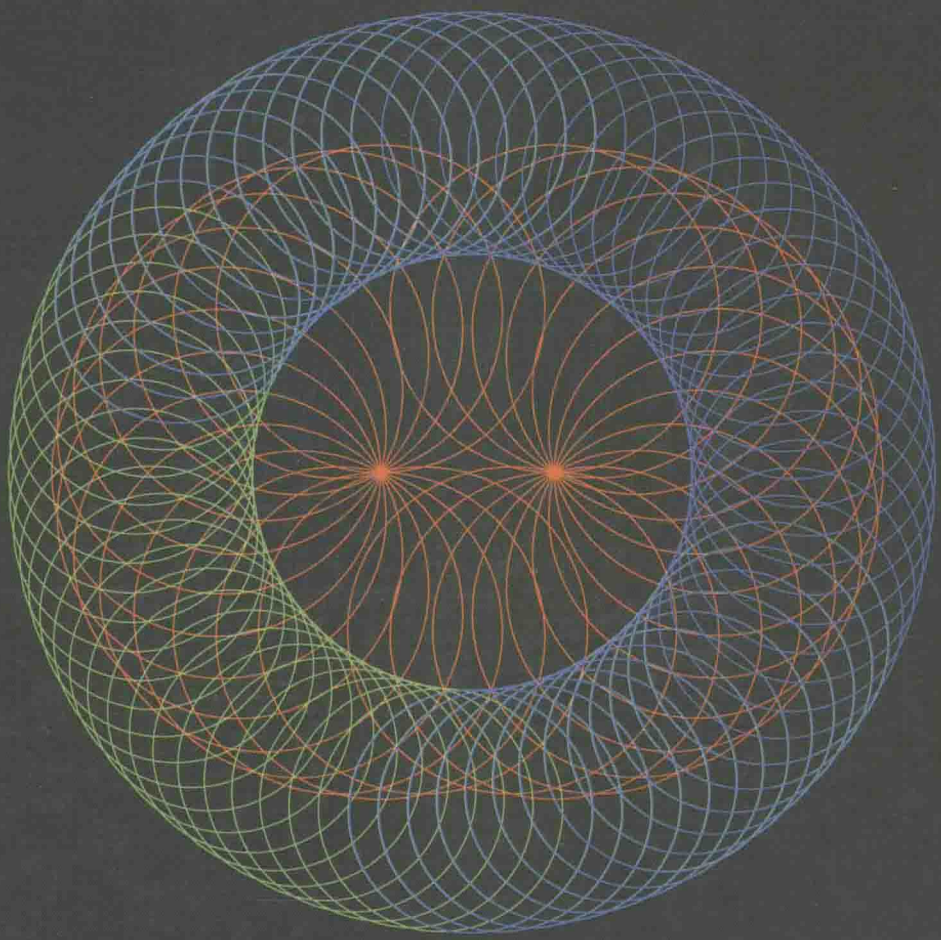
学术专著
Academic Monograph

XML
数据管理技术

王国仁　于 戈　杨晓春　于亚新　著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

XML 数据管理技术

王国仁　于　戈　杨晓春　于亚新　著

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

内 容 简 介

随着 Web 技术的快速发展，如何有效地存储、索引和查询 XML 数据已经成为数据库研究领域的一个热点研究问题。本书的作者在 20 世纪 90 年代末就开始研究 XML 数据管理问题，本书是在他们多年来形成的研究成果的基础上，经过总结和整理而成的。

本书由浅入深，介绍了 XML 数据的聚簇存储技术和外延索引技术；XML 查询处理和优化技术；并行处理技术；基于高级数据模型的 XML 数据更新技术；面向对象的 XML 技术；XML 数据的语义约束；XML 数据的访问控制技术，XML 文档发布中的数据安全定义、基于 XML 约束的数据推演以及信息泄露及其安全技术等。

本书条理清晰，注重理论和实践相结合，对计算机及相关专业研究生或高年级本科生以及从事 XML 数据管理及其相关技术研究开发人员均有帮助。

图书在版编目(CIP)数据

XML 数据管理技术/王国仁等著. —北京：电子工业出版社，2007. 4
ISBN 978-7-121-04014-6

Ⅰ. X… Ⅱ. 王… Ⅲ. 可扩充语言，XML—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 036702 号

责任编辑：竺南直　董亚峰
印　　刷：北京天宇星印刷厂
装　　订：涿州市桃园装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1 092　1/16　印张：19.25　字数：490 千字
印　　次：2007 年 4 月第 1 次印刷
印　　数：4 000 册　定价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系电话：（010）68279077；邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

随着 Web 技术及其应用的快速发展，XML 已经成为万维网上信息表示和数据交换的一个重要的标准，XML 在电子商务、电子数据交换、科学数据表示、数据建模与分析和搜索引擎等领域有着广泛的应用，XML 技术已日益受到更为广泛的关注。到目前为止，在 Web 上已经积累了大量的 XML 文档数据，因此研究 XML 数据的管理技术正成为数据库研究领域中的一个重要挑战，具有十分重要的理论研究意义和实际应用价值。

随着 Web 技术的快速发展，如何有效地存储、索引和查询 XML 数据已经成为数据库研究领域的一个热点研究问题。作者在 20 世纪 90 年代末就开始研究 XML 数据管理问题，对 XML 数据管理所涉及的主要研究问题，包括 XML 数据的存储、索引、聚簇、查询处理、查询优化、XML 文档更新技术、XML 数据的语义约束与访问控制等进行了比较深入和系统的研究工作，在国内外学术期刊和学术会议上发表了一系列有关 XML 数据管理技术方面的学术论文。本书是在作者多年研究成果的基础上，经过总结和整理而成的。

全书共分 9 章，第 1 章和第 2 章介绍所采用的 XML 数据模型和 XML 查询语言，它们是本书研究的基础；第 3 章介绍 XML 数据的聚簇存储技术和外延索引技术；第 4 章描述 XML 查询处理和优化技术，主要包括基于外延的查询处理和优化技术、基于自动机的查询处理技术、基于签名和基于分片的结构连接技术以及基于后缀树的查询处理技术等；第 5 章介绍并行处理技术，主要包括 XML 数据的并行物理分片策略、并行结构化查询处理技术和处理机的并行调度策略等；第 6 章介绍基于高级数据模型的 XML 数据更新技术，主要包括基于高级数据模型的 XML 更新语言的语法和语义，以及实现 XML 数据更新语言 XML-RL 的体系结构等；第 7 章介绍面向对象的 XML 技术，主要包括支持面向对象特征的扩展 DTD、扩展 DTD 的文档有效性验证和基于扩展 DTD 的面向对象的查询语言等；第 8 章介绍 XML 数据的语义约束，主要包括关系数据库参照完整性约束到 XML 文档的映射方法、约束型 XML 文档的实例构造和自动抽取 XML 近似函数依赖等技术；第 9 章介绍 XML 数据的访问控制技术，XML 文档发布中的数据安全定义、基于 XML 约束的数据推演以及信息泄露及其安全技术等。

香港科技大学的陆宏钧教授对东北大学数据库组的发展给予了极为重要和关键的指导，他对我们的指导正是从 XML 开始的，在 20 世纪末将我们引领到 XML 研究领域中来。可以说没有陆老师的指导，就不会有我们今天在 XML 领域中取得的成果。因此，我们谨将此书献给陆宏钧教授，以表达我们对陆老师深深的敬意和感激之情。

本书汇集了作者多年来的研究成果，其中也包括合作者的心血，他们是香港中文大学的 Jeffrey Xu Yu 教授、Kam-Fai Wong 教授，加拿大 Carleton 大学的刘梦赤教授，复旦大学的周傲英教授，武汉大学的彭智勇教授、李石君教授，新加坡国立大学的 Tok Wang Ling 教授和 IBM Almaden 研究中心的 Haifeng Jiang，在此对他们表示衷心感谢。

该书得到国家自然科学基金（60573090，60673139，60573089）、国家重大基础研究计划（2006CB303103）的资助。

参与本书撰写的有：杨晓春（第 1 章、第 8 章），王斌、张恩德（第 2 章、第 9 章），王

波涛（第 3 章），赵相国（第 4 章），于亚新（第 5 章），韩东红（第 6 章）和乔百友（第 7 章）。参加编写工作的还有吕建华、孙冰、肖卫方、周巍、陈杰峰、贾福林、汤南、吴刚、胡军安、周博、王钊、杨川、李建新、张海宁、霍欢、周博以及回晓云。他们为本书付出了辛勤的劳动，在此一并表示衷心的感谢。

本书可作为计算机及相关专业研究生或高年级本科生的教材，也可作为从事 XML 数据管理及其相关技术研究开发人员的参考书籍。

由于作者水平有限，加之时间仓促，难免有疏漏与不当之处，恳请专家、同仁及广大读者批评指正。

王国仁、于戈

2006 年 12 月 20 日

序

自 20 世纪 60 年代末至 20 世纪 90 年代中前期，廿多年间数据库技术经历了一个快速发展的时期，从层次数据库、网状数据库到关系数据库，再到面向对象数据库和对象关系数据库；从集中式数据库到分布式数据库和并行数据库；从商业数据管理技术到各种特种数据管理技术，包括空间数据管理、多媒体数据管理、时态数据管理、文本数据管理等。在这期间，学术界对数据管理技术给予了极大的关注，产生了大量的研究成果，这为信息技术的发展奠定了坚实的基础。

自从 20 世纪 90 年代中期以来，随着应用技术的快速发展，在数据库研究领域涌现出了许多新的研究热点，包括数据流管理技术、传感器网络中的数据管理问题、XML 数据管理技术、P2P 数据管理技术、网格数据管理、生物信息管理等。在这些热点当中，XML 数据管理就是其中的一朵“奇葩”。

随着 Internet 的普及和 Web 技术的快速发展，作为 Internet 上信息表示和数据交换标准的 XML 应运而生。XML 数据已在 Web 上广泛流行，从企业报表、商业广告到技术文档，无所不包。如何有效地管理 XML 数据成了 Web 应用领域中一个亟待解决的重要课题。XML 数据库系统是 Web 技术和数据库技术相结合的产物，引起了数据库学术界的高度关注，产生了很多有意义的研究成果。

数据存储与索引、查询处理与优化、完整性约束与访问控制等技术是数据库领域永恒的研究主题，XML 数据库系统也不例外。该书从 XML 数据特点出发，对 XML 数据的存储、索引、聚簇、查询处理、查询优化、XML 文档更新技术、XML 数据的语义约束与访问控制等方面的技术进行了深入和系统的介绍。本书的作者长期从事数据库理论与技术方面的教学和科研工作，在国内外学术刊物和学术会议上发表了一系列的学术成果，积累了丰富的 XML 数据管理方面的经验。本书是作者在总结和整理他们自己研究成果基础上形成的，具有较高的原创性。

迄今为止，XML 数据管理技术仍然是数据库学术界最热门的研究领域之一，每年在重要的学术会议和学术期刊上都会有大量的学术成果发表。相对而言，有深度地系统地介绍 XML 数据管理技术方面的书籍还是比较少的。我相信这本书会给从事相关研究的学者提供助益，它也是高年级本科生、研究生和从事相关教学工作的教师一本有益的参考书籍。

中国科学院数学与系统科学研究院
周龙骧
2006 年 12 月 29 日

目　录

第 1 章　XML 数据模型

互联网的迅猛发展，尤其是 Web 技术的广泛应用，改变着并冲击着整个世界，带来了经济的发展和生活的丰富，网上飞速流转的信息也越来越庞杂，越来越混乱。互联网的发展需要秩序，XML 注定将成为新互连时代的宠儿。

1.1　XML 的由来

1969 年，IBM 公司由 Goldfarb 领导的研究项目组创建了通用置标语言 GML（General Markup Language），用来编辑和格式化文本，以及在子系统之间实现信息检索。GML 可用于标记任何数据集合的结构，是一种元语言（meta-language）——能够描述其他语言及其语法和词汇表的语言。1986 年 10 月，GML 被国际标准化组织采纳为国际性的数据存储和交换标准，并被更名为“标准通用置标语言”（SGML）[Sun02]。

SGML 是一门非常强大同时也相当复杂的置标语言，它很庞大，功能很强，可选项也很多，适用于需要有严格文档标准的各大组织。在 20 世纪 80 年代，SGML 较多用于科技文献和政府办公文件中。许多国际组织在发布文档时也常常建议用 SGML 的格式。比如世界专利组织（WIPO）就制定了很多的基于 SGML 的规范来记录专利文献资料。SGML 的复杂性使得它十分不适合在 Web 中快速而简便地发布。而且，SGML 文档的处理非常复杂，成本非常高，因此 SGML 的软件通常非常昂贵。Web 发布需要更简单的语言[Sun02]。

1989 年，欧洲粒子物理实验室的 Tim Berners-Lee 提出了适合 HTTP 传输的超文本置标语言（Hyper Text Markup Language，HTML）。Tim Berners-Lee 借鉴了 SGML 的成果，删繁就简。在 Internet 工程任务组（IETF）的支持下，在 1994 年末通用的标记规则被制定为 HTML 2.0，现在 HTML 已经毫无争议地成为了 Web 上的通用语言。HTML 尽管在人机界面方面非常擅长，但是它却不利于机器之间的相互交流和信息的传递。HTML 中的标记大多是设计用来呈现网页布局和外观的，并不能说明所呈现网页的数据含义，它在可扩展性、结构化和语义规范方面存在严重的不足。当 HTML 发展到 4.0 版本以后，人们越来越发现该标准实在到了积重难返的程度，从 1996 年开始，W3C 开始考虑制定新的置标语言——XML，它既具有 SGML 的强大功能和可扩展性，同时又具有 HTML 的简单性[Sun02]。

1996 年 7 月开始，W3C 组织 60 多位精心挑选的 SGML 信息结构专家专门成立了一个 SGML 专家小组，提出了“网络上的 SGML”计划。该小组由 SUN 公司的 Jon Bosak 担任组织者和领导者。该小组删除了 SGML 中非核心的、未被使用的和含义模糊的部分，剩下了短小精干的标记工具——XML。XML 保留了 SGML 80%的功能，而将其复杂程度降低到原来的 20%。1997 年春，可扩展链接语言 XLL 草案已被拟定。最后，W3C 于 1998 年批准了 XML 的 1.0 版本[Sun02]。

由此演变出来的 XML（Extensible Markup Language 的缩写，意为可扩展置标语言），是一套定义语义标记的规则，这些标记将文档分成许多部件，并对这些部件加以标识。它也是元置标语言，即定义了用于定义其他与特定领域有关的、语义的、结构化的置标语言的句法语言。它的出现极大地推动了互联网的发展。

1.1.1 HTML 与 XML

说到这里不得不引出 HTML。HTML（Hypertext Markup Language，超文本置标语言）是现在流行的网页制作语言，它已经成为全球信息网的基础语言。HTML 为人们的传送和接收信息带来了革命性的变化，但是 HTML 主要是被设计为资料显示之用，HTML 的焦点几乎完全放在信息应如何显示上，而不是信息的内容及它的结构。这便是我们需要 XML 出现的原因。

XML 是一个开放的、以文字为基础的卷标语言，它可以给数据提供结构的以及与语义有关的信息。这些关于数据或中继数据（metadata）提供附加的意义和目录给使用那些数据的应用程序，而且也将以网络为基础的信息管理和操作提升到一个新的水平。XML 已经成为 HTML 一个强而有力且标准化的互补，作为在网络上传递信息的另一个重要功能，XML 的目的是同时供网页的创作者和程序设计人员使用。XML 是以文字为基础的，它可以很容易地让非技术性人员了解。但是，它所提供的组织、描述和结构化数据的能力，使它也可为技术性的应用程序所使用。因此，XML 所建立的结构化数据便可同时供数据处理和显示之用。XML 也被称为一种语言，而且能用来创造其他的语言[1]。

综上所述，XML 和 HTML 都来自于 SGML，它们都含有标记，有着相似的语法，HTML 和 XML 的最大区别在于，HTML 是一个定型的置标语言，它用固有的标记来描述，显示网页内容。相对地，XML 则没有固定的标记，XML 不能描述网页具体的外观、内容，它只是描述内容的数据形式和结构。这是一个质的区别，网页将数据和显示混在一起，而 XML 则将数据和显示分开来。XML 不是 HTML 的替代品，XML 和 HTML 是两种不同用途的语言。XML 是被设计用来描述数据的，重点是：什么是数据，如何存放数据。HTML 是被设计用来显示数据的，重点是：显示数据以及如何显示数据更好。HTML 是与显示信息相关的，XML 则是与描述信息相关的[1]。

1.1.2 SGML 与 XML

在这里我们有必要再重新对 SGML 进行更完整的说明，SGML（Standard Generalize Markup Language，标准通用置标语言）属于置标语言，SGML 诞生于 IBM 公司，它具有非常强大的功能，并且已成为被世界各国广泛采用的国际标准。目前，SGML 已被美国波音航空公司用于处理各国每年多达 400 万页的信息资料，被各国的天文机构用于记录数量庞大的天体状态数据，也被众多的跨国公司用于内部数据存储和交换。但其自身也仍然存在一些问题，而且又太复杂，使用和处理起来都绝非易事。为了让结构化的置标语言能够普及使用，特别是在面向信息处理的一些领域，比如 Internet，人们决定使用一套新的标准，这就是 XML[MWL02]。

所以，XML 就是积累了 SGML 20 多年的经验后建立起来的。看名称就知道，它是置标语言的标准，也就是说所有置标语言都是依照 SGML 制定的，当然包括 HTML。SGML 的覆盖面很广，凡是有一定格式的文件都属于 SGML，比如报告、乐谱等等，HTML 和 XML 是 SGML 在网络上最常见的文件格式，因此，人们戏称 SGML 是 HTML 和 XML 的“妈妈”。其中 XML 就是 SGML 的简化版，只不过省略了其中复杂和不常用的部分。和 SGML 一样，XML 也可以应用在金融和科研等各个领域。XML 的语法标准建立在已有的 SGML 的语法标准基础上，是 SGML 的一个兼容子集[MWL02]。

综上所述，XML 和 SGML 的相同点是：都是元置标语言（可以定义其他置标语言的语言）；XML 也适用 SGML 惯用的方式定义和验证文档；XML 文档和 SGML 文档都分为有效文档、成型文档、不成型文档三类；SGML 的编辑软件同样也可以编辑 XML 文档。XML 和 SGML 的不同之处为：XML 的设计目标和 SGML 不同，XML 更适合于网络环境；XML 比 SGML 更容易实现，更容易普及；XML 只是 SGML 的一个子集，而 SGML 则包罗万象[MWL02]。

1.2 XML 数据模型

1.2.1 一个 XML 文档实例

为了感性了解 XML 的文档结构和组成部分，同时为了使本文其他部分的介绍有的放矢，表 1.1 给出了一个 XML 文档的例子[Sun02]。

表 1.1 XML 示例文档

```
1  <?xml version="1.0" standalone="no"?>
2  <?xml-stylesheet type="text/xsl" href="sample.xsl"?>
3  <!DOCTYPE site SYSTEM "sample.dtd">
4  <site>
5      <regions>
6          <africa>
7              <item id="item0">
8                  <name>duteous</name>
9              </item>
10             <item id="item1">
11                 <name>beckon rue</name>
12             </item>
13         </africa>
14         <europe>
15             <item id="item2">
16                 <name>scarce brook</name>
17             </item>
18         </europe>
19     </regions>
20     <closed_auctions>
21         <closed_auction>
22             <seller person="person0"/>
23             <buyer person="person1"/>
24             <itemref item="item1"/>
25             <price>113.87</price>
26         </closed_auction>
27         <closed_auction>
28             <seller person="person5"/>
29             <buyer person="person0"/>
30             <itemref item="item0"/>
31             <price>96.92</price>
32         <closed_auction>
33     </closed_auctions>
34 </site>
```

表的右边一列即为示例的 XML 文档，而左边一列给出了每一行的行号。文档的第一行是一个 XML 声明，XML 声明是 XML 处理指令的一种，XML 文档中的处理指令以“<?”和“?>”标识，一个 XML 文档最好以一个 XML 声明作为开始。XML 声明给出了 XML 文档

采用的 XML 版本号，是否和外部 DTD 配合使用，以及数据所采用的编码方式等信息。文档第二行同样是一个 XML 处理指令，它指明了当显示该 XML 文档时所应使用的样式文件。文档第三行是文档的外部 DTD 使用说明，关于 DTD 的相关知识在下一节中有详细介绍。

从第四行起是 XML 文档的实质内容，即 XML 文档所表示的数据。XML 通过元素来组织 XML 数据，元素是 XML 文档内容的基本单元，XML 元素包括标记和字符数据。从语法上讲，一个元素包含一个起始标记、一个结束标记以及标记之间的数据内容。XML 中有元素、属性、文本等几种基本的数据类型，这些数据类型是我们研究的重点对象，此外，还有一些其他的结构用以实现一些特殊的功能，如 XML 指令等。一个元素的形式是：<标签 属性名=属性值……>数据内容</标签>。其中数据内容可以是元素的值，或者是其他的元素，也可能是两者的混合。

每个 XML 文档有且仅有一个根元素，其他任何元素都是根元素的子孙，即都出现在根元素的内部。如示例中标签为 site 的元素为该文档的根元素。site 有 regions 和 closed_auctions 两个子元素。regions 是一个抽象的地区概念，其子元素代表一系列的地区，如 africa，europe 等，文档中它有 africa 和 europe 两个子元素。每个地区节点可能有若干个名为 item 的子元素，而 item 元素有名为 id 的属性和名为 name 的子元素，name 元素有一个文本类型的元素作为其子节点以表示其取值，如文档中 africa 元素有两个名为 item 的子元素，其中第一个元素的 id 属性值为“item0”，其名为 name 的元素的值为“duteous”。closed_auctions 节点可以有一个或多个名为 closed_auction 的子节点，而 closed_auction 又有 seller、buyer、itemref 和 price 四个子节点，这四个子节点分别有名为 person、person、itemref 的属性和 #PCDATA 类型（文本节点）的子节点。

1.2.2 XML 文档树

一个 XML 文档可以树的形式表现出来，该树被称为 XML 文档树。一棵 XML 文档树被定义为[LV03]：

$$T=(V, chl, lab, val, Vr)$$

其中，V 为 XML 文档树 T 中的节点集合，Vr 为根节点。一棵 XML 树包含三种类型的节点，分别为元素节点、文本节点和属性节点，并且每个节点都标有一个唯一的 ID 号。chl、lab、val 为三个函数。函数 chl 用于求子节点，例如 chl(v)返回在 XML 文档树上的节点 v 的所有儿子节点。函数 lab 为从集合 V 到 $E \cup S \cup A$（E：元素名称集合；S：指代#PCDATA；A：属性名称集合）的映射函数，即 lab 为每一个节点赋予了一个标签，元素类型节点的标签为该元素名；#PCDATA 类型的标签为 S；属性类型的节点为相应属性名。函数 val(v)返回节点的值，具体地，当节点 v 为：

1）属性节点：返回节点的属性值；

2）文本节点：返回字符串值；

3）元素节点：返回节点 v 的 ID 号。

图 1.1 为表 1.1 对应的 XML 文档树，为了清晰区分三种类型的节点，属性节点、文本节点和元素节点分别用三角形、正方形和圆形表示。

以图 1.1 为例，&1，…，&35 为节点的 ID 号。&1 即为 Vr，为文档树的根。&2 为元素节点，用圆形表示；&5 是属性节点，用三角形表示；&7 是文本节点，用正方形表示。chl(&2)=[&3,&12]；chl(&18)=[&19,&21,&23,&25]。lab(&2)为“regions”；lab(&5)返回“id”；

lab（&7）返回 S。val(&2)返回&2；val(&5)返回“item0”；val(&7)返回“duteous”。

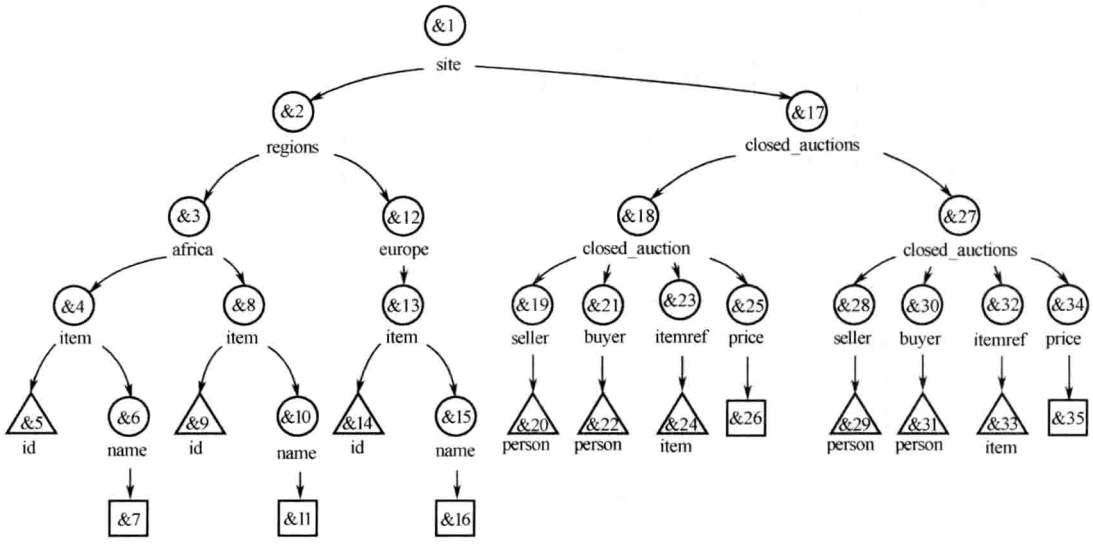


图 1.1　XML 数据树

1.3　XML 文档类型定义 DTD

在具体设计 XML 数据的格式时，要说明 XML 文件将包括哪些元素，这些元素的数据类型，包括他们的子元素，以及子元素的排列顺序等。那么怎么定义这个数据格式呢？答案是使用 DTD（Document Type Definition，文档类型定义）。由于 DTD 具有能够给文档提供正式规范的结构这一优点，所以 XML 也引入了它。按照 W3C 的标准，DTD 应该放在文件头里面。

DTD 可以是一个完全独立的文件，也可以在 XML 文件中直接设定。所以，DTD 分为外部 DTD（在 XML 文件中调用另外已经编辑好的 DTD）和内部 DTD（在 XML 文件中直接设定 DTD）两种。比如，有几十家相互联系的、合作伙伴关系的公司、厂商，他们相互之间的交换电子文档都是用 XML 文档，那么我们可以将这些 XML 文档的 DTD 放在某个地方，让所有交换的 XML 文档都使用此 DTD，这是最方便的做法，同时也适用于公司内部的 XML 文件使用[1]。

1.3.1　DTD 主要语法

DTD 定义的基本格式是 <!DOCTYPE 根元素[……具体规则……]>，其中的具体规则包括很多，例如元素声明 ELEMENT、属性声明 ATTLIST、实体声明 ENTITY、不解析的外部内容格式声明 NOTATION 等。具体的 DTD 语法介绍如下[3]。

1．元素

在 DTD 中，元素类型是通过 ELEMENT 标记声明的。除了关键字，标记还提供所声明类型的名称和内容规范。DTD 中使用元素类型声明 ETD（Element Type Declaration）来声明所有有效的文档元素。ETD 应该采用如下的结构：<!ELEMENT 元素名 元素内容描述>。XML 的标准将元素按内容划分为四类。

空元素类型。定义方式为：<!ELEMENT 元素名 EMPTY>。这类元素在 XML 文档中使用空元素标记，元素中没有内容。

ANY 元素类型。定义方式为：<!ELEMENT 元素名 ANY>。XML 文档里该元素中可以包含任何内容。建议一般只把文档的根元素规定为 ANY 类型。

父元素类型。这类元素中可以包含子元素，在 DTD 中通过正则表达式规定子元素出现的顺序和次数。语法分析器将这些正则表达式与 XML 文档内部的数据模式相匹配，判别出一个文档是否是“有效的”。表 1.2 列出了 DTD 的正则表达式中可能出现的元字符。

表 1.2 DTD 中的元字符

元 字 符	含 义
元素 A 元素 B 元素 C	元素列表，无须遵从顺序要求
,	并（AND），要求严格遵从顺序要求
+	出现一次或多次
*	出现零次或多次
( )	一组要共同匹配的表达式
|	或（OR）
?	可选，不出现或出现一次

混合元素类型。这类元素中可以包含文本，同时文本之间可以有选择地插入子元素，但子元素出现的顺序和次数不受限制。它的定义方式是：

<!ELEMENT 元素名（#PCDATA | 子元素名 1 | 子元素名 2 | ……）*>

2. 属性

属性是对元素的补充和修饰，它能够将一些简单的特性与元素相关联。通过属性，我们可以给元素绑定大量信息。属性在 XML DTD 中是使用 ATTLIST 标记声明的。对于含属性的元素，至少要通过一个 ATTLIST 标记声明其属性列表。ATTLIST 声明由以下部分构成：ATTLIST 关键字、属性修饰的元素名称以及零个或多个属性定义。为了增强可读性，每个属性定义通常占据单独的一行。属性定义包含属性名称、类型和默认声明，在起始标记中出现。属性声明可以要求作者提供属性值，或者完全忽略该属性值，甚至总是使用默认值。这三种类型分别由三个关键词#REQUIRED、#IMPLIED、#FIXED 加以指定。属性的值被解释为纯字符数据，用 CDATA 表示。属性描述的具体格式如下：

<!ATTLIST 元素名称 属性名称 属性类型 默认值>

元素名称指声明具有该属性的元素的名称，属性名称就是属性显示的名称，如前面的元素“item”和属性“id”。在 DTD 的规范中，属性类型比较多，而默认值的四种取值如表 1.3 所示。

表 1.3 属性类型默认值的四种取值

#DEFAULT value：	属性有默认值，其值为“value”
#REQUIRED：	元素的属性必须设定属性值
#IMPLIED：	元素的属性不一定要设置属性值
#FIXED value：	属性的值已经固定了