

第 3 章

关系代数

关系模型的重要部分是关系操纵。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,它是用对关系的运算来表达查询的。利用关系代数可以演示一个查询语言从关系数据库系统中检索信息的潜力,可以用最简单的形式来表达所有关系数据库查询语言必须完成的运算的集合,这些基本的运算对解释标准查询语言 SQL 如何被执行很有帮助,同时也有利于培养关系运算的思维能力。本章主要包括以下两方面的内容。

- (1) 集合运算:包括并、交、差、笛卡儿积。
- (2) 专门的关系操作:包括选择、投影、自然连接、 θ 连接、等值连接等。

3.1 概述

关系代数的运算对象是关系,运算结果也为关系。关系代数用到的运算符包括四类:集合运算符、专门的关系运算符、算术比较符和逻辑运算符。关系代数的运算按运算符的不同可分为传统的集合运算和专门的关系运算两类。其中传统的集合运算将关系看成元组的集合,其运算是从关系的“水平”方向即行的角度来进行。而专门的关系运算不仅涉及行而且涉及列。比较运算符和逻辑运算符是用来辅助专门的关系运算符进行操作的。常见的关系代数运算符如表 3-1 所示。

表 3-1 常见的关系代数运算符

运算符	符号	含义	键盘格式	示 例
集合运算符	\cup	并	UNION	$R \cup S$, 或 $R \text{ UNION } S$
	\cap	交	INTERSECT	$R \cap S$, 或 $R \text{ INTERSECT } S$
	$-$	差	MINUS	$R - S$, 或 $R \text{ MINUS } S$
	\times	乘	TIMES	$R \times S$, 或 $R \text{ TIMES } S$
专门关系运算符	σ	选择	$R \text{ where } C$	$\sigma \text{ 姓名} = \text{'张三'}(S)$ 或 $S \text{ where 姓名} = \text{'张三'}$
	π	投影	$R[\]$	$\pi \text{ 考号, 姓名}(S)$ 或 $S[\text{考号, 姓名}]$
	∞	连接	JOIN	$R \infty S$, 或 $R \text{ JOIN } S$
	\div	除	DIVIDEBY	$R \div S$, 或 $R \text{ DIVIDEBY } S$

3.2 关系中的集合操作

四种最常用的集合操作是：并、交、差和广义笛卡儿积。

定义 3-1 设关系 R 和关系 S 具有相同的目 n (即两个关系都有 n 个属性), 且相应的属性取自同一个域, 则可以定义并、差、交运算如下。

(1) 并(Union)。关系 R 与关系 S 的并记作: $R \cup S = \{t | t \in R \vee t \in S\}$, 其结果仍为 n 目关系, 由属于 R 或属于 S 的元组组成。

(2) 差(Difference)。关系 R 与关系 S 的差记作: $R - S = \{t | t \in R \wedge t \notin S\}$, 其结果仍为 n 目关系, 由属于 R 而不属于 S 的所有元组组成。

(3) 交(Intersection)。关系 R 与关系 S 的交记作: $R \cap S = \{t | t \in R \wedge t \in S\}$, 其结果仍为 n 目关系, 由既属于 R 又属于 S 的元组组成。关系的交可以用差来表示, 即 $R \cap S = R - (R - S)$ 。

(4) 广义笛卡儿积(Extended Cartesian Product)。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 k_1 个元组, S 有 k_2 个元组, 则关系 R 和关系 S 的广义笛卡儿积有 $k_1 \times k_2$ 个元组。记作: $R \times S = \{trts | tr \in R \wedge ts \in S\}$ 。

当在关系上应用这些操作的时候, 需要对关系 R 和关系 S 附加下列条件:

(1) R 和 S 必须具有同样属性集合的表, 同时 R 和 S 的各个属性的类型(域)也必须匹配。

(2) 在做相应的集合操作(并、交、差、广义笛卡儿积)之前, R 和 S 的列必须经过排序, 从而保证他们的属性顺序对于两个关系来说完全相同。

3.3 传统的集合操作综合实例

现有关系 R 和关系 S 如表 3-2 和表 3-3 所示。 $R \cup S$ 的结果如表 3-4 所示。 $R \cap S$ 的结果如表 3-5 所示。 $R - S$ 的结果如表 3-6 所示。 $R \times S$ 的结果如表 3-7 所示。

表 3-2 关系 R

A	B	C
A ₁	B ₁	C ₁
A ₁	B ₂	C ₂
A ₂	B ₂	C ₁

表 3-3 关系 S

A	B	C
A ₁	B ₂	C ₂
A ₁	B ₃	C ₂
A ₂	B ₂	C ₁

表 3-4 $R \cup S$ 的结果

A	B	C
A_1	B_1	C_1
A_1	B_2	C_2
A_2	B_2	C_1
A_1	B_3	C_2

表 3-5 $R \cap S$ 的结果

A	B	C
A_1	B_2	C_2
A_2	B_2	C_1

表 3-6 $R-S$ 的结果

A	B	C
A_1	B_1	C_1

表 3-7 $R \times S$ 的结果

A	B	C	A	B	C
A_1	B_1	C_1	A_1	B_2	C_2
A_1	B_1	C_1	A_1	B_3	C_2
A_1	B_1	C_1	A_2	B_2	C_1
A_1	B_2	C_2	A_1	B_2	C_2
A_1	B_2	C_2	A_1	B_3	C_2
A_1	B_2	C_2	A_2	B_2	C_1
A_2	B_2	C_1	A_1	B_2	C_2
A_2	B_2	C_1	A_1	B_3	C_2
A_2	B_2	C_1	A_2	B_2	C_1

3.4 专门的关系运算

专门的关系运算包括选择、投影、连接、除等。为了叙述方便,先引入几个常用记号。设关系模式为 $R(A_1, A_2, \dots, A_n)$, 它的一个关系设为 R 。 $t \in R$ 表示 t 是 R 的一个元组。 $t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量。

若 $A = \{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$, 其中 $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ 是 A_1, A_2, \dots, A_n 中的一部分, 则 A 称为属性列或域列。

R 为 n 目关系, S 为 m 目关系。 $tr \in R, ts \in S, trts$ 称为元组的连接, 是一个 $n+m$ 列的元组, 前 n 个分量为 R 中的一个 n 元组, 后 m 个分量为 S 中的一个 m 元组。

给定一个关系 $R(X, Z)$, X 和 Z 为属性组。定义 $t[X] = x$ 时, x 在 R 中的像集

(Images Set)为: $Z_x = \{t[Z] \mid t \in R, t[X] = x\}$, 表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合。

3.4.1 选择

定义 3-2 选择(Selection)又称为限制(Restriction),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,是一个逻辑表达式,取逻辑值“真”或“假”。逻辑表达式 F 由逻辑运算符 \neg 、 \wedge 、 \vee 连接各算术表达式组成。算术表达式的基本形式为 $X1\theta Y1$ 。其中, θ 表示比较运算符,可以是 $>$ 、 \geq 、 $<$ 、 \leq 、 $=$ 或 \neq ; $X1$ 、 $Y1$ 等是属性名,或为常量,或为简单函数;属性名也可以用它的序号来代替。

提示: 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,是对行的水平分解。

学生课程数据库的关系如表 3-8~表 3-10 所示。

表 3-8 学生表

学号	姓名	性别	年龄	所在系
200501	李苛	男	19	CS
200502	张峰	男	20	IS
200503	王兰	女	19	MA
200504	李华	男	18	CS

表 3-9 课程表

课程号	课程名	学分
1	数据库	3
2	数学	4
3	英语	3
4	操作系统	3

表 3-10 学生选课表

学号	课程号	成绩
200501	1	92
200502	2	85
200503	3	88
200504	2	90
200504	4	73

例 3-1 查询信息系(IS)的全体学生。

$\sigma_{\text{所在系} = \text{'IS'}}(\text{学生表})$ 或 $\sigma_5 = \text{'IS'}(\text{学生表})$

其中,下角标“5”为所在系的属性序号。或 σ_{where} 所属系 = “IS”,以上查询产生的关系如表 3-11 所示。

表 3-11 信息系的全体学生

学号	姓名	性别	年龄	所在系
200502	张峰	男	20	IS

例 3-2 查询年龄在 19~20 岁的学生。

$\sigma_{\text{年龄} \geq 19 \wedge \text{年龄} \leq 20}$ (学生表) 或 $\sigma_{4 \geq 19 \wedge 4 \leq 20}$ (学生表) 或 σ_{where} 年龄 ≥ 19 和 年龄 ≤ 20

以上查询产生的关系如表 3-12 所示。

表 3-12 年龄在 19~20 岁的学生

学号	姓名	性别	年龄	所在系
200501	李苛	男	19	CS
200502	张峰	男	20	IS
200503	王兰	女	19	MA

3.4.2 投影

定义 3-3 投影(Projection)操作用来从关系 R 生成一个新的关系,这个关系只包含原来关系 R 的部分列。表达式 $\pi_{A_1, A_2, \dots, A_n}(R)$ 的值只包含关系 R 属性 A_1, A_2, \dots, A_n 所代表的列。结果关系模式的属性集合为: $\{A_1, A_2, \dots, A_n\}$, 习惯上按所列出的顺序显示。其中, A 为 R 中的属性列集合。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 3-3 查询学生的姓名和所在系。

$\pi_{\text{姓名}, \text{所在系}}$ (学生表) 或 $\pi_{2, 5}$ (学生表)

其中, 2 表示第 2 列, 5 表示第 5 列。

例 3-4 查询学生关系中都有哪些系。

$\pi_{\text{所在系}}$ (学生表) 或 π_5 (学生表)

3.4.3 笛卡儿积

关系 R 和 S 的笛卡儿积(或者称为叉积或者称为积)是一个有序对的集合。有序对的第一个元素是关系 R 中的任何一个元组,每两个元素是关系 S 中的任何一个元组,表示为 $R \times S$ 。当 R 和 S 都是关系的时候,积本质上仍是关系,但是由于 R 和 S 这两个关系的属性未必只是一个,所产生的结果产生了更长的元组,包括了 R 和 S 中的所有属性。习惯上,在结果中关系 R 的属性出现在关系 S 属性的前面。

结果关系模式是 R 和 S 关系模式的并,但是如果 R 和 S 恰好有相同的属性,就需要

把至少一个属性名更改成不同的名称。为了使意义清楚,如果属性 A 在 R 和 S 中均出现,则结果关系模式中分别用 R. A 和 S. A 表示来自于 R 和 S 的属性。

例 3-5 设关系 R 如表 3-13,关系 S 如表 3-14 所示,则 $R \times S$ 的结果如表 3-15 所示。

表 3-13 关系 R

A	B
1	2
3	4

表 3-14 关系 S

B	C	D
2	5	6
4	7	8
9	10	11

表 3-15 $R \times S$ 的结果

A	R. B	S. B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

3.4.4 自然连接

与笛卡儿积相比,人们经常对两个关系做连接(Join)操作,连接时相应的元组必须在某些方面一致,最简单的就是所谓的自然连接(Nature Join)。关系 R 和 S 的自然连接表示为 $R \bowtie S$ 。它要求两个关系中进行比较的分量必须是相同的属性组,并且在结果中把重复的属性列去掉。如果假设 R 和 S 的模式有公共属性 A_1, A_2, \dots, A_n , r 和 s 是分别来自于关系 R 和 S 的元组,则当且仅当 r 和 s 的 A_1, A_2, \dots, A_n 属性一样时, r 和 s 才能配对,作为结果关系中的元组。

例 3-6 表 3-13 中的关系 R 与表 3-14 中的关系 S 自然连接的结果如表 3-16 所示。

表 3-16 $R \bowtie S$ 的结果

A	B	C	D
1	2	5	6
3	4	7	8

唯一一个 R 和 S 共同的属性 B。只要在属性 B 上相同的元组就可以成功连接。如

果这样,结果元组中就有属性 A(来自 R),B(来自 S 或者 R),C(来自 S),D(来自 S)。

在这个例子中,R 的第一个元组成功地和 S 的第一个元组组成一对,它们在属性 B 上有共同的值 2。这个配对产生了连接结果的第一个元组:(1,2,5,6)。关系 R 的第二个元组只可以跟 S 中的第二个元组配对,结果是(3,4,7,8)。注意,关系 S 的第三个元组不能跟关系 R 的任何一个元组配对,所以不在结果 $R \bowtie S$ 中出现。这一个连接当中,如果一个元组不能和另外关系中的任何一个元组配对的话,这个元组就称为悬挂元组(dangling tuple)。

例 3-7 表 3-17 中的关系 U 与表 3-18 中的关系 V 有两个共同的属性 B 和 C,关系 U 与关系 V 自然连接后产生的结果如表 3-19 所示。

表 3-17 关系 U

A	B	C
1	2	3
6	7	8
9	7	8

表 3-18 关系 V

B	C	D
2	3	4
2	3	5
7	8	10

表 3-19 $U \bowtie V$ 的结果

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

提示: 只有在属性 B 和属性 C 上一致的元组才能配对成功。这样,U 的每一个元组可以和 V 的第一和第二个元组相连接,而 U 的第二和第三个元组与 V 的第三个元组配对。

3.4.5 θ 连接

自然连接必须根据某些特定的条件来将元组配对,虽然相等的公共属性是关系连接最常见的基础,但有时需将满足其他条件的元组配对。为此,就有了 θ 连接操作(theta-join)。 θ 是指任务条件,但现在一般用 C 而不是用 θ 表示这个条件。

关系 R 和关系 S 的 θ 连接可以用符号 $R \bowtie_C S$ 来表示。这个操作的结果构成如下:

- (1) 先得到 R 和 S 的积。
- (2) 在得到的结果中找到满足条件 C 的元组。

就像在积操作中一样,结果关系的模式是模式 R 和模式 S 的并。如果有必要的话,需要在重名的属性前面加上“R.”或“S.”。

例 3-8 关系 U 和关系 V 如表 3-17 和表 3-18 所示,求 $U_{A<D}^{\infty} V$ 的结果。

在这里要考虑 9 个元组的配对方案。看一看来自 U 的元组的属性 A 是不是比来自 V 的元组的属性 D 的值小。U 的第一个属性值是 1,这个元组可以和任何来自 V 的元组配对。但是第二和第三个元组的 A 属性分别是 6 和 9,分别只能和 V 的最后一个元组配对。这样,所得到的结果关系就只有 5 个元组,这个关系的结果如表 3-20 所示。

表 3-20 关系 U 和关系 V θ 连接的结果

A	U. B	U. C	V. B	V. C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

连接结果关系模式是由所有的 6 个属性组成。公共属性 B 和 C 前面加了关系名 U 和 V 以示区别。 θ 连接和自然连接相比,自然连接把公共属性连接成一个属性,而 θ 连接并不去除公共属性。

例 3-9 本例为更为复杂的 θ 连接, $U_{A<D \wedge U. B \neq V. B}^{\infty} V$, 结果不仅要求 U 关系元组的 A 属性小于关系元组的 D 属性,而且 U 和 V 元组的 B 属性不能有同样的值,所以结果只有一个,如表 3-21 所示。

表 3-21 更为复杂的 θ 连接结果

A	U. B	U. C	V. B	V. C	D
1	2	3	7	8	10

3.4.6 等值连接

θ 为“=”的连接运算称为等值连接。它是从关系 R 与 S 的广义笛卡儿积中选取属性值相等的那些元组。

例 3-10 关系 U 和关系 V 如表 3-17 和表 3-18 所示,求 $U_{U. B \wedge U. C = V. C}^{\infty} V$ 的结果,如表 3-22 所示。

表 3-22 关系 U 和关系 V 等值连接的结果

A	U. B	U. C	V. B	V. C	D
1	2	3	2	3	4
1	2	3	2	3	5
6	7	8	7	8	10
9	7	8	7	8	10

3.4.7 除 (Division)

定义 3-4 给定关系 $R(X, Y)$ 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。 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: 元组在 X 上分量值 x 的像集 Y_x 包含 S 在 Y 上投影的集合。记作:

$$R \div S = \{tr[X] \mid tr \in R \wedge \pi_Y(S) \subseteq Y_x\}$$

其中, Y_x 为 x 在 R 中的像集, $x = tr[X]$ 。

除操作是同时从行和列的角度进行运算的。除运算是乘运算的逆运算, 给定两个表 T 和 S , 如果表 R 是通过 $R = T \times S$ 定义的, 那么有 $T = R \div S$ 成立。在这个意义上可解释为什么称作除运算。

例 3-11 关系 R 和关系 S 如表 3-23 和表 3-24 所示, 求 $R \div S$ 的结果。结果如表 3-25 所示。

表 3-23 关系 R

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

表 3-24 关系 S

B	C	D
B ₁	C ₂	D ₁
B ₂	C ₁	D ₁
B ₂	C ₃	D ₂

表 3-25 $R \div S$ 的结果

A
A ₁

在关系 R 中, A 可以取四个值 $\{A_1, A_2, A_3, A_4\}$ 。其中, A_1 的像集为 $\{(B_1, C_2), (B_2, C_3), (B_2, C_1)\}$, A_2 的像集为 $\{(B_3, C_7), (B_2, C_3)\}$, A_3 的像集为 $\{(B_4, C_6)\}$, A_4 的像集为 $\{(B_6, C_6)\}$ 。

S 在 (B, C) 上的投影为 $\{(B_1, C_2), (B_2, C_1), (B_2, C_3)\}$

显然 A_1 的像集包含了 S 在 (B, C) 属性组上的投影, 所以 $R \div S = \{A_1\}$ 。

3.5 关系代数综合实例

本实例建立在 CAP 数据库基础上,各关系如表 3-26~表 3-29 所示。

表 3-26 C(顾客)关系

cid	cname	city	discent
c001	李广	天津	10.00
c002	王开基	北京	12.00
c003	安利德	北京	8.00
c004	曹土雄	天津	8.00
c006	曹土雄	广州	0.00

表 3-27 P(商品)关系

pid	pname	city	quantity	price
p01	梳子	天津	111 400	0.50
p02	刷子	成都	203 000	0.50
p03	刀片	西安	150 600	1.00
p04	钢笔	西安	125 300	1.00
p05	铅笔	天津	221 400	1.00
p06	文件夹	天津	123 100	2.00
p07	盒子	成都	100 500	1.00

表 3-28 A(代理)关系

aid	aname	city	percent
a01	史可然	北京	6
a02	韩利利	上海	6
a03	杜不朗	成都	7
a04	甘瑞	北京	6
a05	敖斯群	武汉	5
a06	史可然	天津	5

表 3-29 O(订单)关系

ordno	month	cid	aid	pid	qty	dollars
1011	01	c001	a01	p01	1000	450.00
1012	01	c001	a01	p01	1000	450.00
1019	02	c001	a02	p02	400	180.00
1017	02	c001	a06	p03	600	540.00
1018	02	c001	a03	p04	600	540.00
1023	03	c001	a04	p05	500	450.00
1022	03	c001	a05	p06	400	720.00