B+树是B树的一种变形，它更适合实际应用中操作系统的文件索引和数据库索引。定义如下：（为和大多资料保持一致，这里使用阶数m来定义B+树，而不像之前的B树中，使用的是最小度t来定义）

1、除根节点外的内部节点，每个节点最多有m个关键字，最少有m/2个关键字。其中每个关键字对应一个子树（也就是最多有m棵子树，最少有m/2棵子树）；

2、根节点要么没有子树，要么至少有2棵子树；

3、所有的叶子节点包含了全部的关键字以及这些关键字指向文件的指针，并且：

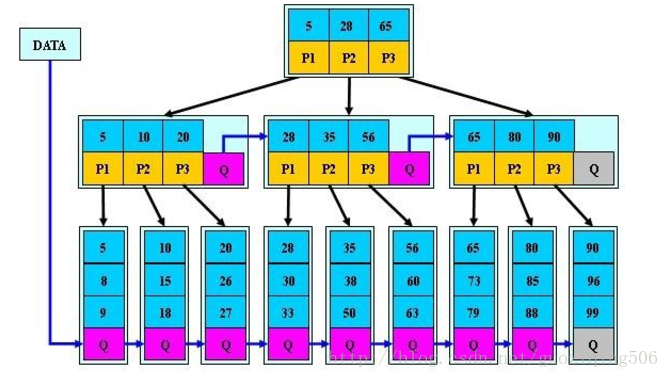
所有叶子节点中的关键字按大小顺序排列

相邻的叶子节点顺序链接（相当于是构成了一个顺序链表）

所有叶子节点在同一层

所有分支节点的关键字都是对应子树中关键字的最大值

比如，下图就是一个非常典型的B+树的例子。



B+树和B树相比，主要的不同点在以下3项：

1、内部节点中，关键字的个数与其子树的个数相同，不像B树种，子树的个数总比关键字个数多1个

2、所有指向文件的关键字及其指针都在叶子节点中，不像B树，有的指向文件的关键字是在内部节点中。换句话说，B+树中，内部节点仅仅起到索引的作用，

3、在搜索过程中，如果查询和内部节点的关键字一致，那么搜索过程不停止，而是继续向下搜索这个分支。

根据B+树的结构，我们可以发现B+树相比于B树，在文件系统，数据库系统当中，更有优势，原因如下：

1、B+树的磁盘读写代价更低

B+树的内部结点并没有指向关键字具体信息的指针。因此其内部结点相对B树更小。如果把所有同一内部结点的关键字存放在同一盘块中，那么盘块所能容纳的关键字数量也越多。一次性读入内存中的需要查找的关键字也就越多。相对来说I/O读写次数也就降低了。

2、B+树的查询效率更加稳定

由于内部结点并不是最终指向文件内容的结点，而只是叶子结点中关键字的索引。所以任何关键字的查找必须走一条从根结点到叶子结点的路。所有关键字查询的路径长度相同，导致每一个数据的查询效率相当。

3、B+树更有利于对数据库的扫描

B树在提高了磁盘IO性能的同时并没有解决元素遍历的效率低下的问题，而B+树只需要遍历叶子节点就可以解决对全部关键字信息的扫描，所以对于数据库中频繁使用的range query，B+树有着更高的性能。