现代数据库索引设计与优化

薛佳楣◎著

国家一级出版社 中国纺织出版社 全国百佳图书出版单位

现代数据库索引设计与优化

薛佳楣 著

中国纺织出版社

图书在版编目（CIP）数据

现代数据库索引设计与优化 / 薛佳楣著. -- 北京：中国纺织出版社，2019.1

ISBN 978-7-5180-5052-9

Ⅰ. ①现… Ⅱ. ①薛… Ⅲ. ①数据库系统—研究 Ⅳ. ①TP311.13

中国版本图书馆CIP数据核字（2018）第112068号

责任编辑：姚　君　　责任印制：储志伟

中国纺织出版社出版发行

地址：北京市朝阳区百子湾东里A407号楼　邮政编码：100124

销售电话：010-67004422　传真：010-87155801

http://www.c-textilep.com

E-mail：faxing@c-textilep.com

中国纺织出版社天猫旗舰店

官方微博 http://weibo.com/2119887771

北京虎彩文化传播有限公司印制　各地新华书店经销

2019年1月第1版第1次印刷

开本：880×1230　1/32　印张：5.5

字数：160千字　定价：51.00元

前　言

数据库设立的一个主要目的就在于实现高效管理。但是伴随着信息化程度的不断加深，网络技术的不断发展，数字信息呈现出爆炸式、几何级数增加的趋势，在这种形势下，数字信息的存储总量越来越大，这给数据存储及管理技术带来了新的挑战。相应地，存放在数据库中的数据形式也从简单的字符串格式升级成为复杂多样的格式，处理数据的方式也由简单的字符处理升级为字符图像处理。这给人们处理与获取信息提出了挑战，因此，如何提高数据库数据的提取速度，使用户以最快的速度从海量数据库中获取所需要的数据成为数据库设计者的主要目的。

索引就是加快检索表中数据的方法，即能帮助找到满足条件的记录ID的辅助数据结构，索引是与表或视图关联的磁盘上结构，可以加快从表或视图中检索行的速度，是提高数据库系统执行效率的一种有效工具，索引选择问题是数据库物理设计中一个重要的优化问题。数据库索引是数据库管理系统中一个排序的数据结构，以协助快速查询、更新数据库表中数据。数据库索引就是为了提高表的搜索效率而对某些字段中的值建立的目录。数据库索引好比是一本书前面的目录，能加快数据库的查询速度，但是索引比字典目录更为复杂，因为数据库必须处理插入、删除和更新等操作，这些操作将导致索引发生变化。

大型的企业或部门每天都需要处理大量的数据业务，随着企业的发展与壮大，其业务量与数据量的必将会逐年增大，这将导致信息系

统的压力越来越大。此时，在系统设计、开发和测试中没有考虑到的隐患也逐渐暴露，使得系统可靠性与高效性明显降低，影响了对业务支持的质量。在系统设计和开发阶段，设计者最注重的是系统的功能实现，测试阶段注重系统功能的正确性，而对于系统性能缺乏关注，主要是由于数据量较少，无法显示出系统性能瓶颈。当系统上线运行后，随着数据量和并发用户的增多，数据库性能问题逐渐显现，尤其是索引对象设计不当、业务流程变更等原因，导致性能降低是最常见的现象。随着大数据时代的来临以及各个学科邻域信息系统的逐步成熟，传统数据库索引技术明显已无法满足时代的需求，因而，对数据库索引进行优化成了当前亟待解决的问题。

基于此种形势，作者写作了此书，本书首先对数据库物理结构及数据库索引进行了介绍，而后分别阐述了空间数据库索引设计与优化、内存数据库索引设计与优化、图像数据库索引设计与优化、嵌入式数据库索引设计与优化及实时数据库索引设计与优化，希望能为数据库索引技术的研究者提供借鉴。

在本书的写作过程中，作者花费了大量时间，翻阅了大量资料，并且就有些问题咨询了相关的专家，以求提高本书的价值。但是，由于作者能力有限，本书可能还存在许多不足之处，希望广大读者批评指正。最后，诚挚地感谢在本书的写作过程中给予作者帮助的广大亲友！

编者

2018.8

目　录

第一章　数据库索引概述

数据库设立的一个主要目的就在于实现对之进行高效的管理。伴随着信息化程度的不断加深，因特网技术的不断发展，数字信息呈现出爆炸式、几何级数增加的趋势，数字信息的存储总量越来越大，这给数据存储及管理技术带来了新的挑战。同时，数据的格式和种类也在不断增加，而且数据的类型也由简单的字符处理向字符及图像处理的方向发展。面对着这样的形势，如何准确高效地从海量的信息中查询到想要的数据，已成为数据库设计人员的首要任务。

第一节　数据库物理结构

一、数据库结构

（一）基本结构

数据库的基本结构分三个层次，反映了观察数据库的三种不同角度。

1. 物理数据层

它是数据库的最内层，是物理存储设备上实际存储的数据的集合。这些数据是原始数据，是用户加工的对象，由内部模式描述的指令操作处理的位串、字符和字组成。

2. 概念数据层

它是数据库的中间一层，是数据库的整体逻辑表示。指出了每个数据的逻辑定义及数据间的逻辑联系，是存储记录的集合。它涉及的是数据库所有对象的逻辑关系，而不是它们的物理情况，是数据库管理员概念下的数据库。

3. 逻辑数据层

它是用户所看到和使用的数据库，表示了一个或一些特定用户使用的数据集合，即逻辑记录的集合。数据库不同层次之间的联系是通过映射进行转换的。

（二）物理结构

在物理层面上，SQL Server数据库由数据文件组成，而这些数据文件可以组成文件组，然后存储在磁盘上。每个文件包含许多区，每个区的大小为64K，由8个物理上连续的页组成（一个页8K），SQL Server数据库中数据存储的基本单位为页。为数据库中的数据文件（.mdf或.ndf）分配的磁盘空间可以从逻辑上划分成页（从0到n连续编号）。页中存储的类型有：数据、索引和溢出。

1. 文件和文件组

在SQL Server中，通过文件组这个逻辑对象对存放数据的文件进行管理。在顶层是数据库，由于数据库是由一个或多个文件组组成，而文件组是由一个或多个文件组成的逻辑组，这样可以把文件组分散到不同的磁盘中，使用户数据尽可能跨越多个设备，多个I/O运转，避免I/O竞争，从而均衡I/O负载，克服访问瓶颈。

2. 区和页

文件是由区组成的，而区由8个物理上连续的页组成，由于区的大小为64K，所以每当增加一个区文件就增加64K。页中保存的数据类型有：表数据、索引数据、溢出数据、分配映射、页空闲空间、索引分配等。

在数据页上，数据行紧接着页头（标头）按顺序放置；页头包含标识值，如页码或对象数据的对象ID；数据行持有实际的数据；最后，页的末尾是行偏移表，对于页中的每一行，每个行偏移表都包含一个条目，每个条目记录对应行的第一个字节与页头的距离，行偏移表中的条目的顺序与页中行的顺序相反。

二、数据库的结构种类

数据库通常分为层次式数据库、网络式数据库和关系式数据库三种。而不同的数据库是按不同的数据结构来联系和组织的。

（一）数据结构模型

1. 数据结构

所谓数据结构是指数据的组织形式或数据之间的联系。如果用D表示数据，用R表示数据对象之间存在的关系集合，则将DS =（D，R）称为数据结构。例如，设有一个电话号码簿，它记录了n个人的名字和相应的电话号码。为了方便地查找某人的电话号码，将人名和号码按字典顺序排列，并在名字的后面跟随着对应的电话号码。这样，若要查找某人的电话号码（假定他的名字的第一个字母是Y），那么只须查找以Y开头的那些名字就可以了。该例中，数据的集合D就是人名和电话号码，它们之间的联系R就是按字典顺序的排列，其相应的数据结构就是DS =（D，R），即一个数组。

2. 数据结构种类

数据结构又分为数据的逻辑结构和数据的物理结构。数据的逻辑结构是从逻辑的角度（即数据间的联系和组织方式）来观察数据、分析数据，与数据的存储位置无关。数据的物理结构是指数据在计算机中存放的结构，即数据的逻辑结构在计算机中的实现形式，所以物理结构也被称为存储结构。这里只研究数据的逻辑结构，并将反映和实现数据联系的方法称为数据模型。比较流行的数据模型有三种，即按图论理论建立的层次结构模型和网状结构模型以及按关系理论建立的关系结构模型。

（二）层次、网状和关系数据库系统

1. 层次结构模型

层次结构模型实质上是一种有根结点的定向有序树（在数学中“树”被定义为一个无回的连通图）。这个组织结构图像一棵树，校部就是树根（称为根结点），各系、专业、教师、学生等为枝点（称为结点），树根与枝点之间的联系称为边，树根与边之比为1 ：N，即树根只有一个，树枝有N个。按照层次模型建立的数据库系统称为

层次模型数据库系统。IMS（Information Management System）是其典型代表。

2. 网状结构模型

按照网状数据结构建立的数据库系统称为网状数据库系统，其典型代表是DBTG（Data Base Task Group）。用数学方法可将网状数据结构转化为层次数据结构。

3. 关系结构模型

关系式数据结构把一些复杂的数据结构归结为简单的二元关系（即二维表格形式）。例如某单位的职工关系就是一个二元关系。由关系数据结构组成的数据库系统被称为关系数据库系统。在关系数据库中，对数据的操作几乎全部建立在一个或多个关系表格上，通过对这些关系表格的分类、合并、连接或选取等运算来实现数据的管理。dBASEII 就是这类数据库管理系统的典型代表。对于一个实际的应用问题（如人事管理问题），有时需要多个关系才能实现。用dBASEII 建立起来的一个关系称为一个数据库（或称数据库文件），而把对应多个关系建立起来的多个数据库称为数据库系统。dBASEII 的另一个重要功能是通过建立命令文件来实现对数据库的使用和管理，对于一个数据库系统相应的命令序列文件，称为该数据库的应用系统。因此，可以概括地说，一个关系称为一个数据库，若干个数据库可以构成一个数据库系统。数据库系统可以派生出各种不同类型的辅助文件和建立它的应用系统。

第二节　数据库索引介绍

一、索引概述

索引（Index）是数据库中的一个独特的结构，提供查询的速度。由于它保存数据库信息，就需要给它分配磁盘空间和维护索引表。创

建索引并不会改变表中的数据，它只是创建了一个新的数据结构指向数据表。在使用字典查字时，首先要知道查询单词起始字母，然后翻到目录页，接着查找单词具体在哪一页，这时目录就是索引表，而目录项就是索引了。当然，索引比字典目录更为复杂，因为数据库必须处理插入、删除和更新等操作，这些操作将导致索引发生变化。

（一）创建索引的优缺点

索引的一个主要目的就是加快检索表中数据的方法，亦即能协助信息搜索者尽快找到符合限制条件的记录ID 的辅助数据结构。从数据搜索实现的角度来看，索引也是另外一类文件/ 记录，它包含着可以指示出相关数据记录的各种记录。其中，每一索引都有一个相对应的搜索码，字符段的任意一个子集都能够形成一个搜索码。这样，索引就相当于所有数据目录项的一个集合，它能为既定的搜索码值的所有数据目录项提供定位所需的各种有效支持。

1. 建立索引的优点

通过建立索引可以极大地提高在数据库中获取所需信息的速度，同时还能提高服务器处理相关搜索请求的效率，从这个方面来看它具有以下优点：

（1）在设计数据库时，通过创建一个唯一的索引，能够在索引和信息之间形成一对一的映射式的对应关系，增加数据的唯一性特点。

（2）能提高数据的搜索及检索速度，符合数据库建立的初衷。

（3）能够加快表与表之间的连接速度，这对于提高数据的参考完整性方面具有重要作用。

（4）在信息检索过程中，若使用分组及排序子句进行时，通过建立索引能有效地减少检索过程中所需的分组及排序时间，提高检索效率。

（5）建立索引之后，在信息查询过程中可以使用优化隐藏器，这对于提高整个信息检索系统的性能具有重要意义。

2. 建立索引的缺点

虽然索引的建立在提高检索效率方面具有诸多积极的作用，但还是存在下列缺点：

（1）在数据库建立过程中，需花费较多的时间去建立并维护索引，特别是随着数据总量的增加，所花费的时间将不断递增。

（2）在数据库中创建的索引需要占用一定的物理存储空间，这其中就包括数据表所占的数据空间以及所创建的每一个索引所占用的物理空间，如果有必要建立起聚簇索引，所占用的空间还将进一步增加。

（3）在对表中的数据进行修改时，例如对其进行增加、删除或者是修改操作时，索引还需要进行动态的维护，这给数据库的维护速度带来了一定的麻烦。

（二）创建索引

使用命令行处理器来创建索引，可输入以下形式的语句：

CREATE INDEX <name> ON <table_name>（<column_name>）

可以创建一个索引，它将允许重复值，即非唯一索引，以便于可以按照非主关键字的列来执行有效搜索。也允许在构成索引的一列或多列中存在的重复值。下列SQL 语句根据EMPLOYEE 表中的LSATNAME 来创建一个非唯一索引，索引名为LNAME 并且按照升序排序：

CREATE INDEX <name> ON EMPLOYEE（LASTNAME ASC）

以下SQL 语句基于电话号码列来创建唯一索引：

CREATE UNIQUE INDEX PHONE MPLOYEE （PHONENO DESC）

唯一索引确保在构成索引的一列或多列中不存在重复值。在更新或插入行的SQL 语句的结尾，实现这个约束。如果一个或多个列组成的集合已经有重复的值，那么就不能在它上面创建这类索引。

关键字ASC 按照列的升序来排列这些索引项，而DESC 则按照列的降序排列。默认时为升序。

（三）创建双向索引

使用CREATE INDEX 语句创建索引时，如果指定ALLOW REVERSE SCANS 参数则创建的索引可以向左或者向右扫描。也就是说，这些索引支持按照在反方向创建和扫描索引时所定义的方向来进行索引。这个SQL 语句如下。

CREATE INDEX iname ON tname（cname DESC）ALLOW REVERSE SCANS

在这种情况下，基于给定列（cname）中的递减值（DESC）形成索引(iname)。尽管列上的索引定义用来按照递减次序扫描，但通过允许反向扫描，也可以按照升序来扫描。实际上没有使用这两个方向上的索引，创建和考虑存取模式时由优化器控制这些索引的使用。

（四）创建集群索引

以下SQL 语句在EMPLOYEE 表的LASTNAME 列上创建一个群集索引，名为INDEX 1：

CREATE INDEX INDEX 1 ON EMPLOYEE （LASTNAME） CLUSTER

为了让语句更加高效，可以通过与ALTER TABLE 语句相关的PCTFREE 参数来使用群集索引，这有利于将新数据插入到正确的页上，从而维护该群集的次序。一般来说，表上的INSERT 操作越多，为维护群集所需要的PCTFREE 值就越大。因为群集索引确定数据在物理页上放置的次序，所以在任何特定的表上只能定义一个群集索引。

另一方面，如果新行的索引关键字值总是新的大关键字值，那么表的群集属性将会尝试把它们放到表的末尾。其他页上有空闲空间对

保持群集没有什么作用。在这种情况下，将表设置为追加方式可能优于使用群集索引，改变表来拥有一个大的PCTFREE 值。可以通过执行如下命令来将表设置为追加方式：ALTER TABLEAPPEND ON。

以上讨论同样适用于增大行大小的UPDATE 操作引起的新的“溢出（overflow）”行。

（五）完全索引访问（index access only）

CREATE INDEX 语句的INCLUDE 子句指定在创建索引时，可以选择包含附加的列，这些附加的列数据将与键存储在一起，但实际上它们不是键自身的一部分，所以不会被排序。在索引中包含附加列的主要原因是为了提高某些查询的性能，这样做DB2有时将不需要访问数据页，因为索引页已经提供了数据值。只能为包含的列定义唯一索引，但在强制执行索引的唯一性时不考虑被包含的列。

假设经常需要获取EMPNO 排序的员工的列表。查询将如下所示：

CREATE EMPNO，EMPNAME FROM EMP ORDER BY EMPNO

下面的语句会创建一个可以提高性能的索引：

CREATE UNIQUE INDEXIEMPNO

ON EMPNO（EMPNO） INCLUDE（EMPNAME）

结果，查询结果所需的所有数据都存在于索引中，不需要检索数据页。那么，为什么不干脆在索引中包含所有的数据呢? 首先，这需要数据库中的更多物理空间，因为本质上数据在索引中是被复制的；其次，只要更新了数据的值，数据的所有副本都需要更新，在数据更新操作频繁的数据库中，这会是一项很大的开销。

（六）索引页合并与分裂

CREATE INDEX 语句的MINPCTUSED 子句指定在索引叶页上最