

数据库 第九章

陈鑫蕾 22920202202877

1. 试述查询优化在关系数据库系统中的重要性和可能性。

答：(1) 重要性：关系系统的查询优化既是关系数据库管理系统实现的关键技术，又是关系系统的优点所在。它减轻了用户选择存取路径的负担；用户只需要提出“干什么”，不必考虑如何最好地表达查询以获得较好的效率，而且系统可以比用户程序的“优化”做得更好；

(2) 可能性：①优化器可以从数据字典中获取许多统计信息，例如关系中的元组数、关系中每个属性的分布情况、这些属性上是否有索引等；优化器可以根据这些信息选择有效的执行计划，而用户程序则难以获得这些信息；

②如果数据库的物理统计信息改变了，系统可以自动对查询进行重新优化以选择相适应的执行计划；在非关系系统中必须重写程序，而重写程序在实际应用中往往是不太可能的；

③优化器可以考虑数十甚至数百种不同的执行计划，从中选出较优的一个，而程序员一般只能考虑有限的几种可能性；

④优化器中包括了很多复杂的技术，这些优化技术往往只有最好的程序员才能掌握；系统的自动优化相当于使得所有人都拥有这些优化技术。

2. 假设关系 R (A, B) 和 S (B, C, D) 情况如下：R 有 20000 个元组，S 有 1200 个元组，一块能装 40 个 R 的元组，30 个 S 的元组，估算下列操作需要多少次磁盘块的读写。

(1) R 上没有索引，`select * from R;`

(2) R 中 A 为主码，A 有 B+ 树索引，`select * from R where A = 10;`

(3) 嵌套循环连接 $R \bowtie S$;

(4) 排序合并连接 $R \bowtie S$ ，区分 R 与 S 在 B 属性上已经有序和无序两种情况。

答：(1) $20000 / 40 = 500$ 次；

(2) 4 (三次索引，一次数据)；

(3) 设在内存中存放 5 块 S 元组和 1 块 R 元组，

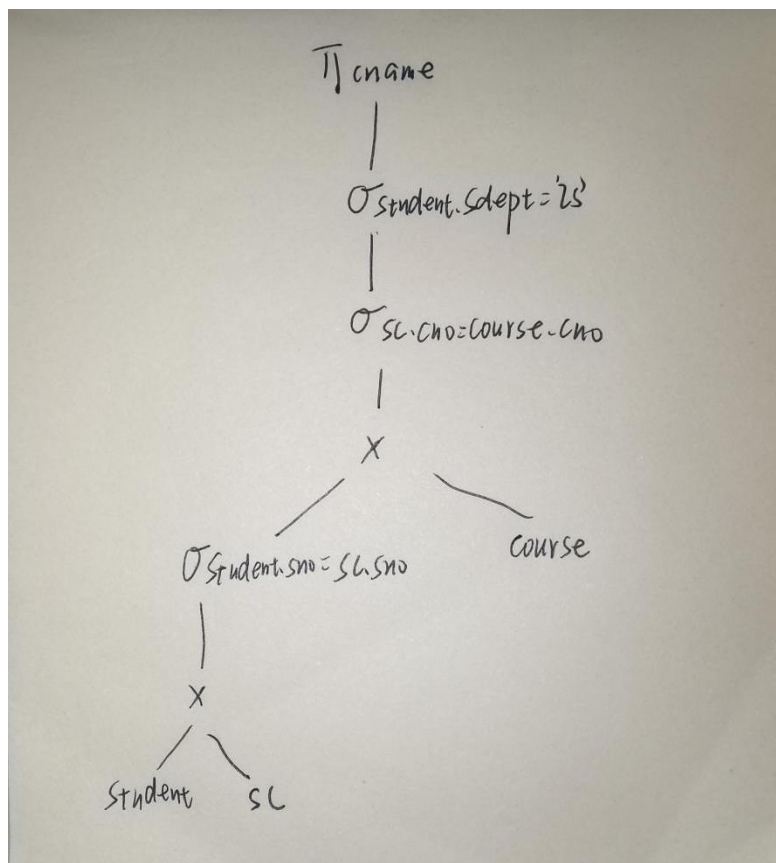
$1200 / 30 + 20000 / 40 * (1200 / 30 / 5) = 4040$;

(4) 有序: $20000 / 40 + 1200 / 30 + \text{新元组的块数}$;

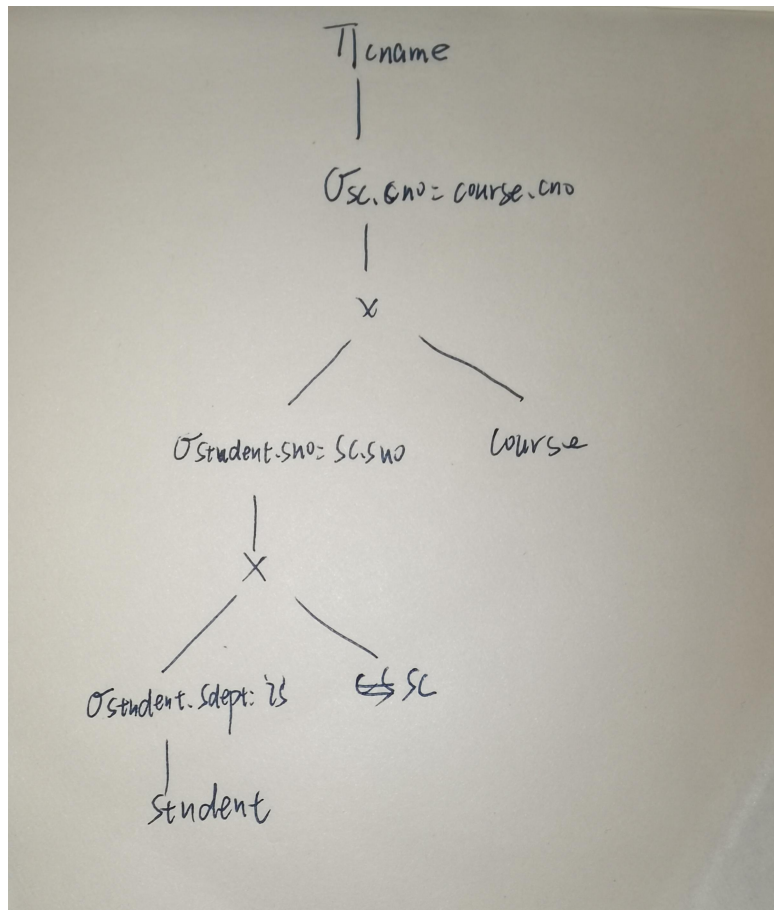
无序: 排序代价 + 有序的块数;

3. 对学生-课程数据库, 查询信息系学生选修了的所有课程名称, 画出语法树, 并优化。

答: 优化前:



优化后:



4. 说明下列查询语句的一种较优的处理方法。

- (1) select * from teacher where Tsex= '女';
- (2) select * from department where Dno<301;
- (3) select * from work where Year<>2000;
- (4) select * from work where Year>2000 and Salary<5000;
- (5) select * from work where Year<2000 or Salary<5000。

答：(1) 全盘顺序扫描；

(2) 估算比例再选择 B+数索引或是全盘扫描；

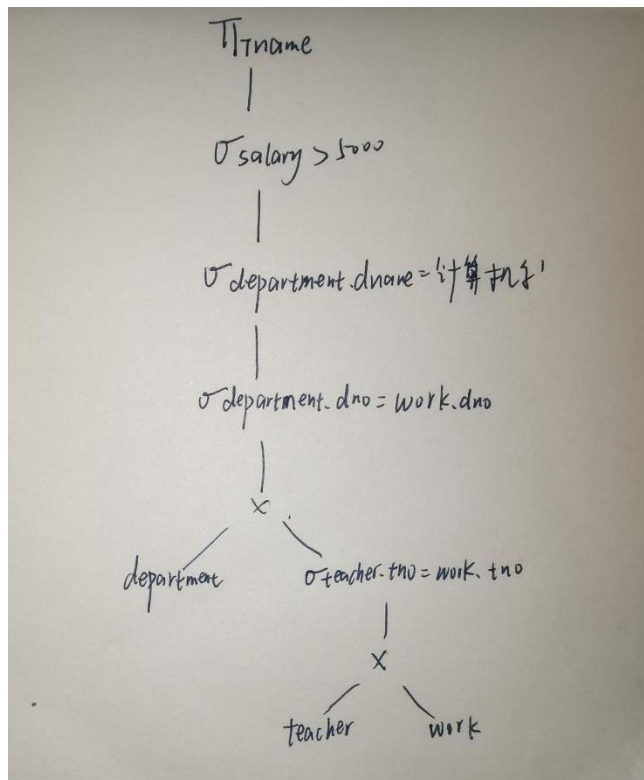
(3) 全盘顺序扫描；

(4) B+数索引；

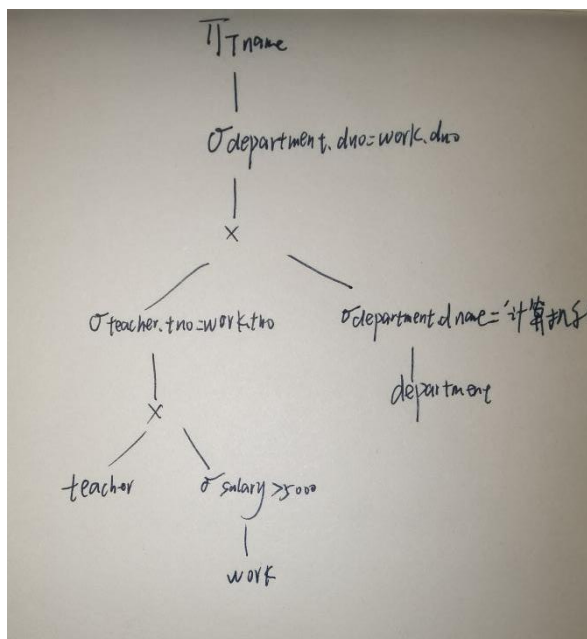
(5) 全盘顺序扫描。

5. 画出以下查询的语法树和优化后的语法树。

答：优化前：



优化后:



6. 试述关系数据库管理系统查询优化的一般准则。

答: (1) 选择运算应尽可能先做;

(2) 投影运算和选择运算同时进行;

(3) 投影同其前后的双目运算结合起来;

- (4) 某些选择同其在它前面要执行的笛卡尔积结合起来成为一个连接运算；
- (5) 找出公共子表达式；
- (6) 选取合适的连接算法。

7. 试述关系数据库管理系统查询优化的一般步骤。

答：(1) 把查询转换成某种内部表示，比如语法树；

(2) 把原始的语法树优化；

(3) 选择底层的存取路径；

(4) 生成查询计划，选择代价最小的。