https://www.cnblogs.com/moxiaotao/p/10144167.html

[**数据库中查询记录时是否每次只能使用一个索引？**](https://www.cnblogs.com/moxiaotao/p/10144167.html)

在网上看一些文章的时候，发现好几次下面这样的话：

如果经常需要同时对两个字段进行AND查询,那么使用两个单独索引不如建立一个复合索引，因为两个单独索引通常数据库只能使用其中一个，而使用复合索引因为索引本身就对应到两个字段上的，效率会有很大提高。

但是，往往都没有说为什么？想知道以下问题：  
1、是不是在任何情况下数据库查询一次只会使用到一个索引？  
2、如果不是，那么什么情况下只会使用一个索引？  
3、那分别是什么造成上面的查询索引使用问题呢？

与其说是“数据库查询只能用到一个索引”，倒不是说是 和全表扫描/只使用一个索引的速度比起来，去分析两个索引二叉树更加耗费时间，所以绝大多数情况下数据库都是是用一个索引。  
如这条语句：

select count(1) from table1 where column1 = 1 and column2 = 'foo' and column3 = 'bar'

我们来想象一下当数据库有N个索引并且查询中分别都要用上他们的情况：  
查询优化器（用大白话说就是生成执行计划的那个东西）需要进行N次主二叉树查找[这里主二叉树的意思是最外层的索引节点]，此处的查找流程大概如下：  
查出第一条column1主二叉树等于1的值，然后去第二条column2主二叉树查出foo的值并且当前行的coumn1必须等于1，最后去column主二叉树查找bar的值并且column1必须等于1和column2必须等于foo。  
如果这样的流程被查询优化器执行一遍，就算不死也半条命了，查询优化器可等不及把以上计划都执行一遍，贪婪算法（最近邻居算法）可不允许这种情况的发生，所以当遇到以下语句的时候，数据库只要用到第一个筛选列的索引（column1），就会直接去进行表扫描了。

select count(1) from table1 where column1 = 1 and column2 = 'foo' and column3 = 'bar'

所以与其说是数据库只支持一条查询语句只使用一个索引，倒不如说N条独立索引同时在一条语句使用的消耗比只使用一个索引还要慢。  
所以如上条的情况，最佳推荐是使用index(column1,column2,column3） 这种联合索引，此联合索引可以把b+tree结构的优势发挥得淋漓尽致：  
一条主二叉树（column=1），查询到column=1节点后基于当前节点进行二级二叉树column2=foo的查询，在二级二叉树查询到column2=foo后，去三级二叉树column3=bar查找。