

数据库原理

第6章 关系数据库设计理论

辽东学院 鲁 琴

本节要点

数据库基础概念

数据库原理

关系数据库

数据库新技术

关系数据模型

关系数据语言

数据库设计

数据库管理

数据库设计步骤

概念设计工具

ER图

UML图

逻辑设计工具

关系数据库设计理论

数据依赖

1NF

2NF

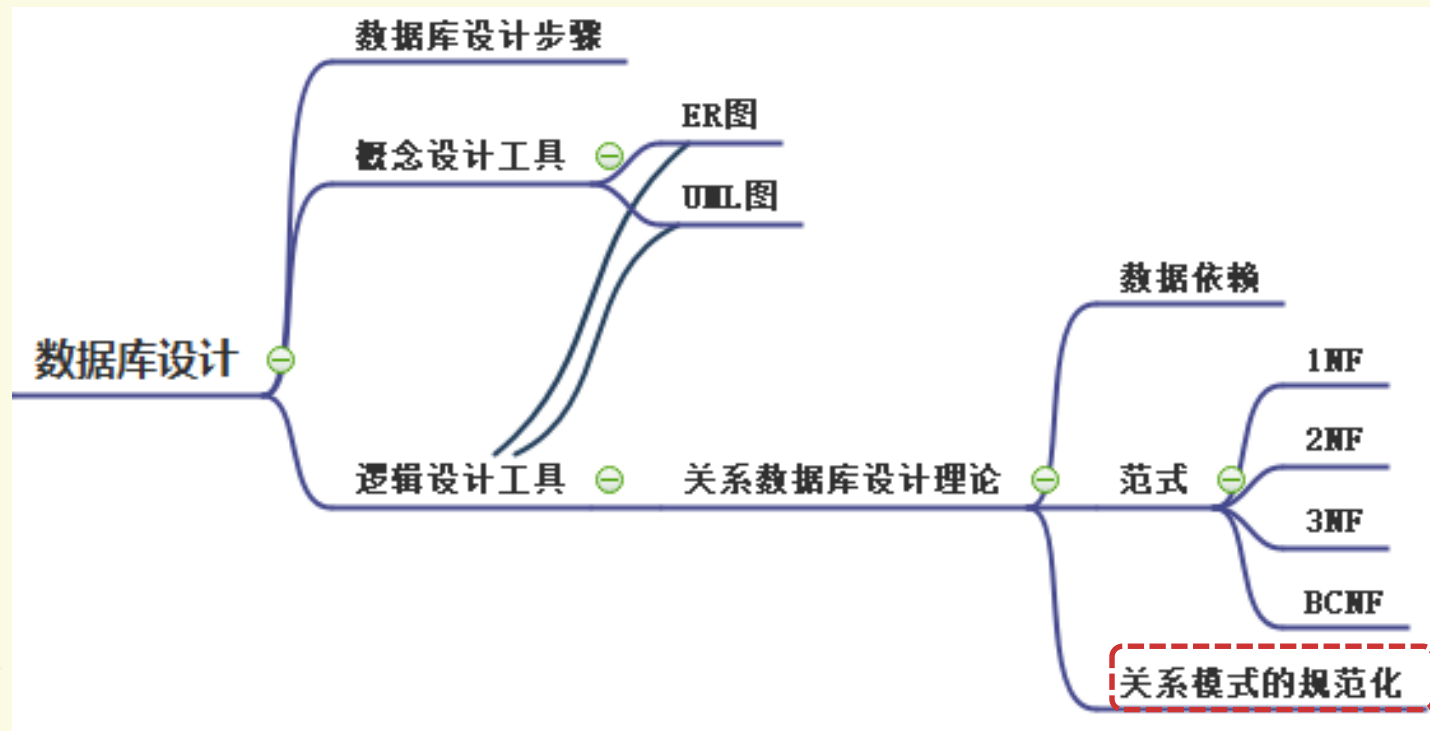
3NF

BCNF

关系模式的规范化

范式

第6章 关系数据库设计理论



第6章 关系数据库设计理论

6.1 数据依赖

6.2 范式

6.3 关系模式的规范化

6.3.1 关系模式规范化的步骤

6.3.2 关系模式的分解

6.3 关系模式的规范化

关系数据库的规范化理论是数据库**逻辑设计**的工具

一个关系只要其分量都是不可分的数据项，它就是规范化的关系，但这只是**最基本的规范化**

规范化程度可以有6个不同的级别，即6个范式

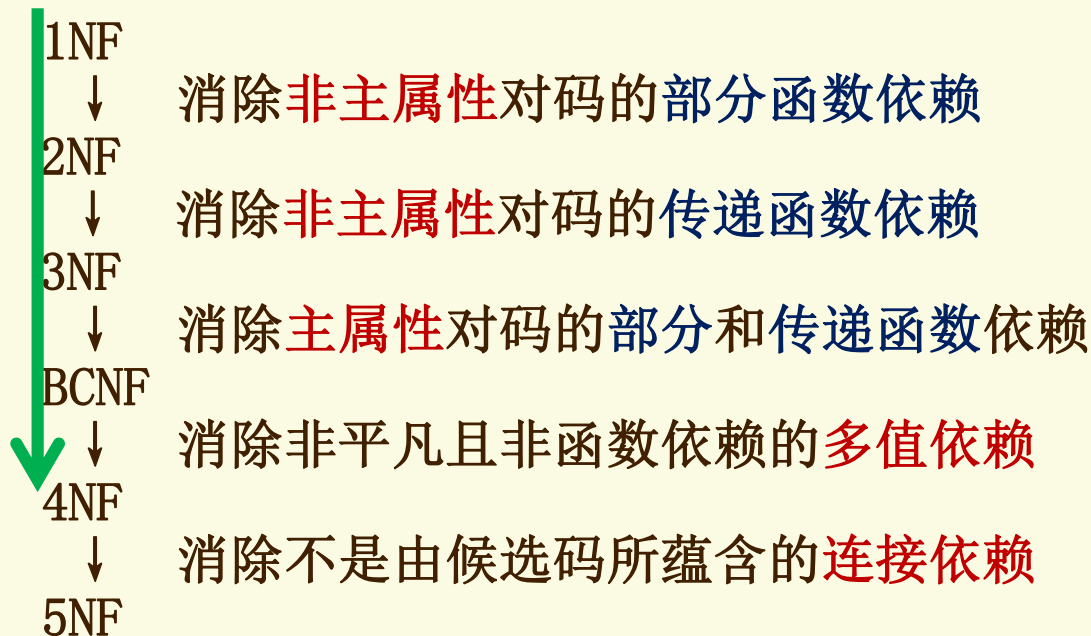
关系模式的规范化

规范化程度过低的关系不一定能够很好地描述现实世界，可能会存在插入异常、删除异常、修改复杂、数据冗余等问题，解决方法就是对其进行规范化，转换成高级范式。

一个低一级范式的关系模式，通过模式分解可以转换为若干个高一级范式的关系模式集合，这种过程就叫关系的规范化。

6.3.1 关系模式规范化的步骤

消除决定属性
集非码的非平
凡函数依赖



关系模式规范化的步骤

- 规范化的**基本思想**是逐步消除数据依赖中不合适的部分，使模式中的各关系模式达到某种程度的“**分离**”
- 即采用“**一事一地**”的模式设计原则，让一个关系描述一个概念、一个实体或者实体间的一种联系
- 若多于一个概念就把它“分离”出去。因此所谓**规范化实质上**是概念的单一化

关系模式规范化的步骤

- 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

6.3.2 关系模式的分解

关系模式的规范化过程是通过对关系模式的分解来实现的

- 把低一级的关系模式分解为若干个高级的关系模式的**方法并不唯一**
- 在这些分解方法中，只有能够保证分解后的关系模式与原关系模式**等价**的方法才有意义

关系模式的分解（续）

将一个关系模式 $R(U, F)$ 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

意味着相应地将存储在一个二维表 t 中的数据分散到若干个二维表 t_1, t_2, \dots, t_n 中去

- ✓ 其中 t_i 是 t 在属性集 U_i 上的投影

关系模式 SL (Sno, Sdept, Sloc)

函数依赖:

$Sno \rightarrow Sdept$

$Sdept \rightarrow Sloc$

$Sno \rightarrow Sloc$



SL属于
几范式?

$SL \in 2NF$, 该关系模式存在插入异常、删除异常、数据冗余度和修改复杂的问题

因此需要分解该关系模式, 使成为更高范式的关系模式
分解方法可以有很多种

关系模式SL的分解

假设下面是该关系模式的一个关系：

SL

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

第一种分解方法

分解后的关系为：

	SN	SD	SO
SN(Sno)			
SD(Sdept)			
SO(Sloc)			
	Sno	Sdept	Sloc
	95001	CS	A
	95002	IS	B
	95003	MA	C
	95004	PH	
	95005		

第一种分解方法

- SN、SD和SO都是规范化程度很高的关系模式（5NF）
- 但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。因此这种分解方法是不可取的
- 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。

第二种分解方法

此分解方法仍然丢失了信息

NL(Sno, Sloc)

DL(Sdept, Sloc)

NL ∞ DL

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

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

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

第三种分解方法

ND(Sno, Sdept)

NL(Sno, Sloc)

无损连接

丢失原
函数依赖

ND ∞ DL

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

具有无损连接性的模式分解

- 设关系模式 $R(U, F)$ 被分解为若干个关系模式 $R_1(U_1, F_1)$, $R_2(U_2, F_2)$, ..., $R_n(U_n, F_n)$
 - ✓ 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 R 与 R_1 、 R_2 、...、 R_n 自然连接的结果相等, 则称关系模式 R 的这个分解具有无损连接性 (Lossless join)
- 只有具有无损连接性的分解才能够保证不丢失信息
- 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

保持原关系中的函数依赖

上面的第三种分解方法虽然具有无损连接性，保证了不丢失原关系中的信息，但它并没有解决插入异常、删除异常、修改复杂、数据冗余等问题。

例如95001学生由CS系转到IS系，ND关系的(95001, CS)元组和NL关系的(95001, A)元组必须同时进行修改，否则会破坏数据库的一致性

之所以出现上述问题，是因为分解得到的两个关系模式不是互相独立的。SL中的函数依赖 $Sdept \rightarrow Sloc$ 既没有投影到关系模式ND上，也没有投影到关系模式NL上，而是跨在这两个关系模式上。也就是这种分解方法没有保持原关系中的函数依赖

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 $R_1(U_1, F_1)$, $R_2(U_2, F_2)$, ..., $R_n(U_n, F_n)$
 - ✓ 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)。

第四种分解方法

ND(Sno, Sdept)

DL(Sdept, Sloc)

无损连接

保持函数依赖

$ND \infty DL$

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

Sdept	Sloc
CS	A
IS	B
MA	C
IS	B
PH	B

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

判断对关系模式的一个分解是否与原关系模式等价的标准

1. 分解具有无损连接性
2. 分解要保持函数依赖

关系模式的分解

- 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

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

分解后的关系为：

SN

SD

SO

SN(Sno)

SD(Sdept)

SO(Sloc)

Sno
95001
95002
95003
95004
95005

Sdept
CS
IS
MA
PH

Sloc
A
B
C

第二种分解方法 保持了函数依赖，但不具有无损连接性

NL(Sno, Sloc)

DL(Sdept, Sloc)

NL ∞ DL

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

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B

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

第三种分解方法

具有无损连接性，但未持函数依赖

ND(Sno, Sdept)

NL(Sno, Sloc)

ND ∞ DL

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

第四种分解方法 既具有无损连接性，又保持了函数依赖

ND(Sno, Sdept)

DL(Sdept, Sloc)

ND ∞ DL

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

Sdept	Sloc
CS	A
IS	B
MA	C
IS	B
PH	B

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

关系模式的分解

规范化理论提供了一套完整的模式分解算法，按照这套算法可以做到

- 若要求分解具有无损连接性，那么模式分解一定能够达到4NF
- 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到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

小结

规范化理论为数据库**逻辑设计**提供了理论的指南和工具

- 也仅仅是指南和工具

并不是规范化程度越高，模式就越好

- 必须结合应用环境和现实世界的具体情况合理地选择数据库模式

练习

现有如下关系模式： $R(A\#, B\#, C, D, E)$

其中：

$A\#B\#$ 组合为码， R 上存在的函数依赖有 $A\#B\# \rightarrow E$ ， $B\# \rightarrow C$ ， $C \rightarrow D$

1. 该关系模式满足2NF吗？为什么？

2. 如果将关系模式 R 分解为：

$R_1(A\#, B\#, E)$

$R_2(B\#, C, D)$

指出关系模式 R_2 的码，并说明该关系模式最高满足第几范式？(在1NF~BCNF之内)。

$R(A\#, B\#, C, D, E)$

1. 不满足

因为 $A\#B\#$ 组合为码，所以 $A\#B\# \rightarrow C$ ，又 $B\# \rightarrow C$ ，即 C 部分函数依赖于 $A\#B\#$ ，所以不满足2NF。

2. R_2 的码： $B\#$ 。满足2NF。

因为 $B\# \rightarrow C$ ， $C \rightarrow D$ ，存在传递函数依赖，不满足3NF