

第 3 章

CHAPTER

关系数据库

本章学习重点：

- 关系完整性。
- E-R 图转换为关系模型。
- 关系运算。

关系数据库系统是支持关系模型的数据库系统。关系数据模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。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。

3.1 关系数据结构

关系模型的数据结构非常简单。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格,但并不是任何二维表都叫作关系,人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里。

关系数据库要求其中的关系必须具有以下性质。

- (1) 在同一个关系中,同一列的数据必须是同一种数据类型。
- (2) 在同一个关系中,不同列的数据可以是同一种数据类型,但各属性的名称都必须互不相同。
- (3) 在同一个关系中,任意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。

(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。

(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,有学生成绩如表 3-1 所示。这种表格就不是关系,应对其修改结构,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,年龄,班级号,数学,语文,英语);也可以转化成两个关系,即学生关系(学号,姓名,性别,年龄,班级号)和成绩关系(学号,数学,语文,英语),如表 3-2 和表 3-3 所示。

表 3-1 学生成绩表

学号	姓名	性别	年龄	班级号	成 绩		
					数学	语文	英语
201501	张鹏	男	20	01	77	81	85
201502	王宏	女	19	01	78	82	86
201503	赵飞	男	21	02	79	83	87
201504	黄明	男	19	02	80	84	88

表 3-2 学生关系表

学 号	姓 名	性 别	年 龄	班 级 号
201501	张鹏	男	20	01
201502	王宏	女	19	01
201503	赵飞	男	21	02
201504	黄明	男	19	02

表 3-3 成绩关系表

学 号	数 学	语 文	英 语
201501	77	81	85
201502	78	82	86
201503	79	83	87
201504	80	84	88

所以,关系是一种规范化了的二维表格,是一个属性数目相同的元组的集合。集合中的元素是元组,每个元组的属性数目应该相同。

在关系数据模型中,实体以及实体之间的联系都是用关系来表示的,它是通过关系中的冗余属性(一般是主码和外码的关系)来实现实体之间的联系。上例中学生关系和成绩

关系就是通过“学号”属性实现的一对一联系,即一个学生只有一行成绩,而一行成绩也只属于一个学生。

3.1.2 关系模式

关系数据库中,关系模式(relation schema)是型,关系是值;关系模式是对关系的描述。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。

关系模式的定义:关系的描述称为关系模式。一个关系模式应是一个五元组。关系模式可以形式化地表示为 $R(U, D, \text{dom}, F)$ 。

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 中属性所来自的域, dom 是属性间域的映像集合, 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

$$R(U) \text{ 或 } R(A_1, A_2, \dots, A_n)$$

其中, R 是关系名, $A_1, A_2, \dots, A_n$ 为属性名,域名及属性间域的映像,常常直接说明为属性的类型、长度。

关系实际上就是关系模式在某一时刻的状态或内容。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更新着数据库中的数据,可以把关系模式和关系系统称为关系。

3.1.3 关系数据库

关系数据库就是采用关系模型的数据库。关系数据库有型和值的区别,关系数据库的型是指对关系数据库的描述,它包括若干关系模式;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。

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,是对关系数据库的描述,它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 关系完整性

关系完整性是指关系模型中数据的正确性与一致性。关系模型允许定义三类完整性规则:实体完整性、参照完整性和用户定义的完整性规则。

1. 实体完整性规则(entity integrity rule)

实体完整性要求关系的主码具有唯一性且主码中的每一个属性都不能取空值。例如,学生表中的学号属性应具有唯一性且不能为空。

关系模型必须满足实体完整性的原因如下。

(1) 现实世界中的实体和实体间的联系都是可区分的,即它们具有某种唯一性标识。相应地,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。主属性取空值,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(reference integrity rule)

设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的码,如果 F 与基本关系 S 的主码 Ks 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 Ks 和参照关系的外码 F 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: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 Ks 相对应(基本关系 R 和 S 可能是相同的关系),则对于 R 中每个元组在 F 上的值必须为空值(F 的每个属性值均为空值);或者等于 S 中某个元组的主码值。

【例 3-1】 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,系号)

系(系号,系名)

学生关系的“系号”与系关系的“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”属性只能取下面两类值:空值或系关系中已经存在的值。

3. 用户定义的完整性规则

用户定义的完整性规则是由用户根据实际情况对数据库中数据的内容进行的规定,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值,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,学生表中的性别数据只能是男和女,选课表中的成绩数据为 1~100 等。

关系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 and 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 将 E-R 图转换为关系数据模型

关系数据模型是一组关系模式的集合,而 E-R 图是由实体、属性和实体之间的联系三要素组成的。所以将 E-R 图转换为关系数据模型实际上是要将实体、属性和实体之间的联系转换为关系模式。

3.3.1 实体的转换

一个实体转换为一个关系模式。实体的属性就是关系的属性。实体的主码就是关系的主码。

【例 3-2】 将图 3-1 所示的学生实体转换为关系模式。

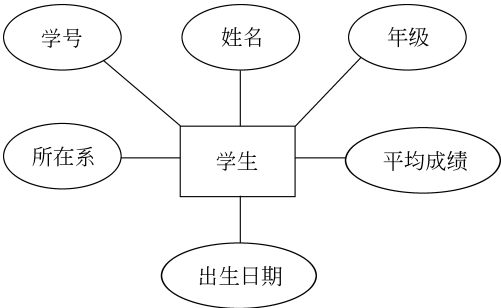


图 3-1 学生实体图

对应的关系模式为
学生(学号, 姓名, 出生日期, 所在系, 年级, 平均成绩)
其中,学号为主码,用下画线标识。

3.3.2 联系的转换

两实体间联系的转换根据联系类型的不同可分为如下 3 种。

1. 1 : 1 联系的转换

两实体间 1 : 1 联系可以转换为一个独立的关系模式,也可以与某一端对应的关系模式合并。

1) 转换为一个独立的关系模式

转换后的关系模式中关系的属性包括与该联系相连的各实体的码以及联系本身的属性;关系的主码为两个实体的主码的组合。

【例 3-3】 将图 3-2 所示的 E-R 图转换为关系模式。

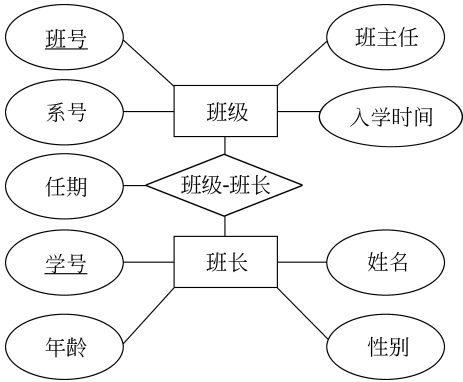


图 3-2 1 : 1 联系的 E-R 图

转换成如下关系模式：
班级(班号, 系号, 班主任, 入学时间)
班长(学号, 姓名, 性别, 年龄)

班级-班长(班号,学号,任期)

2) 与某一端对应的关系模式合并

合并后关系的属性包括自身关系的属性和另一关系的码及联系本身的属性;合并后关系的码不变。

【例 3-4】 将图 3-2 所示的 E-R 图,按照 2)转换为关系模式。

转换成如下关系模式:

班级(班号,系号,班主任,入学时间,学号,任期)

班长(学号,姓名,性别,年龄)

或

班级(班号,系号,班主任,入学时间)

班长(学号,姓名,性别,年龄,班号,任期)

2. 1 : N 联系的转换

两实体间 1 : N 联系可以转换为一个独立的关系模式,也可以与 N 端对应的关系模式合并。

1) 转换为一个独立的关系模式

关系的属性包括与该联系相连的各实体的码以及联系本身的属性;关系的主码为 N 端实体的码。

【例 3-5】 将图 3-3 所示的 E-R 图,关系转换为独立的关系模式。

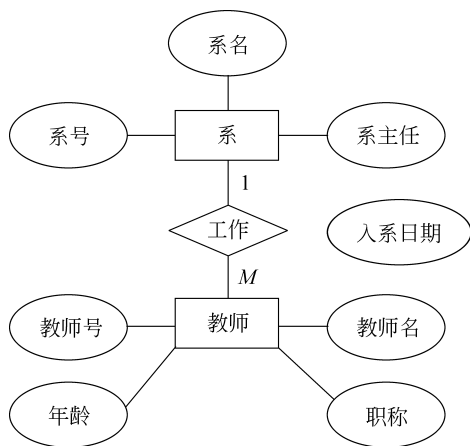


图 3-3 1 : M 联系的 E-R 图

对应的关系模式如下:

系(系号,系名,系主任)

教师(教师号,教师名,年龄,职称)

工作(教师号,系号,入系日期)

2) 与 N 端对应的关系模式合并

合并后关系的属性包括在 N 端关系中加入 1 端关系的主码和联系本身的属性。合并后关系的码不变。

【例 3-6】 将图 3-3 所示的 E-R 图,按照 2)转换为合并的关系模式。

对应的关系模式如下:

系(系号,系名,系主任)

教师(教师号,教师名,年龄,职称,系号,入系日期)

注意: 实际使用中通常采用这种方法以减少关系模式,因为多一个关系模式就意味着查询过程中要进行连接运算,降低查询的效率。

3. M : N 联系的转换

两实体间 $M:N$ 联系必须为联系产生一个新的关系。该关系中至少包含被它所联系的双方实体的主关键字,若联系有属性,也要并入该关系中。

【例 3-7】 将图 3-4 所示的 E-R 图转换成对应的关系模式。

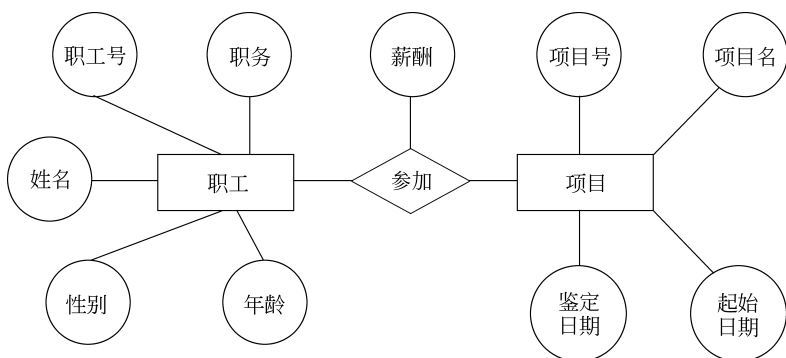


图 3-4 $M:N$ 联系的 E-R 图

对应的关系模式如下:

职工(职工号,姓名,性别,年龄,职务)

项目(项目号,项目名,起始日期,鉴定日期)

参加(职工号,项目号,薪酬)

3.4 关系运算

关系代数是以前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,是关系数据操作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。

关系代数中的操作可以分为如下两类。

传统的集合运算:笛卡儿积、并、差、交。

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

3.4.1 传统的集合运算

传统的集合运算包括笛卡儿积($\times$)、并($\cup$)、差($-$)和交($\cap$)。

1. 笛卡儿积(Cartesian product)

设关系 R 和 S 的属性个数分别为 r 和 s , 定义 R 和 S 的笛卡儿积是一个 $(r+s)$ 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 $R \times S$ 。形式化定义如下:

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

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 $R \times S$ 有 $m \times n$ 个元组。

实际操作时, 可以从 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, 直至 R 的最后一个元组也进行完同样的操作, 即可得到 $R \times S$ 的全部元组。

【例 3-8】 已知关系 R 和关系 S 如表 3-4 和表 3-5 所示, 求 R 和 S 的笛卡儿积。

表 3-4 关系 R

M	N
m1	n1
m2	n2
m3	n3

表 3-5 关系 S

X	Y
x1	y1
x2	y2
x3	y3

R 和 S 的笛卡儿积结果如表 3-6 所示。

表 3-6 R 和 S 的笛卡儿积

M	N	X	Y
m1	n1	x1	y1
m1	n1	x2	y2
m1	n1	x3	y3
m2	n2	x1	y1
m2	n2	x2	y2
m2	n2	x3	y3
m3	n3	x1	y1
m3	n3	x2	y2
m3	n3	x3	y3

2. 并(union)

设关系 R 和 S 具有相同的模式, R 和 S 是 n 元关系, R 和 S 的并是由属于 R 或属于 S 的元组构成的集合, 记为 $R \cup S$ 。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 或属于 S 时, t 属于 $R \cup S$ 。 $R \cup S$ 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 3-9】 已知关系 R 和 S 如表 3-7 和表 3-8 所示,求 R 和 S 的并。

R 和 S 的并结果如表 3-9 所示。

表 3-7 关系 R

M	N	K
m1	n2	k3
m4	n5	k6
m7	n8	k9

表 3-8 关系 S

M	N	K
m1	n2	k3
m4	n5	k6
m10	n11	k12

表 3-9 R 和 S 的并结果

M	N	K
m1	n2	k3
m4	n5	k6
m7	n8	k9
m10	n11	k12

注意：并运算会取消某些元组(避免重复行)。

3. 差(difference)

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

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

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 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,是关系代数的基本操作。

【例 3-10】 已知关系 R 和 S 如表 3-7 和表 3-8 所示,求 R 和 S 的差。

R 和 S 的差结果如表 3-10 所示。

表 3-10 R 和 S 的差结果

M	N	K
m7	n8	k9

4. 交(intersection)

关系 R 和 S 具有相同的模式, R 和 S 是 n 元关系, R 和 S 的交是由属于 R 且属于 S 的元组构成的集合, 记为 $R \cap S$ 。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 $R \cap S$ 。 $R \cap S$ 是一个 n 元关系。

关系的交操作对应于寻找两关系共有记录的操作, 是一种关系查询操作。

【例 3-11】 已知关系 R 和 S 如表 3-7 和表 3-8 所示, 求 R 和 S 的交。

R 和 S 的交结果如表 3-11 所示。

表 3-11 R 和 S 的交结果

M	N	K
m1	n2	k3
m4	n5	k6

3.4.2 专门的关系运算

专门的关系运算包括选择、投影、连接等。

1. 选择(selection)

选择运算是在关系 R 中选择满足给定条件的诸元组, 记作

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

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 $X_1 \theta Y_1 [\phi X_2 \theta Y_2] \dots$ 。其中, θ 表示比较运算符, 它可以是 $>$ 、 $>=$ 、 $<$ 、 $<=$ 、 $=$ 或 $<>$ 。 X_1, Y_1 等是属性名或常量或简单函数。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 $\neg$ 、 $\wedge$ 或 $\vee$ 。 $[\]$ 表示任选项, 即 $[\]$ 中的部分可要可不要, $\dots$ 表示上述格式可以重复下去。

因此, 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。这是从行的角度进行的运算, 如图 3-5 所示。

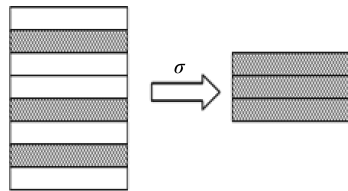


图 3-5 选择运算示意图

接下来举例说明选择运算的操作。

有学生表(student)如表 3-12 所示、课程表(course)如表 3-13 所示和成绩表(score)如表 3-14 所示。

表 3-12 学生表(student)

学 号	姓 名	性 别	年 龄	专 业
20154103101	王 鹏	男	18	软件工程
20154103102	张 超	女	17	大数据

续表

学 号	姓 名	性 别	年 龄	专 业
20154103103	刘明浩	男	19	计算机科学与技术
20154103104	李 明	男	18	软件工程
20154103105	王 晓	女	18	大数据

表 3-13 课程表(course)

课 程 号	课 程 名	学 分
20001	Java	3
20002	C++	4
20003	数据库	3
20004	操作系统	3
20005	Web 设计	2
20006	算法分析	4
20007	C 程序设计	3

表 3-14 成绩表(score)

学 号	课 程 号	成 绩
20154103101	20001	67
20154103101	20001	78
20154103102	20004	98
20154103102	20005	56
20154103103	20005	87
20154103103	20002	68

【例 3-12】 查询软件工程专业学生的信息。

$\sigma_{\text{专业}='软件工程'}(\text{student})$ 或 $\sigma_5='软件工程'(\text{student})$,结果如表 3-15 所示。

表 3-15 查询软件工程专业学生的信息

学 号	姓 名	性 别	年 龄	专 业
20154103101	王鹏	男	18	软件工程
20154103104	李明	男	18	软件工程

【例 3-13】 查询年龄大于 17 的女同学的信息。

$\sigma_{\text{年龄}>17 \wedge \text{性别}='女'}(\text{student})$ 或 $\sigma_{4>17 \wedge 3='女'}(\text{student})$,结果如表 3-16 所示。

表 3-16 查询年龄大于 17 的女同学的信息

学 号	姓 名	性 别	年 龄	专 业
20154103105	王晓	女	18	大数据

2. 投影(projection)

关系 R 上的投影是从 R 中选择出若干属性列组成新的关系,记作

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

其中,A 为 R 中的属性列。投影之后不仅取消了原关系中的某些列,而且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。

这个操作是对一个关系进行垂直分割,投影运算的直观意义如图 3-6 所示。

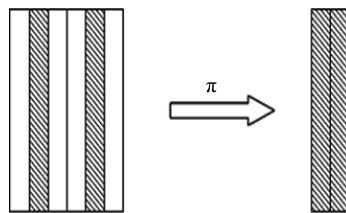


图 3-6 投影运算示意图

【例 3-14】 查询学生的学号和姓名。

$\pi_{\text{学号,姓名}}(\text{student})$ 或 $\pi_{1,2}(\text{student})$,结果如表 3-17 所示。

表 3-17 查询学生的学号和姓名

学 号	姓 名
20154103101	王 鹏
20154103102	张 超
20154103103	刘明浩
20154103104	李 明
20154103105	王 晓

【例 3-15】 查询课程表中的课程名和课程号。

$\pi_{\text{课程号,课程名}}(\text{course})$ 或 $\pi_{1,2}(\text{course})$,结果如表 3-18 所示。

表 3-18 查询课程名和课程号

课 程 号	课 程 名
20001	Java
20002	C++
20003	数据库
20004	操作系统
20005	Web 设计
20006	算法分析
20007	C 程序设计

3. 连接(join)

1) 连接运算的含义：

连接也称 θ 连接,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,记为

$$R \bowtie_{A\theta B} S = \sigma_{A\theta B}(R \times S) = \{t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B]\}$$

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

连接操作是从行和列的角度进行运算,连接运算的直观意义如图 3-7 所示。

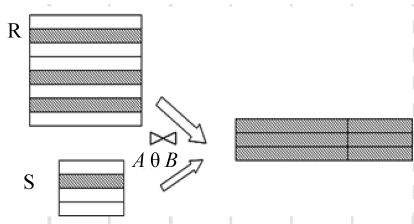


图 3-7 连接运算示意图

2) 连接运算的过程

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) 两类常用连接运算

(1) 等值连接(equal-join)

θ 为 = 的连接运算称为等值连接,它是从关系 R 与 S 的笛卡儿积中选取 A、B 属性值相等的那些元组。等值连接为

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

(2) 自然连接(natural join)

自然连接是一种特殊的等值连接,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：

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

接下来举例说明以上连接类型。

【例 3-16】 已知关系 R 和关系 S 如表 3-19 和表 3-20 所示,求 $R \bowtie_{N<Y} S, R \bowtie_{N=Y} S, R \bowtie S$ 。

表 3-19 关系 R

M	N	K
m1	5	k1
m2	6	k2
m1	10	k3

表 3-20 关系 S

X	Y	K
x1	6	k1
x2	11	k2
x3	2	k3

上述 3 种运算的结果如表 3-21～表 3-23 所示。

表 3-21 $N < Y$ 的运算结果

M	N	R.K	X	Y	S.K
m1	5	k1	x1	6	k1
m1	5	k1	x2	11	k2
m2	6	k2	x2	11	k2
m1	10	k3	x2	11	k2

表 3-22 $N = Y$ 的运算结果

M	N	R.K	X	Y	S.K
m2	6	k2	x1	6	k1
a2	b1	7	7	e1	b1

表 3-23 自然连接的运算结果

M	N	R.K	X	Y
m1	5	k1	x1	6
m2	6	k2	x2	11
m1	10	k3	x3	2

习 题

一、选择题

1. 采用二维表格结构表达实体型及实体间联系的数据模型是()。
- A. 层次模型
- B. 网状模型
- C. 关系模型
- D. 实体联系模型
2. 同一个关系模型的任意两个元组值()。
- A. 不能完全相同
- B. 可以完全相同
- C. 必须完全相同
- D. 以上都不是

3. 参照完整性规定: 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 Ks 相对应,则对于 R 中每个元组在 F 上的值必须为()。

- A. 只能取空值
- B. 只能等于 S 中某个元组的主码值
- C. 或者为空值,或者等于 S 中某个元组的主码值
- D. 可以取任意值

4. 学生实体和系别实体如下,其中主码用下画线标识

学生(学号,姓名,年龄,系别号)

系别(系别号,系名)

则学生关系的外码是()。

- A. 学号
- B. 系别号
- C. 姓名
- D. 年龄

5. 设关系: 职工(职工号,姓名,年龄,性别),下列可以求出年龄大于 50 岁的职工信息的关系代数是()。

- A. $\sigma_{\text{年龄}>50}(\text{职工})$
- B. $\pi_{\text{年龄}>50}(\text{职工})$
- C. $\sigma_{50}(\text{职工})$
- D. $\pi_{50}(\text{职工})$

二、填空题

- 1. ()规则是指若属性 A 是基本关系 R 的主属性,则属性 A 不能取空值。
- 2. 关系模型的三类完整性约束是实体完整性规则、参照完整性规则和()。
- 3. 在专门关系运算中,从表中按照要求取出指定属性的操作称为()。
- 4. 关系代数中 4 类传统的集合运算分别为()、差运算、交运算和广义笛卡儿积运算。

三、操作题

1. 学生和教师管理教学模型如下。

学生: 学号、姓名、性别、年龄。

教师: 编号、姓名、性别、年龄、职称。

课程: 课程号、课程名、课时、学分。

一门课程只安排一名教师任教,一名教师可任多门课程。教师任课包括任课时间和使用教材。

一门课程有多名学生选修,每名学生可选多门课。学生选课包括考核成绩。

完成如下设计:

- (1) 设计学生和教师管理教学模型的 E-R 图。要求标注联系类型,可省略实体属性。
- (2) 把该 E-R 图转换为关系模式结构,并注明每个关系的主键和外键(如果存在)。

2. 设某工厂数据库情况如下。

仓库: 仓库号,仓库面积。

零件: 零件号,零件名,规格,单价。

供应商: 供应商号,供应商名,地址。

保管员: 职工号,职工名。

设仓库与零件之间有存放关系,每个仓库可以存放多个零件,每种零件可以存放在多个仓库,仓库存放时有库存量,每个供应商可供应多种零件,每种零件可由多个供应商供应,供应要考虑供应量,一个仓库有多个保管员,一个保管员只能在一个仓库工作。

- (1) 根据上述语义画出 E-R 图。要求标注联系类型,(可省略实体的属性)。
- (2) 将 E-R 图转换为关系模式,并注明每个关系的主键和外键(如果存在)。