

Oracle数据库中索引树的结构与块尺寸 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_Oracle\\_E6\\_95\\_B0\\_E6\\_c102\\_143720.htm](https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E6_95_B0_E6_c102_143720.htm) 关于Oracle索引树的结构以及它们

对Oracle性能调优是否重要存在大量的、激烈的争论，而且已经有很多文章试图来描述这些重要的Oracle性能工具的内部工作机制。关于这个论题也出现了一些新书，例如由“国际Oracle用户组”（IOUG）主席Kim Floss所著的《Oracle索引

管理秘诀》和《Oracle SQL 性能调优和“基于代价的优化器”内幕》。正如我们知道的，Oracle提供了大量索引结构，

每种索引结构都有其好处和不足：

- \* B树索引。从最早的Oracle发行版本开始，Oracle就一直使用的标准树索引就是B树索引。

- \* 位图索引。当某个索引列含有很少数量的不同的值（即低基数）时，使用位图索引。这对于那些只读数据库而言速度超快，但对需要经常性更新的系统不适合。

- \* 位图连接索引。这是针对来自其他表的数据列出现在某个连接表的多列索引时使用的索引。下面是在from子句和where子句中使用类似SQL的创建索引的惟一语法。

```
create bitmap index  
part_suppliers_state on inventory( parts.part_type, supplier.state  
) from inventory i, parts p, supplier s where i.part_id=p.part_id and  
i.supplier_id=p.supplier_id.
```

尽管有关索引重建的争论仍在激烈进行着，但还是存在每个人都认可的索引管理的某些领域。

在内部机制上，一个Oracle B-树索引的结构和一个UNIX I-结点的结构非常相似。索引中的每个数据块都是索引树中的一个结点，位于最底部的结点（叶数据块）包含一对符号键和行ID值。

Oracle b-树索引 为了正确管理这些数据块，Oracle

控制着每个数据块中指针的分配。随着一棵Oracle树的增长（通过往表里插入新行），Oracle会填充这个数据块，当这个数据块满时Oracle会分裂它，创建新的索引结点（数据块）来管理索引内的符号键。因此，一个Oracle索引块可能包含以下两种类型的指针：\* 指向其他索引结点（数据块）的指针 \* 指向数据库表中特定行的行ID指针 Oracle管理着索引块内指针的分配，这就是为什么我们不能为索引指定一个PCTUSED值（自由列表重链接门槛）的原因。当我们检查一个索引块的结构时，我们发现每个索引结点内部条目的数量是下面两个值的一个函数：1. 符号键的长度 2. 索引表空间的块尺寸 由于块尺寸影响每个索引结点内部的符号键的数量，可以推理出：块尺寸对一棵索引树的结构也会有影响。在其他条件相同的情况下，采用32K的大数据块能容纳更多的符号键，从而能够比在2K表空间中创建的相同的索引更加平整。采用大的数据块也将减少索引访问期间一致获取的数量，从而提高分散读访问的性能。索引中的每个数据块包含索引树中的“结点”，位于最底部的结点（叶数据块）包含一对符号键和行ID值。随着一棵Oracle树的增长（通过往表里插入新行），Oracle会填充这个数据块，当这个数据块满时Oracle会分裂它，创建新的索引结点（数据块）来管理索引内的符号键。因此，一个Oracle索引块可能包含指向其他索引结点或行ID/符号键对的指针。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)