

通过一步一图来深入理解联合索引查询原理以及全值匹配规则

今天我们来通过一步一图的方式，深入理解一下多个字段组成的联合索引查询原理，以及使用索引的全值匹配的规则。

之所以讲解联合索引，那是因为平时我们设计系统的时候一般都是设计联合索引，很少用单个字段做索引，原因之前讲过，我们还是要尽可能的让索引数量少一些，避免磁盘占用太多，增删改性能太差。

另外，单个字段的索引组织结构和查询原理，之前其实我们都讲解的很清楚了，没必要在重复了。

现在我们来假设一下，咱们有一个表是存储学生成绩的，这个表当然有id了，这个id是一个自增主键，默认就会基于他做一个聚簇索引，这个就不用多说了。

然后呢，就是包含了学生班级、学生姓名、科目名称、成绩分数四个字段，平时查询，可能比较多的就是查找某个班的某个学生的某个科目的成绩。

所以，我们可以针对学生班级、学生姓名和科目名称建立一个联合索引。

接着我们画了下面的一个图，这个图就展示了这个三个字段组成的联合索引的部分内容，大家看一下。

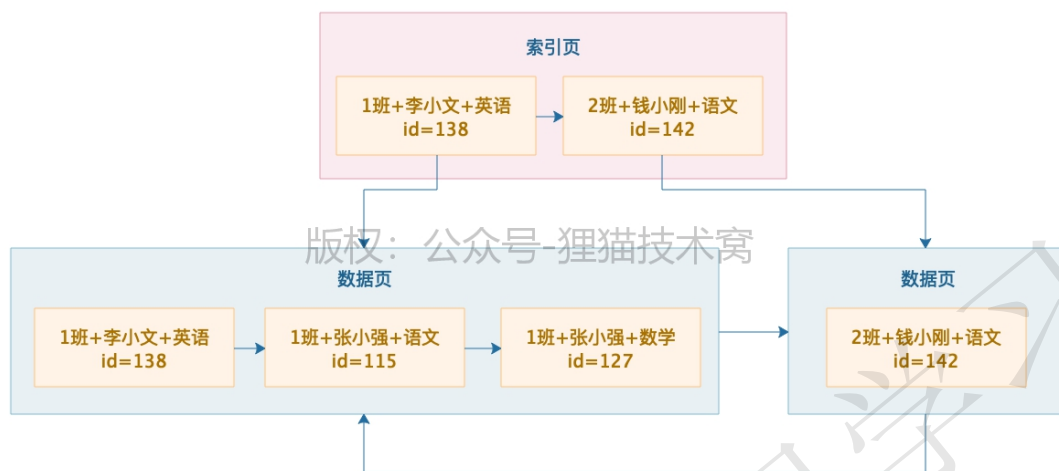
下面有两个数据页，第一个数据页里有三条数据，每条数据都包含了联合索引的三个字段的值和主键值，数据页内部是按照顺序排序的。

首先按照班级字段的值来排序，如果一样则按照学生姓名字段来排序，如果一样，则按照科目名称来排序，所以数据页内部都是按照三个字段的值来排序的，而且还组成了单向链表。

然后数据页之间也是有顺序的，第二个数据页里的三个字段的值一定都大于上一个数据页里三个字段的值，比较方法也是按照班级名称、学生姓名、科目名称依次来比较的，数据页之间组成双向链表。

索引页里就是两条数据，分别指向两个数据页，索引存放的是每个数据页里最小的那个数据的值，大家看到，索引页里指向两个数据页的索引项里都是存放了那个数据页里最小的值！

索引页内部的数据页是组成单向链表有序的，如果你有多个索引页，那么索引页之间也是有序的，组成了双向链表。



好了，那么现在假设我们要搜索：1班+张小强+数学的成绩，此时你可能会写一个类似下面的SQL语句，`select * from student_score where class_name='1班' and student_name='张小强' and subject_name='数学'`。

此时就涉及到了一个索引使用的规则，那就是你发起的SQL语句里，`where`条件里的几个字段都是基于等值来查询，都是用的等于号！而且`where`条件里的几个字段的名称和顺序也跟你的联合索引一模一样！此时就是等值匹配规则，上面的SQL语句是百分百可以用联合索引来查询的。

那么查询的过程也很简单了，首先到索引页里去找，索引页里有多个数据页的最小值记录，此时直接在索引页里基于二分查找法来找就可以了，先是根据班级名称来找1班这个值对应的数据页，直接可以定位到他所在的数据页，如下图。



然后你就直接找到索引指向的那个数据页就可以了，在数据页内部本身也是一个单向链表，你也是直接就做二分查找就可以了，先按1班这个值来找，你会发现几条数据都是1班，此时就可以按照张小强这个姓名来二分查找，此时会发现多条数据都是张小强，接着就按照科目名称数学来二分查找。

很快就可以定位到下图中的一条数据，1班的张小强的数学科目，他对应的数据的id是127，如下图所示。



然后就根据主键id=127到聚簇索引里按照一样的思路，从索引根节点开始二分查找迅速定位下个层级的页，再不停的找，很快就可以找到id=127的那条数据，然后从里面提取所有字段，包括分数，就可以了。

上面整个过程就是联合索引的查找过程，以及全值匹配规则，假设你的SQL语句的where条件里用的几个字段的名称和顺序，都跟你的索引里的字段一样，同时你还是用等号在做等值匹配，那么直接就会按照上述过程来找。

对于联合索引而言，就是依次按照各个字段来进行二分查找，先定位到第一个字段对应的值在哪个页里，然后如果第一个字段有多条数据值都一样，就根据第二个字段来找，以此类推，一定可以定位到某条或者某几条数据！

End