

数据库原理

The Theory of Database System

第二章 关系数据库(续)



中国矿业大学计算机学院



中国矿业大学数据库原理精品课程

第二章 关系数据库

2.1 关系

2.2 关系代数

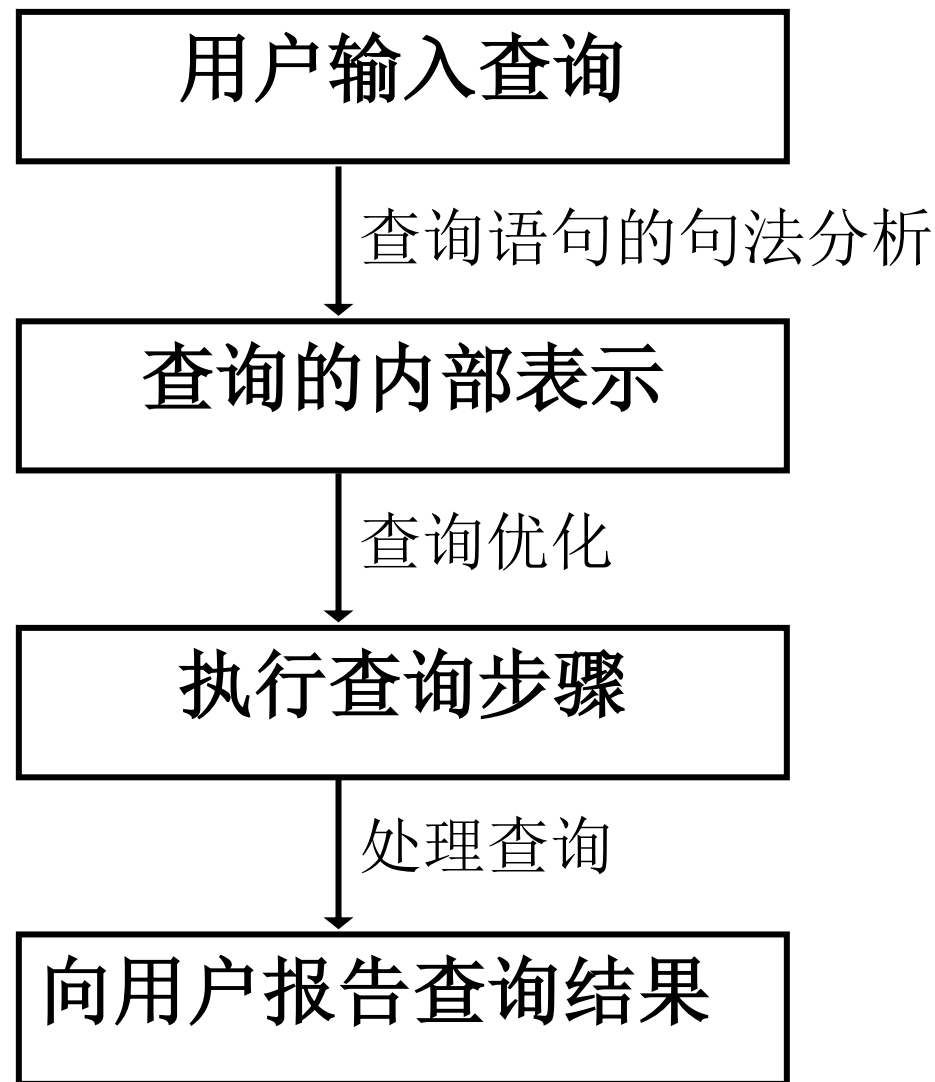
2.3 关系演算

2.4 查询优化

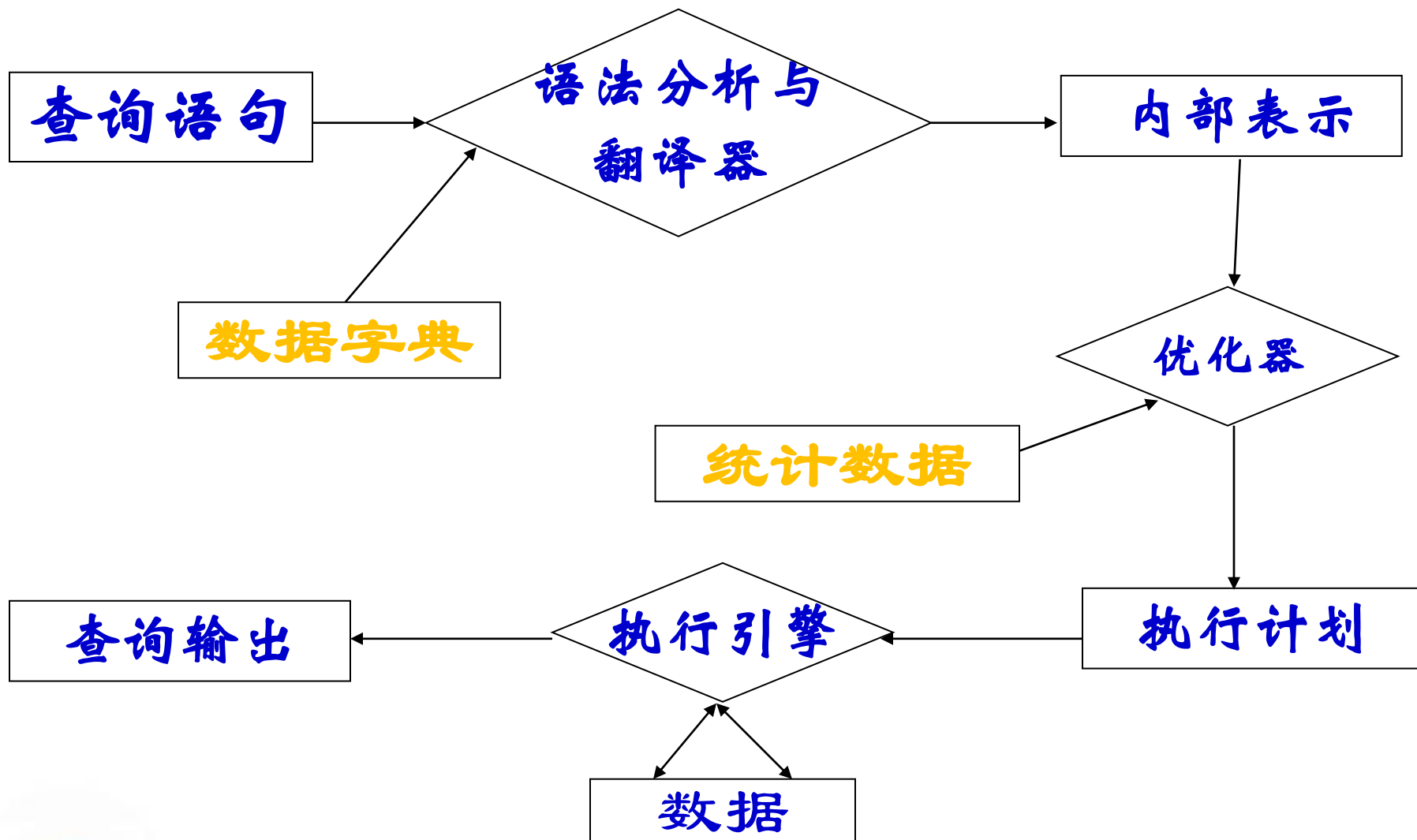
2.5 关系系统



响应用户查询的一般过程



查询优化的组织



引例

例：求选修了2号课程的学生学号和姓名。

$$Q1 = \pi_{\text{姓名}}(\sigma_{SC.\text{课程号}='2'}(S \bowtie SC))$$

$$Q2 = \pi_{\text{姓名}}(S \bowtie \sigma_{\text{课程号}='2'}(SC))$$

查询开销：

总代价 = **I/O代价** + CPU代价 + 内存代价



- 假设有1000个S记录，10000个SC记录，其中选修2号课程的有50个
- 每个物理块可放10个S元组或100个SC元组，在内存中可存放5块S元组和1块SC元组
- p 是系统传送一块所需的时间

$$p=0.005(\text{s/块})$$



第一种情况

$Q1 = \pi_{\text{姓名}}(\sigma_{S.\text{学号}=SC.\text{学号} \wedge SC.\text{课程号}='2'}(S \times SC))$

➤ 计算笛卡尔积

读取的总块数为：

$$1000/10 + (1000/(10*5)) * (10000/100) \\ = 2100 \text{块}$$

读取时间： $2100 * 0.005 = 10.5s$

➤ 写入临时文件

– 连接后的元组数为 $10^3 * 10^4 = 10^7$ ，设每块装10个元组，则写入临时文件的时间为

$$10^6 * 0.005 = 5 * 10^3 s$$



第一种情况:

➤ 作选择操作所花的时间

读取中间文件花费时间: $5 \times 10^3 \text{s}$

符合条件的50个, 均可放在内存

➤ 作投影的时间忽略不计

➤ 总I/O时间: $10.5 + 2 \times 5 \times 10^3$ 约为 10^4s

➤ 相当于2.78个小时



第二种情况

$Q2 = \pi_{\text{姓名}}(S \bowtie \sigma_{SC.\text{课程号}='2'}(SC))$

➤ 对SC作选择

读取SC时间： $10000/100 * 0.005 = 0.5s$

符合条件的元组有50个不再需要临时文件

➤ 读取S时间

只需读入S一遍， $100 * 0.005 = 0.5s$

➤ 作投影的时间忽略不计

➤ 总I/O时间： $0.5 + 0.5 = 1s$



常见查询优化途径

- 1) 代数优化：对查询表达式进行代数变换，从而改变基本查询操作的次序，提高查询语句的执行效率。
- 2) 物理优化：根据系统提供的存取路径，选择合适的存储策略，如运用关键字和索引这些特点进行查询等，这种方法依赖于存取路径的优化。
- 3) 代价估算优化：除根据一些基本规则外，还对可供选择的执行策略进行代价估算，从中选用代价最小的执行策略。



查询优化准则[策略]

- 1、尽可能早地执行选择操作（减少中间运算结果）
- 2、对关系进行预处理（索引、排序）
- 3、同时进行投影和选择运算
- 4、把投影同其前或后的双目运算结合起来。（合并连接的选择与投影操作，以减少扫描的次数）
- 5、合并选择与笛卡尔积组成一个连接
- 6、寻找公共子表达式



关系代数等价变换规则

1) 连接、笛卡尔积交换律

$$E_1 \times E_2 \equiv E_2 \times E_1$$

$$E_1 \bowtie_F E_2 \equiv E_2 \bowtie_F E_1$$

$$E_1 \bowtie E_2 \equiv E_2 \bowtie E_1$$

2) 连接、笛卡尔积结合律

$$(E_1 \times E_2) \times E_3 \equiv E_1 \times (E_2 \times E_3)$$

$$(E_1 \bowtie E_2) \bowtie E_3 \equiv E_1 \bowtie (E_2 \bowtie E_3)$$

$$(E_1 \bowtie_{F1} E_2) \bowtie_{F2} E_3 \equiv E_1 \bowtie_{F1} (E_2 \bowtie_{F2} E_3)$$



关系代数等价变换规则(续)

3、投影的串接定律

$$\pi_{A1,A2,\dots,A_n}(\pi_{B1,B2,\dots,B_m}(E)) \equiv \pi_{A1,A2,\dots,A_n}(E)$$

4、选择的串接定律

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

5、选择与投影的交换律

$$\sigma_F(\pi_{A1,A2,\dots,A_n}(E)) \equiv \pi_{A1,A2,\dots,A_n}(\sigma_F(E)) \quad (F \text{ 只涉及 } A1,A2,\dots,A_n)$$

$$\pi_{A1,A2,\dots,A_n}(\sigma_F(E)) \equiv \pi_{A1,A2,\dots,A_n}(\sigma_F(\pi_{A1,\dots,A_n,B1,\dots,B_m}(E)))$$



关系代数等价变换规则(续)

6、选择与笛卡尔积的交换律

$$\sigma_F (E_1 \times E_2) \equiv \sigma_F (E_1) \times E_2$$

$$\sigma_F (E_1 \times E_2) \equiv \sigma_{F_1} (E_1) \times \sigma_{F_2}(E_2)$$

$$\sigma_F (E_1 \times E_2) \equiv \sigma_{F_2}(\sigma_{F_1}(E_1) \times E_2)$$

7、选择与并的交换

$$\sigma_F (E_1 \cup E_2) \equiv \sigma_F (E_1) \cup \sigma_F (E_2)$$

8、选择与差的交换

$$\sigma_F (E_1 - E_2) \equiv \sigma_F (E_1) - \sigma_F (E_2)$$



9、投影与笛卡尔积的交换律

$$\pi_{A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m} (E_1 \times E_2) \equiv \pi_{A_1, A_2, \dots, A_n} (E_1) \times \pi_{B_1, B_2, \dots, B_m} (E_2)$$

10、投影与并的交换

$$\pi_{A_1, A_2, \dots, A_n} (E_1 \cup E_2) \equiv \pi_{A_1, A_2, \dots, A_n} (E_1) \cup \pi_{A_1, A_2, \dots, A_n} (E_2)$$



关系代数表达式的优化算法

输入：一个关系表达式的语法树

输出：计算该表达式的程序

方法：

1)把 $\sigma_{F_1 \wedge F_2 \wedge \dots \wedge F_n}(E)$ 变换为

[规则4]

$$\sigma_{F_1}(\sigma_{F_2}(\dots(\sigma_{F_n}(E))\dots))$$

2)对每一个选择尽可能把它移到树的叶端。

[规则4-8]

3)对每一个投影尽可能把它移到树的叶端。

[规则3,5,9,10]

4)合并选择和投影或一个选择后跟一个投影。

[规则3-5]

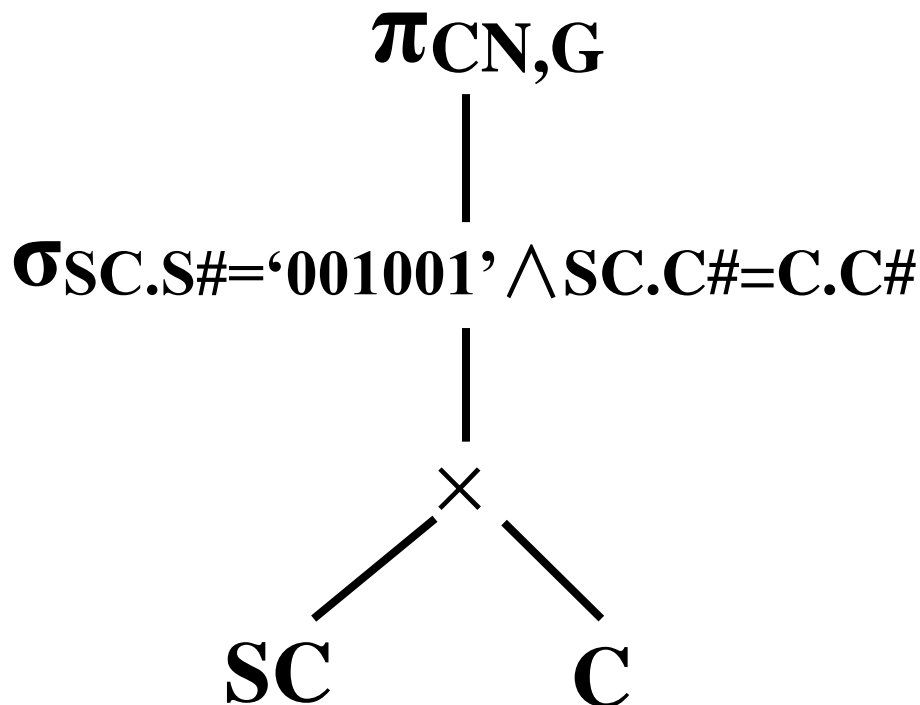
5)将得到的语法树的内节点分组。(每一双目运算和它所有的直接祖先为一组。

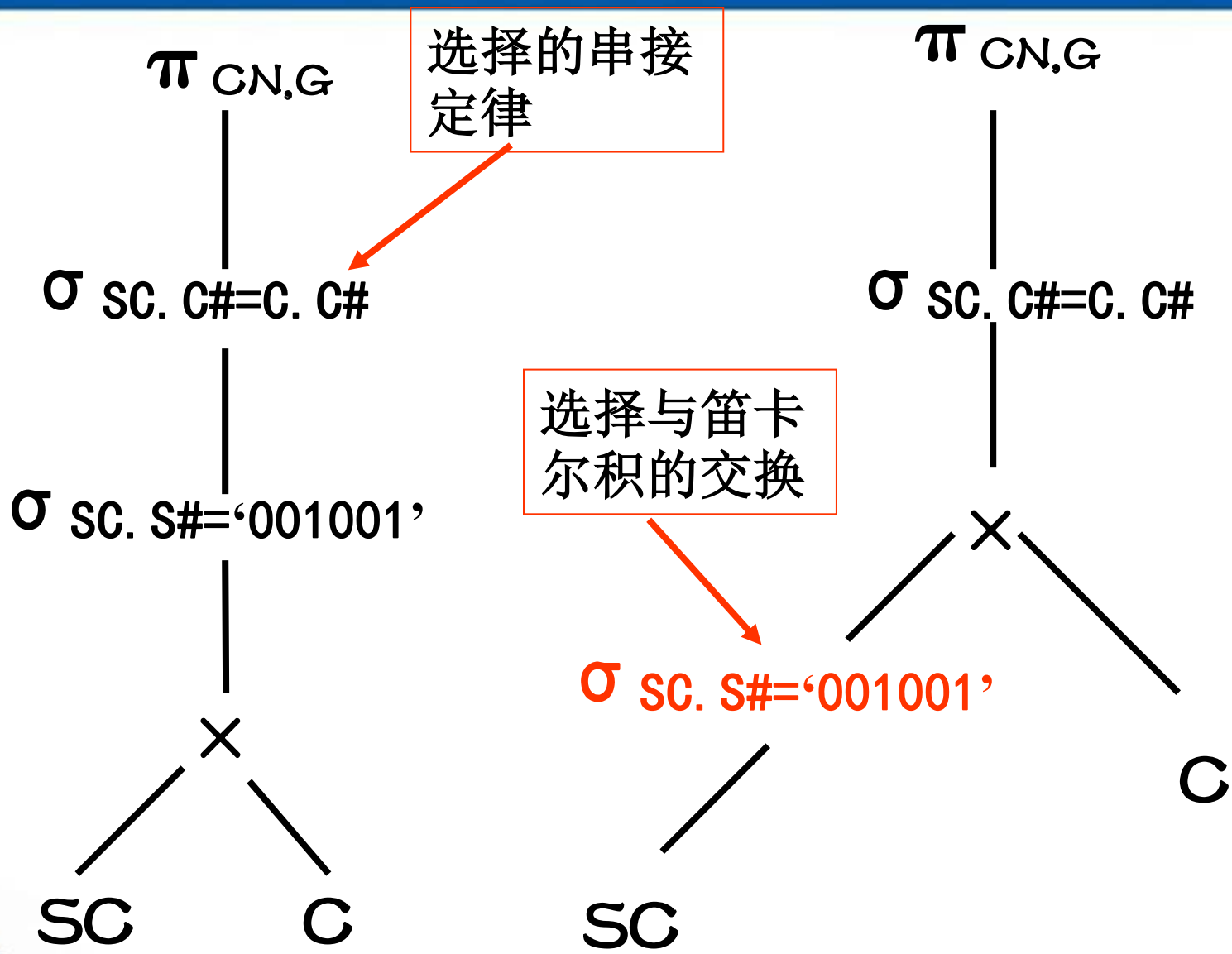
6)生成一个程序，每组节点的计算是程序中的一步。求值顺序为先子孙，后祖先。



例：求001001号学生所选修的课程名及成绩

$\pi_{CN,G} (\sigma_{SC.S\#='001001' \wedge SC.C\#=C.C\#} (SC \times C))$

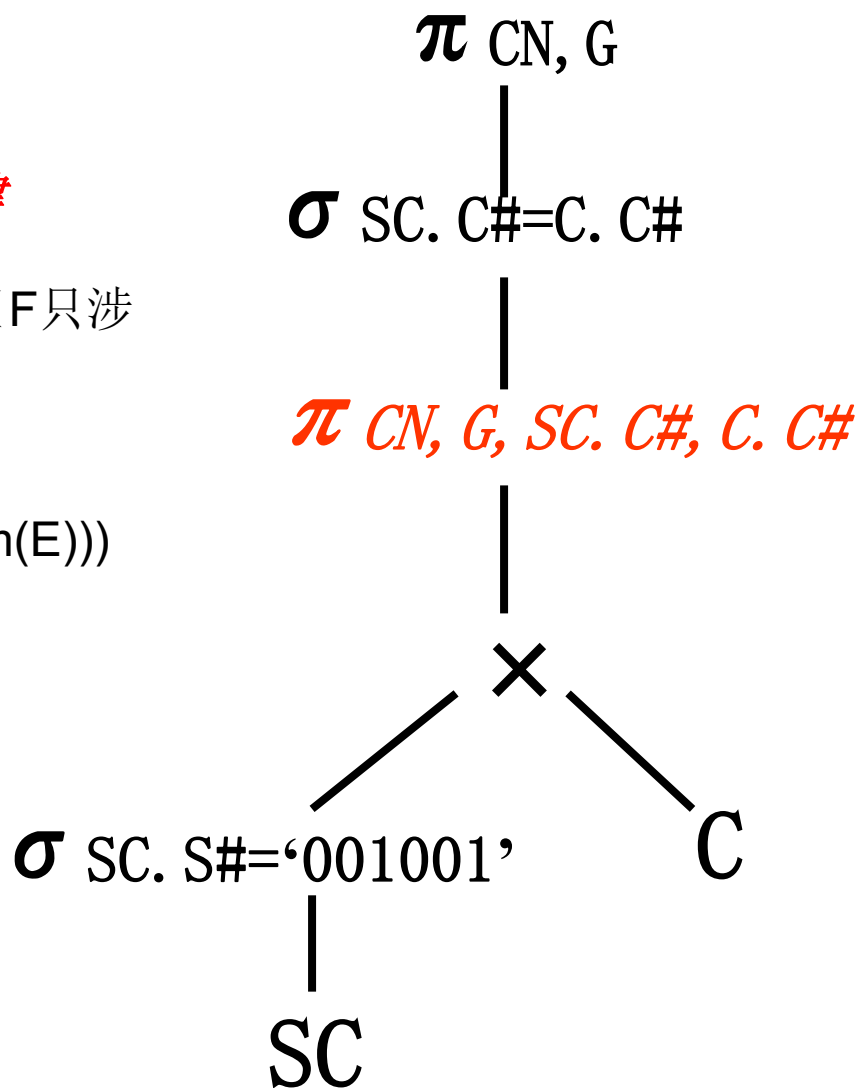


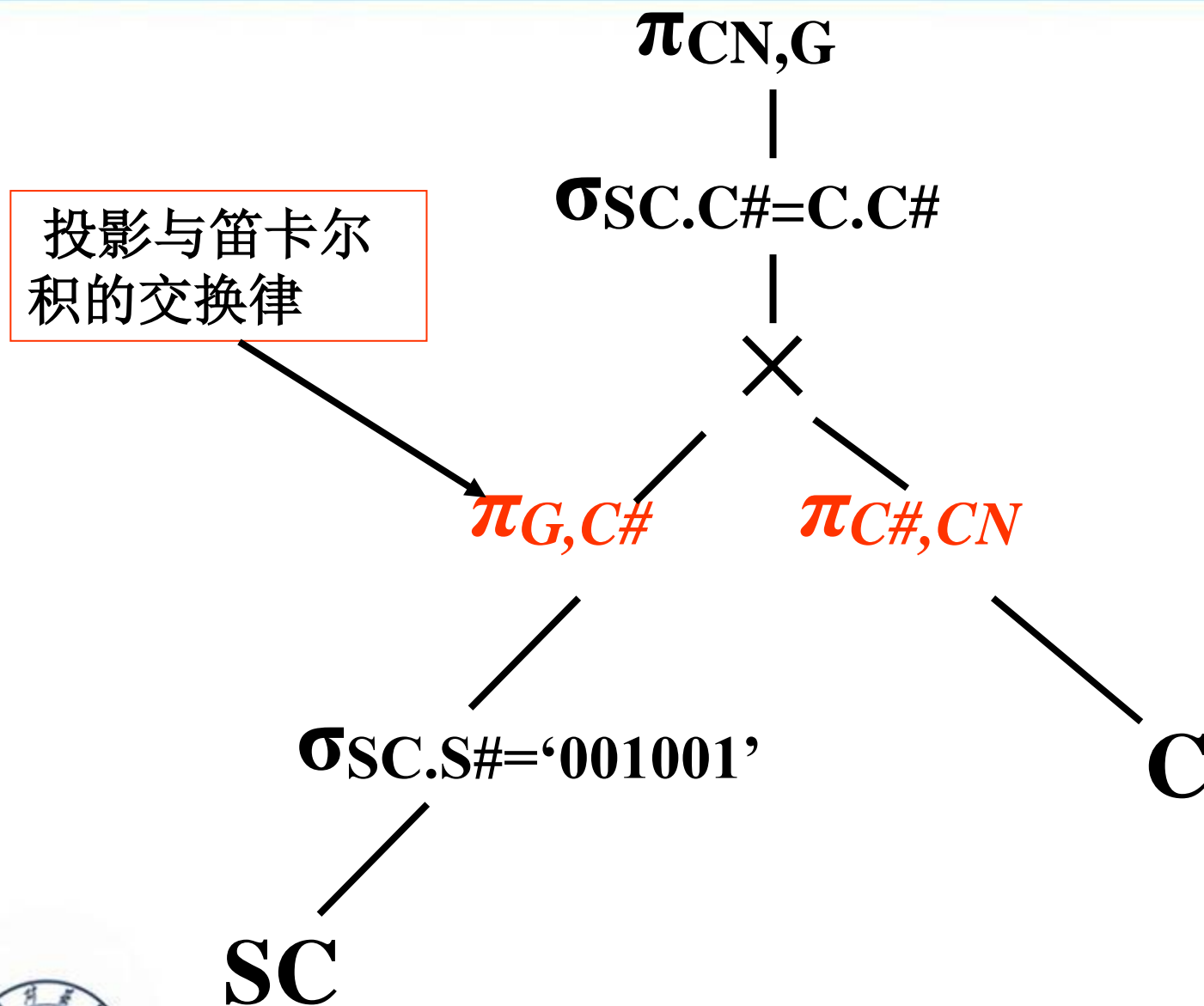


5、选择与投影的交换律

$\sigma_F(\pi_{A_1, A_2, \dots, A_n}(E)) \equiv$
 $\pi_{A_1, A_2, \dots, A_n}(\sigma_F(E))$ (F只涉
 及 A_1, A_2, \dots, A_n)

$\pi_{A_1, A_2, \dots, A_n}(\sigma_F(E))$
 $\equiv \pi_{A_1, A_2, \dots, A_n}$
 $\sigma_F(\pi_{A_1, \dots, A_n, B_1, \dots, B_m}(E))$





例：查询选修了数据库原理的学生姓名和成绩

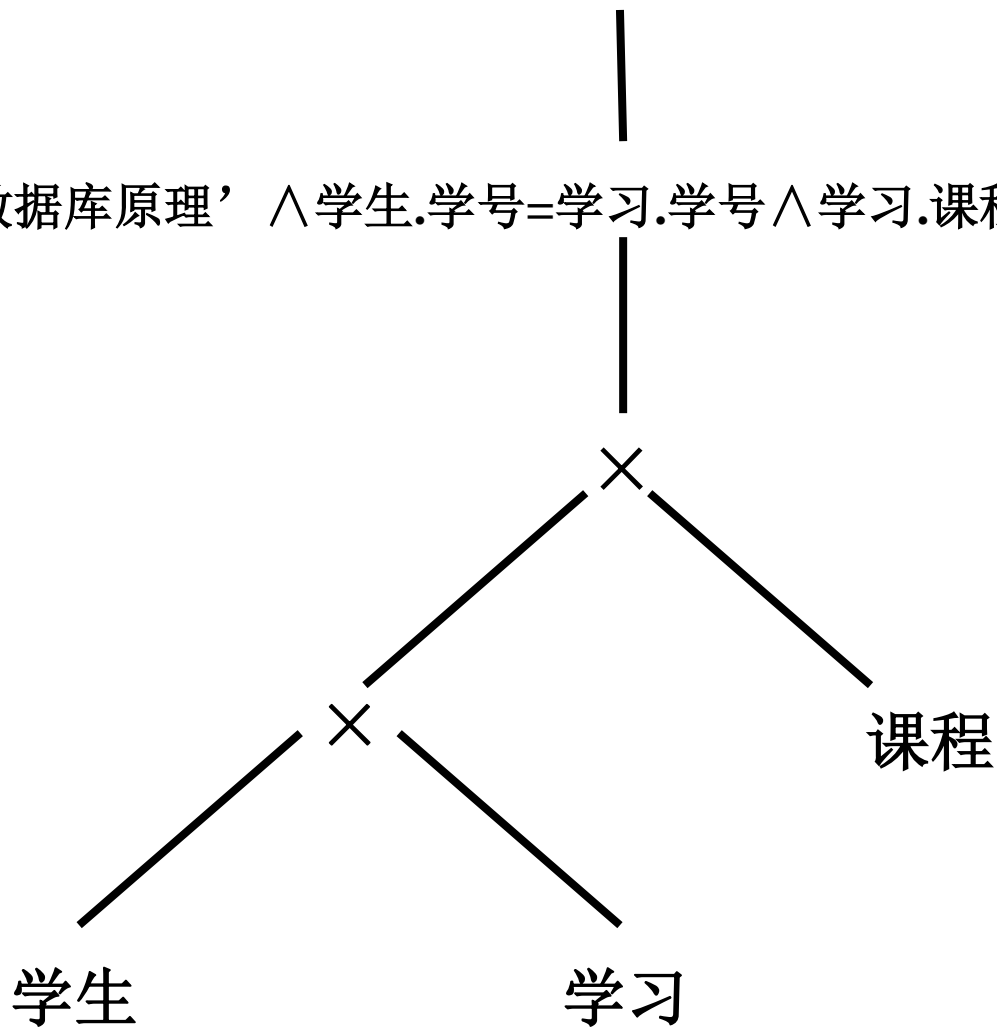
$\pi_{\text{姓名,成绩}} \left(\sigma_{\text{课程名}='数据库原理'} \left(\text{学生} \bowtie \text{学习} \right. \right.$
 $\left. \left. \bowtie \text{课程} \right) \right)$

$\pi_{\text{姓名,成绩}} \left(\sigma_{\text{课程名}='数据库原理'} \wedge \text{学生.学号}=\text{学习.学号} \right.$
 $\left. \wedge \text{学习.课程号}=\text{课程.课程号} \left(\text{学生} \times \text{学习} \times \text{课程} \right) \right)$



Π 姓名,成绩

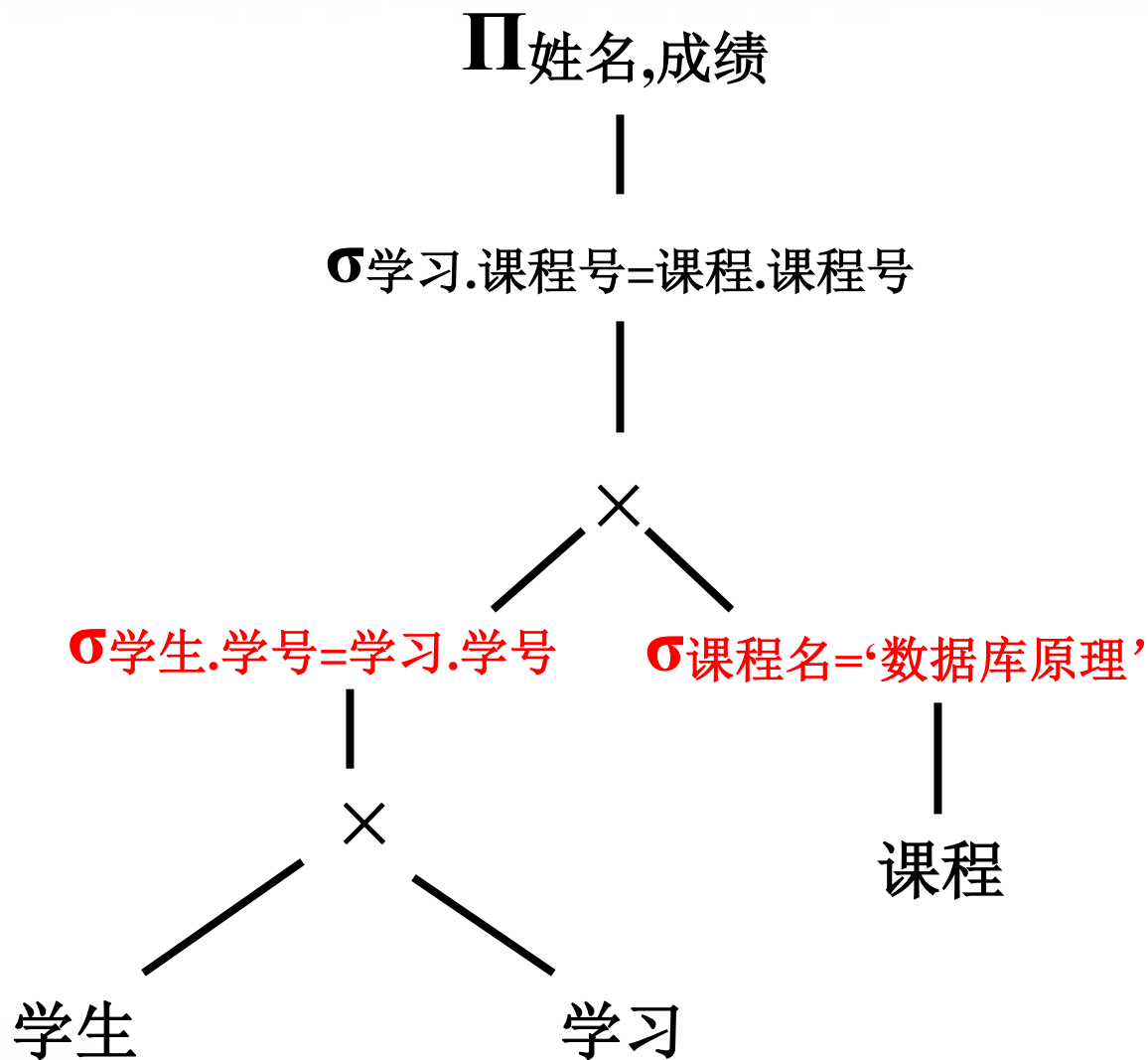
σ 课程名='数据库原理' \wedge 学生.学号=学习.学号 \wedge 学习.课程号=课程.课程号



规则 4、6

选择的串接

选择和笛卡
尔积交换



Π 姓名,成绩

σ 学习.课程号=课程.课程号

Π 姓名,成绩,学习.课程号,课程.课程号

σ 学生.学号=学习.学号

σ 课程名='数据库原理'

课程

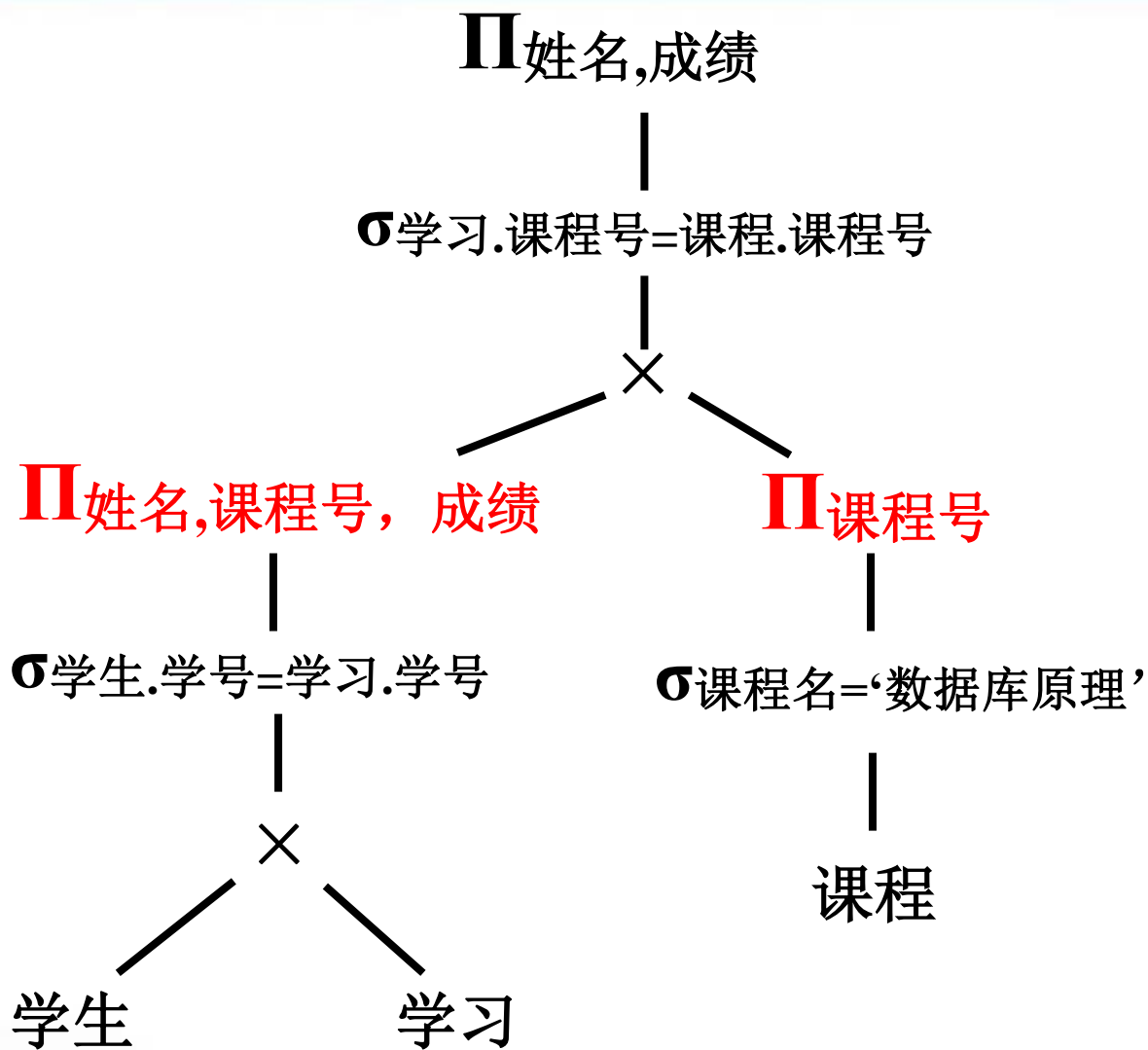
学生

学习

规则 5

选择和投影
交换





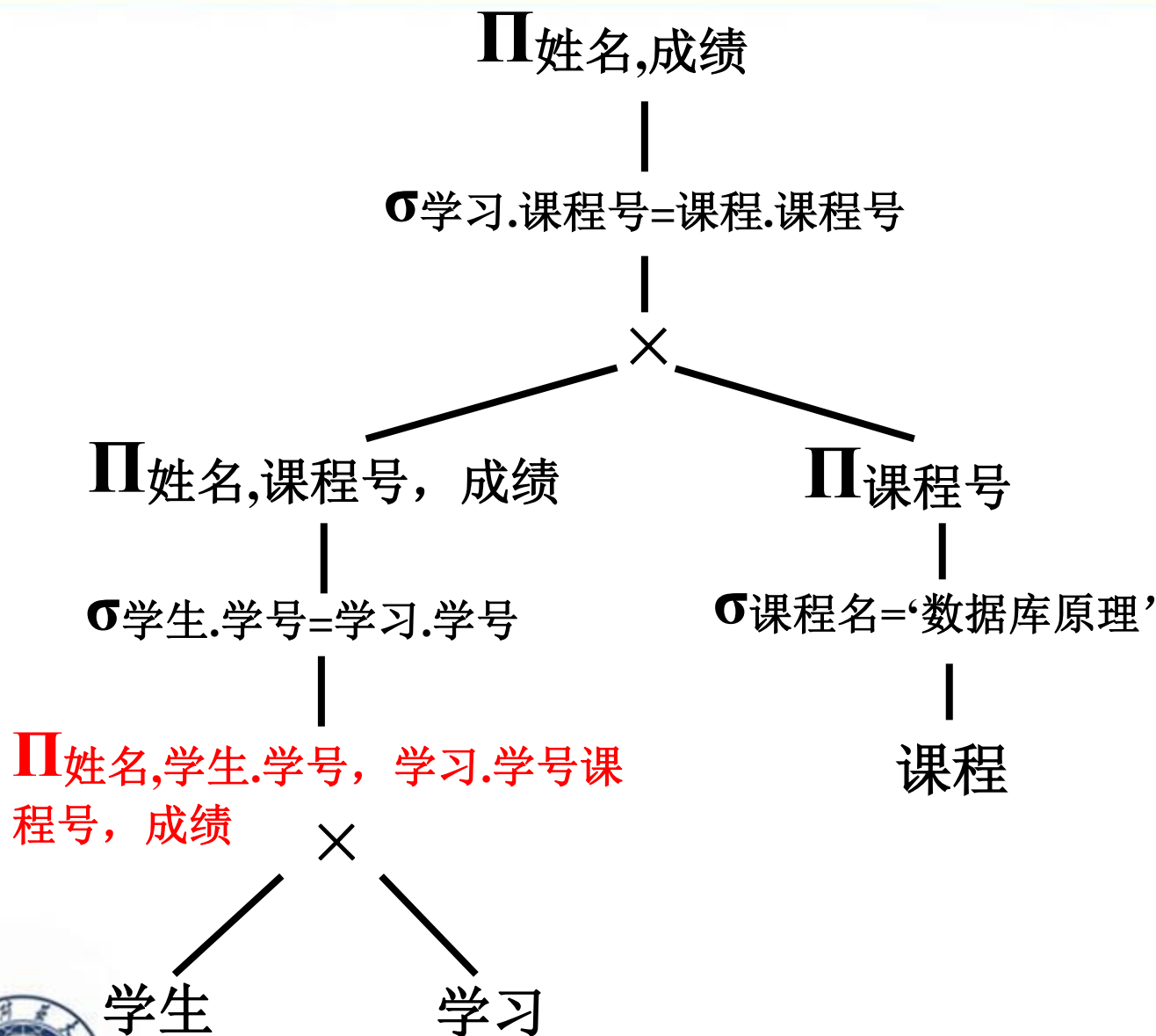
规则 9

投影和笛卡尔积交换



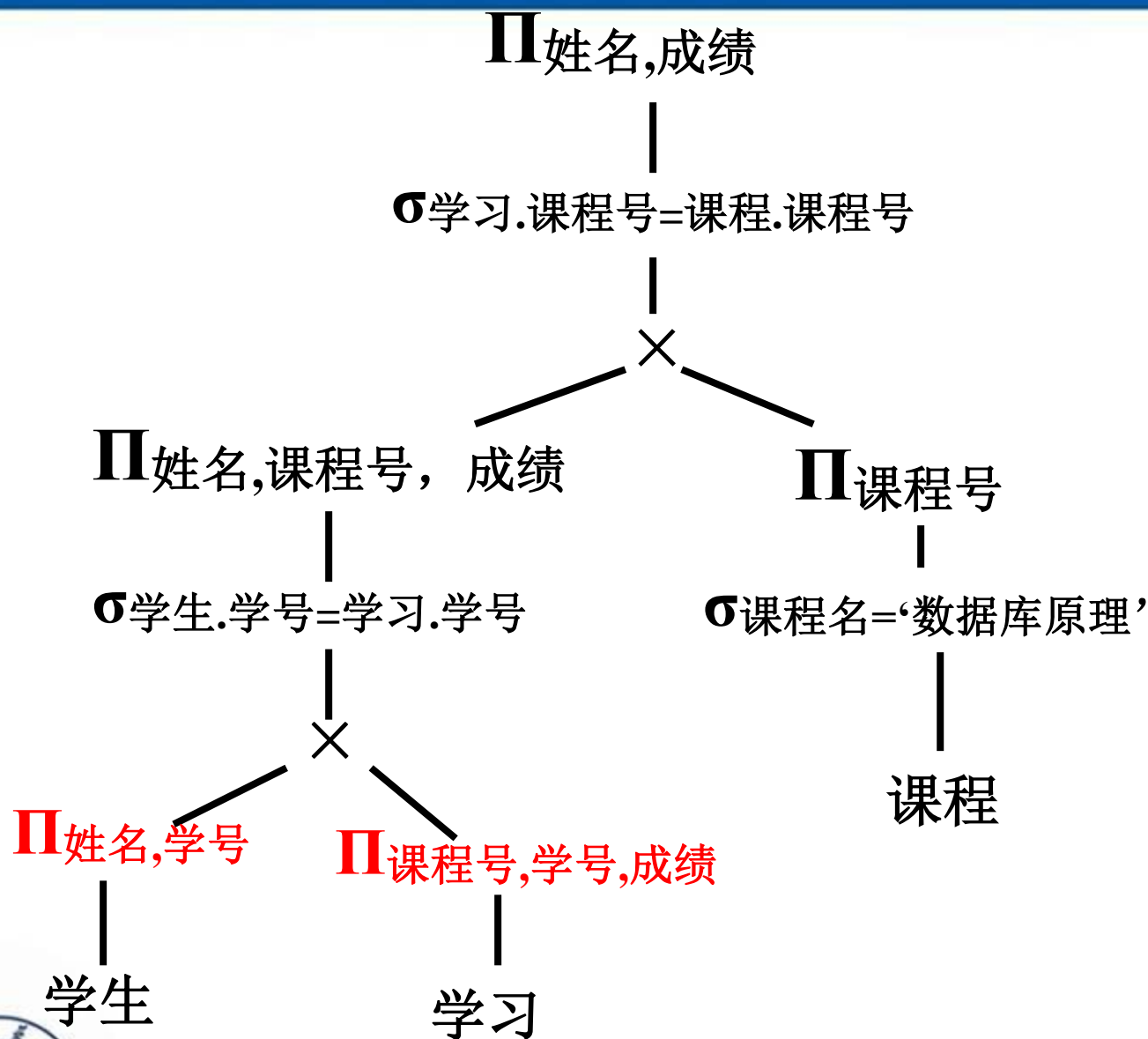
规则 5

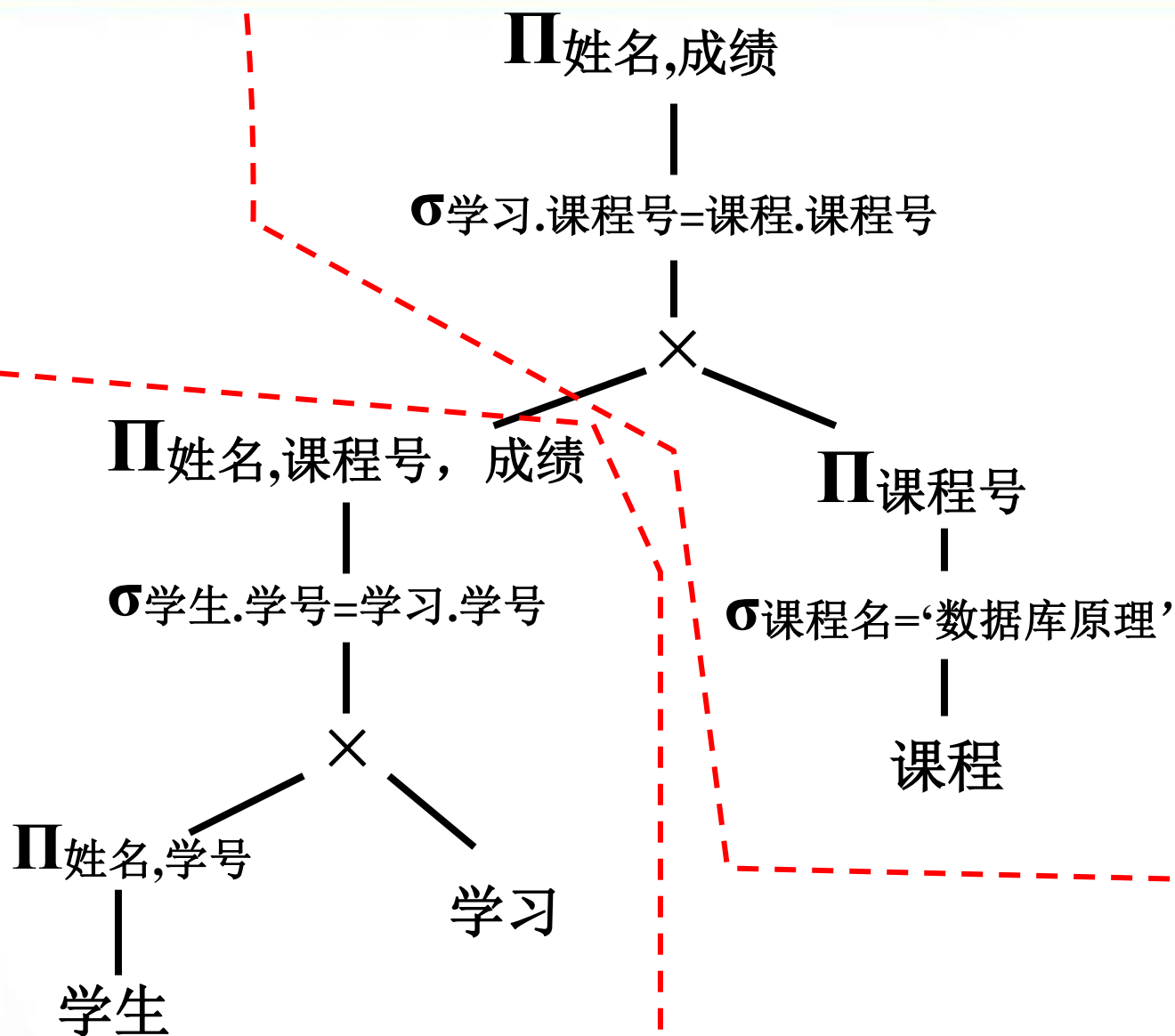
选择和投影
交换



规则 9

投影和笛卡尔积交换





优化后的表达式

$\Pi_{\text{姓名,成绩}} \left(\Pi_{\text{课程号,成绩,姓名}} \left(\Pi_{\text{学号,姓名}} \left(\text{学生} \right) \right) \right) \bowtie \text{学习} \bowtie \Pi_{\text{课程号}} \left(\sigma_{\text{课程名}='数据库原理'} \left(\text{课程} \right) \right)$



练习

学生（学号，姓名，性别，专业）

课程（课程号，课程名，先行课，学分）

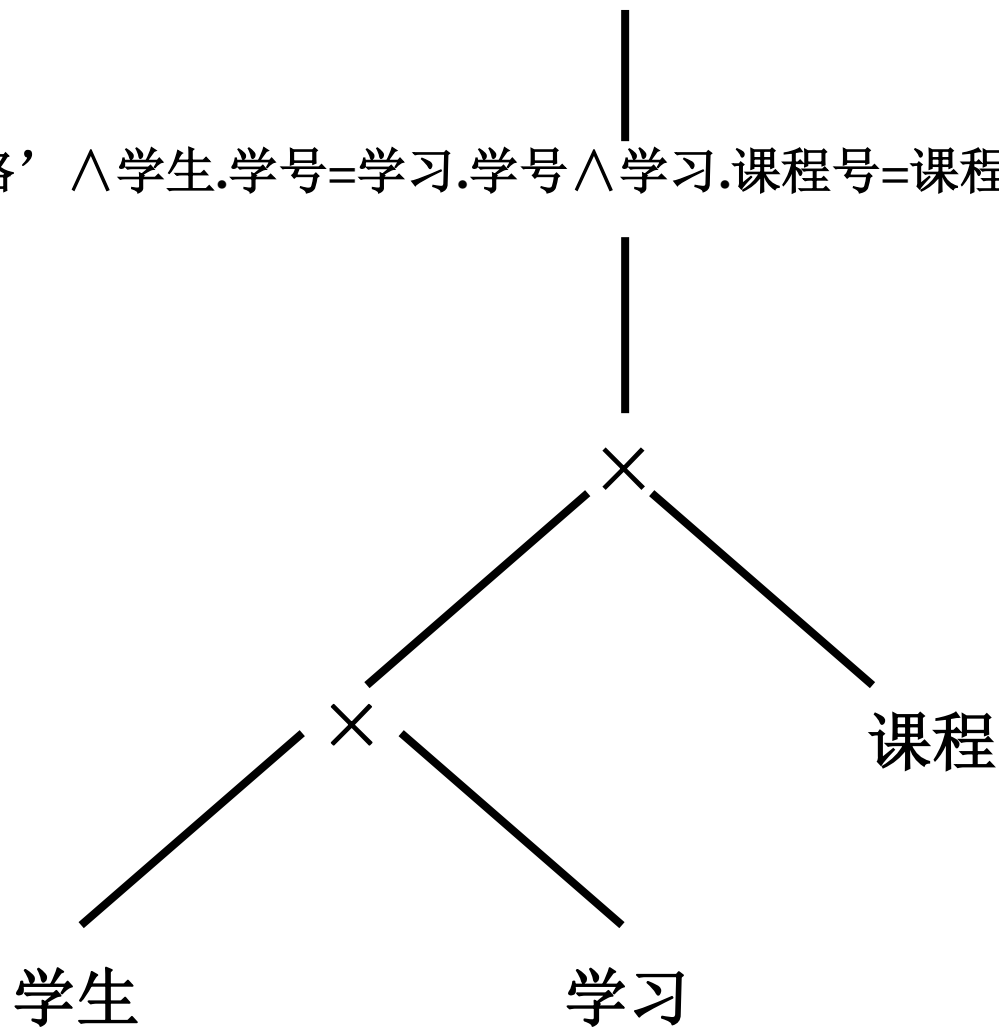
选修（学号，课程号，成绩）

- 1、写出“**查询网络方向学生的学号及其选修的课程名称和成绩**”关系代数表达式
- 2、画出该代数表达式的语法树
- 3、对该语法树进行优化
- 4、写出优化后的代数表达式



II 学生.学号, 课程名, 成绩

σ专业='网络' ∧ 学生.学号=学习.学号 ∧ 学习.课程号=课程.课程号



Π 学生.学号, 课程名, 成绩

σ 学习.课程号=课程.课程号

σ 学生.学号=学习.学号

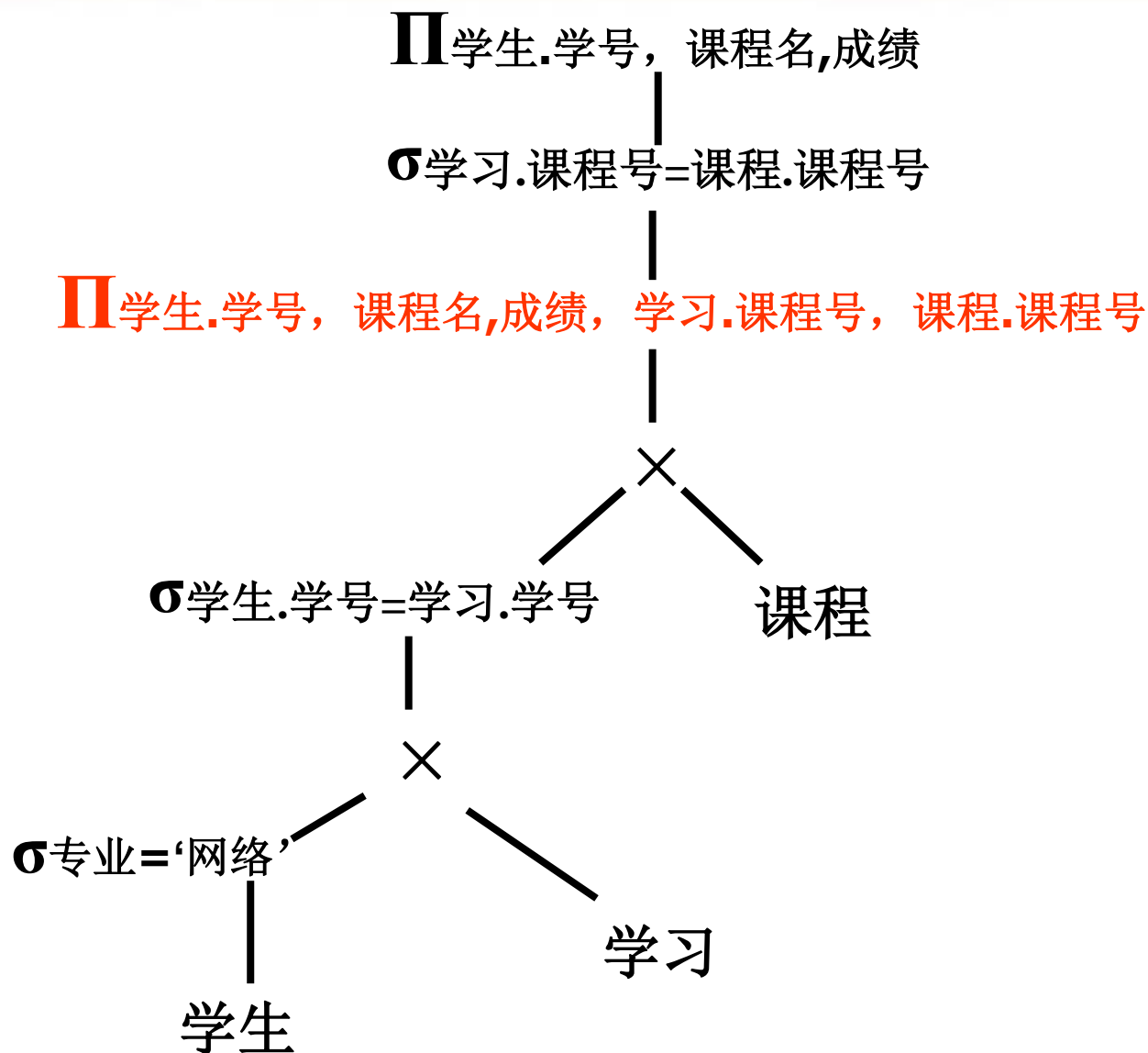
课程

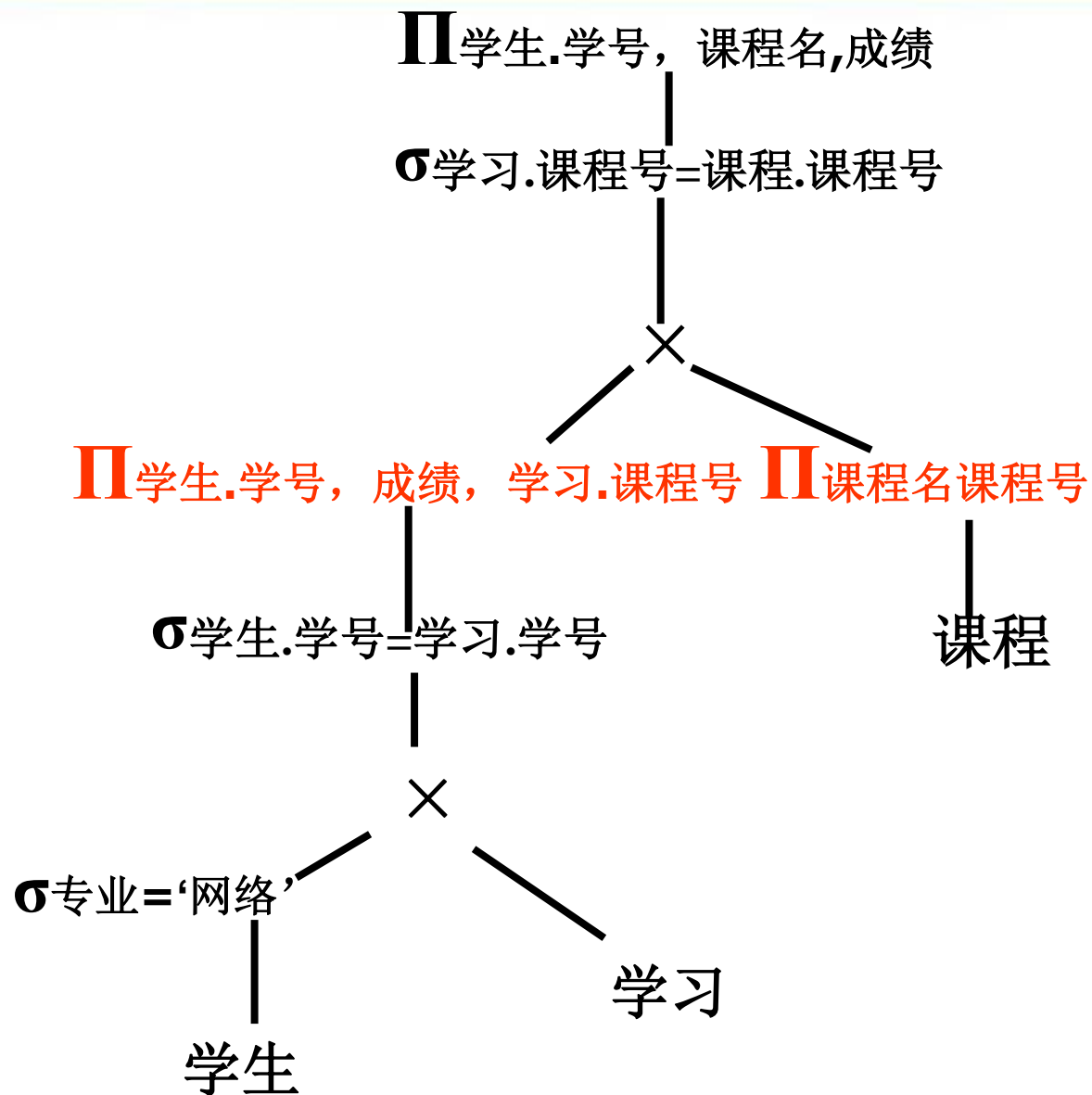
σ 专业='网络'

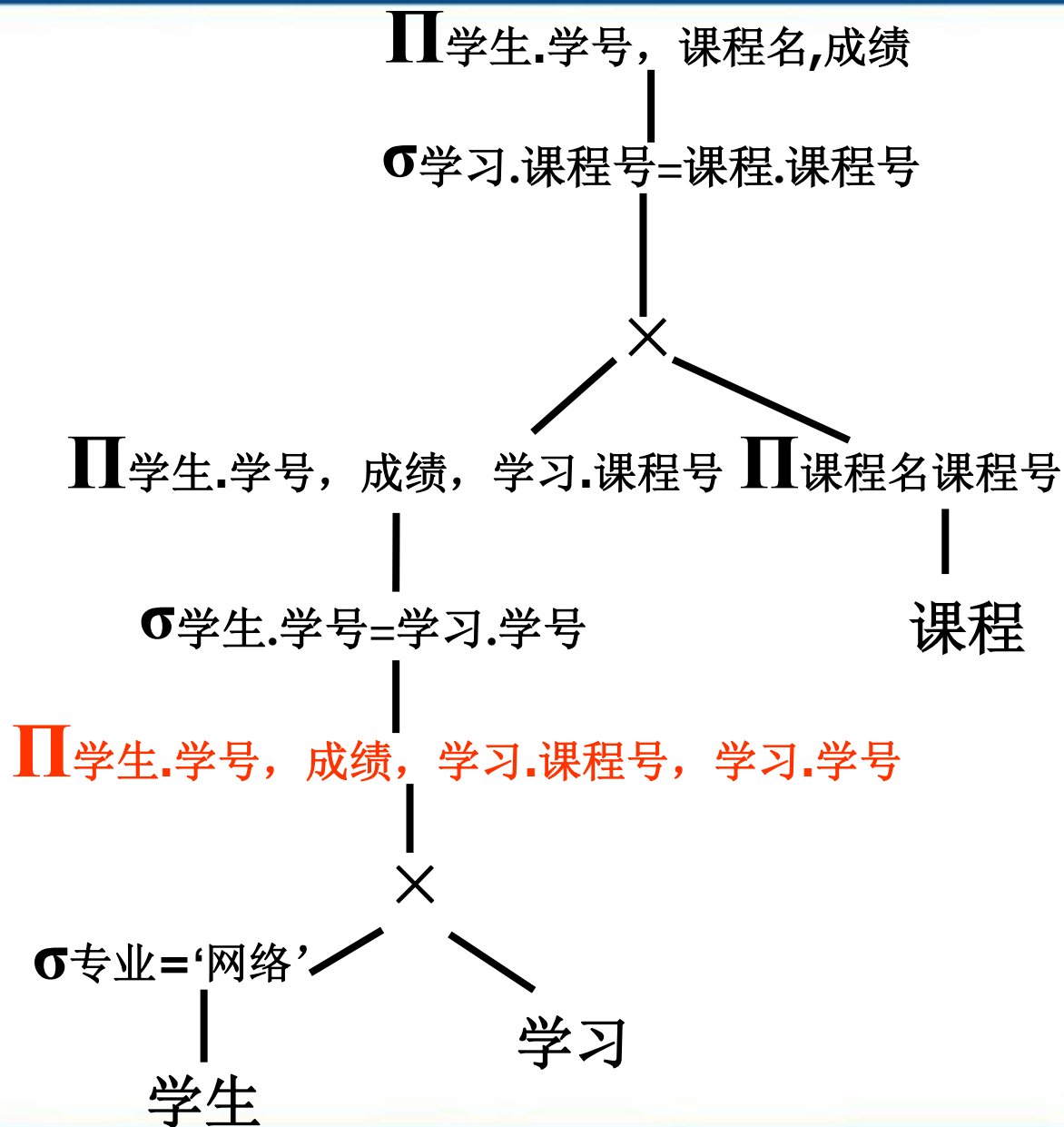
学习

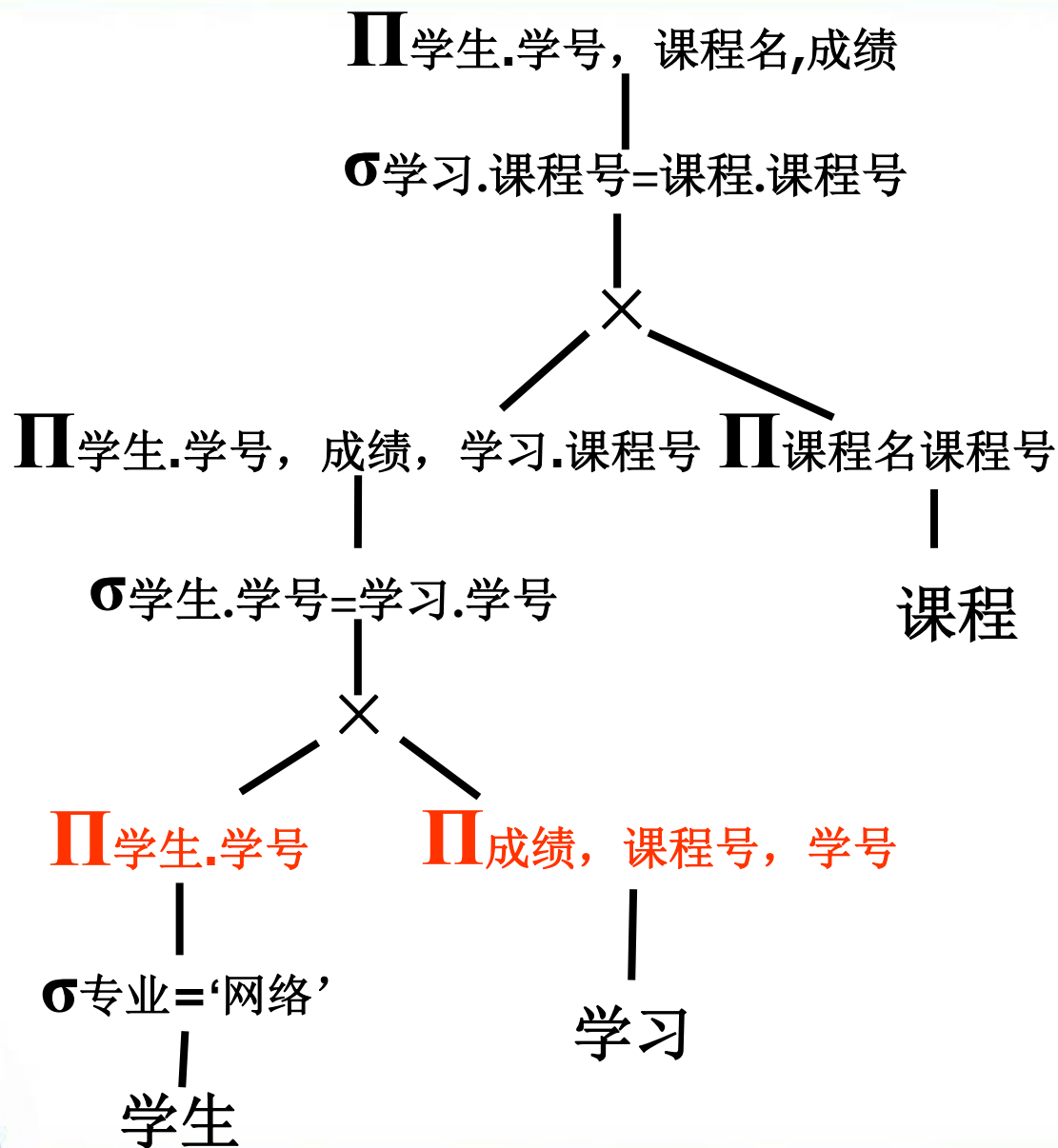
学生

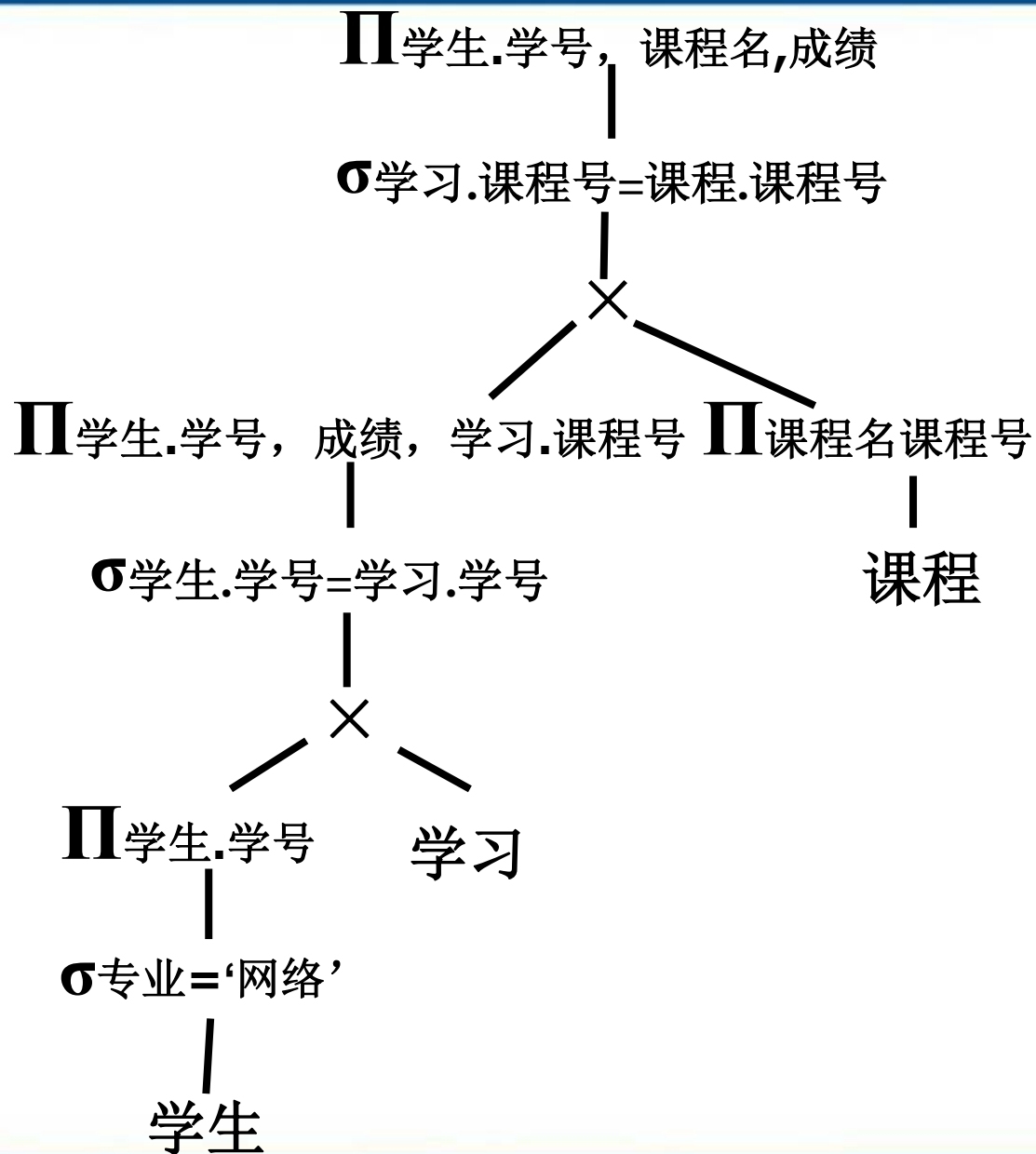












优化后的关系代数表达式

$\Pi_{\text{学生.学号}, \text{课程名}, \text{成绩}} \left(\Pi_{\text{学号}, \text{成绩}, \text{课程号}} \left(\Pi_{\text{学号}} \left(\sigma_{\text{专业}='网络'} (\text{学生}) \right) \right) \right) \bowtie \text{学习} \bowtie \Pi_{\text{课程号}, \text{课程名}} (\text{课程})$



2.5 关系系统

2.5.1 关系系统的定义

一个系统可定义为关系系统，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。

对这些运算不必要求定义任何物理存取路径。



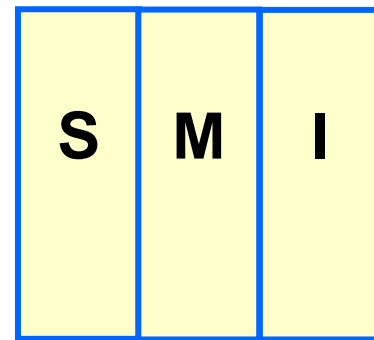
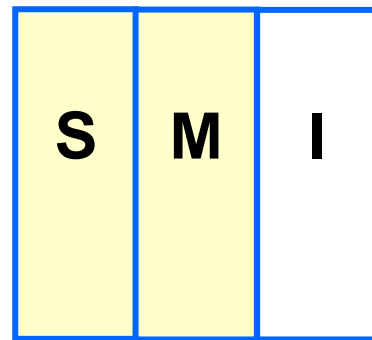
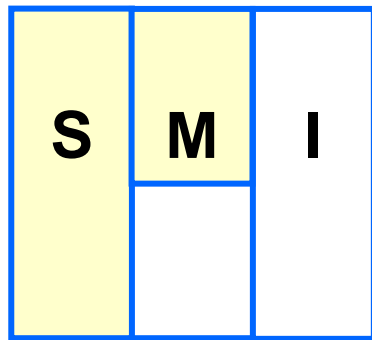
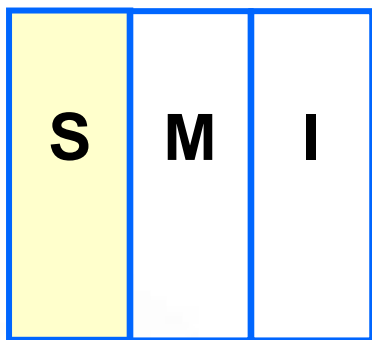
2.5.2 关系系统的分类

- 1) 表式系统：仅支持数据结构。
- 2) (最小) 关系系统：数据结构+三种关系操作。
- 3) 关系完备的系统：数据结构+所有关系代数操作。
- 4) 全关系系统：支持关系模型的所有特征。

S: 结构

M: 数据操纵

I: 完整性



本章总结

- **掌握关系的数学概念**
- **重点掌握关系代数**
- **重点掌握查询优化**
- **掌握关系系统及其分类**

