第 2 章　关系数据库

【本章导读】

本章主要讲述关系模型的基本概念、关系的数据结构、数据操纵和完整性约束以及关系系统的定义和分类。

【本章要点】

- 关系模型的数据结构
- 并、交、差和笛卡儿积 4 种传统的集合运算
- 选择、投影、连接和除 4 种专门的关系运算
- 关系的实体完整性规则和参照完整性规则
- 关系系统的定义和分类

2.1　关系模型的基本概念

2.1.1　数学定义

1. 域

定义 2.1　域是一组具有相同数据类型的值的集合。

例如，整数、实数、字符串、{男, 女}，大于 0 小于等于 100 的正整数等都可以是域。

2. 笛卡儿积

定义 2.2　给定一组域 $D_1, D_2, \cdots, D_n$，$D_1, D_2, \cdots, D_n$ 的笛卡儿积为：

$$D_1 \times D_2 \times \cdots \times D_n = \{(d_1, d_2, \cdots d_n) | d_i \in D_i \ ,\ i=1,2, \cdots, n\}$$

其中，每一个元素$(d_1, d_2, \cdots, d_n)$叫作一个元组，元素中的每一个值 d_i 叫作一个分量。

【例 2-1】假设有两个域：

D_1=animal（动物集合）={猫,狗,猪}

D_2=food（食物集合）={鱼,骨头,白菜}

则：

$D_1 \times D_2$={(猫, 鱼) (狗, 鱼) (猪, 鱼) (猫, 骨头) (狗, 骨头) (猪, 骨头) (猫, 白菜) (猪, 白菜) (狗, 白菜)}。这 9 个元组可列成一张二维表，如表 2-1 所示。

3. 关系

定义 2.3　$D_1 \times D_2 \times \cdots \times D_n$ 的子集叫作在域 $D_1, D_2, \cdots, D_n$ 上的关系，用 $R(D_1, D_2, \cdots, D_n)$ 来表示。这里 R 表示关系的名字。

下面就从上例的笛卡儿积中取出一个子集来构造一个关系 Eat(animal, food)。关系名字为 Eat，属性名为 animal 和 food，如表 2-2 所示。

表 2-1　笛卡儿积结果

animal	food
猫	鱼
猫	骨头
猫	白菜
狗	鱼
狗	骨头
狗	白菜
猪	鱼
猪	骨头
猪	白菜

表 2-2　Eat 关系

animal	food
猫	鱼
狗	骨头
猪	白菜

4. 关系的性质

（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）不同的列可出自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。

（3）列的顺序无要求，即列的次序可以任意交换。

（4）任意两个元组不能完全相同。

（5）行的顺序无要求，即行的次序可以任意交换。

（6）分量必须取原子值，即每一个分量都必须是不可再分的数据项。

2.1.2　关系数据结构

在用户看来，一个关系模型的逻辑结构是一张二维表，它由行和列组成。例如，表 2-3 所示的学生记录就是一个关系模型，它涉及下列概念。

关系：一个关系对应一张二维表，表 2-3 所示的这张学生记录表就是一个关系。

元组：图中的一行称为一个元组，若表 2-3 中有 20 行，就有 20 个元组。

属性：图中的一列称为一个属性，表 2-3 中有 5 列，对应 5 个属性：学号、姓名、性别、年龄和所在系。

码：表中的某个属性（组），它可以唯一地确定一个元组，则称该属性组为“候选码”。若一个关系有多个候选码，则选定其中一个为主码。如表 2-3 所示的学号列，可以作为该学生关系的码来唯一标识一个学生的信息。

域：属性的取值范围。如表 2-3 所示的学生年龄的域应是(16～28)，性别的域是(男, 女)，系别的域是一个学校所有系名的集合。

分量：元组中的一个属性值。

关系模式：对关系的描述，一般表示为：

关系名(属性 1, 属性 2, ……, 属性 n)

表 2-3 所示的学生关系可描述为：学生(学号, 姓名, 性别, 年龄, 所在系)。

表 2-3 学生关系

学号	姓名	性别	年龄	所在系
000101	王萧	男	17	经济系
000207	李云虎	男	18	机械系
010302	郭敏	女	18	信息系
010408	高红	女	20	土木系
⋮	⋮	⋮	⋮	⋮
020309	王睿	男	19	信息系
020506	路旭青	女	21	管理系

2.2 关系代数和关系演算

2.2.1 传统的集合运算

传统的集合运算是二目运算，包括并、交、差和广义笛卡儿积 4 种运算。设关系 R 和关系 S 具有相同的目 n（即两个关系都具有 n 个属性），且相应的属性取自同一个域，则 4 种运算定义如下。

1. 并

关系 R 与关系 S 的并由属于 R 或属于 S 的元组组成，其结果关系仍为 n 目关系。记作 R∪S。

2. 交

关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成，其结果关系仍为 n 目关系。记作 R∩S。

3. 差

关系 R 与关系 S 的差由属于 R 而不属于 S 的所有元组组成。其结果关系仍为 n 目关系。记作 R−S。

4. 广义笛卡儿积

两个分别为 n 目和 m 目的关系 R 和 S 的广义笛卡儿积是一个（n+m）列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。若 R 有 A_1 个元组，S 有 A_2 个元组，则关系 R 和关系 S 的广义笛卡儿积有 $A_1 \times A_2$ 个元组，记作 R×S。

【例 2-2】有关系 R、S，如表 2-4（a）、（b）所示，则 R∪S、R∩S、R−S、R×S 的结果分别为表 2-4（c）、（d）、（e）、（f）所示。

表 2-4　传统的集合运算

R

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

（a）

S

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

（b）

R∪S

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

（c）

R∩S

a	b	c
1	2	3
7	8	9

（d）

R−S

a	b	c
4	5	6

（e）

R×S

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

（f）

2.2.2　专门的关系运算

专门的关系运算包括选择、投影、连接和除等。

1. 选择

选择是在关系 R 中选择满足给定条件的诸元组，记作：

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

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：

$X_1\theta Y_1[\Phi X_2\theta Y_2]\cdots$

θ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¬（非）、∧（与）及∨（或）。[]表示任选项，即[]中的部分可以要也可以不要，…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。

设有一个学生－课程关系数据库，包括学生关系 S、课程关系 C 和选修关系 SC，如表 2-5 所示。下面的例子将对这 3 个关系进行运算。

表 2-5　学生－课程关系数据库

S

学号 S#	姓名 SN	性别 SS	年龄 SA	所在系 SD
000101	李晨	男	18	信息系
000102	王博	女	19	数学系
010101	刘思思	女	18	信息系
010102	王国美	女	20	物理系
020101	范伟	男	19	数学系

C

课程号 C#	课程名 CN	学分 CC
1	数学	6
2	英语	4
3	计算机	4
4	制图	3

SC

学号 S#	课程号 C#	成绩 G
000101	1	90
000101	2	87
000101	3	72
010101	1	85
010101	2	42
020101	3	70

【例 2-3】查询数学系学生的信息。

$\sigma_{SD='数学系'}(S)$

或

$\sigma_{5='数学系'}(S)$

结果如表 2-6 所示。

【例 2-4】查询年龄小于 20 的学生的信息。

$\sigma_{SA<20}(S)$

或

$\sigma_{4<20}(S)$

结果如表 2-7 所示。

表 2-6　查询数学系学生的信息

学号 S#	姓名 SN	性别 SS	年龄 SA	所在系 SD
000102	王博	女	19	数学系
020101	范伟	男	19	数学系

表 2-7　查询年龄小于 20 的学生的信息

学号 S#	姓名 SN	性别 SS	年龄 SA	所在系 SD
000101	李晨	男	18	信息系
000102	王博	女	19	数学系
010101	刘思思	女	18	信息系
020101	范伟	男	19	数学系

2. 投影

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。

投影操作是从列的角度进行的运算。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。

【例 2-5】查询学生的学号和姓名。

$\pi_{S\#,SN}(S)$

或

$\pi_{1,2}(S)$

结果如表 2-8 所示。

表 2-8　查询学生的学号和姓名

学号 S#	姓名 SN
000101	李晨
000102	王博
010101	刘思思
010102	王国美
020101	范伟

【例 2-6】查询学生的所在系，即查询学生关系 S 在所在系属性上的投影。

$\pi_{SD}(S)$

或

$\pi_5(S)$

结果如表 2-9 所示。

表 2-9 查询结果

所在系 SD
信息系
数学系
物理系

3. 连接

连接也称为θ连接。它是从两个关系的笛卡儿积中选取属性间满足一定条件的元组。记作：

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

其中，A 和 B 分别为 R 和 S 上度数相等且可比的属性组，θ是比较运算符。连接运算从 R 和 S 的笛卡儿积 R×S 中选取（R 关系）在 A 属性组上的值与（S 关系）在 B 属性组上的值满足比较关系θ的元组。

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。等值连接可记作：

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

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。这种在相同的属性组间进行比较并去掉了重复的属性的等值连接称为自然连接。自然连接可记作：

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

一般的连接操作是从行的角度进行运算，自然连接还需要取消重复列，所以是同时从行和列的角度进行运算。

【例 2-7】设关系 R、S 分别如表 2-10（a）、（b）所示，则 $R \underset{C<D}{\bowtie} S$ 的结果如表 2-11（a）所示，等值连接 $R \underset{C=D}{\bowtie} S$ 的结果如表 2-11（b）所示。

表 2-10 关系表 R 和 S

R

A	B	C
1	2	3
4	5	6
7	3	0

（a）

S

D	E
3	1
6	2

（b）

表 2-11 R 和 S 的运算结果

$R \underset{C<D}{\bowtie} S$

A	B	C	D	E
1	2	3	6	2
7	3	0	3	1
7	3	0	6	2

（a）

$R \underset{C=D}{\bowtie} S$

A	B	C	D	E
1	2	3	3	1
4	5	6	6	2

（b）

若 R 和 S 有相同的属性组 C，如表 2-12（a）、（b）所示，自然连接的结果如表 2-12（c）所示。

表 2-12　关系表 R 和 S

R

A	B	C
1	2	3
4	5	6
7	3	0

（a）

S

C	E
3	1
6	2

（b）

$R \bowtie S$

A	B	C	E
1	2	3	1
4	5	6	2

（c）

4. 除

R 与 S 的除运算得到一个新的关系 P(X)，P 是 R 中满足下列条件的元组在 X 属性列上的投影。

（1）关系 R(X, Y)和 S(Y, Z)，其中 X、Y、Z 为属性组（R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集）。

（2）元组在 X 上分量值 x 的象集 Yx 包含 S 在 Y 上的投影。

除运算可以记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_y(S) \subseteq Yx\}, x = t_r[X]\}$$

除操作是同时从行和列角度进行运算的。

【例 2-8】R(A, B, C)和 S(B, C, D)两个关系如表 2-13 所示，求 R÷S。

表 2-13　关系表 R 和 S

R

A	B	C
a1	b1	c2
a2	b3	c7
a3	b4	c6
a1	b2	c3
a4	b6	c6
a2	b2	c3
a1	b2	c1

（a）

S

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

（b）

则 R÷S 运算如下：

a1 的象集为 {(b1,c2), (b2,c3), (b2,c1)}

a2 的象集为 {(b3,c7), (b2,c3)}

a3 的象集为 {(b4,c6)}

a4 的象集为 {(b6,c6)}

S 在(B, C)上的投影为：{(b1,c2), (b2,c1), (b2,c3)}

因只有 a1 的象集包含了 S 在(B, C)属性组上的投影，故

R÷S={a1}

结果如表 2-14 所示。

表 2-14 除运算结果

A
a1

可以使用选择、投影、连接和除 4 种操作来编辑复杂的查询。以前面的学生－课程数据库中的 3 个表为例，举例说明它们的综合用法。

【例 2-9】查询选修了 2 号课程的学生的学号。

$\pi_{S\#}(\sigma_{C\#='2'}(SC))$

【例 2-10】查询选修了 3 号课程的学生的姓名。

$\pi_{SN}(\sigma_{C\#='3'}(SC \bowtie S))$

【例 2-11】查询选修了数学课的学生的姓名和成绩。

$\pi_{SN,G}(\sigma_{CN='数学'}(C \bowtie SC \bowtie S))$

2.2.3 关系演算

关系演算是以数理逻辑中的谓词演算为基础的。按谓词变元的不同，关系演算可分为元组关系演算和域关系演算。

2.2.3.1 元组关系演算语言——ALPHA

元组关系演算以元组变量作为谓词变元的基本对象。一种典型的元组关系演算语言是 E.F.Codd 提出的 ALPHA 语言，这一语言虽然没有实际实现，但关系数据库管理系统 INGRES 所用的 QUEL 语言是参照 ALPHA 语言研制的，与 ALPHA 十分类似。

ALPHA 语言语句的基本格式如下：

操作语句 工作空间名(表达式):操作条件

其中操作语句主要有 GET、PUT、HOLD、UPDATE、DELETE 和 DROP 六条语句。表达式用于指定语句的操作对象，它可以是关系名或属性名，一条语句可以同时操作多个关系或多个属性。操作条件是一个关系或逻辑表达式，用于将操作对象限定在满足条件的元组中，操作条件可以为空。除此之外，还可以在基本格式的基础上加上排序要求、定额要求等。

这里仍以前面的学生—课程数据库中的 3 个表为例来介绍 ALPHA 语言的各种操作。

1. 检索操作

检索操作用 GET 语句实现。

（1）简单检索（即不带条件的检索）。

【例 2-12】查询所有学生的姓名。

```
GET  W(S.SN)
```

【例 2-13】查询所有学生的信息。

```
GET  W(S)
```

（2）带条件的检索。

【例 2-14】查询信息系学生的学号和年龄。

```
GET  W(S.S#, S.SA):S.SD='信息系'
```

【例 2-15】查询数学系年龄小于 20 的学生的姓名和年龄。

```
GET  W(S.SN, S.SA):S.SD='数学系'∧S.SA<20
```

（3）带排序的检索。

【例 2-16】查询计算机科学系学生的学号、姓名，并按年龄降序排序。

```
GET  W(S.S#, S.SN):S.SD='计算机科学系' DOWN S.SA
```

（4）指定元组个数的检索

【例 2-17】取出一个数学系学生的姓名。

```
GET  W(1)(S.SN):S.SD='数学系'
```

【例 2-18】查询信息系年龄最大的 3 个学生的学号及其年龄。

```
GET  W(3)(S.S#, S.SA):S.SD='信息系' DOWN S.SA
```

2. 更新操作

（1）插入操作。插入操作用 PUT 语句实现，其步骤如下：

1）用宿主语言在工作空间中建立新元组。

2）用 PUT 语句把该元组存入指定的关系中。

【例 2-19】插入一学号为 020302、姓名为刘青的 18 岁女生到计算机系。

```
MOVE  020302  TO  W.S#
MOVE  '刘青' TO W.SN
MOVE  '女'  TO  W.SS
MOVE  18  TO  W.SA
MOVE  '计算机系'  TO  W.SD
PUT W(S)
```

（2）删除操作。删除操作用 DELETE 语句实现。其步骤如下：

1）用 HOLD 语句把要删除的元组从数据库中读到工作空间中。

2）用 DELETE 语句删除该元组。

【例 2-20】删除学号为 020302 的学生。

```
HOLD  W(S):S.S#='020302'
DELETE  W
```

【例 2-21】删除全部学生。

```
HOLD  W(S)
DELETE  W
```

（3）修改操作。修改操作用 UPDATE 语句实现，其步骤如下：

1）用 HOLD 语句将要修改的元组从数据库中读到工作空间中。

2）用宿主语言修改工作空间中元组的属性。

3）用 UPDATE 语句将修改后的元组送回数据库中。

【例 2-22】将 020101 的姓名改为孟伟。

```
HOLD  W(S.S#, S.SN):S.S#='020101'
MOVE  '孟伟' TO  W.SN
UPDATE  W
```

修改主码的操作是不允许的，如果需要修改关系中某个元组的主码值，只能先用删除操作删除该元组，然后再把具有新主码值的元组插入到关系中。

【例 2-23】将 020101 的学号改为 030201。

```
HOLD  W  (S):S.S#='020101'
DELETE  W
MOVE  '030201'  TO  W.S#
MOVE  '孟伟'  TO  W.S N
MOVE  '男'  TO  W.SS
MOVE  '19'  TO  W.SA
MOVE  '数学系'  TO  W.SD
PUT  W  (S)
```

2.2.3.2 域关系演算语言 QBE

域关系演算以元组变量的分量，即域变量作为谓词变元的基本对象。1971 年由 M.M.Zloof 提出的 QBE 就是一个很有特色的域关系演算语言，该语言于 1978 年在 IBM 370 上得以实现。

QBE（Query By Example）用示例元素来表示查询结果可能的例子。示例元素实质上就是域变量。其最突出的特点是它的操作方式：它是一种高度非过程化的基于屏幕表格的查询语言。用户通过终端屏幕编辑程序以填写表格的方式构造查询要求，而查询结果也是以表格形式显示。因此非常直观，易学易用。

下面仍以学生—课程数据库为例，说明 QBE 的用法。

1. 检索操作

（1）简单查询。

【例 2-24】查询全体学生的姓名。

操作步骤如下：

1）用户提出要求。

2）屏幕显示空白表格，如表 2-15 所示。

表 2-15 显示空白表格

3）用户在最左边一栏输入关系名，如表 2-16 所示。

表 2-16 输入关系名

S			

4）显示该关系的栏名，即 S 的各个属性名，如表 2-17 所示。

表 2-17　显示关系的属性

S	S#	SN	SS	SA	SD

5）用户构造查询要求，如表 2-18 所示。

表 2-18　构造查询要求

S	S#	SN	SS	SA	SD
		P. T			

这里 T 是示例元素，即域变量。QBE 要求示例元素下面一定要加下划线。“P.”是操作符，表示打印（Print），实际上就是显示。

示例元素是这个域中可能的一个值，它不必是查询结果中的元素。比如要求查询信息系学生的姓名，只要给出任意的一个学生名即可，而不必是信息系的某个学生名。

6）屏幕显示查询结果，如表 2-19 所示。

表 2-19　查询结果

S	S#	SN	SS	SA	SD
		李晨 王博 刘思思 王国美			

【例 2-25】查询全体学生的信息，如表 2-20 所示。

表 2-20　查询全体学生的信息

S	S#	SN	SS	SA	SD
	P.000101	P.李晨	P.男	P.18	P.信息系

显示全部数据也可以简单地把“P.”操作符作用在关系名上。因此本查询也可以简单地表示为如表 2-21 所示的形式。

表 2-21　简化查询

S	S#	SN	SS	SA	SD
P.					

（2）条件查询。

【例 2-26】求信息系全体学生的姓名，如表 2-22 所示。

表 2-22　查询信息系全体学生的姓名

S	S#	SN	SS	SA	SD
		P.李晨			信息系

信息系是查询条件，不必加横线。

【例 2-27】求数学系年龄大于19岁的学生的学号。

本查询是两个条件的“与”。在 QBE 中，有以下两种表示方法。

1）把两个条件写在同一行上，如表 2-23 所示。

表 2-23 查询数学系年龄大于 19 岁的学生的学号

S	S#	SN	SS	SA	SD
	P.000101			>19	数学系

2）把两个条件写在不同行上，但使用相同的示例元素值，如表 2-24 所示。

表 2-24 查询数学系年龄大于 19 岁的学生的学号

S	S#	SN	SS	SA	SD
	P.000101 P.000101			>19	 数学系

【例 2-28】查询数学系或者年龄大于 19 岁的学生的学号。

本查询是两个条件的“或”。在 QBE 中把两个条件写在不同行上，并且使用不同的示例元素值来表示条件的“或”，如表 2-25 所示。

表 2-25 查询数学系或者年龄大于 19 岁的学生的学号

S	S#	SN	SS	SA	SD
	P.000101 P.000102			>19	 数学系

（3）查询结果排序。

对查询结果按某个属性值的升序排序，只需在相应列中填入“AO.”，按降序排序则填“DO.”。如果按多列排序，用“AO(i).”或“DO(i).”表示，其中 i 为排序的优先级，i 值越小，优先级越高。

【例 2-29】查询信息系学生的姓名，要求查询结果按年龄升序排序，对年龄相同的学生按性别降序排序，如表 2-26 所示。

表 2-26 对查询结果排序

S	S#	SN	SS	SA	SD
		P.李晨	DO(2)	AO(1)	信息系

2. 更新操作

（1）插入操作。

插入操作符为“I.”，新插入的元组必须具有码值，其他属性值可以为空。

【例 2-30】把学号为 000103，姓名张兰，年龄 17 岁的信息系女生插入表 S，如表 2-27 所示。

表 2-27　插入操作

S	S#	SN	SS	SA	SD
I.	000103	张兰	女	17	信息系

（2）删除操作。

删除操作符为“D.”。

【例 2-31】删除学号为 000103 的学生，如表 2-28 所示。

表 2-28　删除操作

S	S#	SN	SS	SA	SD
D.	000103				

（3）修改操作。

修改操作符为“U.”。关系的主码不允许修改，如果需要修改某个元组的主码，需首先删除该元组，然后再插入新的主码的元组。

【例 2-32】把 000101 的年龄改为 19 岁，如表 2-29 所示。

表 2-29　修改操作

S	S#	SN	SS	SA	SD
	000101			U.19	

（a）

或：

S	S#	SN	SS	SA	SD
U.	000101			19	

（b）

【例 2-33】把所有学生的年龄增加 1 岁，如表 2-30 所示。

表 2-30　把所有学生的年龄增加 1 岁

S	S#	SN	SS	SA	SD
U.	000101 000101			X X+1	

2.3　关系的完整性

关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系系统自动支持。

2.3.1 实体完整性

规则 2.1 实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

例如，在学生关系 S(S#, SN, SS, SA, SD)中，S#属性为主码，则 S#不能取空值。

实体完整性规则规定，基本关系的所有主属性都不能取空值，而不仅是主码整体不能取空值。例如，学生选课关系 SC(S#, C#, G)，(S#, C#)为主码，则 S#和 C#两属性都不能取空值。

2.3.2 参照完整性

现实世界中的实体之间往往存在某种联系，在关系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系间的引用。先来看一个例子。

学生—课程关系数据库，包括学生关系 S、课程关系 C 和选修关系 SC，这三个关系分别为：

学生(学号, 姓名, 性别, 年龄, 所在系)

课程(课程号, 课程名, 学分)

选修(学号, 课程号, 成绩)

其中，添加下划线的属性为该关系的主码。

这三个关系之间存在着属性的引用，即选修关系引用了学生关系的主码“学号”和课程关系的主码“课程号”。显然，选修关系中的学号值必须是确实存在的学生的学号，即学生关系中有该学生的记录。选修关系中的课程号值也必须是确实存在的课程的课程号，即课程关系中有该课程的记录。换句话说，选修关系中某些属性的取值需要参照其他关系的属性取值。

不仅两个或两个以上的关系间可以存在引用关系，同一关系内部属性间也可能存在引用关系。

定义 2.4 设 F 是关系 R 的一个或一组属性，但不是关系 R 的码，如果 F 与关系 S 的主码 Ks 相对应，则称 F 是基本关系 R 的外码（Foreign Key），并称关系 R 为参照关系，关系 S 为被参照关系。

显然，被参照关系 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。

在例子中，选修关系的“学号”属性与学生关系的主码“学号”相对应，因此“学号”属性是选修关系的外码。学生关系为被参照关系，选修关系为参照关系。选修关系的“课程号”属性与课程关系的主码“课程号”相对应，因此“课程号”属性也是选修关系的外码。课程关系为被参照关系，选修关系为参照关系。

参照完整性规则就是定义外码与主码之间的引用规则。

规则 2.2 参照完整性规则：若属性（或属性组）F 是关系 R 的外码，它与关系 S 的主码 Ks 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须为：

（1）或者取空值（F 的每个属性值均为空值）。

（2）或者等于 S 中某个元组的主码值。

对于例子中选修关系中每个元组的学号属性只能取下面两类值：

（1）空值，表示尚未有学生选课。

（2）非空值，这时该值必须是学生关系中某个学生学号，表示某个未知的学生不能选课。

同样，选修关系中每个元组的课程号属性只能取下面两类值：

（1）空值，表示尚未开课。

（2）非空值，这时该值必须是课程关系中某个课程号，表示不能选未开设的课。

2.3.3　用户定义的完整性

实体完整性和参照完整性适用于任何关系数据库系统。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。例如，学生关系的年龄在 15～30 之间；选修关系的成绩必须在 0～100 之间等。

2.4　关系系统

2.4.1　关系系统的定义

一个系统可以定义为关系系统，当且仅当它支持：

（1）关系数据结构。也就是说，从用户观点看，数据库是由表构成的，并且系统中只有表这种结构。

（2）支持选择、投影和（自然）连接运算。对这些运算不要求用户定义任何物理存取路径。

关系模型中并非每一部分都是同等重要的，所以并不苛求一个实际的关系系统必须完全支持关系模型。

不支持关系数据结构的系统显然不能称为关系系统。

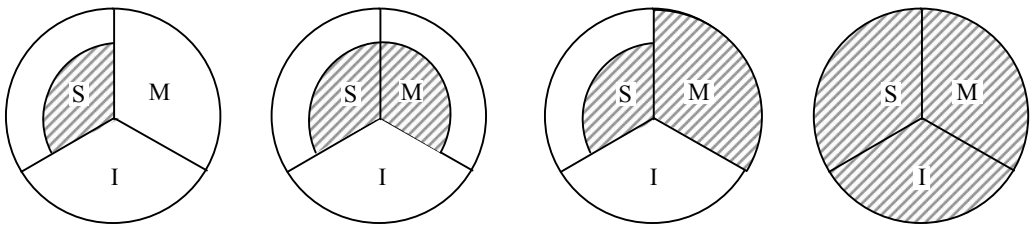
仅支持关系数据结构，但没有选择、投影和连接运算功能的系统，用户使用起来仍不方便，这种系统仍不能算作关系系统。

支持选择、投影和连接运算，但要求定义物理存取路径，这样就降低或丧失了数据的物理独立性，这种系统也不能算作真正的关系系统。

选择、投影、连接运算是最有用的运算，能解决绝大部分实际问题，所以要求关系系统只要支持这 3 种最主要的运算即可，并不要求它必须提供关系代数的全部运算功能。

2.4.2　关系系统的分类

按照 E.F.Codd 的思想，依据关系系统支持关系模型的程度不同，可以把关系系统分为 4 类，如图 2-1 所示。



（a）表式系统　（b）（最小）关系系统　（c）关系完备系统　（d）全关系系统

图 2-1　关系系统分类

图中的圆表示关系数据模型。每个圆分为三部分，分别表示模型的三个组成部分：S 表示数据结构（Structure），M 表示数据操纵（Manipulation），I 表示完整性约束（Integrity）。图中的阴影部分表示各类系统支持模型的程度。

1．表式系统

这类系统仅支持关系数据结构（即表），不支持集合级的操作。表式系统实际上不能算作关系系统。倒排表列（Inverted List）系统就属于这一类系统。

2.（最小）关系系统

这类系统是上面定义的关系系统，它支持关系数据结构和选择、投影、连接三种关系操作。许多微机关系系统如 FoxBase、FoxPro 等就属于这一类系统。

3．关系完备系统

这类系统支持关系数据结构和所有的关系代数操作（功能上与关系代数等价）。目前许多大、中型关系系统如 DB2、Oracle 等就属于这一类系统。

4．全关系系统

这类系统支持关系模型的所有特征，特别是数据结构中域的概念、实体完整性和参照完整性。虽然 DB2、Oracle 等系统已经接近这个目标，但到目前为止尚没有一个系统是全关系系统。

本章小结

一组具有相同数据类型的值的集合称为域。

D1×D2×…×Dn={(d1,d2,…,dn)|di∈Di，i=1,2,…,n}称为域 D1,D2,…,Dn，D1,D2,…,Dn 上的笛卡儿积。其中每一个元素(d1,d2,…,dn)叫作一个元组，元素中的每一个值 di 叫作一个分量。

D1×D2×…×Dn 的子集叫作在域 D1,D2,…,Dn 上的关系，用 R(D1,D2,…,Dn)来表示。

关系的数据结构就是一张二维表。

关系代数包括传统的集合运算和专门的关系运算。传统的集合运算有：并、交、差和广义笛卡儿积；专门的关系运算有：选择、投影、连接和除。关系演算是以数理逻辑中的谓词演算为基础的。按谓词变元的不同，关系演算可分为元组关系演算和域关系演算。

关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有三类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应该由关系系统自动支持。

一个系统可以定义为关系系统，当且仅当它支持：①关系数据结构；②支持选择、投影和（自然）连接运算。对这些运算不要求用户定义任何物理存取路径。按照 E.F.Codd 的思想，依据关系系统支持关系模型的程度不同，可以把关系系统分为 4 类：表式系统、（最小）关系系统、关系完备的系统和全关系系统。

习题二

一、填空题

1．关系数据模型中，实体及实体间的联系都用_____来表示。在数据库的物理组织

中，它以_____形式存储。

2．常用的关系操作有两类：传统的集合操作，如并、交、差和_____；专门的关系操作，如_____、_____、_____和除等。

3．关系数据库的完整性约束包括_____、_____和_____三类。

二、操作题

有以下的 4 个关系：

S（供应商）：

SNO （供应商号）	SNAME （供应商姓名）	CITY （供应商所在城市）
S1	精益	天津
S2	万胜	北京
S3	东方	北京
S4	丰泰隆	上海
S5	康健	南京

P（零件）：

PNO （零件号）	PNAME （零件名称）	COLOR （零件颜色）	WEIGHT （零件重量）
P1	螺母	红	12
P2	螺栓	绿	17
P3	螺丝刀	蓝	14
P4	螺丝刀	红	14
P5	凸轮	蓝	40

J（项目）：

JNO （项目号）	JNAME （项目名称）	CITY （项目所在城市）
J1	三建	北京
J2	一汽	长春
J3	弹簧厂	天津
J4	造船厂	天津
J5	机车厂	唐山
J6	无线电厂	常州

SPJ（供应情况）：

SNO （供应商号）	PNO （零件号）	JNO （项目号）	QTY （供应数量）
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400
S2	P5	J1	400
S2	P5	J2	100
S3	P1	J1	200
S3	P3	J1	200
S4	P5	J1	100
S5	P6	J2	200
S5	P6	J4	500

试用关系代数完成下列操作：

1．求供应商供应的商品的零件号。

2．求供应商 S5 供应的商品的零件号。

3．求供应工程 J1 零件的供应商号。

4．求供应工程 J1 零件 P1 的供应商号。

5．求供应工程 J1 红色零件的供应商号。

三、简答题

1．关系模型的完整性规则有哪几类？在关系模型的参照完整性规则中，外部码属性的值是否可以为空？什么情况下才可以为空？

2．关系系统可以分为哪几类？各类关系系统的定义是什么？