第3章 关系数据库系统

关系数据库系统是支持关系模型的数据库系统，它由关系数据结构、关系操作集合和关系完整性约束3个要素组成。在关系数据库设计中，为使其数据模型合理可靠、简单实用，需要使用关系数据库的规范化设计理论。

本章首先介绍关系数据库的基本概念，围绕关系数据模型的三要素展开，利用集合、代数等抽象的数学知识深刻而透彻地介绍关系数据结构、关系数据库操作及关系数据库完整性等内容；然后讲述函数依赖的概念及分类、常见的几种范式、关系规范化理论及方法。

3.1 关系数据结构

在关系数据模型中，现实世界的实体以及实体间的各种联系均用关系来表示。在用户看来，关系模型中数据的逻辑结构是一张二维表。

3.1.1 关系的定义和性质

关系就是一张二维表，但并不是任何二维表都叫关系，我们不能把日常生活中所用的任何表格都当成一个关系直接存放到数据库中。

1. 关系的数学定义

(1) 域：一组具有相同数据类型的值的集合。

(2) 笛卡儿积：设 $D_1, D_2, \cdots, D_n$ 为任意的 n 个域，定义 $D_1, D_2, \cdots, D_n$ 的笛卡儿积为 $D_1 \times D_2 \times \cdots \times D_n = \{(d_1, d_2, \cdots, d_n) \mid d_i \in D_i, i=1,2,\cdots,n\}$。

例如有两个域，D_1 = 动物集合 = {猫，狗，猪}，D_2 = 食物集合 = {鱼，骨头，白菜}，则 D_1 与 D_2 的笛卡儿积为 $D_1 \times D_2$ = {(猫，鱼)(狗，鱼)(猪，鱼)(猫，骨头)(狗，骨头)(猪，骨头)(猫，白菜)(狗，白菜)(猪，白菜)}。

可以把这个笛卡儿积制作成二维表格的形式，如表3-1所示。

取每种动物最喜欢吃的食物的行中的数据形成动物食物表的子集，即动物食物关系表，如表3-2所示。

(3) 关系：$D_1 \times D_2 \times \cdots \times D_n$ 中有关系的行形成的一个子集称为 $D_1 \times D_2 \times \cdots \times D_n$ 上的一个关系(Relation)，用 $R(D_1, D_2, \cdots, D_n)$ 表示。其中，R 表示关系名，n 表示关系的目或元。

IBM 公司的 E. F. Codd 于 1970 年发表了关于“关系模型”的概念，但在 1980 年才真正实现，然后大规模使用。其真正的实现难度在前 5 年，因为层次模型和网状模型所采用的树和图结构用计算机表达和实现已经非常成熟，而关系模型采用的是数学概念，如何用计算机实现，用计算机表达，需要考虑很多问题(例如时间、空间复杂度等)才能得出一个合理的方法。“关系模型”真正实现后，数据处理变得很简单、很直观，编程也很容易，所以一直沿用至今。

表 3-1　动物食物表

动物	食物
猫	鱼
狗	鱼
猪	鱼
猫	骨头
狗	骨头
猪	骨头
猫	白菜
狗	白菜
猪	白菜

表 3-2　动物食物关系表

动物	食物
猫	鱼
狗	骨头
猪	白菜

2. 关系的性质

关系数据库要求其中的关系必须具有以下性质：

(1) 列是同质的，即每一列中的分量是同一类型的数据，来自同一个域。

(2) 在同一个关系中，不同列的数据可以是同一种数据类型，但各属性的名称必须互不相同。

(3) 在同一个关系中，任何两个元组都不能完全相同。

(4) 在同一个关系中，列的次序无关紧要，即列的排列顺序是不分先后的，但我们一般按使用习惯排列各列的顺序。

(5) 在同一个关系中，元组的位置无关紧要，即排行不分先后，可以任意交换两行的位置。同样，我们一般按使用习惯排列行的顺序。

(6) 关系中的每个属性必须是单值，即不可再分，这就要求关系的结构不能嵌套，这是关系应满足的最基本的条件。

例如有这样一个学生表，见表 3-3 所示的复合表，这种表格就不是关系，应对其进行结构上的修改才能成为数据库中的关系。对于该复合表，可以把它转化成一个关系，即学生成绩关系(学号，姓名，性别，系编号，程序设计，英语，高数)；也可以转化成两个关系(如表 3-4 和表 3-5 所示)，即学生关系(学号，姓名，性别，系编号)和成绩关系(学号，程序设计，英语，高数)。

表 3-3　复合表示例

学号	姓名	性别	系编号	成绩		
				程序设计	英语	高数
2012002	张三	男	01	77	87	86
2012025	李四	女	02	69	89	76
2010023	刘明	男	03	79	84	82
2011033	王晓	女	03	66	90	76

表 3-4　学生表

学号	姓名	性别	系编号
2012002	张三	男	01
2012025	李四	女	02
2010023	刘明	男	03
2011033	王晓	女	03

表 3-5　成绩表

学号	程序设计	英语	高数
2012002	77	87	86
2012025	69	89	76
2010023	79	84	82
2011033	66	90	76

3.1.2 关系数据库的基本概念

1. 关系模式

在关系数据库中,关系模式是型,关系是值,关系模式(Relation Schema)是对关系的描述。因此关系模式必须指出这个元组集合的结构,即它由哪些属性构成,这些属性来自哪些域,以及属性与域之间的映像关系。

一个关系模式应当是一个五元组,关系模式可以形式化地表示为 R(U,D,dom,F)。其中,R 是关系名;U 是组成该关系的属性名集合;D 是属性组 U 中属性来自的域;dom 是属性向域的映像集合;F 是属性间的数据依赖关系集合。

关系模式通常可以简记为 R(U)或 $R(A_1,A_2,\cdots,A_n)$。

其中,R 是关系名,$A_1,A_2,\cdots,A_n$ 为属性名;域名及属性向域的映像常常直接说明为属性的类型和长度。

【例 3-1】 已知"学生情况表"如表 3-6 所示,写出其对应的关系模式。

表 3-6 学生情况表

学号	姓名	性别	年龄	所在系
2015000101	王萧	男	17	计算机系
2015000207	李大虎	男	18	物理系
2015010302	郭敏	女	18	数学系
2013010408	高红	女	20	数学系
2014020309	王睿	男	19	美术系
2013020506	张旭	女	21	美术系

表 3-6 所示的学生情况表的关系模式可以描述为学生情况表(学号,姓名,性别,年龄,所在系)。

关系实际上就是关系模式在某一时刻的状态或内容。也就是说,关系模式是型,关系是它的值。关系模式是静态的、稳定的,而关系是动态的、随时间不断变化的,因为关系操作在不断地更新着数据库中的数据。但在实际应用中,人们常常把关系模式和关系统称为关系。

2. 关系数据库

关系数据库就是采用关系模型的数据库。在一个给定的应用领域中,所有实体及实体之间联系的关系的集合构成一个关系数据库。关系数据库的型也称为关系数据库模式,它是对关系数据库的描述,包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合,通常称为关系数据库。

3.2 关系的完整性

数据完整性是指关系模型中数据的正确性与一致性。

关系模型一般定义三类完整性约束,即实体完整性、参照完整性和用户定义的完整性约束。

1. 实体完整性规则

实体完整性规则(Entity Integrity Rule)要求关系的主码中属性具有唯一性且不能取空值。例如,表 3-6 所示的"学生情况表"中的"学号"属性既具有唯一性又不能为空。

关系模型必须遵守实体完整性规则的原因如下：

(1) 现实世界中的实体和实体之间都是可区分的，即它们具有某种唯一性标识，而关系模型中以主码作为唯一性标识。

(2) 空值就是“不知道”或“无意义”的值。主码中的属性取空值，也就说明存在某个不可标识的实体，这与第(1)条矛盾。

2. 参照完整性规则

设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的主码，如果 F 与另一个基本关系 S 的主码 K 相对应，则称 F 是基本关系 R 的外码(Foreign Key)，并称基本关系 R 为参照关系(Referencing Relation)，基本关系 S 为被参照关系(Referenced Relation)或目标关系(Target Relation)。关系 R 和 S 也可以是相同的关系，即自身参照。

目标关系 S 的主码 K 和参照关系的外码 F 可以不同名，但必须定义在同一个(或一组)域上。参照完整性规则就是定义外码与主码之间的引用规则。

参照完整性规则(Reference Integrity Rule)，若属性(或属性组)F 是基本关系 R 的外码，它与基本关系 S 的主码 K 相对应(基本关系 R 和 S 可能是相同的关系)，则 R 中的每个元组在 F 上的取值或者取空值(F 的每个属性值均为空值)，或者等于 S 中的某个元组的主码值。

【例 3-2】 “学生”实体和“系”实体可以用下面的关系表示，其中主码用下划线标识。

学生(学号，姓名，性别，年龄，系号)
系(系号，系名，系主任)

学生关系的“系号”与系关系的主码“系号”相对应，因此，“系号”属性是学生关系的外码，是系关系的主码。这里系关系是被参照关系，学生关系为参照关系。学生关系中的每个元组的“系号”属性只能取空值或系关系中“系号”已经存在的值。

3. 用户定义的完整性规则

用户定义的完整性规则(User-defined Integrity Rule)由用户根据实际情况对数据库中的数据内容进行规定，也称为域完整性规则。

通过这些规则限制数据库只接受符合完整性约束条件的数据，不接受违反约束条件的数据，从而保证数据库中数据的有效性和可靠性。

例如，表 3-4 所示的学生表中的“性别”数据只能是“男”和“女”，表 3-5 所示的成绩表中的“各科成绩”数据为 1～100 之间等。

数据完整性的作用就是要保证数据库中的数据是正确的。通过在数据模型中定义实体完整性规则、参照完整性规则和用户定义完整性规则，数据库管理系统将检查和维护数据库中数据的完整性。

3.3　关 系 运 算

关系代数是以关系为运算对象的一组高级运算的集合。关系代数是一种抽象的查询语言，是关系数据操纵语言的一种传统表达方式。关系代数的运算对象是关系，运算结果也是关系。

关系代数中的运算可以分为下面两类。

(1) 传统的集合运算：并、差、交、笛卡儿积。

(2) 专门的关系运算：投影(对关系进行垂直分割)、选择(对关系进行水平分割)、连接(关系的结合)、除法(笛卡儿积的逆运算)等。

在两类关系代数运算中，还将用到下面两类辅助操作符。

(1) 比较运算符：>、>=、<、<=、=、<>。

(2) 逻辑运算符：∨(或)、∧(与)、¬(非)。

3.3.1 传统的集合运算

传统的集合运算包括并、差、交和笛卡儿积。

1. 笛卡儿积

设关系R和S的元数(属性个数)分别为r和s，定义R和S的笛卡儿积(Cartesian Product)是一个$(r+s)$元的元组集合，每个元组的前r个分量(属性值)来自R的一个元组，后s个分量来自S的一个元组，记为R×S。形式定义如下：

$$R\times S=\{t \mid t=<t^r,t^s> \wedge t^r\in R \wedge t^s\in S\}$$

其中，t^r、t^s中的r、s为上标。若R有m个元组，S有n个元组，则R×S有$m\times n$个元组。

在实际操作时，可从R的第一个元组开始，依次与S的每一个元组组合，然后对R的下一个元组进行同样的操作，直到R的最后一个元组也进行完同样的操作为止，即可得到R×S的全部元组。

【例3-3】 已知关系R和关系S，如表3-7和表3-8所示，求R和S的笛卡儿积。

R和S的笛卡儿积如表3-9所示。

表3-7 关系R

A	B	C
a1	b2	c1
a2	b1	c3
a3	b3	c2

表3-8 关系S

E	F	D
e1	f2	d2
e2	f3	d1
e3	f1	d3

表3-9 关系R×S

A	B	C	E	F	D
a1	b2	c1	e1	f2	d2
a1	b2	c1	e2	f3	d1
a1	b2	c1	e3	f1	d3
a2	b1	c3	e1	f2	d2
a2	b1	c3	e2	f3	d1
a2	b1	c3	e3	f1	d3
a3	b3	c2	e1	f2	d2
a3	b3	c2	e2	f3	d1
a3	b3	c2	e3	f1	d3

2. 并

设关系R和S具有相同的关系模式，R和S是n元关系，R和S的并(Union)是由属于R或属于S的元组构成的集合，记为R∪S。形式定义如下：

$$R\cup S=\{t \mid t\in R \vee t\in S\}$$

其含义为任取元组 t，当且仅当 t 属于 R 或 t 属于 S 时，t 属于 R∪S。R∪S 是一个 n 元关系。

关系的并操作对应于关系的插入或添加记录的操作，俗称“＋”操作，是关系代数的基本操作。

【例 3-4】 已知关系 R 和 S 如表 3-10 和表 3-11 所示，求 R 和 S 的并。

R 和 S 的并如表 3-12 所示。

表 3-10　R

a	b	c
1	2	3
4	5	6
7	8	9

表 3-11　S

a	b	c
1	2	3
10	11	12
7	8	9

表 3-12　R∪S

a	b	c
1	2	3
4	5	6
7	8	9
10	11	12

注意：并运算可以去掉某些元组，避免表中出现重复行。

3. 差

设关系 R 和 S 具有相同的关系模式，R 和 S 是 n 元关系，R 和 S 的差(Difference)是由属于 R 但不属于 S 的元组构成的集合，记为 R－S。形式定义如下：

$$R-S=\{t \mid t\in R \wedge t\notin S\}$$

其含义为当且仅当 t 属于 R 并且不属于 S 时，t 属于 R－S。R－S 也是一个 n 元关系。关系的差操作对应于关系的删除记录的操作，俗称“－”操作。

【例 3-5】 已知关系 R(表 3-10)和 S(表 3-11)，求 R 和 S 的差。

R 和 S 的差如表 3-13 所示。

4. 交

设关系 R 和 S 具有相同的关系模式，R 和 S 是 n 元关系，R 和 S 的交(Intersection)是由属于 R 且属于 S 的元组构成的集合，记为 R∩S。形式定义如下：

$$R\cap S=\{t \mid t\in R \wedge t\in S\}$$

其含义为任取元组 t，当且仅当 t 既属于 R 又属于 S 时，t 属于 R∩S。R∩S 也是一个 n 元关系。

关系的交操作对应于寻找两关系中共有记录的操作，是一种关系查询操作，是关系代数的基本操作。

【例 3-6】 已知关系 R(表 3-10)和 S(表 3-11)，求 R 和 S 的交。

R 和 S 的交如表 3-14 所示。

表 3-13　R－S

a	b	c
4	5	6

表 3-14　R∩S

a	b	c
1	2	3
7	8	9

3.3.2　专门的关系运算

专门的关系运算包括选择、投影、连接、除等。

1. 选择

选择(Selection)运算是在关系 R 中选择满足给定条件的诸元组，记作：

$$\sigma_F(R)=\{t \mid t\in R \wedge F(t)='真'\}$$

其中,F 表示选择条件,它是一个逻辑表达式,取逻辑值“真”或“假”;逻辑表达式 F 的基本形式为 $X_1\theta\ Y_1[\Phi X_2\ \theta\ Y_2]\ldots$;$\theta$ 表示比较运算符,它可以是>、>=、<、<=、=或<>;X_1、Y_1 等是属性名、常量或简单函数,属性名也可以用它的序号来代替;Φ 表示逻辑运算符,它可以是¬、∧或∨;[ ]表示任选项,即[ ]中的部分可要可不要;“…”部分表示上述格式可以重复下去。

选择运算实际上是从关系 R 中选取使逻辑表达式 F 为真的元组,是从行的角度进行的运算。选择运算的操作示意图如图 3-1 所示。

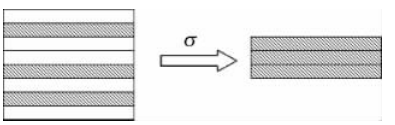

图 3-1 σ运算示意图

设有一个学生-课程数据库,其中包括学生情况表 student、课程表 course 和成绩表 score,其内容如表 3-15～表 3-17 所示。

表 3-15 学生情况表(student)

学号(no)	姓名(name)	性别(sex)	年龄(age)	所在系(dep)
2015001	张超	男	18	物理系
2015002	李岚	女	17	信息系
2015003	王芳	女	19	数学系
2015004	刘娟	女	18	信息系
2015005	赵强	男	19	物理系

表 3-16 课程表(course)

课程号(cno)	课程名(cname)	学分(credit)
1	数据库	4
2	高等数学	3
3	信息系统	2
4	操作系统	3
5	数据结构	5
6	C 程序设计	3

表 3-17 成绩表(score)

学号(no)	课程号(cno)	成绩(grade)
2015001	2	78
2015001	3	88
2015001	5	81
2015002	1	90
2015002	4	68
2015003	4	70
2015003	5	57
2015003	1	89
2015005	2	93
2015005	5	79

【例 3-7】 查询数学系学生的信息。

$$\sigma_{dep='数学系'}(student) \quad 或 \quad \sigma_{5='数学系'}(student)$$

结果如表 3-18 所示。

表 3-18 查询数学系学生的信息

学号(no)	姓名(name)	性别 (sex)	年龄(age)	所在系(dep)
2015003	王芳	女	19	数学系

【例 3-8】 查询超过 17 岁的女同学的信息。

$$\sigma_{age>17\wedge sex='女'}(student) \quad 或 \quad \sigma_{4>17\wedge\ 3='女'}(student)$$

结果如表 3-19 所示。

表 3-19　查询超过 17 岁的女同学的信息

学号(no)	姓名(name)	性别 (sex)	年龄(age)	所在系(dep)
2015003	王芳	女	19	数学系
2015004	刘娟	女	18	信息系

2. 投影(Projection)

关系 R 上的投影是从 R 中选择出若干个属性列组成新的关系,记作:

$$\pi_A(R)=\{t[A]\mid t\in R\}$$

其中,A 为 R 中的属性列。

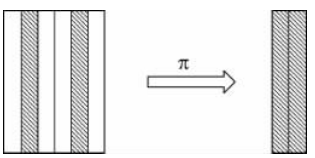

图 3-2　π 运算示意图

投影之后不仅取消了原关系中的某些列,还可能取消某些元组,因为取消了某些属性列后,就可能出现重复行,应取消这些完全相同的行。这个操作是从列的角度进行的运算,是对一个关系进行垂直分割,投影运算的直观意义如图 3-2 所示。

【例 3-9】　查询学生情况表(student)中学生的学号和姓名。

$$\pi_{no,name}(student)\quad 或\quad \pi_{1,2}(student)$$

查询结果如表 3-20 所示。

【例 3-10】　查询课程表中的课程名和课程号。

$$\pi_{cname,cno}(course)\quad 或\quad \pi_{2,1}(course)$$

查询结果如表 3-21 所示。

表 3-20　查询学生的学号和姓名

学号(no)	姓名(name)
2015001	张超
2015002	李岚
2015003	王芳
2015004	刘娟
2015005	赵强

表 3-21　查询课程表中的课程名和课程号

课程名(cname)	课程号(cno)
数据库	1
高等数学	2
信息系统	3
操作系统	4
数据结构	5
C 程序设计	6

3. 连接(Join)

1) 连接运算的含义

连接也称 θ 连接,它是从两个关系的笛卡儿积中选取满足某规定条件的全体元组,形成一个新的关系,记作:

$$R \underset{A\theta B}{\bowtie} S$$

其中,A 是 R 的属性组($A_1,A_2,\cdots,A_k$),B 是 S 的属性组($B_1,B_2,\cdots,B_K$);$A\theta B$ 的实际形式为 $A_1\theta B_1 \wedge A_2\theta B_2 \wedge \cdots \wedge A_k\theta B_k$;$A_i$ 和 B_i($i=1,2,\cdots,k$)不一定同名,但必须可比;$\theta\in\{>,<,<=,>=,=,<>\}$。

连接操作是从行和列的角度进行的运算,连接运算的直观意义如图 3-3 所示。

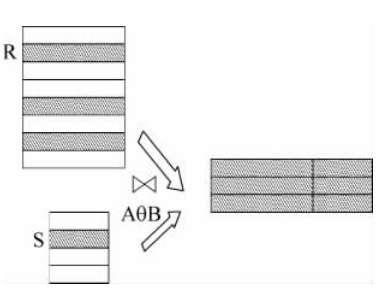

图 3-3　连接示意图

2) 连接运算的过程

首先确定结果中的属性列;然后确定参与比较的属性

列；最后逐一取R中的元组分别和S中与其符合比较关系的元组进行拼接。

3）常用的两种连接运算

（1）等值连接：θ为“=”的连接运算称为等值连接(Equal Join)，它是从关系R与S的笛卡儿积中选取A、B属性值相等的元组。等值连接记作：

$$R \underset{A=B}{\bowtie} S=\{t_r\ t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A]=t_s[B]\}$$

（2）自然连接：自然连接(Natural Join)是一种特殊的等值连接，即若A、B是相同的属性组，就可以在结果中把重复的属性去掉。这种去掉了重复属性的等值连接称为自然连接。自然连接可记作：

$$R \bowtie S=\{t_r\ t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[B]=t_s[B]\}$$

【例3-11】 已知关系R和关系S如表3-22和表3-23所示，求：

$$R \underset{B<E}{\bowtie} S,\quad R \underset{B=E}{\bowtie} S,\quad R \bowtie S。$$

表3-22 R

A	B	C
a1	6	c2
a2	7	c1
a1	9	c3
a3	12	c2

表3-23 S

E	D	C
5	e2	c2
7	e1	c1
10	e3	c3
6	e2	c2

其中，小于连接结果如表3-24，等值连接结果如表3-25，自然连接结果如表3-26。

表3-24 小于连接结果表

A	B	R.C	E	D	S.C
a1	6	c2	7	e1	c1
a1	6	c2	10	e3	c3
a2	7	c1	10	e3	c3
a1	9	c3	10	e3	c3

表3-25 等值连接结果表

A	B	R.C	E	D	S.C
a1	6	c2	6	e2	c2
a2	7	c1	7	e1	c1

表3-26 自然连接结果表

A	B	C	E	D
a1	6	c2	5	e2
a1	6	c2	6	e2
a2	7	c1	7	e1
a1	9	c3	10	e3
a3	12	c2	5	e2
a3	12	c2	6	e2

关系的除操作也是一种由关系代数基本操作复合而成的查询操作，能用其他基本操作表示，这里不再讲述。

4. 专门的关系运算操作举例

设教学数据库中有3个关系，即学生关系S(SNO,SN,AGE,SEX)、学习关系SC(SNO,CNO,SCORE)、课程关系C(CNO,CN,TEACHER)。

(1) 检索学生选修课程的详细情况。

$$S \bowtie SC \bowtie C$$

(2) 检索学习课程号为C3的学生的学号和成绩。

$$\Pi_{SNO,SCORE}(\sigma_{CNO='C3'}(SC))$$

(3) 检索学习课程号为C4的学生的学号和姓名。

$$\Pi_{SNO,SN}(\sigma_{CNO='C4'}(S \bowtie SC))$$

(4) 检索学习课程名为MATHS的学生的学号和姓名。

$$\Pi_{SNO,SN}(\sigma_{CN='MATHS'}(S \bowtie SC \bowtie C))$$

(5) 检索学习课程号为C1或C3的学生的学号。

$$\Pi_{SNO}(\sigma_{CNO='C1' \vee CNO='C3'}(SC))$$

(6) 检索不学习课程号为C2的学生的姓名和年龄。

$$\Pi_{SN,AGE}(S) - \Pi_{SN,AGE}(\sigma_{CNO='C2'}(S \bowtie SC))$$

3.4　关系的规范化

客观世界的实体间有着错综复杂的联系。实体的联系有两类，一类是实体与实体之间的联系；另一类是实体内部各属性间的联系。定义属性值间的相互关联即数据依赖，是数据库模式设计的关键。数据依赖是现实世界实体的属性间相互联系的抽象，是世界内在的性质，是语义的体现。

为使数据库模式设计合理可靠、简单实用，长期以来，形成了关系数据库设计理论，即规范化理论。它是根据现实世界存在的数据依赖进行关系模式的规范化处理，从而得到一个合理的数据库模式设计效果。

3.4.1　数据依赖

数据依赖共有3种，即函数依赖(Functional Dependency,FD)、多值依赖(Multivalued Dependency,MVD)和连接依赖(Join Dependency,JD)，其中最重要的是函数依赖。

1. 函数依赖

函数依赖是关系模式中各个属性之间的一种依赖关系，是规范化理论中的一个最重要、最基本的概念。

所谓函数依赖是指在关系R中X、Y为R的两个属性或属性组，如果对于X的每一个具体值Y都只有一个具体值与之对应，则称属性Y函数依赖于属性X，记作$X \rightarrow Y$，当Y不函数依赖于X时，记作$X \not\rightarrow Y$。当$X \rightarrow Y$且$Y \rightarrow X$时，记作$X \leftrightarrow Y$。

简单表述：如果属性X的值决定属性Y的值，那么属性Y函数依赖于属性X；或者，如果知道X的值，就可以获得Y的值。

【例 3-12】 学生情况表(如表 3-27 所示)对应的关系模式可描述为学生情况(学号,姓名,专业名,性别,出生日期,总学分)。其中,学号为关键字,求其函数依赖关系有哪些。

表 3-27 学生情况表

学号	姓名	专业名	性别	出生日期	总学分
20151101	王林	计算机	男	1996-02-10	50
20151102	程明	计算机	男	1997-02-01	50
20151103	王燕	计算机	女	1995-10-06	50
20151104	韦严平	网络	男	1997-08-26	50
20151106	李方方	网络	女	1996-11-20	50

由函数依赖的定义可知,存在以下函数依赖关系集:

学号→姓名,学号→专业名,学号→性别,学号→出生日期,学号→总学分

2. 几种特定的函数依赖

1) 非平凡函数依赖和平凡函数依赖

设关系模式 R(U),U 是 R 上的属性集,X、Y⊆U;如果 X→Y,且 Y 是 X 的子集,则称 X→Y 为平凡的函数依赖;如果 X→Y,且 Y 不是 X 的子集,则称 X→Y 为非平凡的函数依赖。

【例 3-13】 在学生课程(学号,课程号,成绩)关系中,若存在函数依赖(学号,课程号)→成绩,该函数依赖是非平凡函数依赖。

2) 完全函数依赖和部分函数依赖

设关系模式 R(U),U 是 R 上的属性集,X、Y⊆U;如果 X→Y,并且对于 X 的任何一个真子集 Z,Z→Y 都不成立,则称 Y 完全函数依赖于 X;如果 X→Y,但对于 X 的某一个真子集 Z 有 Z→Y 成立,则称 Y 部分函数依赖于 X。

【例 3-14】 在学生课程(学号,课程号,成绩)关系中,“学号,课程号”是主码,由于“学号→成绩”不成立,“课程号→成绩”也不成立,因此,“成绩”完全函数依赖于(学号,课程号)。

3) 传递函数依赖

设关系模式 R(U),X⊆U,Y⊆U,Z⊆U,如果 X→Y,Y↛X,且 Y→Z 成立,则称 X→Z 为传递函数依赖。

【例 3-15】 学生关系(学号,姓名,性别,年龄,所在系,系主任)的函数依赖集 F={学号→姓名,学号→性别,学号→年龄,学号→所在系,所在系→系主任,学号→系主任},则学号→系主任为传递函数依赖。

3. 码的函数依赖表示

使用函数依赖的概念可以给出关系模式中码的更严格的定义。

(1) 候选码(Candidate Key):设 K 为关系模式 R(U)中的属性或属性集合,若 K→U,则 K 称为 R 的一个候选码。

(2) 主码(Primary Key):若关系模式 R 有多个候选码,则选定其中一个作为主码。

3.4.2 关系规范化的目的

若设计一个描述学校的数据库:一个系有若干学生,一个学生只属于一个系;一个系只有一名主任;一个学生可以选修多门课程,每门课程有若干学生选修;每个学生所学的每门课程都有一个成绩,如表 3-28 所示。

表 3-28 学生信息表

学号	姓名	年龄	系别	系主任	课程号	成绩
S1	赵红	20	计算机	张力	C1	90
S1	赵红	20	计算机	张力	C2	85
S2	王小明	17	数学	王晓	C5	57
S2	王小明	17	数学	王晓	C6	80
S2	王小明	17	数学	王晓	C7	76
S2	王小明	17	数学	王晓	C4	70
S3	吴小林	19	信息	赵钢	C1	75
S3	吴小林	19	信息	赵钢	C2	70
S4	张涛	21	计算机	张力	C1	93

则上述数据库对应的关系模式为学生信息表(学号,姓名,年龄,系别,系主任,课程号,成绩),学号、课程号为主键。

在上述关系模式中存在以下问题。

(1) 数据冗余:数据在数据库中的重复存放称为数据冗余。冗余度大,不仅浪费存储空间,更重要的是在对数据进行修改时容易造成数据的不一致。例如系名,学生姓名、年龄等都要重复存储多次,当它们发生改变时,就需要修改多次,一旦遗漏就使数据不一致。

(2) 更新异常:因为存在数据冗余,在更新数据时,维护数据完整性的代价就会增大。如果某学生改名,则该学生的所有记录都要逐一修改姓名的值;稍有不慎,就有可能漏改某些记录。

(3) 插入异常:无法插入某部分信息称为插入异常,即该插入的数据插不进去。例如,如果一个系刚成立,尚无学生,我们就无法把这个系及其系主任的信息存入数据库。因为学号与课程号是主键,主键不能为空。

(4) 删除异常:不该删除的数据不得不删除。例如,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。

上述关系模式设计不合理,不是一个好的关系模式,“好”的关系模式不会发生插入异常、删除异常、更新异常,数据冗余也应尽可能的少。

关系模式规范化的目的就是解决关系模式中存在的数据冗余、插入和删除异常以及更新异常等问题。其基本思想是消除数据依赖中不合适的部分,使各关系模式达到某种程度的分离,使一个关系描述一个概念、一个实体或实体间的一种联系。因此,规范化的实质是概念的单一化。

关系数据库中的关系必须满足一定的规范化要求,对于不同的规范化程度可用范式来衡量。范式(Normal Form)是符合某一种级别的关系模式的集合,是衡量关系模式规范化程度的标准,只有达到的关系才是规范化的。目前主要有 6 种范式,即第一范式、第二范式、第三范式、BC 范式、第四范式和第五范式。满足最低要求的范式叫第一范式,简称为 1NF。在第一范式的基础上进一步满足一些要求的范式为第二范式,简称为 2NF。其余以此类推,显然各种范式之间存在联系:1NF⊃2NF⊃3NF⊃BCNF⊃4NF⊃5NF。

通常把某一关系模式 R 为第 n 范式简记为 R∈nNF。

范式的概念最早是由 E. F. Codd 提出的。在 1971 年到 1972 年期间,他先后提出了 1NF、2NF、3NF 的概念,1974 年他又和 Boyee 共同提出了 BCNF 的概念,1976 年 Fagin 提出了

4NF的概念,后来又有人提出了5NF的概念。在这些范式中,最重要的是3NF和BCNF,它们是进行规范化的主要目标。

3.4.3 关系规范化的过程

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的集合,这个过程称为规范化。在通常情况下,规范化到3NF就可以了。

1. 第一范式

设R是一个关系模式,如果R的每个属性的值域都是不可分的简单数据项(原子值)的集合,则称这个关系模式属于第一范式,简记作R∈1NF。

也可以说,如果关系模式R的每一个属性都是不可分解的,则R为第一范式的模式,1NF是规范化最低的范式。

在任何一个关系数据库系统中,关系至少应该是第一范式,不满足第一范式的数据库模式不能称为关系数据库。但要注意,第一范式不能排除数据冗余和异常情况的发生。

例如,表3-29描述的是某单位职工情况。

表3-29 职工情况表

职工号	姓名	工资		
		基本工资	职务工资	工龄工资
20011	李岚	3290	800	430
20013	王晓江	3000	800	340

由于上表中的工资项包括3个部分,不满足每个属性不能分解的条件,是非规范化表,不是第一范式,可规范化为表3-30所示。

表3-30 职工情况表

职工号	姓名	基本工资	职务工资	工龄工资
20011	李岚	3290	800	430
20013	王晓江	3000	800	340

2. 第二范式

如果关系模式R属于第一范式,且它的每个非主属性都完全函数依赖于码(候选码),则称R为满足第二范式的关系模式,简记为R∈2NF。

注意:在一个关系中,包含在任何候选码中的各个属性称为主属性;不包含在任何候选码中的属性称为非主属性。

在规范化时,我们采用的是每个关系的最小函数依赖集,最小函数依赖集是符合以下条件的函数依赖集F:

(1) F中任何一个函数依赖的右部仅含有一个属性。

(2) F中的所有函数依赖的左边都没有冗余属性。

(3) F中不存在冗余的函数依赖。

【例3-16】 在学生关系S(学号,姓名,性别,课程号,学分)中,学号和课程号的组合为主码,姓名、性别、学分为非主属性,关系S中的最小函数依赖集为:

学号→姓名,学号→性别,(学号,课程号)→学分

实际上,函数依赖“(学号,课程号)→姓名”也成立,但左边的“课程号”是多余的;函数依赖“学号→学号”也成立,但这是一个冗余的函数依赖。

两个推论:

(1) 关系 R∈1NF,且其主关键字只有一个属性,则关系 R 一定属于第二范式。

【例 3-17】 在关系 R(学号,姓名,出生日期,成绩)中主码为学号,姓名、出生日期、成绩为非主属性,存在下列最小函数依赖集:

学号→姓名,学号→性别,学号→出生日期,学号→成绩

由于每个非主属性都完全函数依赖于码,所以该关系 R∈2NF。

(2) 主关键字是属性的组合,这样的关系模式可能不属于第二范式。

对于例 3-16 中的最小函数依赖集,存在非主属性(姓名和性别)部分函数依赖于码,故关系 S 不属于 2NF。对上述关系模式进行分解,分解方法为每个非主属性与它所依赖的属性组成新关系,新关系要尽可能的少,新关系的主码为函数依赖的左侧属性或属性集。则上述关系模式分解为下面两个关系:

S1(学号,姓名,性别)和 S2(学号,课程号,学分),且 S1∈2NF,S2∈2NF。

【例 3-18】 在职工信息关系 P(职工号,姓名,职称,项目号,项目名称,项目排名)中,主码为职工号、项目号的组合,非主属性为姓名、职称、项目名称、项目排名,关系 P 中的最小函数依赖集如下:

职工号→姓名,职工号→职称,项目号→项目名称,(职工号,项目号)→项目排名

由于非主属性部分依赖于码,故关系 P 不属于 2NF。对上述关系模式进行分解,分解为职工信息表(职工号,姓名,职称)、项目排名表(职工号,项目号,项目排名)和项目表(项目号,项目名称)。

3. 第三范式

如果关系模式 R 属于第二范式,且没有一个非主属性传递函数依赖于码,则称 R 为满足第三范式的关系模式,简记作 R∈3NF。

【例 3-19】 关系 ST(学号,楼号,收费)包含的最小函数依赖集为:

学号→楼号,楼号→收费

函数依赖“学号→收费”也成立,但因为“收费”不是直接而是传递函数依赖于“学号”,所以这是一个冗余的函数依赖。

对上述关系模式进行分解,分解为 ST1(学号,楼号)和 ST2(楼号,收费)两个关系。

推论:如果关系模式 R∈1NF,且它的每一个非主属性既不部分也不传递函数依赖于码,则 R∈3NF。

通过 3NF 的定义,我们也可以得出这样的推论:不存在非主属性的关系模式一定属于 3NF。此推论由读者自行证明。

4. BC 范式

关系模式 R∈1NF,对于任何非平凡的函数依赖 X→Y,X 均包含码,则称 R 为满足 BCNF 的关系模式,简记作 R∈BCNF。

BCNF是从1NF直接定义而成的,可以证明,如果R∈BCNF,则R∈3NF。

由BCNF的定义可以看到,每个BCNF的关系模式都具有以下3个性质:

(1) 所有非主属性完全函数依赖于每个候选码。

(2) 所有主属性完全函数依赖于每个不包含它的候选码。

(3) 没有任何属性完全函数依赖于非码的任何一组属性。

如果关系模式R∈BCNF,由定义可知,R中不存在任何属性传递函数依赖或部分依赖于任何候选码,所以必定有R∈3NF。但是,如果R∈3NF,R未必属于BCNF。

3NF和BCNF是以函数依赖为基础的关系模式规范化程度的测度。

如果一个关系数据库中的所有关系模式都属于BCNF,那么在函数依赖范畴内,它已实现了模式的彻底分解,达到了最高的规范化程度,消除了插入异常和删除异常。

在信息系统的设计中,普遍采用的是"基于3NF的系统设计"方法,就是由于3NF是无条件可以达到的,并且基本解决了"异常"的问题,因此这种方法目前在信息系统的设计中仍然被广泛应用。

如果仅考虑函数依赖这一种数据依赖,属于BCNF的关系模式已经很完美了。但如果考虑其他数据依赖,例如多值依赖,属于BCNF的关系模式仍存在问题,不能算是一个完美的关系模式。而4NF研究的就是关系模式中多值依赖的问题,5NF研究的是关系模式中连接依赖的问题,这里不再讲述。

5. 关系规范化总结

(1) 对1NF关系进行投影,消除原关系中非主属性对码的部分函数依赖,从而产生若干个2NF的关系。

(2) 对2NF关系进行投影,消除原关系中非主属性对码的传递函数依赖,从而产生若干个3NF的关系。

(3) 对3NF关系进行投影,消除原关系中主属性对码的部分函数依赖和传递函数依赖(也就是说,使决定属性都成为投影的候选码),得到一组BCNF的关系。

总之,关系的规范化减少了冗余数据,节省了空间,避免了不合理的插入、删除、修改等操作,保持了数据的一致性;但是也导致了一些缺点,例如信息放在不同表中,查询数据时有时需要把多个表连接在一起,增加了操作的时间和难度,因此关系模式要从实际设计的目标出发进行设计。

习 题 3

1. 关系数据模型由哪3个要素组成?

2. 简述关系的性质。

3. 简述关系的完整性。

4. 传统的集合运算和专门的关系运算都有哪些?

5. 解释下列术语的含义:函数依赖、平凡函数依赖、非平凡函数依赖、部分函数依赖、完全函数依赖、传递函数依赖、范式。

6. 非规范化的关系中存在哪些问题?

7. 简述关系模式规范化的目的。

8. 根据给定的关系模式进行查询。

设有学生-课程关系数据库，它由3个关系组成，即学生S(学号S#，姓名SN，所在系SD，年龄SA)、课程C(课程号C#，课程名CN，先修课号PC#)、SC(学号S#，课程号C#，成绩G)。请用关系代数分别写出下列查询：

(1) 检索学生年龄大于等于20岁的学生姓名。

(2) 检索先修课号为C2的课程号。

(3) 检索课程号C1的成绩为90分以上的所有学生姓名。

(4) 检索001号学生修读的所有课程名及先修课号。

(5) 检索年龄为19岁的学生所修读的课程名。

9. 设有关系模式R(运动员编号，姓名，性别，班级，班主任，项目号，项目名，成绩)，如果规定每名运动员只能代表一个班级参加比赛，每个班级只能有一个班主任；每名运动员可参加多个项目，每个比赛项目也可由多名运动员参加；每个项目只能有一个项目名；每名运动员参加一个项目只能有一个成绩。根据上述语义，回答下列问题：

(1) 写出关系模式R的主关键字。

(2) 分析R最高属于第几范式，说明理由。

(3) 若R不是3NF，将其分解为3NF。

10. 设有关系模式R(职工号，日期，日营业额，部门名，部门经理)，如果规定每个职工每天只有一个营业额，每个职工只在一个部门工作，每个部门只有一个经理。

(1) 根据上述规定，写出模式R的主关键字。

(2) 分析R最高属于第几范式，说明理由。

(3) 若R不是3NF，将其分解为3NF。