国家自然科学基金项目
国家 863 计划项目
上海交通大学学术著作出版基金资助

空间数据库索引技术

郭 薇 郭 菁 胡志勇 编著

上 海 交 通 大 学 出 版 社

内 容 简 介

空间数据库索引是近年来的热点研究领域，是一门前沿交叉学科。本书全面介绍了传统数据库、空间数据库及时空数据库相关的基本概念、应用领域、数据存储机制、数据检索操作及相关的数据索引技术结构，重点分析了空间数据库索引技术的特点、要求及相关实现算法。本书条理清晰、叙述严谨、实例丰富，既适合计算机及相关专业的本科生、研究生教学需要，也适合 IT 业的研究人员、技术人员研究开发需要及阅读参考。

前　言

空间数据库的研究始于20世纪70年代的地图制图与遥感图像处理领域，其目的是为了有效地利用卫星遥感资源迅速绘制出各种经济专题地图。由于传统的关系数据库在空间数据的表示、存储、管理、检索上存在许多缺陷，从而形成了空间数据库这一数据库研究领域。随着地理信息系统(Geographic Information System，GIS)、计算机辅助设计与制造(CAD/CAM)、机器人、多媒体系统、数字地球、移动通信及定位服务等应用领域的发展，对空间数据库以及时空数据库的研究越来越受到人们的重视。

空间数据库索引技术是对存储在介质上的数据位置信息的描述，用来提高系统对数据获取的效率。空间数据库索引技术的提出是由两方面因素所决定的：其一是由于计算机的体系结构将存储器分为内存和外存两种，访问这两种存储器一次所花费的时间大约相差十万倍以上，尽管现在有“内存数据库”的说法，但实际应用中，绝大多数数据是存储在外存磁盘上的，如果对磁盘上数据的位置不加以索引和组织，每查询一个数据项就要扫描整个数据文件，这种访问磁盘的代价就会严重影响系统的效率。因此系统的设计者必须将数据在磁盘上的位置加以记录和组织，通过在内存中的一些计算来取代对磁盘漫无目的的访问，才能提高系统的效率。尤其是在与空间数据库相关的应用中，如地理信息系统、定位服务等，由于其涉及的是各种海量的复杂数据，索引对于处理的效率是至关重要的；其二是空间数据库所表现的空间数据多维性使得传统的数据库索引技术(如B-树等)并不适用，因为传统的数据库索引技术所针对的字符、数字等传统数据类型是在一个良序集之中，即都是在一个维度上，集合中任给两个元素，都可以在这个维度上确定，其关系只可能是大于、小于、等于三种。而空间数据库具有多维性，目前不存在从二维或高维空间到一维空间的映射，使得任何两个在高维空间接近的对象在一维排序序列中也相互接近。因此传统的数据库索引技术并不能对空间数据库进行有效的索引，需要研究特殊的能适应多维特性的空间索引方式。

空间数据库索引技术是提高空间数据库查找性能的关键技术，直接影响到空间数据库系统的成败。因此空间索引技术研究一直是空间数据库研究领域中的一个热点，对它的研究可以追溯至传统数据库中多属性数据的索引研究。多属性数据可以看成是多维空间的点，因此多属性数据索引(如kd-树、网格文件等)可以直接用于索引空间中的点状实体。对于其他形体的空间实体，如曲线、多边形、多面体等，则可以将其先映射成更高维空间的点，再采用点状目标的索引技术，或者采用某种方法将其映射成一维目标，再采用传统的索引技术(如B-树等)，这是空间索引技术的第一种主要研究思路：目标映射。由于复杂的空间形体映射成高维空间的点后，目标间的空间关系不再保持，查找效率很低，因此人们提出了不允许索引子空间重叠的索引法。这种方法将索引空间按照某种策略划分成许多子空间，空间目标属于与其相交的子空间。对于非点状目标的索引，这种方法必然导致目标重复存储，除了存储开销较高以外，还会增加插入、删除操作的复杂度，这是空间索引技术的第二种主要研究思路：目标复制，如R^+-树、mkd-树等。如果允许索引子空间重叠，将目标界定在某一索引子空间之内，则目标的重复存储可以避免，但索引子空间的重叠必然会导致多条查找路径，因此如何组织目标使索

此为试读，需要完整PDF请访问：www.ertongbook.com

引空间的重叠最小是这类索引方法的主要目标，这是空间索引的第三种主要的研究思路：目标界定，如 R-树、R*-树等。

随着数字城市、定位服务等概念的提出与应用，对大型空间数据库的性能提出了更高的要求。它不但要求能够有效地检索海量数据，而且要求能够有效地存储及检索随着时间推移，其位置关系在不断变化的移动数据对象。目前的空间索引技术的性能往往随着索引数据量的巨增及索引数据的不断更新而急剧下降，因此研究针对时空数据库中面向移动数据对象的索引技术迫在眉睫，它正越来越多地受到学术界的广泛关注。

本书系统地介绍了数据组织、数据检索与数据索引的相关概念，详细分析了传统数据库、空间数据库及时空数据库相关的索引结构、实现算法及性能分析，并探讨了目前几种商用数据库所采用的空间索引方法及空间数据库索引技术的发展趋势。本书共分 13 章。第 1 章简要阐述了数据对象与数据组织、数据库、空间数据库、数据库索引结构等基本概念；第 2 章详细介绍了数据文件的存储及组织方式；第 3 章介绍了几种常用的数据检索技术；第 4 章主要介绍了关系数据库检索的相关概念及几种具有代表性的数据库索引技术；第 5 章分析了空间数据库及空间检索的特点，讨论了空间索引技术的需求及分类。第 6 章至第 11 章，重点讨论了几种典型的空间索引结构、算法及性能分析，包括基于二叉树的空间索引、基于四叉树的空间索引、基于 B-树的空间索引、基于动态哈希的网格索引、基于空间目标排序的索引及基于 QR-树的空间索引；第 12 章，重点介绍了时空数据库索引技术的相关概念及几种具有代表性的时空索引技术；第 13 章，简单介绍了空间索引技术在几种商用数据库中的应用及研究发展动向。

由于我们水平有限，再加上空间数据索引技术还处在不断发展和完善阶段，书中错误在所难免，希望相关专家及读者给与批评指正。

本书研究成果得到了国家自然科学基金、国家 863 项目及上海交通大学学术出版基金的大力资助，在此一并表示感谢。

目　　录

第 1 章　概述 …… 1

1.1　**数据对象与数据组织** …… 1
1.2　**数据库管理系统** …… 3
1.3　**数据库索引技术** …… 5
1.4　**空间数据库** …… 6
1.5　**空间数据库索引技术** …… 8
1.6　**时空数据库索引技术** …… 9
1.7　**本章小结** …… 10

第 2 章　数据存储 …… 12

2.1　**数据文件** …… 12
2.2　**存储介质** …… 13
2.2.1　**主存储器** …… 14
2.2.2　**高速缓冲存储器** …… 15
2.2.3　**外部存储器** …… 17
2.3　**文件组织** …… 22
2.3.1　**顺序文件** …… 22
2.3.2　**索引文件** …… 23
2.3.3　**散列文件** …… 26
2.4　**本章小结** …… 28

第 3 章　数据检索及索引结构 …… 29

3.1　**数据检索** …… 29
3.2　**静态检索** …… 30
3.2.1　**顺序检索** …… 30
3.2.2　**折半检索** …… 32
3.2.3　**索引顺序检索** …… 34
3.3　**动态检索** …… 35
3.3.1　**二叉检索树** …… 35
3.4　**基于哈希的检索** …… 40
3.4.1　Hash **表的基本概念** …… 41
3.4.2　**几种常用的** Hash **表** …… 42
3.5　**本章小结** …… 47

第 4 章 数据库索引技术 …… 48

4.1 DBMS 中的数据检索 …… 48
4.2 基于树的索引技术 …… 49
4.2.1 索引顺序存取方法 …… 49
4.2.2 B-树 …… 51
4.2.3 B^+-树 …… 53
4.3 基于哈希的索引技术 …… 57
4.3.1 静态哈希 …… 57
4.3.2 可扩展哈希 …… 58
4.3.3 线性哈希 …… 60
4.4 不同文件组织的性能比较 …… 62
4.4.1 顺序文件的代价 …… 62
4.4.2 排序文件 …… 63
4.4.3 聚簇文件 …… 64
4.4.4 基于树索引的顺序文件 …… 65
4.4.5 基于哈希索引的顺序文件 …… 65
4.4.6 I/O 代价的比较 …… 66
4.5 本章小结 …… 67

第 5 章 空间数据库索引技术 …… 68

5.1 空间数据组织 …… 68
5.1.1 空间数据特征 …… 68
5.1.2 空间数据模型 …… 70
5.2 空间检索 …… 75
5.2.1 空间查询 …… 75
5.2.2 目标近似 …… 76
5.2.3 基于目标近似的空间检索过程 …… 77
5.3 空间索引 …… 78
5.3.1 空间索引的需求 …… 78
5.3.2 空间数据聚类 …… 79
5.3.3 空间索引技术 …… 81
5.4 本章小结 …… 82

第 6 章 基于二叉树的空间索引 …… 84

6.1 kd-树 …… 84
6.1.1 kd-树的定义 …… 84
6.1.2 kd-树的查找 …… 85
6.1.3 kd-树的插入 …… 85

6.1.4 kd-树的删除 …… 86
6.1.5 分析 …… 88
6.1.6 kd-树的变体 …… 88
6.2 K-D-B-树 …… 89
6.3 hB-树 …… 90
6.4 hB*-树 …… 92
6.4.1 hB*-树的基本特点 …… 92
6.4.2 hB*-树的插入和删除算法 …… 92
6.4.3 分裂的避免 …… 93
6.4.4 DAG 的避免和消除 …… 95
6.4.5 效果分析 …… 98
6.5 本章小结 …… 98

第 7 章 基于四叉树的空间索引 …… 100

7.1 点四叉树 …… 100
7.2 区域四叉树 …… 101
7.2.1 MX 四叉树 …… 101
7.2.2 PR 四叉树 …… 102
7.3 CIF 四叉树 …… 102
7.4 本章小结 …… 103

第 8 章 基于 B-树的空间索引 …… 104

8.1 R-树 …… 104
8.1.1 R-树的定义 …… 104
8.1.2 查找 …… 105
8.1.3 插入 …… 106
8.1.4 删除 …… 108
8.1.5 分析 …… 109
8.2 R*-树 …… 109
8.2.1 插入路径的选择 …… 109
8.2.2 结点的分裂 …… 110
8.2.3 强制重新插入 …… 111
8.3 R+-树 …… 112
8.3.1 R+-树及其特点 …… 112
8.3.2 查找 …… 113
8.3.3 插入 …… 114
8.3.4 删除 …… 115
8.3.5 结点分裂 …… 115
8.3.6 分析 …… 116

此为试读，需要完整PDF请访问：www.ertongbook.com

8.4 本章小结 …… 116

第 9 章 基于动态哈希的格网法 …… 118

9.1 网格文件 …… 118
9.1.1 网格文件及其查找 …… 118
9.1.2 插入 …… 119
9.1.3 删除 …… 120
9.1.4 分析 …… 120
9.2 R-文件 …… 120
9.3 G 树 …… 121
9.3.1 G 树的空间模型 …… 121
9.3.2 G 树上的操作算法 …… 123
9.3.3 G 树的效率讨论 …… 124
9.4 本章小结 …… 124

第 10 章 基于空间目标排序的索引方法 …… 125

10.1 Z-排序 …… 125
10.2 Hilbert 曲线 …… 126
10.3 位置键 …… 127
10.4 本章小结 …… 128

第 11 章 QR-树 …… 129

11.1 QR-树的概念 …… 129
11.1.1 QR-树结点结构 …… 130
11.1.2 QR-树类的设计 …… 131
11.2 查找算法 …… 136
11.2.1 查找算法描述 …… 136
11.2.2 查找算法实现 …… 137
11.3 插入算法 …… 139
11.3.1 插入算法描述 …… 139
11.3.2 插入算法实现 …… 140
11.4 删除算法 …… 141
11.4.1 删除算法描述 …… 141
11.4.2 删除算法实现 …… 142
11.5 本章小结 …… 143

第 12 章 时空数据库索引技术 …… 144

12.1 时空数据库 …… 144
12.1.1 时空数据模型 …… 144

12.1.2 移动数据对象 …… 146
12.1.3 时空数据查询 …… 147
12.2 时空数据库索引技术 …… 147
12.3 基于离散数据表示的索引结构 …… 148
12.3.1 3DR-树索引 …… 148
12.3.2 RT-树索引 …… 149
12.3.3 HR-树(History R-树) …… 150
12.3.4 小结 …… 152
12.4 基于连续数据表示的索引结构 …… 152
12.4.1 TPR 树 …… 152
12.4.2 PMR-Quad tree …… 157
12.4.3 Q+R 树 …… 161
12.4.4 IMORS …… 164
12.5 本章小结 …… 169

第 13 章 空间数据库索引技术的应用与发展 …… 171

13.1 商用数据库空间索引技术 …… 171
13.1.1 Oracle Spatial 的空间数据索引 …… 171
13.1.2 IBM 空间数据刀片(Spatial DataBlade) …… 173
13.1.3 MySQL 空间数据扩展 …… 173
13.1.4 ERSI 空间数据引擎(Spatial Data Engine) …… 174
13.2 空间数据库索引技术的发展 …… 175
13.2.1 高维空间索引技术 …… 175
13.2.2 基于空间关系的索引技术 …… 175
13.2.3 基于 Web 技术的空间索引技术 …… 176
13.2.4 基于空间数据仓库的索引技术 …… 176
13.3 本章小结 …… 177

参考文献 …… 179

第1章 概 述

本章1.1节首先介绍了数据对象及数据组织的概念和方法;1.2节介绍了数据库管理系统的特点及基本的管理操作;1.3节介绍了数据库索引的基本思想及常用方法;1.4节介绍了空间数据库的基本概念及空间数据查询对数据库管理系统的基本要求;1.5节重点分析了传统数据库中索引技术运用于空间数据检索时存在的问题,对空间索引的研究现状及各类索引技术的基本思想进行了粗略地总结;1.6节介绍了时空数据库的研究进展及时空数据索引的基本方法;最后,1.7节给出了本章小结。

1.1 数据对象与数据组织

我们正处在一个信息变革的时代。数据(data)作为信息的载体,它是一切文字、符号、声音、图像等有意义的组合,是用符号记录下来的、可以识别的信息。从一般意义上来说,数据是描述现实世界中各种具体事物或抽象概念的可存储并具有明确意义的信息。因此,它在信息系统中起着至关重要的作用。

但同时我们也应该看到,人们所获得的数据正爆炸性地增长,为了从巨大而复杂的数据集合中获得最大效益,用户必须借助相应的工具以简化数据存储、数据组织和数据检索工作,以实现对海量数据的快速地存取、方便地更新(包括插入、删除和修改)及对存储空间的有效利用。否则,数据将变成负担,以至于获取和管理数据的代价将远远超过从数据发掘出的价值。

数据通常以文件(file)的形式加以组织。文件是由大量性质相同的记录组成的集合,它是数据存储和组织的一种基本方法。文件可以有不同的物理存取方式及数据组织形式,本书第2章我们将详细介绍数据文件的存储结构及数据组织方法。文件组织的目的是为文件在物理存储设备上的存储提供有效方法,以便能提高存储空间的利用率和减少存取记录的时间,从而支持各种数据处理的要求。

计算机的体系结构将存储器分为内存和外存两种,访问这两种存储器一次所花费的时间相差十万倍以上。文件的数据量通常很大,尽管随着存储技术的不断发展,计算机内存的价格在不断地下降,但将文件的全部数据均存储在内存中的做法仍然是不可行的。需要将大量的数据存储在外部存储设备(如磁盘、光盘、磁带等)中,并实现相关数据在内存与外部存储设备之间的来回传输和处理。通常,我们将文件存储在磁盘中,且每一个文件由一个或多个存储页构成。文件及访问方法层通过巧妙地组织数据以支持对所要子集或记录的快速访问。

由于外存的访问速度相对来说很慢,因此,对于一个与数据检索相关的应用来说,数据从外部存储设备到内存之间的传输代价(即I/O代价)就是衡量效率的主要标准。在搜索一个数据文件时,如果对磁盘上数据的位置不加以记录和组织,每查询一个数据项就要扫描整个数据文件,这种访问磁盘的代价就会严重影响系统的效率。因此系统的设计者必须将数据在磁盘上的位置加以记录和组织,通过相应的索引机制来取代对磁盘漫无目的的访问,以提高系统的效率。

数据文件可以采用操作系统文件来进行管理，也可以采用数据库管理系统来进行管理。数据管理的任务主要包括对数据进行分类组织、检索、处理和存储等操作。

操作系统(Operation System，OS)可以管理磁盘空间。通常操作系统把文件抽象成字节序列。OS 管理磁盘上的空间，并把诸如“读文件 f 的第 i 个字节”的请求转换成相应的底层命令：“读 d 磁盘的 c 柱面的 t 磁道的 m 块”。磁盘空间管理器负责管理这些 OS 文件的空间。操作系统的文件系统(file system)提供了从逻辑文件到物理文件的映射与转换；处理方式上，既有文件的批处理，又提供联机实时处理能力。文件系统提供的逻辑文件到物理文件的映射与转换，比人工管理阶段数据的逻辑结构和物理结构均需由程序员来设计要方便得多。有了文件系统，程序员可不用再考虑数据具体的物理存储结构，而直接利用文件系统提供的存取方法来编制应用程序。

然而，基于文件系统的数据管理在大容量数据的处理与存储、多用户并发访问、故障恢复、安全性和完整性等方面存在着一些缺陷，重点表现在：

(1) 大容量数据的处理与存储：当计算机数据管理向其他具有巨数据量(如 GB 级、TB 级或 PB 级)的领域(如证券、银行、航空等)延伸时，大量的操作是检索相关信息。然而，对于大数据量场合，既不能一次将所有数据读入内存进行查询，又不能只在一个数据文件中查询。这种情况下，如何保证查询速度，是使用操作系统的文件系统来进行数据管理需要考虑的重要环节。

(2) 多用户并发访问：随着网络的发展和广泛应用，以及人们对数据共享的要求，使得应用系统面向多个用户同时访问和使用成为必需。多用户的同时使用，即“并发访问”(concurrent access)，可能导致多个用户同时存取同一数据，于是出现数据访问的“冲突”(conflict)。而数据访问的冲突可能导致访问数据的不一致，甚至导致数据集合的被破坏。为保证这种并发访问既能顺利进行，又不致引起冲突，同时让用户还感觉不到是多个用户在同时使用同一个系统，操作系统的文件系统亦需做较大改动。

(3) 故障情况下的数据恢复：应用系统在日常化管理运行中，不可避免地会遇到各种各样的故障，如突然断电、系统死机、程序崩溃、磁盘不能读写等。在故障情况下，可能存在数据未完全写入磁盘的现象，从而出现数据丢失，甚至数据的破坏，这对于视管理数据为生命的各行业，特别是商业、证券、银行、保险、电信等，均是一大灾难。为应对这类灾难，实现故障出现后的数据恢复，简称“ 故障恢复”(crash recovery)，基于文件系统的管理应用系统也必须加以改进，以保证出现故障后能顺利恢复数据，避免数据的丢失。

(4) 安全性：企事业单位的日常管理转用计算机化管理后，用户最担心的问题之一是保存在计算机中数据的“安全性”(security)。数据的安全性主要在于对访问数据用户的授权。没有授权的用户不能访问系统的数据；得到授权的用户只能访问他所能访问的数据。因此，要实现现实管理系统所要达到的安全性要求，需要对基于文件的管理应用系统加大改进，以满足数据安全性的要求。

(5) 数据的完整性：同一数据可能会出现在多个数据结构中，同时对应地出现于多个数据文件中。于是问题就出现了，即如何保证多个数据文件中同一数据的一致性。另外，由于计算机化的管理应用系统是现实业务系统的替代者，其管理的数据也应符合现实业务系统中的各种规章制度的要求。事实上，许多规章制度可以转化为对各种数据的“约束”(constraint)，例如，企业人事制度可能规定在职人员的年龄不得超过 60 岁、一个部门的主管不能在其他部门

此为试读，需要完整PDF请访问：www.ertongbook.com

兼任主管、一个学生的学号不能重复等。应用系统中相关数据的管理,就应遵从这种由制度转化而来的数据约束,从而保证数据的完整。基于文件系统的数据管理应用也需在这方面进行改进,以使其真实、完整地成为现实业务系统的替代者,提高管理的水平和效率。

(6) 数据冗余度大:文件系统中文件基本上对应于某个应用程序,也就是说,数据还是面向应用的。当不同的应用程序所需要的数据有部分相同时,也必须建立各自的文件,而不能共享相同的数据。因此数据冗余度大,浪费存储空间。并且由于相同数据的重复存储、各自管理,给数据的修改和维护带来了困难,容易造成数据的不一致性。

(7) 数据和程序缺乏独立性:文件系统是为某一特定应用服务的,文件的逻辑结构对该应用程序来说是优化的。但是,要想对现有的数据再增加一些新的应用是很困难的。系统不容易扩充,一旦数据的逻辑结构改变,必须修改应用程序、修改文件结构的定义。而应用程序的改变,如应用程序所使用的高级语言的变化等,也将影响文件的数据结构的改变。数据和程序缺乏独立性。因此,文件系统仍然是一个不具有弹性的无结构的数据集合。所谓无结构,是指文件之间是孤立的,不能反映现实世界事物之间的内在联系。

基于文件系统的数据管理在大容量数据的处理与存储、多用户并发访问、故障恢复、安全性和完整性、数据和程序独立性等方面存在的问题,导致了目前成为数据库系统核心的数据库管理系统的产生,它使得采用数据库管理系统来管理数据文件成为目前的主流方式,同时也使数据管理从以加工数据的程序为中心,转向以数据共享的管理为核心。

1.2 数据库管理系统

随着数据库技术的发展,目前绝大部分数据文件均是通过数据库(Data Base,DB)及其数据库管理系统(Data Base Management System,DBMS)进行存储和管理。

数据库是指相互关联的数据集合。它是一组长期存储在计算机内,有组织的、可共享的、具有明确意义的数据集合,用于描述一个或多个相关组织的活动。数据库具有以下几个特点:

(1) 它是具有逻辑关系和确定意义的数据集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的冗余度、较高的数据独立性,可为各种用户共享。

(2) 它是针对明确的应用目标而设计、建立和加载的。

(3) 它表现了现实世界的某些方面。

DBMS 是辅助用户对数据库进行有效管理的一组计算机程序。它是位于用户与操作系统之间的数据管理软件,是一种通用的软件系统。数据库管理系统通常由语言处理、系统运行控制和系统维护三大部分组成,给用户提供了一个软件环境,允许用户快速方便地建立、维护、检索、存取和处理数据库中的信息。

通过把数据存入 DBMS,而不是存入一组操作系统文件,就可以利用 DBMS 的特征,以健壮而有效的方式管理数据。随着数据量和用户数量的增长,采用 DBMS 来管理数据文件就变得势在必行。

使用 DBMS 管理数据有如下优点:

(1) 数据独立性:DBMS 提供数据的抽象视图,从而把应用代码与数据细节分开,以保证应用程序尽可能独立于数据表达和存储细节。

(2) 有效的数据存取和索引机制:DBMS 采用各种复杂技术有效地存储和检索数据,这对

管理存储在外部存储设备上的数据尤为重要。

(3) 数据完整性和安全性:如果数据总是通过 DBMS 存取,则 DBMS 能增强数据完整性约束。例如,在插入一个雇员的工资信息之前,DBMS 可以检查是否超过部门预算。另外,DBMS 也可以通过实行存取控制,确保相关数据对不同用户的可见性是不同的。

(4) 数据管理:当多个用户共享数据时,集中数据管理有很大益处。通过合理地组织数据表达方式,可以尽可能地减少冗余。同时,通过调整数据的存储方式,可以获得有效的检索效率。

(5) 并发存储和故障恢复:DBMS 并发调度数据的存取,所有用户在存取数据时感觉就像只有一个用户在操作一样。此外,DBMS 还保护用户免受系统故障的影响。

(6) 减少应用程序开发时间:DBMS 支持很多重要功能,这些功能对很多存取 DBMS 数据的应用程序来说是通用的。DBMS 与高层数据接口相结合便于应用程序的快速开发。而且,由于很多重要任务不再由应用系统实现,而是由 DBMS 处理,因此,这样开发的应用程序也具有更高的稳定性。

由此可见,在需要对大规模数据进行管理的情况下,DBMS 已经成为了必不可少的工具。数据库及其数据库管理系统是信息时代的成功案例,它们已渐渐渗透到我们日常生活的各个方面。随着越来越多在线数据的产生以及计算机网络技术的飞速发展,数据库也变得越来越重要了。今天,众多领域的发展需求,如商业应用管理、贸易、多媒体数据库、互动视频、流媒体、数字图书馆、人类基因图研究、地球观测系统研究等,正推动着数据库领域不断地向前发展。

在 DBMS 中,基本的数据抽象是一个记录的集合,或一个文件,并且每一个文件由一个或多个页构成。文件及访问方法层巧妙地组织数据以支持对所要子集或记录的快速访问。因此,理解文件的组织方式是有效地使用数据库系统的基础。

数据库管理系统结构的最底层软件称为磁盘空间管理器,它管理磁盘上的空间。抽象地说,磁盘空间管理器支持作为数据单元的页的概念,并且提供分配和回收页及读/写页的命令。通常选择磁盘块大小作为页的大小,并且页面作为磁盘块存储起来。这样在一次磁盘的 I/O 操作中就能够完成一页的读/写。磁盘空间管理器隐藏了底层硬件(和操作系统)的细节,并允许高层软件把数据看成是页的集合。

数据库总是随着记录的插入或删除而增长或缩小。磁盘空间管理器除了需要跟踪哪些页在哪些磁盘块上外,还要跟踪哪些磁盘块正在被使用。虽然开始时在磁盘上顺序分配磁盘块是可能的,但后面的分配和回收通常可能产生“空闲空间”。跟踪磁盘利用情况的一个方法是维护一个空闲块的列表。当磁盘块被回收时,通过请求和使用这些块的高层软件把它们放进空闲列表,以备将来使用,并将指向空闲块列表上的第一个块的指针存储在磁盘的一个已知位置上。跟踪磁盘利用情况的第二个方法是维护一个位图,其中的每一位对应一个磁盘块,它将说明一个块是否在使用。位图也支持磁盘连续区域的快速识别和分配,而这对于链式列表方法是很难完成的。

数据库磁盘空间管理器可以建立在 OS 文件之上,也可以不依赖于 OS 的文件系统,通过扩展 OS 功能来实现自己的磁盘空间管理。

1.3 数据库索引技术

数据索引是在磁盘上组织数据记录的一种数据结构，是对存储在存储介质上的数据位置信息的描述。它用于优化某类数据检索的操作，是提高系统对数据获取效率的一种重要手段。索引技术可以帮助我们以多种方式来访问一个记录集合，并有效地支持各种类型的查询。索引使得我们能够有效地检索满足搜索条件的那些记录。在一个给定的数据记录集合上可以创建多个索引，选择一组好的索引是改善系统性能的最有力工具。

在 DBMS 中，针对 DBMS 中存储的数据的提取和访问称为数据库检索或数据库查询。数据查询和或数据检索是 DBMS 最常用的功能之一。根据其应用领域的不同，数据库的查询可以有多种形式。查询优化和求解的有效性很大程度上取决于数据在物理上是如何存储的。数据库索引技术即是研究数据库文件在物理存储设备上的组织及存储方式，它属于数据库物理设计部分。数据库索引技术可以用于加速很多查询。事实上，为底层关系选择好的索引能加速每一个查询。常用的数据库索引技术包括两大类：基于树的索引技术和基于哈希的索引技术。

常用的基于树型组织的索引技术，包括索引顺序存取方法和多层索引树的索引。多层索引树的索引主要包括 B-树和 B^+-树。

索引顺序存取方法(Indexed Sequential Access Method，ISAM)是一个专为磁盘存取设计的文件组织方式，采用的是静态索引结构。由于磁盘是以盘组、柱面和磁道三级地址存取的设备，所以 ISAM 方法对磁盘上的数据文件建立主索引、柱面和磁道三级索引。ISAM 的存储结构分为三个区：索引页、数据页和溢出页。数据页是按记录的关键字值排序的。为了保持 ISAM 结构中数据页的有序性，在插入数据项时，需要频繁地移动记录，且经过多次增、删记录后，文件的结构可能变得很不合理。此时，大量的记录进入溢出页，将产生很长的溢出链，而数据页又有很多因删除而未利用的空间，从而导致性能变差。因此，要周期地进行调整，即重新排序，复制成一个新的索引顺序文件。

B-树和 B^+-树是为克服 ISAM 中存在问题而提出的。B-树是一种动态调节的平衡多路检索树，它具有较高的随机检索效率，但是遍历 B-树中的所有元素却很不方便。在 B-树中，元素被分布在整个 B-树中的各个结点中，并且，在非叶子结点中出现的元素就不再出现在叶子结点中。因此，在 B-树中，很难用一个顺序链将所有的元素链接在一起。

B^+-树可以看作是 B-树的变形，B^+-树具有以下一些主要特征：首先，其树上的操作(插入、删除)能保持树的平衡。其次，在实现删除算法时，除了根结点外，每一个结点都将保证最小 50%的占有率。再者，因为树是平衡的，所以从根到任意叶子的长度都是相同的。B^+-树在各方面都优于 B-树，因此，B^+-树在数据库索引技术中有着广泛的应用。

在基于树的索引技术中，记录在结构中的相对位置是随机的，它和记录的关键字之间不存在确定的关系，因此. 在结构中查找记录时需进行一系列和关键字的比较。其查找所需要的时间总是与文件中的记录个数有关。查找的效率依赖于查找过程中所进行的比较次数。而在基于哈希的索引技术中，记录的存储位置和它的关键字之间建立有一个确定的对应关系函数 f，使每个关键字和结构中一个唯一的存储位置相对应。因而在查找时，只要根据这个对应关系函数 f 找到给定值 k 的映象 $f(k)$。若结构中存在关键字和 k 相等的记录，则该记录必定在

$f(k)$的存储位置上，由此，不需要进行比较便可直接取得所查记录。在此，我们称这个对应关系函数 f 为哈希(Hash)函数，按这个思想建立的表为 Hash 表。

Hash 表技术的关键之一是要处理好元素的冲突。采用不同的方法处理冲突就可以得到各种不同的 Hash 表。Hash 表具有较好的查找效率，但基于哈希的索引技术不支持范围检索。

在实践中，每一个商用关系型数据库都支持一种或多种索引。IBM DB2 及 Informix，Microsoft SQL Server，Oracle8i 和 Sybase ASE 都支持 B^+-树索引。由于基于哈希的索引技术不支持范围检索。而基于树的索引技术能有效地支持范围检索，并且它的等值检索几乎与基于哈希的索引一样有效。所以，很多商业系统只选择支持基于树的索引。但是，哈希技术在实现关系操作如连接操作中是很有用的。本书第 4 章将详细描述基于树的索引技术和基于哈希的索引技术的相关操作及实现算法。

索引虽然可以提高(甚至大大提高)检索速度，但也并非建立得越多越好。如果索引太多，特别是建立一些不可利用的索引，将增加维护索引结构的代价，最终势必增加系统负担，反而降低系统性能。如何组织索引中的数据项以支持对数据项的有效检索是一个重要的问题。

1.4 空间数据库

空间数据是某个空间框架中对象的位置信息。一般来说，空间数据是指与二维、三维或更高维空间的空间坐标及空间范围相关的数据，常用于表示空间物体的位置、形状、大小和分布特征等诸方面信息。一个空间数据对象占据着空间的一个特定区域，称为空间范围，它是用其位置和边界来刻画的。根据空间数据的几何形体特征，空间数据可以粗略地分为点、线、区域等几种类型的目标，例如：地图上的城市、道路、湖泊等。

空间数据是多维的，它不仅要表达空间目标的属性信息，而且要存储目标的几何信息以及目标间的空间关系。另外，针对于空间数据的空间检索、空间分析等操作往往与空间目标的空间位置与空间关系有关。而且通常还会有时间维。空间数据集的重要问题是如何集成来自多个数据源的数据。传统的数据库系统主要针对一维的属性数据而设计，无法有效地表示、存储、管理和检索多维的空间数据。空间数据库就是针对处理空间数据而提出的。

简单地说，空间数据库系统是一个通用的软件系统，它管理空间数据的存储、检索，确保数据的一致性、完整性、安全性，并提供依据空间数据的位置和范围定位它们的工具。空间数据库是对传统数据库的扩展，它除了具有传统数据库的所有功能之外，还必须提供对空间数据的描述、存储、检索能力。

目前，空间数据库已经被广泛应用于多个应用领域。如地理信息系统(GIS)、计算机辅助设计和制造(CAD/CAM)、多媒体数据库、移动数据库等。

GIS 是管理和分析空间数据的计算机系统，在计算机软硬件支持下，对空间数据按地理坐标和空间位置进行处理，完成数据输入、存储、管理、分析、输出等功能。GIS 可以实现对数据的有效管理，包括空间信息(例如城市、州(省)、国家，街道、高速公路、湖泊、河流和其他地理特征)、属性信息(如城市人口数目、道路的建成年代等)及其相互关系，同时支持包含这些空间信息和非空间信息的应用。通过对多因素信息的综合分析，可以快速地获取满足应用需求的信息，并能以图形、数据、文字等形式表示处理结果。在 GIS 中，空间信息通常被认为就是地图

上的各种信息。GIS系统必须有效地管理这些二维或三维空间上的数据集,直观地描述所有类型的空间查询,如“上海至北京之间的高速公路经过哪些城市”、“哪条是从上海至佳木斯的最短公路”等。

和GIS系统一样,CAD/CAM与医学影像系统都涉及到对空间数据的处理,如设计飞机时,必须存储和处理描述飞机机身等对象的表面信息,如一系列的点数据和区域数据。同时,需要经常性地使用区域查询和空间连接查询,在使用查询操作时,还会经常用到一些相关的空间完整性约束(例如“在轮子和机身之间至少要有1英尺的距离”等)。

多媒体数据库,包含诸如图像、文本和其他众多的时间序列的数据(例如音频),也需要空间数据管理。在医学图像数据库中,不得不存储数字化的二维和三维图像(例如X光图像、指纹、人脸图像等),这些图像数据库应用依赖于基于内容的图像检索(例如,找出与给定图像类似的图像等)。除了这些图像以外,多媒体数据库还可以用来存储视频片断,并且搜索场景变化的片断或者有特定对象的片断。另外,还可以存储信号或者时间序列的数据,并且搜索类似的时间序列;同时,它还可以存储文本文档的集合,然后搜索类似的文档。在多媒体数据库中,找出一个与给定目标类似的对象这样的相似性查询运算是一种常用的操作,解决这样一个相似性查询问题的一个流行办法是首先将多媒体数据映射到称为特征向量的点的集合中。表示多媒体对象特征向量最典型的是高维空间中的点。当把多媒体数据映射到点的集合时,重要的一点是要能够保证两个点之间有距离的度量,该度量用于获取两个相对应的多媒体对象之间的近似性。这样,映射到相邻两点的两幅图像必须比映射到相对远的两点的两个图像更相形。一旦对象被映射到适当的坐标空间,搜索类似的图像、类似的文档或者类似的时间序列就可以用搜索相近点来近似。于是,一个相似查询问题即被转换为找出与表示查询对象的点最邻近的点的问题,即空间数据库中最邻近查询问题。只是与GIS和CAD/CAM相比较,这些数据是高维的(通常为10维或者更高)。

移动数据库同样需要对空间数据进行管理。无线通信和定位服务等应用的发展创建了一大批移动数据库用户。一方面,这些用户只是简单地通过网络来访问数据库,类似于分布式数据库管理系统。另一方面,不论是网络,还是数据和用户,都有一些新的特性:首先,通过无线连接的用户带宽是以太网的1/10左右,是ATM网的1/100左右。因此通信开销比I/O和CPU开销更高。其次,用户的位置通常是变动的,而且移动终端的电池使用时间是有限的。因此,除内容传输开销以及因位置的频繁变动而产生的开销外,真正的通信开销体现在连接时间和电池的使用上。通常将数据生成多个副本,以使从不同位置的访问开销最小化。再者,当一个用户移动的时候,一个事务可能需要从多个数据库服务器中访问数据,此时丢失连接的可能性比传统的网络要大。因此,需要为用户程序开发其他的一致性方法。移动数据库的这些新的特性影响了数据库管理系统中的许多构件,包括查询引擎、事务管理程序和恢复管理程序。

由此可见,空间数据库具有十分广阔的应用市场,随着数字地球、数字城市等概念的提出与应用,空间数据库,特别是大型的空间数据库,必将具有更广阔的应用前景。但同时也可以看到,空间数据库领域还存在很多问题有待于进一步研究和解决。

1.5 空间数据库索引技术

与传统的数据库相比，空间数据库涉及对大量多维空间目标的存储与操作。这些空间目标具有其特殊性：

(1) 空间目标往往具有不规则的几何形状，且目标之间的空间关系复杂(如相交关系、相邻关系、包含关系等)。例如在 GIS 中，线状地物往往是蜿蜒曲折的，面状地物也总是覆盖由很多顶点组成的多边形区域；地理对象之间的关系也不再是大小关系，而是在空间位置上所表现出来的拓扑关系，如远离、重合、相交、包含、相邻等。

(2) 存储空间需求量大。假设一个二维空间的点坐标(x,y)需要 8 个字节表示，则描述地图上一条由 20 个顶点表示的公路需要 160 字节的存储空间。事实上，现实世界中的一个空间目标往往远远不只用 20 个点来描述。而且，现实世界中需描述的空间目标数目往往非常巨大。

(3) 对空间目标的空间操作比传统的选择或连接操作复杂、运算量大。例如求两个多边形是否相交、判断点是否在多边形内等，都需要很多计算时间，这是由空间目标的不规则性所决定的。

(4) 难以定义合理的空间目标的空间次序，无法应用通常的排序技术，如归并排序等。

因此，对空间数据的处理是一项时间和空间开销都很高的操作。为了有效提高对空间数据的处理效率，尤其是针对空间位置的实时查询效率，空间数据库必须利用有效的索引机制。

根据对象描述和索引结构的处理，针对空间数据本身的特点及空间数据的点查询、范围查询、最近邻域查询的特点，国内外学者开发了很多空间索引技术。从空间目标映射方法分，空间索引技术可大致分为两大类：

第一类，基于空间目标排序的索引方法

由于现有的数据库管理系统能高效地支持一维索引，提供对一维数据的快速存取，因此如果多维的目标能被映射成一维的目标，则可以丝毫不改地直接使用一维索引技术。问题的关键在于这一映射函数必须较好地保持多维空间目标间的邻近关系以提供较好的空间查找性能。

基于空间目标排序的索引方法的基本思想是：按照某种策略将索引空间细分为许多格子，并给每一网格分配一编号，然后用这些编号为空间目标获得一具有代表意义的数字。这样，多维的空间目标转换为简单的数字，因此一维索引技术可以利用，常规的数据库系统可以用于管理空间目标的属性与几何信息。用一维的值给多维的空间目标排序的技术很多，如：peano 曲线、位置键、Z-排序等。

第二类，专门的空间索引方法

在系统中加入专门的外部空间数据结构，来提供对空间数据的索引。常用的方法有：

(1) 不允许空间重叠的索引法

这种方法将 k 维的数据空间按某种方法(如二叉树划分、四叉树划分、格网划分等)划分成彼此不相交的子空间，然后对属于这些子空间的目标分别存储在对应的磁盘页或数据桶中。对于复杂的非点状空间目标，这类方法必须采取如下两种技术之一：

(a) 目标复制。目标标识重复存储在与该目标相交的所有子空间。

此为试读，需要完整PDF请访问：www.ertongbook.com