

# 查询优化

查询优化在关系数据库管理系统中非常重要，它不仅决定了关系数据库管理系统的查询性能而且降低了数据库的应用难度。在使用关系数据库管理系统时，用户只需要提出“干什么”，而不必指出“怎么干”，系统能够通过某种代价模型计算出各种查询执行计划的执行代价，然后选取代价最小（性能最高）的方案。这是文档数据库管理系统无法比拟的。

## 性能衡量标准

通常，衡量一个算法的性能会考虑算法的时间复杂度和空间复杂度。时间复杂度指执行算法所需要的CPU时钟周期，即CPU代价；空间复杂度指执行算法所需要的内存空间大小，即内存代价。CPU代价越小（时间复杂度越低），内存代价越小（空间复杂度越低）则算法性能越高。

那么，衡量一个查询执行计划的性能需要考虑哪些呢？在集中式关系数据库管理系统中，执行查询执行计划需要将数据从磁盘读入内存，然后再将数据从内存读入寄存器CPU中进行计算。在分布式数据库中，数据还需要经过网络进行传输。以集中式关系数据库系统为例，衡量查询执行计划的性能需要考虑从磁盘读取数据的时间（I/O代价），CPU的计算时间（CPU代价），从内存读取数据的时间和内存空间的开销（合称内存代价），即

查询处理代价=I/O代价+CPU代价+内存代价

在计算机的存储体系结构中，一次CPU的时钟周期约为1ns，一次内存数据访问时延约为100ns，而一次磁盘数据访问时延约为10ms。从数量级上来看，I/O代价远远高于CPU代价和内存代价。因此，计算查询处理代价时只考虑从磁盘读取数据的时间（I/O代价）。通常，一次磁盘访问时延为常量，衡量查询执行计划的代价可以转换为计算查询处理过程中读取的物理数据页。也就是说，读取的物理页越少（I/O次数越少），查询执行计划的性能越高。

## 代数优化

代数优化主要改变查询语句中操作的次序和组合。基于查询处理性能的衡量标准，关系数据库管理系统常使用启发式规则（Heuristic Rules）进行代数优化。典型的启发式规则有：

- 选择运算尽可能先做。这是最重要也是最基本的一条规则。选择运算可以大大减少查询的中间结果。如果内存大小能够容纳中间结果，那么后续的计算就不需要从磁盘读取数据。
- 将投影运算和选择运算同时进行。如果对同一个关系有若干投影和选择运算，那么在读取关系数据同时完成所有的运算，这样可以避免重复地从磁盘读取数据。
- 尽可能地将笛卡尔积转变成连接运算。连接运算（尤其是等值连接）比同等关系上的笛卡尔积更高效。
- 找出公共子表达式。查询语句中重复出现的子表达式称为公共子表达式。如果公共子表达式的查询结果不大，并且从磁盘读取该查询结果的时间少于计算该公共子表达式的时间，则可先计算公共子表达式并把结果写入中间文件。

遵循上述启发式规则，结合一些关系代数表达式等价转换规则就能完成代数优化。关系代数表达式的等价转换规则可以参照其他教材。

## 物理优化

优化后的关系代数表达式中，各代数运算拥有多种操作算法或多种存取路径。比如，选择运算拥有全表扫描（Table Scan）和索引扫描（Index Scan）两种算法；连接运算拥有嵌套循环算法（Nested Loop Join）、排序合并算法（Merge Join）、散列连接算法（Hash Join）和索引连接算法（Index Join）。物理优化就是为各运算选择高效合理的操作算法或存取路径，从而获得优化的查询计划。

物理优化在选择关系运算算法时需要遵循查询处理性能的衡量标准。优化方法包括基于启发式规则的优化、基于代价估算的优化和两者结合的优化方法。

启发式规则是指在大多数情况下都适用，但在不是每种情况下都是最好的规则。选择运算的启发式规则有：

- 小关系的选择运算，通常使用全表扫描，即使选择列上有索引；
- 大关系的选择运算，如果选择列上有索引，且查询结果占整个关系的比例较小（<10%），则使用索引扫描，否则使用全表扫描；如果选择列上没有索引或者由OR连接构成的析取选择条件时，使用全表扫描。

连接运算的启发式规则有：

- 如果连接的两个表都已经按照连接属性排序，则使用排序合并算法；
- 如果一个表在连接属性上有索引，则选用索引连接算法；
- 如果上述两个规则都不适应，且其中一个表较小，则选用散列连接算法；
- 最后选用嵌套循环算法，并将其中较小的表（物理页较少的表）作为左表进行连接运算。

启发式规则的优化是做定性选择，实现简单且优化本身的代价较小。基于代价估算的优化方法是基于数据字典中的统计信息来计算各种操作算法的执行代价，从而选出代价最小的操作算法，这是做定量选择。数据字典中的统计信息主要包括以下几个方面：

- 基本表的信息：该表的元组总数（N）、元组长度（l）、占用的物理页数（B）、占用的溢出物理页数（BO）；
- 基本表属性列的信息：该列不同值的个数（m）、该列最大值、最小值，该列是否建立索引以及索引的类型。基于这些统计信息可以计算出该列值的分布情况以及选择运算的查询结果比例（选择率f）。如果不同值的分布是均匀地，则 $f=1/m$ ；如果不同值的分布不均匀，则 $f=\text{该值的元组数}/N$ ；
- 索引的信息：以B+树索引为例，该索引的层数（L）、索引的叶结点数（Y），不同索引值的个数、索引的选择基数S（有S个元组具有某个索引值）。

如何基于统计信息来计算各操作算法的执行代价将在下一节介绍算法的实现中进行详细的介绍。