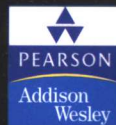




国外经典教材·计算机科学与技术



XML Data Management

Native XML and XML-Enabled Database Systems

XML 数据管理

纯XML和支持XML的数据库系统

Akmal B. Chaudhri

Awais Rashid

Roberto Zicari

编著

邢春晓 张志强 李骅竞

等译



清华大学出版社

国外经典教材·计算机科学与技术

XML 数 据 管 理

纯 XML 和支持 XML 的数据库系统

Akmal B. Chaudhri

Awais Rashid 编著

Roberto Zicari

邢春晓 张志强 李骅竞 等 译

清华大学出版社
北 京

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: XML Data Management: Native XML and XML-Enabled Database Systems, by Akmal B. Chaudhri, Awais Rashid, Roberto Zicari [Eds.], Copyright © 2003

EISBN: 0-201-84452-4

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-4350

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

XML数据管理: 纯XML和支持XML的数据库系统/(美)埃克马尔(Chaudhri, A. B.), (美)阿瓦斯(Rashid, A.), (美)罗伯托(Zicari, R.) 编著; 邢春晓, 张志强, 李骅竞等译. —北京: 清华大学出版社, 2006.2
书名原文: XML Data Management: Native XML and XML-Enabled Database Systems
(国外经典教材·计算机科学与技术)

ISBN 7-302-11983-X

I. X… II. ①埃… ②阿… ③罗… ④邢… ⑤张… ⑥李… III. 可扩充语言, XML-程序设计-教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 119088 号

出版者: 清华大学出版社 地址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮编: 100084

社总机: 010-62770175 客户服务: 010-62776969

责任编辑: 张 靓 龙啟铭

印装者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开本: 185 × 260 印张: 29 字数: 717 千字

版次: 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

书号: ISBN 7-302-11983-X/TP · 7761

印数: 1 ~ 3000

定价: 57.00 元

XML Data Management

Native XML and XML-Enabled
Database Systems



Akmal B. Chaudhri
Awais Rashid
Roberto Zicari
Editors

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师和拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外经典教材·计算机科学与技术

编审委员会

主任委员:

孙家广

清华大学教授

副主任委员:

周立柱

清华大学教授

委员(按姓氏笔画排序):

王成山

天津大学教授

王 珊

中国人民大学教授

冯少荣

厦门大学教授

冯全源

西南交通大学教授

刘乐善

华中科技大学教授

刘腾红

中南财经政法大学教授

吉根林

南京师范大学教授

孙吉贵

吉林大学教授

阮秋琦

北京交通大学教授

何 晨

上海交通大学教授

吴百锋

复旦大学教授

李 彤

云南大学教授

杨宗源

华东师范大学教授

沈钧毅

西安交通大学教授

邵志清

华东理工大学教授

陈 纯

浙江大学教授

陈 钟

北京大学教授

陈道蓄

南京大学教授

周伯生

北京航空航天大学教授

孟祥旭

山东大学教授

姚淑珍

北京航空航天大学教授

徐佩霞

中国科学技术大学教授

徐晓飞

哈尔滨工业大学教授

秦小麟

南京航空航天大学教授

钱培德

苏州大学教授

曹元大

北京理工大学教授

龚声蓉

苏州大学教授

谢希仁

中国人民解放军理工大学教授

译者序

随着 Internet 和信息技术的高速发展, XML 已经成为 Internet 上信息交换和表示的重要标准, XML 数据的数量正在呈指数级增长, 如何高效、系统、科学地管理这些 GB、TB 乃至 PB 级的 XML 文档已成为数据库研究领域中的一个重要挑战。我们非常高兴能够向广大读者推荐这本 XML 数据管理方面系统而全面的专业书籍。到目前为止, 我们还没有看到过任何一本与本书类似的书籍。在本书中, 读者既可以找到对最新出现的纯 XML 数据库的研究和分析, 也可以深入了解支持 XML 的关系数据库系统的相关信息和知识。同时, 本书也详尽地给出了各种评测基准以及用于测试 XML 数据库性能的分析技术。

本书的翻译工作主要由邢春晓、张志强、李骅竞、廖雨果、钱乾完成, 另外, 李超、林玲、徐尧强、李新伟、齐畅、张勇、高凤荣、李越、贺炜、李娟等也协助完成了部分章节及校对。邢春晓对全书进行了最后的统稿和审校。

限于水平, 译文中的疏漏和错误之处在所难免, 敬请广大读者批评指正。

译者

前言

在过去的几年里，XML（可扩展置标语言）得到了快速普及和广泛应用。这种爆炸式增长的主要驱动力来源于它提供了一种标准化、可扩展的方法，将语义信息包含在描述半结构数据的文档内。这使 XML 能够克服现存置标语言（如 HTML）的缺点并支持电子商务环境中的数据交换。

例如，考虑列表 P-1 中一段简单的 HTML 文档，该文档中既包含了数据，也包含了数据的相关表现信息。事实上，标签仅仅描述了数据将如何被格式化，而没有体现出数据是用来表示人的姓名和地址的语义信息。结果造成一个解释器不能做出任何关于语义的有效判断，即使标签内包含了关于车和它的零部件的信息。WIRE 系统（Aggarwal et al. 1998）使用基于 HTML 文件的结构和包含于诸如定义文件头之类的标签中的重要信息，可以解释一些语义信息。但是，这样的解释缺乏有效依据，并且它的准确性是上下文相关的。

列表 P-1 一段关于个人的 HTML 文档

```
<html>
<head>
  <title>Person Information</title>
</head>
<body>
<p> <b>Name: </b>John Doe</p>
<p> <b>Address: </b>10 Church Street, Lancaster LAX 2YZ, UK</p>
</body>
</html>
```

对于动态 Web 页面，数据在一个后端的数据库中，并且利用一个预定义的模板提供服务，从而减少了数据及其表现的耦合程度。但是，数据的语义信息在进行电子商务信息交换时仍然会产生混乱。在一个 B2B（商务对商务）的事务中，一个具体的项目在两个不同的系统中可以使用不同的名称表示（在最简单的情况下），这时语义解析必然依赖于复杂的、私有的文档标准。

XML 提供了对上述问题的固有支持，因为在一个 XML 文档中数据具有自描述性。但是随着 XML 的大量应用也带来了新的挑战性问题。关键问题之一就是大量 XML 文档的管理问题，产生了对有效存储、检索、操纵 XML 数据的工具和技术的需要。这本书的目的就是讨论有关工具和技术的理论和方法。

在综述 XML 数据管理的问题及其解决方法之前，本部分介绍了 XML 的基础和一些相关的技术，关于 XML 和相关技术的介绍仅仅提供了一个概述，因为其他一些书籍已经比较深入地讲解了这些概念。

P.1 什么是 XML

XML 是 W3C 关于文档标记的一个标准，它使得定义描述数据的标签成为可能。列表 P-2 是包含关于一个人的数据的 XML 文档例子。注意 XML 中的标签可以包含属性。但是，为了简化，没有在这个例子中使用属性。

列表 P-2 关于一个人的 XML 文档

```
<?xml version="1.0" standalone="yes"?>
<person>
  <name>
    <surname>Doe</surname>
    <firstname>John</firstname>
  </name>
  <address>
    <housetnumber>10</housetnumber>
    <street>Church Street</street>
    <town>Lancaster</town>
    <postcode>LAX 2YZ</postcode>
    <country>UK</country>
  </address>
</person>
```

不像列表 P-1 的 HTML 文档，列表 P-2 仅仅包含了关于人的数据而没有表现信息。数据和它的意义可以从文档中读出，文档所需要的表现格式定义在另一个文档中，一个标准的方法就是使用可扩展样式语言（XSL）。

XML 的灵活特性使它成为定义任意语言的一个理想基础，其中一个例子就是无线置标语言（WML）。类似地，用于描述 XML 文档结构的 XML 模式语言也是建立在 XML 自身基础上的。

P.1.1 良构和有效的 XML

虽然 XML 的句法是灵活的，但它受限于一个语法，该语法控制着允许的标签名、附着于标签的属性等。所有 XML 文档必须遵循这个基本的语法规则。凡是遵循该语法的文档称为良构的（well formed）并且可以被 XML 解释器解释，这意味着不需要为每个 XML 文档实例写一个解释器。

除了良构以外，一个具体的 XML 文档结构还可以通过一个 DTD（文档类型定义）或 XML 模式进行有效化。一个 XML 文档遵循给定的 DTD 和模式时，被称为是有效的（valid）。

P.1.2 以数据为中心和以文档为中心的 XML

XML 文档可以根据它们所包含的数据进行分类。以数据为中心的文档包含结构化的数

据，如一个产品类目、一个订单或一张发票等。与以数据为中心的文档不同，以文档为中心的文档包含非结构化的数据，如文章、书或电子邮件等。当然，这两类 XML 文档可以合并成一个混合文档，也就是既以数据为中心也以文档为中心的文档。列表 P-3 和列表 P-4 分别提供了以数据为中心的文档和以文档为中心的文档的例子。

列表 P-3 以数据为中心的 XML 文档

```
<order>
  <customer>Doe</customer>
  <position>
    <isbn>1-234-56789-0</isbn>
    <number>2</number>
    <price currency="UKP">30.00</price>
  </position>
</order>
```

列表 P-4 以文档为中心的 XML 文档

```
<content>
  XML builds on the principles of two
  existing languages, <em>HTML</em>
  and <em>SGML</em> to create a simple
  mechanism . . .
  The generalized markup concept . . .
</content>
```

P.2 XML 概念

本节将介绍 XML 的一些基本概念，包括 DTD 和 XML 模式、DOM 和 SAX。

P.2.1 DTD 和 XML 模式

DTD 和 XML 模式都是实现 XML 文档结构定义的工具。它们决定了一个 XML 文档将包括什么元素，如何使用这些元素，元素属性的默认值是什么，等等。当给出了一个 DTD 或者 XML 模式以及与之对应的 XML 文档，XML 解析器就可以判断这个文档是否符合它应有的结构和其他限制。在数据交换的应用背景中，这种功能尤其有用，因为 DTD 和 XML 模式提供并强制要求了一个适用于被交换数据的通用的词汇表。

XML DTD 是标准通用置标语言（SGML）DTD 的子集。一个 XML DTD 列举了在一个文档中出现的不同元素和属性，以及它们使用的上下文环境。同时它还列举了一个文档不能包括的元素。然而，它不能定义一些必要的限制条件，诸如在一个文档中某一个特定元素出现的次数、每个元素包括的数据类型，等等。因此，DTD 更适用于以文档为中心的

XML 内容，而不是以数据为中心的 XML 内容。这是因为在前者中数据类型和实例限制并不如在后者中那般重要。但是，DTD 可以并且正在被以上这两种文档类型使用。

列表 P-5 显示了列表 P-2 中的简单 XML 文档的 DTD 定义。它描述了对三个复合元素 person、name 和 address 来说，哪些简单元素可以组成有效的复合内容。关键词 #PCDATA 表示元素内部不能包括任何标签或者子元素，而只能包括可解析的字符数据。

列表 P-5 列表 2 中简单 XML 文档的 DTD

```
<!ELEMENT person (name, address)>
<!ELEMENT name (surname, firstname)>
<!ELEMENT surname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT address (houzenumber, street, town, postcode, country)>
<!ELEMENT houzenumber (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT town (#PCDATA)>
<!ELEMENT postcode (#PCDATA)>
<!ELEMENT country (#PCDATA)>
```

XML 模式与 DTD 在以下方面不同：XML 模式定义语言本身基于 XML。所以，不像 DTD 的定义，可用于定义 XML 文档的构造集合是可扩展的。XML 模式同时支持命名空间和相对于 DTD 更丰富和更复杂的结构定义。而且，可以更加强大地描述一个标签所包括的数据的类型限制，因为 XML 模式支持一系列简单数据类型，诸如字符串、小数和整数等。这个功能使得 XML 模式非常适合定义以数据为中心的文档。另一个显著的优势在于 XML 模式定义可以利用为 XML 设计的数据管理方法。一个 XML 模式本身就是一个 XML 文档。这与 DTD 形成了一个非常鲜明的对照，DTD 需要在 XML 数据管理系统中加入特定的支持。

列表 P-6 显示了列表 P-2 中简单的 XML 文档的 XML 模式定义。sequence 标签是一个复杂标签定义，它表示子元素是一个有序的序列。同时还有其他的复杂标签定义，比如 choice 和 all。注意，请参见列表中对 address 元素的定义，该定义限制了在一个文档中某个元素出现的最大和最小实例数。虽然在这个例子中没有出现，XML 模式也允许自定义复杂和简单的类型。比如说，可以为元素 address 定义一个叫 Address 的复杂类型。

列表 P-6 列表 2 中简单 XML 文档的 XML 模式

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="surname" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<xs:element name="firstname" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="address" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="houseNumber" type="xs:integer"/>
      <xs:element name="street" type="xs:string"/>
      <xs:element name="town" type="xs:string"/>
      <xs:element name="postcode" type="xs:string"/>
      <xs:element name="country" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

P.2.2 DOM 和 SAX

DOM (Document Object Model, 文档对象模型) 和 SAX (Simple API for XML, XML 简单应用程序接口) 是应用中操作 XML 文档的两种主要 API, 现在是用于 XML 处理 (JAXP 版本 1.1) 的 Java API 的一部分。DOM 是 W3C 定义的标准文档对象模型, 是一个与操作系统和编程语言无关的、用于内存中存储和操作层次化文档的模型。一个 DOM 解析器解析一个 XML 文档, 建立一个对应的 DOM 树, 它可以被用在不同的结点之间进行遍历。然而, 在遍历之前必须完成树的建造。所以, 在处理规模较大的 XML 文档时内存管理会是一个问题。这件工作是非常占用资源的, 尤其是在只需要操作文档的一小部分时。

SAX 是一个实际上通用的标准。它与 DOM 不同, 它使用一个事件驱动模型。每次当遇到一个开始或者结束标签, 或者是遇到一条处理指令时, 程序会得到通知。所以, 在操作文档之前不需要对整个文档进行解析。实际上, 文档的各个部分可以在进行解析的同时进行操作。因此, SAX 相对于 DOM 来说更适合于操作大文档。

P.3 与 XML 有关的技术

本节描述一些与 XML 有关的技术, 包括 XPath、XSL 和 SOAP。

P.3.1 XPath

XPath (XML 路径语言) 为在一个 XML 文档中定位和链接信息提供了通用的语法和

语义。使用 XPath，信息可以用两种方式来定位：

- 基于在一个文档树中元素顺序的分层方式
- 基于在一个文档树中拥有惟一标识符的元素的随机方式

列表 P-7 中显示了基于列表 P-2 中的示例 XML 文档的一些 XPath 表达式例子。例 1 表示所有在当前元素下所有标签名称为 `firstname` 的子元素。例 2 选择在当前元素下所有父标签名称为 `name`、标签名称为 `surname` 的结点。例 3 测试是否有元素满足元素 `name` 和 `address` 的合集。值得注意的是，虽然在例子中没有显示，但 XPath 可以指定约束条件，比如“在文档中出现的第三个人的第一个地址内容”。

列表 P-7 XPath 表达式的例子

1. `select="firstname"`
2. `select="name/surname"`
3. `match="name | address"`

P.3.2 XSL

由于 XML 文档本身不包含任何有关表现方法的信息，因此它可以很灵活地进行格式设计。定义 XML 文档格式的一种标准化的方法就是使用 XSL(可扩展样式语言)。W3C XSL 规范包括两个部分：XSL FO(XSL 格式对象)和 XSL 转换(XSLT)。

XSL FO 提供了为表达 XML 文档而进行的格式化和流程语义。一个 XML 文档创建代理程序负责解释 XSL FO 提供的抽象结构，为了实例化一个特定媒介的表现。

XSLT 提供将一种组织方式的信息转换为另一种组织方式的构造方法。虽然 XSLT 是被设计用于把一个 XML 的词汇转换为一个 XSL FO 的词汇，但它也可以被用于其他一系列的转换工作，比如转换为 HTML 内容，如列表 P-8 所示。示例样式单使用了一些简单的 XSLT 模板和 XPath 表达式，把列表 P-2 中的 XML 文档的一部分转换为 HTML 形式(参见列表 P-9)。

列表 P-8 列表 P-2 中 XML 文档的一个 XSL 样式单

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl=
    "http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <html>
      <head><title>Person Information</title></head>
      <body>
        <xsl:apply-templates select="person/name"/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="name">
    <xsl:apply-templates/>
  </xsl:template>
</xsl:stylesheet>
```

```
</xsl:template>
<xsl:template match="surname">
  <p><b><xsl:text>Surname: </xsl:text></b>
    <xsl:value-of select="."/></p><br/>
</xsl:template>
<xsl:template match="firstname">
  <p><b><xsl:text>First name: </xsl:text></b>
    <xsl:value-of select="."/></p>
</xsl:template>
</xsl:stylesheet>
```

列表 P-9 根据列表 P-8 转换后的 HTML 结果

```
<html>
  <head>
    <title>Person Information </title>
  </head>
  <body>
    <p>
      <b>Surname: </b>Doe
    </p>
    <br>
    <p>
      <b>First name: </b>John
    </p>
  </body>
</html>
```

P.3.3 SOAP

SOAP (Simple Object Access Protocol, 简单对象访问协议) 是在互联网上使用 XML 和 HTTP 调用代码的简单对象访问协议。它的机制与 Java 远程方法调用 (RMI) 类似。在 SOAP 中, 方法调用被转换为 XML, 并且通过 HTTP 协议发送。SOAP 的设计是为了与 XML 模式的兼容, 但是并不强制使用 XML 模式。由于基于 XML, XML 模式可以提供一个描述和发送 SOAP 类型的无缝方法。

P.4 XML 数据管理

到目前为止, 已经讨论了 XML 的基本概念和与之相关的一些技术。这些讨论着重强调了 XML 的最根本优势, 并给出了 XML 广泛流行和普遍采纳的深层原因。随着越来越多的机构和系统在它们的信息管理和交换策略中使用 XML, 经典的数据管理问题就应运而生, 即如何合理并且有效地存储、获取、查询、索引和操作 XML 文档。同时, 以前没有遇到过的信息建模挑战也随之出现。

数据库产品提供商已经对这些新的数据和信息管理需求做出了回应。大部分商业的关系型、对象—关系型和面向对象型数据库系统提供了扩展、插件以及其他方式以支持对 XML 数据的管理。除了在现有数据库管理系统中支持 XML 之外，纯 XML 数据库也出现了。它们的设计可以无缝地存储、获取和操作 XML 数据，并且集成了其他相关技术。

随着市场上大量可用的处理方法和解决方案的出现，XML 数据管理系统的开发者需要面对一系列的挑战：

- 在各种不同的 XML 数据管理方案中哪些是有效的？
- 这些不同的 XML 数据管理系统可以提供哪些特性、服务和工具？
- 如何开发一个定制的解决方案，而不是使用一个现成的商业系统？
- 对某一特定应用来说，哪一个 XML 数据管理系统或方法拥有最佳的性能和效率？
- 在使用 XML 进行信息建模时，是否存在着优良的实践经验，或者存在着针对特定应用的指导原则？
- 在某一特定领域中，是否有其他的 XML 数据管理实例与应用系统？

本书希望通过强调上述挑战性问题给出一种支持机制。书中对一系列产品和应用领域中使用的不同 XML 数据管理方法进行了讨论。同时，也提供了关于 XML 建模的性能和基准测试结果，以及相关原则。

P.5 本书是如何组织的

本书分为 5 个部分，每个部分包括了一系列内容一致并且紧密关联的章节。需要指出的是，这些部分是相对独立的，读者可以采用任何的顺序阅读它们。这 5 个部分列举如下：

第 1 部分：绪论

第 2 部分：纯 XML 数据库

第 3 部分：XML 和关系数据库

第 4 部分：XML 应用系统

第 5 部分：性能和评测基准

下面将对这 5 个部分进行概述。

P.5.1 第 1 部分：绪论

这部分包括一章内容，它着重介绍了使用 XML 进行信息建模时为了获得良好的语法格式和风格所应遵循的一些原则。该章作者 Brandin 阐述了好的语法格式可以减少翻译 XML 的应用程序所需要的冗余的领域知识。另一方面，良好的 XML 风格可以改进应用性能，尤其是在存储、获取和管理信息时。这一章提供了有关 XML 内在的信息建模模式和常见的 XML 信息建模缺陷的深入探讨。

P.5.2 第 2 部分：纯 XML 数据库

这个部分包括了两种纯 XML 数据库系统——Tamino 和 eXist。在第 2 章里，Schöning

在讨论 Tamino 的 XML 存储方法和索引特征之前对 Tamino 的体系结构以及 API 进行了概述，同时也描述了 Tamino 的查询、工具支持和访问其他类型数据仓储中的数据等功能。此章提供了对开发 XML 数据管理应用过程中的重要特性的深入讨论。

Meier 采用了类似的方式，在他所写的第 3 章介绍了开放源系统 eXist 的不同特征和 API。然而，与第 2 章不同的是，该章论述的重点在于如何在 eXist 系统内部进行查询处理。因此，作者更深入地探讨了系统的索引和存储体系。以上两章均提供了一个权衡的讨论，既强调系统的高层应用编程特征，又涉及与高效查询处理有关的底层索引和存储机制。

最后，第 4 章介绍了一个嵌入式 XML 数据库系统的例子。该系统基于一个通用的嵌入式数据库引擎——Berkeley DB。Berkeley DB XML 可以原生地存储 XML 文档，同时提供了索引功能和一个 XPath 查询接口。此产品的一些功能将通过示例代码进行介绍。

P.5.3 第 3 部分：XML 和关系数据库

这部分把一些在关系型和对象—关系型数据库系统中支持 XML 数据管理的产品和办法进行了有趣的结合。第 5、6、7 章讨论了三种商业数据库产品：IBM DB2、Oracle9i 和 MS SQL Server 2000。而第 8、9 两章描述了对关系型和对象—关系型系统更加通用的、用户决定的策略。

Benham 所写的第 5 章突出了 IBM 公司提供的 XML 数据管理和信息集成产品的技术和体系结构。该章的重点在于 DB2 通用数据库和 Xperanto。前者通过 DB2 XML Extender、扩展 SQL 和对 Web service 的支持，实现了在提供关系型和对象—关系型数据管理系列产品上的 XML 应用功能。而后者则是计划中用于满足信息集成需求的一系列产品和功能，它们着眼于通过提供附加的对 XML 以及结构化与非结构化应用的支持，实现对 DB 产品功能的补充。

在第 6 章中，Hohenstein 讨论了在 Oracle9i 中实现的与前者类似的特征：使用 Oracle 的 CLOB 功能与 OracleText Cartridge 处理以数据为中心的 XML 文档；使用 XMLType，一种基于 Oracle9i 的对象—关系功能的新的对象类型，管理以文档为中心的 XML 文档。他向读者展示了为 XML 设计的 Oracle SQL 扩展，并提供了如何使用这些扩展实现从关系型数据中创建 XML 文档的例子。本章中也描述了诸如 URI（统一资源标识符）支持、解析器、类产生器等面向 XML 的特征和工具，以及封装这些功能的 Java Bean。

在第 7 章中，与第 5 章和第 6 章类似，Rys 介绍了 MS SQL Server 2000 中的一系列特征集合。他着重强调了结构化 XML 数据的输出和输入的应用场景。所以，该章的重点在于介绍 HTTP 和 SOAP 访问，可查询和可更新的 XML 视图，基于 XML 的记录集视图，关系型结果的 XML 串行化等功能。记录集视图和 XML 串行化功能的目的在于为更熟悉关系型环境的用户提供 XML 支持。另一方面，XML 视图为熟悉 XML 的用户提供了基于 XML 的数据库访问。

总体上说，第 5、6、7 章对三种商业系统提供的功能进行了一次有趣的比较，读者可以从这些章节中发现它们的 XML 数据管理方案中种种的相似点和不同之处。相反地，分别由 Edwards 和 Brown 编写的第 8 章和第 9 章着眼于一般的、与产品提供商无关的解决方案。

Edwards 描述了在关系数据库环境中存储 XML 文档的一种一般性的体系结构。这种方法的目的旨在避免使用与特定数据库产品相关的扩展功能，并且为数据库应用开发者提供一个在没有机会实现很多先进 XML 技术的条件下体验 XML 数据存储的机会。在这种方案中，数据库模型基于嵌入了嵌套集合模型的 DOM 模型，所以这种方案可以减轻导航的复杂度，并且可以存储任何结构完整的 XML 文档。以上这些设计可以提高串行化和查询的速度，但是在更新的时候会有更多的资源开销。

与 Edwards 着眼于支持传统的关系数据库开发者不同，Brown 提供的方案尽量利用对象—关系模型提供的先进功能以及大部分关系数据库系统提供的不同扩展功能。他讨论了对对象—关系模式设计的方法，这种方法是基于向 DBMS（数据库管理系统）的核心类型与操作符中引入与 XML 标准类型和操作符等价的单元实现的。对 DBMS 核心来说需要的关键功能是一个可扩展的索引系统，它应该允许重载内建的 SQL 类型比较运算符。于是，新的 SQL 3 类型是把 XPath 表达式映射为针对模式的 SQL 3 查询语句的基础。

P.5.4 第 4 部分：XML 应用系统

这部分研究了几个 XML 数据管理的应用系统和用例，范围从生物信息学、地理和工程数据管理到客户服务和资金流改进，甚至到大型分布式系统、数据仓库和归纳数据库系统。

在第 10 章中，Direen 和 Jones 讨论了在生物信息学中面临的数据管理方面的挑战，研究了 XML 在获取和展现复杂的生物学信息过程中扮演的角色。他们认为 XML 灵活的、可扩展的信息模型非常适合这种应用背景。同时，为了能够紧紧跟上生物学数据管理需求的发展，数据库技术必须能够展示相同的特性。他们在这个背景中讨论了 NeoCore XML 管理系统的作用，并且集成了 BLAST（基础的本地联盟检索工具）顺序搜索引擎以提高系统捕获、操作、分析和发展有关组成生物组织的复杂系统信息的能力。

Kowalski 在第 11 章中提供了两个有关 XML 和 IBM DB2 通用数据库的用例研究。她的第一个用例是一个顾客服务单元，它需要首先为最重要的顾客解决问题。第二个用例着眼于在一个学校中通过减少教育部门偿还花费的时间以改进资金的流动过程。作者提供了每个用例的应用场景与研究需要解决的特定问题，并且提供了对阻碍问题解决的现状的研究。对每个用例，书中都对如何使用 XML 和 DB2 设计一个合适的解决方案的描述进行了总结。

Eglin、Hendra 和 Pentakolos 编写的第 12 章描述了 JEDMICS 开放访问接口（一个基于 EJB 的 API，它可以访问存储在多种介质中的图像数据和存储在一个关系数据库中的元数据）的设计与实现。JEDMICS 系统使用 XML 作为一种可移植的数据交换方案。作者在本章中讨论了有关把 XML 与系统的面向对象的核心及提供持久化存储的关系数据库进行集成所带来的问题。本章的一个非常有趣的地方是作者提供了他们在 JEDMICS 环境中对一些 XML 技术的使用体会，这些技术包括 DOM、JDOM、JAXB、XSLT 和 Oracle XSU。

在第 13 章中，Wilson 和她的合作者们提供了使用 XML 提高 GIDB（地理空间信息数据库）系统在互联网上交换地理数据能力的研究情况。他们描述了把通过 METCAST 系统从远程获取的气候和海洋数据集成到 GIDB 的过程。XML 在这里发挥了关键的作用，因为