hash索引快为什么mysql还要用B+TREE做索引？

* 只选一个数据，hash更快（o(1)）。数据库中经常会选择N条如果使用hash索引那就是o(n)，这时候由于B+树索引有序(o(logn))，且叶子节点上有链表相连，它的查询效率比hash就快很多了。
* 索引存储在硬盘上，一般来说索引本身很大，不能一次性全部存在内存中，B+树的设计可以允许数据分批加载，同时树的高度较低，查找效率高。
* 硬盘的I/O速度相比内存来说慢很多，而索引是用于加快查询速度的，需要减少I/O操作，内存和磁盘以页为单位交换数据，为了减少I/O，在新建索引节点的时候，是直接申请一个页的空间，存储分配是按页对齐，就实现了一个节点一次I/O

B+树相对于B树：

单一节点存储的元素更多，使得查询的IO次数更少，所以也就使得它更适合作为数据库MySQL的底层数据结构了；

所有查询都要查找到叶子节点，查询性能稳定，而B树，每个节点都可以查找到数据，所以不稳定

所有的叶子节点形成一个有序链表，更加便于查找。

1 b+树的数据都集中在叶子节点。分支节点 只负责索引。  b树的分支节点也有数据 。 b+树的层高 会小于 B树 平均的Io次数会远大于 B+树

2b+树更擅长范围查询。叶子节点 数据是按顺序放置的双向链表。  b树范围查询只能中序遍历。

3索引节点没有数据。比较小。B+树可以吧索引完全加载至内存中。

索引是建立的越多越好越吗

答案当然是否定的，因为：

数据量小的表不需要建立索引,建立会增加额外的索引开销（就像我们读一个只有两页的书，没有必要建立目录）

数据变更需要维护索引,因此更多的索引意味着更多的维护成本

更多的索引意味着也需要更多的空间

数据库索引什么时候会**失效**

如果条件中有or，即使其中有条件带索引也不会使用(这也是为什么尽量少用or的原因，因为or会让数据库管理系统使用全表搜索，而不是索引，关键字“in”也会发生类似的问题)

对于多列索引，不是使用的第一部分，则不会使用索引

like查询是以%开头

如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则不使用索引

如果mysql估计使用全表扫描要比使用索引快,则不使用索引