

数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼



第六章 关系数据理论

Principles of Database Systems

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

6.1 问题的提出

- 所有数据库应用系统设计的共同问题—— 如何把复杂的现实世界表达成合适的数据库模式？
- 具体到关系数据库应用系统中，就表现为下列问题：
 - 数据库中应该构造几个关系？
 - 每个关系又应该包括哪些属性？
- 这些问题属于数据库逻辑设计的问题。
- 本章讲述关系数据库设计理论，它是数据库逻辑设计的有力工具。主要包括三个方面的内容：
 - 数据依赖
 - 范式
 - 模式设计方法

6.1 问题的提出

□ 一个完整的关系模式用五元组表示： $R(U, D, DOM, F)$ ， 其中：

R： 关系名称

U： 一组属性

D： U中属性所来自的域

DOM： 属性到域的映射

F： 属性组U上的一组数据依赖

□ 关系模式设计中起主要作用的是U和F， 简化的关系模式为3元组： $R<U, F>$

- 表示： **当且仅当**U上的一个关系r满足F时， r称为关系模式 $R<U, F>$ 上的一个关系。

- 有时候， 更简写成 $R(U)$ ， 即 $R(A_1, A_2, \dots, A_n)$ 的形式。

6.1 问题的提出

- 规范化理论主要包括三个方面的内容：
 - 数据依赖
 - 范式 (Normal Form)
 - 模式设计
- 其中,
 - **数据依赖**是一个关系内部属性与属性之间的一种约束关系。这种约束关系通过数据值的相等与否体现出数据间的相关关系。
 - **函数依赖** (Functional Dependancy, FD) 和
 - **多值依赖** (Multi-Valued Depandancy, MVD)
 - 函数依赖起着核心的作用, 是模式分解和模式设计的基础, 范式是模式分解的标准。

6.1 问题的提出

□ 关系模式的存储异常问题:

- 什么是好的关系模式?
- 某些不好的关系模式可能导致哪些问题?
- 数据库的逻辑设计为什么要遵循一定的规范化理论?

6.1 问题的提出

□ 函数依赖

—— 属性间的一种依赖关系。

设有关系SCD:

语义定义如下:

- 一个系有若干名学生, 但一个学生只属于一个系;
- 一个系只有一名系主任;
- 一个学生可以选修多门功课, 每门课程可有若干学生选修;
- 每个学生学习课程有一个成绩。

学号	姓名	年龄	所在系	系主任	课程号	成绩
S1	赵亦	18	计算机	刘伟	C1	90
S1	赵亦	18	计算机	刘伟	C2	82
S2	钱尔	19	信息	王平	C2	78
S2	钱尔	19	信息	王平	C4	98
S4	李思	21	自动化	陈浩	C6	NULL

思考: 关系模式 SCD 中存在哪些函数依赖?

学号→姓名, 学号→年龄, 学号→所在系,
所在系→系主任,
(学号, 课程号)→成绩

这个关系模式好不好?

6.1 问题的提出

□ 问题1：数据冗余

例如：每个系名和系主任的名字存储的次数等于该系的学生人数乘以每个学生选修的课程门数，同时学生的姓名、年龄也都要重复存储多次，数据的冗余度很大。

学号	姓名	年龄	所在系	系主任	课程号	成绩
S1	赵亦	18	计算机	刘伟	C1	90
S1	赵亦	18	计算机	刘伟	C2	82
S2	钱尔	19	信息	王平	C2	78
S2	钱尔	19	信息	王平	C4	98
S4	李思	21	自动化	陈浩	C6	NULL

□ 问题2：更新异常(update anomalies)

例如：如果某系更换系主任，则属于该系的所有学生记录都要进行修改。

6.1 问题的提出

学号	姓名	年龄	所在系	系主任	课程号	成绩
S1	赵亦	18	计算机	刘伟	C1	90
S1	赵亦	18	计算机	刘伟	C2	82
S2	钱尔	19	信息	王平	C2	78
S2	钱尔	19	信息	王平	C4	98
S4	李思	21	自动化	陈浩	C6	NULL

❑ 问题3：插入异常(insertion anomalies)

例如：此关系的码为(学号, 课程号)，依据实体完整性约束，缺少其中一部分时，元组不能插入。因此，如果某个学生没有选课，则这名学生的信息无法插入到数据库中。

❑ 问题4：删除异常(deletion anomalies)

例如：某系学生全部毕业时，要删除全部学生的记录，则系名、系主任也随之删除。虽然这个系依然存在，但在数据库中却无法找到该系的信息。

6.1 问题的提出

□ 问题产生原因：关系SCD中内容太杂；存在不合适的数据依赖。

□ 解决方法：关系分解：

学生关系：S(学号, 姓名, 年龄, 所在系)

选课关系：SC(学号, 课程号, 成绩)

系关系：D(系名, 系主任)

□ 与SCD相比，分解为3个关系模式后的优点：

- 数据的冗余度明显降低；
- 如果某系更换系主任，只需修改关系D中的一条记录，避免了更新异常；
- 如果某个学生尚未选课，只要在S中添加一条学生记录，与选课关系无关，避免了插入异常；
- 当一个系的学生全部毕业时，只需在S中删除该系的全部学生记录，而D中该系的信息仍然保留，避免了删除异常。

6.1 问题的提出

- 一个好的关系模式应该具备以下4个条件：
 - 尽可能少的数据冗余
 - 没有更新异常
 - 没有插入异常
 - 没有删除异常
- 产生关系数据库中这些异常问题的根本原因：
 - 关系模式中存在不合适的属性间联系
- 解决问题的方法：
 - 采用分解策略消去不合适联系

6.1 问题的提出

- 分解后的关系模式是“好”的。
- 不过，一个好的关系模式并不是在任何情况下都是最优的。
- **关系的规范化**：按照一定的规范设计关系模式，将结构复杂的关系**分解**成结构简单的关系，从而把“不好”的关系数据库模式转变为“好”的关系数据库模式。
- 规范化可以根据不同的要求而分成若干级别，即范式。
- 数据库模式的好坏和关系中各属性间的依赖关系有关，因此：
先讨论属性间的依赖关系，然后再讨论关系规范化理论。

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

□ 一个好的关系模式应该具备以下4个条件：

- ❖ 尽可能少的数据冗余
- ❖ 没有更新异常
- ❖ 没有插入异常
- ❖ 没有删除异常

6.2.1 函数依赖

□ **定义6.1** 设 $R(U)$ 是一个属性集 U 上的关系模式, $X, Y \subseteq U$, r 是 $R(U)$ 上的任意一个关系, 如果成立:

对 $\forall t, s \in r$, 若 $t[X] = s[X]$, 则 $t[Y] = s[Y]$

则称 “**X函数确定Y**” 或 “**Y函数依赖于X**”, 记作 $X \rightarrow Y$ 。

(或: 不存在 t, s , $t[X] = s[X]$ 而 $t[Y] \neq s[Y]$,)

X 称为这个函数依赖的**决定属性集(Determinant)**。

若 Y 不函数依赖于 X , 则记为 $X \nrightarrow Y$ 。

□ 类似于数学函数: $y = f(x)$, 对于任意的 x , 有唯一的 y 值与之对应

6.2.1 函数依赖

有关函数依赖的几点说明：

1) 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。

2) 函数依赖是**语义范畴**的概念

只能根据**语义**来确定一个函数依赖，而不能按照其形式化定义来证明一个函数依赖是否成立。

例：对于关系模式S，当学生不存在重名的情况下，可以得到：

姓名→**年龄** 或 **姓名**→**性别**。

函数依赖反映了一种**语义完整性约束**。

6.2.1 函数依赖

有关函数依赖的几点说明：

3) 函数依赖关系的存在与时间无关。

4) 属性间联系与函数依赖的对应关系

- 1:1联系：存在函数依赖 $X \rightarrow Y$ ，且 $Y \rightarrow X$ ，即 $X \leftrightarrow Y$ 。
- 1:m联系：存在函数依赖 $X \rightarrow Y$ ，但不存在 $Y \rightarrow X$ 。
- m:n联系：不存在函数依赖。

例：关系模式SCD中存在的函数依赖——属性间的联系：

- 学号 \rightarrow 姓名，学号 \rightarrow 年龄，学号 \rightarrow 所在系， 1:m
- 所在系 \rightarrow 系主任， 1:1
- (学号, 课程号) \rightarrow 成绩, 学号和课程之间 m:n

6.2.1 函数依赖

2. 平凡函数依赖与非平凡函数依赖

□ 平凡函数依赖

设有关系模式 $R(U, F)$, X 、 Y 都是 U 的子集, 若 $X \rightarrow Y$, $Y \subseteq X$, 则称 $X \rightarrow Y$ 是平凡函数依赖。

□ 非平凡函数依赖

设有关系模式 $R(U, F)$, X 、 Y 是 U 的子集, 若 $X \rightarrow Y$, 但 $Y \not\subseteq X$, 则称 $X \rightarrow Y$ 是非平凡函数依赖。

□ 说明: 对于任何 R , 平凡函数依赖总是成立的。

若不特别声明, 我们总是讨论非平凡的函数依赖。

6.2.1 函数依赖

3. 完全函数依赖与部分函数依赖

□ **定义6.2** 设关系模式 $R(U)$, U 是属性全集, X 和 Y 是 U 的子集:

- 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' , 都有 $X' \not\rightarrow Y$, 则称 Y **完全函数依赖于 X** , 记作 $X \twoheadrightarrow Y$ 。(Full functional dependency)
- 如果对 X 的某个真子集 X' , 有 $X' \rightarrow Y$, 则称 Y **部分函数依赖于 X** , 记作 $X \twoheadrightarrow Y$ 。(partial functional dependency)

□ 例:

在SCD中:

(学号, 课程号) \twoheadrightarrow 成绩。

而学号 \rightarrow 年龄, 所以 (学号, 课程号) \twoheadrightarrow 年龄。

6.2.1 函数依赖

4. 传递函数依赖 (Transitive Dependency)

□ 定义6.3 在关系模式 $R(U)$ 中, 如果 $X \rightarrow Y$, $Y \rightarrow Z$,
且 $Y \not\subseteq X$, $Y \not\rightarrow X$, $Z \not\subseteq Y$, 则称 Z 传递函数依赖于 X 。记为: $X \xrightarrow{T} Z$ 。

■ 注: 如果 $Y \rightarrow X$, 即 $X \longleftrightarrow Y$, 则 Z 直接依赖于 X 。

□ 例: 在前述关系模式SCD中, 有:

$Sno \rightarrow Sdept$, $Sdept \rightarrow Mname$

系主任 $Mname$ 传递函数依赖于学号 Sno 。

6.2.2 码

定义6.4 设 K 为关系模式 $R\langle U, F \rangle$ 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）。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）。

- **单属性码、多属性码、全码。**
- **主码**——任意选候选码之一。
- **主属性**——包含在任何一个候选码中的属性。 **非主属性**
- **超码**——若 $K \xrightarrow{P} U$ ，则 K 称为超码。

定义6.6 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**。

□ 主码和外部码一起提供了表示关系间联系的手段。