

西北工业大学

Northwestern Polytechnical University

数据库系统原理

Database System

第六章 关系数据理论

赵晓南

2024.10

■ 数据库应用系统能力（基础级）

- ① SQL语句运用能力 √
- ② 数据库设计能力 ×
- ③ 数据库应用系统设计能力 ×
- ④ 数据库应用系统开发能力 √
- ⑤ 数据库运维能力



■ 数据库管理系统能力（提高级）

- ① 数据库管理系统设计能力
- ② 数据库管理系统开发能力

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.保持函数依赖的模式分解

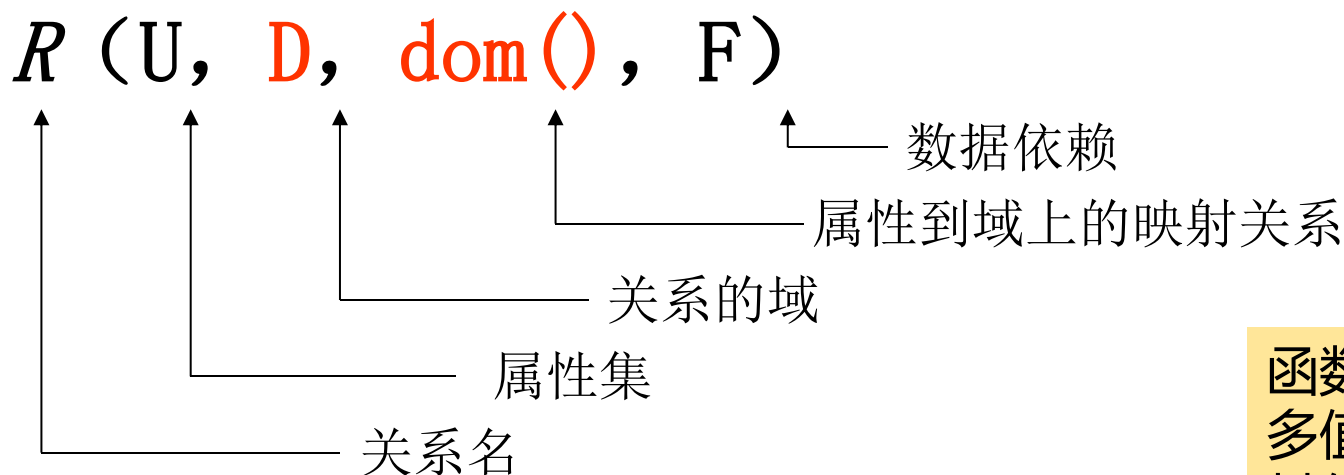
*6.5.无损连接的模式分解



■ 关系的型与值

- 关系的型：关系的结构(字段名、字段个数、域等)
- 关系的值：关系中具体的元组，也称关系的实例。

■ 关系模式(Relation Schema)：关系的型的定义



函数依赖
多值依赖
其他依赖

关系模式通常简记为: $R(U)$ 本章: $R(U, F)$

■ 关系模式的好与不好？

例：描述学校的数据库有如下属性：

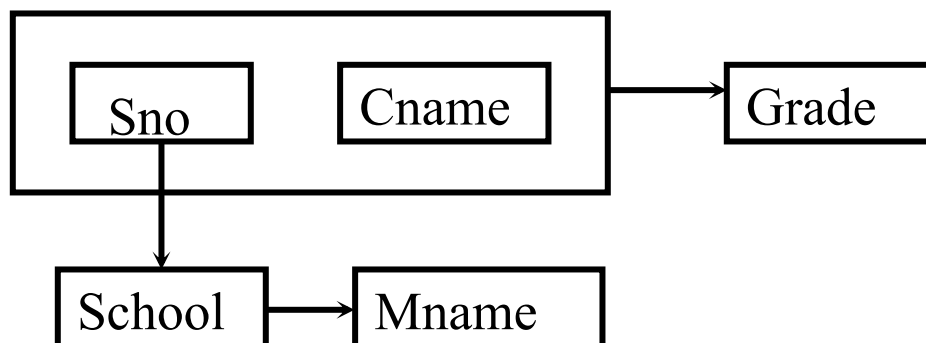
学生的学号 (Sno)、所在学院 (School)、院长姓名 (Mname)、
选课课程号 (Cno)、成绩 (Grade)

关系模式：Student (Sno, School, Mname, Cno, Grade)

根据现实语义可以得出：

函数依赖：

$F = \{Sno \rightarrow School, School \rightarrow Mname, (Sno, Cno) \rightarrow Grade\}$



■ Student (Sno, School, Mname, Cno, Grade)

1. 数据冗余太大:

浪费大量的存储空间

例：每一个学院院长的姓名重复出现

2. 修改复杂

数据冗余，更新数据时，维护数据完整性代价大。

例：某学院更换院长后，系统必须修改与该系学生有关的每个元组

3. 插入异常

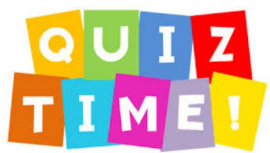
该插的数据无法插入到表中。

例：如果一个系刚成立，尚无学生，我们就无法把这个学院及其院长的信息存入数据库。

4. 删除异常

不该删除的数据被删掉了

例：如果某个学院的学生全部毕业了，我们在删除该系学生信息的同时，把这个学院及其院长的信息也丢掉了



关系模式： Student (Sno, School, Mname, Cno, Grade)。以下哪种设计最好？

- ☐ A S(Sno, Cno, Grade), D(School,Mname)
- ☐ B S(Sno, Cno, Grade), SD(Sno,School,Mname)
- ☐ C S(Sno, Cno, Grade), SD(Sno,School), SM(Sno,Mname)
- ☒ D S(Sno, Cno, Grade), SD(Sno,School), DM(School,Mname)

提交

6.1 问题的提出 - 好的关系模式?



■ Student (Sno, School, Mname, Cno, Grade)

结论: Student关系模式不是一个好的模式。

所谓“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: 由存在于模式中属性间的某些依赖关系引起的如Mname的取值依赖于Sno的值。

解决方法: 通过分解关系模式来消除其中不合适的依赖。

如：分解为三个关系模式

SD (Sno, School, Sno \rightarrow School)

S (Sno, Cno, Grade, (Sno, Cno) \rightarrow Grade)

DM (School, Mname, School \rightarrow Mname)

关系数据库理论

- 什么样的模式是好的关系模式？
- 模式应该怎样分解才是合理的？



度（即现代的尺）



量
(升、斗即现代的体积量具)



衡
(秤砣即重量的衡具)

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.保持函数依赖的模式分解

***6.5.无损连接的模式分解**



■ 规范化(Normalization)

指定义一组关系模式应该符合的条件(范式), 而符合这些条件的关系模式就不存在某些操作异常, 冗余也会减小。

- **数据依赖**: 定义属性值间的相互约束关系

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

■ 函数依赖 (Functional Dependencies, 简写为FD)

定义： 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 $X \rightarrow Y$ （读作 X 决定 Y 或者 Y 函数依赖于 X ）。

- X 称为这个函数依赖的**决定因素** (Determinant)。
- $X \rightarrow Y$ ，且 $Y \subseteq X$ ，称 $X \rightarrow Y$ 是平凡(trivial)的函数依赖。
- 若 Y 不函数依赖于 X ，则记为 $X \nrightarrow Y$ 。

例： 学生关系Student (Sno, Sname, Sgender, Sbirthdate, Smajor) 假设不允许

重名，则有：

$Sno \rightarrow Sgender, Sno \rightarrow Sbirthdate, Sno \rightarrow Smajor,$
 $Sno \rightarrow Sname, Sname \rightarrow Sno, (Sname \leftrightarrow Sno)$
 $Sname \rightarrow Sgender, Sname \rightarrow Sbirthdate, Sname \rightarrow Smajor.$
 $Sno, Sgender \rightarrow Sgender$ (平凡的函数依赖)

6.2 规范化 —— 函数依赖



■如果下表数据即为问题领域，则存在函数依赖有哪些？

属性A	属性B	属性C
1	2	3
4	2	3
5	3	3

$A \rightarrow B$ ✓ $B \rightarrow A$ ×
 $A \rightarrow C$ ✓ $C \rightarrow A$ ×
 $B \rightarrow C$ ✓ $C \rightarrow B$ ×

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

最终成立的FD：
 $A \rightarrow C, D \rightarrow B$

■ 完全函数依赖与部分函数依赖

定义：在关系模式 $R(U)$ 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 完全函数依赖于 X ，记作 $X \xrightarrow{F} Y$ 。
若 $X \rightarrow Y$ ，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 $SC(Sno, Sname, Cno, Grade)$ 中，
由于： $Sno \nrightarrow Grade, Cno \nrightarrow Grade$ ，
因此： $(Sno, Cno) \xrightarrow{F} Grade$
 $(Sno, Sname, Cno) \xrightarrow{P} Grade$

■ 传递函数依赖

定义：在关系模式 $R(U)$ 中，如果 $X \rightarrow Y$ ， $(Y \subseteq X)$ ， $Y \not\rightarrow X$ ， $Y \rightarrow Z$ ，则有 $X \rightarrow Z$ ，称 Z 传递函数依赖于 X ，记为： $X \xrightarrow{T} Z$ 。

注：①如果 $Y \subseteq X$ ，则 $X \xrightarrow{P} Z$ ；

②如果 $Y \rightarrow X$ ，即 $X \leftrightarrow Y$ ，则 Z 直接依赖于 X ；

例：在关系 $Std(Sno, School, Mname)$ 中，有：

$Sno \rightarrow School$ ， $School \rightarrow Mname$ ，则 $Sno \xrightarrow{T} Mname$

例：在关系 $SC(Sno, Sname, Cno, Grade, Sbirthdate)$ 中，

1) 假设： $X: Sno, Cno$ ， $Y: Sno$ ， $Z: Sbirthdate$

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

2) 假设： $X: Sno$ ， $Y: Sname$ ， $Z: Sbirthdate$

$Sno \rightarrow Sname$ ， $Sname \rightarrow Sbirthdate$ ，

$Sname \rightarrow Sno$ ， $(Sname \leftrightarrow Sno)$ $Sno \rightarrow Sbirthdate$ (直接依赖)

■ 码

定义： 设K为关系模式R<U, F>中的属性或属性组。若 $K \xrightarrow{P} U$ ，则K称为R的一个**超码** (Super Key)，如果K的任何一个真子集K'， $K' \not\xrightarrow{F} U$ ，则称K'是**候选码** (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)。

■ 候选码的两个性质：

① 决定性： $K \rightarrow U$

② 最小性： $\neg \exists K' \subset K$ ，使得 $K' \rightarrow U$

- **主属性 (Prime Attribute)：** 所有候选码中出现的属性
- **非主属性 (Nonprime Attribute)：** 不出现在任何候选码中的属性
- **全码 (All Key)：** 由关系模式的所有属性构成码

例： 设关系模式Student (Sno, Sname, Smajor, Sbirthdate)，若无重名，则Sno, Sname是候选码，Sno, Sname是主属性，Smajor, Sbirthdate是非主属性。

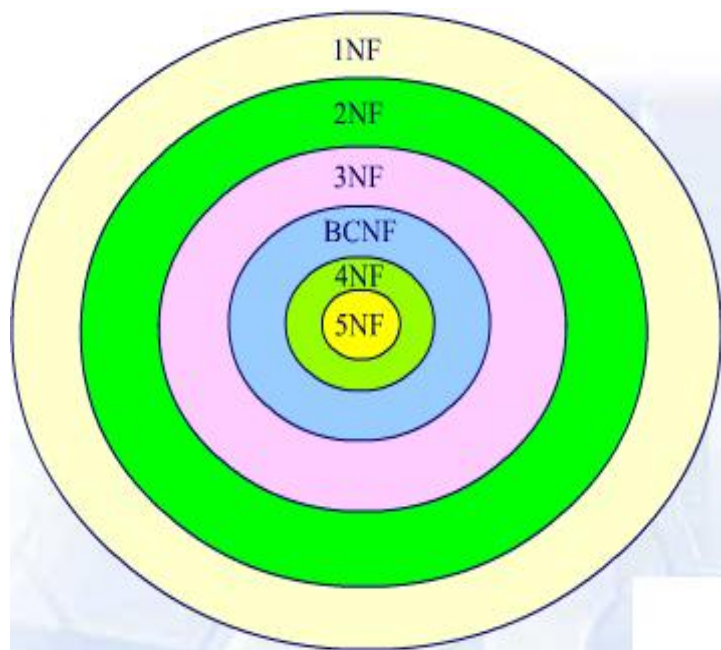
■ 码

外码：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。

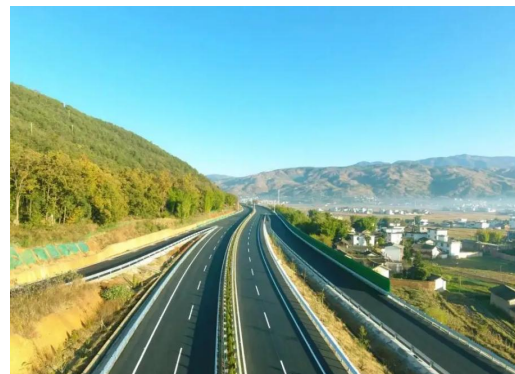
例：在关系 $SC(Sno, Cno, Grade)$ 中，码为 (Sno, Cno) ， Sno 是另一关系 $S(Sno, Sname, \dots)$ 的码，而非 SC 的码，故 Sno 是 SC 的外码，同样 Cno 也是 SC 的外码。

■ 范式

范式(Normal Form): 符合某一类满足一定要求的关系模式的集合。
关系数据库中的关系必须满足一定的要求。



通过模式分解转换为若干个高一级的范式的关系模式的集合，这个过程称为规范化。



6.2 规范化 —— 1NF



■ 1NF

1NF: 关系的每一个分量必须是一个不可分的数据项

例：职工号，姓名，电话号码组成一个表
(一个人可能有一个办公室电话 和一个家里电话号码)

规范成为1NF的方法？

- 1) 关键字只能是电话号码，重复存储职工号和姓名
- 2) 职工号为关键字，电话号码分为单位电话和住宅电话两个属性。
- 3) 职工号为关键字，但强制每条记录只能有一个电话号码。

职工号	姓名	电话号码	
		办公室	家

表中表，
不是1NF

■ 1NF的不足

- (1) 数据冗余。例如，职工号和姓名。
- (2) 插入异常。由于关系的主键电话不能为空值，如果一个职工不安装电话课，则这位职工的姓名等信息就不能插入表中。
- (3) 删除异常。如果所有电话拆除，则有关该电话的其它数据(职工号及所在姓名)也将被删除。
- (4) 修改复杂。如果改变职工编号，则需要修改表中的多行记录，如果部分修改，则会导致数据的不一致。

电话	职工号	姓名
8487777	0111	张三
8487776	0111	张三
。 。 。	。 。 。	。 。 。

6.2 规范化 —— 2NF



■ 2NF

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

例：SC(Sno, Cno, Grade)

① 每个属性都是由不可再分的值构成，故 $SC \in 1NF$

② 候选码：(Sno, Cno) 非主属性：Grade

$(Sno, Cno) \xrightarrow{F} Grade$ ， $SC \in 2NF$ 。

例：SLC(Sno, School, Sloc, Cno, Grade)

① 每个属性都是由不可再分的值构成，故 $SLC \in 1NF$

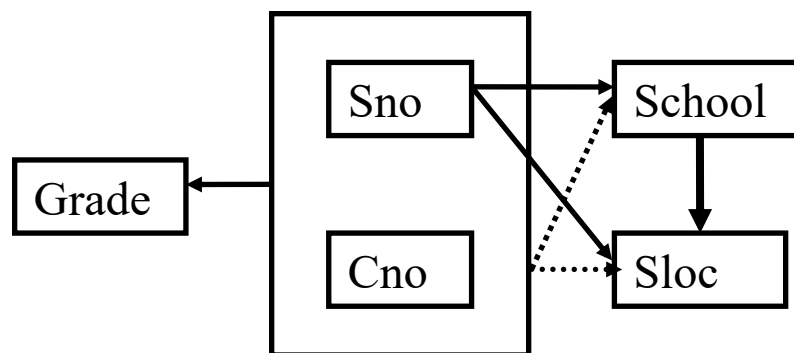
② 候选码：(Sno, Cno)

非主属性：School, Sloc, Grade

$Sno \rightarrow School$ ，故 $(Sno, Cno) \xrightarrow{P} School$ ， $SLC \notin 2NF$ 。

■ 2NF的不足

S-L-C



原因： School、Sloc部分函数依赖于码。

解决方法： 将S-L-C分解为两个关系模式，以消除其中的部分函数依赖。

SC (Sno, Cno, Grade)

S-L (Sno, School, Sloc)

∈ 2NF

对于S-L关系：

(1) **插入异常。** 例如插入一条学生记录 sno, school, sloc。若某个学院暂时没有确定办公地点，学生记录也无法插入。

(2) **删除异常。** 将某个学院学生全部毕业，会将学院的办公地址删除掉。

(3) **修改复杂。** 某个学院的办公地址发生变化，每一条记录都需要更新。

6.2 规范化 —— 3NF



■ 3NF

关系模式 $R\langle U, F \rangle$ 中若不存在这样的码 X 、属性组 Y 及非主属性 Z ($Z \not\subseteq Y$), 使得 $X \rightarrow Y$, $Y \rightarrow Z$ 成立, ($Y \not\rightarrow X$), 则称 $R \in 3NF$ 。

例: $SC(Sno, Cno, Grade) \in 2NF$

码: (Sno, Cno) , 非主属性: $Grade$

非主属性: $Grade$ 直接依赖于码

$(Sno, Cno) \rightarrow Grade$ 而非传递依赖于码, 故 $SC \in 3NF$

例: $SL(Sno, School, Sloc) \in 2NF$

码: Sno , 非主属性: $School, Sloc$

FD: $Sno \rightarrow School, School \rightarrow Sloc$ 。

则有 $Sno \rightarrow Sloc$, 即存在非主属性对码的传递依赖, 故 SL 不属于 $3NF$

6.2 规范化 —— 3NF

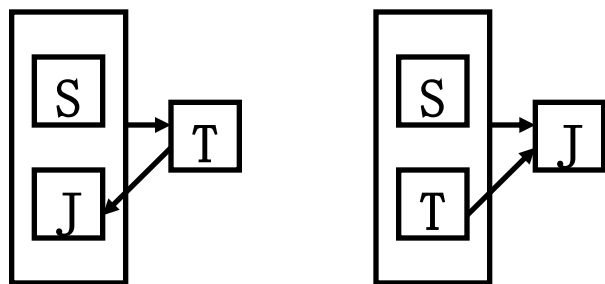


■ 3NF

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。

由语义得如下FD：

$(S, J) \rightarrow T$, $(S, T) \rightarrow J$, $T \rightarrow J$



STJ

码：(S, J), (S, T)

非主属性：无

①不存在非主属性对码的部分依赖 STJ
 $\in 2NF$

②不存在非主属性对码的传递依赖 STJ
 $\in 3NF$

■ 3NF的不足

- (1) **数据冗余**。同一老师的任课信息在多个学生选课的记录中重复存储。
- (2) **插入异常**。学生未开始选修某门课程前，无法输入教师任课信息。
- (3) **删除异常**。删除学生选课信息，教师任课信息一并删除。
- (4) **修改复杂**。由冗余性决定。

学生	老师	课程
李四	张三	数据结构
...

问题：2NF和3NF规范了非主属性和主属性之间的函数依赖关系，但是主属性之间如果存在部分函数依赖或者传递函数依赖呢？

6.2 规范化 —— BCNF



■BCNF:修正(或扩充)的第三范式。

定义：设关系模式 $R\langle U, F \rangle \in 1NF$ ，如果对于R的
每个函数依赖 $X \rightarrow Y$ ($Y \not\subseteq X$)， X 必包含码，则 $R \in BCNF$

例：关系模式 $C(Cno, Cname, Credit)$ ，课程允许重名。

码： Cno FD： $Cno \rightarrow Cname$, $Cno \rightarrow Credit$

非主属性： $Cname, Credit$

①不存在非主属性对码的部分依赖 $C \in 2NF$

②不存在非主属性对码的传递依赖 $C \in 3NF$

③每一个函数依赖的决定因素都包含码 $C \in BCNF$

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。由语义得如下FD：

$(S, J) \rightarrow T$, $(S, T) \rightarrow J$, $T \rightarrow J$ (前面已知 $STJ \in 3NF$)

存在函数依赖 $T \rightarrow J$ ，其中的决定因素 T 不包含码，因此 $STJ \notin BCNF$

■BCNF的不足

例如：一门课程有多个老师讲授，他们使用相同的一套参考书。

每个老师可以教授多门课程，每种参考书可以供多门课程使用。

课程C	教师T	参考书B
数据结构	张三	数据结构与C语言
数据结构	张三	数据结构与算法
数据结构	李四	数据结构与C语言
数据结构	李四	数据结构与算法
C语言	张三	数据结构与C语言
...

■BCNF的不足

课程C	教师T	参考书B
数据结构	张三	数据结构与C语言
数据结构	张三	数据结构与算法
数据结构	李四	数据结构与C语言
数据结构	李四	数据结构与算法
...

结论:

- 1) BCNF消除了任何属性对码的部分依赖和传递依赖，在函数依赖的范畴内解决了数据插入异常和删除异常
- 2) 但可能存在着数据冗余和修改复杂。

码: C, T, B

非主属性: 无

- ①不存在非主属性对码的部分依赖 $CTB \in 2NF$
- ②不存在非主属性对码的传递依赖 $CTB \in 3NF$
- ③每一个函数依赖的决定因素都包含码 $CTB \in BCNF$

■BCNF的特性

- 每个非主属性对每个码都是完全函数依赖；
- 所有的主属性对每一个**不包含它**的码，也是完全函数依赖；
- 没有任何属性完全函数依赖于非码的任何一组属性。

说明：

3NF和BCNF是在**函数依赖的条件**下对模式分解所能达到的**分离程度的限度**。

一个关系数据库中的所有关系模式如果都属于**BCNF**，

这在函数依赖范畴内，就已经实现了彻底的分离，

达到了**最高的规范化程度**，并消除了插入异常和删除异常。

3NF的不彻底性表现在可能存在主属性对码的部分函数依赖和传递依赖

■ 转换为第三范式的步骤

- (1) 把该结构分解成若干个属于第一范式的关系。
- (2) 对那些存在组合码, 且有非主属性部分函数依赖的关系必须继续分解, 使所得关系都是属于第二范式。
- (3) 若关系中有非主属性传递依赖于码, 则继续分解之, 使得关系都属于第三范式。
- (4) 主属性之间存在部分依赖或者传递依赖进行分解。

在分解时应注意满足以下三个条件:

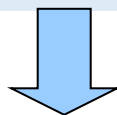
- (1) 分解是无损连接分解, 分解后既不丢失又不增加信息。
- (2) 分解所得的所有关系都是高一级范式的。
- (3) 分解所得关系的个数最少。

规范化方法: 投影分解, 其结果不是唯一的

■ 分解举例

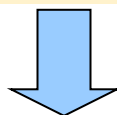
1NF:

SLC{(Sno, School, Sloc, Cno, Grade), ((Sno, Cno) → Grade,
Sno → Sschool, School → Sloc)}



2NF:

SC(Sno, Cno, Grade)	S-L(Sno, School, Sloc)
(Sno, Cno) → Grade	Sno → School, School → Sloc



3NF/BCNF:

SC{(Sno, Cno, Grade), ((Sno, Cno) → Grade)}
S-D{(Sno, School), (Sno → School)}
D-L{(School, Sloc), (School → Sloc)}

■ 多值依赖 (Multivalued Dependencies, 简称为MVD)

课程C (x)	教师T (y)	参考书B (z)
数据结构	张三	数据结构与C语言
数据结构	李四	数据结构与算法
数据结构	张三	数据结构与算法
数据结构	李四	数据结构与C语言
...

定义:

有关系模式 $R(U)$, 其中 X 、 Y 、 Z 是 U 的子集, 并且 $Z = U - X - Y$, 关系模式 $R(U)$ 中, 当且仅当满足下列性质: 对 $R(U)$ 的任一关系 r , 给定一对 (x, z) 值, 就有一组 y 值与之相对应, 而且这组 y 值只依赖于 x 值, 而与 z 值无关。则称 Y 多值依赖于 X , 记作 $X \twoheadrightarrow Y$ 。

例: 给定 (数据结构, 数据结构与C语言), 有一组 {张三, 李四} 与之对应, 并且 {张三, 李四} 这组值只课程有关, 与参考书无关

所以: 课程 \twoheadrightarrow 教师 同理: 课程 \twoheadrightarrow 参考书

一对多

6.2 规范化 —— 多值依赖



■ 多值依赖 (Multivalued Dependencies, 简称为MVD)

	X	Y	Z
	α	β	$R - \alpha - \beta$
t_1	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$a_{j+1} \dots a_n$
t_2	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$b_{j+1} \dots b_n$
t_3	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$b_{j+1} \dots b_n$
t_4	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$a_{j+1} \dots a_n$

$X \twoheadrightarrow Y$

另一种形式化定义:

在 $R(U)$ 的任一关系 r 中, 如果存在元组 t_1, t_2 , 使得 $t_1[X] = t_2[X]$, 那么必然存在元组 t_3, t_4 , 使得 $t_3[X]=t_4[X]=t_1[X]$, 而 $t_3[Y]=t_1[Y], t_3[Z]=t_2[Z]$, $t_4[Y]=t_2[Y], t_4[Z]=t_1[Z]$ 。

(即交换 t_1, t_2 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)。

6.2 规范化 —— 多值依赖



■ 多值依赖举例

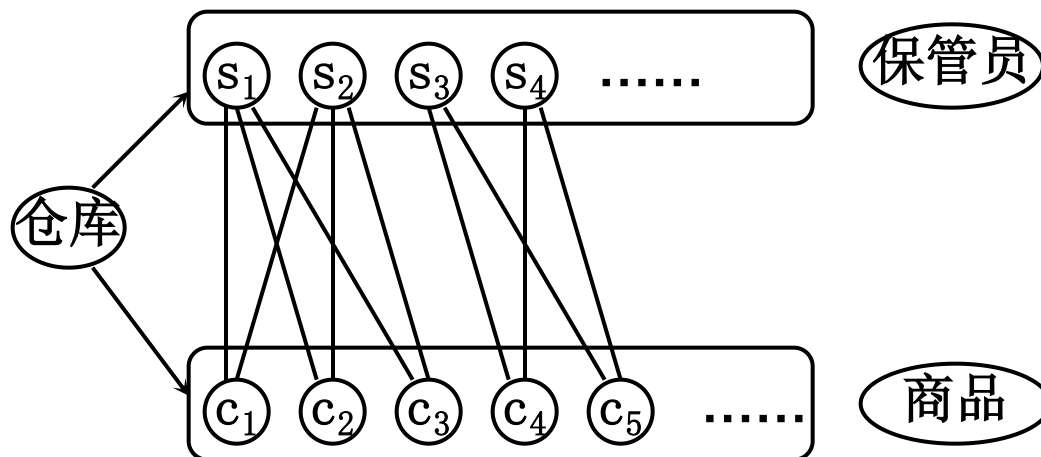
$W \twoheadrightarrow S$



$W \twoheadrightarrow C$

例：关系模式WSC(W, S, C)中，W表示仓库，S表示保管员，C表示商品。每个仓库有若干保管员，有若干商品，每个保管员保管所在仓库的所有商品，每种商品被所有它所在的仓库的保管员所保管。

W	S	C
w ₁	s ₁	c ₁
w ₁	s ₁	c ₂
w ₁	s ₁	c ₃
w ₁	s ₂	c ₁
w ₁	s ₂	c ₂
w ₁	s ₂	c ₃



Y与Z集合间构成完全二分图

■ 特殊的多值依赖：

若 $X \twoheadrightarrow Y$ ，而 $Z = \Phi$ ，则称 $X \twoheadrightarrow Y$ 为**平凡的多值依赖**（仅有 X, Y ）
否则称 $X \twoheadrightarrow Y$ 为**非平凡的多值依赖**。

■ 多值依赖的特性：

- 1) 对称性：若 $X \twoheadrightarrow Y$ ，则 $X \twoheadrightarrow Z$ ，其中 $Z = U - X - Y$ 。
- 2) 传递性：若 $X \twoheadrightarrow Y$ ， $Y \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Z - Y$ 。
- 3) 函数依赖是多值依赖的特例：若 $X \rightarrow Y$ ，则 $X \twoheadrightarrow Y$
- 4) 若 $X \twoheadrightarrow Y$ ， $X \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow YZ$
- 5) 若 $X \twoheadrightarrow Y$ ， $X \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Y \cap Z$
- 6) 若 $X \twoheadrightarrow Y$ ， $X \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Y - Z$ ， $X \twoheadrightarrow Z - Y$

■多值依赖与函数依赖的比较：

1) 有效性：多值依赖的有效性与属性集的范围有关

- 若 $X \twoheadrightarrow Y$ 在 U 上成立，则在 W ($X, Y \subseteq W \subseteq U$) 上一定成立；
反之则不然，即 $X \twoheadrightarrow Y$ 在 W ($W \subset U$) 上成立，在 U 上并不一定成立
- 函数依赖 $X \rightarrow Y$ 在 $R(U)$ 上成立，则在包含 XY 的任一超集中成立

2) 包含性

- 若函数依赖 $X \rightarrow Y$ 在 $R(U)$ 上成立，则对于任何 $Y' \subset Y$ 均有 $X \rightarrow Y'$ 成立。
- 多值依赖 $X \twoheadrightarrow Y$ 若在 $R(U)$ 上成立，不能断言对于任何 $Y' \subset Y$ ，有 $X \twoheadrightarrow Y'$ 成立。

	Loan-number	Customer-id	Customer-Street	Customer-City
1	L-23	99-123	North	Rye
2	L-23	99-123	south	Manchester
3	L-27	99-123	North	Rye
4	L-27	99-123	south	Manchester

$Customer_id \twoheadrightarrow (Customer_street, Customer_city)$

6.2 规范化 —— 4NF



函数依赖和多值依赖集为 D 的关系模式 $r(R)$ 属于第四范式(4NF)的条件是, 对 D^+ 中所有形如 $\alpha \twoheadrightarrow \beta$ 的多值依赖(其中 $\alpha \subseteq R$ 且 $\beta \subseteq R$), 至少有以下之一成立:

- $\alpha \twoheadrightarrow \beta$ 是一个平凡的多值依赖。
- α 是 R 的一个超码。

4NF

4NF: 关系模式 $R \langle U, F \rangle \in 1NF$, 如果对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y$ ($Y \subsetneq X$), X 都含有候选码, 则 $R \in 4NF$ 。

例: 前面的WSC关系模式 (仓库, 保管员, 商品)

码: 全码 (W, S, C)

多值依赖: $W \twoheadrightarrow S, W \twoheadrightarrow C$

根据定义可知, $R \notin 4NF$



例: 前面的WSC关系模式 (仓库, 保管员, 商品)

分解为:

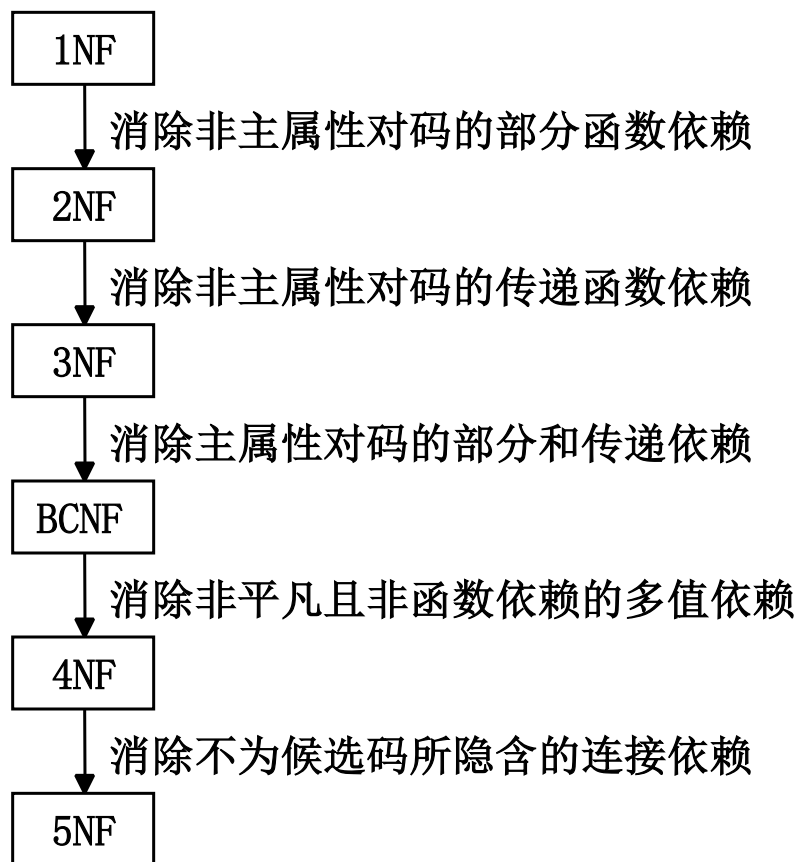
WS (W, S) (一对多)

WC (W, C) (一对多)

注意: 虽然仍然有 $W \twoheadrightarrow S$, 但这是平凡的多值依赖, 所以可以得到 $WS \in 4NF$, 同理 $WC \in 4NF$

4NF: 如果有多值依赖, 则一定依赖于候选码!

■ 规范化小结



规范化的基本原则：

由低到高, 逐步规范, 权衡利弊, 适可而止。
通常, 以满足第三范式为基本要求。

数据库设计满足的范式越高, 其数据处理的开销也越大

1. 判断：所有的BCNF都是3NF？
2. 判断：所有的BCNF都是4NF？
3. 关系R包含属性{A,B,C,D}，若已知该关系中存在多值依赖 $AB \twoheadrightarrow C$ ，根据表以下表中的数据可知该表中一定还存在哪两个元组？

A	B	C	D
3	2	1	4
3	2	2	3

答案： (3, 2, 1, 3) 、 (3, 2, 2, 4)

例题：请分析下列属性集上的函数依赖。

- 学生（学号，姓名，班级，课号，课程名，成绩，
教师，教师职务）

✓ 学号 \rightarrow {姓名，班级}；课号 \rightarrow 课程名；{学号，课号} \rightarrow 成绩
✓ 教师 \rightarrow 教师职务
✓ {班级，课号} \rightarrow 教师

{班级，课号} \rightarrow 教师；

课号 \rightarrow 教师；

{学号，课号} \rightarrow 教师

究竟选哪一个取决于对问题领域的理解

- 客户（客户号，客户名称，类别，联系电话，
产品编码，产品名称，数量，要货日期）

✓ 客户号 \rightarrow {客户名称，类别}
✓ 产品编码 \rightarrow 产品名称
✓ {客户号，产品编码，要货日期} \rightarrow 数量

函数依赖是对属性之间取值的一种约束

■ 练习（参考哈工大 数据库原理MOOC）

请自行分析下列属性集上的函数依赖

- 员工（员工码， 姓名， 出生日期， 联系电话， 最后学历，
毕业学校， 培训日期， 培训内容， 职务变动日期， 变动后职务）
- 图书（书号， 书名， 出版日期， 出版社， 书架号， 房间号）

图书：（书号， 书名， 出版日期， 出版社，
书架号， 楼层号， 楼层志愿者）

如何优化？



■ 练习

- $U = \{A, B, C, D, E\}$
- $F = \{AB \rightarrow C, B \rightarrow D, D \rightarrow E, C \rightarrow B\}$
 - 哪个是候选码?
 - 是第几范式?为什么?
 - 怎么改进?

- 函数依赖，码，范式等基本概念的理解
- 1NF到4NF的概念，理解与应用：

各个级别范式中存在的问题（插入异常、删除异常、数据冗余）和解决方法。能够根据应用语义，完整地写出关系模式的数据依赖集合，并能根据数据依赖分析某一个关系模式属于第几范式。

