



海量数据计算研究中心

实现篇

第九章 查询优化

主讲：程思瑶

海量数据计算研究中心





查询优化技术

- 问题的提出
- 关系表达式的等价转换规则
- 表达式结果大小的估计
- 启发式关系代数优化算法
- 基于复杂性估计的查询优化方法





查询优化技术

- 问题的提出
- 关系表达式的等价转换规则
- 表达式结果大小的估计
- 启发式关系代数优化算法
- 基于复杂性估计的查询优化方法





问题的提出

- 设有如下两个关系：
 - $S(S\#, SN, \dots)$
 - $SC(S\#, C\#, \dots)$
- 有如下SQL语句：
 - `SELECT DISTINCT S.SN`
 - `FROM S, SC`
 - `WHERE S.S#=SC.S# AND SC.C#="C2"`
- 存在多种处理策略：
 - $Q_1 = \Pi_{SN} (\sigma_{(S.S\#=SC.S\#) \wedge (SC.C\#="C2")}(S \times SC));$
 - $Q_2 = \Pi_{SN} (\sigma_{SC.C\#="C2"} (S \bowtie_{S.S\#=SC.S\#} SC));$
 - $Q_3 = \Pi_{SN} ((\sigma_{SC.C\#="C2"} SC) \bowtie_{S.S\#=SC.S\#} S)。$





问题的提出

- $Q1 = \Pi_{SN}(\sigma_{(S.S\# = SC.S\#) \wedge (SC.C\# = "C2")}(S \times SC))$
- Q_1 的执行过程:
 - 计算 S 和 SC 的笛卡儿积:
 - ① 使用一个内存缓冲区 BS 读入某个关系 (如 S) 的未处理元组;
 - ② 使用另一个内存缓冲区 BSC 读入另一个关系 (如 SC) 的未处理元组;
 - ③ 把 BS 中的每个元组和 BSC 中每个元组相连接, 并送入输出缓冲区 BO ;
 - ④ 如果 BO 已满则写到一个临时文件;
 - ⑤ 重复步骤②~④, 直至 SC 中元组全部处理完;
 - ⑥ 重复步骤①~⑥, 直至 S 中元组全部处理完。
 - 从临时文件读入 S 和 SC 的笛卡儿积, 按照选择条件选取满足要求的记录。
 - 把第2步的结果在 SN 上作投影输出, 得到最终结果。





- S中有**1000**个学生记录
- SC中有**10000**个选课记录
- SC中选修C2课程的记录为**50**个

问题的提出

- Q_1 的时间开销:

- 每个磁盘块能装**10**个S的元组或**100**个SC的元组。每个磁盘块能装**5**个结果元组。每秒可读/写**20**个磁盘块。
- (1) 计算S和SC的笛卡尔积:
 - 时间开销: **10秒** (读) + **100000秒** (写)
- (2) 选择:
 - 时间开销: **100000秒** (读)
- (3) 投影:
 - 时间开销: **0秒**
- 忽略CPU时间, Q_1 需要的处理时间大约**200010秒**。





问题的提出

- $Q_2 = \Pi_{SN} (\sigma_{SC.C\#="C2"} (S \bowtie_{S.S\#=SC.S\#} SC))$
- Q_2 的执行过程:
 - (1) 计算自然连接。
 - (2) 读取中间文件块，执行选择运算。
 - (3) 把第2步结果投影输出。





- S中有**1000**个学生记录
- SC中有**10000**个选课记录
- SC中选修C2课程的记录为**50**个

问题的提出

- Q_2 的时间开销：
 - 每个磁盘块能装**10**个S的元组或**100**个SC的元组。每个磁盘块能装**5**个结果元组。每秒可读/写**20**个磁盘块。
 - (1) 计算自然连接：
 - 设自然连接的结果为**1000**个元组。
 - 时间开销：**10秒**（读）+**10秒**（写）
 - (2) 选择：
 - 时间开销：**10秒**（读）
 - (3) 投影：
 - 时间开销：**0秒**
 - 忽略CPU时间， Q_2 需要的处理时间大约**30**秒。





问题的提出

- $Q_3 = \Pi_{SN} ((\sigma_{SC.C\#="C2"} SC) \bowtie_{S.S\#=SC.S\#} S)$
- Q_3 的执行过程:
 - (1) 对SC作选择运算。
 - (2) 读取S表, 把读入的S元组和内存中的SC元组作连接。
 - (3) 把第2步结果投影输出。





- S中有**1000**个学生记录
- SC中有**10000**个选课记录
- SC中选修C2课程的记录为**50**个

问题的提出

- Q_3 的时间开销:

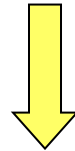
- 每个磁盘块能装**10**个S的元组或**100**个SC的元组。每个磁盘块能装**5**个结果元组。每秒可读/写**20**个磁盘块。
- (1) 对SC作选择运算:
 - 时间开销: **5秒** (读)
- (2) 连接:
 - 时间开销: **5秒** (读)
- (3) 投影:
 - 时间开销: **0秒**
- 忽略CPU时间, Q_3 需要的处理时间大约**10秒**。
- 若SC表的C#字段有索引, Q_3 的处理时间将进一步减少





问题的提出

- 使用不同的策略处理一个查询会得到不同的时间开销。



- 需要选择优化的查询处理策略，以减少查询处理时间，提高系统的处理能力。
- 这个简单的例子不仅充分说明了查询优化的必要性，也给了我们一个查询优化的方法：当一个查询中具有选择和连接时，应当先执行选择后执行连接，尽量减少中间结果的大小，加快连接操作的处理。





目录

- 问题的提出
- 关系表达式的等价转换规则
- 表达式结果大小的估计
- 启发式关系代数优化算法
- 基于复杂性估计的查询优化方法





关系表达式的等价转换规则

- 在很多数据库管理系统中，查询处理的第一步是把查询变换为与关系代数对应的内部表示，如查询树。
- 一个查询可以变换为一个等价的关系代数表达式。
- 在关系数据库管理系统中，**关系代数表达式的等价转换规则非常重要。**





关系表达式的等价转换规则

- 关系代数等价转换规则
 - 设 E_1 和 E_2 是两个关系代数表达式。如果 E_1 和 E_2 表示相同的系，则称 E_1 和 E_2 等价，记作 $E_1 \equiv E_2$ 。





关系表达式的等价转换规则

- 关系代数等价转换规则 (12类)

- 1. 选择串接律

- $\sigma_{c1 \text{ AND } \dots \text{ AND } c_n}(E) \equiv \sigma_{c1}(\sigma_{c2}(\dots(\sigma_{c_n}(E))\dots))$

- 2. 选择交换律

- $\sigma_{c1}(\sigma_{c2}(E)) \equiv \sigma_{c2}(\sigma_{c1}(E))$

- 3. 投影串接律

- $\Pi_{L1}(\Pi_{L2}(\dots(\Pi_{Ln}(E))\dots)) \equiv \Pi_{L1}(E),$

- 其中E是关系代数表达式, L_i 是投影属性集合, 而且 $L_1 \subseteq L_2 \subseteq \dots \subseteq L_n$ 。





关系表达式的等价转换规则

• 关系代数等价转换规则 (12类)

– 4. 选择投影交换律

- $\Pi_L (\sigma_C (E)) \equiv \sigma_C (\Pi_L (E))$

- 其中，E是关系代数表达式，L是投影属性集合，C是选择条件，C只涉及L中的属性。

- $\Pi_L (\sigma_C (E)) \equiv \Pi_L (\sigma_C (\Pi_{L, B_1, \dots, B_m} (E)))$ 。

- C还涉及L以外的属性 B_1 、...、 B_m

– 5. 连接和笛卡儿乘积的交换律

- $E_1 \times E_2 \equiv E_2 \times E_1, E_1 \bowtie_C E_2 \equiv E_2 \bowtie_C E_1$

- 其中， E_1 和 E_2 是关系代数表达式，C是连接条件。

– 6. 集合操作的交换律

- $E_1 \cup E_2 \equiv E_2 \cup E_1, E_1 \cap E_2 \equiv E_2 \cap E_1$

- 其中， E_1 和 E_2 是关系代数表达式。





关系表达式的等价转换规则

• 关系代数等价转换规则 (12类)

– 7. 连接、笛卡儿乘积和集合操作的结合律

- (1) $(E_1 \times E_2) \times E_3 \equiv E_1 \times (E_2 \times E_3),$
- (2) $(E_1 \bowtie_{C_1} E_2) \bowtie_{C_2} E_3 \equiv E_1 \bowtie_{C_1} (E_2 \bowtie_{C_2} E_3),$
- (3) $(E_1 \cup E_2) \cup E_3 \equiv E_1 \cup (E_2 \cup E_3),$
- (4) $(E_1 \cap E_2) \cap E_3 \equiv E_1 \cap (E_2 \cap E_3),$

– 8. 选择、连接和笛卡儿乘积的分配律

- (1) $\sigma_{C_1} (E_1 \bowtie_{C_2} E_2) \equiv (\sigma_{C_1} (E_1)) \bowtie_{C_2} E_2$
 - 其中, C_1 是选择条件, C_2 是连接条件, E_1 和 E_2 是关系代数表达式, C_1 仅涉及 E_1 的属性。
- (2) $\sigma_{C_1} (E_1 \bowtie_{C_2} E_2) \equiv (\sigma_{C_{11}} (E_1)) \bowtie_{C_2} (\sigma_{C_{12}} (E_2))$
 - 其中, C_1 是选择条件, C_2 是连接条件, E_1 和 E_2 是关系代数表达式, $C_1 = C_{11} \wedge C_{12}$, C_{11} 仅涉及 E_1 的属性, C_{12} 仅涉及 E_2 的属性。
- (3) 用 \times 代替上边两个等价式中的 \bowtie_{C_2} , 就可以得到选择与笛卡尔乘积的分配律。





关系表达式的等价转换规则

- 关系代数等价转换规则 (12类)

- 9. 投影、连接和笛卡尔乘积的分配律

- (1) $\Pi_L (E_1 \bowtie_C E_2) \equiv (\Pi_{L_1} (E_1)) \bowtie_C (\Pi_{L_2} (E_2))$

- 其中，C是连接条件， E_1 和 E_2 是关系代数表达式， $L=L_1 \cup L_2$ 是投影属性集合， L_1 仅涉及 E_1 的属性， L_2 仅涉及 E_2 的属性，C仅涉及L的属性。

- (2) 用 \times 代替上边两个等价式中的 \bowtie_C ，就可以得到选择与笛卡尔乘积的分配律。





关系表达式的等价转换规则

- 关系代数等价转换规则 (12类)

- 10. 选择与集合操作的分配律

- (1) $\sigma_C (E_1 \cup E_2) \equiv (\sigma_C (E_1)) \cup (\sigma_C (E_2))$,
 - (2) $\sigma_C (E_1 \cap E_2) \equiv (\sigma_C (E_1)) \cap (\sigma_C (E_2))$,
 - (3) $\sigma_C (E_1 - E_2) \equiv (\sigma_C (E_1)) - (\sigma_C (E_2))$,

- 11. 投影与集合操作的分配律

- (1) $\Pi_L (E_1 \cup E_2) \equiv (\Pi_L (E_1)) \cup (\Pi_L (E_2))$,
 - (2) $\Pi_L (E_1 \cap E_2) \equiv (\Pi_L (E_1)) \cap (\Pi_L (E_2))$,
 - (3) $\Pi_L (E_1 - E_2) \equiv (\Pi_L (E_1)) - (\Pi_L (E_2))$,

- 12. 其他等价变换律

- 除了上述规律以外，还有很多其他等价变换规律。例如，可以把选择或连接条件等价地转换为其他形式的条件。





目录

- 问题的提出
- 关系表达式的等价转换规则
- **表达式结果大小的估计**
- 启发式关系代数优化算法
- 基于复杂性估计的查询优化方法





表达式结果大小的估计

- 一个操作的代价依赖于它的输入的大小和其他统计信息

— 数据库系统的统计信息

- n_r : 关系 r 的元组数
- b_r : 包含关系 r 中元组的磁盘块数
- l_r : 关系 r 中每个元组的字节数
- f_r : 关系 r 的块因子——一个磁盘块能容纳 r 中元组的个数
- $V(A, r)$: 关系 r 中属性 A 中出现的非重复值的个数。

— 如果假设关系 r 在物理上存储于一个文件中，则

$$b_r = \left\lceil \frac{n_r}{f_r} \right\rceil$$





表达式结果大小的估计

- 当 r 中 A 属性上的取值分布是均匀的，运算结果大小的估计如下：

- 投影 $\Pi_A(r)$

- 估计值为 $V(A, r)$

- 选择 $\sigma_{A=a}(r)$

- 估计值为 $n_r/V(A, r)$

- 选择 $\sigma_{A \leq v}(r)$

- 如果 $v < \min(A, r)$ ，则估计值为0
 - 如果 $v \geq \max(A, r)$ ，则估计值为 n_r
 - 否则，估计值为

$$\frac{v - \min(A, r)}{\max(A, r) - \min(A, r)} \times n_r$$





表达式结果大小的估计

- 当r中A属性上的取值分布是**均匀的**，运算结果大小的估计如下：

— **合取** $\sigma_{\theta_1 \wedge \theta_2 \wedge \dots \wedge \theta_k}(r)$

- 对于每个 θ_i ，我们按照之前描述的那样估计选择的大小，记为 s_i ，则

$$n_r \times \frac{s_1 \times s_2 \times \dots \times s_k}{n_r^k}$$

— **析取** $\sigma_{\theta_1 \vee \theta_2 \vee \dots \vee \theta_k}(r)$

- 对于每个 θ_i ，我们按照之前描述的那样估计选择的大小，记为 s_i ，则

$$\left(1 - \left(1 - \frac{s_1}{n_r}\right) \times \left(1 - \frac{s_2}{n_r}\right) \times \dots \times \left(1 - \frac{s_k}{n_r}\right)\right) \times n_r$$





表达式结果大小的估计

- 当 r 中 A 属性上的取值分布是均匀的，运算结果大小的估计如下：

— 取反 $\sigma_{\neg\theta}(r)$

- 在无空值的情况下，估计值为 $n_r - s_\theta$
- 在有空值的情况下，估计值为 $n_r - n_{null} - s_\theta$

— 笛卡尔积 $r \times s$

- 元组个数 $n_r \times n_s$ ，每个元组占 $l_r + l_s$ 个字节





表达式结果大小的估计

- 当 r 中 A 属性上的取值分布是均匀的，运算结果大小的估计如下：

— 自然连接 $r(R)$ 和 $s(S)$

- 若 $R \cap S$ 为空，则类似于笛卡尔积的结果
- 若 $R \cap S$ 为 R 的码，则可知 s 的一个元组至多与 r 的一个元组相连接，因此自然连接结果的元组数小于等于 n_s ，若 $R \cap S$ 构成 S 中参照 R 的外码，则自然连接结果等于 n_s ，反之亦然
- 若 $R \cap S$ 既不是 R 的码也不是 S 的码，则可按如下估计

$$\min \left(n_r \times \frac{n_s}{V(A, s)}, n_s \times \frac{n_r}{V(A, r)} \right)$$





表达式结果大小的估计

- 当 r 中 A 属性上的取值分布是**均匀的**，运算结果大小的估计如下：

- **聚集：**

- $V(A, r)$

- **集合运算：**

- 类似于合取、析取、取反的估计方法

- **外连接(结果上界)：**

- r 左外连接 s ： r 与 s 自然连接的大小估计值 $+n_r$
- r 右外连接 s ： r 与 s 自然连接的大小估计值 $+n_s$
- R 全外连接 s ： r 与 s 自然连接的大小估计值 $+n_r+n_s$

- 当 r 中 A 属性上的取值分布**不均匀的**，可以采用**直方图等**统计方法进行结果大小估计





目录

- 问题的提出
- 关系表达式的等价转换规则
- 表达式结果大小的估计
- **启发式关系代数优化算法**
- 基于复杂性估计的查询优化方法





启发式关系代数优化算法

- 启发式代数优化规则

- 给定一个关系代数表达式，可以应用一组启发式规则对其进行等价变换，产生一个具有较高效率的等价表达式。
- 需要注意，这些规则不能保证一定产生最优等价表达式。





启发式关系代数优化算法

- 启发式代数优化规则

- 规则1 选择和投影操作尽早执行。

- 4. 选择投影交换律

- $\Pi_L (\sigma_C (E)) \equiv \sigma_C (\Pi_L (E))$

- $\Pi_L (\sigma_C (E)) \equiv \Pi_L (\sigma_C (\Pi_{L, B1, \dots, Bm} (E)))$

- 8. 选择、连接和笛卡儿乘积的分配律

- $\sigma_{C1} (E_1 \bowtie_{C2} E_2) \equiv (\sigma_{C1} (E_1)) \bowtie_{C2} E_2$

- $\sigma_{C1} (E_1 \bowtie_{C2} E_2) \equiv (\sigma_{C11} (E_1)) \bowtie_{C2} (\sigma_{C12} (E_2))$

- 10. 选择与集合操作的分配律

- $\sigma_C (E_1 \cup E_2) \equiv (\sigma_C (E_1)) \cup (\sigma_C (E_2)),$

- $\sigma_C (E_1 \cap E_2) \equiv (\sigma_C (E_1)) \cap (\sigma_C (E_2)),$

- $\sigma_C (E_1 - E_2) \equiv (\sigma_C (E_1)) - (\sigma_C (E_2)),$





启发式关系代数优化算法

- 启发式代数优化规则

- 规则2 把某些选择操作与笛卡尔乘积相结合，形成一个连接操作。
- 规则3 同时执行相同关系上的多个选择和投影操作。
- 规则4 把投影操作与连接操作结合起来执行。





启发式关系代数优化算法

- 启发式代数优化规则

- 规则5 提取公共表达式。

- 如果一个反复出现的公共表达式的结果不是一个很大的关系，而且从外存读入它的时间小于计算它的时间，可以只计算这个表达式一次并存储其结果，以后遇到该表达式时不必重新计算。





启发式关系代数优化算法

- 启发式代数优化算法

- 构造查询的内部表示是查询处理的第一步。给定一个用高级语言定义的查询，需要两步来构造该查询的内部表示。

- 第一步把用高级语言定义的查询转换为关系代数表达式。
 - 第二步把关系代数表达式转换为查询树。





启发式关系代数优化算法

- 给定一个用SQL语言定义的查询:
 - SELECT list
 - FROM R_1, R_2, \dots, R_n
 - WHERE C
- 可以按照如下方法把这个查询转换为关系代数表达式:
 - 使用FROM从句中的关系 R_1, R_2, \dots, R_n 构造笛卡儿积:
$$R_1 \times R_2 \times \dots \times R_n;$$
 - 在第1步的基础上构造一个选择操作:
$$\sigma_C (R_1 \times R_2 \times \dots \times R_n);$$
 - 在第2步的基础上构造一个投影操作:
$$\Pi_{list} (\sigma_C (R_1 \times R_2 \times \dots \times R_n)).$$





启发式关系代数优化算法

• 从关系代数表达式到查询树的转换：

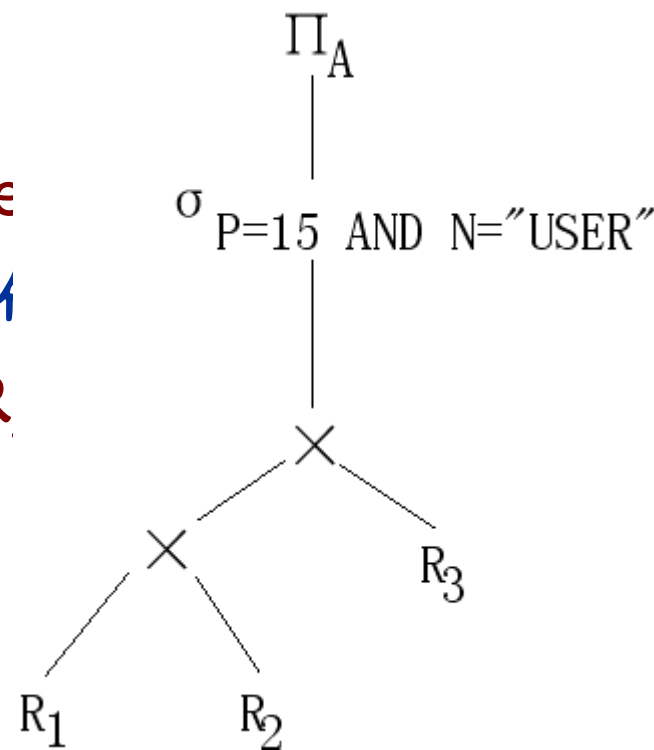
— 给定查询

- SELECT A
- FROM R_1, R_2, R_3
- WHERE $P=15$ AND $N=$ “User”

— 使用上述方法可以得到关系1

- $\Pi_A (\sigma_{P=15 \text{ AND } N= \text{“User”}} ((R_1 \times R_2$

— 对应的查询树：





启发式关系代数优化算法

- 启发式代数优化算法：

- 输入：关系代数表达式。

- 输出：计算输入关系代数表达式的程序。

- 方法：

- (1) 使用规律1把每个选择操作 $\sigma_{C_1 \text{ AND } \dots \text{ AND } C_n} (E)$ 变换为：
 $\sigma_{C_1} (\dots (\sigma_{C_n} (E)) \dots)$;
 - (2) 使用规律2,4,8和10，把查询树上的每个选择操作移到尽可能靠近叶节点；
 - (3) 使用规律3、4、9和11，把查询树上的每个投影操作移到尽可能靠近叶节点
 - (4) 使用规律1、3和4把串接的多个选择或多个投影操作组合为单个的选择或投影操作；
 - (5) 使用规律7重新安排叶节点，使得具有最小选择操作的叶节点最先执行；



启发式关系代数优化算法

- 启发式代数优化算法：

- 输入：关系代数表达式。

- 输出：计算输入关系代数表达式的程序。

- 方法(续)：

- (6) 组合笛卡尔乘积和相继的选择操作形成连接操作；
 - (7) 把最后的查询树划分为多个子树，使得每个子树上的操作可以由单个存取程序一次完成。划分方法如下：
 - ① 每个二目操作在且仅在一个子树中；
 - ② 如果二目操作 α 在子树 T 中，而且从叶到根的方向的路径 $\beta \rightarrow \alpha$ 是仅包含一目操作的最长路径，则 $\beta \rightarrow \alpha$ 也包含在 T 中。
 - (8) 产生一个计算最后查询树的程序，每一步计算一个子树



启发式关系代数优化算法

- 例：

- 图书馆数据库：

- 书目关系: Boo(Ti, Au, Pn, Nc)
- 出版社关系: Pub(Pn, Pa, Pc)
- 借阅者关系: Bor(Na, Ad, Ci, Cn)
- 借阅登记关系: Loa(Cn, Nc, Da)。

- 设由已借出书的信息构成的视图LBI定义如下：

- CREATE VIEW LBI
- AS SELECT Ti,Au,Boo.Pn,Boo.Nc,Na,Ad,Ci,Bor.Cn,Da
- FROM Boo,Bor,Loa
- WHERE Boo.Nc=Loa.Nc AND Bor.Cn=Loa.Cn

- 有如下查询：

- SELECT Ti
- FROM LBI
- WHERE Da<2/1/1994。





启发式关系代数优化算法

–书目关系: Boo(Ti, Au, Pn, Nc)
–出版社关系: Pub(Pn, Pa, Pc)

–CREATE VIEW LBI
–AS SELECT
Ti,Au,Boo.Pn,Boo.Nc,Na,Ad,Ci,Bor.Cn,Da
–FROM Boo,Bor,Loa
–WHERE Boo.Nc=Loa.Nc AND Bor.Cn=Loa.Cn

–SELECT Ti
–FROM LBI
–WHERE Da<2/1/1994

– 查询处理程序首先把这个查询与视图LBI合并形成一个直接定义在基关系上的等价查询并变换为等价的关系代数表达式:

• $\Pi_{Ti} (\sigma_{C_1} (\Pi_L (\sigma_{C_2} ((Loa \times Bor) \times Boo))))$

• 其中:

– $C_1 = \text{“Da} < 2/1/1994\text{”}$,

– $L = \text{“Ti, Au, Boo.Pn, Boo.Nc, Na, Ad, Ci, Bor.Cn, Da”}$,

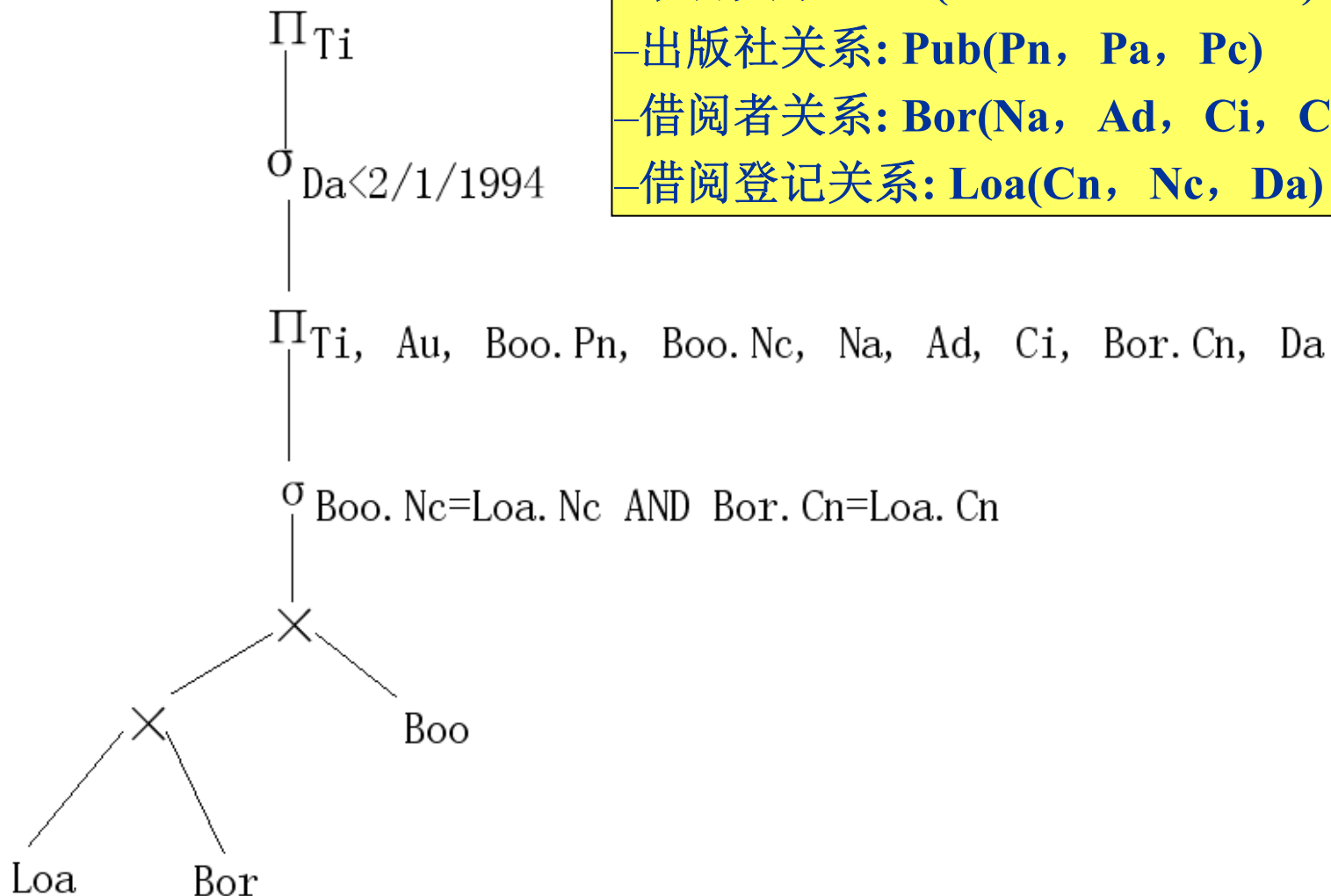
– $C_2 = \text{“Boo.Nc=Loa.Nc AND Bor.Cn=Loa.Cn”}$ 。





启发式关系代数优化算法

- 对应的查询树



—书目关系: **Boo**(Ti, Au, Pn, Nc)
—出版社关系: **Pub**(Pn, Pa, Pc)
—借阅者关系: **Bor**(Na, Ad, Ci, Cn)
—借阅登记关系: **Loa**(Cn, Nc, Da)

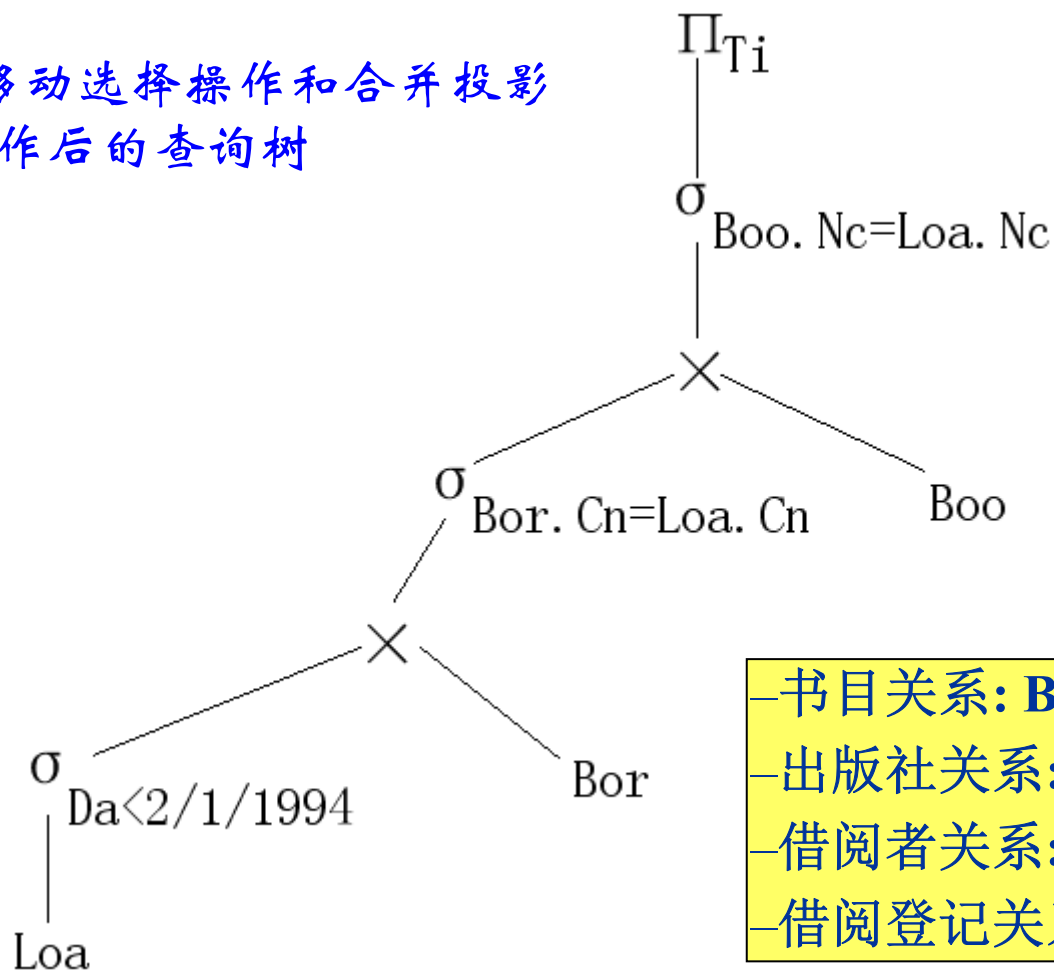




启发式关系代数优化算法

• Q的优化过程

—移动选择操作和合并投影
操作后的查询树



—书目关系: **Boo**(Ti, Au, Pn, Nc)
—出版社关系: **Pub**(Pn, Pa, Pc)
—借阅者关系: **Bor**(Na, Ad, Ci, Cn)
—借阅登记关系: **Loa**(Cn, Nc, Da)

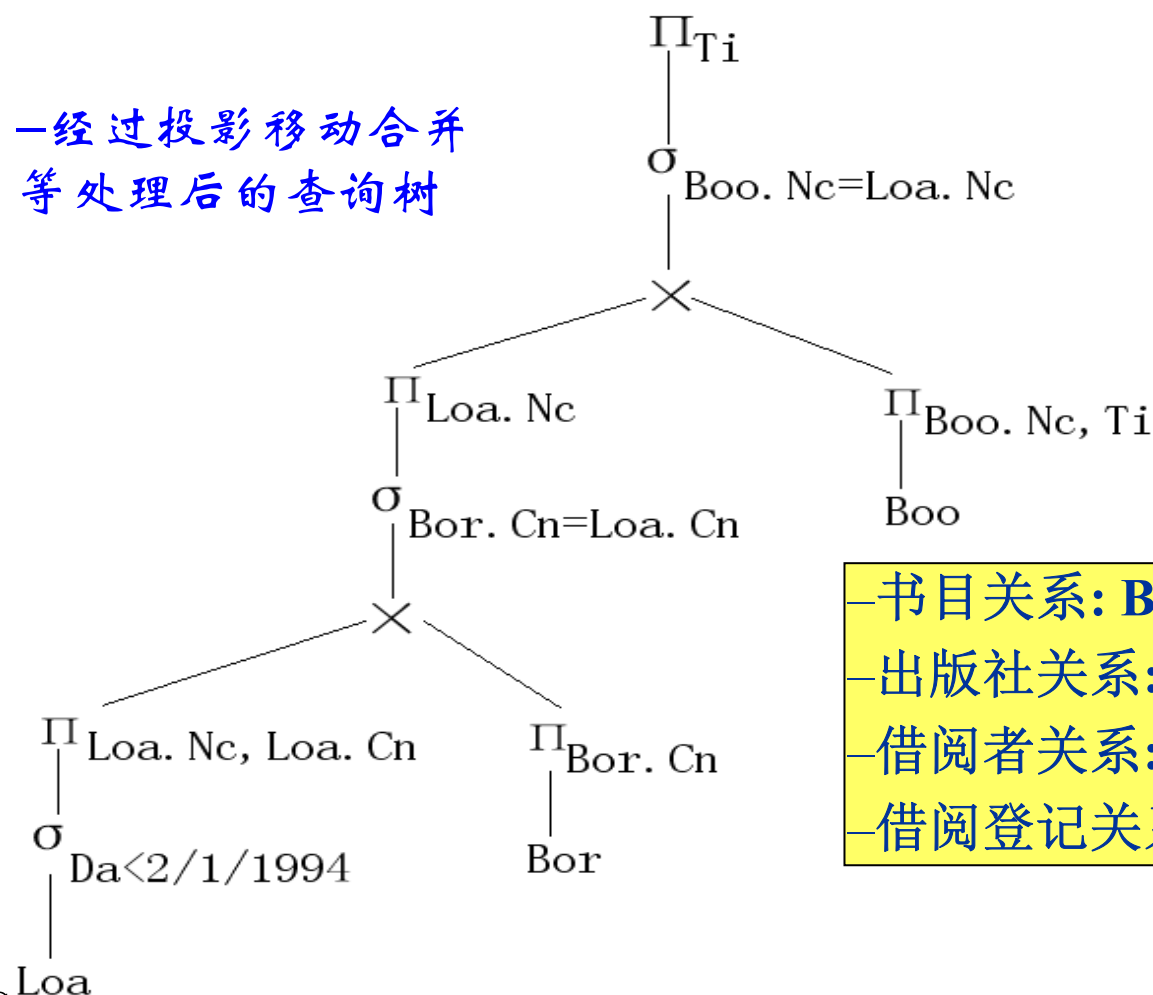




启发式关系代数优化算法

• Q的优化过程

— 经过投影移动合并
等处理后的查询树



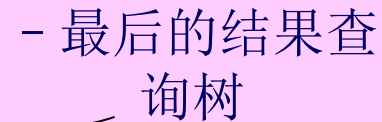
— 书目关系: **Boo(Ti, Au, Pn, Nc)**
— 出版社关系: **Pub(Pn, Pa, Pc)**
— 借阅者关系: **Bor(Na, Ad, Ci, Cn)**
— 借阅登记关系: **Loa(Cn, Nc, Da)**





- 合并选择和笛卡儿积形成连接操作

—借阅登记关系: $\text{Loa}(\text{Cn}, \text{Nc}, \text{Da})$





目录

- 问题的提出
- 关系表达式的等价转换规则
- 表达式结果大小的估计
- 启发式关系代数优化算法
- 基于复杂性估计的查询优化算法



基于复杂性估计的查询优化方法

- 考虑各种查询执行策略的复杂性估计。
- 基于复杂性估计的查询优化方法分为两个阶段。
 - 第一阶段使用启发式优化方法产生逻辑上优化的关系代数表达式或查询计划 P 。
 - 第二阶段为 P 中每个关系代数操作选择具有最小复杂性的实现算法，确定 P 的优化执行策略 $P(A)$ 。
- 影响查询执行效率的主要因素是**磁盘存取块数**。



- 本章重点
 - 熟练掌握关系代数表达式等价变换规则；
 - 掌握启发式查询优化方法。





**Now let's go to
Next Chapter**

