

数据管理基础

第6章 关系数据理论

(函数依赖 与 码)

智能软件与工程学院



❑ 定义6.1 函数依赖

- 函数依赖的定义

- 非平凡函数依赖，平凡函数依赖

❑ 定义6.2 完全函数依赖，部分函数依赖

❑ 定义6.3 传递函数依赖（直接函数依赖）

❑ 定义6.4 候选码

- 候选码、超码

- 全码

- 主属性，非主属性/非码属性

❑ 定义6.5 外码

常用的符号表示方法

A, B, C	属性名
ABC	代表由 A 、 B 、 C 三个属性组成的属性集合，即： $ABC = \{A, B, C\}$ 有时候也用(...)代替其中的{...}，即表示为： $ABC = (A, B, C)$
X, Y, Z	关系的属性子集(subset)
XY	表示 X 和 Y 的并集，即： $XY = X \text{ union } Y$
R, S, T	关系名 (关系模式)
$R(U)$	关系模式：关系名 R ，关系中的属性集合 U
$R(U, F)$	关系模式：关系名 R ，属性集合 U ，函数依赖集 F
r, s, t	关系实例 (一个关系中的元组集合)
r_1, r_2, r_3	关系中的元组
$r_1[A]$	元组 r_1 在属性(集) A 上的取值

函数依赖 1

- ❑ 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 $X \rightarrow Y$ 。
- ❑ 如果存在函数依赖 $X \rightarrow Y$
 - X 称为这个函数依赖的**决定因素** (Determinant)
 - Y 称为这个函数依赖的**依赖因素**
- ❑ 若 $X \rightarrow Y$ ，并且 $Y \rightarrow X$ ，则记为 $X \leftrightarrow Y$
- ❑ 若 Y 不函数依赖于 X ，则记为 $X \nrightarrow Y$

□ [例] 学生关系 **Student(Sno, Sname, Ssex, Sage, Sdept)**

➤ 假设不允许学生重名，则有：

$Sno \rightarrow Ssex$

$Sno \rightarrow Sage$

$Sno \rightarrow Sdept$

$Sno \leftrightarrow Sname$

$Sname \rightarrow Ssex$

$Sname \rightarrow Sage$

$Sname \rightarrow Sdept$

➤ 但是

$Ssex \nrightarrow Sage$

$Ssex \nrightarrow Sdept$

- **Sno**是学生的码
- 学生不允许同名
- 每个学生只能就读于一个院系

函数依赖 3

- ❑ 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。
- ❑ 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 \rightarrow 年龄”：只有在不允许有同名人的条件下成立
 - “学号 \rightarrow 姓名”：只有在每个学生只有一个姓名的条件下成立
- ❑ 数据库设计者可以根据现实世界中的语义约束作强制的规定。
 - 例如规定不允许同名人的出现，函数依赖“姓名 \rightarrow 年龄”成立。
 - 所插入的元组必须满足规定的函数依赖，若发现有同名人的存在，则拒绝插入该元组。

函数依赖 4

违背了 $Sno \rightarrow Sname$

Sno	Sname	Ssex	Sage	Sdept
S1	张三	男	20	计算机系
S1	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

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 ，
 - $X \rightarrow Y$ 但 $Y \not\subseteq X$ ，则称 $X \rightarrow Y$ 是‘非平凡函数依赖’
 - $X \rightarrow Y$ 但 $Y \subseteq X$ ，则称 $X \rightarrow Y$ 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。因此若不特别声明，我们总是讨论非平凡函数依赖。

- 例：在关系 $SC(Sno, Cno, Grade)$ 中，
 - 有非平凡函数依赖： $(Sno, Cno) \rightarrow Grade$

 - 有平凡函数依赖：
 $(Sno, Cno) \rightarrow Sno$
 $(Sno, Cno) \rightarrow Cno$

完全函数依赖 与 部分函数依赖

□ 定义6.2 在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' , 都有 $X' \nrightarrow Y$, 则称 Y 对 X 完全函数依赖, 记作 $X \xrightarrow{F} Y$

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

□ 由于: $Sno \nrightarrow Grade$, $Cno \nrightarrow Grade$, $Sno \rightarrow Sdept$

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

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

□ 定义6.3 在 $R(U)$ 中, 如果 $X \rightarrow Y (Y \not\subseteq X)$, $Y \nrightarrow X$, $Y \rightarrow Z$, $Z \not\subseteq Y$, 则称 Z 对 X 传递函数依赖(transitive functional dependency)。记为:

$X \xrightarrow{\text{传递}} Z$

- 在定义里加上条件 $Y \nrightarrow X$, 是因为: 如果 $X \rightarrow Y$ 且 $Y \rightarrow X$ 即 $X \leftrightarrow Y$, 则 Z 直接函数依赖于 X , 而不是传递函数依赖。
- 在定义里加上条件 $Y \not\subseteq X$ 和 $Z \not\subseteq Y$, 是因为: 如果 $Y \subseteq X$ 或 $Z \subseteq Y$, 则 $X \rightarrow Z$ 就是一个直接函数依赖, 并且是一个部分函数依赖或平凡函数依赖。
- 一般情况下, 传递函数依赖与直接函数依赖在表示上仍然使用函数依赖的表示方法, 并没有区分。只有在规范化设计中需要时, 我们才区分它们。

❑ [例] 在关系 **Std(Sno, Sdept, Mname)** 中

➤ 有: $Sno \rightarrow Sdept, Sdept \nrightarrow Sno, Sdept \rightarrow Mname$

➤ 所以, $Sno \xrightarrow{\text{传递}} Mname$

❑ [例] 在学生关系 **Student(Sno, Sname, Ssex, Sage, Sdept)** 中

➤ 有: $Sno \leftrightarrow Sname$

➤ 所以, $Sno \rightarrow Sdept$ 和 $Sname \rightarrow Sdept$ 都是直接函数依赖

❑ 之所以要区分完全FD与部分FD、直接FD与传递FD, 是因为某些不好的部分FD或传递FD会带来过高的数据冗余, 并因此产生插入、修改、删除等异常。

- 定义6.4 设 K 为 $R(U, F)$ 中的属性或属性组合。若 $K \xrightarrow{F} U$ ，则 K 称为 R 的一个**候选码**(Candidate Key)，简称‘**码**’。
- 如果 $K \rightarrow U$ （可能是部分函数依赖，也可能是完全函数依赖），则 K 称为 R 的一个**超码**（Superkey）。
 - 候选码是最小的超码，候选码的真子集一定不是超码
 - 候选码的超集是超码
 - 如果关系 R 的所有属性 U 是 R 的码，称为**全码**（all-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)。

□ [例6.2]

- $S(\underline{Sno}, Sdept, Sage)$ 中, 单个属性Sno是码 (也是超码), $(Sno, Sdept)$ 是超码但不是码;
- $SC(\underline{Sno}, \underline{Cno}, Grade)$ 中, (Sno, Cno) 是码, $(Sno, Cno, Grade)$ 是超码但不是码, $(Sno, Grade)$ 既不是码也不是超码。

□ [例] Student(Sno, Sname, Ssex, Sage, Sdept)中

- 单个属性Sno是码, 单个属性Sname也是码 (不允许学生同名)

□ [例6.3] 关系 $R(\text{演奏者}P, \text{作品}W, \text{听众}A)$ 中

- 一个演奏者可以演奏多个作品
- 某一作品可被多个演奏者演奏
- 听众可以欣赏不同演奏者的不同作品
- 关系 R 的码为 (P, W, A) , 即**All-Key**

*

□ 主属性 与 非主属性

- 包含在任何一个候选码中的属性，称为**主属性**（Prime attribute）
- 不包含在任何码中的属性称为**非主属性**（Nonprime attribute）或**非码属性**（Non-key attribute）

□ [例6.2]

- $S(\underline{Sno}, Sdept, Sage)$ 中，只有一个候选码Sno，所以：Sno是关系S的主属性，Sdept和Sage是关系S的2个非主属性；
- $SC(\underline{Sno}, \underline{Cno}, Grade)$ 中，只有一个候选码(Sno, Cno)，所以：Sno和Cno是关系SC的2个主属性，Grade是关系SC的非主属性。

□ [例] Student(Sno, Sname, Ssex, Sage, Sdept)中

- 有两个候选码：Sno 和 Sname（不允许学生同名）
- 所以：Sno 和 Sname是关系Student的2个主属性，Ssex, Sage, Sdept是关系Student的3个非主属性

□ [例6.3] 关系 R (演奏者 P , 作品 W , 听众 A) 中

- 只有一个全码(P, W, A)
- 所以： P, W, A 都是关系 R 的主属性，在关系 R 中没有非主属性*

外码

- ❑ 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。
 - $SC(\underline{Sno}, Cno, Grade)$ 中， Sno 不是码， Sno 是 $S(\underline{Sno}, Sdept, Sage)$ 的码，则 Sno 是 SC 的外码
- ❑ 主码与外部码一起提供了表示关系间联系的手段

函数依赖的发现

- ❑ 函数依赖的存在性判定
- ❑ 如何发现一个关系上的函数依赖?
- ❑ 函数依赖发现的例子

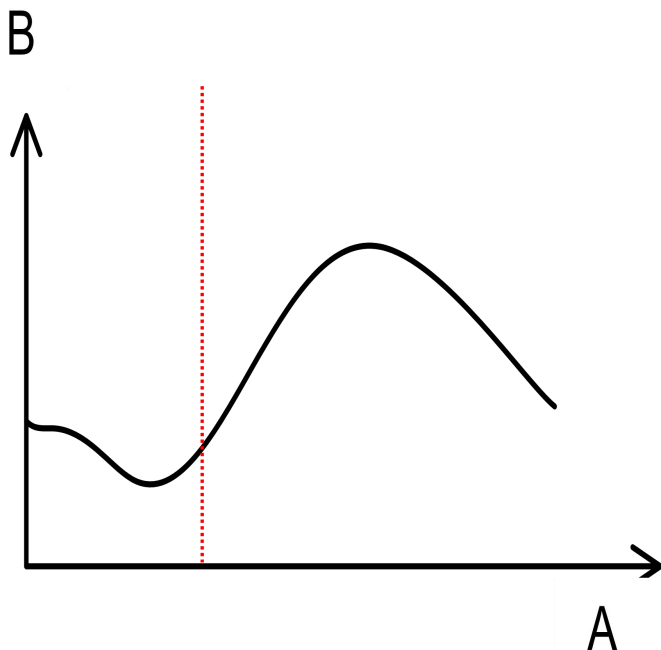
函数依赖

- 假设在关系模式 $R(U)$ 上存在函数依赖: $X \rightarrow Y$
- r 是依据关系模式 $R(U)$ 建立起来的任意一个关系, 那么关系 r 必满足:
 - 从关系 r 中任取两个元组 r_1 和 r_2 , 如果元组 r_1 在 X 这组属性上的取值 $r_1[X]$ 等于元组 r_2 在 X 这组属性上的取值 $r_2[X]$, 即: $r_1[X] = r_2[X]$
 - 则它们在 Y 属性组上的取值也必定相等, 即: $r_1[Y] = r_2[Y]$

$X \rightarrow Y$ if and only if For any rows r_1 and r_2 in any instance of relation R , if $r_1[X] = r_2[X]$ then $r_1[Y] = r_2[Y]$.

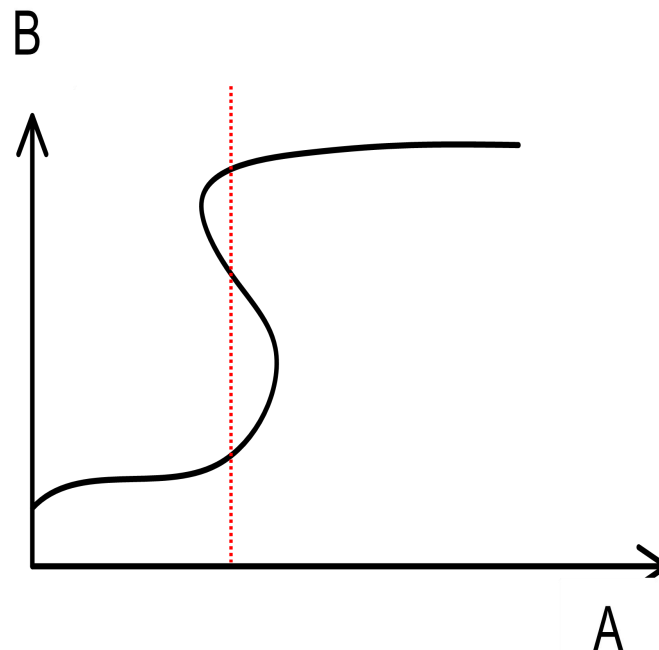
Graphical depiction of functional dependency

- 横坐标是 $\text{domain}(A)$ ，纵坐标是 $\text{domain}(B)$ ，图形反映了在一个关系中的A和B之间的取值对应关系。



A functionally determines B
($A \rightarrow B$)

- Because of
Each value of A corresponds to only one value of B.



A doesn't functionally determines B
 $A \nrightarrow B$

- Because of
Some values of A correspond to more than one value of B.

函数依赖

- 函数依赖是对一个关系中属性与属性之间的数据取值约束的抽象，定于函数依赖的目的是为了确保今后保存在该关系中的数据都能满足这些函数依赖的约束要求。
- 函数依赖是语义范畴上的概念，只有根据属性间固有的语义联系才能归纳出与客观事实相符合的函数依赖关系，而不是仅从现有的一个或若干个关系实例中得出的结论。
- 特定的关系实例虽然不能用于函数依赖的发现，但可以用于否定某些函数依赖。例如，对于如下的关系实例S

关系 S

sno	sn	sd	sa
S0001	王剑飞	CS	17
S0002	陈瑛	MA	17
S0003	方世觉	CS	19

函数依赖

❑ 在关系模式**S(sno,sn,sd,sa)**上，**可能存在**的函数依赖有很多，例如：

$sno \rightarrow sn$, $sn \rightarrow sd$, $sd \rightarrow sn$, $\{sno, sn\} \rightarrow sd$,

❑ 但是，根据下面的关系实例**S** (依据上述的关系模式而建):

sno	sn	sd	sa
S0001	王剑飞	CS	17
S0002	陈瑛	MA	17
S0003	方世觉	CS	19

分析其中元组的取值情况，可以得到如下结论：在关系**S**上，

- 1) 可以确认‘不存在以下的函数依赖’（或者说：可以否定以下函数依赖的存在）：
 $sd \rightarrow sno$, $sa \rightarrow sd$,
- 2) 虽然不能否定（原因：找不到不存在的理由），但也不能确认以下函数依赖的成立（只能说‘可能存在’）：
 $sno \rightarrow sn$, $sn \rightarrow sa$,

函数依赖

- ❑ 以下述的学生选课关系SCG为例，请判断其中可能存在/或不可能存在的函数依赖

sno	sname	dept	sage	cno	cname	grade
S0001	王剑飞	CS	17	C101	ABC	5
S0001	王剑飞	CS	17	C102	ACD	5
S0001	王剑飞	CS	17	C103	BBC	4
S0001	王剑飞	CS	17	C105	AEF	3
S0001	王剑飞	CS	17	C110	BCF	4
S0002	陈瑛	MA	17	C103	BBC	3
S0002	陈瑛	MA	17	C105	AEF	3
S0003	方世觉	CS	19	C107	BHD	4

The SCG database

sno	sname	dept	sage	cno	cname	grade
S0001	王剑飞	CS	17	C101	ABC	5
S0001	王剑飞	CS	17	C102	ACD	5
S0001	王剑飞	CS	17	C103	BBC	4
S0001	王剑飞	CS	17	C105	AEF	3
S0001	王剑飞	CS	17	C110	BCF	4
S0002	陈瑛	MA	17	C103	BBC	3
S0002	陈瑛	MA	17	C105	AEF	3
S0003	方世觉	CS	19	C107	BHD	4

$sno \rightarrow sname$?

$sno \rightarrow dept$?

$sno \rightarrow cno$?

$cno \rightarrow cname$?

$cno \rightarrow sno$?

➤ 请判断上述的函数依赖是否可能成立?

The SCG database

sno	sname	dept	sage	cno	cname	grade
S0001	王剑飞	CS	17	C101	ABC	5
S0001	王剑飞	CS	17	C102	ACD	5
S0001	王剑飞	CS	17	C103	BBC	4
S0001	王剑飞	CS	17	C105	AEF	3
S0001	王剑飞	CS	17	C110	BCF	4
S0002	陈瑛	MA	17	C103	BBC	3
S0002	陈瑛	MA	17	C105	AEF	3
S0003	方世觉	CS	19	C107	BHD	4

$\text{sno} \rightarrow \text{sname}$

$\text{sno} \rightarrow \text{dept}$

$\text{sno} \rightarrow \text{cno} \quad \times$

$\text{cno} \rightarrow \text{cname}$

$\text{cno} \rightarrow \text{sno} \quad \times$

既不能否认、也不能确定这些函数依赖的存在

但可以确定，这些函数依赖是不成立的

函数依赖的发现

□ 如何发现一个关系中的函数依赖？

- 对关系中各个属性的语义进行分析，寻找它们相互之间存在的语义联系，并据此发现在它们之间存在的函数依赖；
- 除了上述方法外，也可以分析属性与属性之间在取值对应关系上的数量特征，进而判断它们之间是否存在某种函数依赖。

□ 属性之间的取值对应关系的数量特征有三种：

- ① ‘一 对 一’ (1:1)
- ② ‘一 对 多’ (1:n) 或者称 ‘多 对 一’ (n:1)
- ③ ‘多 对 多’ (m:n)

□ 说明：

- ‘数量特征’ 只有 ‘一’ 和 ‘多’ 这两种类型，一般用数字 1 代表 ‘一’，用一个英文字母代表 ‘多’；
- ‘一对多’ 要确定方向，标注清楚哪一个是 ‘一’，哪一个是 ‘多’。

函数依赖的发现

□ 设 X 、 Y 是关系 R 的两个属性(集)

- 1) **一对一 (1:1)**: 如果“一个 X 值, 最多只能有唯一一个 Y 值与之对应; 一个 Y 值, 也最多只能有唯一一个 X 值与之对应”, 则称“ X 与 Y 是‘一对一’”。
- 2) **一对多 (1:n)**: 如果“一个 X 值, 允许有多个 Y 值与之对应; 一个 Y 值, 却最多只能有唯一一个 X 值与之对应”, 则称“从 X 到 Y 是‘一对多’” (或者称“从 Y 到 X 是‘多对一’”)。
- 3) **多对多 (m:n)**: 如果“一个 X 值, 允许有多个 Y 值与之对应; 一个 Y 值, 也允许有多个 X 值与之对应”, 则称“ X 与 Y 是‘多对多’”。

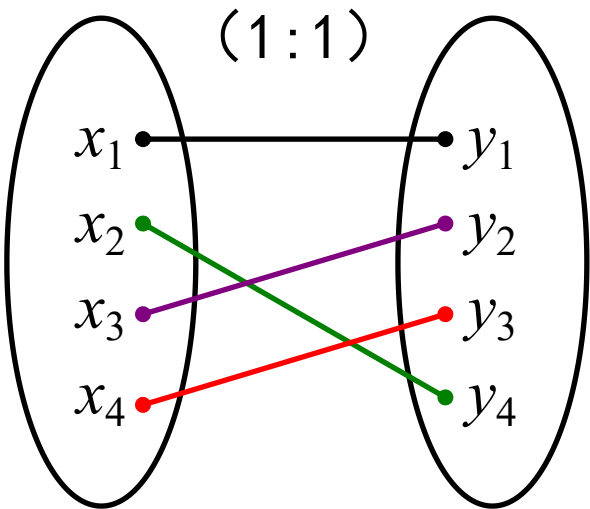
□ 说明:

- 在‘多对多’中允许存在‘多对一’现象, 例如: 允许某个 X 值 x_1 只对应到唯一的一个 Y 值 y_1 , 但 y_1 可能对应着多个不同的 X 值(包括 x_1);
- 同理, 在‘多对一’中允许存在‘一对一’现象;
- 不管关系 R 中的元组集合如何变化, X 和 Y 在关系 R 中的取值对应关系的‘数量特征’是确定不变的!

[例8-2] 根据下列具体的关系实例，判断其中可能存在哪些函数依赖？

T_1

X	Y
x ₁	y ₁
x ₂	y ₄
x ₃	y ₂
x ₄	y ₃

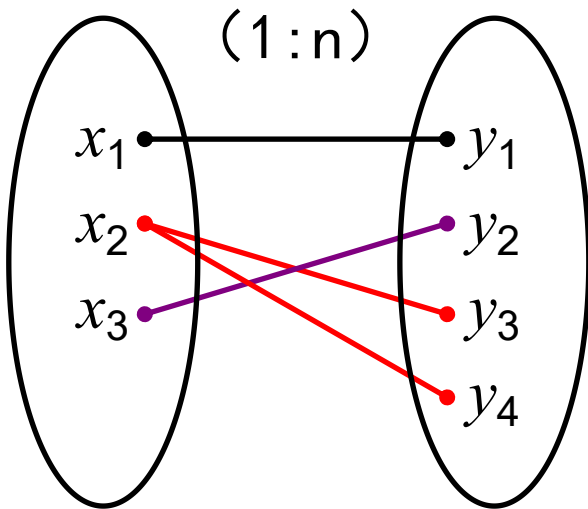


$X \rightarrow Y$

$Y \rightarrow X$

T_2

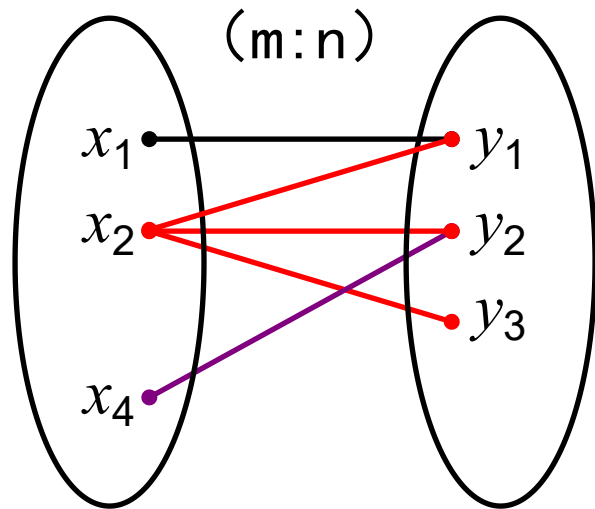
X	Y
x ₁	y ₁
x ₂	y ₃
x ₂	y ₄
x ₃	y ₂



$Y \rightarrow X$

T_3

X	Y
x ₁	y ₁
x ₂	y ₁
x ₂	y ₂
x ₂	y ₃
x ₄	y ₂



在X与Y之间
不存在函数依赖

函数依赖的发现

□ 设有如左图所示的关系R

- 其中可能成立的函数依赖关系有哪些？
- 又有哪些函数依赖关系是不可能成立的？

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

说明：

- ① 根据具体的关系实例，可以否定一个函数依赖的存在，但不能证明某个函数依赖的成立；对于不能被否认的函数依赖，我们用了‘可能成立’这个定语，在本例中，寻找到的所有函数依赖都属于这一类；
- ② 只考虑右边的依赖因素是单个属性的情况；
- ③ 左边的决定因素按照属性数从少到多的顺序依次检查。

关系 R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

- 首先考虑决定因素和依赖因素都是单个属性的情况:

$A \rightarrow B \quad \checkmark$

$B \rightarrow A \quad \times$

$C \rightarrow A \quad \times$

$D \rightarrow A \quad \checkmark$

$A \rightarrow C \quad \times$

$B \rightarrow C \quad \times$

$C \rightarrow B \quad \checkmark$

$D \rightarrow B \quad \checkmark$

$A \rightarrow D \quad \times$

$B \rightarrow D \quad \times$

$C \rightarrow D \quad \times$

$D \rightarrow C \quad \checkmark$

- 根据关系R中的元组在属性B上的取值特殊性(都相同)可知:

① 以B作为决定因素的三个函数依赖都不成立 (很容易找到不满足函数依赖约束的一对例子元组(反例))

② 以B作为依赖因素的函数依赖都成立(找不到反例)

- 另外:

③ 根据属性D的取值可知, 所有以D为决定因素的FD都成立!

④ 对另外四个函数依赖, 可以根据函数依赖的定义来检查其是否成立

❑ 首先考虑决定因素和依赖因素都是单个属性的情况:

$A \rightarrow B \checkmark$ $B \rightarrow A \times$ $C \rightarrow A \times$ $D \rightarrow A \checkmark$
 $A \rightarrow C \times$ $B \rightarrow C \times$ $C \rightarrow B \checkmark$ $D \rightarrow B \checkmark$
 $A \rightarrow D \times$ $B \rightarrow D \times$ $C \rightarrow D \times$ $D \rightarrow C \checkmark$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

关系 R

❑ 做出判定的理由或方法如下:

- ① 以判断 $B \rightarrow A$ 为例: 很容易就能找到一对元组 (a_1, b_1, c_2, d_2) 和 (a_2, b_1, c_1, d_3) , 它们两者在属性 B 上的取值相等, 但在属性 A 上的取值不等, 这不符合 $B \rightarrow A$ 的定义, 故 $B \rightarrow A$ 不成立!
- ② 以判断 $X \rightarrow B$ 为例: 在这个关系中, 我们找不到一对元组且它们在依赖因素 B 上的取值不相等, 所以在该关系中“不存在不满足 $X \rightarrow B$ 的情况”, 故任何以 B 作为依赖因素的函数依赖都成立!

□ 首先考虑决定因素和依赖因素都是单个属性的情况:

$A \rightarrow B \checkmark$ $B \rightarrow A \times$ $C \rightarrow A \times$ $D \rightarrow A \checkmark$

$A \rightarrow C \times$ $B \rightarrow C \times$ $C \rightarrow B \checkmark$ $D \rightarrow B \checkmark$

$A \rightarrow D \times$ $B \rightarrow D \times$ $C \rightarrow D \times$ $D \rightarrow C \checkmark$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

关系 R

(cont.)

- ③ “以D为决定因素的函数依赖 $D \rightarrow X$ 一定成立”，其原因是“在这个关系中，找不到 $D \rightarrow X$ 的反例”，即：找不到一对元组，它们在决定因素D上的取值相等但在X上的取值不等！（函数依赖仅仅强调：当决定因素的取值相等时，依赖因素上的取值一定相等；在决定因素的取值不相等的情况下，依赖因素的取值可以相等也可以不相等！）
- ④ 以判断 $C \rightarrow A$ 为例：可以找到一对元组 (a_1, b_1, c_1, d_1) 和 (a_2, b_1, c_1, d_3) ，它们不符合 $C \rightarrow A$ 的定义。

关系 R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

$A \rightarrow B$

$C \rightarrow B$

$D \rightarrow A$

$D \rightarrow B$

$D \rightarrow C$

□ 其次，再考虑决定因素是多个属性的情况。其中：

1) 在左边的决定因素中不需要考虑含有属性 D 的情况，why？

2) 在左边的决定因素中不需要考虑含有属性 B 的情况，why？

□ 因此只需要考虑下述的FD是否成立：

$AC \rightarrow B$

$AC \rightarrow D$

关系 R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

- ❑ 已发现的函数依赖：{ $A \rightarrow B$, $C \rightarrow B$, $D \rightarrow B$, $D \rightarrow A$, $D \rightarrow C$ }
- ❑ 需检查 $AC \rightarrow B$ 和 $AC \rightarrow D$ 是否成立？
- ❑ 根据关系R中元组值的特殊性以及已经找到的函数依赖，我们不需要再检查 $AC \rightarrow B$ 是否成立，why？

- 任何一个以属性B作为依赖因素的函数依赖 $X \rightarrow B$ 都成立(在关系R中找不到其反例)
- 这样的函数依赖可以从已经发现的函数依赖中推导出来，所以也不用把 $AC \rightarrow B$ 写出来。

- ❑ 因此，最后只需要检查 $AC \rightarrow D$ 是否可能成立？(next slide)

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4



A	C	D
a_1	c_1	d_1
a_1	c_2	d_2
a_2	c_1	d_3
a_2	c_3	d_4



AC	D
(a_1, c_1)	d_1
(a_1, c_2)	d_2
(a_2, c_1)	d_3
(a_2, c_3)	d_4

∴ 不存在两个元组，它们在**AC**上的取值相同，而在**D**上的取值不同

∴ **AC**→**D**

关系 R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

❑ 已发现的函数依赖：

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

❑ 思考：为什么没有写出

- 1) 左边含有属性D的其它的那些可能的函数依赖？
- 2) 右边为单个属性B的其它的那些可能的函数依赖？
- 3) 右边为多个属性的那些可能的函数依赖？

情况1) & 2) 如果存在这样的函数依赖，它们都可以从已知的这六个函数依赖中推导得到；（事实上，在已知的这六个函数依赖中，就有函数依赖是可以被从其他函数依赖中推导出来的，如 $D \rightarrow B$ ）

情况3) 可以用已知的这六个函数依赖来证明它是否成立：

- 如果能够从已知的函数依赖中推导得到，就证明它是成立的；
- 如果无法从已知的函数依赖中推导得到，就证明它是不成立的。

关系 R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_1	c_1	d_3
a_2	b_1	c_3	d_4

X	Y	对应关系
A	B	n : 1
A	C	n : m
A	D	1 : n
B	C	1 : n
B	D	1 : n
C	D	1 : n
AC	D	1 : 1
AC	B	n : 1

- 设X和Y是关系R的两个属性子集，我们也可以通过它们在这个关系中所体现出来的“取值的数量对应关系”来分析它们之间的函数依赖情况（根据前面分析的结果，依赖因素Y只考虑是单个属性的情况；决定因素X只考虑是单个属性或者是AC组合）
- 根据右表可以得到如下函数依赖：
 $\{A \rightarrow B, D \rightarrow A, C \rightarrow B, D \rightarrow B, D \rightarrow C, AC \rightarrow D, D \rightarrow AC, AC \rightarrow B\}$
- 其中，有一些函数依赖是可以从其他函数依赖中推导得到，不一定需要写出来。
- 精简以后的函数依赖集如下： $\{A \rightarrow B, C \rightarrow B, D \rightarrow A, D \rightarrow C, AC \rightarrow D\}$

函数依赖的发现

- ❑ 函数依赖反映的是同一个关系中的两个属性子集之间在取值上的依存关系，这种依存关系实际上也是一种数据完整性约束。因此，我们也可以通过数据完整性约束的分析来寻找属性之间的函数依赖关系。
- ❑ 例：有一个学生课程关系 $R(sno, sn, sd, ss, cno, g)$ ，其中的属性有：学号 sno ，姓名 sn ，就读院系 sd 与就读专业 ss ，修读的某一门课程的课程号 cno 和成绩 g 。该关系模式中的语义约束如下：
 - sno 和 cno 分别是学生和课程的标识属性， sd 和 ss 分别是院系和专业的标识属性；
 - 每个学生只能有一个姓名，并且只就读于一个院系与一个专业；
 - 每个学生修读的每一门课程有且仅有一个成绩；
 - 各院系之间无相同专业（一个专业只能隶属于一个院系，但在一个院系里允许设若干个专业）。

关系 R (sno, sn, sd, ss, cno, g)

❑ sno是学生的标识属性，每个学生只能有一个姓名，并且只就读于一个系与一个专业；

➤ 有函数依赖： $sno \rightarrow sn$
 $sno \rightarrow sd$
 $sno \rightarrow ss$

❑ 每个学生修读的每一门课程有且仅有一个成绩；

➤ 有函数依赖： $(sno, cno) \rightarrow g$

❑ 各系之间无相同专业（一个专业只能隶属于一个系，但在一个系里允许设若干个专业）

• 有函数依赖： $ss \rightarrow sd$

❑ 合并上述分析结果，得到关系R上的函数依赖集为：

$\{ sno \rightarrow sn, sno \rightarrow sd, sno \rightarrow ss, (sno, cno) \rightarrow g, ss \rightarrow sd \}$

复习思考题 1

1. 请举例说明，不好的关系模式设计会带来哪些方面的问题？
2. 请给出以下函数依赖的定义：
 - ① 什么是函数依赖？
 - ② 什么是平凡函数依赖？什么是非平凡函数依赖？
 - ③ 什么是完全函数依赖？什么是部分函数依赖？
 - ④ 什么是传递函数依赖？
3. 码
 - ① 请用函数依赖给出关系的码(key)和超码(superkey)的定义；
 - ② 在关系模式中，什么是主属性？什么是非主属性？请设计一个例子关系，并写出该关系的模式、函数依赖集、关键字、主属性集、非主属性集；
 - ③ 请简要说明下列概念的相互关系：码，主码，主属性，非主属性。

复习思考题 2

4. 设有一个图书借阅关系R (借书证号, 身份证号, 书号, 书名, 借阅日期, 归还日期), 其中的数据约束如下:

- ① 借书证号、身份证号分别是借书证、读者的标识属性;
- ② 一个读者只能办理一张借书证, 一张借书证只能对应一个读者;
- ③ 每一本图书都有一个唯一的书号, 不同的图书可能有相同的书名;
- ④ 一个读者可以同时借阅多本图书, 也可以在不同时候借阅同一本图书;
- ⑤ 系统需要记录一本图书每一次被借阅的借阅日期和归还日期, 并保存所有的借阅历史。

请写出在借阅关系R上的非平凡的完全函数依赖。

复习思考题 3

5. 设有一个期末考试监考安排关系R，其中的属性有：课程的课程号(cno)和课程名(cname)，授课教师的工作证编号(tno)和姓名(tname)，监考老师的工作证编号(in_no)，每一场考试的开始时间(s_date)、结束时间(e_date)和考试教室(room)。其中：课程号和工作证编号分别是课程及教师的标识属性，开始时间和结束时间是date类型（含日期和时间）的字段，并且规定：

- ① 每一门课程至少有一位授课教师，也可能安排多位授课教师；
- ② 一位老师也可以担任多门课程的授课任务；
- ③ 每一门课的期末考试只安排一场，可分在多个教室中同时进行，除了授课教师外，在每一间考试教室中都必须安排一位或多位监考老师；
- ④ 同一时间段、同一间教室中只能安排一门课程的考试；
- ⑤ 一位老师可以担任多门课程的监考任务，但在同一时间段内，一位老师只能在指定的一间教室中监考一门课；
- ⑥ 授课教师必须参加自己承担授课任务的课程监考（不限定教室）。

R(cno, cname, tno, tname, in_no, s_date, e_date, room)

请找出该关系中的所有函数依赖（非平凡的完全函数依赖）。