

67 基于主键的索引是如何设计的，以及如何根据主键索引查询？

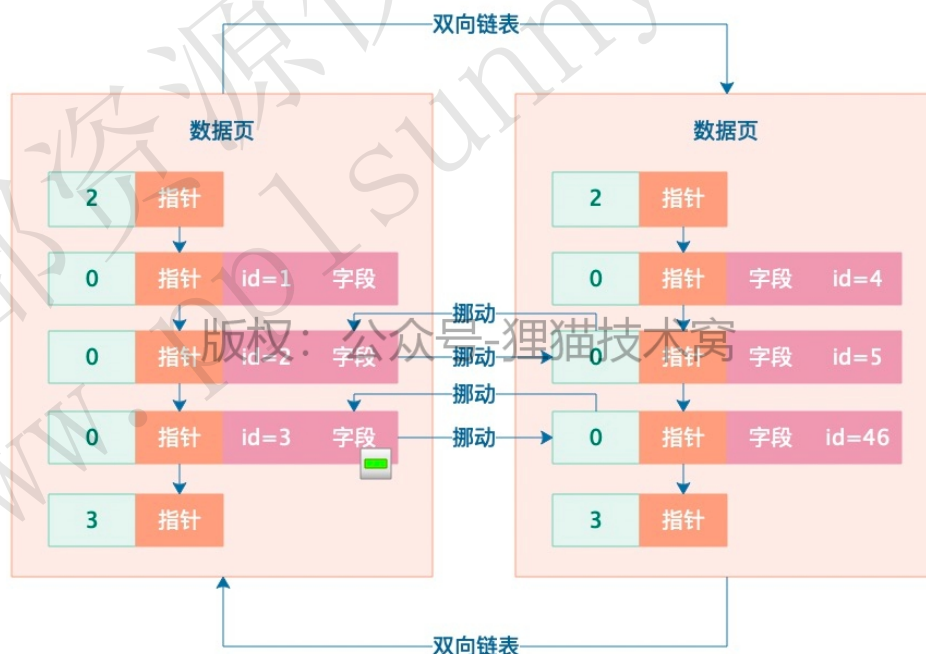
基于主键的索引是如何设计的，以及如何根据主键索引查询？

上回我们说了数据页分裂的过程，在你不停的往表里灌入数据的时候，会搞出来一个一个的数据页，如果你的主键不是自增的，他可能会有一个数据行的挪动过程，保证你下一个数据页的主键值都大于上一个数据页的主键值。

在这个基础之上，我们这一讲终于可以开始正式进入索引原理的分析了，我们就先拿最基础的主键索引来分析，一步一步的给大家把索引原理和查询原理，都讲清楚，接着就可以讲解索引设计案例和SQL调优案例了。

现在是这样的，假设我们有多个数据页，然后我们想要根据主键来查询数据，那么直接查询的话也是不行的，因为我们也不知道主键到底是在哪里，是不是？

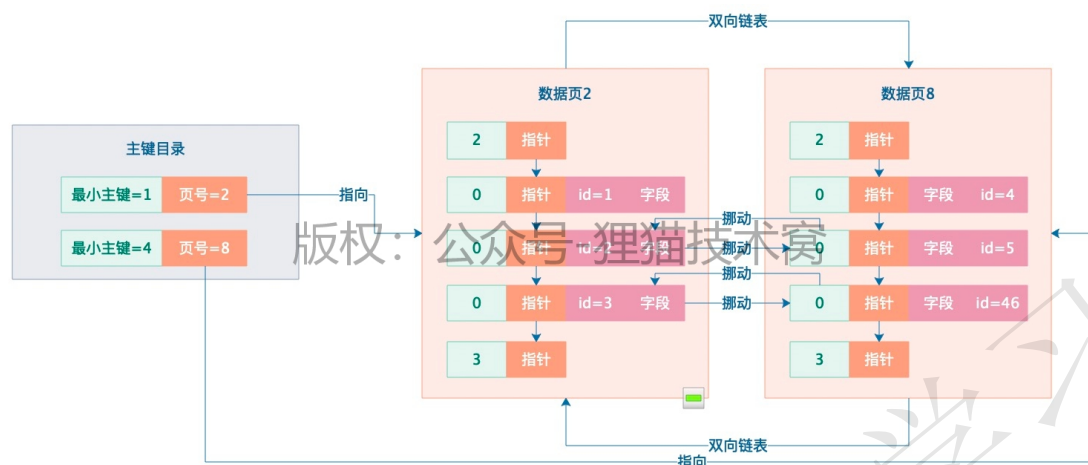
比如下图，大家回顾一下



现在假设你要搜id=4的数据，你怎么知道在哪个数据页里？没有任何证据可以告诉你他到底是在哪个数据页里啊！

所以假设还是这个样子的话，你也就只能全表扫描了，从第一个数据页开始，每个数据页都进入到页目录里查找主键，最坏情况下，所有数据页你都得扫描一遍，还是很坑的。

所以其实此时就需要针对主键设计一个索引了，针对主键的索引实际上就是主键目录，这个主键目录呢，就是把**每个数据页的页号，还有数据页里最小的主键值放在一起，组成一个索引的目录**，如下图所示。



现在有了上图的主键目录就方便了，直接就可以到主键目录里去搜索，比如你要找id=3的数据，此时就会跟每个数据页的最小主键来比，首先id=3大于了数据页2里的最小主键值1，接着小于了数据页8里的最小主键值4。

所以既然如此，你直接就可以定位到id=3的数据一定是在数据页2里的！

假设你有很多的数据页，在主键目录里就会有更多的数据页和最小主键值，此时你完全可以根据二分查找的方式来找你要找到的id到底在哪个数据页里！

所以这个效率是非常之高的，而类似上图的主键目录，就可以认为是主键索引。

而大家都知道我们的数据页都是一坨一坨的连续数据放在很多磁盘文件里的，所以只要你能根据主键索引定位到数据所在的数据页，此时假设我们有别的方式存储了数据页跟磁盘文件的对应关系，此时你就可以找到一个磁盘文件。

而且我们假设数据页在磁盘文件里的位置也就是offset偏移量，你也是可以知道的，此时就可以直接通过随机读的方式定位到磁盘文件的某个offset偏移量的位置，然后就可以读取连续的 一大坨数据页了！

大家看完了今天的文章，不知道有什么感想？

是不是觉得索引也不过尔尔罢了，但是其实我们今天讲的是最简单和基础的一个索引的概念

接下来下一次我们就要讲，到底为什么用B+树来组成一个索引的数据结构，那才是真正的索引！

End