数据仓库 数据库 数据处理 体系结构 (10)

计算机科学 1999Vol. 26№. 2

关于数据仓库若干问题的讨论

Discussion on the Problems of Data Warehouse

39-43,54

吴宏旻 陈 奇 俞瑞钊

(浙江大学人工智能研究所 杭州 310027)

TP274.2

Abstract This paper introduces the development on the modern research of the data warehouse, and some critical problems about the concept, organization, management, design and implementation of data warehouse are discussed here. Our opinions about its development are also put forth in this paper.

Keywords Data warehouse, On-Line transaction processing, On-Line analysis processing

一、决策支持的新手段——从数据库到数据仓库

数据库(DB)是传统的决策支持系统(DSS)中一个重要组成部分。但是,一般决策所需的数据总是与一些维数(每一维代表对数据的一个特定的观察视角,如地区、时间)和不同级别(如部门、领域、地区和国家)的统计或计算有关。此外,随着政府及商业应用的发展,数据量急剧增大,用户的需求也越来越复杂,不仅要能查询或操作数据,还要进行数据分析和信息综合。可以说,以多维数据为核心的多维数据分析是决策的主要内容。

因此,基于传统 DB 的 DSS 已经无法很好地满足需要,出现了许多难以克服的问题[5]:

1)数据缺乏组织性:各种业务数据分散在异构的分布式环境中,各种号称标准 SQL 的关系库实际相互并不兼容,许多微机数据库甚至没有一种标准统一的查询语言;

2)业务数据本身大多以原始的形式存储,难以转化为有用的信息,效率低下;

3)其他问题:DSS 分析需要时间较长,而 OLTP 则要求尽快做出响应。另外,DSS 常常需要通过一段历史时期的数据来分析趋势,而 DB 中一般只存储短期数据,且各个应用领域的保存期限也不一样,在分析时难以满足 DSS 的需要。

人们开始尝试对 DB 中的原始数据进行重新组织、再加工和再利用,形成一个综合的、面向分析的环境,最终提供给高层进行决策,由此,数据仓库(Data Warehouse,DW)的思想逐渐形成。可以说,DW 是由 DB 发展而来的,与传统 DB 目标又有较大的不同。

二、关于数据仓库的含义

对数据仓库这种较新的概念,人们还未形成完全一致的看法。在著名 DB 专家们发表的权威性报告"数据库研究:进入 21 世纪的机遇和成就"[4]中,把 DW 定义为:来自一个或多个数据库的数据的拷贝。

这可能是最广义的一种定义了,它指出了 DW 的最根本的特点,即物理的存放数据,而且这些数据并非最新、专有的,而是来源于其它数据库。至于要存放什么样的数据、如何使用,完全可以根据实际需要而定。

我们认为,这个定义在现在看来仍有其局限性(或者说,应该有更为广义的含义):

- DW 应支持多种数据源:不仅是数据库,还有各种数据文件、文本文件、应用程序等。
- DW 中存放的应该不仅仅是供使用的数据,还有在一定激发条件下能主动起作用的处理规则、算法、甚至是过程等。
- 传统的物理数据仓库方法并非唯一的选择,如 INTERSOLV 公司已经提出以需求驱动建立虚拟数据仓库的解决方案。
- 此外,DW 中的数据并不完全是原始数据的简单归并或搬家,而应该是增值和统一。因此,"汇总并统一"(Consolidation)是一种更可取的描述。

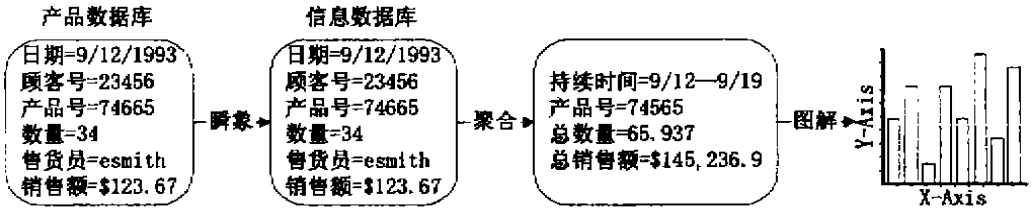
W H. Inmon 是业界公认的 DW 概念的创始人。在他写的《建立数据仓库(Building the Data Warehouse)》[1]一书中，给出的定义是："DW 就是面向主题的、集成的、稳定的、不同时间的数据集合，用以支持经营管理中的决策制订过程。"并作了以下描述："DW 是 90 年代信息技术构架的新焦点，它提供集成化的和历史化的数据；它集成种类不同的应用系统；DW 从发展和历史的角度来组织和存储数据，以供信息化和分析化处理之用"。由于他在 DW 发展中的作用，上述描述在技术性的文献中不断地被引用。

但是，我们认为这个描述性的定义还需要进一步的解释和补充：

·数据的集成化表明数据在结构上具有综合性，并且在语义上是异构的。如前所述，DW 的各个数据源往往是异质的，不同系统对同一数据的意义、用法又存在不同的解释(即语义异构性)，而且其中的原始数据与适合 DSS 分析的数据之间差别很大，因此，数据进入 DW 之前，必须经过加工和集成。这实际上是 DW 建设中最关键、最复杂的一步。

首先，要进行语义"同化"，统一原始数据的所有矛盾之处，如字段的同名异义、异名同义、单位不统一等；然后，要把原始数据结构从面向应用转变到面向主题，组织各种维度；还要对源数据进行必要的变换、净化和综合，以增强其可用性，如分组、获取瞬象数据(Snapshotting)、实施集运算(求和、求平均量等)、填入预测量等等。也就是进行"汇总并统一"。

例如，数据从 POS 系统进入 DW 前，经瞬象抽取某时间区间内的销售记录，然后实施集运算(此处是求销售总量及总销售额)，最后作销售趋势的预测并图形化。对此例而言，进入 DW 的是 3 个不同结构不同形式的数据集，各数据集是相应变换的结果。



·历史化表明它可以截取不同时间尺度上的信息，从瞬态到区段直到全体，DW 以时间为基准来管理(积累、使用并处理)数据，允许用户回顾并了解公司的过去和现在。从数据源中提取数据并加载到 DW 中时，就要在其码键中加入时间项，标明该数据的历史时期。而数据一旦进入 DW，就很少或根本不更新。而且 DW 内的数据时限(一般 5～10 年)要远远长于操作型环境(60～90 天)，这是为了适应 DSS 进行趋势分析的要求。

事实上，任何信息都带有相应的时间标记，但在文件系统或传统的 DB 系统中，时间维的表达和处理一般没有显式化或者是很不自然。例如，DB 的日志文件等等；或是在 DB 的"表"中引入时间字段，但维护和管理困难。

显而易见，需要正确设计出依赖于时间维的数据结构以便于历史信息的处理，而且要有一套办法管理其变化，这是一个必须解决的富有挑战性的课题。

·集成不同的应用系统表明 DW 所解决的问题需要从多个专业应用系统中寻找答案。例如，用户想了解他最可能失去的前 20 个客户是谁，以及用哪种促销方式可以不失去这些客户，DW 就需要从客户服务、销售、订单管理、信用和质量等多个应用系统中提取数据。这也说明了数据源的多样性。

·必须以最终用户最容易理解和使用的方式组织和存储数据。在传统 DB，特别是关系数据库(RDB)基础上建立的各个应用系统，由于功能单一和过于规范化，只能回答很专门的问题。这类应用中，数据的组织方式(结构、索引、编码和访问方式等)只对单一应用是最优的。而 DW 需要为决策提供综合信息，这类信息的组织应当以企业中业务工作的"主题"内容为主线，只有这样才能提供信息的全方位可用性。其中，一个主题基本对应一个宏观的分析领域，例如一个保险公司的主题域可能有：顾客，保险单，保险金，索赔；而按应用来组织则可能是：汽车保险，生命保险，健康保险，伤亡保险。

·需要特别强调的是，DW 保存和管理的是"对象"——数据以及与之相关的处理规则、算法和过程等等的统一体，它们在 DW 中以打包及有序存放的形式被保存和维护，一旦需要即可使用。

因此，我认为当前的关系数据模型的支持能力就显得不足了，DW 所需要的数据模型比这种经典

模型要更为丰富。例如，在DW中对数据作各种复杂变换，并将数据组织成多种多媒体对象，此时就需要一种有更强语义捕获能力和数据组织能力的模型，这种模型应当具有以下的能力和手段：灵活地表达数据的分类层体系；同一对象类中的对象具有可变结构；具有智能性的继承规则；有针对地依赖于时间的历史数据的数据模型；为促进模块化而实现对象的封装；多媒体数据类型，等等。

现在，在DB支持能力及对策上，大致存在三种意见：第一是以当前的RDB技术为基础，按DW的要求作扩展和完善；另一种意见是对现在的DB技术作根本性的改造以适应DW的要求；第三种意见是大力强化前端工具，以弥补支持能力之不足。这些意见各有利弊，再加上种种层出不穷的新主张将使这方面的争论变得空前激烈。

三、数据仓库体系结构中的若干问题

关于DW系统的体系结构，存在着各种不同的说法和建议，但大都包括了W. H. Inmon所区分的3部分在内：①数据源：提供原始数据；②后端加工：实施数据的后处理（包括接收、提取、汇总、变换、打包和储存等）；③前端服务：面向最终用户。

例如，可以按照与传统DB系统相对应的方法划分DW系统：

	DB系统	DW系统
库	DB	DW
管理系统	DBMS：需要频繁地对操作型数据进行更新、删除等，功能较强	DWMS 比DB引擎简单，基本无需更新，但需要从多数据源提取、变换、加载数据
工具	面向OLTP应用，分析功能较弱，只能满足日常应用中信息的提取	不仅需要一般的查询工具，而且需要功能强大的分析工具。

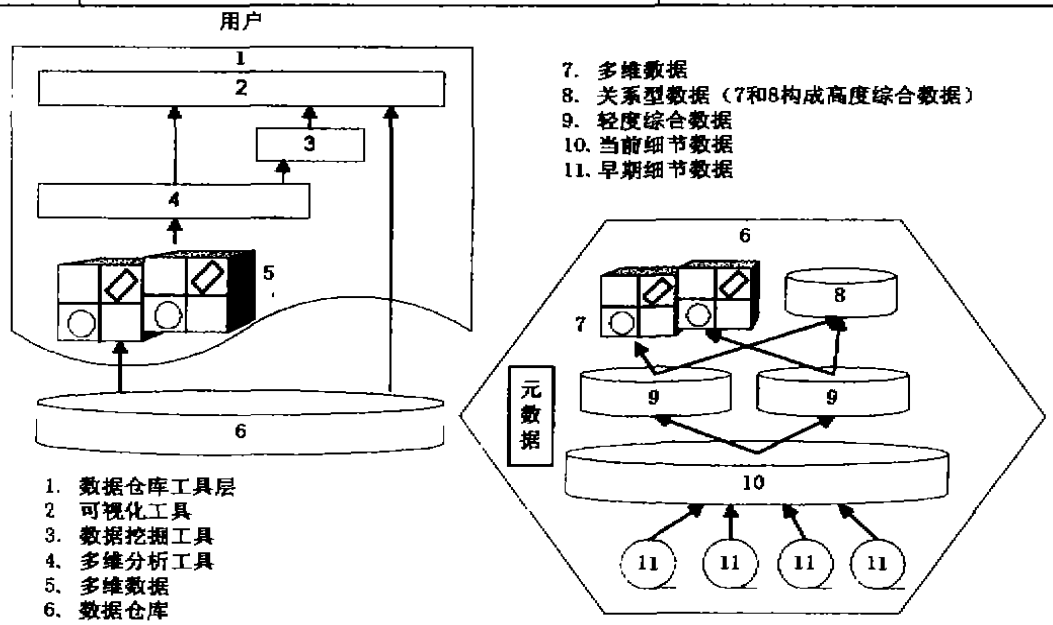


图1 数据仓库体系结构图

1. 关于DW的结构

关于四级结构及其实现问题。DW大体分为四级：早期细节级（远期基本数据）、当前细节级（近期基本数据）、轻度综合级和高度综合级。原始数据（即最近时期的业务数据）经过集成，首先进入当前细节级，这是DB用户最感兴趣的部分，数据量极大。然后系统根据具体需要进一步地综合，并创建索引。随着时间的推移，由DW的时间控制机制将老化的数据转为早期细节级，一般转存在一些转换介质中，如磁带等。

注意，设计轻度综合级的数据结构时，要选取综合处理数据的时间段，明确综合数据包含哪些数据

属性和内容。而高度综合数据层的数据一般十分精练，是一种准决策数据。

当前，细节数据一般用 RDB 系统进行管理，而对于综合性数据有两种方式：①建立专用的多维 DB 系统；②利用现有的 RDB 技术来模拟多维数据。有很多人都认为多维 DB 更适用于建立 DW。

的确，MDDB 在数据存储及综合等方面都有着 RDB 不可比拟的一些优点，实现比 RDB 简明、对开发人员的经验及技术要求不高，维护工作量也比较小。例如用关系表模拟多维数据，其存取就比多维 DB 复杂，首先最终用户的多维分析请求由关系型 OLAP 服务器转为 SQL 请求，然后交由 RDBMS 处理，处理结果经多维处理后返回给用户。而且 SQL 并不能处理所有的分析计算工作，如无法直接处理记录间的计算，只能依靠附加的应用程序来完成。而多维 DB 可以利用多维查询语言或其它方式直接将用户查询转为 MDDB 可以处理的形式、基本不借助附加程序。

但是，由于 RDBMS 在技术上比较成熟，而且它的预综合相当灵活，因此在适应数据的动态变化和适应大数据量及软硬件的能力上优于 MDDB。不过，这种差距是历史造成的，可以想象，MDDB 的技术将会不断成熟，像并行处理等 RDBMS 上用到的技术也会逐渐用到 MDDB 上来。

此外，我们认为，DW 作为一个集成后的核心数据库，应该具有极强的开放性。它可以是分布式的，也可以由多个不同类型的 DB 系统组成，甚至可以由非关系型的、面向对象的以及多维的 DB 等构成。这样既可以把现有的 DB 系统集成到系统中来，又可以针对不同的应用采用最适合的 DB 系统。至于不同 DB 的集成可以采用 DB 指针的方式把各个 DB 连接构成网状结构，从而实现对所有 DB 的透明访问。

·关于元数据和信息目录的作用问题。元数据是关于数据的数据，用于组织和描述整个 DW 的组织结构、如数据结构、从业务环境到 DW 的规划、用于综合的算法等等。而元数据的组织情况反映在信息目录中。它的用途包括：①在辅助 DSS 分析过程中，起定位 DW 的目录作用；②数据从业务环境向 DW 环境传送时，作为 DW 的目录内容；③指导从细节数据到综合数据的综合算法选择。

目录与元数据在 DW 中尤其重要。DW 中所保留的数据时间跨度很大，在此期间，数据结构一般都会有变化，只有元数据能向用户反映这类变化。另外，DW 的最终用户首先是专家和参与决策的人，信息目录是他们了解和使用 DW 的必不可少的工具，对于 DW 系统本身来说，从数据获取、汇总、变换直到形成有用的业务信息的整个过程都需要信息目录的参与。因此可以说，没有信息目录就没有 DW。

理想的情况应当是先有一个正确设计的信息目录，然后据此建立 DW，但信息目录的设计属于概念设计的范围，要得到一个正确的设计往往旷日持久，因此，当务之急是寻求一种方法学的支持，使得 DW 的设计和信息目录的设计能够并行地展开或者滚动地进行。

2 关于 DWMS 和工具层

·关于数据输入接口。如前所述，数据来源应广泛地包含把数据输入 DW 的一切方法、应用程序等。它应当是开放且跨平台的，既能从相同的 DB 系统中利用复制机自动地搜集数据，又能从不同的 DB 系统、应用系统中获取数据。在数据输入接口从各个异质数据源中提取、转换、综合数据的过程中，应该让最终用户、甚至是管理人员无需考虑不同操作系统、DB 系统的差异，用户无需知道数据源的种类和实际分布情况，看到的只是单一的数据来源。

·关于数据输出（分发）。负责把集中的仓库数据分发到多个分设的 DW 数据库服务器和其他供最终用户使用的 DSS 上。从更广泛的意义上说这应当是一个规范，确定数据从中心 DW 导出时应采用何种格式。当所有基于 DW 的应用系统都采用这个规范时，它们就实现了系统互连。

从现实的角度讲，因为许多现有的应用系统采用不同的 DB 和不同的输入格式，为了和它们接口，数据输出部分应当能够定制和修改输出格式和内容，甚至能给输出文件写文件头，这样只要明确其他应用系统所需的输入文件格式和 DB 结构，就能制定出相应的输出文件，把 DW 中的数据转换到其他系统 DB 中。若有条件，数据输出可以采用一些 DB 提供的连接方法（如数据管道和复制管道），把中心 DW 和其他应用系统 DB 连接起来进行数据传输。

目前，作为数据源的主要是 RDB 的表及文件系统的记录，SQL 和与之配合的其他工具（查询和报表等）能把原始数据以行、列的形式提取到屏幕上来，但这还不是最佳表现形式。桌面信息技术的发展，将使我们拥有更多的工具和手段，按照使用的要求组织数据的表达方式：电子报表、直观图形乃至使用多媒体工具以动画方式展现某种变化过程等。

·关于辅助分析。辅助分析用于高级的业务需

求。在DW的基础之上，应用一些先进的技术和工具如数据挖掘、多维分析技术（切片旋转，钻取）、信息可视化技术、决策支持技术等进行二次开发，以期充分地利用现有DB系统，在信息服务和管理决策上更进一步。

现在市场上已经有许多比较优秀的工具，例如BusinessObject、INTERSOLV的Data Direct Explorer、SAS、SPSS等等。其中BO就是一种DB和DW的前端工具，可以综合多种数据源（包括各种RDB、OLAP服务器、应用软件包和本地文件），是集复杂查询、报表、OLAP和数据挖掘等技术为一体的智能决策支持工具。

3. 现有DW产品的缺陷

目前，许多公司和学术界已经开发或正在开发一些DW产品，如IBM的CDF系统、DEC公司的RDB/VMS、Sybase的Warehouse WORKS等等，但是，这些商用DW产品一般都假定DW和数据源使用相同的数据模型，通常是关系模型，这样就无需变换数据结构，而且采用一种离线的批处理方式进行数据采集——即只有自下而上的加载操作，而缺少自上而下的提取操作。因此，这个问题需要学术界更进一步地研究。

四、数据仓库技术发展中的其他问题

DW技术的发展和应用，是信息产业的一个机遇，同时也是一种挑战。透过各种关于DW技术的研究可以窥见该领域在发展之中的种种问题。除了前文提出的一些局限性以外，还有一些问题涉及到DW的设计和管理等各个方面。

（1）在一个相当长的时期，很多人都把DW看作是一个大型信息DB——只是一个对数据实施存储、归总、变换和打包的地方。事实上，DW除此之外还是一项活动，不仅管理数据还管理对数据的处理。

例如，DW也应当是"软件仓库"，即支持软件的分布，这有时比数据的分布更为重要。DW中的对象如果包含由Excel一类的电子报表嵌入的宏（macro）功能，访问这类对象就等价于取得相应的软件及其服务。同样，作为DW基础设施的存取和拷贝管理，同样亦能应用于软件分布。

在很多情况下，系统关注的重点是数据在操作型系统与仓库系统之间的流动，对这类流动及流动中的处理实施调度、监督和恢复，乃是仓库管理的核心任务。

（2）关于DW与操作型系统的分与合的看法。W. H. Inmon强调DW应当独立于操作型系统来建立，理由是这两类系统的需求完全不相同。也有人认为二者应该合一。例如，使用同一个DB2分系统，同时包含这两种系统中的数据并非不可能，当然，大多数操作型系统都建立在大型主机平台上，两个系统合一在技术上有一些复杂的问题需要解决。我们认为，分与合取决于很多其他因素，如原有的软硬件基础、DW系统的目标以及投资状况等，需要视具体情形慎重论证后再作决定。

（3）关于DW的更新问题和规模问题。DW只维持为只读性还是允许更新，是一个对库规模和使用方便性很有影响的决定。有人主张，从效率和充分利用硬件资源的角度衡量，应当将DW与OLTP类系统分开，即把DW看作是只读性的信息DB，当然也有相反的主张。如果组织成只读性DB，就必须将所有需要提供给各级各类用户的信息实实在在地维持在仓库中，这样将会使数据量变得十分庞大，如果组织成可以更新的信息，为维持一种最低限度的数据量，可只存放最详尽一级的基础数据，这里存在着速度与空间的权衡，其依据是DW的功能模式和使用模式。

从已报导的技术资料看，主张组织成只读性的占多数，对DW的更新只发生在从外部数据源提取数据的时候，且只由批处理程序在夜间进行，由于无需设立和管理锁，所以只读性仓库能大大简化DB的并发控制，改善数据的可用性。唯一存在的问题是要解决好仓库系统与日常生产系统之间的数据衔接和协同：基于仓库数据而进行的业务决策必须组织成针对生产系统的事务。从根本上看，这始终是一个艰难而又必须解决的问题。解决好了，DW将在决策者与企业中的各种运行系统的沟通方面，真正起到一种建设性的作用。

（4）DW的功能范围问题。DW一般是针对全企业而建立，特别是它的信息目录的内容是全局有效的。但在DW的实际建设过程中，往往是先从企业内某个急需的部门或者某个已有技术基础的单位作起。另一方面，对于从一开始就针对整个企业建立的DW来说，也希望它对不同的应用部门能有不同的"子部"，既有针对性又有更好的安全保证。总而言之，是建立一个大统一的、集中式的DW，还是一批针对部门业务的、工作组级的DW？这也是一个需要权衡的问题，很难一概而论，但从降低风险、简化项目管理的复杂性出发，从工作组级的仓库系统作起，

（下转第34页）

大量的时间开销。

结论 从理论和技术两方面来看，KDOODB 系统的主要特点表现在：

(1)系统采用了合理有效的页服务器结构，并采用面向对象设计方法，从而使系统具有良好的开放性结构，便于今后对系统的维护和改进以及与其它应用软件的集成；

(2)参照 ODMG—93 国际标准，系统提供了灵活有效、语义表达能力强的标准化的数据模型；

(3)系统扩充当前国际上流行的面向对象编程语言 C++ 作为数据库应用编程语言，提供定义和操作永久对象的功能，使其与 OODBMS 无缝集成，有效地避免了"阻抗失配"问题；

(4)系统充分利用现代硬件和软件的处理优势，提供了对永久对象的高效存储，支持对永久对象的灵活快捷的操作，尤其是对大对象和复杂对象的快速存取提供了良好的支持；

(5)系统提供了使用方便的非过程化的查询语言，既可以嵌入编程语言使用，又可以作为交互式命令使用，实用有效的查询优化算法和循环查询处理策略为查询语言的实现提供了有力支持；

(6)系统针对页服务器结构和嵌套事务模型的特点，研究并实现了基于页的恢复策略和用于事务标识分配的基于位的事务标识分配策略，从而实现了有效的事务管理，为用户开发新型的数据库应用提供有力支持；

(7)系统为用户提供多种使用方式和友好的用户界面，基于 KDOODB 开发了城市道路路面技术数据管理系统和项目合同管理系统等应用系统，显示出该系统具有广阔的应用前景。

参 考 文 献

1 Cattell R, et al. The Object Database Standard: ODMG—93. Morgan Kaufmann, 1994

2 钟武，胡守仁. OQL 逻辑优化准则. 计算机科学，1998，25(2)

3 王意洁，胡守仁 面向对象数据库中循环查询处理技术的研究. 计算机研究与发展，1998，35(12)

4 Kim, et al. Cyclic Query Processing in Object-Oriented Databases. In: Proc. of 5th Intl. Conf. on Data Engineering. 1989. 564～571

(上接第 43 页)

采取自下而上的方式，仍不失是一种务实和可行的办法。

(5)关于 DW 的刷新与归档问题，每时每刻 DW 都需要保存它的信息长河的一个合适的"区段"以维持其可用性。为此就需要不断地刷新和归档。如何确定刷新与归档的时机是 DW 使用和维护中心必然要遇到的问题，时机取决于数据的可用程度。但是，数据的可用性不是一种可预测的事，只能事后处理。真正构成挑战的是在某些使用场合不允许事后处理，如何以事前处理方式解决数据的刷新和归档，仍是一个需要认真研究和解决的问题。

结束语 目前，数据仓库在各个领域已引起了很大的注意。随着信息量的增加，这项技术将会有更为广阔的前景，并给人类带来不可估量的益处。例如，在市场分析、决策支持、金融预测和医学诊断等各个方面，都可以大大地提高信息处理的效率，更有效地发挥数据库的潜在价值。但正如本文所述，开发一个灵活、可伸缩、高效的数据仓库系统，还需要进一步的研究。

参 考 文 献

1 Inmon W H. Building the Data Warehouse(Second Edition). Wiley Computer Publishing, John Wiley & Sons Inc.

2 Hammergren T. Data Warehousing: Building the Corporate Knowledge Base. Ventana Communications Group Inc.

3 Bischoff J. Achieving Warehouse Success. Database Programming & Design, 1996, 7(7)

4 姚卿达，王珊编译. 数据库研究：面向 21 世纪的机遇与成就. 计算机科学，1996，(4)：1～7

5 王珊，罗立. 从数据库到数据仓库. 计算机世界报，1996-7-15

6 周胜，王珊. 论数据仓库系统中工具的重要性. 计算机世界报，1996-7-15

7 姚卿达，黄晓春，刘向民. 数据仓库和数据采掘应用研究. 计算机科学，1996，23(6)：63～65

8 于戈，等 数据仓库管理中的若干关键技术. 计算机科学，1997，24(2)：31～34

9 李云峰，徐新国 OLAP 及其实现. 计算机世界报，1998-4-6