数据库系统原理与设计

（第 3 版）

志存高远，

脚踏实地，

勇于探索！

数据库系统原理与设计

（第 3 版）

第5章　关系数据理论及求精

目 录

本章要解决的两个问题

- 如何判断一个数据库模式是“好”的模式
- 如何设计出一个“好”模式?

数据冗余导致的问题

- **数据冗余**是指同一信息在数据库中存储了多个副本。它可能引起下列问题：
  - **冗余存储**：信息被重复存储，导致浪费大量存储空间。
  - **更新异常**：当重复信息的一个副本被修改，所有副本都必须进行同样的修改。因此当更新数据时，系统要付出很大的代价来维护数据库的完整性，否则会面临数据不一致的危险。
  - **插入异常**：只有当一些信息事先已经存放在数据库中时，另外一些信息才能存入数据库中。
  - **删除异常**：删除某些信息时可能丢失其它信息。

数据冗余关系举例

- [例5.1] 考虑学生选课关系模式：

  SCE(studentNo, studentName, courseNo, courseName, score),

  属性集{studentNo, courseNo}是唯一**候选码**，也是**主码**。

- 如果允许一个学生选修多门课程，且一门课程可被多个学生选修，则该关系实例可能出现：

studentNo	StudentName	courseNo	courseName	score
S0700001	李小勇	C001	高等数学	98
S0700001	李小勇	C002	离散数学	82
S0700001	李小勇	C006	数据库系统原理	56
S0700002	刘方晨	C003	计算机原理	69
S0700002	刘方晨	C004	C语言程序设计	87
S0700002	刘方晨	C005	数据结构	77
S0700002	刘方晨	C007	操作系统	90
S0700003	王红敏	C001	高等数学	46
S0700003	王红敏	C002	离散数学	38
S0700003	王红敏	C007	操作系统	50

冗余问题？

- **冗余存储：** 学生姓名和课程名被重复存储多次
- **更新异常：** 当修改某学生的姓名或某课程的
  部分副本的信息，而其他副本未被修改到；
- **插入异常：** 如果某学生没有选修课程，或某课程未被任何学生选修时，则该学生或该课程信息**不能存入数据库**，因为主码值不能为空；
- **删除异常：** 当某学生的所有选修课程信息都被删除时，则该学生的信息**将被丢失**。对课程也是如此。

[例5.1] 考虑学生选课关系模式：
SCE(studentNo, studentName, courseNo, courseName, score),
属性集{studentNo, courseNo}是唯一候选码，也是主码。

数据冗余产生原因及解决方法

- SCE产生问题的原因：部分函数依赖
  - 该模式中某些属性之间存在如下函数依赖关系
    - studentNo→studentName（即学号决定姓名）
    - courseNo→courseName（即课程编号决定课程名称）
    - {studentNo, courseNo}→score
  - 即studentName、courseName部分依赖于关系的主码。
- 解决方法：关系模式分解
  - 分解为三个关系模式
    - S(studentNo, studentName)
    - C(courseNo, courseName)
    - E (studentNo, courseNo, score)

模式分解导致的问题

- 将一个关系模式分解为较小关系模式集可解决冗余问题。但由此可能产生两个新的问题：
  - 什么样的关系模式需要进一步分解为较小的关系模式集？
    - 根据范式要求决定（后面讨论）
  - 是否所有的模式分解都是有益的？
    - 实例

模式分解问题举例

- [例5.2] 设一关系模式

  STU (<u>studentNo</u>, studentName, sex, birthday, native, classNo),

  其中，**studentNo**为**主码**。

- 假设将STU分解为以下两个子模式：
  - **STU1 (<u>studentNo</u>, studentName)**
  - **STU2 (studentName, sex, birthday, native, classNo)**

分解后会发生什么问题?

模式分解问题举例

studentNo	studnetName	sex	birthday	native	classNo
S0700005	王红	男	1992-4-26	江西省南昌市	CS0702
S0800005	王红	女	1995-8-10	湖北省武汉市	CP0802

studentNo	studnetName
S0700005	王红
S0800005	王红

studnetName	sex	birthday	native	classNo
王红	男	1992-4-26	江西省南昌市	CS0702
王红	女	1995-8-10	湖北省武汉市	CP0802

$\bowtie$

studentNo	studnetName	Sex	birthday	native	classNo
S0700005	王红	男	1992-4-26	江西省南昌市	CS0702
S0700005	王红	女	1995-8-10	湖北省武汉市	CP0802
S0800005	王红	男	1992-4-26	江西省南昌市	CS0702
S0800005	王红	女	1995-8-10	湖北省武汉市	CP0802

图5-2　有损分解举例

模式分解问题举例

假设将STU分解为以下两个子模式：

- STU1 (studentNo, studentName)
- STU2 (studentName, sex, birthday, native, classNo)

■ 模式分解存在的问题

- **有损分解**：两个分解后的关系通过连接运算还原得到的信息与原来关系的信息不一致。
  - 如果能够通过连接分解后所得到的较小关系完全还原被分解关系的所有实例，称之为**无损分解**(lossless decomposition)，也称该分解具有无损连接特性。
- **丢失依赖关系**
  - sex、birthday、age、native、classNo等与属性studentNo依赖关系也就不再存在。
  - 如果被分解关系模式上的所有依赖关系都在分解得到的关系模式上保留，称该分解为**依赖保持** (dependency preserving)分解。

小　结

■一个“好”的关系模式应该是:

- 数据冗余尽可能少(即数据共享尽可能高)
- 不发生插入异常、删除异常、更新异常等问题。
- 模式分解时，分解后的模式应具有无损连接、保持依赖等特性。

目 录

函数依赖定义

- **函数依赖**(functional dependency, 简称FD)是**一种完整性约束**，是现实世界事物属性之间的一种制约关系，它广泛地存在于现实世界之中。

- 定义5.1 设$r(R)$为关系模式，$\alpha \subseteq R$，$\beta \subseteq R$。对任意合法关系r及其中任两个元组t_i和t_j，$i \neq j$，若$t_i[\alpha]=t_j[\alpha]$，则$t_i[\beta]=t_j[\beta]$，则称α**函数确定**β，或 β**函数依赖于**α，记作$\alpha \to \beta$。

	学号	姓名
t1	11	张三
t2	22	李四
t3	11	王五
t4	66	张三

图5-3　$\alpha \to \beta$函数依赖图

	学号	姓名
t1	11	张三
t2	22	李四
t3	11	张三
t4	66	张三

函数依赖举例

- [例5.3] 关系模式*r*(*A*, *B*, *C*, *D*)的一个关系实例如图5-4所示.
  - 对于任意两个在属性集{*A*, *B*}上取值相同的元组，它们在属性*C*上的取值也相同。
  - 因此，满足函数依赖$AB \rightarrow C$。

A	*B*	*C*	*D*
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c1	d1
a2	b1	c3	d1

图5-4　满足函数依赖$AB \rightarrow C$的一个关系实例

如果在图中再增加一个元组(a1, b1, c2, d1)，那么函数依赖$AB \rightarrow C$ 还成立吗？

函数依赖

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

- **检验：A→C？ C→A？ AB→D？**

函数依赖说明

- 对于函数依赖，需做如下说明：
  - 函数依赖不是指关系模式$r(R)$的某个或某些关系实例满足的约束条件，而是指关系模式$r(R)$的所有关系实例均要满足的约束条件。
  - 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，是不能够被证明的。
  - 数据库设计者可以对现实世界作强制的规定。
  - 码约束是函数依赖的一个特例。码属性(集)相当于定义5.1中的α，关系中的所有属性相当于定义5.1中的β。

平凡与非平凡函数依赖

- **定义5.2** 在关系模式$r(R)$中，$\alpha \subseteq R$，$\beta \subseteq R$。若$\alpha \rightarrow \beta$，但$\beta \not\subseteq \alpha$，则称$\alpha \rightarrow \beta$是**非平凡函数依赖**。否则，若$\beta \subseteq \alpha$，则称$\alpha \rightarrow \beta$是**平凡函数依赖**。
- 对于任一关系模式，**平凡函数依赖**都是必然成立的，它不反映新的语义。

(a) 非平凡函数依赖　　　　(b) 平凡函数依赖

图5-5　非平凡及平凡函数依赖图

完全函数依赖和部分函数依赖

- 定义5.3 在关系模式$r(R)$中，$\alpha\subseteq R$，$\beta\subseteq R$，且$\alpha\rightarrow\beta$是非平凡函数依赖。若对任意的$\gamma\subset\alpha$，$\gamma\rightarrow\beta$都不成立，则称$\alpha\rightarrow\beta$是完全函数依赖，简称完全依赖。否则，若存在非空的$\gamma\subset\alpha$，使$\gamma\rightarrow\beta$成立，则称$\alpha\rightarrow\beta$是部分函数依赖，简称部分依赖。
  - 如果$\alpha\rightarrow\beta$是完全函数依赖，且$\alpha\neq\beta$，则$\alpha\rightarrow\beta$一定是非平凡函数依赖。

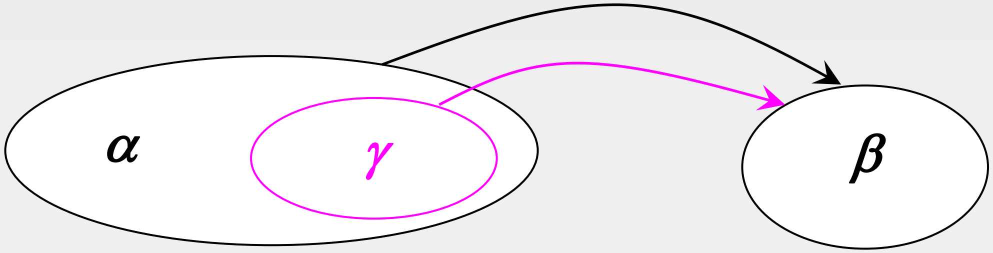


图5-6 部分依赖$\alpha\rightarrow\beta$的依赖图

studentNo	StudentName	courseNo	courseName	score
S0700001	李小勇	C001	高等数学	98
S0700001	李小勇	C002	离散数学	82
S0700001	李小勇	C006	数据库系统原理	56
S0700002	刘方晨	C003	计算机原理	69
S0700002	刘方晨	C004	C语言程序设计	87
S0700002	刘方晨	C005	数据结构	77
S0700002	刘方晨	C007	操作系统	90
S0700003	王红敏	C001	高等数学	46
S0700003	王红敏	C002	离散数学	38
S0700003	王红敏	C007	操作系统	50

完全函数依赖和

- 当α是单属性时，则$\alpha \rightarrow \beta$完
- 例如，在关系SCE中
  - 完全依赖：
    - studentNo → studentName
    - courseNo → courseName
    - {studentNo, courseNo} → score
  - 部分依赖：
    - {studentNo, courseNo} → studentName
    - {studentNo, courseNo} → courseName
- 对候选码的部分函数依赖会导致数据冗余和插入、删除、更新异常！

传递函数依赖

- **定义5.4** 在关系模式$r(R)$中，$\alpha \subseteq R$，$\beta \subseteq R$，$\gamma \subseteq R$，且$\beta \not\subseteq \alpha$，$\beta \not\to \alpha$。若$\alpha \to \beta$，$\beta \to \gamma$，则必存在函数依赖$\alpha \to \gamma$，并称$\alpha \to \gamma$是**传递函数依赖**，简称**传递依赖**。
  - 注意条件：$\beta \not\subseteq \alpha$ 和 $\beta \not\to \alpha$。

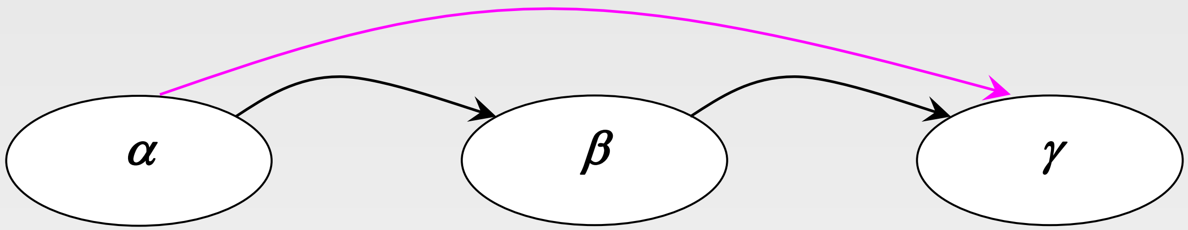


图5-7　传递依赖$\alpha \to \gamma$的依赖图

- 与部分依赖一样，传递依赖也可能会导致数据冗余及产生各种异常。

传递函数依赖举例

- [例5.4] 在关系模式SCI(studentNo, classNo, className, institute)中，存在下列函数依赖：
  - studentNo → classNo
  - classNo → className
  - classNo → institute
- 因此，关系模式SCI中存在下列传递函数依赖：
  - studentNo → className
  - studentNo → institute
- SCI中存在传递依赖，因此可能导致数据冗余、更新异常、插入异常及删除异常。
  - 请自己分析SCI中存在的各种异常问题。

函数依赖小结

- 函数依赖是指关系模式中属性之间存在的一种约束关系。这种约束关系既可以是现实世界事物或联系的属性之间客观存在的约束，也可以是数据库设计者根据应用需求或设计需要强加给数据的一种约束。
- 但不论是那种约束，一旦确定，进入数据库中的所有数据都必须严格遵守。
- 正确了解数据的意义及确定属性之间的函数依赖关系，对设计一个好的关系模式是十分重要的。
- 平凡/不平凡，完全/非完全，传递函数 依赖

目 录

范式概述

- 基于函数依赖理论，关系模式可分成
  - 第一范式(1NF)
  - 第二范式(2NF)
  - 第三范式(3NF)
  - Boyce-Codd范式(BCNF)
- 这几种范式的要求一个比一个严格，它们之间的联系为：

$$\text{BCNF} \subset \text{3NF} \subset \text{2NF} \subset \text{1NF}$$

- 满足BCNF范式的关系一定满足3NF范式，满足3NF范式的关系一定满足2NF范式，满足2NF范式的关系一定满足1NF范式。

第一范式(1NF)——码

- 定义5.5 如果一关系模式*r*(*R*)的每个属性对应的域值都是不可分的(即原子的)，则称*r*(*R*)属于**第一范式**，记为*r*(*R*)∈1NF.
- **第一范式的目标是：将基本数据划分成称为实体集或表的逻辑单元，当设计好每个实体后，需要为其指定主码。**

studentNo	studentName	sex	birthday	age	address			classNo
					province	city	street	

图5-8 非规范化的关系模式

studentNo	studentName	sex	birthday	age	province	city	street	classNo

图5-9 1NF规范化后的关系模式

数据冗余关系举例

studentNo	StudentName	courseNo	courseName	score
S0700001	李小勇	C001	高等数学	98
S0700001	李小勇	C002	离散数学	82
S0700001	李小勇	C006	数据库系统原理	56
S0700002	刘方晨	C003	计算机原理	69
S0700002	刘方晨	C004	C语言程序设计	87
S0700002	刘方晨	C005	数据结构	77
S0700002	刘方晨	C007	操作系统	90
S0700003	王红敏	C001	高等数学	46
S0700003	王红敏	C002	离散数学	38
S0700003	王红敏	C007	操作系统	50

图5-1　学生选课关系SCE实例

第二范式(2NF)——全

- 定义5.6 设有一关系模式$r(R)$，$\alpha \subseteq R$。
  个候选码中，则称α为主属性，否则α
  - 在SCE关系中，属性集{studentNo, courseNo}是SCE的唯一候选码。因此，属性studentNo和courseNo为主属性，其余属性为非主属性。
- 定义5.7 如果一个关系模式$r(R) \in$1NF，且所有非主属性都完全函数依赖于$r(R)$的候选码，则称$r(R)$属于第二范式，记为$r(R) \in$2NF。
  - SCE中存在依赖关系studentNo→studentName和courseNo→courseName，即非主属性studentName和courseName部分依赖于SCE的候选码{studentNo, courseNo}，故SCE$\notin$2NF。

第二范式(2NF)——全部是码

- 第二范式的目标：将只部分依赖于候选码（即依赖于候选码的部分属性）的非主属性移到其他表中。
- 也就是说，在满足第一范式的实体中，如果有复合候选码(即多个属性共同构成的候选码)，那么所有非主属性必须依赖于全部的候选码，不允许依赖于部分的候选码属性。
  - 即不允许候选码的一部分对非主属性起决定作用：全部是码
- 违背2NF的模式，即存在非主属性对候选码的部分依赖，则可能导致例5.1所述的数据冗余及异常问题。

■ 解决方法：关系模式分解

● 分解为三个关系模式

➢S(studentNo, studentName)

➢C(courseNo, courseName)

➢E (studentNo, courseNo, score)

NF) ——

系模式，可

化，以消除部分依赖。

数据冗余关系举例

studentNo	StudentName	courseNo	courseName	score
S0700001	李小勇	C001	高等数学	98
S0700001	李小勇	C002	离散数学	82
S0700001	李小勇	C006	数据库系统原理	56
S0700002	刘方晨	C003	计算机原理	69
S0700002	刘方晨	C004	C语言程序设计	87
S0700002	刘方晨	C005	数据结构	77
S0700002	刘方晨	C007	操作系统	90
S0700003	王红敏	C001	高等数学	46
S0700003	王红敏	C002	离散数学	38
S0700003	王红敏	C007	操作系统	50

图5-1 学生选课关系SCE实例

● 如将关系模式SCE分解为关系模式S、C和E。这样在每个关系模式中，所有非主属性对候选码都是完全函数依赖，因此都属于2NF范式。

■ 2NF范式虽然消除了由于非主属性对候选码的部分依赖所引起的冗余及各种异常，但并没有排除传递依赖。因此，还需要对其进一步规范化。

第三范式(3NF)——仅仅是码

- 第三范式的目标：去掉表中不直接依赖于候选码的非主属性.
- 定义5.8 如果一个关系模式$r(R)\in$2NF，且所有非主属性都直接函数依赖于$r(R)$的候选码（即不存在非主属性传递依赖于候选码），则称$r(R)$属于第三范式，记为$r(R)\in$3NF.
- 也就是说，在满足2NF的实体中，非主属性不能依赖于另一个非主属性（即非主属性只能直接依赖于候选码）
- 总之，所有的非主属性应该直接依赖于(即不能存在传递依赖，这是3NF的要求)全部的候选码(即必须完全依赖, 不能存在部分依赖，这是2NF的要求)。

范式举例

- **[例5.5]** $r(R)=r(A, B, C, D)$，函数依赖集 $F=\{AB\to C, B\to D\}$。$r(R)$的候选码为AB
  - 因为函数依赖$B\to D$中的决定属性B只是候选码的一部分，即D部分依赖于候选码AB。
  - 可将$r(R)$分解为$r_1(R_1)=r_1(A, B, C)$、$r_2(R_2)=r_2(B, D)$。$r_1(R_1)$的候选码为AB，$r_2(R_2)$的候选码为B。
  - 分解得到的$r_1(R_1)$和$r_2(R_2)$都属于3NF范式。

范式举例

- [例5.6] $r(R)=r(A, B, C)$，函数依赖集$F=\{A\rightarrow B, B\rightarrow C\}$。$r(R)$的候选码为$A$，$r(R)\in$2NF?
  - 因为函数依赖$B\rightarrow C$中的决定属性B不是候选码，即C传递依赖于候选码A。
  - 可将$r(R)$分解为$r_1(R_1)=r_1(A, B)$、$r_2(R_2)=r_2(B, C)$。$r_1(R_1)$的候选码为A，$r_2(R_2)$的候选码为B。
  - 则分解得到的$r_1(R_1)$和$r_2(R_2)$都属于3NF范式。

范式举例

- [例5.7] $r(R)=r(A, B, C, D, E)$，函数依赖集$F=\{AB\rightarrow C, B\rightarrow D, C\rightarrow E\}$。$r(R)$的候选码为$AB$
  - 因为函数依赖$B\rightarrow D$中的决定属性B只是候选码的一部分，即D部分依赖于候选码AB；另外E传递依赖于候选码AB。
  - $r(R)$分解为$r_1(R_1)=r_1(A, B, C)$、$r_2(R_2)=r_2(B, D)$、$r3(R3)=r3(C, E)$
  - $r_1(R_1)$的候选码为AB，$r_2(R_2)$的候选码为B，$r_3(R_3)$的候选码为C。 它们都属于3NF范式。

范式举例

- [例5.8] $r(R)=r(A, B, C)$，函数依赖集$F=\{AB\to C, C\to A\}$. $r(R)$的候选码为AB或BC，属于第几范式？
- $r(R)\in$3NF。
  - 因为关系模式$r(R)$没有非主属性，也就不可能有非主属性对候选码的部分依赖和传递依赖。

总结

- **数据冗余：**冗余存储、更新异常、插入异常、删除异常。
- “好”的关系模式应该是：
  - 数据冗余应尽可能少；
  - 不发生插入异常、删除异常、更新异常等问题；
  - 模式分解时，需要满足无损连接、保持依赖等特性。
- 函数依赖、完全/部分函数依赖、传递/直接函数依赖。
- **范式：** $3NF \subset 2NF \subset 1NF$。
- **2NF：**所有非主属性都完全函数依赖于$r(R)$的候选码。
- **3NF：**所有非主属性都直接函数依赖于$r(R)$的候选码。

Boyce-Codd范式(BCNF)

- 基于函数依赖理论，关系模式可分成：
  - 第一范式(1NF)：所有属性都是原子的；
  - 第二范式(2NF)：不存在非主属性对候选码的部分依赖；
  - 第三范式(3NF)：不存在非主属性对候选码的传递依赖。
  - Boyce-Codd范式(BCNF)：？

函数依赖集闭包

函数依赖集闭包

■对于给定关系模式$r(R, D, DOM, F)$（简记为 $r(R)$）及其函数依赖集F，有时只考虑给定的函数依赖集是不够的，而需要考虑在$r(R)$上总是成立的所有函数依赖。

■[例5.13] 给定关系模式$r(R)=r(A, B, C)$及函数依赖集 $F=\{A\rightarrow B, B\rightarrow C\}$，证明$A\rightarrow C$成立。

证明：假设对于关系实例r中的任意两个元组$t_i, t_j, i\neq j$，满足$t_i[A]=t_j[A]$。由于存在$A\rightarrow B$，则可推出$t_i[B]=t_j[B]$。

又由于$B\rightarrow C$，则又可推出$t_i[C]=t_j[C]$。

因此，$t_i[A]=t_j[A] \Rightarrow t_i[C]=t_j[C]$。按定义5.1有$A\rightarrow C$。证毕。

函数依赖集闭包

F={A->B,B->C}, A->C

- 定义5.11 若给定函数依赖集*F*，可以证明其他函数依赖也成立，则称这些函数依赖被*F*逻辑蕴涵。
- 定义5.12 令*F*为一函数依赖集，*F*逻辑蕴涵的所有函数依赖组成的集合称为*F*的闭包，记为F^+。
- 函数依赖集*F*的闭包计算方法
  - Armstrong公理的推理规则

Armstrong公理及推论

Armstrong公理

- 自反律(reflexivity rule)：若存在$\beta \subseteq \alpha$，则有$\alpha \rightarrow \beta$
- 增补律(augmentation rule)：若存在$\alpha \rightarrow \beta$，则有$\gamma\alpha \rightarrow \gamma\beta$
- 传递律(transitivity rule)：若存在$\alpha \rightarrow \beta$且$\beta \rightarrow \gamma$，则有$\alpha \rightarrow \gamma$

Armstrong公理三个推论

- 合并律(union rule)：若有$\alpha \rightarrow \beta$且$\alpha \rightarrow \gamma$，则有$\alpha \rightarrow \beta\gamma$
- 分解律(decomposition rule)：若有$\alpha \rightarrow \beta\gamma$，则有$\alpha \rightarrow \beta$和$\alpha \rightarrow \gamma$
- 伪传递律(pseudotransitivity rule)：若有$\alpha \rightarrow \beta$且$\beta\gamma \rightarrow \delta$，则有$\alpha\gamma \rightarrow \delta$

请自己证明三个推论。

函数依赖集闭包计算举例

- [例5.14] 令$r(R)=r(A, B, C, G, H, I)$，函数依赖集$F=\{A\rightarrow B, A\rightarrow C, CG\rightarrow H, CG\rightarrow I, B\rightarrow H\}$。我们可列出$F^+$中的几个依赖：
  - 由传递律可得$A\rightarrow H$，因为$A\rightarrow B$且$B\rightarrow H$；
  - 由合并律可得$CG\rightarrow HI$，因为$CG\rightarrow H$，$CG\rightarrow I$；
  - 由伪传递律可得$AG\rightarrow I$，因为$A\rightarrow C$且$CG\rightarrow I$。
- 还可以使用上述规则推导出更多的函数依赖，…

如果想要判断一个给定的函数依赖$\alpha\rightarrow\beta$是否在函数依赖集F的闭包中？

属性集闭包

- 如果想要判断一个给定的函数依赖$\alpha \rightarrow \beta$是否在函数依赖集F的闭包中，不用计算F^+就可以判断出来。
- 定义5.13 令$r(R)$为关系模式，F为函数依赖集，$A \subseteq R$，则称在函数依赖集F下由A函数确定的所有属性的集合为**函数依赖集F下属性集A的闭包**，记为A^+。
- 属性集闭包的计算算法：

```
closure :=A;
repeat     /* 外循环 */
   temp := closure;
   for each α→β ∈F do     /* 内循环 */
      if α⊆closure
         closure := closure ∪ β ;
         if closure=R
            break;
until (closure=temp or closure=R);
```

图5-12 计算F下A^+算法

属性集闭包计算举例

- [例5.15] $r(R)=r(A, B, C, G, H, I)$，$F=\{A\to B, A\to C, CG\to H, CG\to I, B\to H\}$，计算$(AG)^+$。
  - 算法的执行步骤如下：

步骤	FD	closure
1.	赋初值	AG
2.	$A\to B$	ABG
3.	$A\to C$	$ABCG$
4.	$CG\to H$	$ABCGH$
5.	$CG\to I$	$ABCGHI$

```
closure :=A;
repeat     /* 外循环 */
  temp := closure;
  for each α→β∈F do     /* 内循环 */
    if α⊆closure
      closure := closure ∪ β;
      if closure=R
        break;
until (closure=temp or closure=R);
```

图5-12 计算F下A^+算法

结果为：$closure=ABCGHI$。

  - 算法在外循环的第一次执行过程中，当内循环循环执行4次(即遍历到F中的函数依赖$CG\to I$)后，$closure$就已经为$ABCGHI$(即R)，算法终止。最后结果为：$(AG)^+=ABCGHI$。

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**

BCNF

S#	T#	C#
s1	t1	c1
s2	t2	c2
s3	t3	c2
s3	t1	c1

- 示例

STC(S# , T# , C#),

语义：每位老师只教授一门课,每门课有若干教师,某一学生选定某门课,就对应一个固定的教师.

T# → **C#**，每位老师只教授一门课

（**S#**，**T#**）→ **C#**

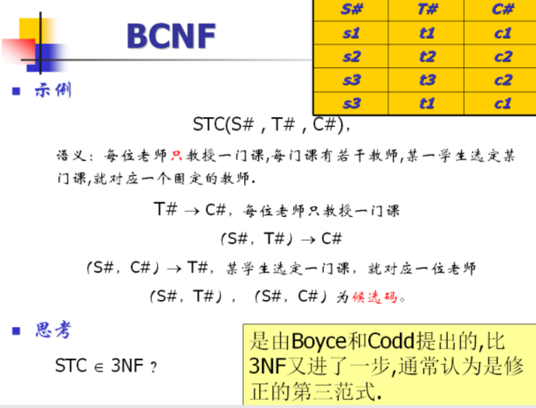
（**S#**，**C#**）→ **T#**，某学生选定一门课，就对应一位老师

（**S#**，**T#**），（**S#**，**C#**）为候选码。

- 思考

STC ∈ **3NF** ？

是由Boyce和Codd提出的,比3NF又进了一步,通常认为是修正的第三范式.



BCNF

S#	T#	C#
s1	t1	c1
s2	t2	c2
s3	t3	c2
s3	t1	c1

- 示例

STC(S# , T# , C#),

语义：每位老师只教授一门课,每门课有若干教师,某一学生选定某门课,就对应一个固定的教师.

T# → C#，每位老师只教授一门课

（S#，T#）→ C#

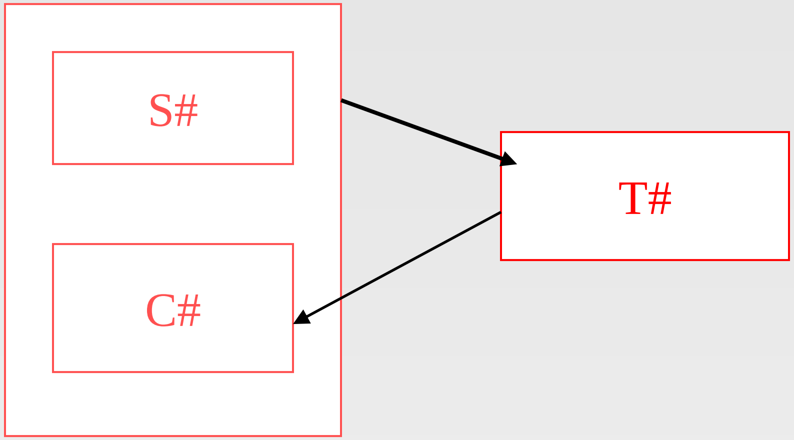
（S#，C#）→ T#，某学生选定一门课，就对应一位老师

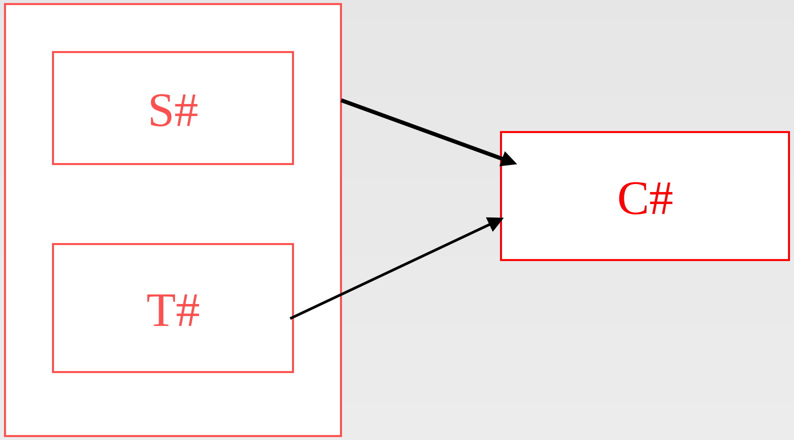
（S#，T#），（S#，C#）为候选码。

- 思考

STC ∈ 3NF ?

是由Boyce和Codd提出的,比3NF又进了一步,通常认为是修正的第三范式.





C#对码的传递依赖

C#对码的部分依赖

BCNF

S#	T#	C#
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：

主属性对码的不良依赖

BCNF

S#	T#
s1	t1
s2	t2
s3	t3
s3	t1

T#	C#
t1	c1
t2	c2
t3	c2

对于属

U , F >

→ C#，而T#不含有码

改造前：STC(S# , T# , C#)，

■ 改造后

将S分解为（S#，T#），（T#，C#）

T# → C#，每位老师只教授一门课

（S#，T#）→ C#

（S#，C#）→ T#，某学生选定一门课，就对应一位老师

（S#，T#），（S#，C#）为候选码。

小结—范式之间的关系

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

↓ 消除主属性对码的传递函数依赖

BCNF

↓ 消除非平凡的多值依赖

4NF

61

目 录

计算属性集闭包的作用

■ 计算属性集闭包的作用可归纳如下：

- 验证$\alpha \to \beta$是否在F^+中：看是否有$\beta \subseteq \alpha^+$。
- 判断α是否为$r(R)$的超码：计算α^+，看其是否包含R的所有属性。如$(AG)^+ = ABCGHI$，则AG为$r(R)$的超码。
- 判断α是否为$r(R)$的候选码：若α是超码，可检验α包含的所有子集的闭包是否包含R的所有属性。若不存在任何这样的属性子集，则α是$r(R)$的候选码。

判断属性集是否为候选码举例

- [例5.16] $r(R)$和F见例5.15，判断AG是否为$r(R)$的候选码.
  - 例5.15已计算出$(AG)^+$＝$ABCGHI$，则还要进一步分别计算A^+和G^+。
  - 经计算得，$A^+=ABCH$、$G^+=G$，它们都不包含R的所有属性。因此，AG为$r(R)$的候选码。
- 对于一个给定的关系模式$r(R)$及函数依赖集F，如何找出它的所有候选码?
  - 这是基于函数依赖理论和范式概念判断该关系模式是否是“好”模式的基础;
  - 也是对一个“不好”的关系模式进行分解的基础。

判断属性集是否为候选码

- 给定关系模式$r(R)$及函数依赖集F，找出它的所有**候选码**的一般步骤如下：
  - 找出函数依赖集F中在所有函数依赖右方都没有出现的属性集X，属性集X中的属性都一定是**候选码**中的属性。
  - 找出F中在所有函数依赖右方出现但左方没有出现的属性集Y，属性集Y中的属性都不可能是**候选码**中的属性。
  - 如果X非空，则基于F计算X^+，并开始发现所有**候选码**：
    - 如果$X^+=R$，则X是关系$r(R)$的唯一**候选码**；
    - 如果$X^+\neq R$，则

- ✓ 首先，试着发现是否能够通过增加1个属性与X联合起来构成候选码，例如，若存在$\alpha \in R-X-Y$，使$(X\cup\{\alpha\})^+=R$，则$(X\cup\{\alpha\})$是关系$r(R)$的一个候选码；继续试着增加另一个属性，若存在$\beta \in R-X-Y-\{\alpha\}$，使$(X\cup\{\beta\})^+=R$，则$(X\cup\{\beta\})$是关系$r(R)$的另一个候选码；……。记找到的所有属性的集合为$Z$，即$\forall \alpha \in Z$，使$(X\cup\{\alpha\})^+=R$。
- ✓ 接下来，还可以试着发现是否能够通过增加2个或多个属性与X联合起来构成候选码，例如，若存在$\{\alpha, \beta\} \subseteq R-X-Y-Z$，使$(X\cup\{\alpha, \beta\})^+=R$；则$(X\cup\{\alpha, \beta\})$也是关系$r(R)$的一个候选码；……。

- 如果X非空，则基于F计算X^+，并开始发现所有候选码：
  - ➢如果$X^+=R$，则X是关系$r(R)$的唯一候选码；
  - ➢如果$X^+\neq R$，则

- 如果X为空，则从F中的每一个函数依赖$\alpha \rightarrow u$开始(先从左边属性较少的函数依赖开始)：

- ✓ 首先，试着发现是否能够通过**增加1个属性**与X联合起来构成**候选码**，例如，若存在$\alpha \in R-X-Y$，使$(X\cup\{\alpha\})^+=R$，则$(X\cup\{\alpha\})$是关系$r(R)$的一个候选码；继续试着增加另一个属性，若存在$\beta \in R-X-Y-\{\alpha\}$，使$(X\cup\{\beta\})^+=R$，则$(X\cup\{\beta\})$是关系$r(R)$的另一个候选码；……。记找到的所有属性的集合为$Z$，即$\forall \alpha \in Z$，使$(X\cup\{\alpha\})^+=R$。
- ✓ 接下来，还可以试着发现是否能够通过**增加2个或多个属性**与X联合起来构成**候选码**，例如，若存在$\{\alpha, \beta\} \subseteq R-X-Y-Z$，使$(X\cup\{\alpha, \beta\})^+=R$；
  - ➢ 如果$\alpha^+=R$，则α是关系$r(R)$的一个候选码；
  - ➢ 如果$\alpha^+\neq R$，类似地，试着发现是否能够通过**增加1个属性**与α联合起来构成**候选码**；再试着发现是否能够通过**增加2个或多个属性**与α联合起来构成**候选码**。

- 如果X为空，则从F中的每一个函数依赖$\alpha \rightarrow u$开始(先从左边属性较少的函数依赖开始)：

判断属性集是否为候选码举例

- **[例5.17]** 给定关系模式$r(R)=r(A, B, C, D)$，函数依赖集$F=\{B\rightarrow C, D\rightarrow A\}$，找出$r(R)$的所有**候选码**。
  - 属性集BD没有在函数依赖的右部出现，故BD为候选码的一部分；
  - 由于$(BD)^+=BDCA=R$，所以BD为关系模式$r(R)$的唯一**候选码**。

判断属性集是否为候选码举例

- [例5.18] 给定关系模式$r(R)=r(A, B, C, D, E)$，函数依赖集$F=\{A\rightarrow B, BC\rightarrow E, ED\rightarrow A\}$，找出$r(R)$的所有候选码。
  - 属性集CD没有在函数依赖的右部出现，故$X=CD$为候选码的一部分；
  - 因$(CD)^+=CD\neq R$，故CD不是候选码；
  - 因没有在函数依赖右部出现但左部不出现的属性，故$Y=\varnothing$;
  - 在集合$R-X-Y=ABE$中寻找与X联合构成候选码的属性(集)：
    - $(\{A, CD\})^+=ACDBE=R$，故ACD为候选码；
    - $(\{B, CD\})^+=BCDEA=R$，故BCD为候选码；
    - $(\{E, CD\})^+=ACDBE=R$，故ECD为候选码。
  - 因此，关系模式$r(R)$的候选码有ACD、BCD和ECD。

判断属性集是否为候选码举例

- [例5.19] 设关系模式$R=\{A, B, C, D, E, G\}$，函数依赖集$F=\{B\rightarrow ADE, A\rightarrow BE, AC\rightarrow G, BC\rightarrow D\}$，找出$r(R)$的所有候选码。
  - 因C没有在函数依赖的右部出现，故$X=C$为候选码的一部分;
  - 因$C^{+}=C\neq R$，故C不是候选码;
  - 在函数依赖右部出现但左部不出现的属性有DEG, 故$Y=DEG$;
  - 在$R-X-Y=AB$中寻找与X联合起来构成候选码的属性(集):
    - $(\{A, C\})^{+}=ACBEGD=R$，故AC为候选码;
    - $(\{B, C\})^{+}=BCADEG=R$，故BC为候选码。
  - 因此，关系模式$r(R)$的候选码有AC和BC。

*正则覆盖：无关属性的概念

- **定义5.14** 给定函数依赖集F及$\alpha\rightarrow\beta\in F$，如果去除$\alpha$或$\beta$中的某属性$A$之后不会改变$F^+$，则称属性$A$是无关的。
- **定义5.15** 给定函数依赖集F及$\alpha\rightarrow\beta\in F$，若$A\in\alpha$，且$F$逻辑蕴涵$\{(\alpha-A)\rightarrow\beta\}$（即$\{(\alpha-A)\rightarrow\beta\}\in F^+$），则属性$A$在$\alpha$中是无关的（左无关）。
- **定义5.16** 给定函数依赖集F及$\alpha\rightarrow\beta\in F$，若$A\in\beta$，且$(F-\{\alpha\rightarrow\beta\})\cup\{\alpha\rightarrow(\beta-A)\}$逻辑蕴涵$F$（即$\{\alpha\rightarrow\beta\}\in((F-\{\alpha\rightarrow\beta\})\cup\{\alpha\rightarrow(\beta-A)\})^+$），则属性$A$在$\beta$中是无关的（右无关）。

*正则覆盖：定义与计算方法

- 对于正则覆盖，需做如下说明：
  - 可以证明F_c与F具有相同的闭包；
  - F_c不包含无关属性，每个依赖是最小的，且是必要的；
  - 正则覆盖不一定唯一。

无损连接分解-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

无损连接分解

- 定义5.18 给定关系模式$r(R)$及函数依赖集F，记$r_1(R_1)$、$r_2(R_2)$为由$r(R)$分解得到的子模式，如果对任意一个满足函数依赖集F的关系实例r都有

$$\prod_{R_1}(r) \bowtie \prod_{R_2}(r) = r ,$$

则称该分解对于F是无损连接的。

- 无损连接分解能够根据分解后的关系通过连接还原原来的关系实例。

- 如何判定一个分解是否是无损连接的？

无损连接分解

- 定义5.19 给定关系模式$r(R)$及函数依赖集F，则将$r(R)$分解成$r_1(R_1)$、$r_2(R_2)$的分解是无损连接分解，当且仅当F^+包含函数依赖$R_1 \cap R_2 \rightarrow R_1$或$R_1 \cap R_2 \rightarrow R_2$.
- 即：在F下，$R_1 \subseteq (R_1 \cap R_2)^+$ 或 $R_2 \subseteq (R_1 \cap R_2)^+$。
- 因此，当一个关系模式分解为两个关系模式时，该分解为无损连接分解的充要条件是两分解关系的公共属性包含$r_1(R_1)$的码 或 $r_2(R_2)$的码。
- 即： $R_1 \cap R_2$ 是关系 $r_1(R_1)$ 或 $r_2(R_2)$ 的超码。

无损连接分解

■ 如果将关系模式$r(R)$分解为M个(M>2)子关系模式$r_1(R_1)$、$r_2(R_2)$、…、$r_M(R_M)$，且$R_1 \cup R_2 \cup \ldots \cup R_M = R$ 。则如何判断此分解是不是无损连接分解？

■ 方法一：逐步连接判断法

- 先选择满足定义5.19 (即满足无损连接条件)的2个子关系模式进行无损连接，不仿假设将子关系模式$r_1(R_1)$和$r_2(R_2)$进行无损连接，并将连接后的关系模式记为$r_{12}(R_{12})$；
- 对连接后剩下的子关系模式$r_{12}(R_{12})$、$r_3(R_3)$、…、$r_M(R_M)$，重复上面的连接步骤，直到连接为一个关系模式——则该分解是无损连接分解；或者剩下的任意2个子关系模式之间都无法进行无损连接 ——则该分解是有损连接分解。

无损连接分解

- [例5.22] 假设关系模式$r(R)$=$r(A, B, C, D, E)$, F ={$A\rightarrow BC$, $CD\rightarrow E$, $B\rightarrow D$, $E\rightarrow A$}，则可将$r(R)$进行两种不同的分解:
  - 分解1：$r_1(R_1)=r_1(A, B, C)$，$r_2(R_2)=r_2(A, D, E)$；
  - 分解2：$r_1(R_1)=r_1(A, B, C)$，$r_2(R_2)=r_2(C, D, E)$。
- 对于分解1，$R_1 \cap R_2 = A$，且$A\rightarrow R_1$，故此分解是无损连接分解。
- 而对于分解2，$R_1 \cap R_2 = C$，且$C \not\rightarrow R_1$、$C \not\rightarrow R_2$，故此分解不是无损连接分解。

无损连接分解

■ 方法二：表格判断法（参考王珊《数据库系统概论》第4版）

- 设$R=\{A_1, A_2, \ldots, A_N\}$，$F=\{FD_1, FD_2, \ldots, FD_K\}$。

无损连接分解

- ■ 定义

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

ρ = {R_1<U_1 , F_1> , R_2<U_2 , F_2>, … , 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。

那么有效判别一个分解很多个小模式是无损分解呢？

■ 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

只是检测，不证明！

解

接性）

$$\rho = \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_k<U_k, F_k>\}$$

1.建立一个n列k行的矩阵

$$TB = \{C_{ij} \mid 若A_j \in U_i, C_{ij} = a_j, 否则C_{ij} = b_{ij}\}$$

	A_1	A_2	…	A_n
U_1 … U_k			C_{ij}	

无损连接分解

2.对F中每一个函数依赖X→Y，若TB中存在元组 t_1，t_2，使得 $t_1[X]=t_2[X]$，$t_1[Y]\neq t_2[Y]$，则对每一个$A_i \in Y$：

①若$t_1[A_i]$，$t_2[A_i]$中有一个等于a_j，则另一个也改为a_j；

②若①不成立，则取$t_2[A_i] = t_1[A_i]$（t_1的行号小于t_1）。

如：**C->D, T1[C]=T2[C],T1[D]?T2[D],则对每个D ∈{D},修改之**

	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

$U=\{A_1, A_2, \ldots, A_n\}$

$= \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_k<U_k, F_k>\}$

1. 建立一个n列k行的矩阵

$TB = \{C_{ij} \mid$ 若 $A_j \in U_i$, $C_{ij} = a_j$, 否则 $C_{ij} = b_{ij}\}$

	A_1	A_2	...	A_n
U_1				
...			C_{ij}	
U_k				

无损连接分解

3.反复执行 2.，直至：

①TB中出现一行为$a_1, a_2, \ldots, a_n$ 的一行。

② TB不再发生变化，且没有一行为$a_1, \ldots, a_n$。

在①情况下， ρ为无损分解，否则为有损分解。

	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

2. 对F中每一个函数依赖X→Y，若TB中存在元组t_1，t_2，使得$t_1[X]=t_2[X]$，$t_1[Y]\neq t_2[Y]$，则对每一个$A_i \in Y$：

①若$t_1[A_i]$，$t_2[A_i]$中有一个等于a_j，则另一个也改为a_j；

②若①不成立，则取$t_1[A_i] = t_2[A_i]$（t_2的行号小于t_1）。

3.反复执行 2.，直至：

①TB中出现一行为$a_1, a_2, \ldots, a_n$的一行。

② TB不再发生变化，且没有一行为$a_1, \ldots, a_n$。

在①情况下，ρ为无损分解，否则为有损分解。

无损连接分解

- 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

2. 对F中每一个函数依赖X→Y，若TB中存在元组t_1，t_2，使得$t_1[X]=t_2[X]$，$t_1[Y]\neq t_2[Y]$，则对每一个$A_i \in Y$：

①若$t_1[A_i]$，$t_2[A_i]$中有一个等于a_j，则另一个也改为a_j；

②若①不成立，则取$t_1[A_i] = t_2[A_i]$（t_2的行号小于t_1）。

无损连接分解

- 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

2. 对F中每一个函数依赖X→Y，若TB中存在元组t_1，t_2，使得$t_1[X]=t_2[X]$，$t_1[Y]\neq t_2[Y]$，则对每一个$A_i \in Y$:

①若$t_1[A_i]$，$t_2[A_i]$中有一个等于a_j，则另一个也改为a_j；

②若①不成立，则取$t_1[A_i] = t_2[A_i]$（t_2的行号小于t_1）。

无损连接分解

- 示例二：$U=\{A,B,C,D,E\}$,

$F=\{A\rightarrow C, B\rightarrow C, C\rightarrow D, DE\rightarrow C, CE\rightarrow A\}$

$\rho =\{(A, D), (A, B), (B, E), (C, D, E), (A, E)\}$

$B\rightarrow C$

	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_{52}	b_{13}	b_{54}	a_5

$C\rightarrow D$

	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_{52}	b_{13}	a_4	a_5

无损连接分解

- 示例二：$U=\{A,B,C,D,E\}$,
  $F=\{A\rightarrow C, B\rightarrow C, C\rightarrow D, DE\rightarrow C, CE\rightarrow A\}$
  $\rho=\{(A, D), (A, B), (B, E), (C, D, E), (A, E)\}$

$DE\rightarrow C$

	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	a_3	a_4	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	a_3	a_4	a_5

$CE\rightarrow A$

	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	a_1	a_2	a_3	a_4	a_5
CDE	a_1	b_{42}	a_3	a_4	a_5
AE	a_1	b_{52}	a_3	a_4	a_5

无损连接分解

■ 无损连接分解

● 定义（分解为两个关系模式证明）

关系模式R(U)，U_1，$U_2 \subseteq U$，$U_1 \cup U_2 = U$，r是R上的一个关系，$r_1=\prod_{U_1}(r)$，$r_2=\prod_{U_2}(r)$，若$r = r_1 \bowtie r_2$，则称(r_1, r_2)是r的一个无损连接分解。

注：$r \subseteq \prod_{U_1}(r) \bowtie \prod_{U_2}(r)$

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

无损连接分解

- **定义5.19** 给定关系模式$r(R)$及函数依赖集F，则将$r(R)$分解成$r_1(R_1)$、$r_2(R_2)$的分解是无损连接分解，当且仅当F^+包含函数依赖$R_1 \cap R_2 \rightarrow R_1$或$R_1 \cap R_2 \rightarrow R_2$.
- 即：在F下，$R_1 \subseteq (R_1 \cap R_2)^+$ 或 $R_2 \subseteq (R_1 \cap R_2)^+$。
- 因此，当一个关系模式分解为两个关系模式时，该分解为无损连接分解的充要条件是两分解关系的公共属性包含$r_1(R_1)$的码 或 $r_2(R_2)$的码。

- 算法：（判别一个分解的无损连接性）

$U=\{A_1, A_2, \ldots, A_n\}$

$\rho = \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_k<U_k, F_k>\}$

1. 建立一个n列k行的矩阵

$TB = \{C_{ij} \mid 若A_j \in U_i, C_{ij} = a_j, 否则C_{ij} = b_{ij}\}$

	A_1	A_2	…	A_n
U_1 … U_k			C_{ij}	

证明：

无损连接分解

- **定理**

关系模式**R(U)**的分解$\rho=\{R_1, R_2\}$，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

无损连接分解一例题

$R=ABC$, $F=\{A \rightarrow B\}$,

$\rho_1=\{R_1(AB), R_2(AC)\}$ 是否是无损连接分解？

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

由 **$A \rightarrow B$**， 得到ρ_1是无损连接分解

$\rho_2=\{R_1(AB), R_2(BC)\}$是否是无损连接分解？

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

$B \rightarrow A$, $B \rightarrow C$均不成立，所以ρ_2不是无损连接分解

保持依赖分解

- 关系数据库模式分解的另一个目标是**保持依赖**。
- **定义5.20** 给定关系模式$r(R)$及函数依赖集F，$r_1(R_1)$, $r_2(R_2)$, ..., $r_n(R_n)$为$r(R)$的分解。**F在R_i的投影**为闭包F^+中所有只包含R_i属性的函数依赖的集合，记为F_i。即如果$\alpha \rightarrow \beta$在F_i中，则α和β的所有属性均在R_i中。
- **定义5.21** 称具有函数依赖集F的关系模式$r(R)$的分解$r_1(R_1)$, $r_2(R_2)$, ..., $r_n(R_n)$为**保持依赖分解**，当且仅当$(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$。

保持依赖分解

- [例5.23] 设关系模式$r(R)=r(A, B, C)$，$F=\{A \rightarrow B, B \rightarrow C\}$，有两种分解：
  - $r_1(R_1)=r_1(A, B)$、$r_2(R_2)=r_2(B, C)$
    - $F_1=\{A \rightarrow B\}$、$F_2=\{B \rightarrow C\}$
    - 该分解**保持函数依赖**，因为$(F_1 \cup F_2)^+=F^+$。
  - $r_1(R_1)=r_1(A, B)$、$r_2(R_2)=r_2(A, C)$。
    - $F_1=\{A \rightarrow B\}$、$F_2=\{A \rightarrow C\}$ （注：$\{A \rightarrow C\} \subseteq F^+$）
    - 该分解**不保持函数依赖**。因为分解后函数依赖$B \rightarrow C$被丢失。

示例

- 设有关系模式R=ABCDE，F={A→D， E→D， D→B， BC→D， DC→A}，现有如下分解：ρ={AB, AE,CE, BCD,AC}。请给出求解过程。
  - 判断上述分解ρ是否无损连接。
  - 给出函数依赖集F在ρ的各个模式上的投影。
  - 判断分解ρ是否保持函数依赖。

示例

示例

■ 设有关系模式R=ABCDE，F={A→D，E→D，D→B，BC→D，DC→A}，现有如下分解：ρ={AB, AE,CE, BCD,AC}。请给出求解过程。

- 判断上述分解ρ是否无损连接。
- 给出函数依赖集F在ρ的各个模式上的投影。
- 判断分解ρ是否保持函数依赖。

（1）

	A	B	C	D	E
AB	a1	a2	b13	b14→a4	b15
AE	a1	b22→a2	b23	b24→b14→a4	a5
CE	b31→a1	b32→a2	a3	b34→b14→a4	a5
BCD	b41→a1	a2	a3	a4	b45
AC	a1	b52→a2	a3	b54→b14→a4	b55

由函数依赖A→D，可将b24和b54都改为b14，由函数依赖E→D，可将b34改为b14，由函数依赖D→B，可将b22、b32和b52改为a2，由函数依赖BC→D，可将所有的b14改为a4，由函数依赖DC→A，可将b31和b41改为a1，这时第3行成为全a行，所以该分解ρ**具有无损连接性**。

示例

（2） 函数依赖集F在各关系模式上的投影如下：

在AB上的投影：F1={ A→B }

在AE上的投影：F2=∅

在CE上的投影：F3=∅

在BCD上的投影：F4={ BC→D, D→B }

在AC上的投影：F5=∅

示例

- 设有关系模式R=ABCDE，F={A→D，E→D，D→B，BC→D，DC→A}，现有如下分解：ρ={AB, AE,CE, BCD,AC}。请给出求解过程。
  - 判断上述分解ρ是否无损连接。
  - 给出函数依赖集F在ρ的各个模式上的投影。
  - 判断分解ρ是否保持函数依赖。

（3）因F1 ∪ F2 ∪ F3 ∪ F4∪ F5={A→B, BC→D, D→B }与F不等价，所以该**分解不具有函数依赖保持。**

总结

- **函数依赖集闭包：** **Armstrong公理。**
- **函数依赖集F下属性集A的闭包：**
  - 验证$\alpha\rightarrow\beta$是否在F^+中；计算F^+。
  - 判断α是否为$r(R)$的超码、**候选码** ——候选码的计算方法。
- **(左、右)无关属性、正则覆盖。**
- **无损连接分解：** $R_1 \cap R_2$ 是关系 $r_1(R_1)$ 或 $r_2(R_2)$ 的**超码**。
- **保持依赖分解：** $(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$。
- **BCNF：** F^+中非平凡函数依赖的决定属性集α都包含**候选码.**
  - 不存在任何属性(包括主属性和非主属性)对**候选码**的**部分依赖**和**传递依赖**，以及主属性之间的**传递依赖**。
  - 模式分解：是**无损连接分解**，可能不是**保持依赖分解**。
- **3NF：** 允许存在主属性对**候选码**的**传递依赖**和**部分依赖**。
  - 模式分解：是**无损连接分解**，且是**保持依赖分解**。

目 录

模式分解方法

- 数据库设计目标为(基于函数依赖):
  - BCNF
  - 无损连接
  - 保持依赖
- 如果不能同时达到这3个目标，就需要根据实际应用需求在BCNF和3NF中做出选择。
- 主要模式分解算法
  - BCNF分解
  - 3NF分解

BCNF分解

- 设$r(R)$为关系模式，$r(R)\notin$BCNF，若非平凡函数依赖$\alpha\rightarrow\beta$违反了BCNF的函数依赖条件(即α不是**超码**)，则将$r(R)$分解为$r_1(R_1)$和$r_2(R_2)$，其中：
  - $R_1=\alpha\beta$ $F_1=\{\alpha\rightarrow\beta\}$ —— 如果$\alpha\cap\beta=\varnothing$，则$\alpha$是候选码
  - $R_2=R-(\beta-\alpha)$ —— 如果$\alpha\cap\beta=\varnothing$，则$R_2=R-\beta$
  - 若$r_2(R_2)$不属于BCNF，则继续分解下去，直到所有结果模式都为BCNF。

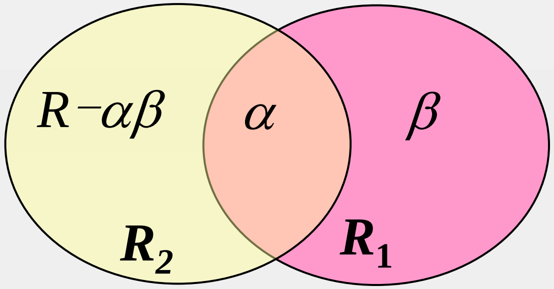


图5-14 不满足BCNF的关系分解

■ 设$r(R)$为关系模式，$r(R)\notin$BCNF，若非平凡函数依赖$\alpha\to\beta$违反了BCNF的函数依赖条件(即α不是超码)，则将$r(R)$分解为$r_1(R_1)$和$r_2(R_2)$，其中：

- $R_1=\alpha\beta$　$F_1=\{\alpha\to\beta\}$ —— 如果$\alpha\cap\beta=\varnothing$，则$\alpha$是候选码
- $R_2=R-(\beta-\alpha)$ —— 如果$\alpha\cap\beta=\varnothing$，则$R_2=R-\beta$

BCNF分解举例

■ [例5.24] $r(R)=r(A, B, C)$，$F=\{AB\to C, C\to A\}$，判断关系模式$r(R)$是否属于BCNF范式？如果不是，则进行BCNF分解。

- [例5.10]已经证明$r(R)\notin$BCNF(因为候选码为AB或BC，所以$C\to A$的决定属性C不是超码)。按上述算法，$r(R)$可分解为
  - ➢$r_1(R_1)=r_1(A, C)$，$F_1=\{C\to A\}$

    ——该关系$r_1(R_1)$中，C是候选码
  - ➢$r_2(R_2)=r_2(B, C)$，$F_2=\{\varnothing\}$

    ——该关系$r_2(R_2)$中，BC是候选码
- 分解后的$r_1(R_1)$和$r_2(R_2)$都属于BCNF，不需再做分解。
- 注意：F中的函数依赖关系$AB\to C$，在分解后丢失了！

⊄

$R=\{ABCDE\}$
$\alpha\to\beta$
$\alpha=AB$
$\beta=CD$
$AB\to CD$
$AB\to\{ABCDE\}\notin F^+$
$\alpha\cap\beta=\phi$

BCNF分解算法

- BCNF分解算法的形式化描述如下：

第一次循环时，$r_i(R_i)$就是$r(R)$

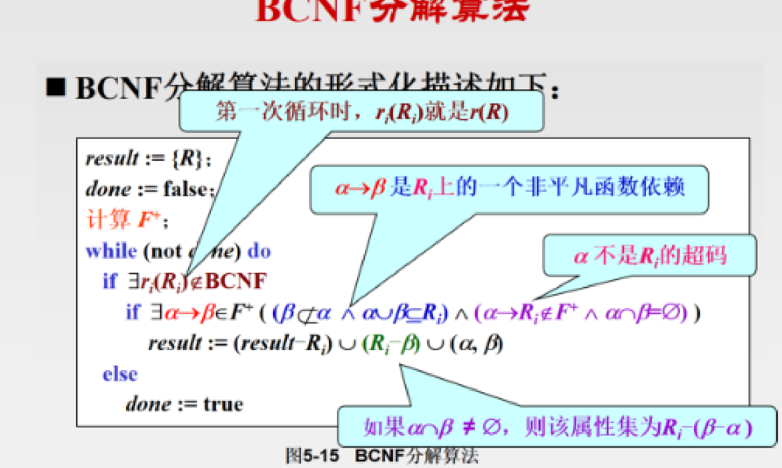
$\alpha\to\beta$是R_i上的一个非平凡函数依赖

α不是R_i的超码

如果$\alpha\cap\beta\neq\varnothing$，则该属性集为$R_i-(\beta-\alpha)$

```
result := {R};
done := false;
计算 F+;
while (not done) do
  if ∃r_i(R_i)∉BCNF
    if ∃α→β∈F+ ( (β⊄α ∧ α∪β⊆R_i) ∧ (α→R_i∉F+ ∧ α∩β=∅) )
      result := (result−R_i) ∪ (R_i−β) ∪ (α, β)
  else
    done := true
```

图5-15 BCNF分解算法

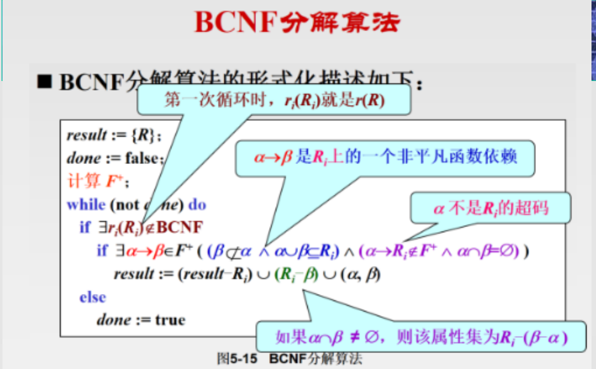


NF分解举例

D, G, H)， F={$A\rightarrow BC$, $DG\rightarrow H$, $D\rightarrow A$},

？如果不是，则进行BCNF分解。

- $r(R)\notin$BCNF（因为候选码为DG，所以$A\rightarrow BC$的决定属性A不是超码）。按上述算法，$r(R)$可分解为
  - ➢$r_1(R_1)=r_1(A, B, C)$， F_1={$A\rightarrow BC$} —— A是候选码
  - ➢$r_2(R_2)=r_2(A, D, G, H)$， F_2={$DG\rightarrow H$, $D\rightarrow A$}—DG是候选码
- $r_2(R_2)\notin$BCNF（因为$D\rightarrow A$的决定属性D不是超码）。按上述算法，$r_2(R_2)$可分解为
  - ➢$r_{21}(R_{21})=r_{21}(D, A)$， F_{21}={$D\rightarrow A$} —— D是候选码
  - ➢$r_{22}(R_{22})=r_{22}(D, G, H)$， F_{22}={$DG\rightarrow H$} —— DG是候选码
- 最后，$r_1(\underline{A}, B, C)$、$r_{21}(\underline{D}, A)$和$r_{22}(\underline{D}, \underline{G}, H)$都属于BCNF。



BCNF分解举例

***ACD*是候选码！**

■ **例：** $r(R)=r(A, B, C, D, G, H)$， $F=\{AB\rightarrow GH, CD\rightarrow GH, D\rightarrow B\}$， $r(R)$是否属于BCNF范式？如果不是，则进行BCNF分解。

- $r(R)\notin$BCNF，因为$AB\rightarrow GH$的决定属性AB不是超码。$r(R)$可分解为：
  - $r_1(R_1)=r_1(A, B, G, H)$， $F_1=\{AB\rightarrow GH\}$ —— AB是候选码
  - $r_2(R_2)=r_2(A, B, C, D)$， $F_2=\{D\rightarrow B\}$ —— ACD是候选码

  —— 丢失函数依赖$CD\rightarrow GH$！
- $r_2(R_2)\notin$BCNF（$D\rightarrow B$的决定属性D不是超码）。$r_2(R_2)$可分解为：
  - $r_{21}(R_{21})=r_{21}(D, B)$， $F_{21}=\{D\rightarrow B\}$ —— D是候选码
  - $r_{22}(R_{22})=r_{22}(A, C, D)$ ， $F_{22}=\{\varnothing\}$ —— ACD是候选码
- 最后，$r_1(\underline{A}, \underline{B}, G, H)$、$r_{21}(\underline{D}, B)$和$r_{22}(\underline{A}, \underline{C}, \underline{D})$都属于BCNF。

BCNF分解

- 上述算法得到的分解不仅是BCNF分解，而且是无损分解（但可能不是保持函数依赖分解）。
- 算法中使用的函数依赖集是F^+而不是F。
- 用该算法生成的BCNF分解不是唯一的。

目 录

模式求精的必要性

- E-R图设计是一个复杂且主观的过程，并且有些约束关系并不能通过E-R图来表达。一些不“好”的关系模式可能忽略数据之间的约束关系而产生冗余，特别是在大型数据库模式设计时更是如此。
- 另外，关系模式不一定都是严格地由E-R图转换得到，也可能是设计者的即席产生。因此，有必要对关系模式进行模式求精。
- **模式求精**是运用关系理论(如函数依赖理论、多值依赖理论等)对已有关系模式进行结构调整、分解、合并和优化的过程，以满足应用系统的功能及性能等需求。

模式求精步骤

■ 基于函数依赖理论的模式求精步骤：

- **确定函数依赖**。根据需求分析得到的数据需求，确定关系模式内部各属性之间以及不同关系模式的属性之间存在的数据依赖关系。
- **确定关系模式所属范式**。按照数据依赖关系对关系模式进行分析，检测是否存在部分依赖或传递依赖，以确定该模式属于第几范式。
- **分析是否满足应用需求**。按照需求分析得到的数据处理要求，分析现有模式是否满足应用需求，并决定是否需要进行模式合并或分解。
- **模式分解**。根据范式要求(是选择BCNF还是3NF)，运用规范化方法将关系模式分解成所要求的关系模式。
- **模式合并**。在分解过程中可能进行模式合并。如当查询经常涉及到多个关系模式的属性时，系统将经常进行连接操作，而连接运算的代价是相当高的。此时，可考虑将这几个关系合并为一个关系。

模式求精举例

- [例5.29] 假设大学选课系统中课程与教师的关系模式设计为:
  - CourseTeacher (<u>courseNo</u>, courseName, creditHour, courseHour, <u>teacherNo</u>, teacherName, title, degree, teachNumber)

  其中属性集{courseNo, teacherNo}是主码。试对该……模式求精，以达到BCNF/3NF范式要求。

  讲授次数

- 步骤1. 分析函数依赖关系及判断范式
  - 关系模式CourseTeacher存在以下函数依赖:
    - courseNo → courseName, creditHour, courseHour
    - teacherNo→ teacherName, title, degree
    - {courseNo, teacherNo} → teachNumber
  - 显然，存在非主属性对主码的部分依赖，故CourseTeacher不属于2NF范式，更不属于BCNF范式。

模式求精举例

■ 步骤2. 模式分解

- 由于存在部分函数依赖：courseNo → courseName, creditHour, courseHour，违背了BCNF/3NF条件，依BCNF/3NF分解算法，可将关系模式CourseTeacher分解为以下两个关系模式：
  - Course (courseNo, courseName, creditHour, courseHour)；
  - Teaching (*courseNo*, teacherNo, teacherName, title, degree, teachNumber)
- 可验证关系模式Course已满足BCNF/3NF要求，且是**无损分解**(因为公共属性courseNo是关系模式Course的主码)。
- 而在关系模式Teaching中，由于存在部分函数依赖：teacherNo→ teacherName, title, degree，因此可以进一步分解为：
  - Teacher (teacherNo, teacherName, title, degree)
  - NewTeaching (*courseNo*, *teacherNo*, teachNumber)
- 可验证关系模式Teacher和NewTeaching都已满足BCNF/3NF要求，且是**无损分解**(因为公共属性teacherNo是关系模式Teacher的主码)。

模式求精举例

- 综合上述分解结果，关系模式CourseTeacher可以分解为如下满足BCNF/3NF要求的三个关系模式：
  - **Course (courseNo, courseName, creditHour, courseHour)**
  - **Teacher (teacherNo, teacherName, title, degree)**
  - **NewTeaching (*courseNo*, *teacherNo*, teachNumber)**

模式求精是数据库设计过程中非常重要的一步，设计者应在关系数据理论的指导下检查和改进设计中存在的不足和缺陷，以保证最终的设计结果尽可能地满足应用需求。

总结

- **数据库设计目标** ——基于函数依赖
  - BCNF；
  - 无损连接；
  - 保持依赖。
- 如果不能同时达到这3个目标，则需要在BCNF、3NF间选择。
- **模式分解算法**
  - BCNF分解；
  - 3NF分解。
- **模式求精**
  - 确定函数依赖；
  - 确定关系模式所属范式；
  - 分析是否满足应用需求；
  - 模式分解；
  - 模式合并。