TURING 图灵程序设计丛书 数据库系列

Apress®

Expert Indexing in Oracle Database 11g

Maximum Performance for your Database

Oracle索引技术

- 唯一专注Oracle数据库索引技术的图书
- 有的放矢，针对性地提升Oracle性能
- Oracle DBA进阶必备

[美] Darl Kuhn Sam R. Alapati Bill Padfield 著
卢涛 译
李颖 审

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 数据库系列

Expert Indexing in Oracle Database 11g

Maximum Performance for your Database

Oracle索引技术

[美] Darl Kuhn Sam R Alapati Bill Padfield 著
卢涛 译
李颖 审

人民邮电出版社
北京

图书在版编目（CIP）数据

Oracle索引技术 /（美）库恩（Kuhn,D.），（美）阿拉帕提（Alapati,S.R.），（美）帕德菲尔德（Padfield,B.）著；卢涛译. -- 北京：人民邮电出版社，2013.1
（图灵程序设计丛书）
ISBN 978-7-115-29626-9

Ⅰ. ①O… Ⅱ. ①库… ②阿… ③帕… ④卢… Ⅲ. ①关系数据库－数据库管理系统－索引方法 Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2012)第241036号

内 容 提 要

本书由三位经验丰富的顶级 Oracle DBA 联手打造，介绍了 Oracle 的各种索引类型。索引是用于提高查询性能的数据库对象，但如果使用不当，也可能导致查询性能下降。作者分享了多年的实践经验和个人智慧，帮助读者避免误用索引。此外，本书还提出了很多索引管理和维护的建议。

无论是 Oracle 数据库开发人员还是 DBA，都能从本书中获益。

图灵程序设计丛书

Oracle索引技术

◆ 著 [美] Darl Kuhn Sam R. Alapati Bill Padfield
译 卢 涛
审 李 颖
责任编辑 王军花
执行编辑 李 静

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：13.75
字数：325千字 2013年1月第1版
印数：1-3 500册 2013年1月北京第1次印刷

著作权合同登记号 图字：01-2012-1812号

ISBN 978-7-115-29626-9

定价：49.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223

反盗版热线：(010)67171154

版权声明

译 者 序

在应用程序开发工作中，之所以要用到数据库，很大一方面原因是数据库能确保数据的安全和高效存取，这样我们可以专心开发与业务相关的功能，而不必操心对底层数据结构进行操作的具体实现。数据库利用各种方式维护数据的完整性、一致性，并提高数据查询和更改操作的效率，其中索引是非常重要的一种方式。Oracle体系结构庞大，它的索引种类繁多、特点各异，每种都有适用的场合和值得注意之处。随着版本的更新，Oracle不断增加新的索引类型。要开发正确、高效的应用程序，必须在一开始就计划使用索引，并在设计中包含它们。而不是等应用程序编写完成后，再让DBA去创建索引，虽然有时需要根据业务的发展修改索引。要在编写应用程序过程中利用好索引，需要深入了解各种索引的功能和特点。否则如果用错索引，不但不能获益，还会给应用程序带来不良的影响。在应用程序投入运行之后，监控、管理和维护索引，对于保证应用程序正常运行必不可少。所以，索引的使用贯穿应用程序的生命周期，不是用几条简单的规则就能概括的，值得用一本书去讲解。

本书由资深Oracle数据库专家编写，专业技术出版公司Apress出版。深入浅出地介绍了Oracle各种类型索引的特点和实际用法，并对11g版本新增的几种类型，如不可见索引和虚拟索引进行了详细讲解。本书既介绍了利用索引改善查询和进行数据完整性约束，也提出了管理和维护索引的有用建议，最后两章还介绍了两种与索引有关的调优工具，可以帮助我们正确地创建真正有用的索引。无论是Oracle数据库开发人员还是DBA，都可从本书中获益。

这是我翻译的第二本书。第一本书是《Oracle PL/SQL实战》，原书也是由Apress出版的。

快要翻译完《Oracle PL/SQL实战》时，我通过傅志红副总经理联系到了朱巍编辑，有幸通过了试译。在接下来4个月的时间里，完成了初译、复查、修改等工作。

首先感谢我妻子李颖，在帮我审阅了《Oracle PL/SQL实战》后，又帮助我检查本书译文中存在的很多问题，并把它修改得更加通顺。另一方面，她花了很多时间照顾我儿子的起居和学习，这样我才有时间做这项工作。

感谢图灵公司朱巍编辑、岳新欣编辑和李静编辑，她们对译文进行了很多润色，使之错误更少。我从她们修改的译文中也收获良多。

最后还要感谢我儿子卢〇一，他知道我在翻译文章就不再缠着我陪他玩，让我能够专心翻译。本书的出版也有他的一份贡献。

最后希望这本书对读者有所帮助。由于译者经验和水平有限，译文中难免有不妥之处，恳请读者批评指正！

致　谢

特别感谢首席编辑Jonathan Gennick对本书内容和组织结构提出的许多宝贵建议。非常感谢Karen Morton提出的大量建议，这些建议大大提高了本书的质量和技术含量。能找到像Karen这样技艺精湛、睿智又有名望的人做本书的技术评审，实在是荣幸之至！当然，书中遗留的任何错误，都是作者的责任。还要感谢协调编辑Anita Castro做的许多“分外”工作，使本书能如期完成。除了她“分内”的工作，轮番处理三个作者的写作内容本身就是一项艰巨的任务。同时感谢执行编辑Mary Behr出色的工作。感谢这个敬业且才华横溢的团队，没有你们，就没有本书。

个人致谢

感谢辛勤工作的合著者伙伴Sam R. Alapati和Bill Padfield，也感谢多年来帮助过我的众多DBA和开发人员，他们是：Scott Schulze、Dave Jennings、Bob Suehrstedt、Ken Toney、Pete Mullineaux、Janet Bacon、Sue Wagner、Mohan Koneru、Arup Nanda、Charles Kim、Bernard Lopuz、Barb Sannwald、Tim Gorman、Shawn Heisdorffer、Sujit Pattanaik、Ken Roberts、Roger Murphy、Mehran Sowdaey、Kevin Bayer、Guido Handley、Dan Fink、Nehru Kaja、Tim Colbert、Glenn Balanoff、Bob Mason、Mike Nims、Brad Blake、Ravi Narayanaswamy、Abdul Ebadi、Kevin Hoyt、Trent Sherman、Sandra Montijo、Jim Secor、Maureen Frazzini、Sean Best、Patrick Gates、Krish Hariharan、Buzzy Cheadle、Lori Beer、Liz Brill、Ennio Murroni、Gary Smith、Dan Truman、Joey Canlas、Eric Wendelin、Mark Lutze、Kevin Quinlivan、Dave Bourque、John Lilly、Dave Wood、Laurie Bourgeois、Steve Buckmelter、Casey Costley、John DiVirgilio、Valerie Eipper、John Goggin、Brett Guy、Kevin O'Grady、Peter Schow、Jeff Shoup、Mike Tanaka、Todd Wichers、Doug Cushing、Kye Bae、Will Thornburg、Ambereen Pasha、Steve Roughton、Sudha Verma、Dinesh Neelay、Ann Togasaki、Thom Chumley、Lea Wang、Steve Odendahl、Ken Kadonaga、Vasa Dasan、Erik Jasiak、Tae Kim、Jeff Sherard、Aaron Isom、Kristi Jackson、Karolyn Vowles、Terry Roam、Darin Christensen、Max Rose、Doug Drake、Jim Johnson、Marilyn Wenzel、Doc Heppler、Mert Lovell、Ken Sardoni、Kimball Moore、Brian Beasly、Clair Larsen、Odean Bowler、Jim Stark、Robbie Robertson、Gary Plessinger、Donna Zwiller、Brighton Bigler、Kit Ashworth、Lasse Jansen、Debra Rimmer和 Harmon Faleono。

Darl Kuhn

这是我们三个合著的第二本书，我真的很幸运有机会与这么牛的两位专家合作。他们两人都是顶尖的Oracle数据库管理员，给我不断的鼓舞及支持。他们极具幽默感，这已使本书的写作过程充满乐趣，更不用说他们极其慷慨无私的帮助了。

迟来的感谢献给纽约花旗银行的Ram Janardhanan和Anil Sinha，感谢他们在我的职业生涯中的巨大帮助。

和往常一样，当我写书时，我的家人会作出很大的牺牲，使我能全身心地投入到这本书的规划和写作中去。我非常感谢我的妻子Valerie和孩子Shannon、Nicholas和Nina的鼎力支持和帮助。最后，我要感谢其他家庭成员：我的母亲Swarna Kumari、我的父亲Appa Rao、我的兄弟Hari Hara Prasad和Siva Sankara Prasad以及Aruna、Vanaja、Ashwin、Teja、Aparna和Soumya，感谢他们对我的不断支持、鼓励、亲情和爱。

Sam R. Alapati

我想感谢亲密的合著者Sam R. Alapati和Darl Kuhn，感谢他们接纳我这个写作新秀，以及对我的所有帮助和支持。若没有他们的帮助，我不可能完成这项任务。

在我的职业生涯中，有这么多曾帮助过我的人值得我感谢，所以请相信，我感谢每一位鼓励和帮助过我的人。首先，我要感谢Bob Ranney，是他让我有机会成为一名DBA。我还要感谢我的一些领导一直以来对我的帮助，他们是Beth Bowen、Larry Wyzgala、John Zlamal、Linda Scheldrup、Amy Neff和Maureen Frazzini。

当然，也有许多DBA、开发人员、系统管理员和架构师在我的职业生涯中给予我巨大帮助。首先，我要感谢我目前工作团队中的DBA，他们使每天都充满激情。这些人曾经给予我很多专业的帮助，我们一起工作过多年，并已成为很好的朋友。他们是Dave Carter、Debbie Fitzgerald、Pankaj Guleria、Pete Sardaczuk、Brad Strom和Rebecca Western。

多年来，我从下面名单中列出的人那里学到了很多东西，他们一直以来慷慨地花费大量时间来帮助我，并非常耐心地解答我的问题：Mark Nold、Mick McMahon、Sandra Montijo、Jerry Sanderson、Glen Sanderson、Jose Fernandez、Mike Hammontre、Pat Cain、Dave Steep、Gary Whiting、Ron Fullmer、Becky Enter、John Weber、Avanish Gupta、Scott Bunker、Paul Mayes、Bill Read、Rod Ermish、Rick Barry、Sun Yang、Sue Wagner、Glenn Balanoff、Linda Lee Burau、Deborah Lieou-McCall、Bob Zumpf、Kristi Sargent、Sandy Hass、George Huner、Pad Kail、Curtis Gay、Ross Bartholomay、Carol Rosenow、Scott Richards、Sheryl Gross、Lachelle Shambe、John Piel、Rob Grote、Rex Ellis、Zane Warton、Steve Pearson、Jim Barclay、Jason Hermstad、Shari Plantz-Masters、Denise Duncan、Bob Mason、Brad Blake、Mike Nims、Cathie Wilson、Rob Coates、Shirley Amend、Rob Bushlack、Cindy Patterson、Debbie Chartier、Blair Christensen、Meera Ganesan、Kedar Panda、Srivatsan Muralidaran、Kevin Tomimatsu、John Townley和Brent Wagner。

Bill Padfield

目　录

第1章 Oracle索引

索引是一种可以选择创建的数据库对象，它主要用于提高查询性能。数据库索引的用途与一本书后面的索引类似。书的索引把书的主题和页码进行关联。想在一本书中查找信息时，首先检查索引，从中找到要查的主题，确定相关的页码，通常比直接翻书查找要快得多。有了索引提供的信息，就可以直接翻到这本书中的具体页码。如果某个主题只在书的几页内出现，那么读取的页面数量是很少的。采用这种方式，一个主题在书中出现的次数越多，索引对它产生的作用就越小。

与书的索引类似，数据库索引把用户感兴趣的列值连同其行标识符（ROWID）存储在一起。ROWID包含了存储列值的表行在磁盘上的物理位置。有了ROWID，Oracle可以通过最少量的磁盘读取，有效地检索表中的数据。采用这种方式，索引的功能就像表中数据的快捷方式。如果没有可用的索引，那么Oracle就必须读取表中的每一行，才能确定该行是否包含所需的信息。

注意 除了提高性能，Oracle还使用索引来协助强制执行已启用的主键和唯一键约束。此外，当为外键列建立索引时，Oracle可以更好地管理表锁定的情况。

虽然也可以构建不带索引的数据库应用程序，但该程序性能往往是很差的。即使对于非常大的数据集，索引也具有出色的可伸缩性。那么，既然索引对数据库性能是如此重要，为什么不为所有的表和列组合创建索引呢？答案很简单，索引不是没有代价的，它们消耗磁盘空间和系统资源。在列值被修改的同时也必须更新相应的索引。因此，索引使用了存储空间、I/O、CPU和内存资源。创建十分糟糕的索引，会浪费磁盘空间，并且会过度消耗系统资源，也会导致数据库的性能下降。

由于这些原因，在设计和构建基于Oracle数据库的应用程序时，必须从极专业的角度考虑索引策略。作为一名应用程序架构师，你必须了解索引的属性，知道有什么类型的索引可用，并懂得应该选择哪些表和列的组合来创建索引。要想让数据库实现最高性能，正确的索引方法是十分关键的。

本章将介绍Oracle索引的概念。开始我们用一个简明扼要的例子介绍索引是如何提高查询性能的。然后，解释Oracle中可用的索引类型并提供一些指导方针和建议，帮你决定选择为哪些列

创建索引。如果你不太清楚如何使用索引，或需要重温相关知识，那么请从这里开始。

1.1　用索引提高性能

索引究竟是如何提高查询性能的呢？要了解索引的工作原理，可以参考如下简单的例子。假设你创建了一个表用来保存客户信息，它的结构类似下面这样：

```
create table cust
(cust_id    number
,last_name  varchar2(30)
,first_name varchar2(30));
```

假设由于业务增长很快，因此在很短的时间内就创建了数以百万计的客户。每日你都对此表执行报表查询，并注意到发出下面这样的查询语句时，性能在逐步降低。

```
select cust_id, last_name, first_name
from cust
where last_name = 'STARK';
```

当表中数据很少时，该查询一瞬间就返回结果。现在，表中有超过一百万行的数据，而且数据还在继续增长，这个查询花费的时间就越来越长。这是怎么回事？

当执行一个SELECT语句时，Oracle查询优化器快速计算出每一步的执行计划，这个计划详细地说明了将如何检索查询中指定的列值。在计算该计划时，优化器确定检索数据将用到哪些表和索引。

当不存在索引时，表本身是能满足查询结果的唯一访问路径。在这种情况下，Oracle别无选择，只能检查表的每一数据块内的每一行（这被称为*全表扫描*），看看是否有姓（last_name）是STARK的行。随着表中插入越来越多的数据，查询需要的时间越来越长。此查询的成本（以CPU、内存和I/O资源消耗来衡量）与表的数据块的数量成正比。要想使这个查询运行速度更快，只能购买更好的硬件或使用一种增强性能的功能，如索引。

你可以“向前”翻阅一下本章内容，这样可以确定SQL查询的WHERE子句中出现的列的索引可能会提高性能，于是我们为CUST表的LAST_NAME列创建索引，像这样：

```
create index cust_idx1
on cust(last_name);
```

这条语句创建一个B树索引（稍后详细解释）。这是Oracle默认的索引类型。创建索引后，按last_name选择的查询用时就能恢复到一秒内。不错吧？

要了解索引是如何提高性能的，首先要记得索引存储了两种类型的信息：表中的列值和相应的ROWID。在数据库内，ROWID可唯一标识一行（对于堆组织表），并包含其物理位置（数据文件和数据块内行的位置）。创建了索引并执行后续查询后，查询优化器判断该索引是否能减少返回查询结果所需的资源量。

提示　ROWID唯一标识堆组织表的一行。然而，对于聚簇表，有可能出现这种情况：不同表中的行位于同一数据块内，并具有相同的ROWID。

在前一个例子中，假设CUST表中的记录有数百万，但表中只有一个记录的LAST_NAME是STARK。那么查询优化器可以检查索引，并仅通过几个磁盘读，就能找到表中某个块的确切位置（通过ROWID），其中包含符合查询条件的记录，这使得查询速度非常快。在这种情况下，即使表中有数百万行记录也不要紧，只要该索引所包含的值是相当独特的，Oracle都能够通过极少量的磁盘读取操作返回所需的行。

相反，考虑一下LAST_NAME列的值不是很独特的情况。假设CUST表中有几百万的记录都包含了LEE值。如果查询优化器使用索引，就必须从索引中读取数百万次，检索出所有的ROWID，然后再（通过ROWID）从表读数百万次。在这种情况下，不使用索引而直接扫描表中每个块的速度更快。出于这个原因，有时优化器计算出使用索引并不能提高性能，就会忽略它。

提示 B树索引中值的独特性程度越高，它就越有效。在数据库术语中，与表的总行数相比，有很多可选择的（很独特的）列值，这种情况被称为具有高基数。相反，低基数是指与表的总行数相比，独特值很少的情况。

我们还应该指出另外一个有趣的情况。假设不是查询CUST表的所有列值，而是只查询LAST_NAME列，会怎么样呢？

```
select last_name
from cust
where last_name = 'STARK';
```

在这种情况下，因为该索引包含SELECT子句中的所有列值，Oracle只需访问索引就能够满足查询的结果，而不必读取表结构本身。当SELECT子句中的列都具有索引时，这种索引称为覆盖索引。这些索引特别有效，因为只需要读取索引块。

在继续学习之前，先回顾一下本章到目前为止所介绍的内容。

- 索引是一种可选对象，它是在一个表的一个或多个列上定义的。
- 索引要消耗资源。
- B树索引是Oracle默认的索引类型。
- 在表中值最独特的列上创建B树索引效率最高。
- 创建合适的索引能提高性能。
- 在某些情况下，查询优化器会选择不使用索引。换言之，查询优化器这时计算出全表扫描的成本低于使用索引时的成本。
- 在某些情况下，Oracle只需访问索引就可以检索出要查询的数据，而无需对表进行访问。

了解这些有关索引的基础知识，为理解本章和本书其余部分将介绍的概念提供了良好的基础。现在，让我们把注意力转到确定使用哪种类型的索引上。

1.2　确定使用哪种类型的索引

Oracle提供了丰富的索引类型和功能。正确地使用索引可以产生良好的性能和可伸缩的数据库应用程序。相反，如果不正确或不明智地实现某一个功能，有可能对性能造成损害。表1-1总结了Oracle的各种索引类型。乍一看，这是个长长的清单，在Oracle的初学者看来可能有点难以招架。实际上，作出使用哪个索引类型的决定可能不像最初看上去那么困难。对于大多数应用程序，只需要使用默认的B树索引类型即可。

注意　有几个在表1-1中列出的索引类型，其实只是基本的B树索引的变种。例如，反向键索引，仅仅是一种当索引值是顺序生成并插入类似值时，为了均匀地分散I/O而进行了优化的B树索引。

表1-1　Oracle索引的类型和功能说明

索引类型	用　途
B树	默认的索引类型，平衡树索引，适用于高基数（不同值的程度高）的列。除非有特殊原因需要使用不同的索引类型或功能，否则用正常的B树索引即可
索引组织表	当主键包含大多数的列值时很有效率。访问这种索引就像访问表一样。数据存储在一个类似B树的结构中
唯一索引	B树索引的一种形式，用于强制执行列值的唯一性。经常与主键和唯一键约束一起使用，但也可以独立于约束而创建
反向键索引	B树索引的一种形式，在索引有许多顺序插入的情况下，用于平衡I/O
键压缩索引	适用于前导列经常重复的组合索引，压缩叶块条目。此功能适用于B树索引或IOT（索引组织表）索引
降序索引	B树索引的一种形式，在索引对应的列值按降序（默认的顺序是升序）排序时使用。反向键索引不能指定降序，如果是位图索引，那么Oracle忽略降序
位图索引	对于包含低基数列以及在SQL语句的WHERE子句中使用许多AND或OR运算符的数据仓库环境，非常适合使用这种索引。位图索引不适合经常更新行的在线事务处理（OLTP）数据库。无法创建唯一的位图索引
位图连接索引	在数据仓库环境中，对于利用连接事实表和维表的星型模式结构的查询非常有用
基于函数的索引	适用于应用了SQL函数的列。可与B树索引类型或位图索引类型结合使用
虚拟列索引	在表的虚拟列上定义的索引，适用于应用了SQL函数的列，可用来替代基于函数的索引
虚拟索引	允许通过CREATE INDEX的NOSEGMENT子句创建没有物理段或区的索引，在调优SQL时有用，因无需建立物理索引从而避免消耗资源。任何类型的索引都可以创建为虚拟的
不可见索引	该索引对查询优化器是不可见的。然而，在表中的数据被修改的同时也维护索引结构。用于在使索引对应用程序可见之前测试它。任何类型的索引都可以创建为不可见的
全局分区索引	跨分区表的所有分区或常规表的全局索引。它的类型可以是B树索引，而不能是位图索引
本地分区索引	本地索引基于分区表的单个分区。它的类型可以是B树索引或位图索引
域索引	用于具体的应用程序或程序模块
B树聚簇索引	用于聚簇表
散列聚簇索引	用于散列聚簇

以下几节简要介绍B树索引和另外几种索引。必要时，我们会指出本书接下来哪一章中将全面讨论某种索引。

1.2.1 B树索引

首先说明一下，B树索引将在第2章完整介绍。本节介绍它们是为了与其他类型的索引进行比较。如前所述，Oracle默认的索引类型是B树索引。对于高基数的列值，该索引类型是非常有效的。对于大多数应用程序，该索引类型是合适的。

不指定任何选项的情况下，CREATE INDEX语句创建B树索引，需要提供索引的名称、表名和列名。

```
create index cust_idx2
on cust(first_name);
```

除非有已核实的性能方面的原因需要使用其他的索引类型，否则应当使用B树索引。DBA或开发人员读到一种新的索引功能，就以为供应商的夸张宣传的功能符合应用程序的实际实现的利益，这种情况比比皆是。在你选择实现一种新的索引类型或功能时，请确认是否有充足的理由。

B树索引有多个子类型：

- 索引组织表；
- 唯一索引；
- 反向键索引；
- 键压缩索引；
- 降序索引。

接下来的几个小节将简要介绍这些B树索引的子类型。

1. 索引组织表

索引组织表（IOT）在一个B树索引结构中存储表行的全部内容。使用索引组织表，能缩短具有精确匹配和主键范围搜索的查询时间。

尽管索引组织表是作为B树索引结构实现的，但它还是通过CREATE TABLE...ORGANIZATION INDEX语句创建的。例如，

```
create table prod_sku
(prod_sku_id number
,sku         varchar2(256),
constraint prod_sku_pk primary key(prod_sku_id, sku)
) organization index;
```

注意 关于索引组织表的实现细节，请参阅第4章。

---

2. 唯一索引

当创建B树索引时，可以定义它是唯一索引。在这方面，它就像唯一键约束。当插入数据到相应的表时，唯一索引将保证插入到表中的非空值都是不同的。出于这个原因，唯一索引通常与

主键和唯一键约束联合使用（完整的详细信息，请参阅第2章）。

通过CREATE UNIQUE INDEX语句指定创建唯一索引。

```
create unique index cust_uidx1
on cust(last_name, first_name);
```

注意　请参阅第2章，全面了解创建唯一索引与允许Oracle在定义主键或唯一键约束时，自动创建索引这两种方法的优点和缺点。

3. 反向键索引

反向键索引对于平衡有大量顺序插入的索引的I/O是非常有用的。在需要一种方式均匀地分布索引数据，以避免将相似的值聚集在一起时，这些索引表现更好。因此，当插入大量顺序值时，如果使用反向键索引，就可以避免I/O集中在索引内的某个物理磁盘位置。这种索引类型将在第5章进一步讨论。

反向键索引通过REVERSE子句指定，如下所示：

```
create index cust_ridx1
on cust(cust_id) reverse;
```

注意　不能对位图索引或索引组织表指定REVERSE子句。另外，反向键索引不能是降序类型的。

4. 键压缩索引

键压缩索引有助于减少前导列经常重复的组合索引的存储和I/O要求。使用compress N子句创建压缩的索引。

```
create index cust_cidx_1
on cust(last_name, first_name) compress 2;
```

注意　不能在位图索引上创建键压缩索引。

5. 降序索引

默认情况下，Oracle用升序方式存储B树索引。例如，如果在一个列值为数值型数据的列上创建索引，最小的数值将首先出现在索引（最左边的叶节点）中，而最大的数值将被储存在最右边的叶节点上。

通过对一列指定DESC关键字可以指示Oracle反转这种顺序为降序。这将创建降序索引。例如，

```
create index cust_didx1
on cust(cust_id desc);
```

降序索引对于某些列以升序排序而另一些列以降序排序的查询是有用的。