

第九章 关系查询处理与查询优化

- 关系数据库系统的查询处理

- 查询处理步骤

- 查询分析

- 词法分析
 - 语法分析
 - 语义分析
 - 符号名转换

- 查询检查

- 查询优化

- 代数优化
 - 物理优化
 - 存取路径和底层操作算法选择
 - rule based、cost based 和 semantic based

- 查询执行

- 查询计划 由code generator 生成

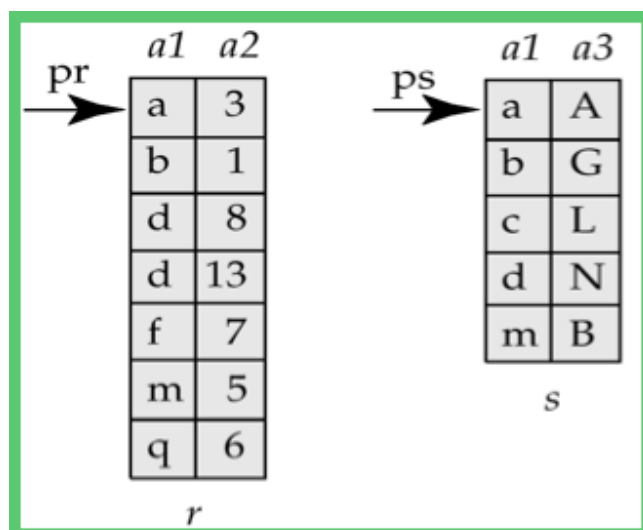
- 实现查询操作的算法示例

- 操作主要考虑I/O块数(相对而言CPU快)

- 选择操作

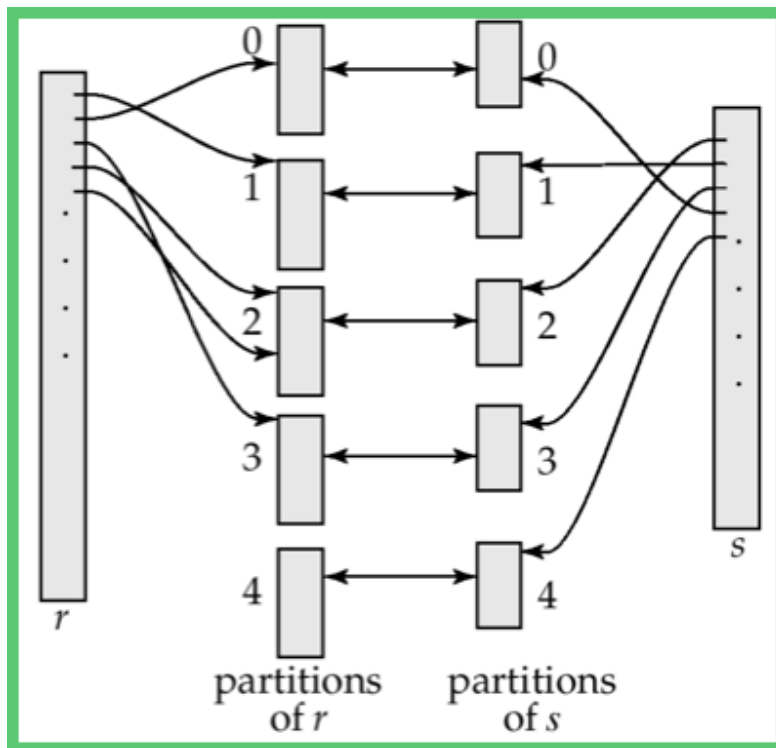
- 连接操作的实现

- 嵌套循环方法
 - 归并连接 (merge join)



- Pr指向r第一个元组,ps指向s第一个元组
 - While (r和s的元组都未处理完)

- 比较pr和ps指向的元组
- 相等，连接对应元组，ps后移
- $(*pr) < (*ps)$, pr后移
- $(*pr) < (*ps)$, ps后移
- 索引连接法(index join)
 - 如果s在连接属性上没有索引，建立索引
 - 对r中每一元组，由s上连接属性的索引查找对应的元组
 - 连接对应元组
- Hash连接
 - 划分阶段



- 只有在相同桶号中的元组才能被连接（大划小）
 - 探查阶段(probing phase)
- 关系数据库系统的查询优化
 - 思想：由系统代替用户优化。
 - 经验
 - 笛卡儿积运算最好能和相应的选择一起构成连接运算；
 - 选择能早做就尽量早做
- 代数优化
 - 代数优化策略
 - 通过对关系代数表达式的等价变化来提高效率
 - 代数表达式的等价
 - 指用相同的关系代数代替两个表达式中相应的关系所得到的结果相同

- 等价变换规则

- 连接、笛卡儿积**交换律**

$$E1 \times E2 = E2 \times E1$$

$$E1 \bowtie E2 = E2 \bowtie E1$$

$$E1 \bowtie_F E2 = E2 \bowtie_F E1$$

- 连接、笛卡儿积的**结合律**

- $(E1 \times E2) \times E3 = E1 \times (E2 \times E3)$

- $(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$

- $(E1 \bowtie_{F1} E2) \bowtie_{F2} E3 = E1 \bowtie_{F1} (E2 \bowtie_{F2} E3)$

- 投影的串接定律

- $\Pi_{A1,A2,\dots,A_n}(\Pi_{A1,A2,\dots,A_n,B1,B2,\dots,B_n}(E))$
 $= \Pi_{A1,A2,\dots,A_n}(E)$

- 选择的串接定律

- $\sigma_{F1}(\sigma_{F2}(E)) = \sigma_{F1 \wedge F2}(E)$

- 常常**反向使用**以**增加选择数目**以便与6合用

- 选择与投影操作的交换律

- $\sigma_F(\Pi_{A1,A2,\dots,A_n}(E)) = \Pi_{A1,A2,\dots,A_n}(\sigma_F(E))$
 - 选择条件F仅涉及A1,A2,...,An

- $\Pi_{A1,A2,\dots,A_n}(\sigma_F(E)) = \Pi_{A1,A2,\dots,A_n}(\sigma_F(\Pi_{B1,B2,\dots,B_m}(E)))$
 - 选择条件F涉及A1,A2,...,An中不存在的B1,B2,...,Bm

- 选择与笛卡儿积的交换律

- $\sigma_F(E_1 \times E_2) = \sigma_F(E_1) \times E_2$
 - F仅涉及E₁的属性
- $\sigma_F(E_1 \times E_2) = \sigma_{F_1}(E_1) \times \sigma_{F_2}(E_2)$
 - F=F₁∧F₂,且F₁只涉及到E₁的属性, 而F₂只涉及到E₂的属性
- $\sigma_F(E_1 \times E_2) = \sigma_{F_2}(\sigma_{F_1}(E_1) \times E_2)$
 - F=F₁∧F₂,且F₁只涉及到E₁的属性, 而F₂只涉及到E₁和E₂的属性

- 选择与并的交换律

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

- 选择与差的交换律

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

- 选择与自然连接的交换律

$$\sigma_F(E_1 \bowtie E_2) = \sigma_F(E_1) \bowtie \sigma_F(E_2)$$

- F只涉及E₁与E₂的公共属性

- 投影与笛卡儿积的分配率

$$\Pi_{A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n}(E_1 \times E_2) = \Pi_{A_1, A_2, \dots, A_n}(E_1) \times \Pi_{B_1, B_2, \dots, B_n}(E_2)$$

- A₁, A₂, ..., A_n是E₁的属性, 而B₁, B₂, ..., B_n是E₂的属性

- 投影与并的分配率

$$\Pi_{A_1, A_2, \dots, A_n}(E_1 \cup E_2) = \Pi_{A_1, A_2, \dots, A_n}(E_1) \cup \Pi_{A_1, A_2, \dots, A_n}(E_2)$$

- 查询树的启发式优化

- 选择尽可能早做
- 让投影运算和选择运算同时进行 (避免重复扫描关系)
- 把投影与其前或后的双目运算结合起来 (避免为了去掉某些字段而扫描一遍关系)
- 把选择与其前面要执行的笛卡儿积结合成连接运算
- 找出公共子表达式
- 关系表达式优化算法

- 五大步：
 - 选择变串连
 - 尽可能先做选择
 - 尽可能先做投影
 - 同时执行多个选择和投影
 - 分组
- 物理优化
 - 概述
 - 代数优化负责优化各操作的顺序
 - 每种操作有多种实现算法
 - 物理优化就是要选择高效合理的操作算法→优化的查询计划
 - 充分考虑索引、数据的存储分布等存取路径。利用它们进一步改善查询效率
 - 方法
 - 基于规则的启发式优化
 - 基于代价估算的优化
 - 两者结合的优化
 - （前者用于产生较优计划；后者负责比较）
- 选择操作的启发式规则
- 基于代价的优化

以上内容整理于 [幕布文档](#)