

命名

- 关系：表（行的完整集合）
- 元组：行
- 属性：列
- 关系模式：标题：列集合（稳定）
- 内容：行集合（变化）
- 基数：行数
- 数据库模式，所有关系模式

关系规则

NF：不允许多值属性（重复字段）和内部结构

- 解决：分表
- 非第一范式NFNF

只能基于内容存取元组

- 行无次序
- 不能指针再次检索行

元组唯一性

- 同一时刻不能有相同元组

键

最小的列集合能区分行

- 超键：非最小
- 必有键

- 主键是设计者选定的唯一标识符候选键
- 属性主键列不允许有空值

关系代数运算

集合运算			
名称	符号	键盘格式	示例
并	\cup	UNION	$R \cup S$, 或 $R \text{ UNION } S$
交	\cap	INTERSECT	$R \cap S$, 或 $R \text{ INTERSECT } S$
差	$-$	- 或 MINUS	$R - S$, 或 $R \text{ MINUS } S$
乘积	\times	TIMES	$R \times S$, 或 $R \text{ TIMES } S$

自然关系运算			
名称	符号	键盘格式	示例
投影	$R [\quad]$	$R [\quad]$	$R [A_{i_1} \dots A_{i_k}]$
选择	$R \text{ where } C$	$R \text{ where } C$	$R \text{ where } A_1 = 5$
连接	\bowtie	JOIN	$R \bowtie S$, 或 $R \text{ JOIN } S$
除	\div	DIVIDEBY	$R \div S$, 或 $R \text{ DIVIDEBY } S$

赋值&别名

定义2.6.3 赋值、别名 R 是一个表, $\text{Head}(R)=A_1 \cdots A_n$ 。假定 B_1, \dots, B_n 是 n 个属性。对于所有的 $1 \leq i \leq n$, 它们的域 $\text{Domain}(B_i)=\text{Domain}(A_i)$ 。通过赋值

$$S(B_1, \dots, B_n) := R(A_1, \dots, A_n)$$

我们定义一个新表 S , 且 $\text{Head}(S)=B_1 \cdots B_n$ 。新表 S 的内容恰好和旧表 R 的内容相同, 也就是说行 u 在 S 中当且仅当在 R 中存在一个行 t , 对于所有的 $1 \leq i \leq n$ 有 $u[B_i]=t[A_i]$ 成立。在赋值中使用的符号 $:=$ 称为赋值运算符。

我们已经努力使得赋值运算允许对源表的标题里的属性名进行重定义, 我们不久就可以体会到这一点的价值。现在我们需要指明, 属性的重定义并不总是必要。如果两个表中所有的属性名都是一对一的, 并且对于所有的 $1 \leq i \leq n$ 都有 $B_i=A_i$ 成立, 我们可以简单地称 S 为表 R 的别名, 简写为 $S:=R$ 。 ■

注意到在公式右边的表 R 也可以是关系代数的表达式求值, 因此, 只要我们在编程语言中使用赋值。这就允许我们把表达式计算的中间结果“保存”下来。在公式左边的表 S 必须是一个命名的表, 它不可以是一个表达式。

以下按优先级顺序

优先级	运算	符号
高	投影	R []
	选择	R where C
	乘积	×
	连接、除法	⋈, ÷
	交	∩
低	并、差	∪, −

图2-7 关系运算优先级

1.投影

- 取一部分列
- 删除重复行
- 查询：投影的表达式

2.选择where

定义2.7.2 选择 给定一个表S, $\text{Head}(S)=A_1 \cdots A_n$, 选择运算定义一个新表, 用 $S \text{ where } C$

表示, 具有相同的属性集合, 包含了S中满足选择条件(简称条件, 记为C)的元组。根据是否满足条件来决定对于每个给定的表的元组是否应该被选择出来保留在行集合中。条件C的形式可以通过如下的递归定义表示:

1) C可以是任何形如 $A_i \propto A_j$ 或者 $A_i \propto a$ 的比较, 这里 A_i 和 A_j 都是S中具有相同域的属性, a 是 $\text{Domain}(A_i)$ 中的一个常数, \propto 是比较符号, 可以是 $<$, $>$, $=$, $<=$, $>=$ 和 $<>$ (比较符 $<>$ 在两个被比较值不等时为真)等。表 $S \text{ where } A_i \propto A_j$ 包含所有满足条件 $t[A_i] \propto t[A_j]$ 的行 t 。而表 $S \text{ where } A_i \propto a$ 包含所有满足条件 $t[A_i] \propto a$ 的行。

用图2-2中的CUSTOMERS表举例, 条件可以写为 $\text{city}='Dallas'$ 和 $\text{discnt} \geq 8.00$ 。注意到条件中的字符常量是放在引号中间的。在CUSTOMERS表中没有属性对具有相同的值, 但是对于比如 $\text{city} > \text{cname}$ 这样的条件, 这里操作符 “ $>$ ” 要求第一个操作数中的字符常量在字母表中的位置位于第二个操作数之后。

2) C和C'都是条件, 那么新的条件可以是 $C \text{ AND } C'$, $C \text{ OR } C'$ 或者是 $\text{NOT } C$, 可能需要把新生成的条件放入圆括号中去。如果 $U := S \text{ where } C_1$, $V := S \text{ where } C_2$ 那么我们有:

- AND连接符: $S \text{ where } C_1 \text{ and } C_2$ 意思如同 $U \cap V$ 。
- OR连接符: $S \text{ where } C_1 \text{ or } C_2$ 意思如同 $U \cup V$ 。
- NOT连接符: $S \text{ where not } C_1$ 意思如同 $S - U$ 。

表 “ $S \text{ where } C$ ” 包含了S中所有满足条件C的行。条件的计算是通过检查定义2.7.2中的第一部分的子表达式的比较, 然后再检查复杂的逻辑语句是否正确。 ■

3.乘积

捏造数据, 虚拟分析

$m \text{ 列} * n \text{ 列} = m + n \text{ 列}$

- 互逆
- 不允许 $A * A$, 可以先 $B := A$
- 注意列名限定

4.连接, 除

除是乘积的逆, 但不可抵消, 除损失数据

“所有”: 除

对指定列内容的其他属性的筛选使用除

连接是横向的列拼接, 但需要能对应上

连接但没有相同列就是乘, 相当于没有限制了

连接但全相同列就是交

果相同，这也就是说，如果 $B_1 \cdots B_k$ 为空，即 $k=0$ ，那么 $R \bowtie S = R \times S$ ；另一方面，如果 R 和 S 具有相同的属性集，那么两个表的连接运算和两个表的交集相同，这就是说，如果 $A_1 \cdots A_n$ 和 $C_1 \cdots C_m$ 都为空集，即 $n=m=0$ ，那么 $R \bowtie S = R \cap S$ 。两者的等价关系在本章后面的习题可以验证，

- 先投影选择好，再连接！！

例2.7.8 现在我们想要查询所有订购了至少一种价值为\$0.50的商品的顾客名字。我们首先列出：

```
CHEAPS := (PRODUCTS where price = 0.50) [pid]
```

这个表达式取出了所有价值为\$0.50的商品的pid，然后我们计算

```
((ORDERS  $\bowtie$  CHEAPS)  $\bowtie$  CUSTOMERS) [cname]
```

从 $(ORDERS \bowtie CHEAPS)$ 中得出了包含价值\$0.50的商品的订单，然后把这些中间结果的

订单信息和表CUSTOMERS做连接运算可以找到顾客的名字。我们也可以通过把上面两步合并来避免赋值语句：

```
((ORDERS  $\bowtie$  (PRODUCTS where price = 0.50) [pid])  $\bowtie$  CUSTOMERS) [cname]
```

很重要的一点是在我们建立CHEAPS表的时候我们必须把价值\$0.50的商品信息投影到pid上。表达式

```
((ORDERS  $\bowtie$  PRODUCTS where price = 0.50)  $\bowtie$  CUSTOMERS) [cname]
```

可能无法正确解决这个问题，因为在第二次连接操作中有一些无法预计的额外列对于两个表来说是共有的：这个表达式要求商品的city和订购该商品的顾客所在的city也必须相同。这个练习之一就是要求你显示出后面这个查询的结果，并获得原先正确的关系代数表达式查询的结果。这个比较的意义在于告诉我们，在做连接的时候必须使用投影来避免一些连接上的不确定影响。 ■

- 合并语句可以避免使用赋值，但不太清晰

5.交并差

- 兼容表才有

标题相同，属性同域、相同含义

最小运算符集

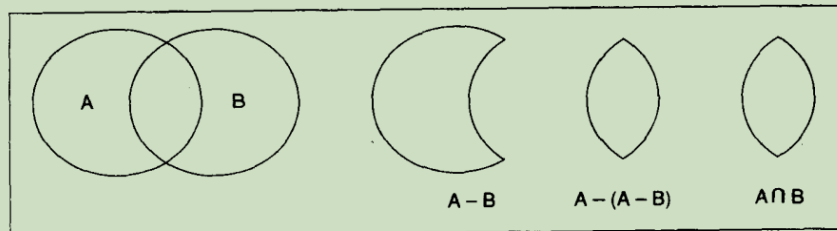
交，连接，除是非必要的

*交

定理2.8.1 R和S是两个兼容表， $\text{Head}(R)=\text{Head}(S)=A_1 \cdots A_n$ 。交运算可以被减运算重定义为： $A \cap B = A - (A - B)$ 。

证 我们用文氏图说明：

这个说明可以很容易用自然语言表达，如果需要的话。



*连接

其实T就差不多了

但还是要T1删去重复列

T2重命名

定理2.8.2 给定两个表R和S， $\text{Head}(R)=A_1 \cdots A_n B_1 \cdots B_k$ ， $\text{Head}(S)=B_1 \cdots B_k C_1 \cdots C_m$ ，其中 $n, k, m \geq 0$ ，R和S的连接可以用积、选择、投影运算以及赋值算子重写。

证 为了证明这一点，考虑表

$T:=(R \times S) \text{ where } R.B_1=S.B_1 \text{ and } \cdots \text{ and } R.B_k=S.B_k$

表T由积和选择操作构成，这和R和S的连接类似。现在我们只要简单地用投影运算把不需要的重复列剔除以及利用赋值算子重定义列名。

$T_1: T[R.A_1, \cdots, R.A_n, R.B_1, \cdots, R.B_k, S.C_1, \cdots, S.C_m]$

$T_2(A_1, \cdots, A_n, B_1, \cdots, B_k, C_1, \cdots, C_m):=T_1$

表T2结果和 $R \bowtie S$ 相同。

*除

• 双重减法

全部的减去不公共的、不满足的

不公共的（没有过交易的二元对）=乘积（所有可能二元对关系）-原来（有过交易的）

定理2.8.3* [难] 除法可以用投影、乘积和差表达。

证 考虑两个表R和S, $\text{Head}(R)=A_1 \cdots A_n B_1 \cdots B_m$ 且 $\text{Head}(S)=B_1 \cdots B_m$ 。我们可以证明:

$$R \div S = R[A_1, \dots, A_n] - ((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]。$$

假设 u 是表 $R[A_1, \dots, A_n] - ((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]$ 的一个元组。这意味着 u 在表 $R[A_1, \dots, A_n]$ 中, 且 u 不在表 $((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]$ 中。再假设 S 中存在元组 s , 使得 u 和 s 的串接不在 R 中。由于 u 在表 $R[A_1, \dots, A_n]$ 中, 这就说明了 u 是表 $((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]$ 的成员, 这就与假设相矛盾。所以对所有的 S 中的元组 s , u 和 s 的连接都在 R 中, 这意味着 u 属于表 $R \div S$ 。

反过来, 如果 u 是 $R \div S$ 中的一个元组, 显然 u 在表 $R[A_1, \dots, A_n]$ 中。另一方面, u 不在表 $((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]$ 中, 因此在 S 中存在一个元组 s 使得 u 和 s 的串接在 $R[A_1, \dots, A_n] \times S$ 中而不在 R 中。因此 u 属于表 $R[A_1, \dots, A_n] - ((R[A_1, \dots, A_n] \times S) - R)[A_1, \dots, A_n]$, 这就推出了我们的命题。你也可以从特定的表中用刚给出的除运算示例中推出结果。

应用

书写细节

- 连等要写成 $a=b$ and $b=c$
- 以N开头: $\text{where name} \geq 'N'$ and $\text{name} < 'O'$
- 重命名陷阱: 课程&先导课程

5. 满足下述条件的学生的学号和姓名: 选修过c009号课程的所有前导课程;

step 1: $R(\text{cno}) := (P \text{ where } \text{cno} = 'c009')[\text{pno}]$

step 2: $((L[\text{sno}, \text{cno}] \div R) \text{ join } S)[\text{sno}, \text{sn}]$

- 键陷阱: 只能用键找所有

4. 查询满足下述条件的供应商的名称aname: 至今为止还没有销售过商品;

解:

$((A[\text{aid}] - O[\text{aid}]) \text{ join } A)[\text{aname}]$

注: 在本题中不能直接用两个aname的集合去执行‘减法’运算。

技巧

- 取别名时先把投影取了
- 只有一次：乘，全部减多次的（判断重复：是同一个人但单号不同）
- 每次连接之前检查重复属性，有就先取投影
- 没有干过：用减法

最大值

1. 用除法：两元对比较，保留>关系满足的，除法消列
2. 不用除法：两元对比较，保留<满足的，再用所有的减去

4. 查询每一个供应商单笔销售金额最高的订单（即：在一个供应商的所有销售订单中，订单金额dols最高的订单），结果返回供应商编号，订单编号ordno，订单金额dols；

解：

(1) 查询每个供应商订单金额dols并非最高的订单：令 $O_1 := O, O_2 := O$

$H := ((O_1 \times O_2) \text{ where } O_1.\text{aid} = O_2.\text{aid} \text{ and } O_1.\text{dols} < O_2.\text{dols})[O_1.\text{aid}, O_1.\text{ordno}, O_1.\text{dols}]$

(2) 查询每个供应商销售金额最高的订单：

$O[\text{aid}, \text{ordno}, \text{dols}] - H$

二元对

$PAIRS := (L \times M) \text{ where } L.\text{city} = M.\text{city}$

有了表M，它和L的区别仅仅在于具有不同的名字，我们就可以在乘积 $L \times M$ 上定义一个条件，来检索满足具有相同城市的记录。基于图2-2的PAIRS表达式的结果如下：

PAIRS

L.aid	L.aname	L.city	L.percent	M.aid	M.aname	M.city	M.percent
a01	Smith	New York	6	a01	Smith	New York	6
a01	Smith	New York	6	a04	Gray	New York	6
a02	Jones	Newark	6	a02	Jones	Newark	6
a03	Brown	Tokyo	7	a03	Brown	Tokyo	7
a04	Gray	New York	6	a01	Smith	New York	6
a04	Gray	New York	6	a04	Gray	New York	6

- 去除冗余

$PAIRS2 := (L \times M) \text{ where } L.\text{city} = M.\text{city} \text{ and } L.\text{aid} < M.\text{aid}$

这个表达式的结果如下：

PAIRS2

L.aid	L.aname	L.city	L.percent	M.aid	M.aname	M.city	M.percent
a01	Smith	New York	6	a04	Gray	New York	6

或选择

- 用并

例2.7.5 假设，我们需要找出满足条件的城市，这个城市要求有顾客折扣率低于10%或者有一家佣金百分率不低于6%的代理商。这个查询可以用以下的关系代数表达式表示：

```
(CUSTOMERS where discont <10)[city] U (AGENTS where percent <6)[city]
```

注意到并操作在此处是必需的。我们无法用定义2.7.2中的选择条件中的OR连接符来取代。因为结果中的城市是从完全不同的表中得出，需要有不同的检索条件才能构成最终结果。 ■

用乘不用连接

4. 满足下述条件的学生的学号和姓名：只选修过自己就读院系开设的课程（没有选修过其他院系开设的课程）

$$(S \text{ join } L)[\text{sno}, \text{sn}] - ((S \times L \times C) \text{ where } S.\text{sno} = L.\text{sno} \text{ and } C.\text{cno} = L.\text{cno} \text{ and } S.\text{dept} \neq C.\text{dept})[\text{sno}, \text{sn}]$$

连接也可以，但要处理很多重复列

3. 查询满足下述条件的订单的编号ordno：顾客、供应商、商品三者所处的城市互不相同；

解：

$$((O \times C \times A \times P) \text{ where } O.\text{cid} = C.\text{cid} \text{ and } O.\text{aid} = A.\text{aid} \text{ and } O.\text{pid} = P.\text{pid} \text{ and } C.\text{city} \neq A.\text{city} \text{ and } C.\text{city} \neq P.\text{city} \text{ and } A.\text{city} \neq P.\text{city})[\text{ordno}]$$

注：在本题中，无法直接使用‘自然联接’运算实现关系的合并。