第九章 关系查询处理和查询优化

9.1 关系数据库系统的查询处理

9.1.1 查询处理步骤

查询分析->查询检查->查询优化->查询执行

1、查询分析

判断查询语句是否符合 SQL 语法规则

2、查询检查

合法权检查（关系名，属性名等是否存在且有效）

视图转换（消解视图）

安全性检查

完整性初步检查（用户是否有权限）

检查通过后把 SQL 查询语句转换成内部表示，即等价的**关系代数表达式**

DBMS 一般用查询树表示扩展的关系代数表达式

3、查询优化

有代数优化和物理优化

代数优化：关系代数表达式的优化

物理优化：存取路径和底层造作算法的选择

选择依据：基于规则/代价/语义

4、查询执行

代码执行器生成执行查询计划的代码

两种执行方法：自顶向下，自底向上

9.1.2 实现查询操作的算法示例

1、选择操作的实现

（1）全表扫描方法

适合小表，不适合大表

（2）索引扫描方法

使用于选择条件中的属性上有索引（B+树索引或是 Hash 索引）

通过索引找到元组指针，再找到元组

2、连接操作的实现

（1）嵌套循环算法 最简单可行

（2）排序-合并算法

先进行排序

若两表原来无序，执行时间要加上对两个表的排序时间

（3）索引连接算法

（4）hash join 算法

把连接属性作为 hash 码

划分阶段：对元组少的表进行操作，按 hash 函数映射到桶中

试探阶段：对另一个表进行处理，用同样 hash 函数映射，匹配的连起来

划分阶段又叫创建阶段，试探阶段又叫连接阶段

前提：第一阶段的小表可以完全放入 hash 的桶中

9.2 关系数据库系统的查询优化

查询优化是影响关系数据库管理系统性能的关键因素

9.2.1 查询优化概述

在集中式数据库数据库中，查询开销：键盘存取块数（I/O/代价）、处理机时间（CPU 代价）、查询的内存开销。I/O 最主要

分布式数据库：总代价=I/O 代价+CPU 代价+内存代价＋通信代价

计算查询代价时一般用查询处理读写的块数作为衡量单位

查询优化总目标：选择有效策略，求得关系表达式值，使代价最小（较小）

9.2.2 一个实例 P281

有选择和连接操作时，先做选择操作，这是代数优化

对于两表的连接，利用表上的索引，这是物理优化

9.3 代数优化

9.3.1 关系代数表达式等价变换规则

优化策略：通过对关系代数表达式的等价变换来提高查询效率

9.3.2 查询树的启发式优化

典型的启发式规则

（1） 选择运算尽可能先做 最重要最基本

（2） 投影运算和选择运算同时进行 避免重复扫描

（3） 把投影同其前后的双目运算结合起来

（4） 选择同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 可以先计算一次子表达式把结果写入中间文件

9.4 物理优化

代数优化改变查询语句中操作的次序和组合，不涉及底层的存取路径

物理优化就是要选择高效合理的操作算法或是存取路径来优化查询计划

可选：基于规则的启发式优化：启发式规则是大多数情况都适用的规则不是全部

基于代价估算的优化：优化器估算不同执行策略的代价

两者结合：先使用启发式规则选出一部分再计算代价，最终选出最优

9.4.1 基于启发式规则的寻去路径选择优化

1、选择操作的启发式规则

对于小关系，使用全表顺序扫描，即使选择列上有索引

对于大关系，启发式规则有：

（1） “**主码＝值**”的查询，查询结果最多是一个元组，可以选择主码索引。

（2） “**非主属性＝值**”：并且选择列上有索引，要估算查询结果的元组数目，如果比例较小(<10%)可以使用索引扫描方法，否则还是使用全表顺序扫描。

（3） 对于选择条件是属性上的**非等值查询或者范围查询**，并且选择列上有索引，要估算查询结果的元组数目，如果比例较小(<10%)可以使用索引扫描方法，否则还是使用全表顺序扫描。

（4） AND 连接的合取选择条件

如果有涉及这些属性的组合索引，优先采用组合索引扫描方法

如果某些属性上有一般的索引，可以用索引扫描方法。通过分别查找满足每个条件的指针，求指针的交集；通过索引查找满足部分条件的元组，然后在扫描这些元组时判断是否满足剩余条件

其他情况：使用全表顺序扫描。

（5） OR 条件，全表顺序扫描

2、选择操作的启发式规则

（1）如果 2 个表都已经按照连接属性排序，选用排序-合并算法

（2）如果一个表在连接属性上有索引，选用索引连接算法

（3）如果上面 2 个规则都不适用，其中一个表较小，选用 Hash join 算法

（4）可以选用嵌套循环方法，并选择其中较小的表，确切地讲是占用的块数(b)较少的表，作为外表(外循环的表) 。

9.4.2 基于代价的优化

1、统计信息 P287

2、代价估算示例