如何提升查询优化的效果？

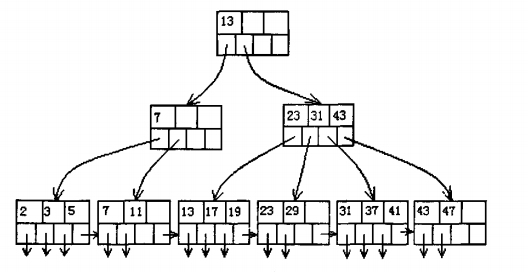
数据库系统是管理信息系统的核心，基于数据库的联机事务处理（OLTP）以及联机分析处理(OLAP)是银行、企业、政府等部门最为重要的计算机应用之一。从大多数系统的应用实例来看，查询操作在各种数据库操作中所占据的比重最大，而查询操作所基于的SELECT语句在SQL语句中又是代价最大的语句。那么如何提升查询优化的效果呢？可以从以下几方面来说。

1、索引技术

在数据相对来说比较少的情况下，一般的select查询语句查询起来速度还是比较快的。但是在查询海量数据的时候，简单的查询语句速度会变得非常的慢，这个时候通过在特定字段上建立索引。索引是一个单独的、物理的数据库结构。它根据表中的一列或者是若干列，按照一定的顺序建立的列值和record之间的对应关系表。首先，主键索引，这个一般是在建表的时候创建的，查询的时候我们通过指定主键可以很快地查询到特定的数据行。需要注意的是主键索引不允许有空值。当然也可以创建基本的普通索引和复合索引。通过创建索引，可以很大程度上提高系统的性能。但是创建索引并且维护索引也需要一定的代价。所以为了提高查询优化的效果，在创建索引的时候，索引创建的字段必须是要是在查询的时候经常会使用到的。一般来说，可以在主键的列，经常被join的外键列，经常需要排序的列，经常使用在where字句中的列上建立索引。当然，要清楚的是索引应该尽量建立在小字段上，对于一些文本字段，最好的原则是不要建立索引，或者可以通过建立前缀索引。当然，索引不应该随便乱建，重复，冗余的索引反而会影响性能。通过索引技术，可以尽量避免进行全表扫描，最主要的是减少了因为查询而造成的I/O性能开销。

由于索引是排好序的，所以可以增加查询的速度。例如在MySQL中没有使用索引的表是一个无序的数据行的集合。这样在一个有1000条数据行的表中，我们查找一个特定ID的数据行，这个需要从头到尾扫描数据表中的每一个数据行，从而进行相应的匹配，这个是非常慢的。但是如果在ID列上建立了索引，那么在查询的时候，会直接扫描索引找到对应的ID列，这就避免了进行线性的扫描，从而可以减少查找的时间。

由于在一个表的存储是由两个部分构成的，其中一部分用来存储表的数据页面，另一部分是存放索引页面的。在进行查询的时候。索引页面首先会被调进内存，然后通过索引找到数据的指针，由指针便可以到数据页面中读取相应的数据。数据库中索引按照存储结构的不同可以分为clustered index和nonclustered index。现在一般来说数据库系统可以使用的索引是B树索引，而B树也能够自动的保持和数据文件大小相适应的索引层次。



B树的结构如上面所示，一般来说分为根、中间非叶子节点，和叶子节点。应用B树进行查找文件，只需要较少的磁盘I/O。还可以使用的索引结构为散列表，这是由散列函数直接计算出来的记录存储得知的一种索引结构。还有常用的索引结构为位图索引和连接索引。

2、join的优化