第六章 关系数据理论

函 数 依 赖

- 设计任何一个数据库应用系统，现实世界→ E-R模型→关系模型，都会遇到如何构造合适的数据模式，即逻辑结构的问题；
- 对于初步的关系模式，可能存在这样或那样的问题（如插入异常、删除异常等），需要利用**关系数据库设计理论**进行**规范化**，以逐步消除存在的问题，从而得到一定规范程度的关系模式，这就是本章要讲的内容；

- 本章是关系模型的理论基础之二，该理论是指导数据库设计的重要依据。
- 本章将阐述关系数据库中最深的一些特性——**函数依赖**、**多值依赖**和**连接依赖**，以及由此引出的**诸多异常**，如插入异常、更新异常、删除异常及冗余等，对于出现的问题，通过理论引入，对关系模式的规范化进行系统阐述。

本次课主要内容

- 关系模型概念回顾
- 关系模式中可能存在的异常
- 关系模式中存在异常的原因
- 关系模式的规范化

一、概念回顾

- 关系：描述实体、属性、实体间的联系。从形式上看，它是一张二维表；
- 关系模式：对关系的描述；关系数据库中，关系模式是型，关系是值；
- 关系数据库：基于关系模型的数据库；
- 关系数据库的模式：对关系数据库的描述，关系数据库的型也称为关系数据库模式。

关系模式的形式化定义

- 关系模式由五部分组成，即它是一个五元组：R(U, D, DOM, F)

  R:　关系名；

  U:　组成该关系的属性名集合——组属性U；

  D:　属性组U中属性所来自的域；

  DOM: 属性向域的映象集合

  F:　属性间数据的依赖关系集合

二、关系模式中可能存在的异常

例1：建立一个描述学校教务的数据库：学生的学号（Sno）、所在系（Sdept）、系主任姓名（Mname）、课程号（Cno）成绩（Grade）；

- 假设用单一的关系模式 Student 来表示，则该关系模式的属性集合 U ＝｛ Sno, Sdept, Mname, Cname, Grade ｝
- 现实世界已知事实的语义：
  - 一个系有若干学生， 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课程， 每门课程有多名学生选修；
  - 每个学生所学的每门课程都有一个成绩。

Student表

Sno	Sdept	Mname	cno	grade
S1	计算机系	张明	C1	95
S1	计算机系	张明	C2	85
S1	计算机系	张明	C3	96
S4	计算机系	张明	C1	73
S5	计算机系	张明	C1	86
…	…	…	…	…

- 数据冗余：如果一个系有多名学生或该学生选修了多门课程，则系名称和系主任姓名就要出现多次，浪费了存储设备；
- 操作异常：由于数据的冗余，在对数据操作时会引起各种异常：
  - 插入异常：如果一个系刚成立，无学生，或者虽然有学生但尚未安排课程，那么就无法把这个系及其负责人的信息加入数据库。——应该插入而未被插入
  - 删除异常：如果某个系的学生全部毕业了， 我们在删除该系学生信息的同时，把这个系及其系主任的信息也丢掉了。——不该删除的数据不得不删
- 更新异常：如果某系更换系主任或更换系名后，系统必须修改与该系学生有关的每一个元组；若有一个元组中的数据未更改，就会造成这个系主任姓名不一致现象。

- **结论：**
  - Student关系模式不是一个好的模式。
  - **“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。

- 如果将关系模式Student划分为三个关系模式S(Sno,Sdept) SC(Sno,Cno,Grade) Dept(Sdept, Mname) 可基本解决以上问题，但分解后是否是最佳的模式，也不是绝对的。

三、关系模式中存在异常的原因

- 事实上，异常现象产生的根源，就是由于关系模式中属性间存在着复杂的**依赖关系**。如学生学号和学生姓名、学生学号和院系名称、院系名称和院系系领导之间都存在着依赖关系，这种依赖都称为数据依赖；
- 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。
- 数据冗余的产生和数据依赖有着密切的关系。

- **数据依赖**
  - 是通过一个关系内部属性与属性之间的一种**约束关系**；
  - **是语义的体现；**
  - 是现实世界属性间相互联系的抽象；
  - 是数据内在的性质；
- 数据依赖的类型
  - 函数依赖（Functional Dependency，简记为FD）
  - 多值依赖（Multivalued Dependency，简记为MVD）
  - 其他

- 函数依赖

函数依赖极普遍地存在现实生活中，如描述一个学生的关系：

学生（学号、姓名、系名、系领导），

如果语义是：一个学号只对应的学生，一个学生只在一个系学习；

那么，学号→姓名；学号→系名；系名→系领导

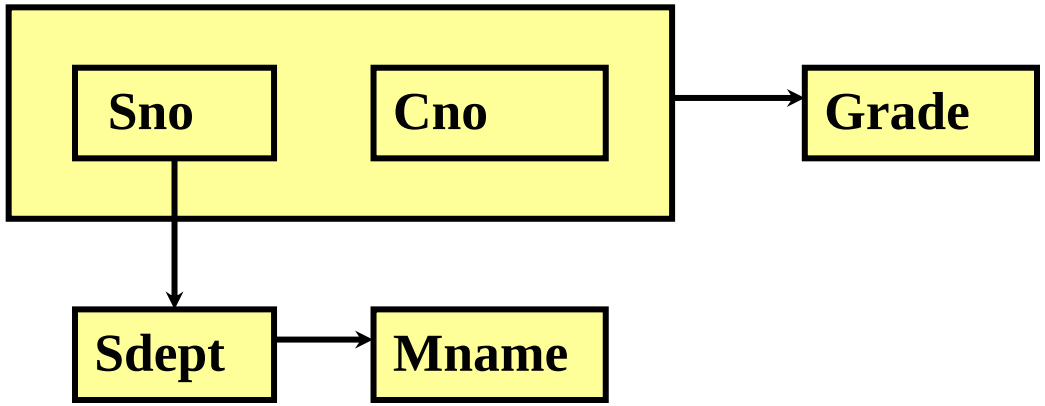
类似于 Y = F(x) 函数，变量x确定后，相应y的值也确定了；

可写为：name =f（sno）,dept =f（sno）,mname =f（sdept）

可称为：name和dept函数依赖于sno，mname函数依赖于dept；

- 对于关系模式Student＝｛Sno, Sdept, Mname, Cno, Grade ｝
- 由语义可得出一组函数依赖：

F＝｛Sno→Sdept，Sdept→Mname, (Sno, cno)→Grade ｝



- 如何构造一个好的关系模式呢？
  - 消除数据冗余，更新异常，插入异常和删除异常的关系模式
- 如何改造关系模式？
  - 如何把一个不好的关系模式分解改造为一个好的关系模式，分析一个关系模式有哪些数据依赖，如何消除那些不合适的数据依赖，这就是关系数据库设计过程中要讨论的规范化理论问题。

四、关系模式的规范化

- **关系规范化理论**正是用来改造关系模式，通过分解关系模式来消除其中不合适的数据依赖，以解决插入异常、删除异常、更新异常和数据冗余问题。
  - 函数依赖
  - 其他相关定义（候选码）
  - 关系模式的范式

1、函数依赖

- **函数依赖定义**
  - 设R(U)是属性集U上的关系模式，X,Y是U的子集。若对于R(U)的**任意**两个元组t_1和t_2，如果$t_1[X]= t_2[X]$,则$t_1[Y]= t_2[Y]$，那么称**X函数确定Y**或**Y函数依赖X**，记作 **X→Y** 。
    - **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；**
    - 任意：关系模式R中的所有关系都要满足约束条件。

- 对于关系模式R，U为属性集合，X、Y为其属性子集，根据函数依赖定义和实体间联系的定义，可得出如下结论：

  若X和Y是1：1的联系，则存在函数依赖X→Y 和Y →X；

  若X和Y是1：n的联系，则存在函数依赖Y →X；

  若X和Y是m：n的联系，则XY之间不存在函数依赖关系。

以关系模式student为例

系与系主任是 1：1，有sdept →Mname和Mname→sdept；

系与学生是1： n，故有sno → sdept

学生与课程是m ： n，故sno和cno之间不存在函数依赖

例: S (Sno, Sname, Ssex, Sage, Sdept)

假设学生姓名不允许重名，则有:

Sno → Ssex，Sno → Sage , Sno → Sdept，

Sno ←→ Sname, Sname → Ssex， Sname → Sage

Sname → Sdept

但Ssex $\not\rightarrow$ Sage

备注： 1）若X→Y，并且Y→X, 则记为X←→Y。

2）若Y不函数依赖于X, 则记为X$\not\rightarrow$Y。

函数依赖说明

- 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→年龄”这个函数依赖只有在不允许有同名人时成立。
  - 例如，如果允许一个职工只有一个电话号码，那么，职工号确定了，则其电话号码也就随之确定了。
- 设计者可以对现实世界做强制规定；
- 关系模式的**所有关系实例**都要满足约束条件。

- **函数依赖分类及其定义**
  - 平凡函数依赖（Trivial FD）
  - 非平凡函数依赖（Nontrivial FD）
  - 完全函数依赖（Full FD）
  - 部分函数依赖（Partial FD）
  - 传递函数依赖（Transitive FD）

1）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：

如果$X \rightarrow Y$，但$Y \not\subseteq X$，则称$X \rightarrow Y$是**非平凡的函数依赖；**

如果$X \rightarrow Y$，但$Y \subseteq X$, 则称$X \rightarrow Y$是**平凡的函数依赖；**

若$X \rightarrow Y$，则X称为这个函数依赖的决定属性组，也称为决定因素。

备注：对于任一关系模式，**平凡函数依赖必然成立**，它不反映新的语义，若不特别声明， 我们总是讨论非平凡函数依赖。

- **例1：在关系Student(Sno, Cno, Grade)中**

  (Sno, Cno) → Grade，但 Grade $\subsetneq$ （Cno，Sno）

  (Sno, Cno) → Sno，但 Sno $\subseteq$ (Sno, Cno)

  非平凡函数依赖：(Sno, Cno) → Grade

  平凡函数依赖：(Sno, Cno) → Sno

  (Sno, Cno) → Cno

2）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$X' \not\rightarrow Y$, 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$X \xrightarrow{P} Y$。

例1: 在关系SC(Sno, Cno, Grade)中

由于： $(Sno, Cno) \rightarrow Grade$ ， $Sno \not\rightarrow Grade$， $Cno \not\rightarrow Grade$

因此：$(Sno, Cno) \xrightarrow{F} Grade$（完全函数依赖）

由于： $(Sno, Cno) \rightarrow Sdept$ ， $Sno \rightarrow Sdept$

故： $(Sno, Cno) \xrightarrow{P} Sdept$ （部分函数依赖）

3）传递函数依赖

定义6.3

- 在R(U)中，如果$X \to Y$ ($Y \not\subseteq X$)，$Y \not\to X$，$Y \to Z$，则称Z对X传递函数依赖。记为：$X \to Z$；

- 若上述定义中有$Y \to X$，则$X \longleftrightarrow Y$，实际上为$X \xrightarrow{\text{传递}} Z$，是直接函数依赖而不是传递函数依赖。

例1: 在关系Student(Sno, Sdept, Mname , cno , Grade)中，有

Sno→ Sdept， Sdept $\not\subseteq$ Sno ，（非平凡函数依赖）

Sdept $\not\rightarrow$ sno，Sdept → Mname

则称：Mname传递函数依赖于Sno，记作 Sno $\xrightarrow{\text{传递}}$ Mname

例：模式R（Sno,Sname,Sdept,Mname,Cno, Cname, Grade),根据其语义，有如下函数依赖关系：

语义1：每个学生只会有一个学号； Sno→Sname

语义2：系与学生是一对多的关系；Sno→Sdept

语义3：系与系主任是一对一的关系； Sdept←→Mname

语义4：每门课程只有一个课程号； Cno→Cname

语义5：每个学生学习每门课程有一个成绩；

（Sno，Cno）$\xrightarrow{F}$ Grade

可推出： Sno $\xrightarrow{传递}$ Mname

- 省略部分函数依赖：如（Sno，Cno）$\xrightarrow{P}$ Sname等