

使用Oracle9i全索引扫描快速访问数据 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__E4_BD_BF_E7_94_A8Orac_c102_143695.htm 为了实现Oracle关于增加SQL优化查询智能的承诺，Oracle9i增强了全索引SQL执行计划以支持基于功能的索引（function-based index）。在Oracle8中，SQL优化器添加了判断查询是否可以专门用一个现存的索引来解决的智能。一旦存在索引，Oracle就会绕过对表的访问，索引组织表（index-organized table，IOT）结构就是一个例子。在IOT结构中，所有的数据都载入索引的b-树结构，这样表（table）就成为一个多余的东西了。一旦Oracle SQL优化器检测到查询无需访问表时，Oracle就调用全索引扫描并快速读取每一个索引块而无需接触表本身。有一点很重要：全索引扫描并没有读取索引节点，而是一块一块的执行扫描并快速捕获索引节点。最好，Oracle调用多块读取功能，调用多个过程来读取表。Oracle和多块读取 为了加快表和索引的访问速度，Oracle使用了db_file_multiblock_read_count参数（默认参数为8）来辅助把全表扫描和全索引扫描所获得的数据块尽快送到数据缓冲区中。然而，这个参数只有当SQL查询执行全表扫描时才可用，并且，在绝大多数情况下，查询要使用索引来访问表。Oracle对全索引扫描有如下限制：SQL请求的全部列（column）必须驻留在索引树中；也就是说，SELECT和WHERE字句中的所有数据列必须存在于索引中。查询访问大量的行（row）。根据你查询的范围，比例变化范围为10%到25%之间，这个比例参数db_file_multiblock_read_count的设置和查询的并行程度极大的影响到这个比例。由于索引节

点并没有按索引顺序排列，所以列并没有顺序。这样，ORDER BY子句将要求附加的排序操作。Oracle提供了一个SQL提示（hint）来强制全索引扫描。你也可以通过指定index_ffs提示来强制快速索引扫描，这常常与parallel_index提示组合来提高性能。例如，下面的查询强迫使用并行快速全索引扫描：

```
0select distinct /* index_ffs(c,pk_auto)
parallel_index_ (automobile, pk_auto) color, count(*) from
automobiles group by color.
```

由于涉及了所有的变量，所以全索引是否会加快查询速度并不能简单的加以回答。所以，大多数有经验的SQL调试者（tuner）会对符合快速全索引扫描标准的查询进行手工计时，看看使用全索引扫描的反映时间是否会降低。在Oracle9i之前，全索引扫描只有当创建的索引没有空值时才可以使用，也就是说，Oracle建立索引时必须用一个NOT NULL子句才可以使用该索引。由于Oracle9i支持用函数（function-based）索引实现的唯索索引扫描，这种情况大大改观。简单回归一下，函数索引是Oracle8的一个重要改进，因为它提供了一种有效的消除无必要长表全扫描的机制，由于函数索引可以在任何查询语句中的WHERE子句中精确复制，Oracle总会用一个索引来匹配上SQL查询的WHERE子句。现在，我要会用一个student表来举一个简单的例子来解释全索引扫描如何与函数索引一起工作的过程。创建student表：(student_name varchar2(40), date_of_birth date).使用这个表，创建与表中所有列相关联的函数索引。在本例中，该函数为initcap（即大写每个单词的首字母）和to_char（即把一个数字变成字符）：

```
create index whole_student on student
(initcap(student_name), to_char(date_of_birth, ' MM-DD-YY
```

')).定义完函数索引后，Oracle9i中任何可以引用这些列（column）的SQL语句都可以使用全索引扫描。下面是SQL匹配函数索引的查询例子：

```
0select * from student where  
initcap(student_name) = ' Jones ' .0select * from student where  
to_char(date_of_birth, ' MM-DD=YY ' ) = ' 04-07-85 ' .
```

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