

第九章 关系查询处理和查询优化

1. 答:

重要性: 关系系统的查询优化既是 RDBMS 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。用户只要提出“干什么”, 不必指出“怎么干”。

查询优化的优点不仅在于用户不必考虑如何最好地表达查询以获得较好的效率, 而且在于系统可以比用户程序的“优化”做得更好。

可能性:

这是因为:

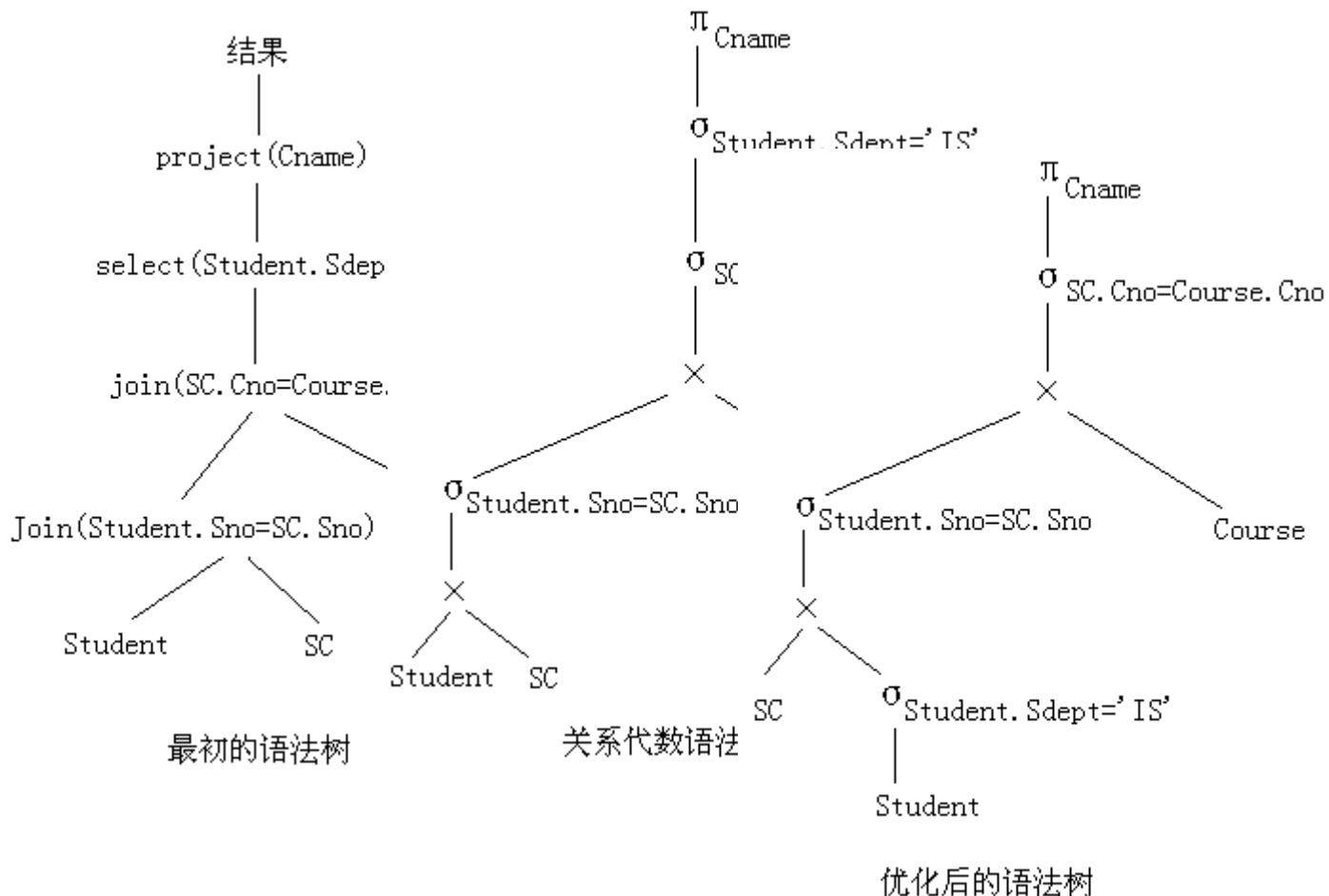
(1) 优化器可以从数据字典中获取许多统计信息, 例如各个关系中的元组数、关系中每个属性值的分布情况、这些属性上是否有索引、是什么索引, B+树索引还是 HASH 索引? 唯一索引? 组合索引? 等等。优化器可以根据这些信息选择有效的执行计划, 而用户程序则难以获得这些信息。

(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优化技术, 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. 答:



3. 答:

下面的优化策略一般能提高查询效率:

- (1) 选择运算应尽可能先做。
- (2) 把投影运算和选择运算同时进行。
- (3) 把投影同其前或其后的双目运算结合起来执行。
- (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。
- (5) 找出公共子表达式。
- (6) 选取合适的连接算法。

解析:

(1) - (5) 是指代数优化策略。(6) 涉及物理优化了。

(1) 选择运算应尽可能先做。因为满足选择条件的元组一般是原来关系的子集, 从而使计算的中间结果变小。这是最基本的也是很有效的优化策略。

(2) 把投影运算和选择运算同时进行。如果在同一个关系上有若干投影和选择运算, 则可以把投影运算和选择运算结合起来, 即选出符合条件的元组后就对这些元组做投影。

(3) 把投影同其前或其后的双目运算结合起来。双目运算有 JOIN 运算、笛卡尔积, 与上面的理由类似, 在进行 JOIN 运算、笛卡尔积时要选出关系的元组, 没有必要为了投影操作 (通常是去掉某些字段) 而单独扫描一遍关系。

(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。连接特别是等连接运算要比在同样关系上的笛卡尔积产生的结果小得多。执行代价也小得多。

(5) 找出公共子表达式。先计算一次公共子表达式并把结果保存起来共享。以避免重复计算公共子表达式。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。我们可以把视图计算出来, 称为视图的实体化。计算结果称为实体化视图。

(6) 选取合适的连接算法。连接操作是关系操作中最费时的操作, 人们研究了许多连接优化算法。例如索引连接算法、排序合并算法、HASH 连接算法等。

选取合适的连接算法属于选择“存取路径”, 是物理优化的范畴。

许多 RDBMS 提供了多种连接算法供优化子系统选择。

有时需要在执行这些连接算法前对关系进行预处理。如对于索引连接算法, 有时要在连接属性上建立索引, 对于排序合并算法、要对连接的两个关系首先进行排序, 然后执行连接。这就是在执行连接前对关系的预处理。

4. 答:

各个关系系统的优化方法不尽相同, 大致的步骤可以归纳如下:

- (1) 把查询转换成某种内部表示, 通常用的内部表示是语法树。
- (2) 把语法树转换成标准(优化)形式。
即利用优化算法, 把原始的语法树转换成优化的形式。
- (3) 选择低层的存取路径。
- (4) 生成查询计划, 选择代价最小的。

解析

读者要把 SQL 查询处理和查询优化的概念结合起来学习, 了解 SQL 查询处理工作的步骤。

如《概论》书图 9.1 所示, 了解查询优化在查询处理中的核心地位。