China-pub.com
下载

第1章 绪 论

- XML是Web和数据库技术的一个完美结合吗？
- 这种结合会持久吗？
- 它是昙花一现吗？会不会随着不兼容性的出现而注定灭亡呢？
- 或者，它是不是一个长期的结合呢，每个技术都支持和加强对方吗？

本书对XML数据库技术进行了详细的介绍：怎样设计数据库，怎样创建系统管理数据库，以及怎样平衡现实世界中的系统等等。本书除了讲述必需的有趣的例子外，还列举了许多生物学领域的例子，甚至还提供了一个复杂的实例，更加适合于利用其广泛的应用范围来讨论 XML数据库设计的复杂性。

本章将介绍XML和数据库，并将详细说明如下问题：

- 什么是XML数据库
- 如何使用、设计、创建XML数据库
- 什么技术是可用的
- 它们的优势和局限性是什么

XML数据库可以通过三个方法集成Web技术和数据库系统：

- 通过把文档转换为关系或对象，XML文档可以存储在一个关系型或面向对象的数据库管理系统（DBMS）中。
- 用数据库软件把数据库中现有的关系或对象表示为XML。
- 用一个基于XML的数据模型创建一个新的DBMS。

本书通过描述数据库系统来表示这些任务，利用描述过程来支持这些任务。好的设计强调理论和实践相结合，重点在于建立强健的各种领域的应用系统结构。

1.1 XML

1.1.1 什么是XML

XML代表可扩展的标记语言。它是一种把数据表示为一个文本字符串的语言，这个文本字符串还包括用于描述数据的散布的“标记”。使用标记允许把文本和与它的内容或形式相关的信息散布在一起。

标记（tag）用角括号“<”和“>”包含字符（例如，“<this>”）来区别字符数据（非标记文本）。因此一个字符串、一个文档由标记和字符数据组成，这些结合起来就形成了元素。一个元素（element）以一个开始标记开始，以一个结束标记结束。结束标记用角括号和一个用来区分开始标记的正斜杠“/”来表示，像“</this>”。<keyword>interspersed</keyword>是标记，在标记里面的内容是文本，而从一个开始标记到它的结束标记的文本区域是一个元素，像

需要全本请访问 www.ertongbook.com

China-pub.com 下载

“ <keyword>interspersed</keyword> ”就是一个元素。

尤其是，标记提供一种机制来给文档添加元内容和结构信息。标记为注释元素中的字符数据提供了一种机制。元素是 XML的基本的组成部分。元素可以包含嵌套在其中的其他元素，叫做子元素（subelement)。一个文档由一个单一的最外面的（或最高级的）元素组成，它包含其他元素和／或字符数据，并且每个子元素可以包含其他带字符的“ 子元素的子元素 ”（sub-subelement），等等。示例 1-1给出了一个 XML文档例子。这个文档由后面跟着元素“ <story>...</story> ”的处理指令“ <?xml version= “ 1.0 ” > ”组成。“ story ”元素有一个开始标记“ <story> ”和一个结束标记“ </story> ”。“ story ”元素包含四个元素，每一个都以“ para ”标记标示。

示例1-1 一个XML文档示例

```
<?xml version="1.0"?>
<story>
<para>
John and Bob were walking down to the lake to go fishing. They enjoyed
fishing together and had been going regularly for almost twenty
years. Today was different. There was a man already in their spot with
his line in the water.
</para>
<para>
John and Bob walked around the man and shifted down the shore to the
side of a little inlet. It was not as good of a spot, but they had
done well there years ago, and it would do.
</para>
<para>
John and Bob each cast their line in the water. John using the same
rod he always had, and Bob using the new graphite one his son got him
for his birthday. John got a tug on the line and reeled in a 7-inch
rockfish. It was too small to keep, and John tossed it in the water.
</para>
<para>
The fish was glad to be alive when it hit the surface of the lake and
could breathe the water again. In relief, the fish started swimming
furiously and  yelled out to the rest of the lake, "Hello World!!!".
</para>
</story>
```

一个XML文档由一个元素（和它的子元素）组成。嵌套的元素也可以用于描述另一个元素的一部分。图1-1中的图形标记可以用示例1-2中的XML来表示。为了注释，创建了标记的层次。

两个示例文档也说明了 XML的两个特性。第一个特性就是一个文档可以被标记，并且还要可以被理解；实际上，添加 XML标记保存了文档中的信息。 XML的第二个特性就是它可以表示层次的（嵌套的）信息。因此， XML标记可以给一个文档添加关于它的内容和结构的信息。

示例1-2 一个XML标记的例子

```
<?xml version="1.0"?>
<sentence language="English">
<subjectOfBook><noun>XML</noun></subjectOfBook><verb type="be">is</
```

China-pub.com
下载


```
verb> a
<annotation>
  <text>language</text>
  <note>
    actually an artificial language rather than a
    <annotation>
      <text>natural</text>
      <note>
        by natural language I mean a language developed over time
        by humans for communicating with each other rather than an
<annotation>
          <text>artificial</text>
  <note>
    by artificial I do not mean that it is not real,
            but that it is an
    <definition>
      <text>artifact</text>
      <def>
        an artifact is any object created or shaped by humans
      </def>
           </definition>
  </note>
</annotation>
language, such as a programming language
      </note>
    </annotation>
    language
   </note>
</annotation>
<prepPhrase>for <colloqial>marking up</colloqial> text</prepPhrase>.
</sentence>
```

一个具有相同属性的元素集合称作为一个元素类型，元素类型通过给那些元素的标记一个惟一的名字（有时候叫做标记名）来区别。例如，例子中的每个“ <para> ”标记标示“ para ”元素类型的一个成员，并且“ <a> ”、“ <a href= “ there ” > ”和“ <a index= “ 1 ” > ”所有都是名字为“ a ”的元素类型的开始标记。XML中的类型与编程语言中的类型使用方法相同，编程语言中，“ 1 ”、“ 2 ”和“ 3 ”都是名字为“ integer ”的数据类型的成员。

一些编程语言具有可扩展的数据类型，诸如 Java这样的面向对象的编程语言。如果你创建一个名叫“ color ”的对象类，那么它可能具有名字为“ red ”、“ blue ”、“ yellow ”和“ violet ”的实例。面向对象编程语言可扩展性的一个优点就是你不必重载其他数据类型，例如可以数字“ 1 ”表示“ red ”，数字“ 2 ”表示“ blue ”，等等。同样， XML的可扩展性避免了 HTML的重载，在 HTML中，“ <b> ”（表示粗体），有时候表示强调，有时候表示节头，还有时候表示一个关键字搜索结果。在 XML中，元素类型是可扩展的，因此，你可以创建一个新的名字为“ color ”，或为“ keyword ”，或为“ babyPicture ”的元素类型，这些元素类型具有属性，属性被表示为标记的一部分，例如<color rgb= “ 110 ” >或<story language= “ English ” keyword= “ fish ” >。一个属性由一个名字 —— 值对组成。

XML为注释和标记字符数据提供一种可扩展的机制，这种机制扩展了与文档和数据库中的数据相关联的信息类型。

China-pub.com 下载

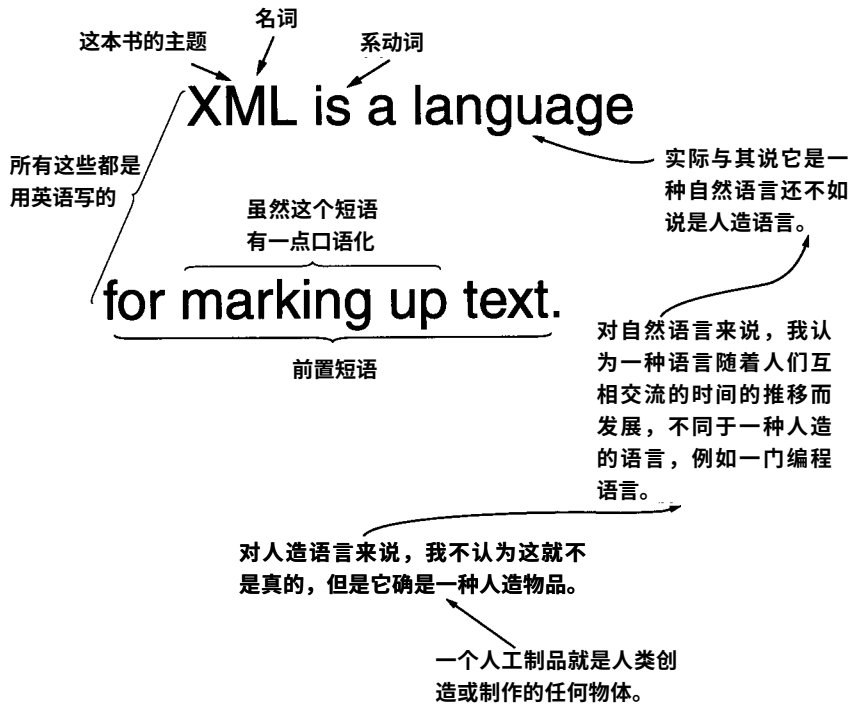


图1-1 被标记的文本

1.1.2 XML来自哪里

第一个标记语言GML发明于1969年，主要用于支持文档处理应用。1974年，出现了标准通用标记语言（Standard Generalized Markup Language,SGML），并且从1978年到1980年它不断被国际标准化组织修改和发展，以满足系统的独立性和国际交换的要求，这些要求使得用于处理那时已有的相异系统的广泛选择成为必要。

HTML是SGML的为了简单、设备独立再现和超链接的一个应用。相反，XML是SGML的一个配置，设计于90年代后期，它给SGML提供在World Web环境中的可扩展能力。XML不完全象SGML，它不允许选择自由，它只支持为Web作出的选择，因此它就易于学习和使用。

SGML允许内容和表示的分离。HTML在同一个框架中混合了内容、表示和处理，这样使得文档难以使用和维护。XML和与它有关的标准鼓励内容（作为抽象元素类型）、表示（作为格式对象 —— 一组特定的元素类型）和处理（作为样式表）相分离，每部分都可以独立的发展，不需要在一个统一的框架中折衷。

XML通过Web传送。数据使用一个XML文档的结构来表示。元素和属性提供有用的元数据 —— 关于数据的数据，并且用一种可以通过Web传送、可以被不同应用使用和可以被转化为满足不同需求的方式来组织数据。

1.1.3 为什么使用XML

XML作为SGML的一个配置，最初被设计通过分离内容和格式来面对大规模的电子出版业

China-pub.com
下载

的挑战。把一个文档的内容和它怎样被格式化分离开来，简化了开发和维护。具有不同专门技术的不同的人员（或团队）可以独立的操作从文档中抽取的关于格式、样式和美化的信息。

XML在数据交换中也是有用的，XML可以用来在Web上的应用之间或应用与用户之间交换数据。XML本身并不给终端用户提供表示，但是当与样式表（提供样式）一起使用时，容易浏览XML文档了。XML文档提供一种方法来获取数据、样式表（例如，可扩展的样式表语言，XSL）或层叠样式表（提供把XML转化为HTML或其他表示的机制，CSS），XSL和CSS控制XML在Web浏览器中的表示。

XML另一个关键的优势就是集成数据和文档的能力。大多数语言或者在表示严格的、绝对的内容和数据结构上设计得好一点，或者在表示灵活的、自由形式的文档文本上设计得好一些，但是XML在两方面都做的很好。它能获取科学的或金融的数据，也能格式化通过email发送的情书，或者用来发表一首诗。XML能表示数据和自由格式的文本，它也能在同一个文档上做到这两方面。详细的评论或完整的科学论文能够以一种广泛的、复杂的数据集来解释一块简单的数据，科学论文可以在文档中嵌套大量复杂的层次数据集。

XML用一种灵活的可扩展的表示来实现内容通信。因此，数据的描述可以反复的被优化为基础域的变化，这使得XML对具有一个复杂的，频繁遭受修订的组织的领域是特别有用的。一些早期的XML技术的采纳者包括一些科学领域，例如生物、化学和数学。存在于这些领域的基于XML的科学标准分别是BioML、GML和MathML。这样，非常复杂的领域可以使用XML很好的合理的表示出来。

与其他数据交换语言相比，XML的两个关键特性就是它具有相当的表现力和可扩展性。实际上，你可以在你选择的主题上表述任何你想表述的。它具有一致的语法，这使它很容易的被解析（并且为它编写解析器）。因此，

- XML可以获取在应用之间交换的信息的类型。
- XML文档可以被剪裁以适合特殊的需求。
- XML解析器和其他通用工具已经是可用的。

XML可以使用用于定义处理的样式表来表示Web页面上的内容。Web页面可以是静态的或动态的，例如在线目录。静态的Web页面中，内容和表示的分离可以简化开发，也可以减少页面的大小。动态的Web页面非常容易表示，因为内容可以随着需求改变，而样式表保持不变。当内容上具有同样的条件时，使用同样的样式表，可能会有非常不同的表示。例如，数据可能从一张大表变化成一个简单的实例，并且表示可以被定制反应不同的格式。

XML文档可以用多种形式发布：HTML、PDF、Postscript等等。虽然这本书主要着眼于使用XML来表示数据库应用数据，但是对于用户来说XML并不仅限于此，它还为用户提供了一个非常好的信息表示语言。

在几种情况下，你可以考虑使用XML。当你需要为用户和应用提供数据时，应该使用XML。相同的文档可以使用样式表来提供给用户，并被应用程序解析和操作。使用同一个文档比为用户和应用程序开发独立的表示要容易得多。

当你需要一个简单的用户界面时，建议使用XML。一个简单的样式表可以被定义来表示一个XML文档，并且支持用户对它的浏览。与其他的需要解析不同的数据格式并生成一个可浏览

需要全本请访问 www.ertongbook.com

China-pub.com 下载

应用程序的方法比较起来，创建一个简单样式表的工作是微不足道的。

当你需要在应用程序之间交换数据时，特别在数据结构发生变化时，建议使用 XML。有时候使用XML可能不是最好的选择，比如当数据总是一个简单的格式时，例如表格，那么使用一个比较简单的表示可能会更高效。如果因为某些原因希望在 XML中也具有一些可选的格式文档，那么文档可以被封装为一个简单的顶级元素和一个 CDATA部件，这就避免了文档的主体被 XML解析器解析。

当数据结构并不简单时，可能需要 XML来存储复杂的关系，许多复杂的关系可以在 XML中捕获到。如果关系非常复杂，或者没有被完全的详细的理解，那么在决定最适合用 XML表示的结构之前，可能需要花大量的精力来建立关系模型。这个主题将在第 2章详细的阐述。

如前所述，当你需要把数据库数据和发布的内容结合起来时， XML是特别有用的。你可以为数据和发布开发独立的表示，然后把它们结合成一个 XML文档。

还有几个使用 XML的商业原因。它被普遍使用的 Web浏览器所支持，这样就降低了传送的成本。XML —— 作为一个W3C（World Wide Web Consortium）的推荐技术 —— 对比较大的系统的开发具有一定级别的保证，因为较大的软件公司，例如 Microsoft、IBM、Sun和Oracle，都支持这个技术。这些原因对数据库来说是特别重要的，因为大的数据库系统一定非常昂贵，并且企业的重要部分可能依赖它们。

W3C在他们的Web站点上（www.w3c.org）提供了一系列的其他商业原因， XML将：

- 使国际化的媒体无关的电子出版成为可能。
- 允许工业制定平台无关协议，来交换数据，特别是电子商业数据。
- 以一种允许在收到数据之后进行自动处理的形式给用户代理发送信息。
- 使得开发软件来处理分布在 Web上的特殊信息变得更容易。
- 对人们来说，容易使用便宜的软件来处理数据。
- 在样式表的控制下，允许人们按照他们所希望的方式来显示信息。
- 更容易提供元数据 —— 关于信息的数据，来帮助人们查找信息，并帮助信息生产者和消费者互相查找。

通常，这些原因适用于 XML，它们也简化了 XML数据库在Web应用中的使用。

最后，另一个使用 XML的原因就是它能够用在数据库中。数据库可以把数据存储为 XML，并且／或者使用XML与其他应用进行交互。

1.2 数据库系统

一个数据库系统由一个数据库和周围的环境，包括软件、操作系统、硬件和使用它的用户所组成的。软件由一个 DBMS，它提供对数据库的访问，和其他与 DBMS进行交互的软件组成。其他软件包括终端用户工具、管理工具、报表生成器、备份软件等等。 DBMS和其他软件依赖操作系统和硬件才能正确运行。一个数据库系统的模式图如图 1-2所示。用户通常根据他们在与DBMS进行交互中的角色来建模。用户角色包括终端用户、数据库管理员（负责建立和维护系统）、操作员（负责执行备份和日常操作）和数据管理员（负责确保数据的一致性），其他的用户角色包括设计数据库结构的数据库开发者和编写与 DBMS交互的软件的应用程序员。这本书主要讨论数据库、 DBMS和其他软件。

China-pub.com
下载

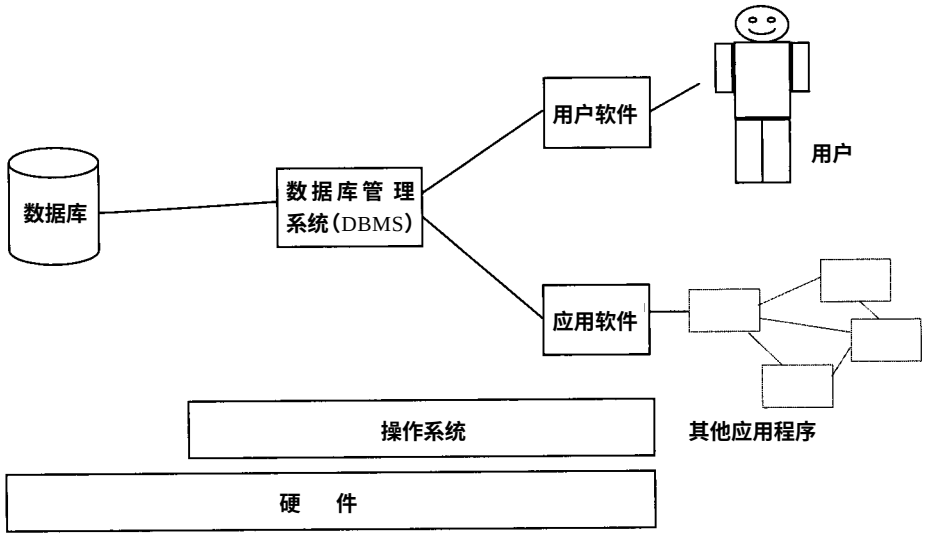


图1-2 数据库系统

1.2.1 什么是数据库

数据库是数据以一种能够保存持久，并且可以被操作的方式来保存的数据集合。保存持久意味着在任何人停止使用它并且在关机之后，数据库仍能继续保留。大多数文件都是可持久的，包括文本文件、电子数据表格和图象。

DBMS操作数据。在系统中它给用户提供插入、删除、检索和更新数据的操作来保护操作和数据的完整性。一个现代的DBMS也提供机制来确保事务的完整性，事务是组成一个完整有意义活动的操作集合。

DBMS把数据库用户从实现DBMS的硬件中隔离出来，并提供对数据的集中控制。隔离和控制保护了数据库中的数据，使它们免受用户和开发者引入的数据可能造成的干扰。遗憾的是，它有时也阻止了人们必要时对数据的灵活访问。很多时候我们需要在安全、速度、简洁性、并发性、灵活性、准确度、自然性和精确性多方面作权衡。

DBMS提供在数据库中存储新数据、从数据库中检索数据、改变数据，并且删除数据的机制。大多数文件，例如，电子数据表格中的那些文件，并不提供对数据库中的个别数据的访问。为了改变一个电子数据表格中的一个数据，一个应用程序读入整个电子数据表格，改变一个值，并且写回整个电子数据表格。这在电子数据表格具有一千个数据时可以工作得很好，但是如果它有十亿个数据，情况就大不一样了，而DBMS就具有修改个别元素的能力。

大多数DBMS最终把数据存储在一组文件中，文件是以一种可以检索或修改独立数据项（或者固定数量的数据项）而不需要访问数据的其他部分的方式创建的。为此有一个方法是创建一个文件，它的大小是操作系统文件块大小的整数倍。大多数操作系统可以创建文件并访问数据的独立块（例如1 024字节）。数据库的数据存储工具创建一个从每个数据项惟一标志符到数据库文件中的一个物理块的映射。有很多方法可以有效的创建这个映射。

China-pub.com 下载

一个比较简单的存储机制叫做平面文件数据库。在一个平面文件数据库中，数据被分为文本文件和目录结构，而不是文件块和映射结构。这样的文件通常比单个的块的大小要大得多，因此在被搜索时，它要慢一些。平面文件数据库的优点就是文本文件也可以被其他应用程序访问，例如文本编辑器、查找工具、浏览器和脚本语言。平面文件数据库已经用作为“伪关系型”的DBMS，其中每个表被存储成单个文件，并且提供一些（但不必是全部的）关系型操作。平面文件数据库的最大的局限性就是它几乎不可能查找一个文件格式，这个文件格式甚至不具备足够的直观性以备浏览。文件格式获取复杂的层次结构，可以被划分成文件大小的块，也可以与应用程序一起使用。现在，XML正致力于研究这些问题，并且平面文件数据库基于XML是可行的。

1.2.2 什么是XML数据库

XML数据库是一个XML文档的集合，这些文档是持久的并且是可以操作的。从历史观点上说，文档是为人类之间的交流而发展起来的。随着计算机的出现，文档也可以用于计算机与计算机之间交流，人与计算机之间的交流，或者计算机与人之间的交流。每个交流范例都有它自己的需求，例如美感、不明确许可范围、准确度和灵活性。

XML文档趋向于面向文档处理，或趋向于面向数据处理。面向文档处理的文档是使用XML来获取自然（人类）语言的那些文档，例如用户手册、静态的Web页面和市场手册都是自然语言。它们以复杂的或无规则的结构和混合内容为特征，并且它们的物理结构是很重要的。文档的处理侧重于给用户提供信息的最终表示；因此，它们可以被称作为面向表示的文档。面向数据处理的文档是XML主要用于数据传送的文档，这些文档包括销售定单、病人记录和科学数据。面向数据处理的文档的物理结构，例如元素的顺序，或者数据被存储为属性还是子元素，通常不是重要的。它们的特征是高度有序的结构，并且同时带有那些数据结构的多个副本。文档的处理通常侧重于它被应用程序的使用和交换，因此，它们可以被称作为面向消息的文档。面向文档处理的文档和面向数据处理的文档有时分别被称为面向文档的文档和面向数据的文档，但是这些术语是不明确的，因为每个文档总是包含数据。

这本书的一个目的就是给出一些深入的数据处理范例，以此来介绍数据库如何处理面向文档处理的文档中的多种复杂结构。面向文档处理的文档的结构可能比面向数据处理的文档的结构更灵活。例如，英语散文的结构比以制表符为分隔符的文件的结构具有更加丰富的语法。面向数据处理的文档趋向于拥有相同结构的许多副本，并且没有例外。XML为获取XML元素的嵌套层次提供了一种语言，它扩展了容易被传统的数据库掌握的规则。

面向数据处理或面向文档处理的XML文档之间的区别不大，并且有一些文档，例如带有描述性文本和数据的动态Web页面，可以用任一种方式表示，然而，文档上需要执行的操作是不一样的。在一个面向文档处理的文档中，需要执行的操作包括检索整个文档、搜索一个单词、查找先前的单词或后续的单词、修改一部分或重排一部分。在一个面向数据处理的文档中，需要执行的操作包括检索文档的一个指定部分、搜索元素和数据的一个特定的结合、修改或删除一个简单的元素或一块简单的数据，或者给文档添加一个新元素。一些需要执行的操作在这两类文档之间是都有的，例如决定元素结构的概况，或浏览文档以执行一些其他的操作。

China-pub.com
下载

XML数据库是一个面向数据处理的文档的集合。XML数据库提供的操作更加适合数据操作而不太适合文本处理。基于SGML的文本处理系统已非常的成熟，并且可能提供一些特性来操作XML文档。第三章将更加详细的描述这些操作是什么。实际上，在XML的DBMS中提供一些文本处理命令，对简化面向文档处理的文档或部分文档的操作也可能是有用的。

1.2.3 为什么使用XML数据库

如前面的章节所讨论的那样，XML在数据库与应用程序之间和在多对应用程序之间进行数据交换是很有用的。处理控制可以被表示为XML，或者被单独定义。XML不具有预定义机制来定义过程，但是有几种过程代码语言构建在XML之上，例如SOAP。

但是，在数据交换中，跟踪或对传送的数据作日志记录可能是很重要的，特别是在应用程序和／或用户属于不同企业的时候。被交换的数据可能需要被保留在一个存档文件中，并且偶尔被搜索。一个可搜索的XML仓库可以作为数据交换的中心部件，并且它也可以与其他使用XML的审计应用程序进行交互，或者获取Web服务的摘要信息。

XML仓库可能满足其他需求来保留、修改和搜索XML文档，例如将互连的XML文档应用于Web服务器、在一个安全的系统中保留重要信息，或者获取更清楚的表示在XML文档结构中信息。

XML的一个优点就是它的结构比关系型数据库中使用的关系更有表现力。一种关系就是一个无序元组集合，集合中每个元组有一个固定的字符集（或属性）。一个XML元素的结构要比关系的结构更有表现力，因为它可以包含其他元素、变化顺序和属性的数量，并且可以包含具有相同元素类型的多个元素，这使得它更加容易表示复杂的数据。

XML有一些类似于面向对象编程语言中的对象的特性，例如元素类型、命名属性和表示层次结构的能力。另外，XML元素具有一些对象本质上没有的特性，例如序列性，即表示为一个字符串的能力。同样对象具有一些XML没有的特性，例如方法和继承。对象和XML元素在封装上采取相对的立场，对象隐藏所有的内部结构，而XML显示的暴露所有的内部结构。所有的这些特性使得对象成为更好的编程基础，但是使得XML成为更好的数据表示的基础。因此，XML比对象更容易与数据交换一起使用，但是需要一种不同语言来操作应用程序中的数据。

如果数据库把数据保存为XML，那么在存储粒度上存在几个选择。XML数据可以被保存为一个文档、被分割为更小的部分并被保存为片段，或者被分解成单独的元素。即使数据不保存为XML，它也可以作为XML被输出，提供给其他应用程序或用户。通过解析文档，并把数据保存在数据库结构（例如关系）中，数据可以作为XML被添加到数据库中。

1.3 Web数据库

一个Web数据库系统就是一个通过Web发送持续数据的数据库系统。它是由数据库、用户、硬件、操作系统、网络协议、DBMS、Web服务器和其他软件组成的。一个Web数据库系统扩展了一个数据库系统的常规定义，它包括一个Web服务器和网络协议。一个Web数据库系统的模式图如图1-3所示。

China-pub.com 下载

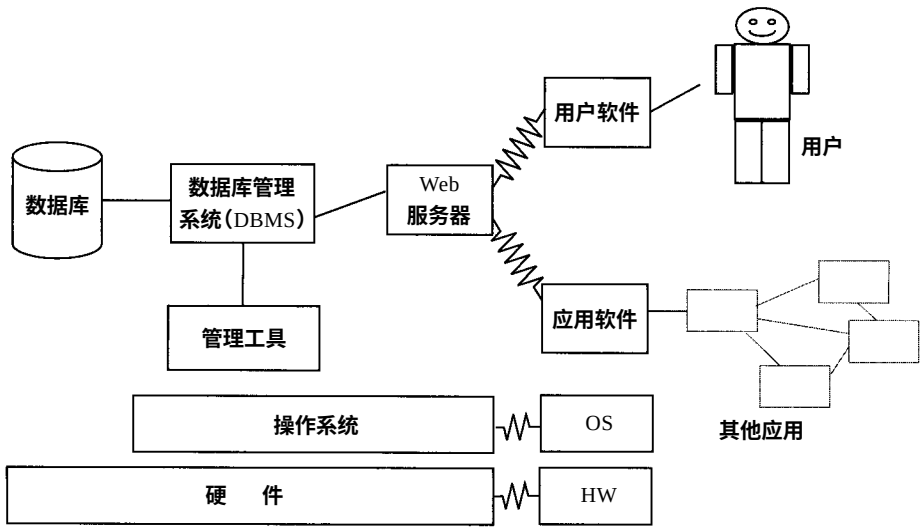


图1-3 Web数据库系统

在一个Web数据库系统中使用XML至少有三种方法。

1) 数据可以被一个把数据保存为 XML文档的平面文件数据库通过 Web服务器访问。 Web服务器不但能够提供对面向文档处理的 XML文档的访问，因为它已经被设计好了，而且它还能够提供对平面文件数据库的面向数据处理的有组织的 XML文档的访问。

2) 可以通过Web服务器到传统数据库的接口访问数据，并且在检索时将数据格式化为 XML。例如，来自关系型DBMS上的查询报表可以被格式化为 XML。

3) 可以通过Web服务器到XML的DBMS的接口访问数据。

1.3.1 平面文件数据库

XML平面文件数据库首先通过文件系统的目录结构，然后通过 XML文档的元素结构来提供对数据的层次访问。

平面文件数据库的优势是它很小，容易构建和维护，并且容易被其他工具访问。平面文件数据库的缺点是数据没有很好的被保护，并且不方便查询，而且平面文件数据库通常不提供并发控制、数据锁或恢复工具。在开发 XML平面文件数据库中，必须明确，什么信息是通过目录结构可以获取到，什么信息是通过 XML文档结构获取到。如下面的示例所说明那样。

鲁道夫圣诞树农场正在建设一个 Web站点，以允许邻近市区的顾客可以通过 Web订购圣诞树。不是十一月份砍树，然后把它们批发卖给城市内的圣诞树零售商，鲁道夫而是可以租用很多的卡车、雇佣很多的司机、并且在十一月底和十二月把刚刚砍的树发送到城市的不同地区。因为树是刚刚砍下的，并且直接从农场送出，所以顾客可能甚至愿意付比零售价格更高的费用。从商业角度看，最大化直销树的数量，同时将因没有批发出售圣诞树的损失最小化，这两者之间存在一个权衡。因为鲁道夫一家以前从来没有通过 Web出售圣诞树，所以他们没有原始数据，他们决定建立一个可调整的价格计划，根据需求、存货数量和圣诞节之前的天数来变动价格。这

China-pub.com
下载

就排除了拥有静态页面的可能性，因为价格信息必须变化。因为这是一个新的、高风险的、季节性的风险投资，所以他们不希望建设一个大的存货数据库应用来支持站点。

现在一个平面文件数据库已经被设计好了，如图 1-4所示，它包含顾客、产品和销售。在Web服务器上建立一个目录来容纳数据。下一步就是决定目录结构可以表示哪种关系以及 XML文档可以表示哪一种关系。一个极端方法就是把所有的信息都存放在目录结构中，同时文件只包含单一的信息片段，这是细粒度的平面文件数据库，因为数据被优化的存放为很多的小片段。一个细粒度数据库的数据库资源管理器如图 1-5所示。另一个极端方法就是拥有一个包含整个数据库的XML文档。这是一个非常粗粒度的平面文件数据库，因为数据被粗略的存放为一些大的片段，也就是一个片段。示例 1-3给出了只有一个XML文档的粗粒度数据库的例子。一个比较适度的方法就是让每个顾客、产品和事务成为一个单独的文件，这是一个中粒度的平面文件数据库。只有一个XML文档的中粒度数据库的例子如图 1-6所示。

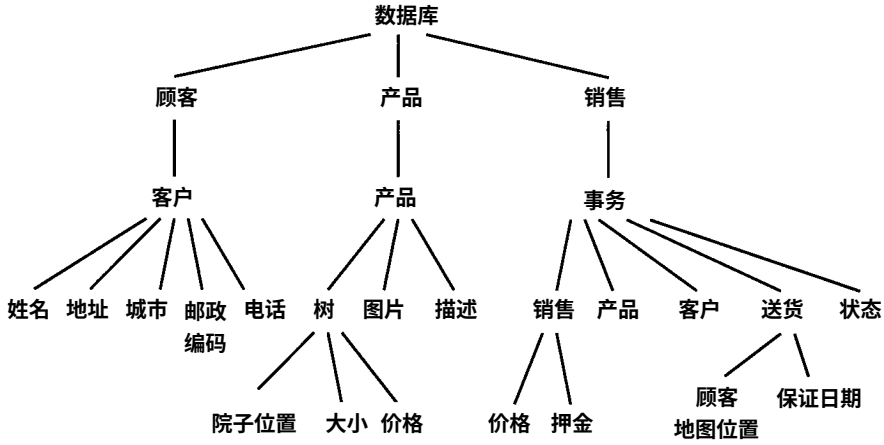


图1-4 鲁道夫圣诞树数据库

示例1-3 粗粒度的圣诞树数据库

```
<?xml version="1.0"?>
<treedb>
 <customers>
  <customer id="1">
   <name>Ebenezer Scrooge</name>
   <address>7 Main Street</address>
   <city>Anytown</city>
   <zip>12345</zip>
   <phone>6785559823</phone>
  </customer>
 </customers>
 <products>
  <product id="1">
   <tree>
    <yard_location>3F2</yard_location>
    <size>8</size>
    <price>59.95</price>
```

需要全本请访问 www.ertongbook.com

China-pub.com 下载

```
    </tree>
    <picture img="1.jpeg"/>
    <description>Beautiful fir tree with solid body and smooth shape.</
description>
   </product>
   <product id="2">
    <tree>
     <yard_location>2A6</yard_location>
     <size>6</size>
     <price>39.95</price>
    </tree>
    <picture img="2.jpeg"/>
    <description>Clean young tree with full coverage.</description>
   </product>
  </products>
  <sales>
   <transaction id="1" status="open">
    <sale>
     <price>39.95</price>
     <deposit>20.00</deposit>
    </sale>
    <customer id="1"/>
    <product id="2"/>
    <delivery>
     <customer_map_location>689A5</customer_map_location>
     <date_promised>1223</date_promised>
    </delivery>
   </transaction>
  </sales>
</treedb>
```

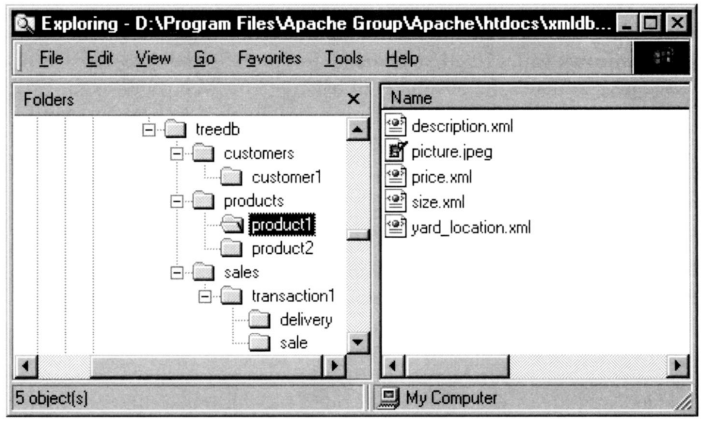


图1-5 细粒度的圣诞树数据库

建立订购系统，给顾客提供基于地址和邮政编码的送货日期。它也为“ available ”和“ sold ”目录中的每个产品组织 XML文件，并且在销售完成时把产品文件从“ available ”移动到“ sold ”中去。编写一个脚本在每天晚上执行事务，并且创建价格的摘要报表。编写另一个脚本把所有未售出产品的价格调整一定的百分比。最后，编写一个脚本生成在特定日期需要被送出的所有

China-pub.com
下载

树的一个报表，包含顾客姓名、地址和电话号码，并且按照圣诞树在院子中的位置和顾客地图的坐标来排序。



图1-6 中粒度的圣诞树数据库

1.3.2 关系型DBMS

到关系型DBMS的XML接口提供对健壮的数据库技术的访问，同时可以进行 XML传输。它在一个关系型数据库已经存在时特别有用。其主要的缺点是数据的表示受限于关系，这对某些领域来说可能不是问题。在关系和 XML元素可以表示什么之间存在着不匹配，因为 XML元素比关系具有更多的固有的灵活性。一种关系（或表中的一行）由固定数量的特性（或列）组成，并且每个特性是由一个数据项组成，而一个 XML元素可以包含任意数量的特性（作为子元素或属性），并且每个子元素可能包含不止一个数据项。在 XML中关系可以容易的被表示出来，但是从一个任意的XML文档中把数据导入到一个关系型数据库中可能是很难的。你也许需要修改关系模式，以便在数据结构中存在比平常还要多的变化。例如，为了获取具有相同元素类型的子元素的多次出现位置，可能有必要创建一个独立的关系来获取这个数据集并且把它链接到原有的关系上。这可能增加关系型数据库的复杂度，这时建议考虑使用 XML的DBMS。

使用关系型DBMS的一个优点是复杂的查询可以用 SQL来写，SQL是关系型DBMS的标准查询语言。报表可能被格式化为 XML，为一些基础数据提供多个视图。特别的，这在多个应用程序在数据库上需要各自的视图时是有用的。应用程序可以查询数据库得到一个结构中需要的特别信息，这个结构模仿应用程序的数据结构；应用程序也可以把数据作为 XML接收，并且可以轻易的把数据插入到数据结构或对象中去。例如，如果一个面向对象的跟踪应用程序需要来自复杂的顾客、产品和销售事务数据库中的数据，那么它可以执行一个复杂的查询来把数据检索为一个XML文档，这个文档非常接近的模拟了为应用程序开发的对象模型，即使它和数据是怎样存储到数据库中的大不一样。

使用关系型数据库的其他优点还包括它容易与其他关系型数据库的互操作能力、精确的事务管理和恢复机制以及大量工具对它的支持。在数据库必须被与其他关系型数据库进行互操作应用程序访问时，使用到关系数据库的XML接口比较合适。

开发一个到关系型数据库的XML接口将在第5章介绍。

1.3.3 XML DBMS

XML DBMS提供对XML文档和文档片段的直接访问，并且提供对文档和文档片段的交叉查

China-pub.com 下载

询。当从一个具有复杂层次关系的领域（例如科学数据库、大型组织和生产系统）中获取数据时使用一个XML DBMS特别合适。

一个XML DBMS可能从零开始建造、购买或者在另一个 DBMS基础上改造。从零开始建造一个XML DBMS时需要特别注意内存管理和对文档片段的高效访问，这将在第 9章讨论。当一个已有的XML DBMS满足项目的需求，并且没有遗留数据时，购买一个 XML DBMS可能是最好的解决方法。商业系统将在第6章中介绍。在另一个DBMS上构建一个XML DBMS更快捷，并且提供了许多用于存储的基础DBMS的优点，把已有的应用和遗留下来的数据集成起来也可能更容易一些。第4章介绍利用关系型或面向对象的DBMS构建一个XML DBMS用于存储，而通过Web接口访问XML DBMS系统将在第5章描述。

特别的，在另一个关系型 DBMS上构建一个XML DBMS有很多的优点。 XML OBMS中的数据与关系型数据库中的数据一样都受到强健的保护；这些数据可被已有的关系型数据库应用访问，包括查询工具、浏览器、资源管理器、数据引导程序和数据输出程序；在查询和视图中可以将这些数据和老系统中的遗留数据相集成。当一些数据可以被关系型表单获取，或者被采用下层关系型数据库的工具获取时，在已有的 XML DBMS 上构建新的XML DBMS也是有用的。

例如，如果鲁道夫圣诞树农场的 Web站点是一个辉煌的成功，然后第 2年他们希望购买财务软件来管理销售事务。财务软件来自一个平凡的、极端保守主义的公司，并且这个公司几年内根本没有升级过这个软件，但是它具备鲁道夫财务需求的所有特性，并且它现在还不贵。不幸的是，它采用的所有销售事务在一个关系型的数据库中是可用的。与其在一个关系型的数据库中开发一个自定义的圣诞树订购软件的解决方案，还不如把数据导入到一个构建在关系型 DBMS上的XML DBMS中。预先定义好的视图创建在关系型数据库中的 XML数据上，以满足财务软件的需求，并且脚本只需要很少的修改就能通过到 XML DBMS，而不是到平面文件的Web接口来访问XML数据。鲁道夫很高兴，甚至能够售出更多的圣诞树。

1.4 应用程序

XML是一种在应用程序之间交换信息的语言。多种应用程序在与用户的接近程度、所交换信息的类型、以及通常对数据的处理等方面大有不同。一些应用与终端用户非常接近，例如 Web浏览器；其他应用与用户离得较远，例如对客户购买产品作日志记录的应用程序。应用程序在它们对信息做何处理方面存在着不同。下列是用来刻画应用程序特征的问题：

- 信息是不是最初被设计供人阅读（例如，新闻故事）？
- 信息是不是最初设计用于计算或操作的（例如，报价、制造业传感器状态、实验结果）？
- 应用程序是否需要为将来的使用保留文本或数据？

有时数据交换是暂时的（例如，当前报价、当前存货、或每日新闻），在信息被生成和使用后，应用程序就不需要保留文档了。有些文本或数据必须保留（或持久），例如合同、销售事务、绑定报价单或订购收据，在这些情况下，文档或文档的某些部分需要保留。持久保留的文本通常在文档管理系统中获得，而持久保留的数据经常在 DBMS中获得。

这本书主要讲解要求数据持久保留的应用程序。有时数据具有一个结构，这个结构可以容

China-pub.com
下载

易的被提取或保存在关系型或面向对象的数据库中。而其他时候将很难提取数据，比如当结果数据库比原始XML要复杂得多的时候。一个获取XML数据的方法就是解析XML文档、把信息保存在关系型（或面向对象的）数据库中，然后必要时根据数据生成一个XML文档。另一种方法是把XML文档保存在一个XML数据库中，在XML数据库中XML文档作为整个文档或部分文档都是可用的。这里提到这两种方法，第2章中介绍如何描述数据，从而轻松地判定选择哪种方法。

1.5 资源

你可以找到许多资源，并从那里获得关于XML和XML数据库的更多信息。你将在第6章中找到关于XML数据库软件的更多信息。

1.5.1 杂志

很多Internet、Web开发、应用开发和Java开发杂志上都定期的刊登关于XML的文章，数据库杂志也刊登XML文章，偶然的在所有计算机杂志上也能找到这样的文章。专门的XML的杂志www.xmlmag.com同时具有印刷版和电子版。另一个专门的XML杂志是《XML Journal》(xml-journal.com)。

有一些XML应用程序开发者的邮件列表（和文摘），他们交流频繁，并且发表XML开发观点和XML标准化活动。它们不是为没有思想的人准备的。

- www.xml.org上的xml-dev邮件列表
- www.mulberrytech.com上的xsl-dev邮件列表

你可以在下面网站上找到联机文档：

- www.xml.com/pub —— 来自xml.com的XML文章

1.5.2 通用站点

一些通用站点如下：

- www.xml.com —— 大量的XML信息
- msdn.microsoft.com/xml/ —— 具有指南的Microsoft的XML主页
- www.ibm.com/developer/xml/ —— IBM开发者使用XML综述
- scentral.alphaworks.ibm.com/xmlguide.html —— 另一个IBM的XML站点
- www.ddj.com/topics/xml/ —— Dobb博士期刊的XML站点
- www.builder.com/Authoring/XmlSpot/ —— CNET有很多链接的主要的XML站点
- www.oasis-open.org/cover/xml.html —— Robin Cover的XML站点（或者www.oasis-open.org/cover/ 关于SGML/XML的站点）；您想知道的XML的任何信息
- www.smlinfo.com —— 关于XML的更多信息
- www.xml101.com —— 链接、文章和指南
- www.xmlpitstop.com——XML开发者的站点
- www.csclub.uwaterloo.ca/u/relander/XML/ —— XML资源的概况

China-pub.com 下载

- www.xmltimes.com —— XML世界的新闻和指南
- www.xmlbooks.com —— 出版的XML书籍列表

1.5.3 XML门户

- www.xmltree.com/xmlTree —— Web上XML内容的目录；给许多例子提供搜索功能
- www.schema.net —— 一个专门领域内的XML规范的目录
- www.xml.com/pub/Guide/Dirctories —— 一个基于XML的Web站点列表的目录
- www.xml.org —— 一个包含XML规范、模式和词汇的XML门户

1.5.4 XML工具

除了在通用的XML Web站点上列出的工具以外，下列站点有一些XML工具的特定信息

- www.xmlsoftware.com —— XML软件列表
- xml.apache.org —— Apache的XML站点

1.5.5 XSL

可以获得更多的关于XSL的信息的一些站点列举如下：

- www.xmlinfo.com/xsl/ —— XMLINFO站点上的可扩展的样式表语言XSL
- www.xml101.com/xsl/ —— XSL手册
- msdn.microsoft.com/xml/XSLGuide/ —— Microsoft的XSL站点
- www.xmlsoftware.com/xsl/ —— XMLSOFTWARE站点上的XSL编辑器工具
- www.w3.org/Style/XSL/ —— W3C XSL活动的主页
- www.w3.org/TR/xsl —— XSL规范
- www.dpawson.freeserve.co.uk/xsl/xslfaq.html —— XSL的FAQ
- www.mulberrytech.com/xsl/xsl-list/ —— Mublerry技术下的XSL列表 —— 关于XSL的开放论坛

1.5.6 W3C文档

- www.w3.org/XML/ —— XML主要的W3C站点
- www.w3.org/XML/Activity —— 提供一个W3C XML活动的综述
- www.w3.org/TR/REC-xml —— 官方的XML规范
- www.w3.org/TR——列出所有的W3C技术报告和出版物
- www.w3.org/DOM/Activity.html —— 提供一个W3C DOM的活动的概述
- www.w3.org/Style/XSL/ —— W3C XSL活动的主页

XML规范包括：

- www.w3.org/TR/xsl/ —— XSL规范
- www.w3.org/TR/xslt —— XSL转化规范（XSLT）
- www.w3.org/TR/xlink —— XML链接语言（XLink）
- www.w3.org/TR/xpath —— XML路径语言（XPath）

China-pub.com
下载

- www.w3.org/TR/xmlschema-1/ —— XML模式（XSDL）部分1结构
- www.w3.org/TR/xmlschema-2/ —— XML模式（XSDL）部分2数据类型

1.5.7 专门领域XML规范的示例

- www.w3.org/TR/PR-math/ —— 数学标记语言（MathML）
- www.voicexml.org —— 声音XML论坛
- www.xml-cml.org —— 化学标记语言

1.5.8 更多的XML信息

关于XML的更多信息可以在下列站点上找到：

- www.finetuning.com —— 一些可读的文章包括了其他地方没有的信息
- www.oasis-open.org/cover/elementsAndAttrs.html收集了在第3章中讨论的关于属性和子元素之间权衡的更多信息

需要全本请访问 www.ertongbook.com