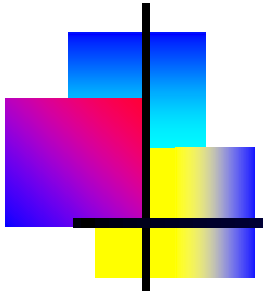
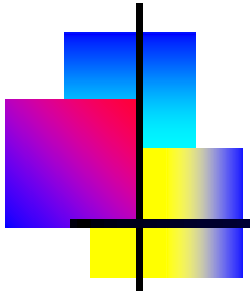


教材：数据库实用教程（第四版）

《数据库原理》课程

清华大学出版社

2024年12月30日



《数据库原理》

第三章 关系运算

清华大学出版社
2024年12月30日

第三章 关系运算

教学内容：

- 基本概念：关系模型、关键码、关系的定义和性质、三类完整性规则。
- 关系代数：基本操作、组合操作、扩充操作。
- 关系演算：元组演算、域演算、关系演算的安全性和等价性。
- 查询优化：关系代数表达式的等价转换规则、优化算法。

教学重点：　关系代数运算

§1 关系数据模型的基本概念

一、基本术语

用二维表格表示实体集；

外键表示实体间联系的模型称为关系模型；

关系：对应二维表格；

元组：表中的行；

属性：表中的列；

域：属性的取值范围。

二、数学定义

定义一： 域（Domain）是值的集合。即域是属性的取值范围。

定义二： 关系的定义

①用集合论的观点定义关系：关系是一个元数为K的元组的集合。

即这个关系中有若干个元组，每个元组有K个属性值。 把关系看成一个集合，集合中的元素是元组。

②用值域的观点定义关系： 关系是属性值域笛卡儿积的一个子集。

三、关系的性质

1、列具有相同的性质，不同的列可有相同的域，

2、任意两个元组不能相同，元组的次序可交换；

3、每个属性值(分量)都是不可分的数据项 (即属性值为最小单位).

四、关键码

超　键： 能唯一标识元组的属性组合（可能存在多余的属性）。

侯选键： 能唯一标识元组的最小属性组合。

主键： 若一个关系中有多个侯选键,则选其中的一个为关系的主键。

外键： 若一个关系R中包含有另一个关系S的主键所对应的属性组F， 则称F为R的外键。称关系S为参照关系，R为依赖关系。

五、关系模型的三类完整性规则

实体完整性规则： 关系中元组的主键值不能为空。

参照完整性规则： 关系中元组的外键值只允许有两种可能值：空值、或与相应的参照关系中的某个主键值相同。

用户定义的完整性规则： 由应用环境决定，反映某一具体应用涉及的数据必须满足的约束条件。

§2 关系代数

一、关系数据语言

关系数据库语言由查询语句（描述用户的检索操作）和更新语句（描述用户的插入、修改和删除等操作）两大类组成。

关系查询语言分为：

关系代数语言： 以集合操作为基础；

关系演算语言：以谓词演算为基础；
- 元组关系演算语言
- 域关系演算语言

基于关系代数和关系演算语言双重特点的语言： SQL

二、关系代数的基本运算

1. 并（union）：设关系R和关系S具有相同的元数，且相应的属性列具有相同的特征：

$$R \cup S \equiv \{t \mid t \in R \vee t \in S\}$$

其中：t是元组变量，R和S的元数相同。

并运算要求两个关系属性的性质必须一致且并运算的结果要消除重复的元组。

举例：

2. 差(differedce)：设关系R和关系S具有相同的元数，

且相应的属性列具有相同的特征：

$$R-S\equiv\{t \mid t\in R\wedge t\notin S\}$$

其中：t是元组变量，R和S的元数相同。

举例：

3. 笛卡儿积（cartesian product）

设关系R和关系S的元数分别为r和s。定义R和S的笛卡儿积R×S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个分量是S的一个元组，记为R×S。

形式定义如下：

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

若R有n个元组，S有m个元组，则R×S有n×m个元组。

举例：

4. 投影（projection）：

对一个关系进行垂直分割，消去某些列，

并重新安排列的顺序，再删去重复元组。

设关系R是k元关系，R在其分量$A_{i1},\ldots,A_{im}$($m\leqslant k,\ldots,i1,\ldots,im$为1到k之间的整数)上的投影用$\pi_{i1,\ldots,im}(R)$表示，它是从R中选择若干属性列组成的一个m元元组的集合，形式定义如下：

$$\pi_{i1,\ldots,im}(R)\equiv\{t\mid t=\langle t_{i1},\ldots,t_{im}\rangle\wedge\langle t_1,\ldots,t_m\rangle\in R,1\leq m\leq k\}$$

举例：

5. 选择(selection)

根据某些条件对关系做水平分割,选择符合条件的元组。

条件用命题公式F（即计算在括号中的条件表达式）表示.

F中的运算对象：　常量(用引号括起来），

元组分量（属性名或列的序号），

算术比较运算符:＜，≤，＞，≥，＝，≠,

逻辑运算符：∧，∨，¬。

关系R关于公式F的选择操作用σ_F（R）表示，形式定义如下：

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

举例：

三、关系代数的组合操作

1、交（intersection）

设关系R和关系S具有相同的元数n（即两个关系都有n个属性），而且相应的属性取自同一个域。关系R和S的交记为R∩S，结果仍为n元的关系。由即属于R又属于S的元组组成。形式定义如下：

$$R \cap S \equiv \{t \mid t \in R \wedge t \in S\}$$

t是元组变量，R和S的元数相同。

关系的交可以由关系的差来表示，即：

$$R \cap S \equiv R - (R-S)$$

举例：

2、 联接(join)

联接操作是笛卡儿积、选择和投影操作的组合。

联接分成 θ 联接和F联接两种。

① θ 联接： $R \underset{i\,\theta\,j}{\bowtie} S \equiv \sigma_{i\,\theta\,(r+j)}(R \times S)$

如果 θ 为等号“=”，那么这个联结操作称为等值连接。

② F 联接： F联接是从关系R和S的笛卡儿积中选取属性间满足某一公式F的元组,记为： $R \underset{F}{\bowtie} S$

这里F是形为F1∧F2∧...∧Fn公式，每个Fi是形为i θ j的式子。

而i和j分别为关系R和S的第i个分量和第j个分量的序号。

举例：

3、自然联接(natural join)---特殊的等值连接

将关系R和S中公共属性组满足对应分量相等的元组联接起来， 并且要在结果中将重复的属性去掉。

$$R \bowtie S \equiv \pi_{i1,\ldots im}(\sigma_{R.A1=S.A1 \wedge \ldots \wedge R.AK=S.AK}(R \times S))$$

举例：

4、 除(division)

设关系R和S的元数分别为:r、s（r>s>0），

R÷S: 是一个（r-s）元的元组的集合，

是满足下列条件的最大关系：

其中每个元组t与S中每个元组u组成的

新元组<t,u>必在关系R中。

R÷S的具体计算过程如下：

① $T = \pi_{1,2,\ldots,r-s}(R)$

② $W = (T \times S) - R$

③ $V = \pi_{1,2,\ldots,r-s}(W)$

④ $R \div S = T - V$

$$R \div S \equiv \pi_{1,2,\ldots,r-s}(R) - \pi_{1,2,\ldots,r-s}((\pi_{1,2,\ldots,r-s}(R) \times S) - R)$$

举例：

四、关系代数表达式及其应用实例

工程项目零件供应数据库PROJECTY有四个关系模式有四个：

供应商关系：　S(SNO, SNAME, SADDR)

零件关系：　P(PNO, PNAME, COLOR, WEIGHT)

工程项目关系：J(JNO, JNAME, JCITY, BALANCE)

供应情况关系：SPJ(SNO, PNO, JNO, PRICE, QTY)

试用关系代数表达式表示每个查询语句。

1. 检索供应零件给工程J1的供应商编号SNO与零件编号PNO。

$\pi_{SNO,PNO}(\sigma_{JNO='J1'}(SPJ))$　或：$\pi_{1,2}(\sigma_{3='J1'}(SPJ))$

2. 检索供应零件给工程J1，且零件编号为P1的供应商编号SNO。

$$\pi_{SNO}(\sigma_{JNO='J1' \wedge PNO='P1'}(SPJ))$$

3. 检索使用了编号为P3零件的工程编号和名称。

$$\pi_{JNO,JNAME}(\sigma_{PNO='P3'}(J \bowtie SPJ))$$

4. 检索供应零件给工程J1,且零件颜色为红色的供应商名称和地址。

$$\pi_{SNAME,SADDR}(\sigma_{JNO='J1' \wedge COLOR='红色'}(S \bowtie SPJ \bowtie P))$$

5. 检索使用了零件编号为P3或P5零件的工程编号JNO。

$$\pi_{JNO}(\sigma_{PNO='P3' \vee PNO='P5'}(SPJ))$$

6. 检索至少使用了编号为P3和P5 零件的工程编号JNO。

$$\pi_{3}(\sigma_{3=8 \wedge 2=\text{'P3'} \wedge 7=\text{'P5'}}(SPJ \times SPJ))$$

7. 检索不使用编号为P3零件的工程编号JNO和工程名称JNAME。

$$\pi_{JNO,JNAME}(J) - \pi_{JNO,JNAME}(\sigma_{PNO=\text{'P3'}}(S \bowtie SPJ \bowtie P))$$

8. 检索使用了全部零件的工程名称JNAME。

$$\pi_{JNAME}(J \bowtie (\pi_{JNO,PNO}(SPJ) \div \pi_{PNO}(P)))$$

9. 检索使用零件包含编号为S1的供应商所供应的全部零件的工程编号JNO。

$$\pi_{JNO,PNO}(\sigma_{SNO=\text{'S1'}}(SPJ)) \div \pi_{PNO}(\sigma_{SNO=\text{'S1'}}(SPJ))$$

举例　假设学生数据库中的关系模式如下：

S（SNO, SNAME, AGE, SEX, SDEPT）

C（CNO, CNAME, CDEPT, TNAME）

SC（SNO, CNO, GRADE）

试用关系代数表达式表示每个查询语句。（板书举例）

1. 检索计算机系的全体学生的学号、姓名和性别。

2. 检索学习课程号为C2的学生的学号和姓名。

3. 检索选修课程名为“数据结构”的学生的学号和姓名。

4. 检索选修课程号为C2或C4的学生的学号。

5. 检索至少选修课程号为C2和C4的学生的学号。

6. 检索没有选修C2课程的学生的姓名和年龄。

7. 检索选修了全部课程的学生的姓名和所在系。

8. 检索选修课程包含学生S2所选的全部课程的学生的学号。

五、扩充的关系代数操作

1. 外联接（outer join）⟗

$$R \bowtie S \equiv \pi_{i1,\ldots im}(\sigma_{R.A1=S.A1 \wedge \ldots \wedge R.AK=S.AK}(R \times S))$$

在R和S做自然联接时，把原该舍弃的元组也保留在新关系中，同时在这些元组新增加的属性上填上空值（null），这种操作称为“外联接”操作，用符号R ⟗ S表示。

i、如果R和S做自然联接时，把R中原该舍弃的元组放到新关系中，那么这种操作称为“左外联接”操作，用符号： R  S表示。

ii、如果R和S做自然联接时，把S中原该舍弃的元组放到新关系中，那么这种操作称为“右外联接”操作，用符号：R  S表示。

A	B	C
a	b	c
b	b	f
c	a	d

关系R

B	C	D
b	c	d
b	c	e
a	d	b
e	f	g

关系S

A	B	C	D
a	b	c	d
a	b	c	e
c	a	d	b
b	b	f	null
null	e	f	g

⋈

A	B	C	D
a	b	c	d
a	b	c	e
c	a	d	b
b	b	f	null

⋈

A	B	C	D
a	b	c	d
a	b	c	e
c	a	d	b
null	e	f	g

⋈

2. 外部并（outer union）

如果R和S的关系模式不同，构成的新关系属性有R和S的所有属性组成（公共属性只取一次），新关系的元组由属于R或属于S的元组构成，同时元组在新增加的属性上填上空值(null)。

A	B	C	D
a	b	c	null
b	b	f	null
c	a	d	null
null	b	c	d
null	b	c	e
null	a	d	b
null	e	f	g

3. 半联接（semijoin）

关系R和S的半联接操作记为$R \ltimes S$：

R和S的自然联接在关系R的属性集上的投影

即：$$R \ltimes S \equiv \pi_R(R \bowtie S)$$

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

关系R

B	C	D
b	c	d
b	c	e
a	d	b

关系S

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

R ⋈ S

A	B	C
a	b	c
d	b	c
c	a	d

R ⋉ S

B	C	D
b	c	d
b	c	e
a	d	b

S ⋉ R

§4 关系演算

以数理逻辑中的谓词演算为基础，用公式表示关系演算的条件。

关系演算按所用到的变量不同可分为：

元组关系演算———以元组为变量；

域关系演算 ———以域为变量。

一、元组关系演算

形式： $\{t \mid \varphi(t)\}$

其中 t：元组变量；

φ：公式，公式有原子公式及运算符组成。

1. 原子公式有下列三种形式：

① R（t）：　　R 是关系名，t 是元组变量。

② t[i] θ u[j]：t 和 u是元组变量， θ 是算术比较运算符。

③ t[i] θ C 或 C θ u[i]：　t 和 u 是元组变量，c 是常量。

2. 约束变量和自由变量

在一个公式中，如果一个元组变量的前面没有存在量词 ∃ 或全称量词∀的符号定义，称之为自由元组变量，否则称为约束元组变量。

3. 运算符及优先次序为：

i. 括号中运算符优先级最高

ii. 算术比较运算符最高；

iii. 量词 ：存在量词 ∃

全称量词∀

iv. 逻辑运算符：¬ 、∧、∨

4. 公式中变量的递归定义如下：

①每个原子公式是一个公式。其中的元组变量是自由变量；

②如果φ1和φ2是元组关系演算公式，则：

φ1∧φ2, φ1∨φ2, ┓φ1也是元组关系演算公式；

③若φ是元组关系演算公式，则(∃t)(φ)也是元组关系演算公式；

④若φ是元组关系演算公式，则(∀t)(φ)也是元组关系演算公式；

⑤在元组演算公式中，各种运算符的优先次序为：

括号, 算术比较运算符, ∃, ∀, ¬ 、∧、∨

⑥有限次地使用上述五条规则得到的公式是元组关系演算公式，

其他公式不是元组关系演算公式。

5. 用元组关系演算表达式表示关系运算

并： $R \cup S$ $\{t \mid R(t) \vee S(t)\}$

差： $R-S$ $\{t \mid R(t) \wedge \neg S(t)\}$

笛卡儿积:$R \times S$ $\{t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge t[1]=u[1] \wedge \ldots \wedge$

$t[r]=u[r] \wedge t[r+1]=v[1] \wedge \ldots \wedge t[r+s]=v[s])\}$

投影: $\pi_{i1,i2,\ldots ik}(R)$ $\{t \mid (\exists u) \mid R(u) \wedge t[1]=u[i_1] \wedge \ldots \wedge t[iK]=u[iK]\}\}$

选择: $\sigma_F(R)$ $\{t \mid R(t) \wedge F'\}$

F'是F在元组演算中的等价表示形式。

- 如果P1和P2是公式，在元组关系演算的公式中，存在下列等价转换规则：

$$P1 \wedge P2 \equiv \neg(\neg P1 \vee \neg P2)$$

$$P1 \vee P2 \equiv \neg(\neg P1 \wedge \neg P2)$$

$$(\forall s)(P1(s)) \equiv \neg(\exists s)(\neg P1(s))$$

$$(\exists s)(P1(s)) \equiv \neg(\forall s)(\neg P1(s))$$

$$P1 \Rightarrow P2 \equiv \neg P1 \vee P2$$

6. 实例:用元组表达式形式表示下列查询

例： 用元组表达式形式表示下列查询：

1. 检索供应零件给工程J1，且零件编号为P1的供应商编号SNO。

$\{t \mid (\exists u)(SPJ(u) \wedge u[2]='P1' \wedge u[3]='J1' \wedge t[1]=u[1])\}$

2. 检索使用了编号为P3零件的工程编号和名称。

$\{t \mid (\exists u)(\exists v)(J(u) \wedge SPJ(v) \wedge v[2]='P3' \wedge u[1]=v[3] \wedge t[1]=u[1]$

$\wedge t[2]=u[2])\}$

3. 检索至少使用了编号为P3和P5零件的工程编号JNO。

$\{t \mid (\exists u)(\exists v)(SPJ(u) \wedge SPJ(v) \wedge u[3]=v[3] \wedge u[2]='P3'$

$\wedge v[2]='P5' \wedge t[1]=u[3])\}$

4. 检索不使用编号为P3零件的工程编号JNO和工程名称SNAME。

$$\{t|(\exists u)(\forall v)(J(u)\wedge SPJ(v)\wedge((u[1]=v[3])\Rightarrow v[2]\neq\text{'}P3\text{'})$$
$$\wedge t[1]=u[1]\wedge t[2]=u[2])\}$$

5. 检索使用了全部零件的工程名称JNAME。

$$\{t|(\exists u)(\forall v)(\exists w)(J(u)\wedge P(v)\wedge SPJ(w)\wedge u[1]=w[3]\wedge v[1]=w[2]$$
$$\wedge t[1]=u[2])\}$$

6. 检索使用零件包含编号为S1供应商所供应的全部零件的工程编号。

$$\{t|(\exists u)(SPJ(u)\wedge(\forall v)(SPJ(v)\wedge(v[1]=\text{'}S1\text{'}\Rightarrow(\exists w)(SPJ(w)$$
$$\wedge w[3]=u[3]\wedge w[2]=v[2])))\ \wedge t[1]=u[3])\}$$

举例　假设学生数据库中的关系模式如下：

S（SNO, SNAME, AGE, SEX, SDEPT）

C（CNO, CNAME, CDEPT, TNAME）

SC（SNO, CNO, GRADE）

用元组表达式形式表示下列查询：　（板书举例）

1. 检索计算机系的全体学生的学号、姓名和性别。

2. 检索学习课程号为C2的学生的学号和姓名。

3. 检索选修课程名为“数据结构”的学生的学号和姓名。

4. 检索选修课程号为C2或C4的学生的学号。

5. 检索至少选修课程号为C2和C4的学生的学号。

6. 检索没有选修C2课程的学生的姓名和年龄。

7. 检索选修了全部课程的学生的姓名和所在系。

8. 检索选修课程包含学生S2所选的全部课程的学生的学号。

二、域关系演算

域演算表达式的定义类似于元组演算表达式的定义，

所不同的是公式中用域变量代替元组变量的每一个分量，

域变量的变化范围是某个值域而不是一个关系。

1、域演算表达式：

一般形式：$\{t_1\ t_2 \cdots t_k \mid P(t_1, t_2, \cdots, t_k)\}$

其中t_1、t_2、…、t_k分别是元组变量t的各个分量的域变量，

P是域演算公式。

①原子公式有下列两种形式：

i. R（t1...tk）：R是K元关系，每个ti是域变量或常量。

ii. x θ y,其中x,y是域变量或常量，但至少有一个是域变量， θ 是算术比较运算符。

②域关系演算的公式中也可使用∧、∨、¬ 和=>等逻辑运算符

也可用(∃x)和（∀x）形成新的公式，但变量x是域变量，不是元组变量。

自由域变量、约束域变量等概念和元组演算中一样。

2. 元组表达式到域表达式的转换规则：

① 对于k元的元组变量t ， 引入k个域变量$t_1, t_2, \ldots, t_k$，

在公式中t用$t_1, t_2, \ldots, t_k$替换，元组分量t[i]用t_i替换。

② 对于每个量词(∃u)或（∀u），若u是m元的元组变量，

则引入m个新的域变量$u_1, u_2, \ldots, u_m$。

在量词的辖域内，u用$u_1u_2\ldots u_m$替换，u[i]用u_i替换，

(∃u)用$(\exists u_1)\ldots(\exists u_m)$ 替换，

(∀u）用（$\forall u_1$）…$(\forall u_m)$替换。

3. 举例(板书举例)

三、关系运算的安全性和等价性

1. 关系运算的安全性

$\{t|\neg R(t)\}$ 无限关系

验证$(\forall u)(w(u))$为’T’或$(\exists u)(w(u))$为’F’ 无穷验证

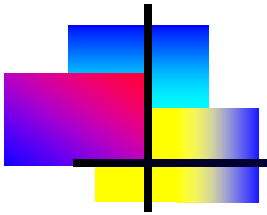
安全运算：

不产生无限关系和无穷次验证的运算称为安全运算，相应的表达式称为是安全表达式。所采用的措施称为安全约束。

安全约束集DOM(φ)：公式φ中的常量和φ中所出现关系的所有属性值组成的集合。

2. 关系运算的等价性

并、差、笛卡儿积、投影和选择是关系代数最基本的操作，并构成了关系代数运算的最小完备集。可以证明，在这个基本上，关系代数、安全的元组关系演算、安全的域关系演算在关系的表达和操作能力上是等价的。



§5 查询优化

关系代数表达式的优化问题:

在关系代数表达式中需要指出若干关系的操作步骤。

系统应该以什么样的操作顺序，才能做到既省时间，又省空间，而且效率也比较高呢？

如何花费较少的时间和空间，有效地执行笛卡儿积操作.

一、查询优化的总目标

选择有效的策略，

求得给定关系代数表达式的值，

达到提高DBMS系统效率的目标。

例:学生数据库： S(SNO, SNAME, SEX, SDEPT)

C(CNO, CNAME, TEACHER, CDEPT)

SC(SNO, CNO, GRADE)

检索选修C2课程的学生的学号和姓名,用关系代数表达式表示：

$$E1 = \pi_{SNAME}(\sigma_{S.SNO=SC.SNO \wedge SC.CNO='C2'}(S \times SC))$$

$$E2 = \pi_{SNAME}(\sigma_{SC.CNO='C2'}(S \bowtie SC))$$

$$E3 = \pi_{SNAME}(S \bowtie \sigma_{SC.CNO='C2'}(SC))$$

这三个关系代数表达式是等价的，但执行的效率大不一样。显然，求E1，E2，E3的大部分时间是花在联接操作上的。

二、代数表达式的等价变换规则

1、 联接和笛卡儿积的交换律

$$E1 \underset{F}{\bowtie} E2 \equiv E2 \underset{F}{\bowtie} E1$$

$$E1 \bowtie E2 \equiv E2 \bowtie E1$$

$$E1 \times E2 \equiv E2 \times E1$$

2. 联接和笛卡儿积的结合律

$$(E1 \underset{F1}{\bowtie} E2) \underset{F2}{\bowtie} E3 \equiv E1 \underset{F1}{\bowtie} (E2 \underset{F2}{\bowtie} E3)$$

$$(E1 \bowtie E2) \bowtie E3 \equiv E1 \bowtie (E2 \bowtie E3)$$

$$(E1 \times E2) \times E3 \equiv E1 \times (E2 \times E3)$$

3. 投影的串接

设L1，L2，…，Ln为属性集，并且L1⊆L2⊆…⊆Ln， 成立：

$$\pi_{L1}(\pi_{L2}(\ldots(\pi_{Ln}(E))\ldots)) \equiv \pi_{L1}(E)$$

4. 选择的串接

$$\sigma_{F1}(\sigma_{F2}(E)) \equiv \sigma_{F1 \wedge F2}(E)$$

由于F1∧ F2 = F2∧ F1，选择的交换律也成立：

$$\sigma_{F1}(\sigma_{F2}(E)) \equiv \sigma_{F2}(\sigma_{F1}(E))$$

5. 选择和投影操作的交换

$$\pi_L(\sigma_F(E)) \equiv \sigma_F(\pi_L(E))$$

要求条件F只涉及到L中的属性，如果F还涉及到不在L中的属性集L1：

$$\pi_L(\sigma_F(E)) \equiv \pi_L(\sigma_F(\pi_{L \cup L1}(E)))$$

6. 选择对笛卡儿积的分配律

$$\sigma_F(E1 \times E2) \equiv \sigma_F(E1) \times E2$$

$$\sigma_F(E1 \times E2) \equiv \sigma_{F1}(E1) \times \sigma_{F2}(E2)$$

$$\sigma_F(E1 \times E2) \equiv \sigma_{F2}(\sigma_{F1}(E1) \times E2)$$

7. 选择对并的分配律

$$\sigma_F(E1 \cup E2) \equiv \sigma_{F1}(E1) \cup \sigma_{F2}(E2)$$

8. 选择对集合差的分配律

$$\sigma_F(E1 - E2) \equiv \sigma_F(E1) - \sigma_F(E2)$$

9. 选择对自然联接的分配律

$$\sigma_F(E1 \bowtie E2) \equiv \sigma_F(E1) \bowtie \sigma_F(E2)$$

10. 投影对笛卡儿积的分配律

$$\pi_{L1 \cup L2}(E1 \times E2) \equiv \pi_{L1}(E1) \times \pi_{L2}(E2)$$

11. 投影对并的分配律

$$\pi_L(E1 \cup E2) \equiv \pi_L(E1) \cup \pi_L(E2)$$

12. 选择与联接操作的结合

$$\sigma_{F1}(E1 \underset{F2}{\bowtie} E2) \equiv E1 \underset{F2 \wedge F2}{\bowtie} E2$$

13. 并和交的交换律

$$E1 \cup E2 \equiv E2 \cup E1$$

$$E1 \cap E2 \equiv E2 \cap E1$$

14. 并和交的结合律

$$(E1 \cup E2) \cup E3 \equiv E1 \cup (E2 \cup E3)$$

$$(E1 \cap E2) \cap E3 \equiv E1 \cap (E2 \cap E3)$$

三、优化的一般策略（1--6）

(1) 在关系代数表达式中尽可能早地执行选择操作。

(2) 把笛卡儿积和其后的选择操作合并成F联接运算。

(3) 同时计算一连串的选择和投影操作，以免分开运算造成多次扫描文件，节省操作时间。

(4) 如在一个表达式中多次出现某个子表达式，可先对该子表达式进行计算并保存结果，以免重复计算。

(5) 适当地对关系文件进行预处理。

(6) 在计算表达式前应先估计一下怎么计算合算。

三、关系代数表达式的优化算法

输入：一个关系代数表达式的语法树。

输出：计算表达式的一个优化序列。

方法：依次执行下面每一步。

① 使用等价变换规则(4)把每个形为$\sigma_{F1\wedge\ldots\wedge Fn}(E)$的子表达式转换成选择串接形式：$\sigma_{F1}(\ldots\sigma_{Fn}(E))\ldots)$

② 对每个选择操作，使用规则(4)～(9)，尽可能把选择操作移近树的叶端（即尽可能早地执行选择操作）。

③ 对每个投影操作，使用规则(3)，(5)，(10)和(11)，尽可能把投影操作移近树的叶端。

④ 使用规则(3)～(5)，把选择和投影合并成单个选择、单个投影或一个选择后跟一个投影。

⑤ 将上述步骤得到的语法树的内结点分组。分组原则：

a、每个二元运算(×、∪、－)结点与其直接祖先(不超过别的二元运算结点）的一元运算结点(σ或π)分为一组。如果它的子孙结点一直到叶都是一元运算符(σ或π)，则也并入该组。

b、如果二元运算是笛卡儿积，而且后面不是与它组合成等值联接的选择时，则不能将选择与这个二元运算组成同一组。

⑥ 生成一个程序，每一组结点的计算是程序中的一步，各步的顺序是任意的只要保证任何一组不会在它的子孙组之前计算。

举例：　学生数据库：S(SNO, SNAME, SEX, SDEPT)

C(CNO, CNAME, TEACHER, CDEPT)

SC(SNO, CNO, GRADE)

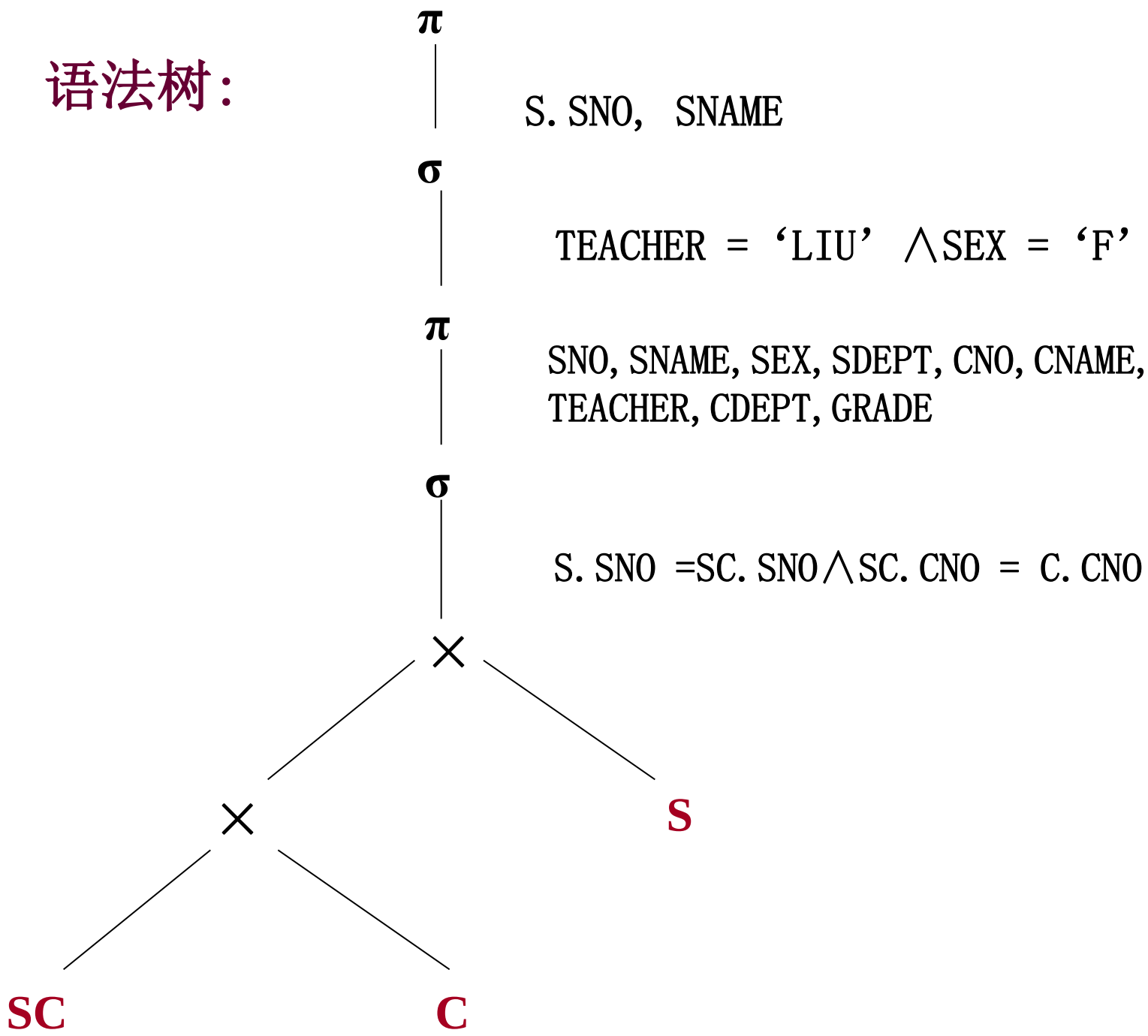
“检索选修LIU老师所授课程的女学生的学号和姓名”。

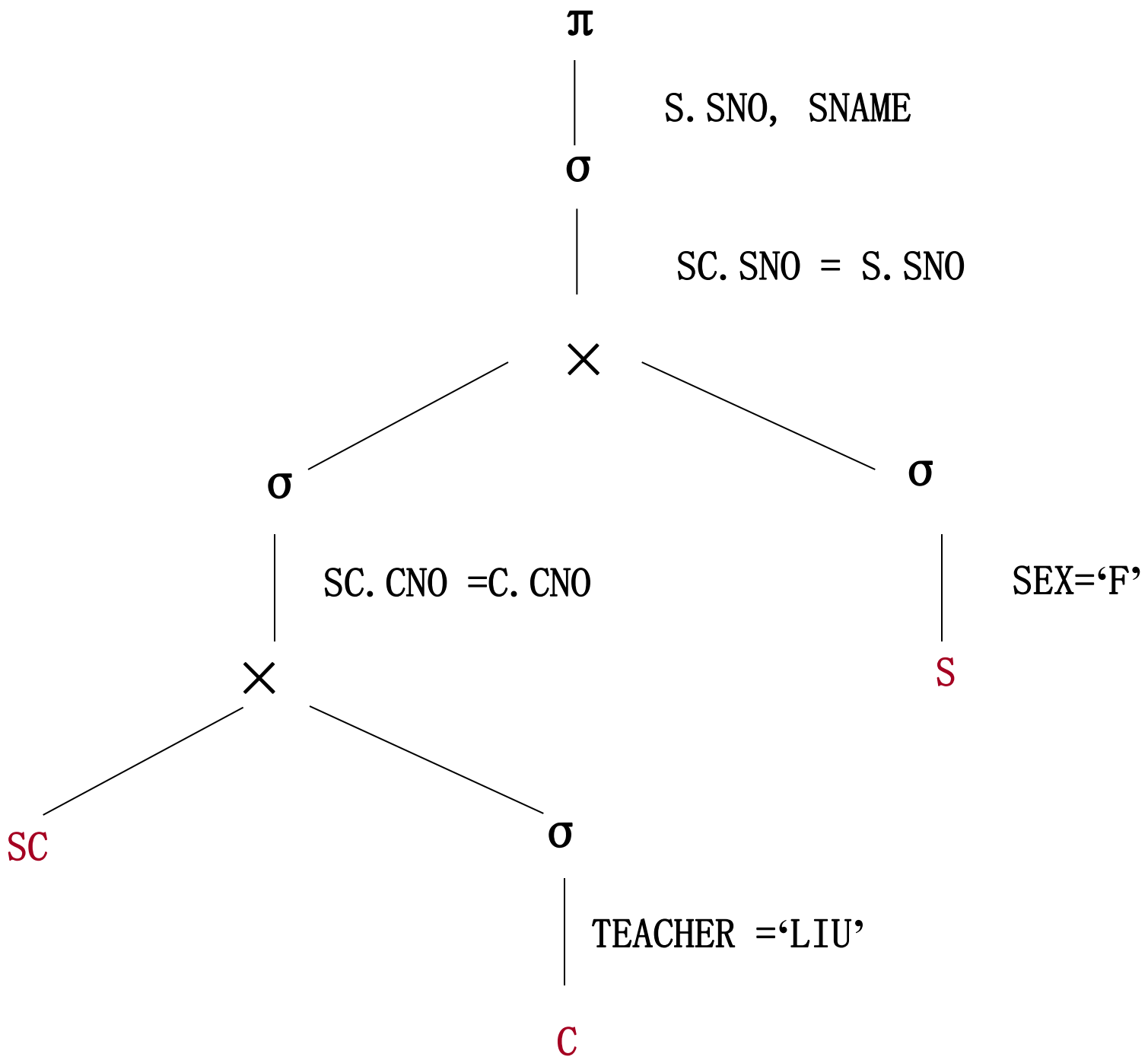
用关系代数表达式表示：

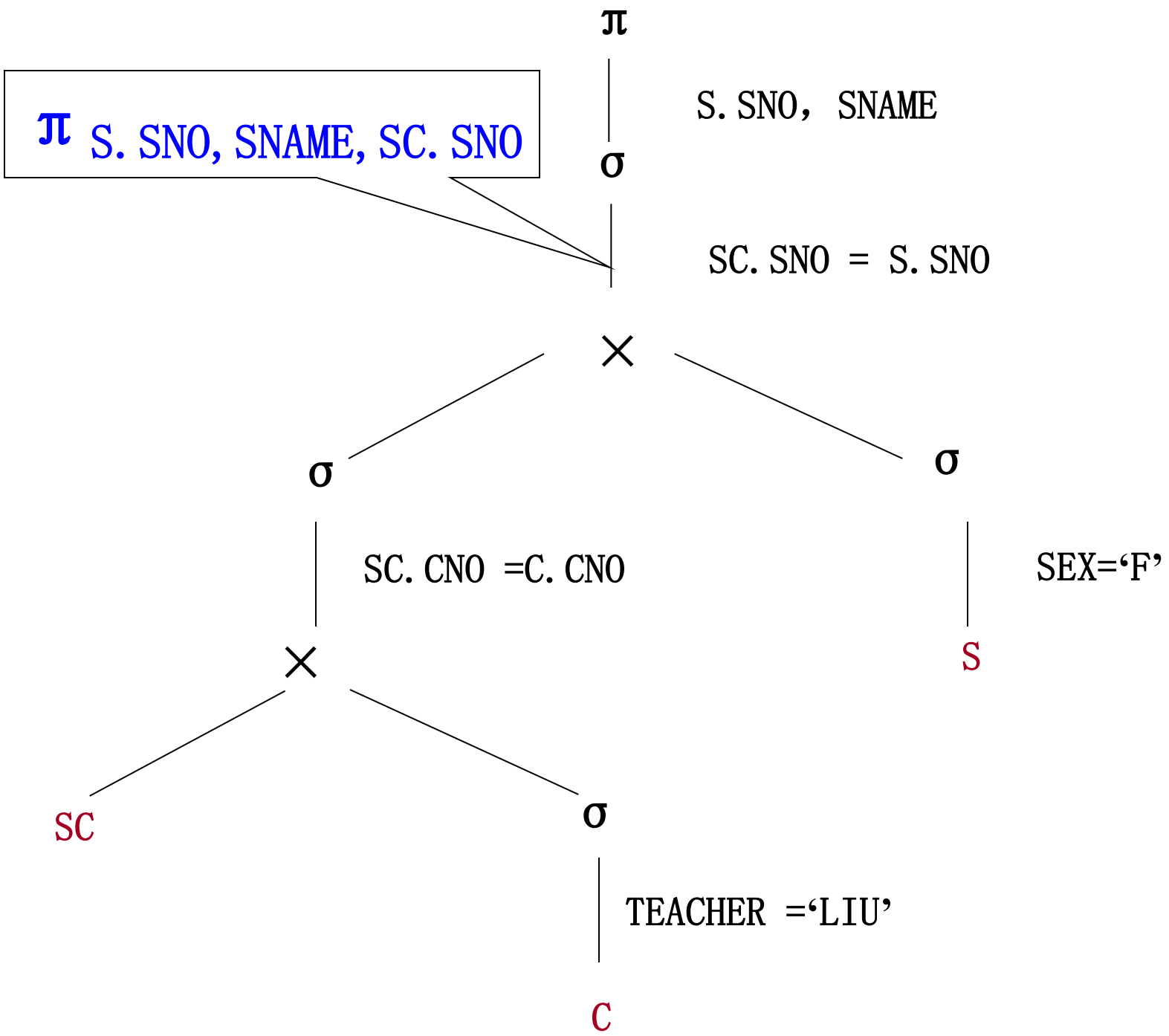
$$\pi_{SNO,SNAME}(\sigma_{TEACHER='LIU' \wedge SEX='F'}(S \bowtie SC \bowtie C))$$

该表达式构成的语法树：

语法树：


π
S.SNO, SNAME
σ
TEACHER = ‘LIU’ ∧SEX = ‘F’
π
SNO, SNAME, SEX, SDEPT, CNO, CNAME,
TEACHER, CDEPT, GRADE
σ
S.SNO =SC.SNO∧SC.CNO = C.CNO
×
×
S
SC
C


π
S.SNO, SNAME
σ
SC.SNO = S.SNO
×
σ
σ
SC.CNO =C.CNO
SEX='F'
×
S
SC
σ
TEACHER ='LIU'
C

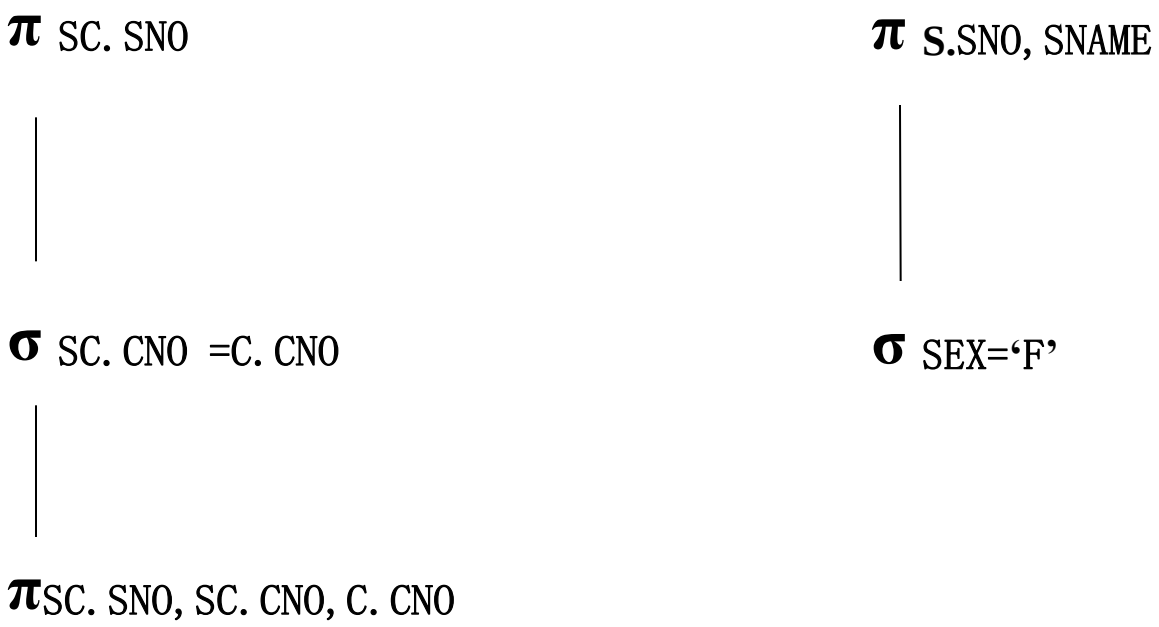

π
S. SNO, SNAME, SC. SNO
π
S. SNO, SNAME
σ
SC. SNO = S. SNO
×
σ
σ
SC. CNO =C. CNO
SEX='F'
×
S
SC
σ
TEACHER ='LIU'
C

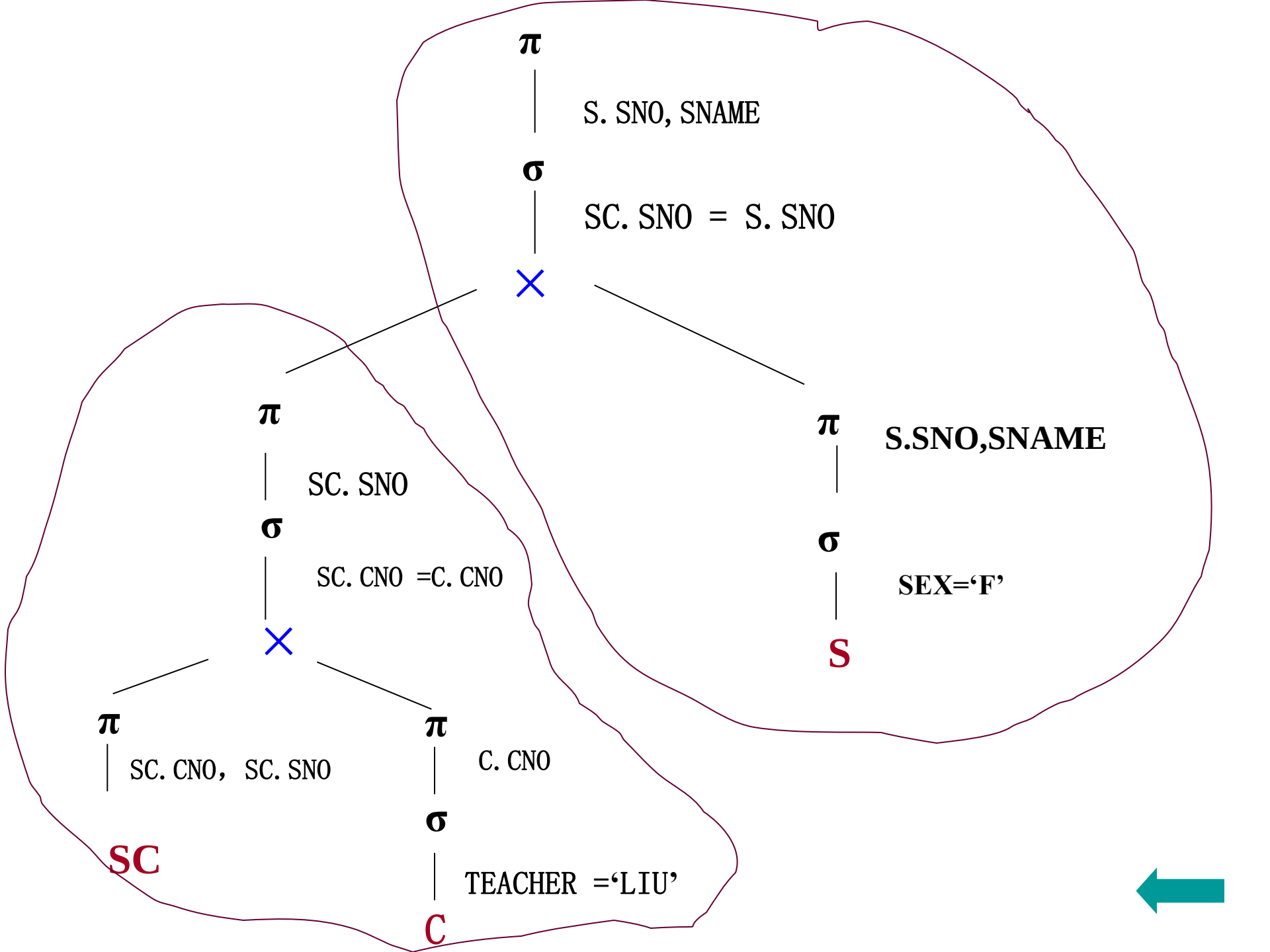
把 $\pi_{S.SNO, SNAME, SC.SNO}$ 分成：

$\pi_{SC.SNO}$ 和 $\pi_{SNO, SNAME}$

使它们分别对 $\sigma_{SC.CNO=C.CNO}(\ldots)$ 和 $\sigma_{SEX='F'}(S)$ 做投影操作。

再据规则(5)，将投影 $\pi_{SC.SNO}$ 和 $\pi_{SNO, SNAME}$ 分别与前面的选择运算形成两个串接运算：




π
S. SNO, SNAME
σ
SC. SNO = S. SNO
×
π
S.SNO,SNAME
σ
SEX='F'
S
π
SC. SNO
σ
SC. CNO =C. CNO
×
π
SC. CNO, SC. SNO
SC
π
C. CNO
σ
TEACHER ='LIU'
C

精读和习题要求

精 读： 教材 P.40 ~ P.66

习 题3： P.67 6 ~ 15