案例背景

一、书店管理系统需求

1．概述

此系统用于书店的日常处理业务，包括供书目录、库存管理、采购管理、客户管理、售书系统、顾客订单管理、供应商管理及网上浏览等。

总体设计要求：书店内部日常处理业务通过 C/S 模式来实现。

2．系统功能需求

该系统的主要功能如下：

（1）供书目录及库存管理。供书目录管理及库存管理为书店的主要日常业务，应包括建立和更新供书目录，新书入库功能等。需要记录供书的信息有：书号、书名，作者（最多可包括 4 个作者，有序）、出版社、价格、关键字（最多 10 个）、存货量、供书商等，存放位置（可采用文字表示）。

（2）采购管理。采购管理为书店的主要日常业务之一，包括缺书登记管理及采购单管理。缺书登记，对于缺货的图书可直接进行缺书登记，包括书号、书名、出版社、供书商、数量、登记日期。采购单管理，根据缺书记录单可选择生成采购单，到货时对采购单作相应处理，并增加库存量，删除缺书记录。

（3）售书 POS 系统。店员通过该模块实现书籍零售功能，一个销售单可以包含多本书销售清单。同时需要计算金额以及打印销售单据。销售单需要存入数据库。销售单包括销售基本信息和销售清单。

销售基本信息包括销售单号、日期、金额、操作员、金额。

销售清单包括销售单号、书号、书名、数量、价格。

（4）销售单处理。经理可以实现销售单的统计和分析，报表打印等。

（5）供应商管理。供应商相关信息包括供应商基本信息、供应商供货信息，并且不同的供应商发布自己现有的书目信息。

（6）员工管理。可以维护员工信息表，包括添加、删除、修改员工信息。

3．数据结构设计

根据业务需要，以及系统的需要分析，设计相应的数据表。

（1）书籍信息表。

BookISBN:char(30) pk
BookName:varchar(30) not null
AuthorName: varchar(50) not null
Publish:varchar(30) not null
Price : money not null
Keyword : varchar(30) not null
Qunantity : int not null
ProviderID : char(10) not null fk
Place : varchar(30)

（2）供应商信息表。

ProviderID:char(10) pk
ProviderName : varchar(30) not null
Phone : varchar(20)
Address : varchar(30)

（3）员工信息表。

EmployeeID: char(10) pk
EmployeeName: varchar(30) not null
EmployeeLevel: (manager(经理),clerk(店员)) not null
Password: varchar(10) not null

（4）采购单信息表。

StockID:char(10)pk
BookISBN:char(30)fk notnull
BookName:varchar(30) not null
Publish:varchar(30) notnull
ProviderID:char(10)fk
Price:money
Quantity:int not null
RecordDate:date notnull
StockState:Boolean not null

（5）缺书记录单实体中的属性。

BookISBN:char(20) pk
BookName:varchar(30) not null
ProviderID:char(10) fk
Quantity:int not null
RecordDate:date notnull
Price:money
Publish:varchar(30)
StockTable:boolean not null

（6）销售清单。

SellNo:char(10) pk
BookISBN:char(20) pk
BookName:varchar(30) not null
BookPrice:money not null
BookQty:int not null
TotalPrice:money not null

（7）销售单基本信息。

```
SellNo:char(10) pk
SellDate:Date not null
EmployeeID:char(10) fk not null
TotalMoney:money not null
```

二、酒店管理系统需求

1．概述

传统的酒店管理方式极大地影响了工作流程效率和数据的正确性、完整性、安全性，已经逐渐落后于时代。利用软件管理系统代替手工的酒店管理，将会大大提高工作效率。

酒店在正常运营中需要对客房资源、顾客信息、结算信息进行管理，利用酒店管理系统及时了解各个环节中信息的变更，有利于提高管理效率。

2．系统目标

系统开发的总体任务是实现酒店各种信息的系统化、规范化和自动化管理。

3．系统功能分析

系统功能分析是在系统开发的总体任务的基础上完成。主要功能如下：

（1）有关客房标准的制定、标准信息的输入，包括标准编号、标准名称、房间面积、床位数量、住房单价、是否有空调、电视机、电话以及单独卫生间等。

（2）客房标准信息的修改、查询等。

（3）客房基本信息的输入，包括客房编号、客房类型、客房位置、客房单价、备注等。

（4）客房基本信息的查询、修改，包括客房编号、客房类型、客房位置、客房单价、备注等。

（5）剩余客房信息的查询。

（6）订房信息的输入，包括客房编号、客房类型、客房位置、客房单价、顾客姓名、顾客身份证号码、入住日期、折扣、备注信息等。

（7）订房信息的修改和查询，包括客房编号、客房类型、客房位置、客房单价、顾客姓名、顾客身份证号码、入住日期、折扣、备注信息等。

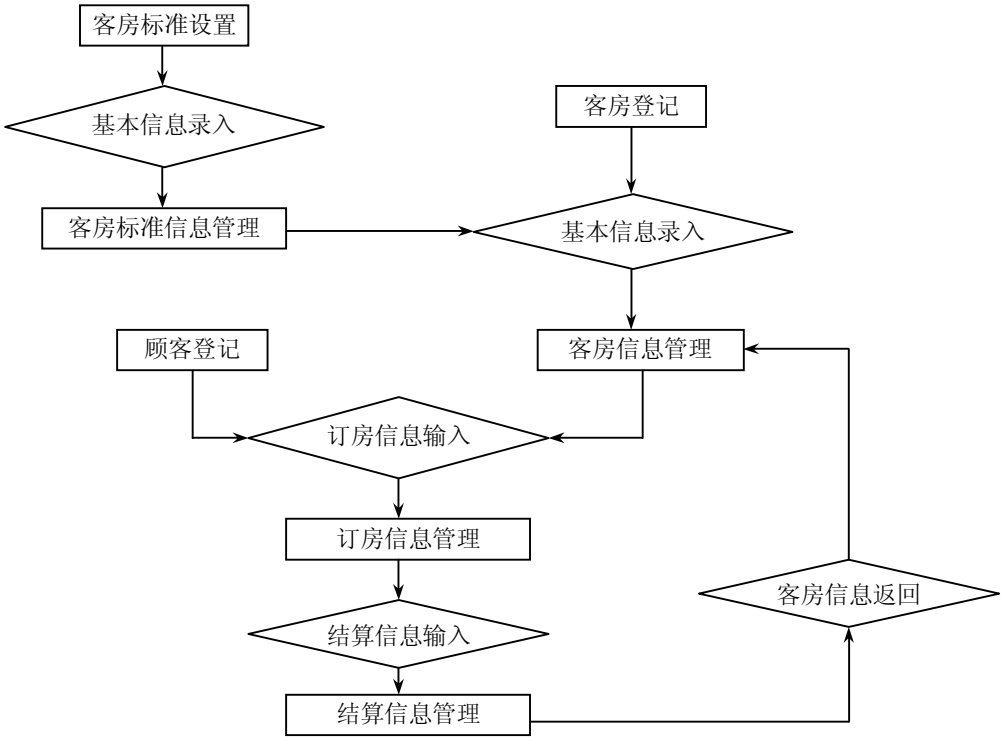
（8）结算信息的输入，包括客房编号、客房类型、客房位置、客房单价、顾客姓名、顾客身份证号码、入住日期、折扣、结算日期、备注信息等。

（9）结算信息的修改和查询，包括客房编号、客房类型、客房位置、客房单价、顾客姓名、顾客身份证号码、入住日期、折扣、结算日期、备注信息等。

（10）结算后账单打印。

（11）每日报表。

4．数据流程图



5．数据结构设计

针对一般酒店管理信息系统的需求，通过对酒店管理工作过程和数据流程的分析，设计如下数据项和数据结构。

（1）客房标准信息，包括的数据项有：标准编号、标准名称、房间面积、床位数量、住房单价、是否有空调、电视机、电话以及单独卫生间等。

```
TypeNo: char(20) pk
TypeName : varchar(20) not null
RoomArea: float notnull
BedNum : int not null
Price : money not null
AirCondition : Boolean not null
TV : Boolean not null
Telephone : Boolean not null
Toilet : Boolean not null
```

房间标准：豪华双人房、标准三人房、标准双人房、标准单人房。

（2）客房信息，包括的数据项有：客房编号、客房标准编号、客房状态、备注等。

```
RoomNo : char(10) pk
TypeNo : char(20) not null 外键
RoomState : varchar(1) not null
Desc : varchar(100)
```

客户状态有三种：在住、未清洁、已清洁。

（3）订房信息，包括的数据项有：订房编号、客房编号、顾客姓名、顾客身份证号码、入住日期、折扣、预交金额、备注等。

```
BookNo : char(10) pk
RoomNo : char(10) fk not null
Discount : int
GuestIDCardNo : char(25) not null
InnDate : date not null
GuestName : varchar(30) not null
BeforeHandBalance : money not null
Desc : varchar(100)
```

（4）结算信息，包括的数据项有：订房编号、客房编号、客房备注信息、顾客姓名、顾客身份证号码、入住日期、折扣、结算时间、备注等。

```
BookNo : char(10) pk
RoomNo : char(10) fk not null
Discount : int
GuestIDCardNo : char(25) not null
InnDate : date not null
CheckOutDate : date not null
GuestName : varchar(30) not null
Balance : money not null
Desc : varchar(100)
```

（5）用户信息，包含的数据项有：用户编号、用户姓名、用户密码、用户类型。

```
UserID : char(10) pk
UserName : varchar(30) not null
UserType : (manager\clerk) notnull
Password : varchar(10) not null
```

第1章 XML 基础知识

本章提要

- XML 的起源
- XML 的特点
- XML 文档的结构及组成要素

1.1 XML 概述

1.1.1 XML 出现的背景

在信息技术发达的今天，人们可以通过互联网从世界各地实时地接收和发送大量、最新的信息，给人们的工作和生活带来极大的方便和更多的乐趣。但在信息交换的过程中也存在着一个突出的问题，那就是数据格式的多样性，给信息的有效使用带来了障碍。因此，在信息时代如何以最便捷、最可靠、最有效的方式获取所需的信息是一个很大的困扰。人们期待着能够找到一种可以描述任何逻辑关系的数据格式来统一电子数据的存储，从而不再因为数据格式的不统一而苦恼和困惑。目前，只有 XML（eXtensible Markup Language，可扩展标记语言）可以满足人们对数据存储的诸多要求。

1.1.2 XML 的起源

说到 XML 的起源，需要先了解一下标准通用标记语言（SGML，Standard Generalized Markup Language）。SGML 最初是由 IBM 开发的一种用于排版的标记语言，称为 GML。经过若干年的发展，1984 年国际标准化协会（ISO）开始对此提案进行讨论，于 1986 年正式承认 SGML 为国际标准规范（ISO8879）。

SGML 实际上是一种通用的文档结构描述标记语言，主要用来定义文献模型的逻辑和物理类结构。一个 SGML 文件由三部分组成，即语法定义、文件类型定义（DTD，Definition Type Document）和文件实例。语法定义部分定义了文件类型定义和文件实例的语法结构；文件类型定义部分定义了文件实例的结构和组成结构的元素类型；文件实例是 SGML 语言程序的主体部分。

在 SGML 的实际使用中，每一个特定的 DTD 都定义了一类文件。因此，人们习惯上把具

有某一特定 DTD 的 SGML 语言称为某某标记语言。这样 SGML 就成为那些派生语言的元语言。

HTML 和 XML 都是源自 SGML 的标记语言。

1.1.3　HTML 简介

我们所熟知的网页就是由超文本标记语言（HTML，Hypertext Markup Language）所编写的。1989 年，欧洲物理量子实验室（CERN）的信息专家 TimBerners-Lees 发明了超文本链接语言, 使用此语言能轻松地将一个文件中的文字或图形连到其他的文件中去，这就是 HTML 的前身。1991 年，TimBerners-Lees 在 SGML 思想的启发下，定义了 HTML 语言的第一个规范，之后成为 W3C 组织为专门在互联网上发布信息而设计的标记语言规范。可以说，HTML 是 SGML 的一个实例，它的 DTD 作为标准被固定下来。

HTML 可用于显示网上任何类型的文档。HTML 是一种可被任何平台使用的标记语言，也就是说，它具有平台无关性。下面通过一个简单的 HTML 文档了解 HMTL 的语法结构。

```
<HTML>
    <HEAD>
        <TITLE>HTML 文档</TITLE>
    </HEAD>
    <BODY>
        <P>
            <FONT  COLOR=green  SIZE=4>欢迎您 </FONT> </BR>
            <FONT  COLOR=red  SIZE=6>进入万维网世界！ </FONT>
        </P>
    </BODY>
</HTML>
```

HTML 文档包含文本和标签。文本用于将数据转换成文档，而标签用于区分文档的不同类型的内容，即标示网页中各种 HTML 元素，以及控制它们的显示状态，但其本身在网页中是不会显示出来的。标签的结构以尖括号“<>”开始，以尖括号“</>”结束，尖括号中是标签名。

HTML 文档由两个部分组成：文件头和主体。文件头部分包含在<HEAD>、</HEAD>标签对之间。文件头的内容是关于当前网页的基本信息，显示在网页的标题栏中。而主体部分包含在<BODY>、</BODY>标签对之间，是 HTML 文档的主要内容，如需要显示的文字、表格、图像等信息，因此，需要在网页中显示的信息应在主体部分中定义。

上例中，主体部分显示的是一个段落元素，包含在<P>、</P>标签对之间；在段落中显示了两行文字，定义了字体元素，包含在<FONT>、</FONT>标签对之间。显示的网页效果如图 1-1 所示。

在 HTML 出现后的短短几年里，HTML 语言发展很快，迄今为止它已历经了 HTML1.0、HTML2.0 和 HTML3.0、HTML4.0 等多个版本，同时 DHTML（动态）、VHTML（虚拟）、SHTML 等也飞速发展起来。HTML 以简单精练的语法、极易掌握的通用性与易学性，使 Web 网页可以亲近于每一个普通人，互联网也因此得以普及和发展，呈现出今日的辉煌。

图 1-1　上例在 IE 浏览器中显示的效果

但是，随着电子商务、电子图书、远程多媒体信息访问等新技术领域的逐步成熟，HTML 本身所具有的缺陷日益明显，表现在以下几个方面：

（1）由于 HTML 的 DTD 是固定不变的，用户不能在原有标记定义的基础上进行扩充。

（2）HTML 是静态的，需要通过脚本编程和访问服务器，实现与用户的交互性操作。

（3）对于不同的浏览器，显示效果会有较大的差异。

（4）处理图形、图像、音频、视频等多媒体数据的能力较弱。

（5）对双字节或多国文字的支持，如中文信息在不同的平台上，格式不同。

正是由于 HTML 的不足，近年来，人们开始研究对 HTML 语言的改进，以寻求一种更加理想的制作 Web 页面的语言，也取得了不少的成效。前面所提到的 DHTML，就是一种改进的 HTML——动态 HTML 标记语言，可以动态地显示 Web 页面；级联样式表（CSS，Cascading Style Sheet），为 Web 页面的显示提供丰富的字体、色彩、显示模式；可扩展标记语言 XML，作为一种新的标记语言，比 HTML 更具有灵活性、高效性和规范性。

1.1.4　XML 简介

XML 是由万维网联盟（W3C，World Wide Web Consortium）于 1998 年 2 月发布的一种标准。它同样是 SGML 的一个简化子集，它将 SGML 的丰富功能与 HTML 的易用性整合到 Web 的应用中，以一种开放的、自我描述方式定义了数据结构。在描述数据内容的同时能突出对结构的描述，从而体现出数据之间的关系。这样所组织的数据对于应用程序和用户都是友好的、可操作的。

虽然与 HTML 一样，XML 也源自通用标记语言，但它却拥有 HTML 语言所缺乏的伸缩性与灵活性。一方面，XML 不再像 HTML 一样有着一成不变的格式，实际上，它成为一种定义语言，即用户可以定义有意义的、任意数目的标记来描述文件中的任何数据元素，从而突破 HTML 固定标记集合的约束，使文件的内容更丰富、更复杂、更容易组成一个完整的信息体系。另一方面，由于 XML 文档中能标记任意多的信息，因此，对于信息量大的数据文件，可实现为不同的用户提供不同部分的信息。如 XML 格式的考试数据文件，包含考题、监考需知、考试答案等内容，可为考生提供考题；为监考教师提供监考需知；为评卷教师提供试题答案。

可以将 XML 的特点归纳为以下几点：

（1）XML 文档设计简单，清晰易读。XML 的设计规范简单明了，由若干规则组成，使用这些规则可以创建标记语言，并通过解析程序处理所有新创建的标记语言（关于解析程序，在后面的章节将详细讲述）。由 XML 创建的标记语言，具有成对出现、编码标准统一的特点，是任何人都能读出和写入的世界语。

（2）具有开放性。XML 是一种开放式标准，它的基础是经过验证的标准技术，并针对网络做到最佳化。许多 IT 业界著名的公司，都积极协助 W3C 制定、完善 XML 标准的工作，使之对系统和浏览器的兼容性更好。

（3）具有高效性、可扩充性。用户可以自定义任意多个标签，也可与他人共享，其可延伸性大。对于 XML 文档，用户可选择多种方式处理，如可以直接在浏览器中显示，也可以转交到其他程序做进一步的处理；利用 DTD 可以灵活地共享 XML 文档中的数据，不同用户通过使用相同的 DTD 进行数据交换，还可以利用 DTD 验证用户自己的数据或接受到的数据的有效性。通过有意义的标签搜索指定的内容，可以提高搜索的效率，减轻浏览器的负担。

（4）标准国际化。一般的文档，如 HTML 网页中，通常是由一种语言编写的，若阅读文档的软件不能识别该种文字时，则无法读取文档中的内容。而 XML 使用新的编码标准，这种标准可以支持世界上大多数文字编写的文本，凡是能阅读 XML 文档的软件，就可以处理不同种语言字符的任意组合，这使得 XML 文档可以跨越国界、文化疆界，在不同的计算机系统之间畅通无阻地进行信息交换。

尽管 XML 具备了作为通用数据描述语言的优秀特质，但是 XML 毕竟只是一种数据描述技术，而不是一种编程语言，因而，XML 还需要结合一些相关技术，才能达到用户所希望的显示内容和效果。这些技术包括：

（1）文档类型定义（DTD），用于定义显示数据的结构和类型。

（2）级联样式表（CSS），用于定义显示或打印需要的排版信息，可以为每个 XML 文档元素指定排版信息。

（3）可扩展样式语言（XSL，eXtensibleStyle Sheet Language），是一种来描述如何显示 XML 文档内容的机制。与级联样式表（CSS）不同的是，XSL 提供了更为丰富、灵活的显示机制。XSL 包括两部分：一个用来转换 XML 文档的方法；一个用来格式化 XML 文档的方法。

（4）可扩展链接语言（XLL，eXtensible Link Language），是 XML 的链接语言，它与 HTML 中的链接类似，但功能更强大。使用 XLL，可以支持多方向的链接，并且可以存在于对象层级，而不仅仅是页面层级。

XML 是基于文本的标记语言，它通过使用有意义的标签以结构化的格式存储数据，这种格式可被任何一种计算机系统所解释。因此，XML 是一种跨平台、硬件和软件独立的标记语言，可以在不同类型的系统间传输结构化的数据，成为各种应用程序的通用数据交换格式。目前，XML 作为引人注目的核心技术，不仅应用于银行数据交换、证券公司对上市公司情况的数据统计、图书馆对图书的查询检索、企事业文件档案管理等领域，而且开始应用于电子商务、电子图书、移动设备应用等许多其他新的领域。

电子商务作为一种利用电子手段通过互联网进行商务活动的交易方式，在互联网上传输或交换数据是必不可少的一个环节。随着电子商务的发展，从最初通过浏览形式单一的网页购物，到购物个性化、新的电子商务模式出现，要求计算机不仅仅是实现网页的呈现，还需要能理解数据的语义，并且能够将数据和表现分离开来。这是侧重表现网页的 HTML 所无法胜任

的，而 XML 弥补了 HTML 的缺陷，成为电子商务中的核心技术。

今日的互联网以其覆盖面广、信息海量等优势，已成为继报刊、电台、电视台之后的一种新型媒体。在 1998 年 5 月举行的联合国新闻委员会年会上，互联网这一新型媒体被正式冠以“第四媒体”的称号。电子图书的出版最初是以 HTML 技术为信息的主要发布手段，但由于 HTML 本身的缺陷，无法满足同一本电子图书制作不同版本的需求。XML 技术具有数据与表现分离的特点，通过利用相关技术，可以实现一次图书内容制作，多个版本出版。

手机可以说是现代人们必备的通信工具。而今的手机不仅是一种通信工具，还被赋予了更多的内涵，如短信、上网等。在 1997 年 6 月，Phone.com 联合了 Nokia、Ericsson、Motorola 建立了 WAP 论坛，旨在利用已有的互联网技术和标准，为移动设备连接互联网建立全球性的统一规范。在 1998 年 5 月，推出了 WAP 规范 1.0 版，并于 1999 年 11 月发布最新的 1.2 版。WAP 规范包括 WAP 编程模型、无线置标语言 WML、WAP 网关、微浏览器规范、轻量级协议栈、无线电话应用（WTA）框架几个组件。其中 WML 是利用 XML 定义的专为手持设备的标记语言。关于 WML 将在后面的学习中进一步介绍。

XML 现在已被公认为一种拥有强大生命力的下一代网络标记语言。作为数据描述语言，XML 是非常优秀的，但并不是所有的电子数据转换到 XML 都是最有效的。例如 XML 的文本表现手法、标记的符号化等会导致 XML 数据比二进制表现方法数据量增加，尤其当数据量很大时，会成为很大的问题。也就是说，使用 XML 技术应根据具体需求，充分发挥 XML 的优势。

1.2 编写 XML 的要素

1.2.1 完整 XML 文档的组成

XML 文档是纯文本文件，其扩展名为 XML。在学习编写 XML 文档之前，先介绍一下编辑 XML 文档的工具。一般来说，XML 编辑工具分为通用编辑工具和专用编辑工具。通用编辑工具是指编辑文本文件的工具，如微软的记事本（Notepad）；而专用编辑工具是专门用于编辑 XML 文档的工具软件，微软的.NET 2003、Wattle 公司的 XML writer、ALTOVA GMBH 的 XMLSPY 等。下面对于通用编辑与专用编辑工具的使用，分别进行介绍。

（1）通用编辑工具。记事本是用于编写纯文本文件的编辑器。利用记事本创建 XML 文件，先创建一个新文本文件，写入 XML 文件的代码，然后保存。在“保存”对话框的“文件名”栏中输入“文件名.xml”，并用双引号括起来。

（2）专用编辑工具。微软的.NET 2003、Wattle 公司的 XML writer、ALTOVA GMBH 的 XMLSPY，这几种工具都提供可视化的编辑界面，除了编辑 XML 文件之外，还可以编辑 DTD、XSL、CSS、HTML 等纯文本文件，并且提供 XML 文档的有效性验证和浏览器功能。XML writer 软件可从 http://www.xmlwriter.net/download.shtml 下载，XMLSPY 软件可从 http://www.xmlspy.com/download.html下载。

在本书中采用 XMLSPY 工具软件进行 XML 文档的编辑、有效性验证以及验证后的浏览处理。

下面通过两个简单的例子来介绍编写 XML 文档的基本知识。

程序清单 1-1　（显示一个用户资料）

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
    <name>jone</name>
    <sex>famle </sex>
    <phone>426377382</phone>
</users>
```

以浏览器方式打开，显示结果如图 1-2 所示。

图 1-2　显示结果

图 1-2 中所显示的 XML 文档，以一种树型结构显示用户资料，内容为用户名：jone，性别：famle，电话：426377382，这类似于一般用户资料表中的一行内容。

文档中的第一行语句：<?xml version="1.0" encoding="UTF-8">，其中 version 指明了 XML 的版本号，encoding 指明了 XML 要处理的编码方案。而每个标签均为用户资料某项内容的名称，标签对间是该项内容的值。

通常存放数据的信息表中不止一条记录。下面来看另一个稍复杂些的例子。

表 1-1 是存放 DVD 信息的数据表，利用 XML 文档对该表中的数据进行描述，显示 DVD 信息表。代码如下：

表 1-1　DVD 信息表

DVD 名称	发行公司	代理商行	价格
春天	天籁之声音像公司	流行风	26
古筝曲	白天鹅音像公司	音乐吧，民乐坊	30
巴赫	北京音像公司	音乐吧	68

程序清单 1-2　（Music.xml）

```
<?xml version="1.0" encoding="GB2312"?>
<!DOCTYPE musiclibrary SYSTEM "Music.dtd">
<!--DVD information in Music.xml-->
```

```
<musiclibrary>
    <music>
        <title>春天</title>
        <publisher>天籁之声音像公司</publisher>
        <shop>流行风</shop>
        <price pricetype="RMB">26.00</price>
    </music>
    <music>
        <title>古筝曲</title>
        <publisher>白天鹅音像公司</publisher>
        <shop>音乐吧</shop>
        <shop>民乐坊</shop>
        <price pricetype="RMB">30.00</price>
    </music>
    <music>
        <title>巴赫</title>
        <publisher>北京音像公司</publisher>
        <shop>音乐吧</shop>
        <price pricetype="RMB">68.00</price>
    </music>
</musiclibrary>
```

Music.xml 文档的 DTD 文件代码如下：

程序清单 1-3 （Music.dtd）

```
<?xml version="1.0" encoding="UTF-8">
<!ELEMENT musiclibrary (music*)>
<!ELEMENT music (title,publisher,shop+,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT shop (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ATTLIST price pricetype (RMB|DOLLAR) 'RMB'>
```

将 Music.xml 和 Music.dtd 放在同一文件夹中，在浏览器中打开 Music.xml，运行结果如图 1-3 所示。

图 1-3　显示结果

DTD 文件用于定义 XML 文档中的标记，定义 XML 文档中的元素（如 title）等内容，以及文档中所显示数据的结构。上例中 DTD 的<!ELEMENT ……>定义了 Music.xml 中的元素 title、publisher、shop、price 以及这些元素排列的方式等。

如果希望显示的页面更加美观，还可以通过级联样式表（CSS）来简单地控制显示的效果。创建一个 CSS 文件 Music.css，代码如下：

程序清单 1-4　（Music.css）

```
<?xml version="1.0" encoding="UTF-8" ?>
music{
   display: block;
   font-weight: bold;
   font-size: 10pt;
     }
title{
   display: block;
   font-weight: bold;
   font-size: 10pt;
   color: red;
      }
publisher{
   display: block;
   font-weight: bold;
   font-size: 30pt;
   color: green;
}
```

修改 Music.xml 文件，在 Music.xml 文件中的第一行语句<?xml version="1.0" encoding="GB2312">后，加一条语句：

```
<?xml-stylesheet type="text/css" href="Music.css" ?>
```

保存该 XML 文档。再次在浏览器中打开 Music.xml，显示的效果如图 1-4 所示。

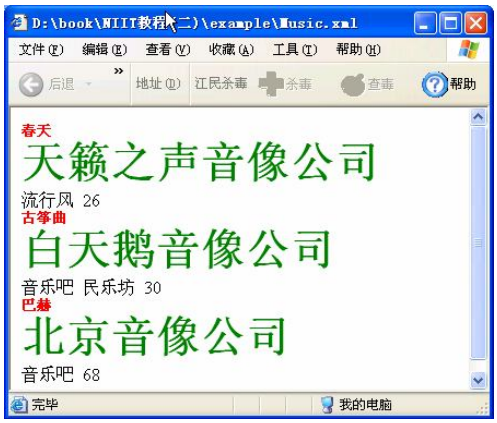


图 1-4　显示结果

从前面的例子可以了解到，一个完整的 XML 文件，应有三个组成部分：XML 文件本身、控制 XML 内容的文件、控制 XML 显示的文件。并且这三个部分是相对独立存在的，一方面

体现 XML 的内容与显示是分离的，另一方面无论是内容还是显示控制，都可以为多个 XML 文档所共享，而且它们的改变也互不影响。

XML 文档的结构可用表 1-2 来说明。

表 1-2　XML 文档结构

组成部分	语句表述	说明
序码	<?xml version="1.0" encoding="GB2312"?>	XML 文档声明
	<!DOCTYPE 根元素名 SYSTEM　DTD 文件名>	与之关联的 DTD 文档声明
	<?xml-stylesheet type="text/css" href=CSS 文件名 ?>	用于连接 CSS 文档的处理指令
	<!--注释内容-->	注释
数据正文	<根元素>……</根元素>	有仅有一个
	<子元素 属性=属性值> 子元素值 </子元素>	包含在根元素内的元素，至少有一个

一个完整的 XML 文档，除了在结构上要与表 1-2 相符外，还应该是一个具有良好规范的 XML 文档和有效的 XML 文档。具有良好规范的 XML 文档是指遵循 XML 语法规则；而有效的 XML 文档是指符合 DTD 对元素或属性的定义。

1.2.2　实例：编写一个简单的 XML 文档

问题

书店管理系统的库存数据存储在总店的数据库服务器，各个分店都可以通过网络连接到该服务器，并都可获得一致的库存数据。由于各个分店分布在不同的地区，所采用的系统平台或硬件也不同。因此，需要将库存数据存储在一个独立于硬件和软件的数据文件中。

库存书信息包括 ISBN、书名、作者（最多为 4 人）、出版社、价格、关键字、供书商、库存量、库存位置。

实现效果（如图 1-5 所示）

分析

（1）确定存储库存数据的文件格式。
（2）理解数据存储文档的结构。
（3）创建一个 XML 文档存储库存数据。

解决

1．确定存储库存数据的文件格式

各个分店所用的系统不同，对数据文件的格式要求也不同；而访问数据库服务器的硬件也不同（如有采用台式机，也有采用手机设备）。这要求存储数据的文件必须独立于硬件和软

件，XML 文档提供了独立于系统平台和设备的数据存储格式，无论采用何种系统平台或设备同样都可以访问数据。因此，采用 XML 文档可以满足系统的要求。

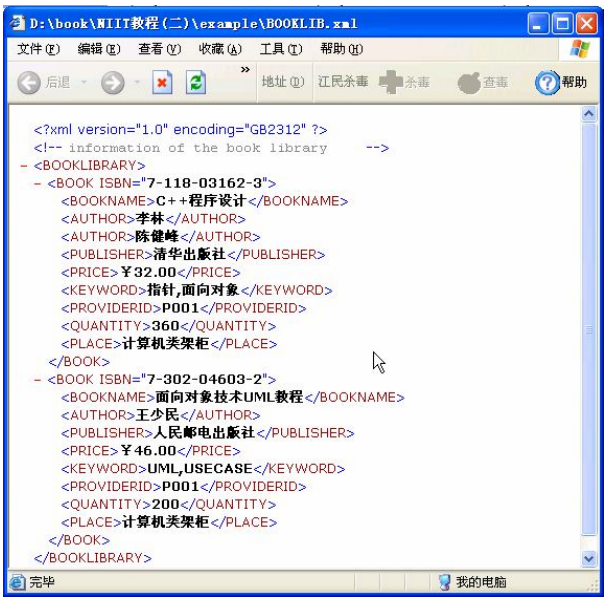


图 1-5　实现效果

2．理解数据存储文档的结构

通过对 1.2 节的学习，对于 XML 文档的结构有了一定了解。但要编写一个 XML 文档存储数据，还需要对每个组成部分更详细地了解。

（1）处理指令（PI，Processing Instruction）。PI 是以“<?”开始，以“?>”结束。通常 PI 出现在文档的序码中（如例 1-2 中，XML 文档的第一句），但也可以在 XML 文档的其他任何地方出现。它的作用是向使用 XML 文档的程序传递某种信息。例如：

```
<?xml version="1.0" encoding="GB2312" standalone="yes"?>
```

一个 XML 文档通常是由 XML 声明开始的。上述 XML 声明表示当前 XML 文档的版本为 1.0，采用的编码标准是中文 GB2312 字符集，不需要引用其他文件。

encoding 属性指定 XML 文档的编码所采用语言的字符集。常用的有 Unicode UTF-8、Unicode UTF-16、GB2312 等，当省略该属性时，表示使用 Unicode 字符编码。 UTF-8 是用于创建英语的标准字符集，它是用 8 位信息表示每个字符，支持使用 ASCII 码字符的计算机系统。UTF-16 是由 16 位信息表示每个字符，用于创建需要用 16 位表示的语言，如日语。一旦设置 encoding 属性，浏览器根据该属性可对 XML 文档进行适当解释。standalone 属性为可选项，表示 XML 文档是否要引用其他文件。该属性取值为“yes”或“no”，当取值为“yes”时，表示不引用其他文件，取值为“no”时，表示可能引用其他文件。缺省值为“no”。

注意　声明中 version，encoding 必须使用小写字母。

例如：

```
<?xml-stylesheet type="text/css" href="Music.css" ?>
```

xml-stylesheet 表示传递给应用程序的目标名称，通常以“XML”、“xml”开始。type 属性用于指定该文件的类型，这里是将 CSS 类型的文档与 XML 文档相关联。href 属性用于指明文件所在的路径和名称，本例中 Music.css 与 XML 文档在同一文件夹中。该语句的含义是用名为 Music.css 的 CSS 文档来格式化输出 XML 文档。

（2）标签。在 XML 文档中，标签是用于标记信息的。标签由尖括号组成，通常成对出现，开始标签中只包含标签名，结束标签包含“/”和标签名。标签名可以是用户定义或 HTML 预定义的标签。开始和结束标签中的标签名应完全相同，因为 XML 文档对大小写敏感。例如：

```
<NAME>Smith</NAME>
```

上例就是一个标签。NAME 是标签名，是用户创建的标签，存储了一个客户的名字 Smith。若存储多个客户的名字，可用多个 NAME 标签来存储，即：

```
<NAME>Smith</NAME>
<NAME>JOHN</NAME>
<NAME>BILL</NAME>
……
```

（3）注释。注释是用于对文件中的某段代码加以标注。例 1-2 中语句<!--DVD information in Music.xml-->，表明该文件是处理关于 DVD 信息的 XML 文档，为整个 XML 文档的内容进行了标注。注释只是对代码的说明，它不是必不可少的内容，但可增加程序的可读性，程序执行时将被忽略。

注释语句的语法：

```
<!-- 注释内容    -->
```

若注释有多行内容，需要用多个注释语句来表示。

（4）元素和属性。元素是用来在 XML 中定义和描述数据的最基本单位。在 XML 中用标签表示元素，通过定义有意义的元素，提高程序的可读性和信息的查询效率。例如：

```
<BOOKNAME>C++程序设计</BOOKNAME>
```

其中“BOOKNAME”是元素名，“C++程序设计”是元素的值。

XML 文档包含一个或多个元素，它是 XML 文档的主要内容。由表 1-2 可知，XML 文档的数据正文包括根元素、子元素和属性。一个 XML 文档有且仅有一个根元素，可有一个或多个子元素，每个元素可以拥有自己的属性，但属性不是必需的。

属性用于提供所声明元素的有关信息。属性表示为一个“名—值”对。一个元素可以拥有一个或多个属性，并且属性值可以是强制的或可选择的。

一般元素的表示方法：

```
<元素名>内容</元素名>
```

具有属性元素的表示方法：

```
<元素名 属性名="属性值">内容</元素名>
```

例如：

```
<musiclibrary>
    <music type="民乐">春天</music>
    <music type="古典">月光</music>
    </musiclibrary>
```

其中 musiclibrary 是根元素，它是唯一的，并包含了所有其他元素。music 是子元素，“春

天”是元素 music 包含的内容，type 是 music 元素的属性，“民乐”是属性的一个值。

XML 文档中的信息可以用元素表示，也可以用属性方式表示。因此，创建一个 XML 文档前，首先需要确定 XML 文档的结构，哪些信息用元素表示，哪些信息用属性表示。决定使用元素还是属性表示某个信息，没有强制的规定，但可以依据以下几个原则来考虑。

- 元素表示一个可定义的信息，属性用于表示对该元素的进一步限定。如上例中 music 是元素，type 是 music 的一个属性，使用 type 对 music 的类型加以限定。
- 如果必须将信息显示出来，则使用元素表示。通常，元素属性用于表示无形的、抽象的性质。如产品的 ID，可用属性表示。
- 假如信息内容需要经常更新，最好使用元素表示。这是由于使用 XML 编辑工具，元素比属性更易编辑。如书店每日有许多次交易，书的库存量会随之频繁变更，可以将书的库存量作为元素来描述。
- 如果需要经常使用某种信息作为查询的索引项，可以将该信息表示为一个属性。因为 XML 处理器检索文档中的属性值内容比检索元素内容要容易些。如查询产品详细资料时，需要按产品的 ID 来检索，就可以将产品 ID 表示为属性。

（5）内容。XML 文档中元素所表示的数据被称为元素的内容。如<music>春天</music>语句，music 是元素名，“春天”为 music 元素的内容。再看一个例子：

```
<music>
    The detail description of DVD
    <title>spring</title>
    <publisher>天籁之声音像公司</publisher>
    <shop>
        <shopname>流行风</shopname>
        <address>中山路 120 号</address>
      </shop>
    <price pricetype="RMB">26.00</price>
</music>
```

分析上述代码，可列出每个元素所包含的内容，如表 1-3 所示。

表 1-3　元素所包含的内容

元素名	内容	内容类型
music	title，publisher，shop，price	元素
	The detail description of DVD	文本
title	spring	字符
publisher	天籁之声音像公司	中文字符
shop	shopname，address	元素
price	26.00	数据

由上例可知，XML 中元素所包含的内容可以是其他元素，也可以是字符或数据，还可以是元素和文本的混合。

对于实例中的问题，可以创建一个根元素 BOOKLIBRARY，将库存书的 ISBN 作为属性来表示，其他信息用元素来表示。库存书的 XML 文档结构描述如图 1-6 所示。

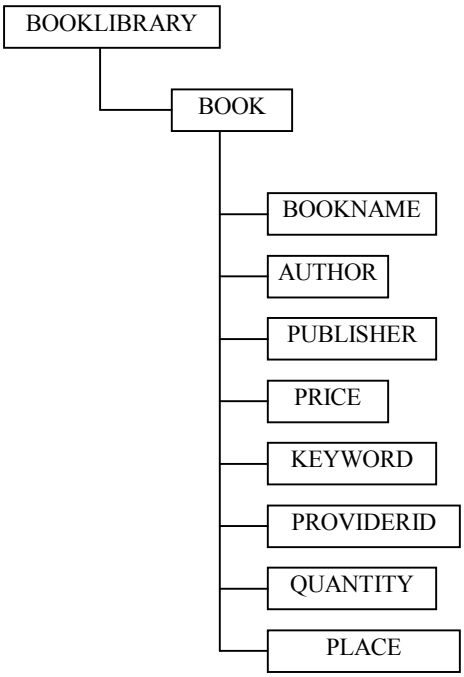


图 1-6　存储书的库存信息的 XML 结构

3．创建一个 XML 文档存储库存数据

一旦设计好 XML 文档的结构，就可以开始创建 XML 文档了。为了使编写的 XML 文档符合 W3C 所规定的 XML 标准及语法规则，还需要了解创建良好规范的 XML 文档应遵循的规则。

（1）文档中有且只有一个根元素，它包含所有其他元素。非根元素之间应为并列关系，或是完全包含关系，不可交叉。如：

```
<shop>
    <shopname>流行风</shopname>
    <address>中山路 120 号</address>
</shop>
```

上例的表示方式是正确的。元素 shop 包含元素为 shopname、address，shopname 与 address 之间为并列关系。如：

```
<shop><shopname>流行风</shop></shopname>
```

上述语句的表示方式是错误的，其中的两个元素之间为交叉关系，在 XML 中这是不允许的。

（2）标签一般是成对出现的，除了空标签外，每个开始标签必须有相应的结束标签，并且开始标签与结束标签必须匹配，否则会导致错误。例如：

```
<NAME>李明
<NAME>周华
```

上述语句中，只有开始标签，没在结束标签，XML 处理器无法判断第二个 NAME 标签是开始还是结束标签，造成语法上的模糊，这是不允许的。例如：

```
<NAME>李明</Name>
```

这条语句的错误在于，开始与结束标签不匹配，由于 XML 对大小写敏感，这样写将被认为是开始与结束标签名不同而导致出错。

（3）上面提到的空标签，是指一种只有开始标签无结束标签，以开始标签开始，以“/”结束。它不包含任何元素，但可以包含属性。例如：

```
<product prodid="p001">
```

上述语句表示 product 是一个空元素，仅包含一个属性 prodid，属性值为 p001。

（4）当出现元素嵌套时，标签必须按照一定的顺序嵌套。嵌套中在前的元素，开始标签在前，而结束标签在后。例如：

```
<shop>
        <shopname>流行风</shopname>
        <address>中山路 120 号</address>
</shop>
```

其中元素 shop 包含元素 shopname、address，shop 的开始标签在 shopname 的开始标签之前，而 shop 的结束标签则在 shopname 的结束标签之后。即，最里层最后开始，最先结束。

（5）属性的赋值必须使用双引号。例如：

```
<product prodid="p001">
```

根据以上 XML 的编写规范，可以创建一个存储书的 XML 文档 BOOK.XML。

程序清单 1-5

```
<?xml version="1.0" encoding="GB2312"?>
<!--information of the book library  -->
<BOOKLIBRARY>
   <BOOK ISBN="7-118-03162-3 ">
       <BOOKNAME>C++程序设计</BOOKNAME>
       <AUTHOR>李林</AUTHOR>
        <AUTHOR>陈健峰</AUTHOR>
       <PUBLISHER>清华出版社</PUBLISHER>
       <PRICE>￥32.00</PRICE>
       <KEYWORD>指针,面向对象</KEYWORD>
       <PROVIDERID>P001</PROVIDERID>
       <QUANTITY>360</QUANTITY>
       <PLACE>计算机类架柜</PLACE>
    </BOOK>
   <BOOK ISBN="7-302-04603-2 ">
       <BOOKNAME>面向对象技术 UML 教程</BOOKNAME>
       <AUTHOR>王少民</AUTHOR>
       <PUBLISHER>人民邮电出版社</PUBLISHER>
       <PRICE>￥46.00</PRICE>
       <KEYWORD>UML,USECASE</KEYWORD>
       <PROVIDERID>P001</PROVIDERID>
       <QUANTITY>200</QUANTITY>
       <PLACE>计算机类架柜</PLACE>
    </BOOK>
</BOOKLIBRARY>
```

单元练习

酒店管理系统中，需要存储客房信息。要求对于酒店的各部门（如客房部、办公室等）、客户都可以通过不同的系统或网络了解客房的有关信息，客房信息数据应存储在独立于软件和硬件的数据格式中。客房信息包括客房编号、客房标准（房间类型）、客房位置、客房单价、备注。

1.3 XML 的相关技术

1.3.1 CSS

样式表定义如何显示 HTML 元素，就像 HTML 3.2 的字体标签和颜色属性所起的作用那样。样式通常保存在外部的.css 文件中。通过仅仅编辑一个简单的 CSS 文档，外部样式表使你有能力同时改变站点中所有页面的布局和外观。由于允许同时控制多重页面的样式和布局，CSS 可以称得上 Web 设计领域的一个突破。作为网站开发者，你能够为每个 HTML 元素定义样式，并将之应用于你希望的任意多的页面中。如需进行全局变换，只需简单地改变样式，然后网站中的所有元素均会自动更新。

1.3.2 DTD 和 XSD

文档类型定义（DTD）可定义合法的 XML 文档构建模块。它使用一系列合法的元素来定义文档的结构。DTD 可被成行地声明于 XML 文档中，也可作为一个外部引用。通过 DTD，每一个 XML 文件均可携带一个有关其自身格式的描述。通过 DTD，独立的团体可一致地使用某个标准的 DTD 来交换数据。同时应用程序也可使用某个标准的 DTD 来验证从外部接收到的数据。

XML Schema 是基于 XML 的 DTD 替代者。它描述了 XML 文档的结构，也称作 XML Schema 定义（XML Schema Definition，XSD）。XML Schema 的作用是定义 XML 文档的合法构建模块，类似 DTD。但是一般认为 XML Schema 很快会在大部分网络应用程序中取代 DTD。原因如下：

- XML Schema 可针对未来的需求进行扩展。
- XML Schema 更完善，功能更强大。
- XML Schema 基于 XML 编写。
- XML Schema 支持数据类型。
- XML Schema 支持名称空间。

1.3.3　XSLT 和 XPath

XSLT 称为 XSL 转换，是 XSL 中最重要的部分。通常用于将一种 XML 文档转换为另外一种 XML 文档，或者可被浏览器识别的其他类型的文档，如 HTML 和 XHTML。通常，XSLT 是通过把每个 XML 元素转换为(X)HTML 元素来完成这项工作的。通过 XSLT，可以从输出文件添加或移除元素和属性。也可重新排列元素，执行测试并决定隐藏或显示哪个元素，等等。

XPath 使用路径表达式在 XML 文档中进行导航，通过使用路径表达式来选取 XML 文档中的节点或节点集。这些路径表达式和我们在常规的文件系统中看到的表达式非常相似。它还包含一个标准函数库，库中含有超过 100 个内建的函数。

本章练习

1-1　XML 与 HTML 的区别是什么?XML 的主要优点有哪些?

1-2　XML 文档的结构是怎样的？

1-3　将下表中的信息用 XML 文档表示。

课程名称	教课教师	班级	学时数
计算机导论	陈迹	计算机 041	60
C++程序设计	林清	计算机 032,计算机 031	80

1-4　分析下面的 XML 文档片段，说明 STUDENT、CONTRACT、PHONE 三个元素的内容及内容类型。

```
<STUDENT Sno="20040112">
      The detail description of STUDENT
        <NAME>刘成</NAME>
        <AGE>20</AGE>
        <ADDRESS>中山路 100 号</ADDRESS>
<CONTRACT>
    <PHONE>89012343</PHONE>
    <MOBILEPHONE>135012340987</MOBILEPHONE>
    <EMAIL>lucheng@163.com</EMAIL>
</CONTRACT>
</STUDENT>
```