarrow 学号

2. 确定候选码 (Candidate Keys)

(学号, 课程号) 可以唯一确定一条记录 (哪个学生选了哪门课, 以及他的成绩和身份证号); (身份证号, 课程号) 同样可以唯一确定一条记录 (哪个身份证的人选了哪门课)。没有更小的属性集能唯一标识元组。

因此, 候选码是: CK1 = (学号, 课程号), CK2 = (身份证号, 课程号)。。

3. 确定主属性 (Prime Attributes) 与非主属性 (Non-Prime Attributes)

主属性：出现在至少一个候选码中的属性。

学号 (在CK1中), 课程号 (在CK1和CK2中), 身份证号 (在CK2中)

所以，主属性有三个：学号，课程号，身份证号。

非主属性：不属于任何候选码的属性。

在这个表中，只有 成绩 一个属性不属于任何候选码。所以，唯一的非主属性是 成绩

4. 判断第二范式

2NF要求：每一个非主属性必须完全函数依赖于每一个候选码。

检查非主属性 成绩：

成绩 是否完全依赖于 CK1 (学号, 课程号)? 是的。

成绩 是否完全依赖于 CK2 (身份证号, 课程号)? 是的。

结论：该关系模式满足第二范式。

5. 判断第三范式

3NF要求：满足2NF，且所有非主属性既不部分依赖于候选码，也不传递依赖于候选码。

非主属性 成绩 直接、完全地依赖于两个候选码，不存在传递依赖。

结论：该关系模式也满足第三范式。

6. 判断BCNF

BCNF要求：对于每一个非平凡的函数依赖 $X \rightarrow Y$ ， X 必须是一个超码。

我们来列出所有重要的函数依赖：

学号 \rightarrow 身份证号 (问题所在！ 学号 本身不是超码)

身份证号 \rightarrow 学号 (问题所在！ 身份证号 本身不是超码)

结论：该关系模式不满足BCNF。

习题

■ 关系模式 WPE (Wno, Pno, Eno, Quantity) 记录仓库的配件管理情况，其中 Wno 表示仓库号，Pno 表示配件号，Eno 表示职工号，Quantity 表示数量。

- 一个仓库有多个职工
- 一个职工仅在一个仓库工作
- 每个仓库里一种型号的配件由专人负责
- 一个人可以管理几种配件
- 同一种型号的配件可以分放在几个仓库中

■ 问题：符合2NF?3NF?BCNF?

候选码：

(Wno, Pno) 和
(Pno, Eno)

由于 $Eno \rightarrow Wno$ ，
所以不满足 BCNF
的要求

关系模式的分解算法

■ 算法：（达到BCNF无损连接分解算法）

```
result := {R};  
done := false;  
compute  $F^+$ ;  
while (not done) do  
  if (there is a schema  $R_i$  in result that is not in BCNF)  
    then begin  
      let  $\alpha \rightarrow \beta$  be a nontrivial functional dependency that holds  
      on  $R_i$  such that  $\alpha \rightarrow R_i$  is not in  $F^+$ , and  $\alpha \cap \beta = \emptyset$ ;  
      result := (result -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ );  
    end  
  else done := true;
```

Figure 8.11 BCNF decomposition algorithm.

关系模式的分解算法

■ 示例： $R = \langle U, F \rangle$, $U = \{A, B, C, D, E\}$

$F = \{A \rightarrow B, B \rightarrow C, (A,D) \rightarrow E\}$,求 R 的一个BCNF 分解

解： 1. 求候选码

码是 $\{AD\}$

2. 检查 $A \rightarrow B$, 由于 A 不是候选码，因此

$U_1 = \{A, B\}$, $F_1 = \{A \rightarrow B\}$

$U_2 = \{A, C, D, E\}$, $F_2 = \{A \rightarrow C, (A,D) \rightarrow E\}$

3. 检查 $A \rightarrow C$, 由于 A 不是候选码，因此

$U_1 = \{A, B\}$, $F_1 = \{A \rightarrow B\}$

$U_2 = \{A, C\}$, $F_2 = \{A \rightarrow C\}$

$U_3 = \{A, D, E\}$, $F_3 = \{(A,D) \rightarrow E\}$

关系模式的分解算法

■ 示例: $U=(SNO, TNO, CNO)$

$F=\{(SNO, CNO)\rightarrow TNO, TNO\rightarrow CNO\}$

解: 不属于BCNF, 分解为

$U_1=(SNO, TNO),$

$U_2=(TNO, CNO), F_2=\{TNO\rightarrow CNO\}$

丢失了函数依赖 $(SNO, CNO)\rightarrow TNO$, 原来一个学生选修一门课程时, 只能对应一个老师; 在新的关系模式下现在一个学生选修一门课程时, 可能会对应多个老师。

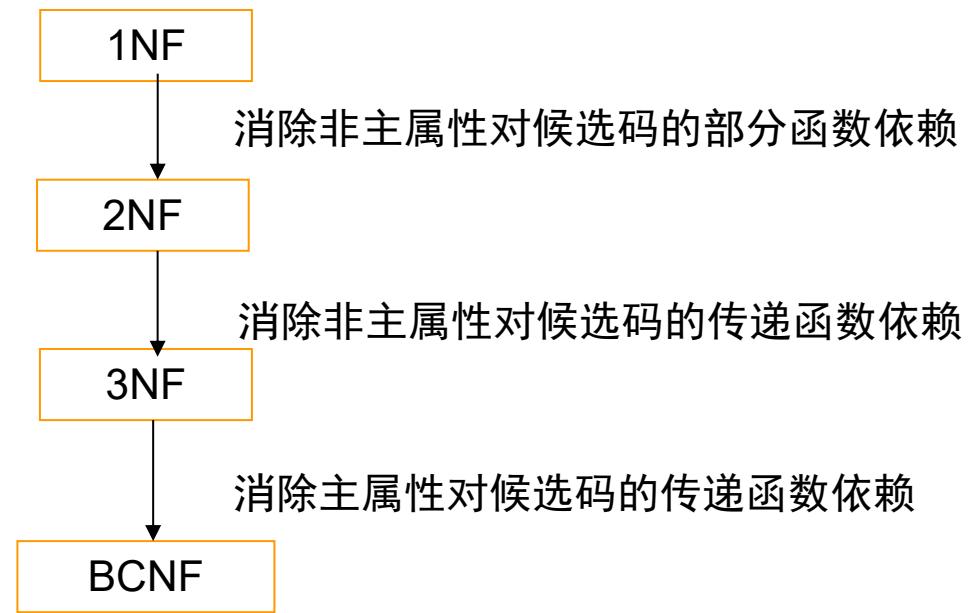
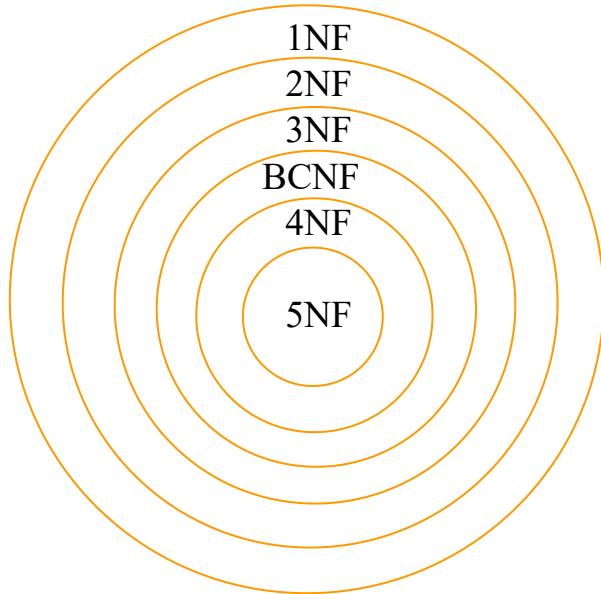
关系分解为BCNF, 不一定能保持函数依赖

关系模式的分解算法

- 结论：若要求分解保持函数依赖，那么分解后的模式总可以达到3NF，但不一定能达到BCNF
- 达到3NF且保持函数依赖的无损分解算法
 - ▶ 原书图8-12

- P-PROVINCE,C-CITY,S-STREET,Z-ZIP
- 设有关系模式R(SNO, SN, P, C, S, Z)
 $F=\{SNO \rightarrow SN, SNO \rightarrow P, SNO \rightarrow C, SNO \rightarrow S, SNO \rightarrow Z, \{P, C, S\} \rightarrow Z, Z \rightarrow P, Z \rightarrow C\}$, 试分解R为3NF。
- 解:
 - 求F的最小覆盖 $F_C = \{SNO \rightarrow \{SN, P, C, S\}, \{P, C, S\} \rightarrow Z, Z \rightarrow \{P, C\}\}$
 - 根据上述算法, 则分解为 $R1 = \{SNO, SN, P, C, S\}$, $R2 = \{P, C, S, Z\}$,
 $R3 = \{Z, P, C\}$, $\because R3 \subseteq R2$, 去掉R3
 - $R1 = \{SNO, SN, P, C, S\}$, $F1 = \{SNO \rightarrow \{SN, P, C, S\}\}$
 - $R2 = \{P, C, S, Z\}$, $F2 = \{\{P, C, S\} \rightarrow Z, Z \rightarrow \{P, C\}\}$
 - 注意 $R1, R2$ 均属于3NF, 但 $R1$ 属于BCNF, $R2$ 不属于BCNF,
若进一步将 $R2$ 分解为 $\{P, C, Z\}$ 和 $\{S, Z\}$, 则均为BCNF, 但丢失了
函数依赖 $Z \rightarrow \{P, C\}$.

范式之间的关系



一般看法

- 分解通常使得对复杂查询的回答的效率更差，因为在查询求值期间必须执行额外的连接。
- 分解使得对简单查询的回答更有效，因为这种查询通常涉及相同关系的一小部分属性
- 分解通常使得简单的更新事务更有效
- 分解能降低存储空间的要求，因为它一般能消除冗余数据
- 如果冗余级别低，则分解会增加存储的需求

反范式/规范化设计

- 通过规范化可以分解得到一些结构简单的表，但是在查询时可能需要连接多个表，这造成了大量的数据库逻辑运算，从而降低了查询效率
- 在具体环境下，为了提高查询效率，可以在特定的表中引入一些额外的属性
- 数据库设计运行存在合理冗余
- 规范化和反规范化是相互矛盾的，在具体应用环境中，需要数据库设计者仔细分析和平衡

反范式/规范化设计

◆ 实例分析

提出问题

计算stuNo='S001'学生的已获学分，需要对三个表进行连接

stuNo	stuName	stuSex	stuAge
S001	白玉芬	女	19
S002	唐嘉芝	女	18

classNo	classCourseName	classStartTime	classCredit
C001	微观经济学	2008-3-1	2
C002	大学英语	2008-3-1	4
C003	建筑艺术	2008-3-16	2

stuNo	classNo	score
S001	C001	78
S001	C002	67
S001	C003	56
S002	C001	92

反范式/规范化设计

◆ 实例分析

☛ 分析问题

查询和计算过程在经过数据库编译后，将会变成一个比较繁琐的程序在后台运行，尤其在数据库记录非常多、用户访问量很大的情况下将会造成用户难以忍受的延迟。

在表中适当增加一定冗余，可以减少数据库服务器的负荷，提高查询速度，如果需要经常查询和统计学生已获得学分，可以在原来的学生表中增加一个新的字段stuTotalCredi来表示已取得的总学分，虽然这样设计在一定程度上会导致数据的冗余，但是可以极大提高查询速度。

反范式/规范化设计

◆ 实例分析

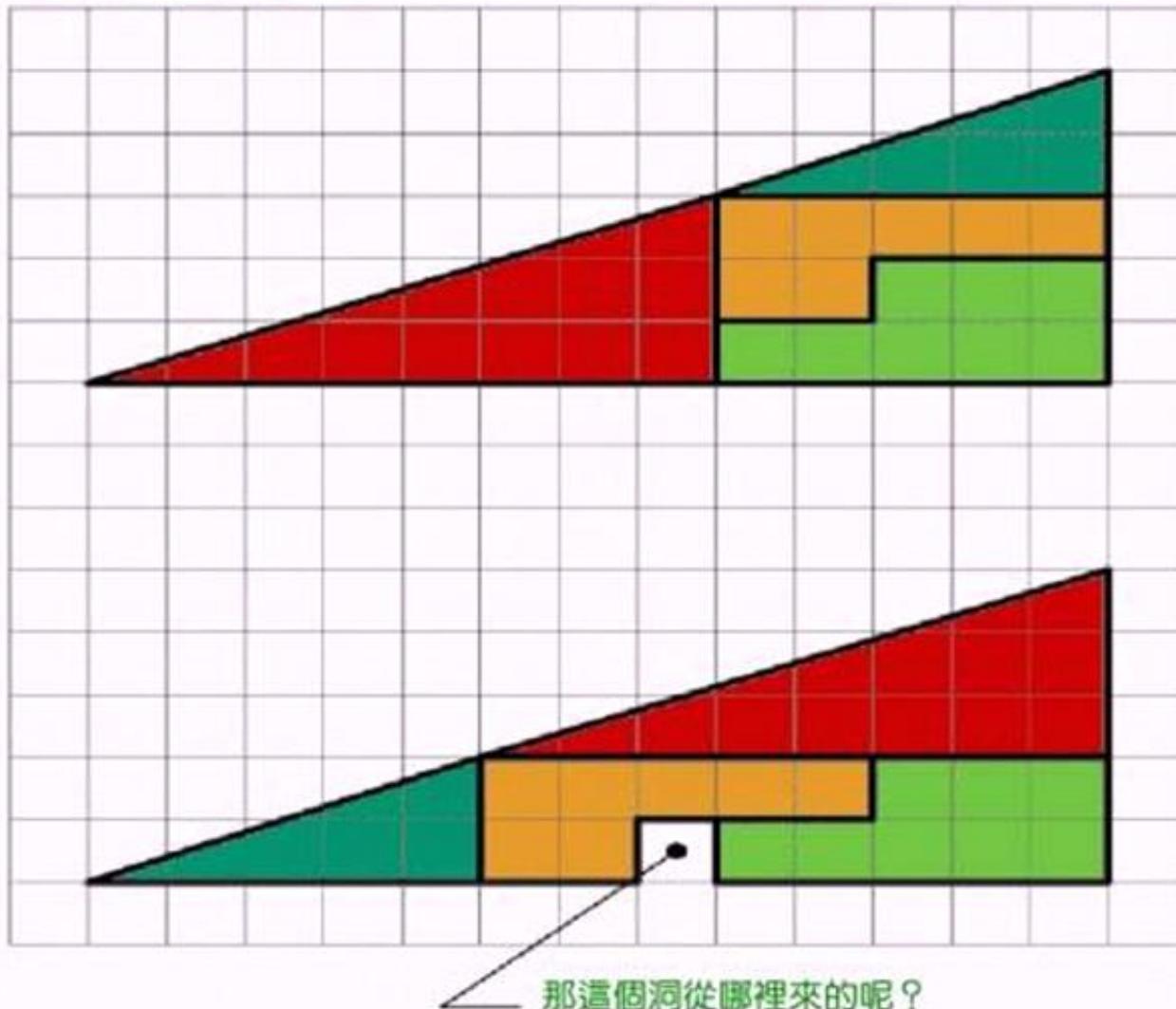
☞ 问题求解

增加一个新列来存储学生已获总学分



stuNo	stuName	stuSex	stuAge	stuTotalCredit
S001	白玉芬	女	19	6
S002	唐嘉芝	女	18	2

這怎麼可能呢？



將圖上的四區
拆開來移到下
圖各位置。

每一部份與上
圖完全相同。

Assignment

- 1、 8.29 d
- 2、 Consider a relation schema $R(A,B,C,D,E)$ and its functional dependencies, $F=\{ A \rightarrow C, C \rightarrow A, B \rightarrow AC, D \rightarrow AC \}$, complete the following questions:
 - a) Compute $(AD)^+$
 - b) Compute the candidate keys for R .
 - c) Compute the canonical cover F_c .
 - d) Is R in 3NF? If it is, justify your answer. If not, produce a decomposition of R into 3NF.(Ref. Figure 8.12)

- 3. Suppose that we have a schema $R(A, B, C, D, E)$. You are given the following dependencies:

$$A \rightarrow B$$

$$BC \rightarrow E$$

$$ED \rightarrow A$$

- a) List all candidate keys for R .
 - b) Is R in 3NF? If it is, justify your answer. If not, produce a decomposition of R into 3NF. (Ref. Figure 8.12)
4. Which normal form do the schemas you obtained in Chapter 7 Exercise belong to? Is it possible to transform them into 3NF? And how?

End of Chapter 8(2)