

# 第9章 关系查询处理和查询优化

9.1 关系数据库系统的查询处理

9.2 关系数据库系统的查询优化

9.3 代数优化

9.4 物理优化

## 9.3 代数优化

### 9.3.1 关系代数表达式等价变换规则

### 9.3.2 查询树的启发式优化

### 9.3.1 关系代数表达式等价变换规则

- 代数优化策略：通过对关系代数表达式的等价变换来提高查询效率
- 关系代数表达式的等价：指用相同的关系代替两个表达式中相应的关系所得到的结果是相同的
- 两个关系表达式 $E_1$ 和 $E_2$ 是等价的，可记为 $E_1 \equiv E_2$



## ❖ 常用的等价变换规则：

- 1. 连接、笛卡尔积交换律
- 2. 连接、笛卡尔积的结合律
- 3. 投影的串接定律
- 4. 选择的串接定律
- 5. 选择与投影操作的交换律
- 6. 选择与笛卡尔积的交换律
- 7. 选择与并的分配律
- 8. 选择与差运算的分配律
- 9. 选择对自然连接的分配律
- 10. 投影与笛卡尔积的分配律
- 11. 投影与并的分配律

## 9.3 代数优化

### 9.3.1 关系代数表达式等价变换规则

### 9.3.2 查询树的启发式优化

## 9.3.2 查询树的启发式优化

- 典型的启发式规则

- (1) 选择运算应尽可能先做

在优化策略中这是最重要、最基本的一条。

- (2) 把投影运算和选择运算同时进行

如有若干投影和选择运算，并且它们都对同一个关系操作，则可以在扫描此关系的同时完成所有的这些运算以避免重复扫描关系。



(3) 把投影同其前或其后的双目运算结合起来，没有必要为了去掉某些字段而扫描一遍关系。

(4) 把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算，连接特别是等值连接运算要比同样关系上的笛卡尔积省很多时间。

## (5) 找出公共子表达式

- 如果这种重复出现的子表达式的结果不是很大的关系
- 并且从外存中读入这个关系比计算该子表达式的时间少得多
- 则先计算一次公共子表达式并把结果写入中间文件是合算的。
- 当查询的是视图时，定义视图的表达式就是公共子表达式的情况



## ❖ 遵循这些启发式规则，应用9.3.1的等价变换公式来优化关系表达式的算法。

算法：关系表达式的优化

输入：一个关系表达式的查询树

输出：优化的查询树

方法：

**规则4：选择的串接定律**

$$\sigma_{F_1}(\sigma_{F_2}(E)) \equiv \sigma_{F_1 \wedge F_2}(E)$$

(1) 利用等价变换规则4把形如 $\sigma_{F_1 \wedge F_2 \wedge \dots \wedge F_n}(E)$ 变换为 $\sigma_{F_1}(\sigma_{F_2}(\dots(\sigma_{F_n}(E))\dots))$ 。

(2) 对每一个选择，利用等价变换规则4 ~ 9尽可能把它移到树的叶端。

**规则4：合并或分解选择运算**

**规则5-9：选择运算与其他运算交换**

(3) 对每一个投影利用等价变换规则3, 5, 10, 11中的一般形式尽可能把它移向树的叶端。

■注意：

规则3： 合并或分解投影运算  
规则5,10,11： 投影运算与其他运算交换

- 等价变换规则3使一些投影消失或使一些投影出现
- 规则5把一个投影分裂为两个，其中一个有可能被移向树的叶端

(4) 利用等价变换规则3 ~ 5，把选择和投影的串接合并成单个选择、单个投影或一个选择后跟一个投影，使多个选择或投影能同时执行，或在一次扫描中全部完成

规则3： 合并或分解投影运算  
规则4： 合并或分解选择运算  
规则5： 投影运算与选择运算交换

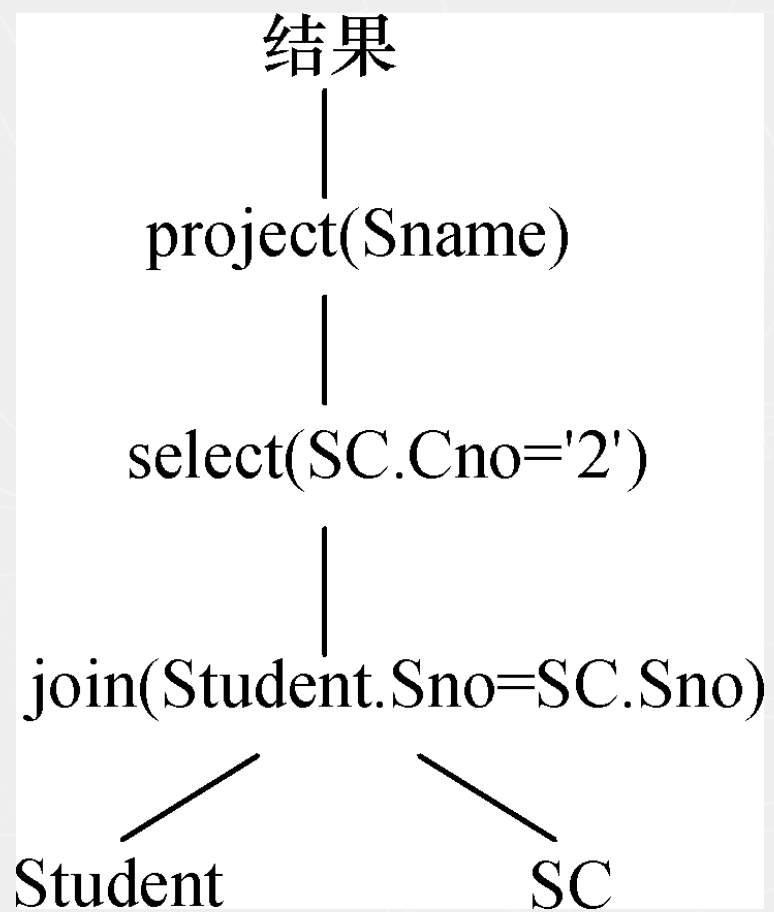
(5) 把上述得到的语法树的内节点分组。

- 每一双目运算( $\times$ ,  $\bowtie$ ,  $\cup$ ,  $-$ )和它所有的直接祖先为一组(这些直接祖先是( $\sigma$ ,  $\pi$ 运算))。
- 如果其后代直到叶子全是单目运算, 则也将它们并入该组
- 但当双目运算是笛卡尔积( $\times$ ), 而且后面不是与它组成等值连接的选择时, 则不能把选择与这个双目运算组成同一组

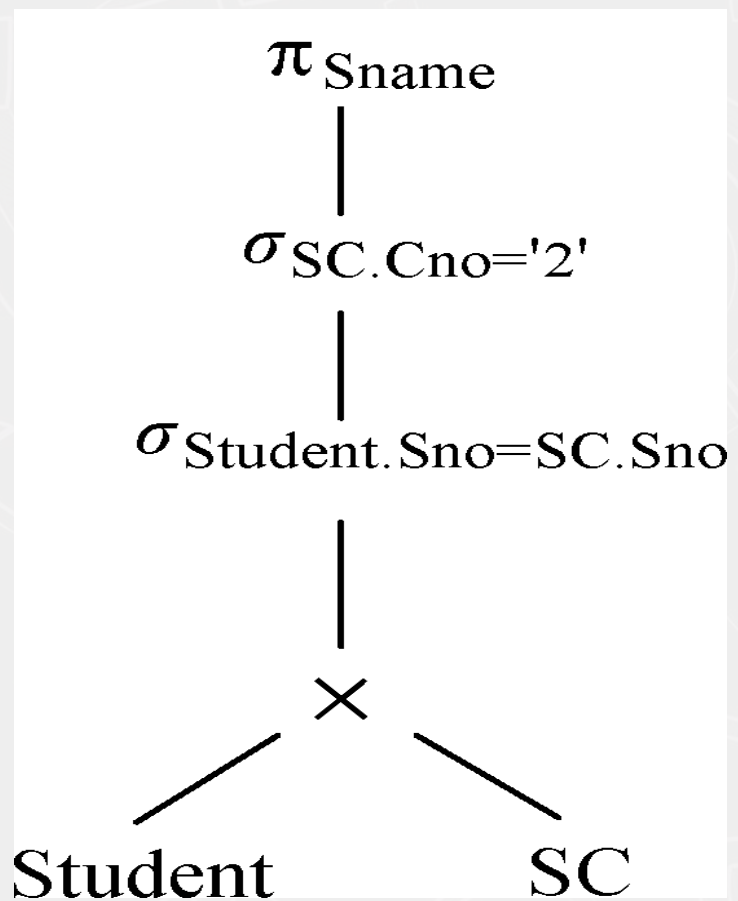


- [例9.4] 下面给出[例9.3]中 SQL语句的代数优化示例

(1) 把SQL语句转换成查询树，如下图所示



为了使用关系代数表达式的优化法，假设内部表示是关系代数语法树，则上面的查询树如图9.4所示。



## (2) 对查询树进行优化

- 利用规则4、6把选择 $\sigma_{SC.Cno='2'}$ 移到叶端，图9.4查询树便转换成下图优化的查询树。这就是9.2.2节中Q3的查询树表示

