

71 插入数据时到底是如何维护好不同索引的B+树的?

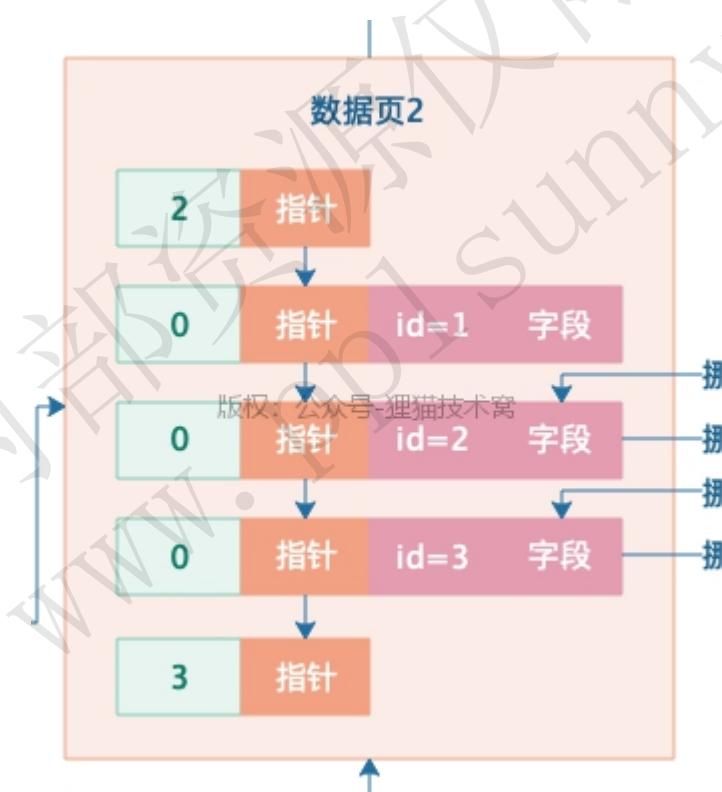
插入数据时到底是如何维护好不同索引的B+树的?

之前我们已经给大家彻底分析清楚了MySQL数据库的索引结构了，大家都知道不同索引的结构是如何的，大致是如何建立的，然后搜索的时候是如何根据不同的索引去查找数据的。

那么今天我们来给大家彻底讲清楚，你在插入数据的时候，是如何维护不同索引的B+树的。

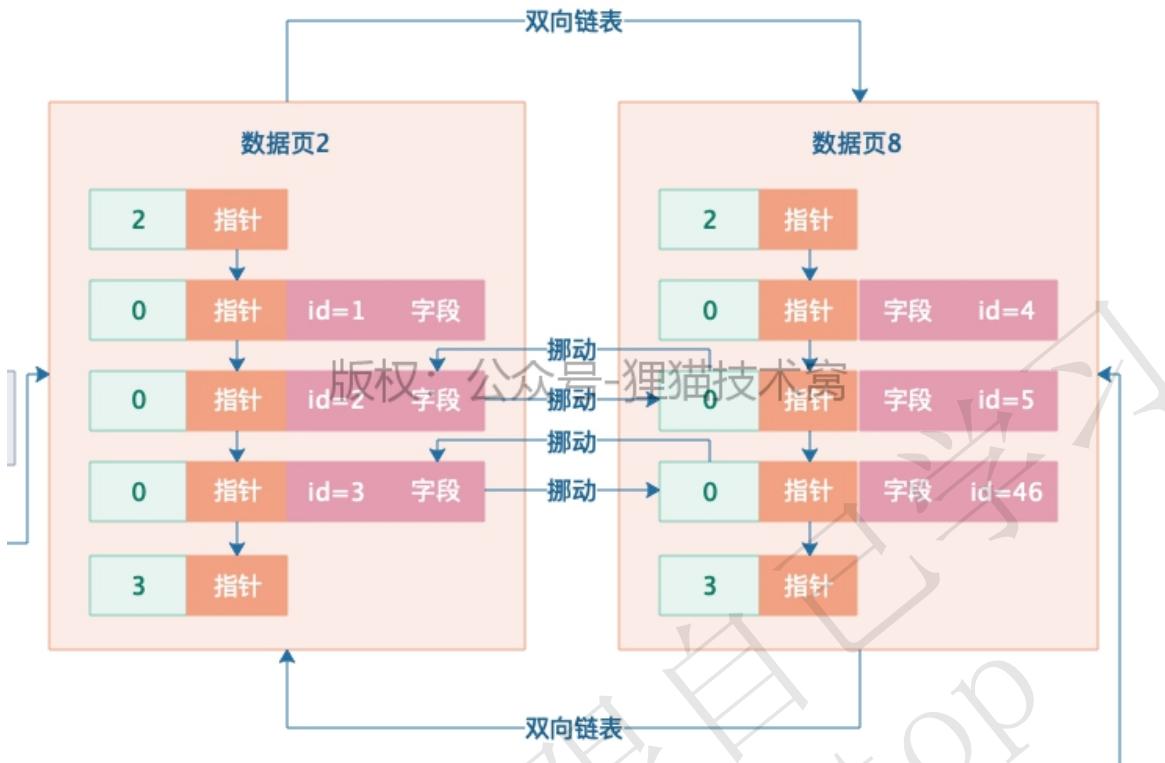
首先呢，其实刚开始你一个表搞出来以后，其实他就一个数据页，这个数据页就是属于聚簇索引的一部分，而且目前还是空的

此时如果你插入数据，就是直接在这个数据页里插入就可以了，也没必要给他弄什么索引页，如下图。



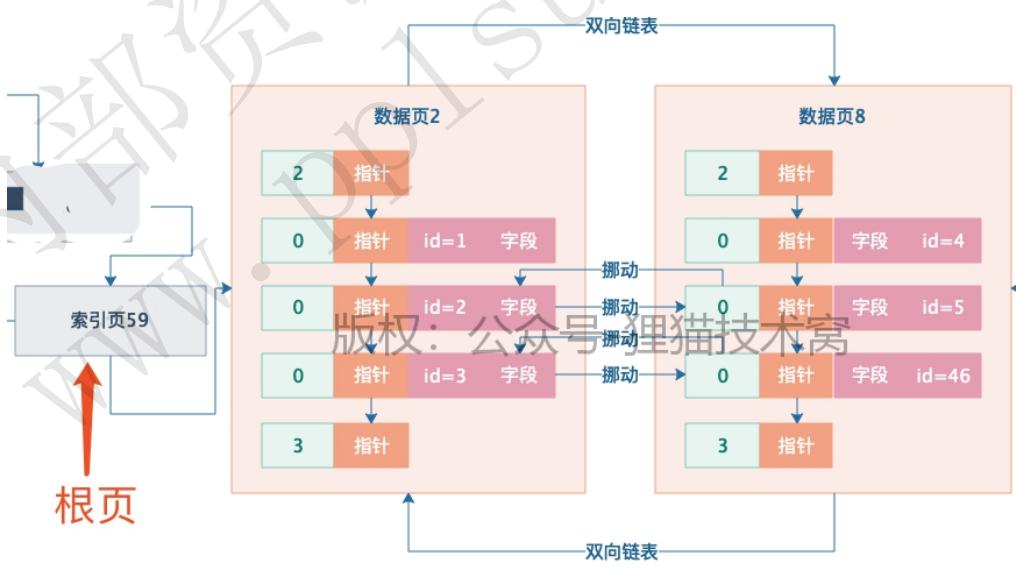
然后呢，这个初始的数据页其实就是一个根页，每个数据页内部默认就有一个基于主键的页目录，所以此时你根据主键来搜索都是ok没有问题的，直接在唯一一个数据页里根据页目录找就行了。

然后你表里的数据越来越多了，此时你的数据页满了，那么就会搞一个新的数据页，然后把你根页面里的数据都拷贝过去，同时再搞一个新的数据页，根据你的主键值的大小进行挪动，让两个新的数据页根据主键值排序，第二个数据页的主键值都大于第一个数据页的主键值，如下图。



那么此时那个根页在哪儿呢？

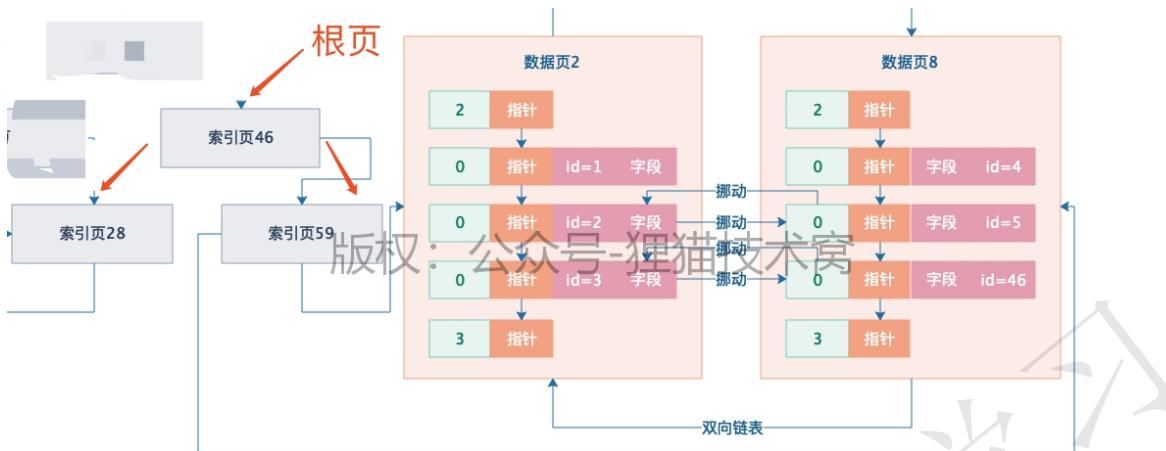
此时根页就升级为索引页了，这个根页里放的是两个数据页的页号和他们里面最小的主键值，所以此时看起来如下图，根页就成为了索引页，引用了两个数据页。



接着你肯定会不停的在表里灌入数据，然后数据页不停的页分裂，分裂出来越来越多的数据页

此时你的唯一一个索引页，也就是根页里存放的数据页索引条目越来越多，连你的索引页都放不下了，那你就让一个索引页分裂成两个索引页，然后根页继续往上走一个层级引用了两个索引页

如下图。



接着就是依次类推了，你的数据页越来越多，那么根页指向的索引页也会不停分裂，分裂出更多的索引页，当你下层的索引页数量太多的时候，会导致你的根页指向的索引页太多了，此时根页继续分裂成多个索引页，根页再次往上提上去一个层级。

这其实就是你增删改的时候，整个聚簇索引维护的一个过程，其实其他的二级索引也是类似的一个原理

比如你name字段有一个索引，那么刚开始的时候你插入数据，一方面在聚簇索引的唯一的数据页里插入，一方面在name字段的索引B+树唯一的数据页里插入。

然后后续数据越来越多了，你的name字段的索引B+树里唯一的数据页也会分裂，整个分裂的过程跟上面说的一样的，所以你插入数据的时候，本身就会自动去维护你的各个索引的B+树。

另外给大家补充一点，你的name字段的索引B+树里的索引页中，其实除了存放页号和最小name字段值以外，每个索引页里还会存放那个最小name字段值对应的主键值

这是因为有时候会出现多个索引页指向的下层页号的最小name字段值是一样的，此时就必须根据主键判断一下。

比如你插入了一个新的name字段值，此时他需要根据name字段的B+树索引的根页面开始，去逐层寻找和定位自己这个新的name字段值应该插入到叶子节点的那个数据页里去

此时万一遇到一层里不同的索引页指向不同的下层页号，但是name字段值一样，此时就得根据主键值比较一下。

新的name字段值肯定是插入到主键值较大的那个数据页里去的。

好了，基本上讲到这里，大家应该对整个索引的数据结构，如何基于索引查询，插入的时候如何维护索引B+树，都有了一个比较清晰地理解了

接下来我们就要讲解MySQL中到底在查询语句里是如何使用索引的，然后单表查询语句的执行原理、多表join语句的执行原理、MySQL执行计划、SQL语句调优。

讲完这些之后，再给大家讲解一些查询优化的调优案例，索引设计案例，基本上大家就对MySQL的日常使用和优化，都有了一个系统性的知识体系了。

End