

7.3 元组演算、域演算与查询优化

主要考点

- 1、元组演算
- 2、域演算
- 3、查询优化

元组演算

- 元组关系演算是非过程化查询语言，简称元组演算。它只描述所需信息，而不给出获得该信息的具体过程。
- 在元组演算中，其元组演算表达式中的变量是以元组为单位的，其一般形式为：

$$\{t|P(t)\}$$

其中， t 是元组变量， $P(t)$ 是元组演算公式，公式是由原子公式组成。

1、原子公式：

(1) $R(t)$

R 是关系名， t 是元组变量， $R(t)$ 表示： t 是关系 R 中的一个元组。

(2) $t[i] \theta C$ 或 $C \theta t[i]$

$t[i]$ 表示元组变量 t 的第 i 个分量， C 是常量， θ 为算术比较运算符。

(3) $t[i] \theta u[j]$

t 和 u 是两个元组变量。 $t[i] \theta u[j]$ 表示元组变量 t 的第 i 个分量与元组变量 u 的第 j 个分量之间满足 θ 运算。

元组演算

2、公式的定义

- 若一个公式的一个元组变量前有全称量词 \forall 或存在量词 \exists 符号，则称该变量为约束变量，否则称之为自由变量。公式可递归定义如下：
 - (1) 原子公式是公式。
 - (2) 如果 φ_1 和 φ_2 是公式，那么， $\neg\varphi_1$ ， $\varphi_1\vee\varphi_2$ ， $\varphi_1\wedge\varphi_2$ ， $\varphi_1\Rightarrow\varphi_2$ 也都是公式。分别表示如下命题：
 $\neg\varphi_1$ 表示“ φ_1 不为真”， $\varphi_1\vee\varphi_2$ 表示“ φ_1 或 φ_2 为真”， $\varphi_1\wedge\varphi_2$ 表示“ φ_1 和 φ_2 都为真”； $\varphi_1\Rightarrow\varphi_2$ 表示“若 φ_1 为真则 φ_2 为真”。
 - (3) 如果是 φ_1 公式，那么， $\exists t(\varphi_1)$ 是公式。 $\exists t(\varphi_1)$ 表示这样一个命题：
“如果有一个 t 使 φ_1 为真，则 $\exists t(\varphi_1)$ 为真，否则 $\exists t(\varphi_1)$ 为假”。
 - (4) 如果是 φ_1 公式，那么， $\forall t(\varphi_1)$ 是公式。 $\forall t(\varphi_1)$ 表示这样一个命题：
“如果对所有的 t 使 φ_1 为真，则 $\forall t(\varphi_1)$ 为真，否则 $\forall t(\varphi_1)$ 为假”。
- 公式中运算符的优先顺序为：
- $\theta > \forall$ 和 $\exists > \neg > \wedge$ 和 $\vee > \Rightarrow$ ，加括号时，括号中的运算符优先。

例：设有关系R、S如下图所示，对如下所示的元组演算表达式，求出它们的值。

- (1) $R1 = \{ t \mid R(t) \wedge \neg S(t) \}$
- (2) $R2 = \{ t \mid S(t) \wedge t[3] > t[2] \wedge t[2] < 8 \}$
- (3) $R3 = \{ t \mid (\exists u) (R(t) \wedge S(u) \wedge t[3] < u[2]) \}$
- (4) $R4 = \{ t \mid (\forall u) (R(t) \wedge S(u) \wedge t[3] > u[1]) \}$
- (5) $R5 = \{ t \mid (\exists u) (\exists v) (R(u) \wedge S(v) \wedge u[2] > v[1] \wedge t[1] = u[1] \wedge t[2] = v[1] \wedge t[3] = v[3]) \}$

R		
A	B	C
1	2	3
4	5	6
7	8	9
10	11	12

S		
A	B	C
3	7	11
4	5	6
5	9	13
6	10	14

A	B	C
1	2	3
7	8	9
10	11	12

(1) R1

A	B	C
3	7	11
4	5	6

(2) R2

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

(3) R3

$$(3) R3 = \{ t \mid (\exists u) (R(t) \wedge S(u) \wedge t[3] < u[2]) \}$$

$\exists u$ 表示：只要有一个（存在一个） u 使得后面的公式成立就可以了，即 $t[3]$ 只要小于 $u[2]$ 中的其中一个就可以了，如果 $t[3] \geq$ 所有的 $u[2]$ ，就说明不存在任何一个 u 满足条件。

$$(4) R4 = \{ t \mid (\forall u) (R(t) \wedge S(u) \wedge t[3] > u[1]) \}$$

$\forall u$ 表示：对任意一个（所有的） u ，都要使后面的公式成立，只要有一个 u 没有满足这个条件，就不行。即：满足条件的 $t[3]$ 应该大于所有的 $u[1]$ 。

$$(5) R5 = \{ t \mid (\exists u) (\exists v) (R(u) \wedge S(v) \wedge u[2] > v[1] \wedge t[1] = u[1] \wedge t[2] = v[1] \wedge t[3] = v[3]) \}$$

R		
A	B	C
1	2	3
4	5	6
7	8	9
10	11	12

S		
A	B	C
3	7	11
4	5	6
5	9	13
6	10	14

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

(3) R3

A	B	C
7	8	9
10	11	12

(4) R4

R.A	S.A	S.C
4	3	11
4	4	6
7	3	11
7	4	6
7	5	13
7	6	14
10	3	11
10	4	6
10	5	13
10	6	14

(5) R5

域演算

- 域关系演算简称域演算。在域演算中，表达式中的变量是表示域的变量，可将关系的属性名视为域变量，域演算表达式的一般形式为：

$$\{t_1, \dots, t_k | P(t_1, \dots, t_k)\}$$

其中， t_1, \dots, t_k 是域变量， $P(t_1, \dots, t_k)$ 是域演算公式。

1、原子公式：

(1) $R(t_1, \dots, t_i, \dots, t_k)$

R 是 k 元关系， t_i 是元组变量 t 的第 i 个分量， $R(t_1, \dots, t_i, \dots, t_k)$ 表示这样一个命题：

以 $t_1, \dots, t_i, \dots, t_k$ 为分量的元组在关系 R 。

(2) $t_i \theta c$ 或 $c \theta t_i$

t_i 表示元组变量 t 的第 i 个分量， c 是常量， θ 为算术比较运算符。

(3) $t_i \theta u_j$

t_i 和 u_j 是两个域变量。 $t_i \theta u_j$ 表示元组变量 t 的第 i 个分量与元组变量 u 的第 j 个分量之间满足 θ 运算。

域演算

2、公式的定义

若一个公式的一个元组变量前有全称量词 \forall 或存在量词 \exists 符号，则称该变量为约束变量，否则称之为自由变量。公式可递归定义如下：

(1) 原子公式是公式。

(2) 如果 φ_1 和 φ_2 是公式，那么， $\neg\varphi_1$ ， $\varphi_1\vee\varphi_2$ ， $\varphi_1\wedge\varphi_2$ ， $\varphi_1\Rightarrow\varphi_2$ 也都是公式。

(3) 如果是 φ_1 公式，那么， $\exists t_i(\varphi_1)$ 是公式。 $\exists t_i(\varphi_1)$ 表示这样一个命题：

“如果有一个 t_i 使 φ_1 为真，则 $\exists t_i(\varphi_1)$ 为真，否则 $\exists t_i(\varphi_1)$ 为假”。

(4) 如果 $\varphi_1(t_1, \dots, t_i, \dots, t_k)$ 是公式，那么， $\forall t_i(\varphi_1)$ 是公式。 $\forall t_i(\varphi_1)$ 表示这样一个命题：

“如果对所有的 t_i 使 $\varphi_1(t_1, \dots, t_i, \dots, t_k)$ 为真，则 $\forall t_i(\varphi_1)$ 为真，否则 $\forall t_i(\varphi_1)$ 为假”。

公式中运算符的优先顺序为：

$\theta > \forall$ 和 $\exists > \neg > \wedge$ 和 $\vee > \Rightarrow$ ，加括号时，括号中的运算符优先。

例：设有关系R、S如下图所示，对如下所示的域演算表达式，求出它们的值。

$$(1) R1 = \{ t_1 t_2 t_3 \mid R(t_1 t_2 t_3) \wedge t_1 < t_2 \wedge t_2 > t_3 \}$$

$$(2) R2 = \{ t_1 t_2 t_3 \mid (R(t_1 t_2 t_3) \wedge t_1 > 4) \vee (S(t_1 t_2 t_3) \wedge t_2 < 8) \}$$

$$(3) R3 = \{ t_1 t_2 t_3 \mid (\exists u)(\exists v)(\exists w) R(ut_2v) \wedge S(t_1 wt_3) \wedge u \geq 7 \wedge v > w \}$$

R		
A	B	C
1	2	1
4	5	7
7	8	6
10	11	9

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

A	B	C
1	2	1
7	8	6
10	11	9

(1) R1

A	B	C
7	8	6
10	11	9
1	2	3
4	5	6

(2) R2

S.A	R.B	S.C
1	8	3
4	8	6
1	11	3
4	11	6
7	11	9

(3) R3

查询优化

- 查询优化是指为查询选择最有效的查询计划的过程。一个查询可能会有多种实现方法，关键是如何找出一个与之等价的且操作时间又少的表达式，以节省时间、空间，提高查询效率。
- 在关系代数运算中，笛卡儿积、连接运算是最耗费时间和空间的。

1、优化的准则：

- (1) 提早执行选取运算。对于有选择运算的表达式，应优化成尽可能先执行选择运算的等价表达式，以得到较小的中间结果，减少运算量和从外存读块的次数。
- (2) 合并乘积与其后的选择运算为连接运算。
- (3) 将投影运算与其后的其他运算同时进行，以避免重复扫描关系。
- (4) 将投影运算和其前后的二目运算结合起来，使得没有必要为去掉某些字段再扫描一遍关系。
- (5) 在执行连接前对关系适当地预处理，就能快速找到要连接的元组。方法有两种：索引连接法、排序合并连接法。
- (6) 存储公共子表达式。对于有公共子表达式的结果应存于外存（中间结果），这样，当从外存读出它的时间比计算的时间少时，就可节约操作时间。

例：查询成绩大于90的男生的学号，姓名。

(1) $\pi_{\text{学号,姓名}}(\sigma_{\text{成绩}>'90' \wedge \text{性别}='男'}(S \bowtie SC))$

$S \bowtie SC$

学号	姓名	性别	年龄	成绩
1001	李娜	女	18	94
1003	王华	男	18	92
1004	吴莉	女	20	86

学号	姓名	性别	年龄
1001	李娜	女	18
1002	赵聪	男	19
1003	王华	男	18
1004	吴莉	女	20

学号	成绩
1001	94
1003	92
1004	86
1005	72

$\sigma_{\text{成绩}>'90' \wedge \text{性别}='男'}(S \bowtie SC)$

学号	姓名	性别	年龄	成绩
1003	王华	男	18	92

$\pi_{\text{学号,姓名}}(\sigma_{\text{成绩}>'90' \wedge \text{性别}='男'}(S \bowtie SC))$

学号	姓名
1003	王华

(2) $\pi_{\text{学号,姓名}}(\sigma_{\text{性别}='男'}(S)) \bowtie \pi_{\text{学号}}(\sigma_{\text{成绩}>'90'}(SC))$

$\sigma_{\text{性别}='男'}(S)$

学号	姓名	性别	年龄
1002	赵聪	男	19
1003	王华	男	18

$\pi_{\text{学号,姓名}}(\sigma_{\text{性别}='男'}(S))$

学号	姓名
1002	赵聪
1003	王华

$\pi_{\text{学号,姓名}}(\sigma_{\text{性别}='男'}(S)) \bowtie \pi_{\text{学号}}(\sigma_{\text{成绩}>'90'}(SC))$

学号	姓名
1003	王华

$\sigma_{\text{成绩}>'90'}(SC)$

学号	成绩
1001	94
1003	92

$\pi_{\text{学号}}(\sigma_{\text{成绩}>'90'}(SC))$

学号
1001
1003

例：供应商数据库中有：供应商S、零件P、项目J、供应SPJ四个基本表（关系），其关系模式如下所示：

S (Sno, Sname, Status, City)	供应商编号，供应商名称，供应商城市
P (Pno, Pname, Color, Weight)	零件号，零件名称，颜色，重量
J (Jno, Jname, City)	工程号，项目名称，所在城市
SPJ (Sno, Pno, Jno, Qty)	供应商编号，零件编号，工程号，数量

若用户要求查询使用“上海”供应商生产的“红色”零件的工程号，请解答如下问题：

(1) 试写出该查询的关系代数表达式；

$$\pi_{Jno}(\sigma_{City='上海' \wedge Color='红'}(S \bowtie SPJ \bowtie P))$$

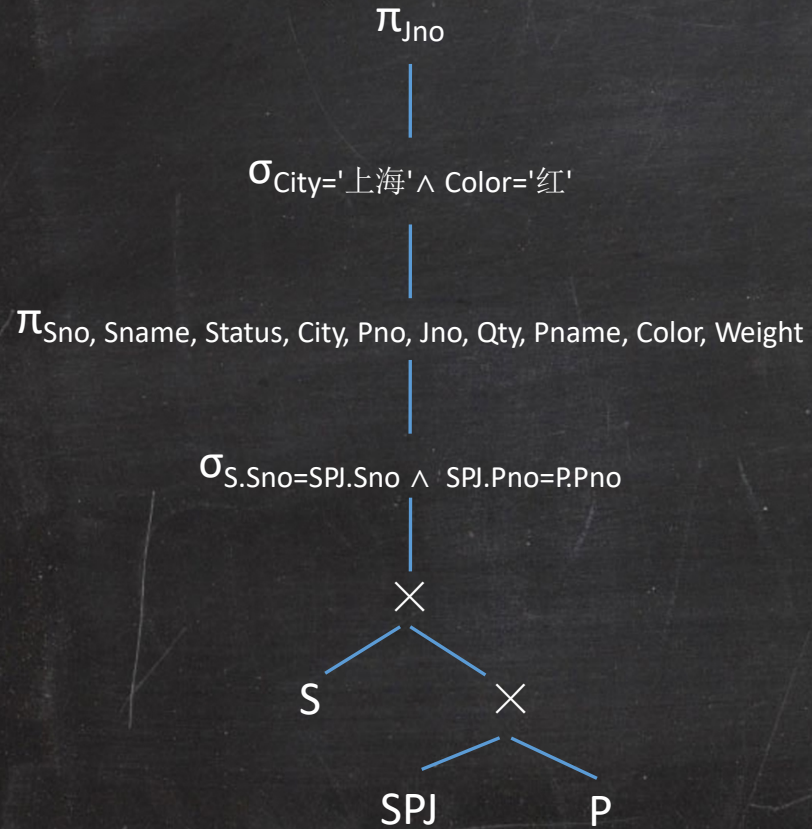
(2) 试写出查询优化的关系代数表达式。

$$\pi_{Jno}(\pi_{Sno}(\sigma_{city='上海'}(S)) \bowtie \pi_{Sno, Pno, Jno}(SPJ) \bowtie \pi_{Pno}(\sigma_{Color='红'}(P)))$$

(3) 画出该查询初始的关系代数表达式的语法树。

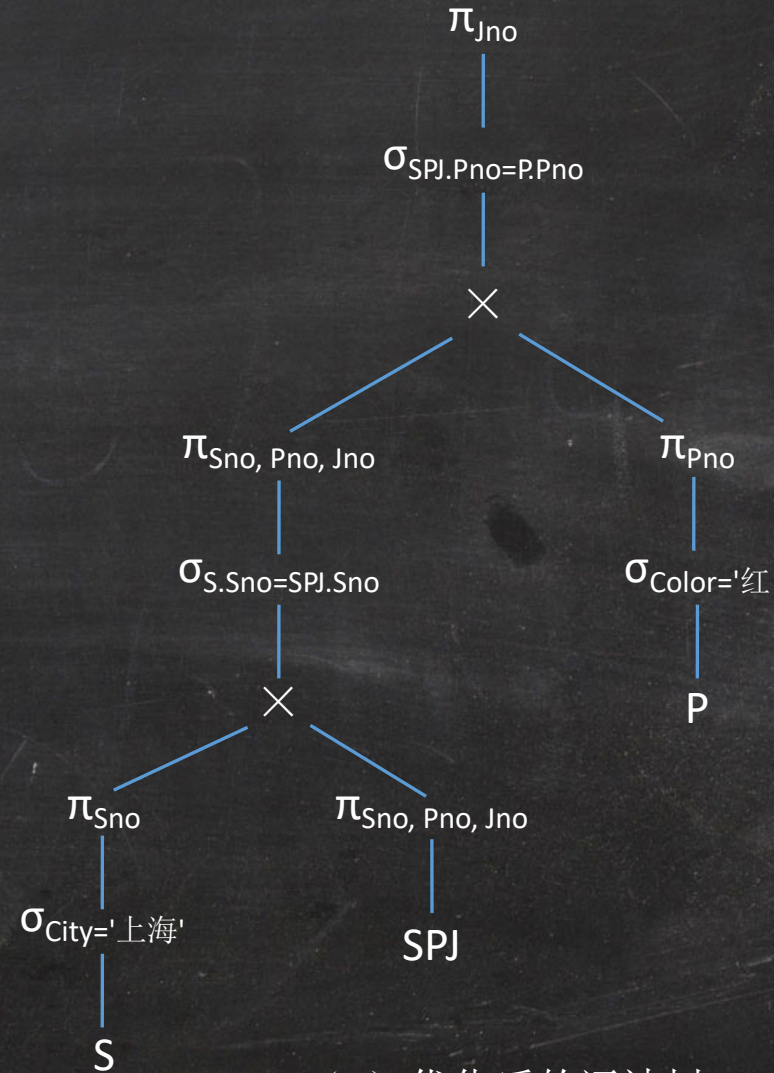
(4) 使用优化算法，对语法树进行优化，并画出优化后的语法树。

(1) $\pi_{Jno}(\sigma_{City='上海' \wedge Color='红'}(S \bowtie SPJ \bowtie P))$



(a) 优化前的语法树

(2) $\pi_{Jno}(\pi_{Sno}(\sigma_{city='上海'}(S)) \bowtie \pi_{Sno, Pno, Jno}(SPJ) \bowtie \pi_{Pno}(\sigma_{Color='红'}(P)))$



(b) 优化后的语法树

1、19年第38题

关系代数表达式的查询优化中，下列说法错误的是（ C ）。

- A、提早执行选择运算
- B、合并乘积与其后的选择运算为连接运算
- C、如投影运算前后存在其它的二目运算，应优先处理投影运算
- D、存储公共的子表达式，避免重新计算

2、16年第37题

关系R，S如下表所示，元组演算表达式 $T=\{t \mid R(t) \wedge \forall u(S(u) \rightarrow t[3] > u[1])\}$ 运算的结果为（ C ）。

R			S		
A	B	C	A	B	C
1	2	3	3	7	11
4	5	6	4	5	6
7	8	9	5	9	13
10	11	12	6	10	14

A、

A	B	C
1	2	3
4	5	6

B、

A	B	C
3	7	11
4	5	6

C、

A	B	C
7	8	9
10	11	12

D、

A	B	C
5	9	13
6	10	14

3、15年第30题

在关系R (A1, A2, A3) 和S (A2, A3, A4) 上进行关系运算的4个等价的表达式E1, E2, E3和E4如下所示：

$$E1 = \pi_{A1,A4}(\sigma_{A2 < '2015' \wedge A4 = '95'}(R \bowtie S))$$

$$E2 = \pi_{A1,A4}(\sigma_{A2 < '2015'}(R) \bowtie \sigma_{A4 = '95'}(S))$$

$$E3 = \pi_{A1,A4}(\sigma_{R.A2=S.A2 \wedge R.A3=S.A3 \wedge A2 < '2015' \wedge A4 = '95'}(R \times S))$$

$$E4 = \pi_{A1,A4}(\sigma_{R.A2=S.A2 \wedge R.A3=S.A3}(\sigma_{A2 < '2015'}(R) \times \sigma_{A4 = '95'}(S)))$$

如果严格按照表达式运算顺序，则查询效率最高的是 (B) 。

A. E1

B. E2

C. E3

D. E4

4、13年第33~35题

关系R、S如下图所示，关系代数表达式 $\pi_{R.A,S.B,S.C}(\sigma_{R.A>S.B}(R \times S)) = (D)$ ，它与元组演算表达式 $\{t | (\exists u)(\exists v)(R(u) \wedge S(v) \wedge (B) \wedge (C))\}$ 等价。

A	B	C
a	b	c
d	e	f
h	i	j
k	m	n

R

A	B	C
c	h	m
d	h	f
e	n	p
f	k	q

S

(33) A.

R.A	S.B	S.C
a	n	p
a	k	q

B.

R.A	S.B	S.C
e	h	m
e	h	f

C.

R.A	S.B	S.C
h	n	p
h	k	q

D.

R.A	S.B	S.C
k	h	m
k	h	f

(34) A. $u[1] < v[2]$ B. $u[1] > v[2]$

(35) A. $t[1]=v[1] \wedge t[2]=u[5] \wedge t[3]=v[6]$

C. $t[1]=u[1] \wedge t[2]=v[2] \wedge t[3]=v[3]$

C. $u[1] < v[5]$ D. $u[1] > v[5]$

B. $t[1]=u[1] \wedge t[2]=u[2] \wedge t[3]=u[3]$

D. $t[1]=u[1] \wedge t[2]=v[2] \wedge t[3]=u[3]$