

第四章 关系系统及其查询优化

第四章进一步讲解关系数据库的基本概念。简单介绍关系数据库管理系统查询优化的重要概念和实现技术。

查询优化是 RDBMS 的内部实现技术,对于一般用户应该是透明的。用户不必了解 RDBMS 是如何对他给出的查询语句进行优化的。

由于 RDBMS 的优化技术并不都是做得很好,对于不同的 SQL 语句、不同的数据库状况优化效果也不一样,有些优化效果好,有些优化得不够好。因此用户有必要了解查询优化的概念。希望用户能够写出“好”的查询,执行效率高的语句。特别对于 DBA 人员来说,“系统调优(Performance Tuning)”是他(们)的职责之一,更需要掌握查询优化的概念和 RDBMS 的内部优化技术。

一、基本知识点

本章进一步讲解关系模型和关系系统这两个基本概念。希望读者知道这是两个不同的、又是紧密相关的概念。

为了提高关系数据库系统的执行效率,RDBMS 必须进行查询优化;由于关系查询语言,例如 SQL,具有较高的语义层次,使 RDBMS 可以进行查询优化。这就是 RDBMS 查询优化的必要性和可能性。

需要了解的:关系系统的定义和分类;全关系系统的十二条准则。

需要牢固掌握的:最小关系的系统、关系上完备的系统和全关系型的关系系统等基本概念;什么是关系系统的查询优化。

需要举一反三的:能够画出一个查询的语法树以及优化后的语法树。

难点:本章的难点在于优化算法,包括代数优化算法和物理优化算法。

二、习题解答和解析

1. 试给出各类关系系统的定义:最小关系系统;关系上完备的系统;全关系型的关系系统。

答

最小关系系统:

一个系统可定义为最小关系系统,当且仅当它:

(1) 支持关系数据库(关系数据结构),从用户观点看,关系数据库由表构成,并且只有表这一种结构;

(2) 支持选择、投影和(自然)连接运算,对这些运算不要求定义任何物理存取路径。

关系上完备的系统:

这类系统支持关系数据结构和所有的关系代数操作(或者功能上与关系代数等价的操作)。

全关系型的关系系统:

这类系统支持关系模型的所有特征。即不仅是关系上完备的而且支持数据结构中域的概念,支持实体完整性和参照完整性。

解析

(1) 通过本题,同学要清楚知道不同的关系系统支持关系模型的程度是不同的。

(2) 最小关系系统是指一个 RDBMS 最起码的条件。如果一个数据库厂商声称他的 DBMS 是关系的,那么它必须满足这两个最基本的要求。否则,就不是 RDBMS。例如表式系统、倒排表系统就不能算关系系统。

(3) 关系数据模型是由数据结构、关系操作和完整性约束条件这三部分组成,是按照这三部分内容来考察一个关系系统,并进行分类的。

(4) 《概论》上图 4.1 很直观地给出了不同的系统支持关系模型的程度,读者可以用这个图帮助理解和记忆。

* 2 试述全关系型系统应满足的十二条准则,以及十二条基本准则的实际意义和理论意义。

答

关系模型的奠基人 E. F. Codd 具体地给出了全关系型的关系系统应遵循的十二条基本准则。从实际意义上看,这十二条准则可以作为评价或购买关系型产品的标准。从理论意义上看,它是对关系数据模型具体而又深入的论述,是从理论和实际紧密结合的高度对关系型 DBMS 的评述。

准则 0 一个关系型的 DBMS 必须能完全通过它的关系能力来管理数据库。

准则 1 信息准则。关系型 DBMS 的所有信息都应在逻辑一级上用一种方法即表中的值显式地表示。

准则 2 保证访问准则。依靠表名、主码和列名的组合,保证能以逻辑方式访问关系数据库中的每个数据项(分量值)。

准则 3 空值的系统化处理。全关系型的 DBMS 应支持空值的概念,并用系

统化的方式处理空值。

准则 4 基于关系模型的动态的联机数据字典。数据库的描述在逻辑级上应该和普通数据采用同样的表示方式,使得授权用户可以使用查询一般数据所用的关系语言来查询数据库的描述信息。

准则 5 统一的数据子语言准则。

准则 6 视图更新准则。所有理论上可更新的视图也应该允许由系统更新。

准则 7 高级的插入、修改和删除操作。

准则 8 数据物理独立性。无论数据库的数据在存储表示或存取方法上作任何变化,应用程序和终端活动都保持逻辑上的不变性。

准则 9 数据逻辑独立性。当对基本关系进行理论上信息不受损害的任何改变时,应用程序和终端活动都保持逻辑上的不变性。

准则 10 数据完整性的独立性。关系数据库的完整性约束条件必须是用数据库语言定义并存储在数据字典中的,而不是在应用程序中加以定义的。

准则 11 分布独立性。关系型 DBMS 具有分布独立性。

准则 12 无破坏准则。如果一个关系系统具有一个低级(指一次处理一个记录)语言,则这个低级语言不能违背或绕过完整性准则。

解析

不要求读者背熟全关系型系统应满足的这十二条准则,而是要理解每一条准则的含义。当在选择或购买 RDBMS 时可以按照这些准则的内容来衡量和评价实际的 RDBMS 产品。要学以致用,把这些准则具体化。

3 试述查询优化在关系数据库系统中的重要性和可能性。

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 索引或惟一索引或组合索引)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。

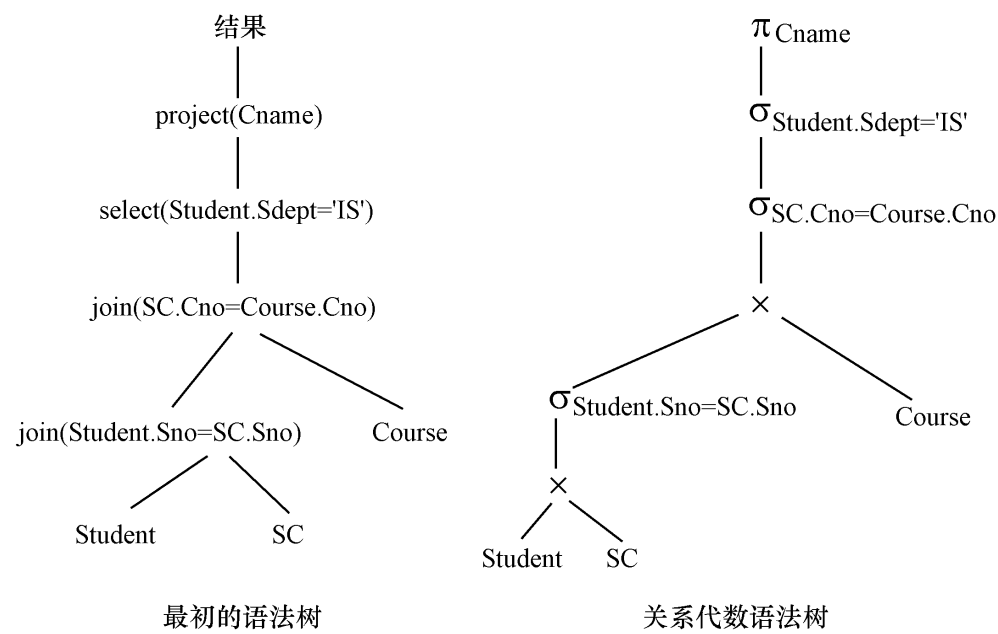
(4) 优化器中包括了很多复杂的优化技术,这些优化技术往往只有最好的程序员才能掌握。系统的自动优化相当于使得所有人都拥有这些优化技术。

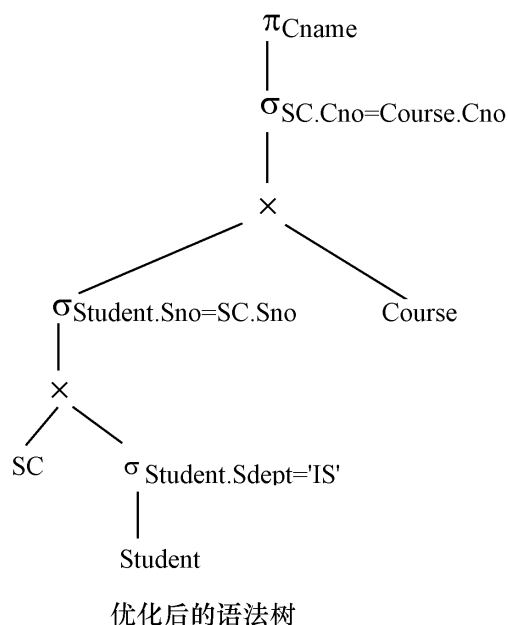
4 对学生 - 课程数据库有如下的查询:

```
SELECT Cname
FROM Student, Course, SC
WHERE Student.Sno = SC.Sno
      AND SC.Cno = Course.Cno
      AND Student.Sdept = IS ;
```

此查询要求信息系学生选修了的所有课程名称。试画出用关系代数表示的语法树,并用关系代数表达式优化算法对原始的语法树进行优化处理,画出优化后的标准语法树。

答





5 试述查询优化的一般准则。

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儿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、笛卡儿积, 与上面的理由类似, 在进行 JOIN 运算、笛卡儿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儿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儿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。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。

6 试述查询优化的一般步骤。

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

为了帮助同学进一步理解查询优化的概念,在这里粗略地画出 SQL 查询处理工作的框图。

查询处理工作的第一步是对 SQL 等高级查询语言所表示的查询进行扫描、语法分析和有效性检查。扫描器从查询语句表达式中识别出语言符号,如 SQL 关键字、属性名和关系名等。

语法分析程序对查询进行语法检查,判断查询表达式是否符合查询语言的语法规则。另外,还必须检查查询的有效性,即根据被查询数据库的模式检查所有的属性名和关系名是否有效,属性名和关系名是否有语义内涵的名字。随后建立查询的内部表示。

查询的内部表示一般用查询树来表达。

查询优化器采用表达式转换、代价估算等优化方法,选择较优的适当的查询处理策略,并生成查询计划。

代码生成器的任务是生成执行这个查询计划的代码。

运行时处理器则负责执行查询代码,并生成查询结果。如果运行时发生了错误,运行时处理器将返回一条错误消息。

