WILEY

数据仓库管理

Data Warehouse Performance

提高数据仓库的性能

减少运作和维护费用

〔美〕 W.H.Inmon Ken Rudin
Christopher K. Buss Ryan Sousa 著
王天佑 等译

電子工業出版社
Publishing House of Electronics Industry
URL:http://www.phei.com.cn

Data Warehouse Performance

数据仓库管理

〔美〕 W. H. Inmon Ken Rudin
Christopher K. Buss Ryan Sousa 著

王天佑 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 提 要

本书介绍了如何设计、建立和管理一个高性能数据仓库环境。

数据仓库技术是近几年来出现并迅速发展的一种技术。数据仓库是面向主题的、集成化的、稳定的、随时间变化的数据集合，用以支持决策管理的一个过程。本书从一个全新的角度出发，详细介绍如何达成一个高性能的数据仓库环境。

本书内容新颖、全面，图文并茂，适合于需要建立数据仓库环境的设计者和开发者，而且也是数据仓库管理员必备的参考指南。

图书在版编目（CIP）数据

数据仓库管理/（美）因曼（W. H. Inmon）等著；王天佑等译. - 北京：电子工业出版社，2000.5

书名原文：Data Warehouse Performance

ISBN 7-5053-5343-8

Ⅰ. 数… Ⅱ. ①因… ②王… Ⅲ. 数据库系统 - 系统管理 Ⅳ. TP311.13

中国版本图书馆CIP数据核字（2000）第08270号

书　　名：**数据仓库管理**

著 作 者：〔美〕W. H. Inmon　Ken Rudin　Christopher K. Buss　Ryan Sousa

译　　者：王天佑　等

责任编辑：陈　宇

印 刷 者：北京天竺颖华印刷厂

装 订 者：三河金马印装有限公司

出版发行：电子工业出版社　URL:http://www.phei.com.cn

北京市海淀区万寿路173信箱　邮编：100036　电话：68279077

北京市海淀区翠微东里甲2号　邮编：100036　电话：68207419

经　　销：各地新华书店

开　　本：787×1092　1/16　印张：15.5　字数：390千字

版　　次：2000年5月第1版　2000年5月第1次印刷

书　　号：ISBN 7-5053-5343-8 / TP·2670

定　　价：25.00元

版权贸易合同登记号　图字：01-1999-0944

序 言

数据仓库在很短的时间内就从理论发展成为了常规学识。早些时候的信息处理，是以有利于收集和存储的方式来存储和组织数据的。这些结构包括：分层目录结构，网状结构和反向目录结构。这些早期的运作结构可以很好地满足收集和存储数据的基本需要，但是却根本不利于数据的访问与分析。其结果是，市场、销售和财务部门无处访问他们所知的公司系统集成中的数据，并以此来支持他们的商务需要。他们一直被运行系统的信息系统（IS）部门拒绝在外。最终，市场、销售和财务部门开始自己着手处理这些事务并建立自己的数据存储，这些早期的数据存储就是现在的数据仓库。

数据仓库开启了公司内部的数据并允许容易地、毫无拘束地访问。与技术人员一样，业务人员同样具有对数据仓库的强烈需求。他们都可以从建立和维护一个数据仓库中获取很多利益。有很多非常好的理由可用来解释为什么建立数据仓库已经成为整个公司中一个持久的固定项目。

数据仓库领域根本不同于传统运作的事务处理领域。在很多方面，两者有着根本性的区别：

开发方法不同。运作领域依赖一种规范的瀑布式方法来进行系统开发。数据仓库领域要求一种迂回式的、螺旋式的方法来进行系统开发，在这里也就是用短时间的快速迭代法来产生结果。数据仓库最吸引人的一个方面就是最终用户不需要长时间等待就能够看到答案。

事务处理完全不同。运作的事务处理运行在一个固定的基础上，通常只需要2秒或3秒，展示一个可预测的访问模式。与此相反，数据仓库的事务处理既可在短期内也可以在长时间内运行，并且它展示的是一个不可预测的访问模式。

运作领域服务于办事员和业务执行人员。数据仓库领域则服务于经理和商务规划人员。

从运作领域中获得的决策是非常短期性的。从数据仓库中得出的决策则是长期性的、战略性的。

运作事务所访问的数据量很少——每次只有几行或几条记录。而数据仓库事务所访问的数据量每次可以是数万的甚至是几百万的记录，甚至更多。

运作处理领域和数据仓库领域有如此多的显著区别。因此如你所想，在建立和管理数据仓库时所利用的概念、技巧和技术与以前所应用的完全不同（在形式上和应用上）。要想获得成功就必须学习整个一套新的规范。本书所讲述的就是有关交付和管理一个高性能数据仓库环境所需要的最重要规范。

当大量数据开始在数据仓库中堆积时，数据仓库环境的性能就开始成为一个问题。在只有少量数据时，可以很容易地控制硬件资源环境，并且还可以让所有用户都获得非常好的查询响应时间。只有在大量数据开始聚积时，性能问题才开始出现。而且一旦大量数据开始增加，对性能的关注就要上升到合适的程度。不久，数据仓库管理员就会发现自己被不满意的用户所困扰。只有在这个时候，数据仓库管理员才开始认认真真地考虑优化性能。

然而，查询响应时间并不是唯一的性能测量尺度。一旦数据仓库管理员控制查询性能，他就会立刻认识到查询响应时间只不过是性能中分解出的多个测量尺度中的一个。除了查询性能之外，还有例如可用性、数据质量、数据流通和吞吐量等因素都必须考虑在数据仓库环境的性能测量中（也许甚至优先于查询响应时间）。

可以说高性能数据仓库环境不仅可以支持商业的直接期望（也就是需要），而且也可以扩展（即增长）来支持发展的商业需要。在定义和发布这些性能目标时必须考虑许多因素。

本书从开始就提到了性能，用运行于一块硬件上的一些特殊设计技巧和技术来构造的一种单一体系结构，它并不是一种高性能的数据仓库（虽然在开始阶段也许足够）。更确切地说高性能数据仓库是这样一种环境：该环境包含了多种体系结构、多种技巧和技术，所有这些都被仔细地融合在一起，从而得到一个能够满足用户群多种分析目的的平衡的解决方案。听起来十分复杂？是的。然而，本书将帮助你解决这些难题并获得这样一个数据仓库环境——它不仅仅可以满足用户群的最初需求，而且还可以扩展来支持他们的发展需求。

本书第1章是导论，介绍了高性能数据仓库环境的设计者、实现者和管理者所面临的一些问题的基础知识。接着，我们将从几个非常重要的观点出发（如图1所示），讨论建立和维护一个高性能数据仓库环境所需要的知识。

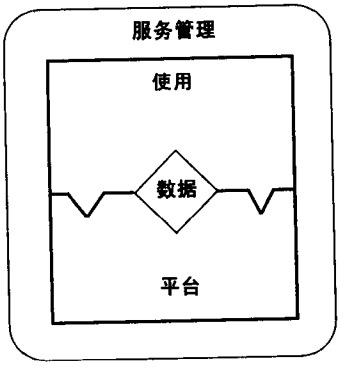


图1 一个高性能数据仓库环境的要素

接下来的章节详细论述了这些观点，并且分成了四个部分。第一部分——使用，数据和性能——重点介绍根据“使用”和“数据”观点如何建立高性能的数据仓库环境，因为“使用”和“数据”层基本上形成了该环境的轮廓。第二部分——平台和性能——重点介绍如何构造“使用”和“数据”所依赖的平台。这是一个很重要的观点，因为如果平台构造不合适，数据仓库环境将不可能扩展以支持商务的发展需求。第三部分——服务管理和性能——重点介绍高性能数据仓库环境的“精华”，即服务管理。数据、使用和平台的观点强调商务能力的交付，然而，服务管理观点的着重点是确保这些能力可以满足客户的发展需要。

最后我们用一个部分来综合整理本书的内容——交付和性能——即把建立一个高性能数据仓库环境所需要的所有这些观点应用于实践。

本书中所提到的一些重要论点包括：

- 用户群——信息使用者和探索者——他们是谁，他们的习惯是什么，以及他们如何与性能相关。

- 休眠数据——它是什么，以及如何管理它。
- 服务管理协议——数据仓库服务管理协议是什么，以及如何使用它来管理性能。
- 硬件结构——MPP、SMP和NUMA——它们是什么，以及它们如何被用作理解性能的基础。
- 数据仓库监视器——数据仓库监视器是什么，如何利用它，以及它有什么局限性。
- 并行性和数据仓储。
- 可扩展性和数据仓库。
- 星型模式和ER建模——在哪里使用它们。
- 数据仓库和数据集市，等等。

本书适合于需要建立数据仓库环境的设计者和开发者。管理这种环境的数据仓库管理员也会从本书中受益。但是，本书对于需要理解底层硬件平台和DBMS（数据库管理系统）软件的系统结构设计师却没有任何用途。最后，信息系统技术专业的学生将会发现这本书非常有用，因为它是从一个全新的角度来讨论性能的。

致　谢

我们要感谢许多同事。在理解如何规划、建立及管理一个高性能数据仓库环境中，我们得到了他们大力的帮助。没有下列这些有才能的人的支持，我们就很难完成这本书的写作。

Bob Barthelow—Hewlett-Packard Co.
Bruce Jenks—Hewlett-Packard Co.
Claudia Imhoff—Intelligent Solutions, Inc.
Corinne jullian—Bristol-Myers Squibb
Debra Colombona—Pine Cone Systems
Dennis McCann—Pine Cone Systems
Dhamodhar Ramanathan—Hewlett-Packard Co.
Doug McBride—Hewlett-Packard Co.
Greg Battas—Tandem Computers
Greg Blair—Bristol-Myers Squibb
Jim Meyerson—Hewlett-Packard Co.
JD Welch—Datawing Consulting
Joe Reese—Cornerstone Concepts, Inc.
Joel Cyr—MCI Communications, Inc.
John Geiger—Intelligent Solutions, Inc.
John Michael Dunn, Jr.—FMG Marketing
John Zachman—Zachman International
Joyce Norris-Montanari—Intelligent Solutions, Inc.
Ken Jacobs—Oracle Corporation
Ken Richardson—Pine Cone Systems
Kevin Waugh—MCI Communications
Lowell Fryman—Intelligent Solutions, Inc.
Lynne Uyehara—Hewlett-Packard Co.
Marc Weinberg—Pacific Bell
Mark Mays—Arrowhead Consulting
Mark Sturdevant—Hewlett-Packard Co.
Martin Haworth—Hewlett-Packard Co.
Pam Munsch—Hewlett-Packard Co.
Pete Simcox—Informix
Roger Geiwitz—Pine Cone Systems

Ron Patterson—MCI Communications, Inc.
Tom Connolly—S3 Consulting
warehouseMCI team—MCI Communications

特别要感谢我们的父母、妻子以及孩子，他们鼓励我们努力工作，并且一直与我们一起分享牺牲和共度难关。最后，我们还要感谢Bill Inmon在数据仓储领域的永恒贡献。他是一位导师和朋友，帮助我们认识到一个人能够写一本书、有一份职业并且还有时间陪伴家人和朋友。

译 者 序

如何有效地管理企业在运营过程中产生的大量数据和信息一直是信息系统工作人员面临的重要问题。70年代出现并被广泛应用的关系型数据库技术为解决这一问题提供了强有力的工具。然而从80年代中期开始，随着市场竞争的加剧，信息系统的用户已经不满足于用计算机仅仅去管理日复一日的事务数据，他们更需要的是信息——支持决策制定过程的信息。这种需求使得在80代中后期出现了数据仓库思想的萌芽，为数据仓库概念的最终提出和发展打下了基础。90年代初期，业界公认的数据仓库概念创始人W.H.Inmon在《建立数据仓库》一书中将数据仓库定义为“数据仓库是面向主题的、集成化的、稳定的、随时间变化的数据集合，用以支持决策管理的一个过程”。

本书是W.H.Inmon等编写的有关数据仓库的另一本著作。数据仓库不同于传统运作的事务处理，它们在开发方法、事务处理方法、事务所访问的数据量及数据存储方式等方面有着根本性的区别。本书从建立高性能数据仓库环境的四个基本要素——使用、数据、平台和服务管理出发，详细介绍了交付和管理一个高性能数据仓库环境所需要的技巧和技术。本书除了序言和第1章之外，共分四个部分。第一部分，包括第2章到第7章，主要论述了使用、数据和性能。这部分的内容包括：用户类型及其作业类型、数据集市、休眠数据、数据清理和监视器等。第二部分，包括第8章到第10章，主要论述建立一个性能平台。这部分讨论了构成平台层的组件，并且我们还详细论述了如何构造、设计、评估和实现这些组件成为一个平衡的平台。最后，我们概述了平台发展的一些高级主题。第三部分，包括第11章和第12章。这部分我们详细论述了服务管理。特别地，我们讨论了服务管理的重要性，服务管理协议的作用，以及实施服务管理协议的步骤。第四部分，包括第13章。这部分我们使用一个实例研究来应用前三个部分所论及的大部分技术和技巧。

本书适合于需要建立数据仓库环境的设计者和开发者，数据仓库管理员也会从本书中受益，尤其是信息系统技术专业的学生将会发现这本书非常有用，因为它是从一个全新的角度来讨论性能的。

本书原文作者W.H.Inmon是Pine Cone Systems（一个编制数据仓库环境管理软件的公司）的首席技术官员，被誉为“数据仓库之父”。Ken Rudin是Emergent公司（一个主要研究大型数据仓库的系统集成公司）的创始人和CEO，也是一流的性能专家。Christopher K. Buss是惠普公司Open Warehouse Advance Technology Center（开放仓库发展技术中心）的高级技术顾问，他参与了惠普公司多个国际性数据仓库工程。Ryan Sousa是一名在数据仓库领域国际公认的专家，他专门研究支持客户关系和市场拓展。

本书翻译人员如下：王天佑、王苍宇、郝青青、周月祥和贲美华。

由于译者水平有限，时间仓促，错误之处在所难免，望广大读者不吝指正。

译 者

2000年元月于北京

目　录

第1章 数据仓库性能导论

起初，成功的数据仓库环境相对较小，工业企业在实践中用迭代方式来建立它，以便数据仓库环境的商业价值得以迅速体现。因而，性能不是问题。只有很少的用户、少量的数据和大量的硬件来运行查询和处理分析。在这些条件下，讨论性能问题毫无意义。但是这些条件将会消失得非常快。数据仓储的成功将会使其中的数据量开始迅速增长。一旦哪里有少量数据，那里不久就会有海量数据。哪个数据仓库环境中有少量用户，那里不久就会涌现大量用户。其结果是，在查询时间、可用性和数据质量等方面最终用户将觉得数据仓库的性能每况愈下。

为理解性能是如何成为一个问题的，让我们用最终用户的眼光来观察这种变化。在某一天，某个最终用户按照规范使用数据仓库。他编写一个查询并输入执行，在这一天，查询操作只用了几秒钟，所以他根本就没有注意数据仓库的性能。但在六个月后，这个用户发现同样的查询操作需要几分钟，他开始注意到性能但没有抱怨，并期望情况将来会变好。不幸的是，随着时间的推移，用户的响应时间变得愈来愈长，再次的六个月之后，同样的查询操作用了一个小时。这个最终用户将会很困惑。他并没有改变查询的内容，甚至连一行SQL（结构化查询语言）也没有改动，然而响应时间却变得越来越长。

事实是这样的：虽然没有改动查询，但是查询操作所依赖的环境已经发生了很大的变化。最终用户指出他对查询操作并没有做任何改变，没错。然而所不同的是，查询操作要面临越来越多的数据并且还要与越来越多的用户抢占资源。就在最终用户从数据仓库环境中获取最大利益的同时，此环境也会让用户失望，操作如此缓慢且不安全将使得用户只能放弃。

测量性能

测量数据仓库环境中查询性能的依据就是从提交查询到返回查询结果的时间。就测量依据而言，数据仓库查询性能的测量与OLTP（联机事务处理）查询性能的测量相同。但是OLTP的查询性能与数据仓库的查询性能有着重要的区别。最显著的区别就是：典型的OLTP查询操作只需2到3秒钟，而数据仓库的查询操作将需要30秒到5分钟，甚至24小时或者更多时间来完成。一个数据仓库查询和一个OLTP查询两者所要执行的功能也完全不同。OLTP查询更具有管理功能的特征。例如，一个服务代理商正在更改一个联机客户的帐目。在这种环境下，生产率和客户满意程度与查询时间之间有着紧密的相关性。与此不同，数据仓库的查询特征就是用户负责制定相对于短期决策来说更为长期的、战略性的决策。对于这种类型的最终用户来说，要作出有效的决策，数据的深度和广度要比1秒或2秒的查询响应时间显得更为重要。

OLTP查询和数据仓库查询的另一个重要区别就是测量响应时间的方法不同。OLTP查询响应时间的测量是从提交时刻到数据第一次返回给最终用户的时刻。与此大不相同的是，数据仓库的查询响应时间不是一个，而是有两个重要的测量：

- 从提交查询的时刻到第一行或第一个记录返回给最终用户的时刻之间的时间长度。
- 从查询的提交直到返回所有行之间的时间长度。

在数据仓库领域，一个查询也许需要许多行数据。为此，响应时间需要测量第一行以及最后一行数据返回的时间。相反，在OLTP领域，每次查询只是访问有限的数据。结果，第一行和最后一行数据返回的时间差异很小。例如，在OLTP领域，你感兴趣的也许是有关一个特定客户或帐户的信息。因而，只返回很少的几行数据。而在数据仓库领域，你可能对比较和对照客户或客户帐目感兴趣。因此，将返回很多很多的行。

生产率和性能

数据仓库的查询响应时间特征不同于OLTP的查询响应时间特征，并不意味数据仓库环境中的性能不重要。事实上，它对于分析员十分重要，只不过程度有所不同。在数据仓库分析员的生产率和数据仓库的性能之间有着一个非常有趣的间接关系。为说明这种关系，假设有两个数据仓库分析员——Ann和Bob。Ann提交查询并在30秒内收到响应。Bob提交查询而在一天之后才收到答复。系统性能对Ann和Bob的工作方式有着很大的影响。因为Ann获得较好的系统性能，所以Ann可以很坦然地构造并测试假设。她有充足的时间来做试验，一个一个地来尝试。如果某个假设被证明不正确，她可以再尝试另一个。Ann没有被她的环境所约束，结果她有非常高的生产效率并且富有创造性。

另一方面，Bob很少有机会来做试验。因为Bob的查询时间已经接近24小时，所以他很少有机会来反复分析。因此Bob必须仔细规划哪些可以和哪些不可以被用来反复分析测试。结果，和Ann相对比，Bob根本没有机会来发挥其创造力和做试验。另外，Bob还有一个约束，就是当Bob收到问题的解答时，他也许已经忘记了询问的原由或者这个查询已经过期了。由于上述原因，Bob的生产率和利用其直觉的能力远远不如Ann，这仅仅是因为两个人所使用系统的性能不同而已。因此在数据仓库环境中，性能和生产率之间有着影响力很大的间接的关系。

数据量和性能

数据仓库环境性能的一个关键问题是管理和调整访问大量数据的能力。图1.1显示了数据仓库环境中产生的大量数据。

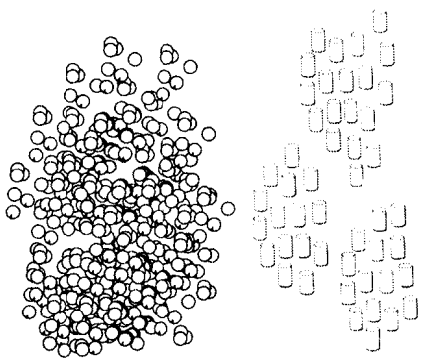


图1.1 数据仓库环境的关键性能就是管理大量数据的能力

数据仓库环境所接纳的数据量是以前的信息处理环境中从来没有经历过的。在以前的环境中，数据量的测量单位是千（KB）和百万（MB）字节。而在数据仓库环境中，数据量的测量单位是千兆（GB）和兆兆（TB）字节。当然，在这些测量单位之间有着相差悬殊的数量级。

为什么数据会增长？

在数据仓库环境中数据量成块增长有很多原因：

数据仓库收集历史数据。以前系统考虑的只是当前的数据。因而，早期的系统只操作限量的数据。但是，数据仓库中收集的是5到10年的数据，所以根本无法避免大块数据的聚集。

数据仓库包含满足未知需求的数据收集。数据仓库的设计者必须同时满足已知需求和未知需求。为此，数据仓库设计者必须将一些无关的和不明显的数据合并到数据仓库中。这种用来满足未知或潜在需求的要求将导致数据存储量增长，其中的一些数据或许会用到，而有些并不会被使用。

数据仓库既包括了详细数据也包含了概括数据。同时容纳详细数据和概括数据的需要也会导致大量数据的累积。

数据仓库还包含外部数据（例如，人口统计学数据、心理学数据等等）。收集大量有效的外部数据可以用来支持多种可预测性的数据挖掘活动。例如，数据挖掘工具利用这种外部数据来预测谁可能是一个好的客户或某些公司在市场上有可能如何运作。

无庸置疑，还有更多的原因可用来说明为什么数据仓库要贮存海量数据。然而，仅是上述原因就要求数据仓库所包含的数据比以前数据处理专家曾经处理的数据要多得多。

数据障碍

为什么数据量在数据仓库环境中对性能有如此大的影响？答案很简单。大量数据会妨碍分析员查找所需要的多行数据。在OLTP环境中，OLTP用户每次查询只期望找到1行或2行数据。只要数据可以用某种最佳的方式来检索或分离，无论有多少数据，OLTP用户都可以很快找到相关数据。但是在数据仓库环境中，用户通常希望能够查看多行数据——某些时候则需要更多行的数据。查找多行数据的要求意味着单独一个简单的索引或一个有效的数据分割模式都不足以得到好的性能。

此外，优化数据仓库环境中的数据结构相当困难，因为有多种多样的需求必须要同时得到满足。某个分析员希望用这种方式来查询数据仓库，另外一个分析员希望用那种方式来查询，而第三个分析员想用的查询方式与前两个又不相同。终在某一天，数据仓库只能用一种方式实现物理优化。在同一个数据库中不可能有多个物理优化方案。利用数据集市作为数据仓库的扩充，这也是其中的原因之一。数据集市允许有数据仓库的定制化窗口来支持多种分析需要（例如，多维分析和数据挖掘）。因而必须非常小心仓库中的数据物理结构，因为数据只能用一种方式来物理优化。

查找隐含数据

大量的数据如何隐藏了分析员所需要的数据？图1.2演示了分析员如何找到一行数据。

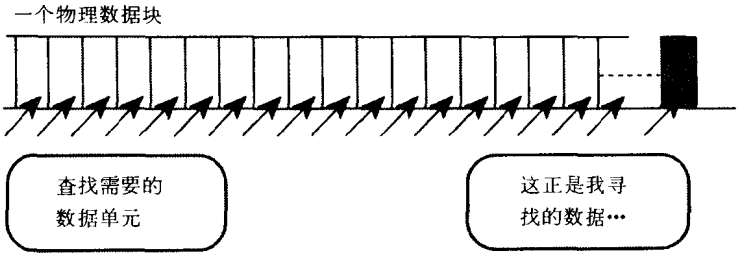


图1.2 为了找到所需要的数据单位，首先必须处理大量不需要的数据

为了找到感兴趣的某行数据，分析员首先必须艰苦地处理物理块中与任务毫不相关的多行数据。即使可以在高速内存中处理这些数据行，浏览那些不需要的行也会花费性能。数据仓库管理员可以利用多种调节技巧——即通过减少检索隐含数据所需要的时间来提高查询性能。这些技巧包括：

- 非规格化
- 建立索引
- 并行性
- 归档

在第3章，我们讨论了这些技巧如何用来支持多种类型的最终用户（信息使用者和探索者）。在第9章，我们讨论了这些技巧被作为数据仓库平台的一部分是如何被配置的。为了更好地熟悉这些技巧，我们先对它们做一个简单介绍。

非规格化 非规格化是一种在关系数据库中引入控制数据冗余的处理。这种冗余的目的就是通过减少频繁的连接、聚集和推导来提高查询性能。这种技巧通过最佳化同一个物理块中的数据物理存放位置、预导出与预计算频繁需求的属性来提高查询性能。利用这些技巧，数据仓库管理员能够最小化查询处理所需要的数据量和I/O（输入/输出）。

但是，在有效利用数据的非规格化之前，有几点重要事项需要考虑。

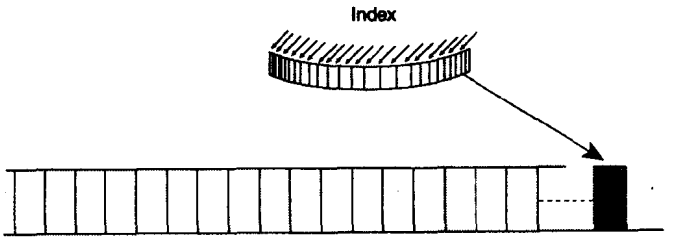
首先要考虑的事项就是数据的优化是仅为一类用户还是多类用户。如果选择了某一类数据的物理优化方案，那么其他类数据的使用就不可能是最优的。这里强烈建议数据的非规格化应该仅用于信息使用者这种类型的数据仓库分析员，因为只有信息使用者才会在处理之前考虑应该如何来使用数据。而探索者并不知道他们该如何处理数据，所以设想用某种方式来物理组织数据以优化探索者所需要的性能，这非常困难。其次要考虑的事项就是非规格化经常使数据变得难以使用和脆弱。通常，数据越不规范，构造数据就越不可思议并更加困难，在管理中也就越复杂、需要更多的努力。使用非规格化必须经过仔细考虑和谨慎利用，因为非规格化技巧将影响数据的结构，并且非规格化的优化处理只能是为某一类用户。

有4种常用的方法可以实现非规格化，在第3章“信息使用者和探索者”和第9章“建立一个高性能数据仓库平台”中将详细讨论。这些方法包括：预聚集，列复制，预连接和阵列。

建立索引 索引技巧就是通过忽略一些无关的数据来搜索所需要的数据。事实上，建立索引可以很好地改善数据仓库性能。图1.3演示了一个索引的创建。

索引可以避免物理地搜索不相关行的数据，如图1.2所示。利用此方式，可以提高性能。但是有一些好的理由完全可以说明为什么建立索引并不能够全面地提高性能。作为数据仓库

环境中性能问题的通用解决方法，为什么建立索引并不能够增强性能，其原因有：



一个物理数据块
当然，建立索引可以避免在物理块中顺序地搜索数据，但是…
- 必须装载索引。
- 索引需要自己的数据管理。
- 索引需要独自的数据量。
- 索引需要维护。
- 不是所有值都能够/应该被索引。
- 如果某个索引非常大，处理这个索引需要相当多的时间。

图1.3　建立一个索引

建立索引需要系统开销和资源。如果数据仓库管理员指定了太多的索引，那么建立和维护这些索引的系统开销将是惊人的，并且这些系统开销对性能的影响也是难以接受的。

索引并不能够解决与性能相关的每个问题。当只有少量值时，用一个索引就可以很容易地搜索整个数据库；在有多变量值的情况时，就必须建立并协调多个索引来检索一个单一的数据逻辑值。

建立索引并不能满足未知需求。索引的本质就是根据已知的需求来组织数据。如果不知道应该如何访问数据，则不可能逻辑化地和最优化地定义一个索引。一个索引预先假定了访问需求是已知的。在数据仓库环境中，许多场合在确实需要数据之前，访问需求一直是未知的。在这种情况下，直接访问数据比停下来建立一个索引来访问数据要快得多。

由于以上原因，很明显索引只能够解决数据仓库环境中部分性能问题。通常，由于信息使用者工作的可选择性和可预见性，索引可以满足信息使用者的需求。但令人遗憾的是，探索者的工作是不可预测的，所以索引很少能够满足探索者的需求。

并行性　管理大量数据的结构方法之一就是利用并行硬件结构。在并行机中，用多个处理器来负担一个处理单元。与一个处理器顺序处理数据集合不同，多个并行处理器同时处理一个数据集合的不同部分。访问数据集合所需要的处理总量是相同的，但是处理所需的时间将随着处理器的数量线性下降。给定数据仓库环境中的数据量之后，问题不在于是否要使用并行处理，而是哪种并行处理形式是最合适的。在第9章“建立一个高性能数据仓库平台”中将讨论以下4种可扩展的硬件结构：

- 对称多处理器（SMP）
- 簇
- 海量并行处理器（MPP）
- 非均匀内存访问（NUMA）