



数据库技术与应用

关系数据理论

6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

多值依赖

- [例9] 学校中某一门课程由多个教师讲授，他们使用相同的一套参考书。每个教员可以讲授多门课程，每种参考书可以供多门课程使用。

❖ 非规范化关系

课 程 C	教 员 T	参 考 书 B
物理	{ 李 勇 王 军 }	{ 普通物理学 光学原理 物理习题集 }
数学	{ 李 勇 张 平 }	{ 数学分析 微分方程 }
计算数学	{ 张 平 周 峰 }	{ 高等代数 数学分析 }
⋮	⋮	⋮

多值依赖（续）

❖ 用二维表表示Teaching

课程C	教员T	参考书B
物理	李勇	普通物理学
物理	李勇	光学原理
物理	李勇	物理习题集
物理	王军	普通物理学
物理	王军	光学原理
物理	王军	物理习题集
数学	李勇	数学分析
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	数学分析
数学	张平	微分方程
数学	张平	高等代数
...


多值依赖（续）

❖ Teaching具有唯一候选码(C, T, B), 即全码

❖ Teaching \in BCNF

Teaching模式中存在的问题

- (1) 数据冗余度大
- (2) 插入操作复杂
- (3) 删除操作复杂
- (4) 修改操作复杂



存在
多值依赖

多值依赖（续）

❖ 定义6.9

设 $R(U)$ 是一个属性集 U 上的一个关系模式， X 、 Y 和 Z 是 U 的子集，并且 $Z=U-X-Y$ 。关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立，当且仅当对 $R(U)$ 的任一关系 r ，给定的一对 (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

❖ 例 Teaching (C, T, B)

多值依赖（续）

❖ 课程C: **X**, 教员T: **Y**, 参考书B: **Z**

课程C	教员T	参考书B
物理	李勇	普通物理学
物理	李勇	光学原理
物理	李勇	物理习题集
物理	王军	普通物理学
物理	王军	光学原理
物理学	王军	物理习题集
数学	李勇	数学分析
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	数学分析
数学	张平	微分方程
数学	张平	高等代数
...

多值依赖（续）

❖ 平凡多值依赖和非平凡的多值依赖

- 若 $X \twoheadrightarrow Y$ ，而 $Z = \varphi$ ，则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖
- 否则称 $X \twoheadrightarrow Y$ 为非平凡的多值依赖

多值依赖的性质

(1) 多值依赖具有对称性

若 $X \twoheadrightarrow Y$ ，则 $X \twoheadrightarrow Z$ ，其中 $Z = U - X - Y$

(2) 多值依赖具有传递性

若 $X \twoheadrightarrow Y$ ， $Y \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Z - Y$

(3) 函数依赖是多值依赖的特殊情况。

若 $X \rightarrow Y$ ，则 $X \twoheadrightarrow Y$ 。

(4) 若 $X \twoheadrightarrow Y$ ， $X \twoheadrightarrow Z$ ，则 $X \twoheadrightarrow Y \cup Z$ 。

(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（续）

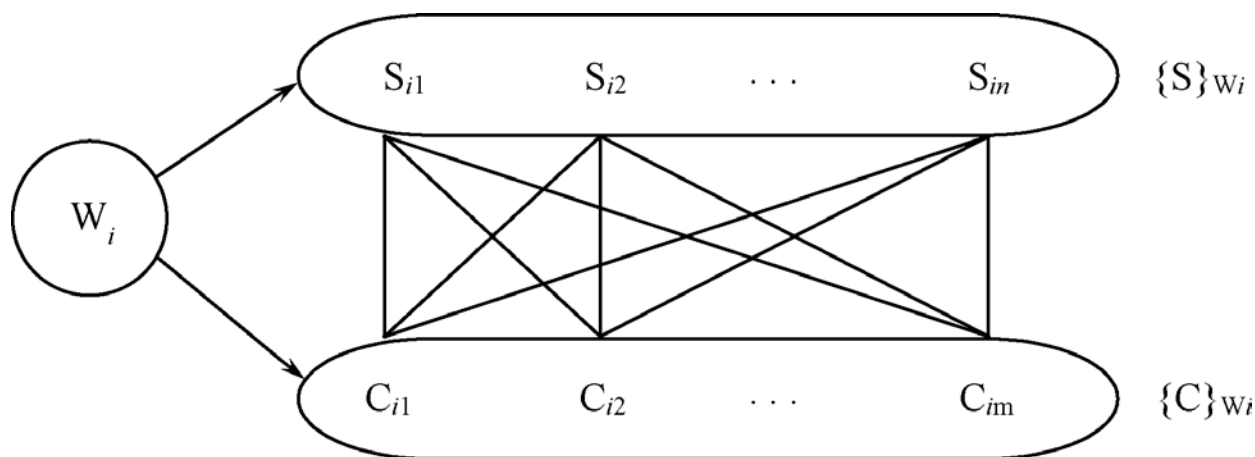
[例10] 关系模式WSC (W, S, C)

- W表示仓库，S表示保管员，C表示商品
- 假设每个仓库有若干个保管员，有若干种商品
- 每个保管员保管所在的仓库的所有商品
- 每种商品被所有保管员保管

W	S	C
W1	S1	C1
W1	S1	C2
W1	S1	C3
W1	S2	C1
W1	S2	C2
W1	S2	C3
W2	S3	C4
W2	S3	C5
W2	S4	C4
W2	S4	C5

多值依赖（续）

用下图表示这种对应



$$W \twoheadrightarrow S \text{ 且 } W \twoheadrightarrow C$$

6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.8 4NF

- ❖ **定义6.10** 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 $X \twoheadrightarrow Y$ ($Y \not\subseteq X$)， X 都含有码，则 $R \in 4NF$ 。
- ❖ 如果 $R \in 4NF$ ， 则 $R \in BCNF$
 - 不允许有非平凡且非函数依赖的多值依赖
 - 允许的非平凡多值依赖是函数依赖

4NF (续)

例: $\text{Teaching}(C, T, B) \notin 4NF$

存在非平凡的多值依赖 $C \twoheadrightarrow T$, 且 C 不是码

■ 用投影分解法把 Teaching 分解为如下两个关系模式:

$\text{CT}(C, T) \in 4NF$

$\text{CB}(C, B) \in 4NF$

$C \twoheadrightarrow T$, $C \twoheadrightarrow B$ 是平凡多值依赖

4NF (续)

例: BC(loan-number, customer-name, customer-street, customer-city)

假设一个客户可以有多个住址

BC \in BCNF

存在非平凡的多值依赖 $\text{customer-name} \twoheadrightarrow \text{loan-number}$, 且 customer-name 不是码

■ 用投影分解法把BC分解为如下两个关系模式:

BC1(loan-number, customer-name, T) \in 4NF

BC2(customer-name, customer-street, customer-city) \in 4NF

$C \twoheadrightarrow T$, $C \twoheadrightarrow B$ 是平凡多值依赖

6.2 规范化

6.2.1 函数依赖

6.2.2 码

6.2.3 范式

6.2.4 2NF

6.2.5 3NF

6.2.6 BCNF

6.2.7 多值依赖

6.2.8 4NF

6.2.9 规范化小结

6.2.9 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

规范化小结（续）

❖ 关系模式规范化的基本步骤

消除决定属性
集非码的非平
凡函数依赖

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

第六章 关系数据理论

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 模式的分解

6.5 小结

6.4 模式的分解

- ❖ 把低一级的关系模式分解为若干个高一级的关系模式的方法不唯一
- ❖ 只有保证分解后的关系模式与原关系模式**等价**，分解方法才有意义

三种模式分解等价的定义：

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

模式的分解（续）

定义6.16 关系模式 $R\langle U, F \rangle$ 的一个分解：

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

$$U = \bigcup_{i=1}^n U_i, \text{ 且不存在 } U_i \subseteq U_j, F_i \text{ 为 } F \text{ 在 } U_i \text{ 上的投影}$$

模式的分解（续）

例: SL (Sno, Sdept, Sloc)

$F = \{ Sno \rightarrow Sdept, Sdept \rightarrow Sloc, Sno \rightarrow Sloc \}$

$SL \in 2NF$

存在插入异常、删除异常、冗余度大和修改复杂等问题

分解方法可以有多种

模式的分解（续）

SL

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

模式的分解（续）

1. SL分解为下面三个关系模式：

SN(Sno), SD(Sdept), SO(Sloc)

SN

Sno

SD

Sdept

SO

Sloc

95001

CS

A

95002

IS

B

95003

MA

C

95004

PH

95005

模式的分解（续）

分解后的数据库**丢失了许多信息**，例如无法查询95001学生所在系或所在宿舍。

如果分解后的关系可以通过**自然连接恢复为原来的关系**，那么这种分解就**没有丢失信息**

模式的分解（续）

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：

NL

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

DL

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

模式的分解（续）

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

模式的分解（续）

NL \bowtie DL比原来的SL关系多了3个元组

无法知道95002、95004、95005

究竟是哪个系的学生

元组增加了，信息丢失了

第三种分解方法

3. 将SL分解为下面二个关系模式：

ND(Sno, Sdept), NL(Sno, Sloc)

ND

Sno	Sdept
-----	-------

95001	CS
95002	IS
95003	MA
95004	IS
95005	PH

NL

Sno	Sloc
-----	------

95001	A
95002	B
95003	C
95004	B
95005	B

模式的分解（续）

ND \bowtie NL

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

与SL关系一样，因此没有丢失信息

具有无损连接性的模式分解

- ❖ 关系模式 $R\langle U, F \rangle$ 的一个分解 $\rho = \{ R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle \}$

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

模式的分解（续）

ND(Sno, Sdept), NL(Sno, Sloc)

第三种分解方法具有无损连接性

问题: 插入异常、删除异常

这种分解方法没有保持原关系中的函数依赖

SL中的函数依赖 $Sdept \rightarrow Sloc$ 没有投影到关系模式ND、NL上

保持函数依赖的模式分解

设关系模式 $R\langle U, F \rangle$ 被分解为若干个关系模式

$$R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$$

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**）。

第四种分解方法

将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法就保持了函数依赖。

模式的分解（续）

- ❖ 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。

模式的分解（续）

- ❖ 1. SN(Sno), SD(Sdept), SO(Sloc)
 - ❖ 2. NL(Sno, Sloc), DL(Sdept, Sloc)
 - ❖ 3. ND(Sno, Sdept), NL(Sno, Sloc)
 - ❖ 4. ND(Sno, Sdept), DL(Sdept, Sloc)
-
- ❖ 第一种分解方法既不具有无损连接性，也未保持函数依赖，它不是原关系模式的一个等价分解
 - ❖ 第二种分解方法保持了函数依赖，但不具有无损连接性
 - ❖ 第三种分解方法具有无损连接性，但未持函数依赖
 - ❖ 第四种分解方法既具有无损连接性，又保持了函数依赖

数据依赖的公理系统

❖ 逻辑蕴含

定义6.11 对于满足一组函数依赖 F 的关系模式 $R \langle U, F \rangle$, 其任何一个关系 r , 若函数依赖 $X \rightarrow Y$ 都成立, (即 r 中任意两元组 t, s , 若 $t[X] = s[X]$, 则 $t[Y] = s[Y]$), 则称 F 逻辑蕴含 $X \rightarrow Y$

1. Armstrong公理系统

关系模式 $R \langle U, F \rangle$ 来说有以下的推理规则：

- A1. 自反律 (Reflexivity)：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所蕴含。
- A2. 增广律 (Augmentation)：若 $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所蕴含。
- A3. 传递律 (Transitivity)：若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 F 所蕴含，则 $X \rightarrow Z$ 为 F 所蕴含。

2. 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 $X \rightarrow Y$, $X \rightarrow Z$, 有 $X \rightarrow YZ$.
(A2, A3)
- **伪传递规则**: 由 $X \rightarrow Y$, $WY \rightarrow Z$, 有 $XW \rightarrow Z$.
(A2, A3)
- **分解规则**: 由 $X \rightarrow Y$ 及 $Z \subseteq Y$, 有 $X \rightarrow Z$.
(A1, A3)

3. 函数依赖闭包

定义6.12 在关系模式 $R\langle U, F \rangle$ 中为 F 所逻辑蕴含的函数依赖的全体叫作 F 的闭包，记为 F^+ 。

例: SL (Sno, Sdept, Sloc)

$F = \{ \text{Sno} \rightarrow \text{Sdept}, \text{Sdept} \rightarrow \text{Sloc} \}$

$F^+ = ?$

判定分解的无损连接性

- ❖ 关系模式 $R\langle U, F \rangle$, R_1 和 R_2 为 R 的分解。该分解为 R 的无损连接分解只要 F^+ 中至少有如下函数依赖中的一个：
- $R_1 \cap R_2 \rightarrow R_1$
 - $R_1 \cap R_2 \rightarrow R_2$

例：SL (Sno, Sdept, Sloc)

$F = \{ \text{Sno} \rightarrow \text{Sdept}, \text{Sdept} \rightarrow \text{Sloc} \}$

分解一：ND(**Sno**, Sdept), NL(**Sno**, Sloc)

分解二：ND(Sno, **Sdept**), DL(**Sdept**, Sloc)

无损连接分解生成BCNF的算法

```
result := {R};  
done := false;  
计算 $F^+$ ;  
while (not done) do  
    if (result中存在模式 $R_i$ 不属于BCNF)  
        令 $\alpha \rightarrow \beta$ 是 $R_i$ 上不满足BCNF的非平凡依赖, 且  
         $\alpha \cap \beta = \emptyset$ ;  
        result := (result -  $R_i$ )  $\cup$  ( $R_i$  -  $\beta$ )  $\cup$  ( $\alpha, \beta$ );  
    else done := true;
```

BCNF分解举例

- ❖ Lending-scheme = (branch-name, branch-city, asset, customer-name, loan-number, amount)
- ❖ 函数依赖集
 - branch-name \rightarrow assets, branch-city
 - loan-number \rightarrow amount, branch-name
- ❖ 候选码
 - { loan-number, customer-name }

BCNF分解举例

- ❖ Lending-scheme = (branch-name, branch-city, asset, customer-name, loan-number, amount)
- ❖ 函数依赖集
 - **branch-name** → **assets, branch-city**
 - loan-number → amount, branch-name
- ❖ 分解步骤一
 - **Branch-schema = (branch-name, branch-city, assets)**
 - Loan-info-schema = (branch-name, customer-name, loan-number, amount)

BCNF分解举例

- ❖ Lending-scheme = (branch-name, branch-city, asset, customer-name, loan-number, amount)
- ❖ 函数依赖集
 - branch-name → assets, branch-city
 - **loan-number → amount, branch-name**
- ❖ 分解步骤二
 - Branch-schema = (branch-name, branch-city, assets)
 - Loan-schema = (loan-number, amount, branch-name)
 - Borrower-schema = (customer-name, loan-number)

BCNF分解举例

- ❖ Lending-scheme = (branch-name, branch-city, asset, customer-name, loan-number, amount)
- ❖ 函数依赖集
 - branch-name → assets, branch-city
 - loan-number → amount, branch-name
- ❖ 检验结果是否为BCNF及保持函数依赖
 - Branch-schema = (branch-name, branch-city, assets)
 - Loan-schema = (loan-number, amount, branch-name)
 - Borrower-schema = (customer-name, loan-number)

BCNF分解不一定保持函数依赖

- ❖ Banker-schema = (branch-name, customer-name, banker-name)
- ❖ 一个客户在某一分行有一个“银行账户负责人”
- ❖ 函数依赖集
 - $\text{banker-name} \rightarrow \text{branch-name}$
 - $\text{branch-name, customer-name} \rightarrow \text{banker-name}$
- ❖ BCNF分解
 - Banker-branch = (banker-name, branch-name)
 - Customer-banker = (customer-name, banker-name)
- ❖ 分解后的模式集不保持函数依赖
- ❖ 原模式集是3NF

小结

- ❖ 若仅要求分解是无损连接，那么模式分解一定能够达到BCNF
- ❖ 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF
- ❖ BCNF vs 保持依赖的3NF
 - 通常倾向于选择后者

课堂练习

- ❖ Teaching(Course, Teacher, Time, Room, Student, Grade)
- ❖ 函数依赖
 - Course→Teacher
 - Time, Room → Course
 - Time, Teacher → Room
 - Time, Student → Room
 - Course, Student → Grade
- ❖ 1、是否是BCNF或3NF?
- ❖ 2、BCNF分解
- ❖ 3、分解后的模式集是否保持函数依赖?

练习

- ❖ Teaching(Course, Teacher, Time, Room, Student, Grade)
- ❖ (Course, Teacher)
- ❖ (Course, Student, Grade)
- ❖ (Time, Room, Course)
- ❖ (Time, Room, Student)
- ❖ 不保持函数依赖

作业

- ❖ 下下次课交：设计数据库模式（**ER图**、关系表）
 - 淘宝网
 - 微信