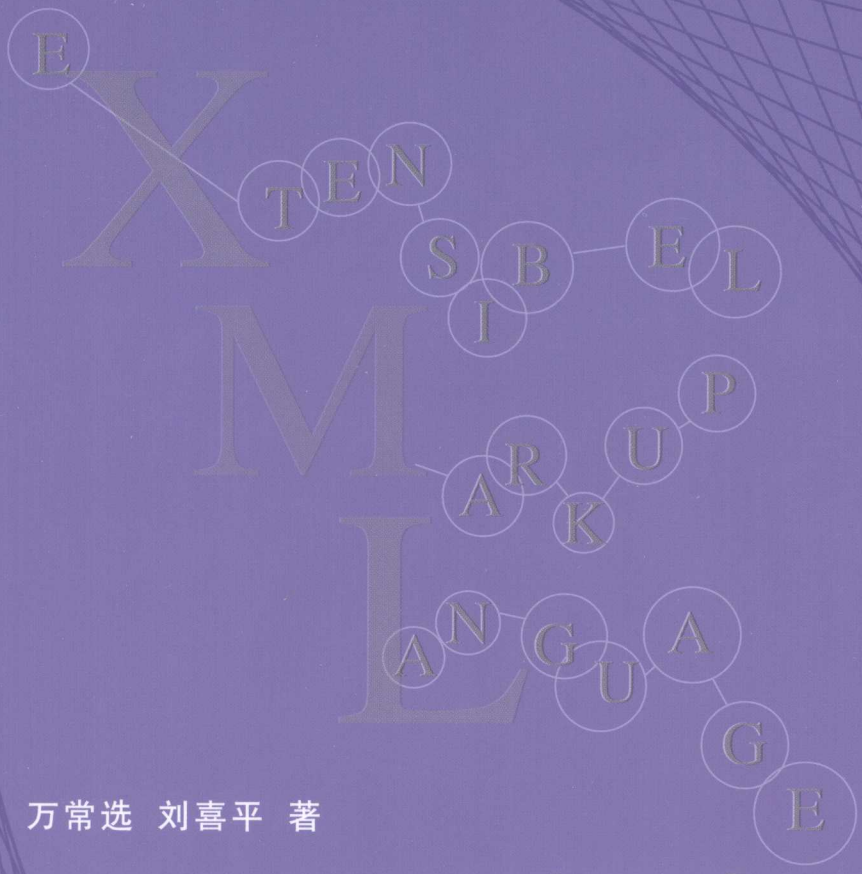


万常选 刘喜平 著

XML数据库技术

（第2版）

清华大学出版社

XML数据库技术
（第2版）

万常选 刘喜平 著

清华大学出版社
北京

内容简介

本书从数据库技术的几大构成来构思，以“存储—索引—查询处理—查询优化”为主线，涵盖了XML数据库技术的主要方面。全书共分7章。第1章介绍了XML的基础知识、相关的技术标准和XML数据库的基本概念，包括XML、XML模式(DTD和XML Schema)、XML数据模型、XPath路径语言、XQuery查询语言和XML数据库的概念。第2章介绍了XML数据库的存储技术，包括基于关系的XML数据存储技术和原生的XML数据存储技术。第3章介绍了XML索引技术，包括结构概要索引、结点编码索引和整体索引。第4章对XML查询处理技术进行了概览和分类，并介绍了XQuery查询的处理和XML-to-SQL查询翻译技术。第5章介绍了XML查询处理中的结构连接算法。第6章介绍了XML查询优化技术，包括XML查询最小化技术、基于代价的XML查询优化、XML结构连接顺序的选择和XML视图查询优化技术。第7章介绍了XML数据库性能评测。

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数据库研究或应用开发人员的参考资料。

图书在版编目(CIP)数据

XML数据库技术/万常选，刘喜平著. —2版. —北京：清华大学出版社，2008.12
ISBN 978-7-302-18985-5

Ⅰ. X…　Ⅱ. ①万… ②刘…　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(2008)第186734号

责任编辑：焦　虹　张为民
责任校对：时翠兰
责任印制：杨　艳

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　　**印　张**：28.25　　**字　数**：698千字
版　　次：2008年12月第2版　　**印　次**：2008年12月第1次印刷
印　　数：1～3000
定　　价：39.00元

序

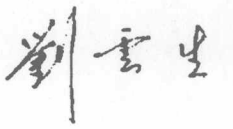
数据库与网络是计算机科学技术领域中近年来发展最快、最受人关注、应用最广的两个分支学科。数据库技术与网络技术已成为当今社会最重要、与社会各界关系最为密切的技术之一，它们共同构成了支持现代信息社会的技术基础。

在数据库领域中，30 多年来人们在数据的表示（包括逻辑模型和物理模型）、数据的处理（包括存储和操作）、数据的使用（包括传输与控制）等方面做了大量创造性的工作，取得了卓越的成绩。数据库系统从“面向记录”的第一代（层次型、网状型）系统，到“面向值”的第二代（关系型、逻辑型）系统，再到“面向对象”的第三代（扩展关系型、语义与函数型、对象关系型、永久对象型等）系统，发展到今天，“数据库园”中已是百花争艳、万紫千红。现代数据库技术发展的一个重要方向，是经典数据库技术与其他一种或多种现代信息/数据处理技术，如面向对象、多媒体、智能/规则处理、时序和实时处理、网络和 Internet/WWW 等的“有机融合”（seamless integration）。其中，与 WWW 的结合是最为必要且迫切的。

Internet/WWW 的出现是网络技术发展史上的一个里程碑。它促使了大量 Internet/WWW 数据的涌现与使用，同时也促进了数据库和 Internet/WWW 两种技术的结合。在当今信息社会，人们迫切希望和要求建立基于 Internet 的数据库系统，盼望能提供 4Wh-ever（Whatever、Whenever、Wherever、Whoever）式的数据处理服务。XML 数据库的应运而生，正是 Internet/WWW 和数据库这两种技术相结合的产物。

面对社会的挑战，广大数据库研究、开发和应用工作者在 Internet/WWW 数据库或 XML 数据库领域中进行了大量、广泛而深入的探究，取得了令世人瞩目的成绩。有关方面的文章如雨后春笋，但相对而言，这方面的专门书籍则要少得多。本书是经过作者多年辛勤工作（尤其是在攻读博士学位期间，远离家人，效“颜回之乐”，作了深入细致的开拓性研究，阅读了大量国内外相关文献），在博士论文的基础之上充实完善而成的，为“数据库沧海”添上了一粟。

当今的信息时代正是数据库的春天，我们广大园丁的辛劳与汗水必将换来数据库百花园中争奇斗艳、姹紫嫣红的美丽景色。



2004 年 11 月

第 2 版前言

1996 年，W3C 的一个工作组开始着手关于可扩展标记语言即 XML 的工作，当时他们也许没有想到 XML 会取得如此大的成功。在随后的十余年里，XML 得到了迅速的发展和极其广泛的应用。2006 年，在 XML 诞生十周年之际，IBM 的科学家撰文指出，XML 构成了计算机科学历史中的一个重要里程碑，并将产生重要的经济、政治和文化影响。他们认为，在对世界产生最重要影响的技术标准中，XML 将占有一席之地。

随着 XML 应用的普及，XML 成为了信息交换和编码的主流格式和事实标准。日益增长的以 XML 形式表示的数据给数据的管理提出了很多新的需求，对数据库技术带来了极大的挑战，为了应付这种挑战，数据库的研究者付出了艰辛和努力，进行了深入的研究，取得了丰硕的成果，形成了现代数据库技术的一个分支——XML 数据库技术。XML 数据库技术的发展不过十年左右的时间，但已经取得了巨大进展，XML 数据库受到了学术界和工业界的普遍认可。今天，在主流的商业数据库中，XML 数据已经成为了“一等公民”，XML 数据的管理已经得到了良好的原生支持。为了充分反映 XML 数据库技术这一新兴领域的现状，作者写作了本书。

本书是《XML 数据库技术》的第 2 版。第 1 版于 2005 年 1 月出版，当时是国内第一本关于 XML 数据库技术的专著。第 1 版出版以来，受到了读者的热烈欢迎。第 1 版出版时，XML 数据库技术本身尚在发展之中，诸多技术前景尚不明了，此后的几年，XML 数据库技术又有了很多新的发展，第 1 版的内容已经不能反映最新的发展状况，为了使读者能够了解较新的现状，作者开始写作第 2 版。本书作者一直在跟踪 XML 数据库技术的发展，对 XML 数据库技术有了一些新的思考，这些思考反映在第 2 版中。

本书不是在第 1 版的基础上的简单修订，而是突破了第 1 版的结构，重新构思和写作。全书分 7 章。

第 1 章介绍了 XML 的基础知识、相关的技术标准和 XML 数据库的基本概念，包括 XML、XML 模式(DTD 和 XML Schema)、XML 数据模型、XPath 路径语言、XQuery 查询语言和 XML 数据库的概念。

第 2 章介绍了 XML 数据库的存储技术，包括基于关系的 XML 数据存储技术和原生的 XML 数据存储技术。

第 3 章介绍了 XML 索引技术，包括结构概要索引、结点编码索引和整体索引。

第 4 章和第 5 章介绍了 XML 查询处理技术。其中，第 4 章对 XML 查询处理技术进行了概览和分类，并介绍了 XQuery 查询的处理和 XML-to-SQL 查询翻译技术；第 5 章着重介绍了 XML 查询处理中的结构连接算法。

第 6 章介绍了 XML 查询优化技术，包括 XML 查询最小化技术、基于代价的 XML 查询优化、XML 结构连接顺序的选择和 XML 视图查询优化技术。

第 7 章介绍了 XML 数据库性能评测，重点是常用的 XML 数据库性能基准和评测数据集。

与第 1 版相比,第 2 版的结构更加合理,内容更加成熟,素材更加新颖。本书从数据库技术的几大构成来构思,以"存储—索引—查询处理—查询优化"为主线,涵盖了 XML 数据库技术的主要方面,脉络清晰。全书内容取自 XML 数据库领域国内外研究的重要成果,主要参考了 ACM SIGMOD、VLDB、IEEE ICDE 等国际会议以及 ACM TODS、VLDB Journal、IEEE TKDE、计算机学报和软件学报等国内外权威期刊,基本上反映了 XML 数据库技术的全貌。本书写作过程中更加注重对于技术的分析、梳理和总结,重点突出,思路明晰,有利于读者把握技术的来龙去脉和发展趋势,可读性更强,参考价值更高。在第 1 版的基础上,本书舍弃或者弱化了一些过时的技术,并加入了大量 2005—2007 年的新研究成果,使得全书在内容上始终反映国际国内研究前沿。

作者所在课题组的研究得到了国家自然科学基金项目(60763001)、国家社会科学基金项目(07BTQ025)、江西省教育厅科技项目(赣教计字[2001]387 号、赣财教[2003]73 号、赣教技字[2006]320 号)和江西省自然科学基金项目(0411009、2007GZS0082)的资助,关于 XML 数据库技术的研究成果曾获得 2006 年度江西省自然科学三等奖,本书第 1 版也获得江西省高校 2005—2006 年度科技成果二等奖。

本书得到了中国计算机学会数据库专业委员会主任委员王珊教授,副主任委员唐常杰教授、于戈教授,秘书长孟小峰教授等的热情鼓励与帮助,同时在编写过程中,参阅了许多专家的研究成果。另外,硕士生王芳为本书绘制了大量图表。在此一并表示衷心的感谢。

本书适合作为计算机相关专业研究生的教材,也可作为从事 XML 数据库研究或开发应用人员的参考资料。

虽然作者力求通过本书反映 XML 数据库技术的全貌和最新进展,但是由于学识所限,加之新技术层出不穷,遗漏与疏忽之处在所难免,恳请专家、同仁和广大读者批评指教。

作 者

2008 年 5 月

第 1 版前言

网络技术的飞速发展改变了人们的学习、生活和工作方式，拓宽了人们获得知识和信息的途径，并且也改变了人们的思维方式。XML 就是这样一种迅速发展的技术。

XML 起初主要是为了增强应用程序从 Web 上获得文档的解释和操作能力而产生的。然而，从数据库的角度来看，XML 从另一个方面提出了一个令人兴奋的可能性：查询这些文档的内容。随着大量 XML 数据的出现，如何有效地存储、管理和查询这些 XML 数据，就成为一个值得研究的重要课题。

对 XML 数据库系统的研究主要有两种途径：一是纯(native)XML 数据库系统，它是为 XML 数据量身定做的数据库。它的优点是充分考虑到 XML 数据的特点，以一种自然的方式来处理 XML 数据，能够从各个方面较好地支持 XML 的存储和查询，但是，纯 XML 数据库要走向成熟还有很长的路。二是 XML 使能(XML-enabled)数据库系统，它是在已有的关系数据库系统或面向对象数据库系统的基础上扩充相应的功能，使其能够胜任 XML 数据的处理。目前，XML 使能数据库的研究主要是基于关系数据库的。这种方法的优点是可以充分利用已有的非常成熟的关系数据库技术，集成现有的大量存储在关系数据库中的商用数据。

对 XML 数据库系统的研究主要集中在 6 个方面：一是对 XML 的查询数据模型、查询语言和查询代数等进行研究；二是对 XML 数据的编码方案和索引结构进行研究；三是对纯 XML 数据库系统进行研究，包括存储结构、索引技术、查询技术和事务管理等；四是对基于关系的 XML 使能数据库进行研究，包括 XML 数据的关系存储模式、XML 查询技术和查询算法，以 XML 文档发布关系数据的技术，以及通过中间件实现以 XML 格式和 XML 查询语言查询关系数据并进行异构数据源的信息集成等；五是对 XML 的查询技术、查询算法，XML 查询的包含与等价、树模式查询最小化，以及查询优化技术等进行研究；六是对 XML 查询测试数据集、测试查询范例进行研究。

目前，XML 数据库技术是数据库领域的研究热点，近年来国际重要的数据库学术会议都收录了大量关于 XML 数据库方面的论文，例如，2004 年国际 VLDB(Very Large Data Bases)会议录用了 79 篇研究论文，其中专门为 XML 开辟了 4 个专题(Session)，录用研究论文 16 篇，占 20%；2004 年国际 ACM SIGMOD(Association for Computing Machinery, Special Interest Group on Management of Data)会议共录用 69 篇研究论文，其中专门为 XML 开辟了 4 个专题，收录论文 13 篇，占 19%。

本书共分 6 章。第 1 章是有关的基础知识，包括 XML、DTD、XML 模式、XPath 和 XQuery 等；第 2 章在介绍了 XML 数据的编码方案之后，对纯 XML 数据库的存储结构、索引技术和事务管理进行了综述；第 3 章讨论了基于关系的 XML 数据库技术，首先对各种映射 XML 数据到关系存储的方法进行了综述，然后重点讨论了我们提出的X-RESTORE 索引结构、关系存储模式以及查询中间件；第 4 章讨论了 X-RESTORE 下的 XML 查询的计算策略和转换 XPath 路径表达式到 SQL 查询的算法；第 5 章讨论了 XML 结构连接技术，

包括各种计算祖先/后裔关系(含双亲/孩子关系)结构连接的直接归并结构连接算法、基于缓存的归并结构连接算法和 twig 模式结构连接算法,以及计算文档位置关系的结构连接算法;第 6 章讨论了 XML 的查询优化技术,主要包括查询最小化、视图查询、估算查询结果大小和选择结构连接顺序等。

本书是在作者博士论文的基础上扩充而成的,首先要感谢我的导师刘云生教授,他对本人在博士研究生阶段的学习和研究工作等进行了悉心指导和无倦教诲。在本书撰写完成之后,刘云生教授又仔细审阅了全书,提出了许多宝贵的意见,并为本书作序。

本书在编写过程中,作者参阅了许多专家的学术论文,特别是香港科技大学陆宏均教授、中国人民大学孟小峰教授等在该领域的大量研究成果,也得到了中国计算机学会数据库专业委员会主任王珊教授的热情鼓励和帮助。在此表示衷心感谢。

本书是作者所在课题组全体师生徐升华、刘喜平、林大海、凌传繁、钱忠胜、张治、唐志涌、江腾蛟等共同的成果。课题的研究得到了江西省教育厅科技项目(赣教计字[2001]387 号、赣财教[2003]73 号)和江西省自然科学基金(0411009)的资助,对江西省教育厅和江西省科技厅的资助表示衷心感谢。在成书的过程中,刘喜平硕士生对全书的初稿进行了阅读和修改,付出了辛勤的劳动,在此也表示感谢。

本书适合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 数据库研究或应用开发人员的参考资料。

尽管我们十分努力,以求本书尽善尽美,但是由于作者的水平有限,加之 XML 数据库技术目前还不成熟,每年都会有大量的研究成果出现,疏漏与谬误之处在所难免,恳请专家、同仁及广大读者批评指教。

万常选

2004 年 12 月

目　录

第1章 绪 论

XML是一种可以用来创建自己的标记的标记语言，即所谓的元标记语言(meta-markup language)，它由万维网协会(World Wide Web Consortium，W3C)创建，用来克服HTML的局限。XML提出以后，迅速风靡了全世界，在各行各业中得到了大量的应用，并成为了众多技术的基础。本章介绍XML的基础知识以及XML数据库的基本概念。

1.1 XML与模式

1.1.1 XML简介

XML(extensible markup language，可扩展标记语言)是由W3C的XML工作组定义的。这个工作组是这样描述该语言的[1]："XML是SGML(standard generalized markup language，标准通用标记语言)的子集，其目标是允许普通的SGML在Web上以目前HTML(hypertext markup language，超文本标记语言)的方式被服务、接收和处理。XML被设计成易于实现，且可在SGML和HTML之间互操作。"

XML和HTML一样，都是SGML的一个子集。SGML由于过于复杂，实际上根本没有流行起来，而HTML则过于简单且缺乏灵活性。XML是一种介于SGML和HTML之间的语言，它保持了SGML强大的标记功能，同时又没有损失灵活性和开放性。XML是一种元标记语言，用户可以定义自己需要的标记，只要这些标记是满足某些最低准则。基于XML，很多行业都已定义了自己的标记语言，如数学家们定义了MathML，化学家们定义了CML等。

相对于HTML而言，XML具有很多优点：

(1) XML是自描述的。XML的最大能量来源于它不仅允许用户定义自己的一套标记，而且这些标记不必像HTML一样局限于对于显示格式的描述。XML可以定义自己的标记集来说明文档的内容，比如可以用一个标记<postcode>来说明330013是一个邮编，用标记<orderno>来说明20080308是一个订单编号等。

(2) XML支持对文档内容的验证。XML文档的结构和内容必须符合一定的文档规则，最起码它要遵守基本的XML语法，除此之外，还往往受到一定模式的约束。有了模式，就可以方便地验证文档的有效性。

(3) XML允许开发各种不同专业的特定领域的标记语言。有了这些语言，这个领域的专业人士就可以自由地交换短文、数据和信息，而不必担心对方是否能够解析和理解这些数据。

(4) XML是非专有并易于阅读和编写的。这使得它成为在不同的应用间交换数据的理想格式。

(5) XML是基于W3C定制的开放标准，从而使得基于XML的应用具有广泛性。

(6) 支持高级搜索。因为可以知晓文档内容的结构和含义(根据它的语法规则),所以很容易在 XML 文档中进行搜索。在 Internet 上如果 Web 页是 XML 格式的,则搜索会更高效,而且不仅可以搜索数据,还可以在搜索中加入与数据相关的上下文信息,这样就形成了更精确的搜索机制。

有人说,XML 是下一代 Web 语言,更有甚者说,XML 是 21 世纪的"世界语"。不管是否确切,这些说法都显示出了 XML 的巨大潜力。下面是 W3C 阐述的 XML 的 10 个设计目标[1]:

- XML 应该可以直接用于 Internet;
- XML 应该支持各种应用程序;
- XML 应该与 SGML 兼容;
- 编写处理 XML 文档的应用程序应该简单;
- XML 中可选特性的数目应该尽可能地少,理想情况是零;
- XML 文档应该便于人阅读,而且相当清晰;
- XML 设计应快速完成;
- XML 的设计应该形式化且简洁;
- XML 文档应该易于创建;
- XML 标记的简洁性是次要的。

目前,XML 在很多方面都有应用。如果需要获得更多的 XML 应用列表,包括每个应用的详细描述,请参见 SGML/XML 的 Web 页(http://www.oasis-open.org/cover/xml.html/#applications)。

XML 的语法十分简单易学。只要了解其中一些简单的规则,就可以轻易上手。虽然它也有一些比较复杂的规则,但是这些规则不难理解。

1. XML 声明

XML 声明必须在文档的第一行,而且其中的字母是区分大小写的。首先,声明使用的 XML 版本号,如<?xml version="1.0"?>,虽然 XML 1.0 是目前唯一的版本,但是仍然要声明版本属性。

文字编码声明位于版本属性之后,其形式为 encoding="UTF-8"。文字编码声明指出文档是使用何种字符集建立的,默认值是 Unicode 编码(UTF-8 或 UTF-16)。

独立文件声明位于文字编码声明之后,如 standalone="yes",独立文件声明使用的属性值可以为 yes 或 no。属性值 yes 表示所有与文件相关的信息都已经包含在文件中,即文件中没有指定外部的实体,也没有使用外部的模式;属性值 no 表示应用程序需要取得文件以外的信息才能完成文件解析。

完整的 XML 声明如:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

2. 元素

通常,元素典型地组成了 XML 文档中的大部分内容。元素有名字(即标记名),它可能

有后裔，后裔可能是元素、处理指令、注释、字符数据(CDATA)段或者字符。一个良构的(well-formed，也称为格式正确的)XML文档必须至少包含一个元素，即文档中必须有根元素。元素由一对标记(即起始标记和终止标记)串行化而成，起始标记的形式是<标记名>，终止标记的形式是</标记名>，元素的后裔则位于起始标记和终止标记之间。如果元素没有后裔，则称为空元素。空元素也可以用一种速记法来表示，即<标记名/>。

XML中的元素名称是区分大小写的。它必须开始于字母或下划线(_)，后面可跟任意长度的字母、数字、句点(.)、连接符(-)、下划线或冒号。

3. 属性

元素可以用属性来注释。属性通常用来给元素提供所显示内容的额外信息。元素的属性在元素的起始标记中给出，形式为：<属性名>=<属性值>。属性名与元素名有相同的构造规则，属性值必须出现在单引号或双引号中。一个元素可以有任意数目的属性，但是它们的名称必须不同。

4. 处理指令

处理指令通常用来为处理XML文档的应用程序提供信息，这些信息包括如何处理文档，如何显示文档等。处理指令可以作为元素的后裔出现，也可以作为文档的顶层结构出现在根元素的前面或后面。处理指令由两部分组成：处理指令的目标或名称、数据或信息，其格式为<?target data?>，目标的构造规则与元素名的构造规则一样。例如，处理指令：

```
<?display table-view?>
```

5. 注释

XML支持注释，注释可以作为元素的后裔出现，也可以作为文档的顶层结构出现在根元素的前面或后面。注释分别使用字符序列“<!--”和“-->”作为开始和结束，注释的文本内容在这两个字符序列之间。

6. 命名空间

因为XML允许设计者自己选择标记名，所以就可能出现重名的情况。XML提供了一种机制即命名空间来解决这个问题。命名空间相当于一个词汇表，它限定了与之关联的所有元素的作用范围。命名空间本身也有名称，它的名称是统一资源标记符(uniform resource identifier，URI)。命名空间的名称和元素的本地名称组合在一起形成了一个全局唯一的名字，即限定名(qualified name，QName)。

命名空间声明通常出现在元素的起始标记中。通常习惯于把一个命名空间的名字映射为另一个很短的字符串，即命名空间前缀(namespace prefix)。命名空间声明的语法为xmlns：prefix='URI'，如果不使用前缀，则语法为xmlns='URI'，这两种情况下，URI都出现在单引号或者双引号中。一个元素只能有一个默认的命名空间声明出现，但是非默认的命名空间声明则是不限的。

所有的XML文档都应该是良构的。良构的XML文档应该是这样的：所有的构造从

语法上都是正确的;只有一个顶层元素,即根元素;所有的起始标记都有与之对应的终止标记,或者使用空元素速记语法;所有的标记都正确地嵌套;每一个元素的所有属性都是不同名的。

图 1-1 给出了一个良构的 XML 文档的例子,这个例子将在本书中多次使用。它是一个包含出版物信息的 XML 文档,它有一个 pub 根元素,该元素中包含一个 library 子元素,以及若干个(即零个或多个)book、article 和 editor 子元素;每个 book 元素中有一个 title 子元素、一个 price 子元素和至少一个(即一个或多个)author 子元素,以及一个属性 year;每个 article 元素中有一个 title 子元素和至少一个 author 子元素,以及一个属性 editorID(引用该文的编辑元素);每个 editor 元素有一个 ID 类型的属性 id,文档中的 editor 元素的 id 属性

```
<?xml version="1.0"?>
<pub>
    <library>Beijing Library</library>
    <book year="2000">
        <title>Database System Concepts</title>
        <price>26.50</price>
        <author id="101">
          <name>Kaily Jone</name>
          <email>kjone@research.bell-labs.com</email>
        </author>
        <author id="102">
          <name>Silen Smith</name>
        </author>
    </book>
    <book year="2001">
        <title>Introduction to XML</title>
        <price>18.80</price>
        <author id="103">
          <name>Kaily Jone</name>
        </author>
    </book>
    <article editorID="105">
        <title>A Query Language for XML</title>
        <author id="104">
          <name>Kaily Jone</name>
        </author>
    </article>
    <editor id="105">
        <name>A. Deutsch</name>
    </editor>
</pub>
```

图 1-1 一个 XML 文档实例

的值为105，一个文档中所有元素的ID类型属性的值在该文档中必须唯一；每一个article元素通过它的IDREF类型的属性editorID来引用另一个元素，被引用元素的id属性值与该元素的editorID属性的值相等，如文档中的article元素引用editor元素。

1.1.2 DTD简介

XML文档本质上是保存信息的结构化载体。为了得到有效的XML文件，还必须要明确文件中的信息必须遵守哪些结构，即需要一种用来描述XML文档中信息结构的机制，这种机制不仅建立了XML文档中可以使用的XML词汇表，还定义了XML文档中元素的顺序、元素的嵌套关系和内容模型，并建立了文档数据的数据类型。解决方案之一是DTD（document type definition，文档类型定义）。DTD列出了可用在文档中的元素、属性、实体和符号表示法，以及这些内容之间可能的相互关系。DTD指定了文档结构的一系列规则。例如，DTD可以确切地规定每个book元素只有一个title子元素、一个price子元素、一个或多个author子元素，等等。

DTD的基本用途是规范标记的使用，并使XML语法分析器能够确认文档。DTD有助于不同的人们和程序互相阅读文件。例如，如果使用一个DTD表示基本的化学符号，就可以确保相互之间都能够阅读和理解对方的文章。DTD显示了文档的一般结构，它确切地定义了在文档内允许出现什么，不允许出现什么。

DTD最初出现在SGML中，它依靠特定的语法来描述XML文档的结构，使用DTD来描述文档规则的一个主要好处是，原有的SGML工具可以很容易地被修改为支持XML。图1-2显示的是一个XML DTD实例，图1-1所示的XML文档符合该DTD的描述。

```
<!ELEMENT pub(library, (book | article | editor) * )>
<!ELEMENT book(title, price, author+)>
<!ATTLIST book year CDATA #IMPLIED>
<!ELEMENT article(title, author+)>
<!ATTLIST article editorID IDREF #IMPLIED>
<!ELEMENT author(name, contact?, email * )>
<!ATTLIST author id ID #REQUIRED>
<!ELEMENT editor(name, contact?, email * )>
<!ATTLIST editor id ID #REQUIRED>
<!ELEMENT library (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT contact (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

图1-2 一个XML DTD实例

一个DTD通过具体说明每一个元素和属性的名称、元素与子元素之间的嵌套关系、子元素的出现次数等来定义XML文档的结构模型，其中可以利用操作符*（0次或多次）、+（至少1次）、?（0次或1次）、|（或选）来定义子元素的出现次数。DTD假设所有取值都只能