

程序开发人员进阶与提高黑皮书

全新推出
精心制作

Borland®

Delphi 数据库

应用开发技术与实例

张大年 廖智勇 刘剑锋 等编著 文清源工作室 审校

```
procedure TfmMain.StartWordApp;  
begin  
    if not Assigned (WordApp) then  
    try  
        // 创建WordApplication对象  
        WordApp:=CreateCOMObject (CLASS_WordApplication, 'Word.Application'  
        with WordApp do  
        begin  
            // 指定COM对象的属性  
            Caption:= 'This copy of WinWord is launched by '+Forms.Application.ExeName;  
            Visible:=True;  
        end;  
        ShowMsg ('启动 Word Application 成功');  
    except  
        ShowError ('启动 Word Application 失败');  
    end;  
end;  
procedure TfmMain.btnStartWordAppClick (Sender: TObject);  
begin  
    StartWordApp;
```



本书含光盘



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



498

TP311.56
Z310

Borland Delphi 数据库应用 开发技术与实例

张大年 廖智勇 刘剑锋 等编著

文清源工作室 审校

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书结合丰富的程序实例全面深入地探讨了 Delphi 6.0 数据库编程技术,论述了数据库应用程序设计的常用技巧和方法,重点剖析了 Delphi 数据库的高级编程技术。同时介绍了 Delphi 6.0 数据库方面的新功能和特点,反映了当前数据库技术的主流和最新发展情况。

本书力图使用简洁明了、通俗易懂的语言来阐述内容,在一些比较难于理解的地方力求讲明讲透,同时辅以大量的实例。另外,本书作者将以往编程过程中积累的经验、技巧也穿插在书中,以求对读者有所帮助。

作为比较专门的参考资料,本书主要面向中高级读者,已经基本掌握 Delphi 入门技术而想进一步提高数据库程序设计技术的读者也可以参考本书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Borland Delphi 数据库应用开发技术与实例

作 者: 张大年 廖智勇 刘剑锋 等

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×960 1/16 **印张:** 33.75 **字数:** 752 千字

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书 号: ISBN 7-900637-78-8

印 数: 0001~5000

定 价: 55.00 元

前 言

1. Delphi 6.0 及其数据库开发

Borland 公司的 Delphi 开发工具一直是软件开发的一把利器，它的完全可视化的开发界面，丰富的控件支持，与 Windows 系统的高度融合以及可以说是速度最快的编译器等，一直为人所称道。即使不能说是 Windows 下最好的开发工具，它也是 Windows 下最快的开发工具。现在 Delphi 已经推出了 6.0 版本，它不仅继承了上述优点和特性，而且紧紧抓住当前软件业技术发展的大趋势，在 Web 应用上有了极大的加强和发展，例如对 XML/XSL、SOAP、和 Web Service 这些技术，Delphi 6.0 都有非常良好的支持。对于跨平台这一软件技术发展趋势，Delphi 6.0 也没有忽略，它拥有跨平台控件库 CLX，通过它程序员可以在 Delphi 6.0 中开发 Linux 下的应用程序。至于其他方面，如新控件的增加，编译器的改善，IDE 中编辑功能的增强等，读者在亲身使用中就会有更深的体会，这里就不一一细述了。

上面介绍了 Delphi 6.0 的总体概况，本书讲述的是 Delphi 在数据库程序开发中的应用。众所周知，数据库程序开发一直是 Delphi 的强项，Delphi 支持 Access 到 Oracle 各种数据库类型，开发者可通过 BDE、ADO 和文件存取的方式与数据库相连并开发出不同的数据库应用程序。BDE 是 Borland 公司自主开发的数据存储引擎，它的特点是速度快，支持类型广泛，易于使用；而通过 ADO 几乎可以和现在所有的数据库类型连接，文件方式存储则是实现移动办公（公文包模式）的最佳途径。此外，还可利用 MIDAS 技术开发出基于三层以上的、分布计算的应用程序，而这种结构正是当今数据库应用程序的主流。

Delphi 6.0 在数据库方面同样有了很大的变化，诸如新增加的 dbExpress 数据存取方式（它同样适用于 Linux 环境），对 XML 支持的增强，新的字段类型等。这些在本书中都有详细介绍。但作者觉得最有意义的发展变化还在于 Web 服务功能的增强。在 Delphi 6.0 中 MIDAS 已经改名为 Datasnap 了，结合其他数据控件方面的变化，可以看出软件技术呈现出从 MIDAS 向基于 Web 的架构演变这一趋势。相信将来的软件会以 Web 服务的方式通过网络发布，而用户也不用再购买、安装、使用并自己维护软件，而是通过网络以执行文件的方式使用这些 Web 服务。这些已经超出了本书的范围，读者如果想深入了解，应该参考其他相关书籍和资料。

2. 关于本书

本书的目标定位于已对 Delphi 一般编程技术有相当了解的中高级读者。但在撰写的过

程中,力图使用简洁明了、通俗易懂的语言来阐述内容,在一些比较难于理解的地方力求讲明讲透,所以对于刚入门和正准备学习 Delphi 的初级读者,只要配合阅读其他关于 Delphi 的通用书籍,本书也是完全适用的。

本书是以功能为讲解单元的,基本上一个单元介绍一个功能模块。所以本书不以介绍各种控件为主要内容,而是在讲述功能的同时自然而然地将所涉及到的控件予以介绍。而在介绍具体的控件时,也不是控件属性、方法和事件的简单罗列,而是根据功能来介绍这些属性、方法和事件。这样就可以做到有的放矢,将最常用、最重要的东西呈现给读者,从而使读者可以更好更快地学到有用的东西。

本书的另一个特点就是有丰富的实例,这些实例基本覆盖了所介绍的功能点,通过这些实例,读者的学习将会容易一些,掌握内容的速度也会提高很多。在本书的配套光盘上能找到所有示例程序的源代码和编译好的可执行程序。

另外,在本书的编写过程中,作者将以往编程过程中积累的经验,使用过的技巧和所遇到的常见问题用“注意”、“说明”等字样和较深的背景颜色标识出来,这些内容对读者在实际的编程过程中是会有相应帮助的。

3. 本书导读

本书共分为 14 章,从基础开始、循序渐进地介绍了 Delphi 数据库程序开发,下面分别介绍各章的主要内容。

第 1 章 数据库基础知识

本章主要介绍了数据库及数据库应用程序的基本概念和知识。这是进行数据库程序开发所必需的基础。分为“数据库基本概念”、“数据模型与数据表”、“数据库管理系统”和“数据库应用程序”这四个小节。

第 2 章 Delphi 数据库应用程序开发概述

本章共有四个小节:“Delphi 6.0 数据库功能新特点”、“Delphi 数据库应用程序的体系结构”、“基于文件的数据库应用程序”和“一个基本 Delphi 数据库程序实例”。其中第一小节介绍了 Delphi 6.0 在数据库程序开发方面的一些新功能和新特点。第二、三小节讲述了 Delphi 数据库应用程序的体系结构,最后一小节用一个简单的 Delphi 数据库程序实例生动地展示了开发 Delphi 数据库应用程序的一般步骤。

第 3 章 Delphi 6.0 数据库工具

本章介绍了 Delphi 6.0 提供的一些常用的数据库工具如 Database Desktop、Data Pump、SQL Explorer、SQL Monitor、BDE 等的使用方法。有如下小节:“桌面数据库平台”、“数据库引擎”、“数据库资源管理器”、“SQL 监视器”、“数据字典”、“数据模块”、“Datapump 工具”和“数据库向导”。

第 4 章 连接数据库

本章分成“数据集控件”、“BDE 数据库连接”和“BDE 会话期”三小节详细地介绍

了如何使用 BDE 来建立和管理数据库的连接。第一小节讲述了连接数据库所使用的核心——数据集控件。第二、三小节则介绍了任何使用 TDataBase、TSession 控件管理数据库的连接和会话期。

第 5 章 字段对象

本章主要介绍了 Delphi 6.0 中的字段对象及其操作,包括字段对象的编辑和使用,动态字段和永久字段的区别以及字段纠错限制,最后还讲述了 Delphi 6.0 支持的几种对象字段。共分为以下七个小节:“字段控件”、“动态字段对象和永久字段对象”、“编辑字段对象”、“设置永久字段对象”、“使用字段对象”、“字段纠错限制”和“对象字段”。

第 6 章 数据查询和显示

本章分成“TTable 控件及其操作”、“TQuery 控件及 SQL 查询”和“显示数据”三个小节介绍了数据的查询和显示。本章的内容是数据库程序开发中最琐碎但也是所占比例最大的一部分。查询部分介绍了 TTable 和 TQuery 控件,显示部分介绍了数据控制控件。本章介绍的控件都是最基本、最常用的,读者应该全面掌握。

第 7 章 存储过程

本章结合实例讲解了如何创建、维护和使用存储过程。本章是为了提高初学者应用程序设计的能力而引入的,它分成“存储过程控件”、“创建、编辑存储过程”、“使用存储过程”、“存储过程的参数”和“存储过程的一个实例”五个小节。

第 8 章 数据报表制作

在数据库应用程序中,经常需要对数据进行统计、汇总等工作,此时需要使用报表。所以,报表在数据库应用程序中占有很重要的地位。本章结合实例介绍了使用 QuickReport 工具制作各种报表的方法。共分成“报表简介”、“报表控件”和“制作报表”三个小节。

第 9 章 TeeChart 图表制作

和上一章一样,本章所介绍的利用 TeeChart 控件制作数据图表的内容也是 Delphi 数据库程序的重要组成部分。分成“TeeChart 使用概述”、“创建数据库图表”、“创建 QuickReport 图表”、“图表的预览,打印和导出”、“使用 TeeChart 向导创建图表”和“综合实例”六个小节介绍了制作和使用 TeeChart 图表的方法,对 TeeChart 系列控件和诸如 TChart、TSeries 一些常用对象属性和方法也有讲述,并辅以实例。

第 10 章 数据库决策分析

本章介绍了使用决策控件组 TDecision Cube 进行数据库决策分析的方法。主要讲述了各控件的属性、方法和使用,并附以实例。分为五个小节:“使用决策数据集”、“建立数据仓库”、“决策源”、“使用决策控件”和“一个综合例程”。

第 11 章 InterBase 数据库程序开发

InterBase 数据库是一种常用数据库类型,特别是在 Delphi 应用程序中经常使用。本章介绍了 InterBase 数据库应用程序的开发。共分两小节。第一小节“InterBase 数据库”介绍了 InterBase 数据库及其创建、使用和维护方法;第二小节“开发访问 InterBase 数据库程

序”介绍了 InterBase 数据库程序的开发, 主要包括数据查询、数据更新、数据库信息监视和事件响应以及相应的各种 InterBase 控件的使用方法。

第 12 章 开发 dbExpress 程序

dbExpress 是 Delphi 6.0 推出的跨平台的数据库访问机制。本章讲述了 dbExpress 数据库应用程序的开发, 包括基本程序开发、数据库更新和元数据信息查询, 共分为三小节: “开发 dbExpress 程序的基本步骤”、“数据库更新”和“数据库元数据信息”。

第 13 章 ADO 编程开发

ADO 应用程序是数据库程序开发中很重要的一部分, 这一章介绍了 ADO 应用程序开发, 对 ADO 机制、数据库连接、各种控件的使用方法都有详尽的介绍, 并提供了相应实例。本章内容分成“ADO 编程概述”、“ADO 数据库连接”、“使用 ADO 数据集”、“使用 TADOCommand 控件”和“TRDSCONNECTION 控件的使用”五个小节。

第 14 章 分布式多层结构体系

本章介绍了多层数据库应用程序的体系结构, 并介绍了利用 Delphi 6.0 开发多层数据库应用程序的方法和技巧。而基于 MIDAS 技术的多层分布计算的应用程序结构正是当今数据库应用程序的主流。这一章共分为四个小节: “分布式多层体系结构和技术概述”、“应用服务器”、“客户端应用程序”和“一个多层应用程序实例”。

另外, 本书所配光盘提供了书中所介绍的应用程序的所有代码, 供读者演试和借鉴。

4. 致谢

本书在创作过程中, 受到众多同仁的支持和帮助, 在此表示感谢。

本书之所以能够出版, 离不开清华大学出版社的大力支持和帮助, 在此代表此书的所有作者对清华大学出版社表示诚挚的感谢。

此外, 由于作者自身水平和客观条件的限制, 本书中难免会有错误和不足之处, 敬请广大读者谅解并欢迎批评指正。

作 者

2001 年 8 月

第 1 章 数据库基础知识

数据库应用开发是软件开发中的重要组成部分。而且很多非数据库应用程序，也要涉及数据存取和管理的问题。在阐述 Delphi 6.0 数据库开发之前，本章先介绍有关数据库的基础知识。

1.1 数据库基本概念

1.1.1 数据库系统与数据库

广义的数据库系统是指一个存储、维护和为应用系统提供数据的软件系统。它由三个部分组成：数据库管理系统（DBMS——它是专门负责组织和管理数据信息的程序）、数据库应用程序（它使我们能够获取、显示和更新由 DBMS 存储的数据）、数据库（按一定结构组织在一起的相关数据的集合）。而狭义的数据库系统是由一个或几个数据表格组成的，数据表格是由数据组成的。

1.1.2 本地数据库与远程数据库

数据库按所在的物理地址的不同可分为本地数据库和远程数据库两种。本地数据库位于本地计算机上，如 Paradox、dBASE、FoxPro、Access 数据库等；远程数据库指非本地的，位于远程计算机或服务器上的数据库，如 Oracle、Sybase、SQL Server、DB2 等。

本地数据库采用基于文件的机制访问数据库，因此本地数据库又叫基于文件的数据库。远程数据库采用 SQL（Structured Query Language 结构化查询语言）访问数据库，因此远程数据库又叫 SQL 服务器或是 RDBMS（Remote DataBase Management Systems 远程数据库管理系统）。

因为本地数据库位于本地计算机上，访问速度比访问远程数据库速度快。但它不可能支持多用户同时访问，数据库存储容量也没有远程数据库大。远程数据库提供了基于事物的多用户支持，适用于多个用户同时访问，数据存储容量也大得多，如果是分布式数据库，数据甚至可以分布存储在多个计算机上。

1.1.3 分布式数据库

分布式数据库是指由一个系统管理，分布在本地和远程计算机上的多个数据库组成的系统。常见的分布式数据库有 Oracle、SQL Server 等。

1.1.4 客户端/服务器应用结构

客户端/服务器结构（C/S 结构）是指数据库应用程序分为前台和后台两部分。前台任务运行客户端（用户计算机），执行用户输入和数据输出显示；后台运行在服务器上，用于执行对数据库的基本操作。

1.1.5 存储过程

存储过程（Stored Procedure）是一组由 SQL 语句组成的程序，存放在 SQL 数据库服务器上，用来执行特定的任务。使用存储过程可以加快程序的执行速度。

1.1.6 事务

数据库事务实际上是一组动作（对数据库的操作），这些动作必须在一个或几个数据表被提交之前成功地执行。如果有某个动作执行失败，所有的动作都将被撤消(Undo)。大部分本地数据库不支持事务，而 SQL 数据库和 ODBC 兼容的数据库本身就提供了处理事务的能力。Delphi 的 BDE 驱动程序也提供了有限的事务处理能力。

1.1.7 数据字典

数据字典提供了一个可自定义的不依赖于应用程序的存储区域，在这个区域中，可以创建扩展的字段属性集和描述数据的内容及外观。有时在设计期创建数据集时，就不需要手工设置数据显示格式，只要从数据字典中选择一个合适的属性集，让当前数据集中的数据共享它的属性，数据就可以直接用属性集里设计好的显示格式了。因而使用数据字典能保证数据具有一致的外观。

关于数据库的其他一些概念（如数据模型，SQL 语句等），在下面的小节里有更详细的论述。

1.2 数据模型与数据表

数据库中的数据是高度结构化的,不仅数据项之间存在着联系,记录之间也存在着联系。而数据模型就是指描述这种联系的数据结构的形式。不同的数据模型决定了不同类型的数据库。比较重要的数据模型有以下几种。

- ❖ 层次模型:用树形结构描述数据及其之间的联系;
- ❖ 网状模型:用网状结构描述数据及其之间的联系;
- ❖ 关系模型:用二维表结构描述数据及其之间的联系;
- ❖ 实体关系模型:用实体关系图描述数据及其之间的联系。

因为现在大多数数据库都是基于关系模型的,这里将主要介绍一下关系数据模型。

说明:

在本书中,除非特别声明,所提到的数据库都是关系数据库。

在关系模型中,数据信息是以二维表的结构组织的,这些二维表被称为关系表格,简称表(Table)。每张表描述的是事物某一方面的信息,它由表名、列名(表头)和若干行(数据)组成,如表1-1所示。

表 1-1 学籍信息

学号	姓名	所在系部	入学日期	入学成绩	籍贯
0001122	张平	经济管理系	2000 / 09	595	河北
0002094	李一凡	计算机科学系	2000 / 09	612	山西
0004101	王洪	通信工程系	2000 / 09	603	陕西
9904045	廖强	通信工程系	1999 / 09	620	广东
9901006	罗浩	经济管理系	1999 / 09	610	北京
9802079	陈得胜	计算机科学系	1998 / 09	632	辽宁

说明:

当然,严格说来,上述表格只是一个完整的学籍信息表的一部分。但它并不影响我们对问题的分析。

在上表中,表名为学籍信息,列名就是学号、姓名、所在系部等,一个学生的情况就是一行(数据)。下面结合表介绍一下关系模型数据库的几个基本概念。

- ❖ 列(Field):也被称为字段、域或属性。每一列描述一类信息。在关系数据库中,列之间的顺序并不重要。但列名在一个表中必须是惟一的。

- ❖ 行 (Row): 也称元组 (Tuple) 或记录 (Record)。一行数据是由若干字段组成, 描述了一个对象的信息。而每个字段对应的是这个对象某一方面的性质或属性。上表的每一行就表示了一位同学的基本信息。
- ❖ 键 (Key): 也称码。在表中可以标识每一行的字段或字段的集合称为键。如在表 1-1 中, 学号就是一个键, 因为一个学号对应唯一的一行。通过学号, 就能找到确定的行, 而不会出现两个以上的行。同理, 姓名也是一个键 (排除同名的情况)。利用键字段能够高效地与其他数据表格建立链接。
- ❖ 值域 (Domain): 也就是字段的数据类型, 它是数据库管理系统中的基本信息类型。例如入学成绩的值域为 NUMBER。
- ❖ 索引 (Index): 为了加快访问数据库的速度, 许多数据表格都使用索引。

此外, 合理的数据表格还应满足以下几点要求。一是各列的信息要有一定联系; 其次表格中至少要有一列的记录是不重复的; 三是至少能与同数据库的另外一个数据表格链接; 最后是与数据库其他任意数据表格不要有多对多的链接。

1.3 数据库管理系统 (DBMS)

数据库管理系统 (DataBase Management System) 是用于定义、控制、管理和维护数据库的程序系统的, 是数据库系统的核心组成部分。它建立在操作系统的基础上, 对数据库进行统一的管理和控制。根据所基于的数据模型, DBMS 主要有四种类型: 文件管理系统、层次数据库系统、网状数据库系统和关系数据库系统, 当然, 目前关系数据库系统应用最为广泛。一般说来, DBMS 的主要功能有:

- ❖ 描述数据库 定义数据库的逻辑结构、存储结构、语义信息和保密要求以及其他各种数据库对象。
- ❖ 管理数据库 控制系统运行、数据存取和用户访问, 检验数据的安全与一致性, 执行数据检索、修改、插入、删除等操作。
- ❖ 建立维护数据库 初始数据装入, 数据库结构维护, 重建数据库, 故障恢复, 数据库性能监视, 记录工作日志。
- ❖ 数据通信 组织数据的传输。

从内容组成上讲, DBMS 一般包括以下三部分:

- ❖ 数据描述语言 (DDL) 及其翻译程序;
- ❖ 数据操纵/查询语言 (DML) 及其翻译程序;
- ❖ 管理程序。

除此之外, 现代的数据库系统, 为了提高数据库开发的效率和水平, 还提供了一些应用开发的辅助支持工具。如 ORACLE 的交互式命令语言 SQL PLUS, 报表生成器

SQLFORMS, SYBASE 的四代语言报表生成器 SQR 4GL 等。

1.4 数据库应用程序

DBMS 存储数据信息的目的是为用户提供数据信息服务, 而利用数据库应用程序能够与 DBMS 进行通信, 并访问 DBMS 中的数据, 它是沟通用户和 DBMS 的桥梁。用户正是通过数据库应用程序来访问、插入、修改、删除数据库中的数据。虽然一般的数据库管理系统自身也具有上述功能, 但这些使用功能命令要求用户必须熟悉所使用的数据库系统; 而且不灵活, 没有针对性, 实现一些特定功能很麻烦, 并且缺乏友好的界面。数据库应用程序屏蔽了复杂的 DBMS 具体细节, 它能根据用户的要求实现特定的功能, 并具有方便的接口, 易于操作的友好界面。传统上数据库应用程序是用一种或多种通用或专用的程序设计语言编写的, 但是近年来出现了多种面向用户的数据库应用程序开发工具, 这些工具可以简化使用 DBMS 的过程, 并且不需要专门编程。目前开发数据库应用程序的语言主要分为三大类型:

1. 过程化语言

如 Pascal、Basic 和 C 都是过程化语言, 这些语言可以通过应用程序接口 (API) 来创建数据库应用程序, API 函数调用扩展了语言的功能, 使之能访问数据库中的数据。过程化语言的特点是程序以应用的执行过程为中心组织结构, 每一段代码就是一个过程, 每个过程执行应用的某一部分的功能, 过程间有严格的顺序。如一个过程查询数据库中的数据, 而另一过程更新数据库中的数据, 然后不同的过程通过其他的用户界面过程 (例如菜单系统) 联系在一起, 并且在应用中的适当地方运行。

上述这些过程化语言通常被称为“第三代语言”(3GL)。还有一些过程化程序设计语言是某种特定的 DBMS 提供和专用的, 这些语言一般被称为“第四代语言”(4GL), 即数据库专用语言。常见的如 dBASE 语言, Paradox 数据库的 PAL 语言等。

2. 结构化查询语言 (SQL)

结构化查询语言 (Structure Query Language) 是基于关系模型的数据库查询语言, 已经为大多数数据库管理系统所应用。它是一种非过程化的程序语言, 也就是说, 程序只需告诉数据库系统做什么, 而不用告诉它怎么做。写出的语句可看做是给数据库的一个问题, 称为“查询”(Query), 针对这个查询, 从数据库中得到所需的结果。SQL 语句可以完成许多功能, 如建立数据库, 查询、插入、修改和删除数据, 控制数据存取, 确保数据库的

致和完整。下面是一个例子：

```
Select Name,No from Student where Total>600
```

这个查询是从数据库学籍表 Student 中将总分 (Total) 大于 600 的所有人选出来，并列出他们的学号 (No) 和姓名 (Name)。

SQL 没有任何界面处理或用户输入/输出的能力。它的主要目的是为了提供访问数据库的标准方法，而不管数据库应用的其余部分是用什么语言编写的，它既是为数据库的交互式查询而设计的（因此被称为动态 SQL），同时也可在过程化语言编写的数据库应用程序中使用（因此被称为嵌入式 SQL）。关于 SQL 在 Delphi 中的详细用法后面的章节有专门论述。

3. 面向对象的语言

开发数据库应用程序中，还可以使用目前常见的“面向对象程序设计”（OOP）语言，如 C++、Object Pascal 等，OOP 代表了一种完全不同的程序设计方法，在这种方法中，程序设计不再以应用的执行过程为中心，代码根据所描述的事物不同被划分成一个个的“对象”，每个对象都有自己的数据、属性和方法。应用的执行就是程序向这些对象发送消息，对象根据内部的属性方法对消息做出不同反应的过程。如程序执行时，用户的敲击键盘、移动鼠标就是向程序发送消息，程序把它转发给相应的对象，对象再做出应有的反应，来完成应用的执行。OOP 方法更符合人的思维习惯，它使同属一个对象的数据和算法紧密联系在一起，有利于代码的重用和软件的升级，从而大大提高了软件开发的效率。在数据库应用程序中使用 OOP 语言已经是主流了。

开发数据库应用程序使用的另一种语言是“宏”。宏语言不是一种完全的程序设计语言，它实际上是一段代码，它被输入到应用程序中，以便自动执行一定的任务。对于某个特定应用的高级语言，宏语言通常可以在低档 DBMS 软件中或数据库服务器的前端中找到。

最后，还有一种叫“Query-By-Example”（QBE，范例查询）的语言。严格地讲它也不是一种语言，它是面向用户提供了一个或多个空表的界面，这些空表对应于数据库中的表。用户可以在这些空表中选择需要查询的列，并在适当的列中填入条件从而定义查询的检索条件，然后 DBMS 就把 QBE 转换成相应的动作，以完成用户要求的查询任务。

1.5 本章小结

本章主要介绍了数据库及数据库应用程序的基本概念和知识。这是进行数据库程序开

发所必需的基础。本章首先介绍了数据库的一些基本概念，接下来是数据表和数据库管理系统，最后讲述了数据库应用程序开发的常用语言类型。

本章里读者应重点掌握数据库的基本概念，特别是要理解数据表的含义，因为以后所涉及的数据库应用程序其实都是针对数据表的。另外一个重点就是理解并掌握面向对象的编程语言。

第2章 Delphi 数据库应用程序开发概述

随着世纪之钟敲响 21 声，我们的脚步已经迈入信息社会的大门。在这信息化的社会里，人们对信息服务需求的范围越来越广，质量越来越高。工作和生活中要涉及到大量的信息查询与修改，信息存储和共享，这种情况导致数据库这门技术越来越重要并得以飞速发展。

数据库应用程序，顾名思义，是从数据库管理系统（DataBase Management System, DBMS）中获得数据并进行相应的处理的程序。从简单的数据浏览和查询、数据批处理程序到复杂的客户/服务器应用程序都属于数据库应用程序的范畴。不仅如此，平常的一些程序看起来不像传统的数据库应用程序，但是它会使用数据库来存储数据和信息。可以相信，数据库技术在现代社会中将会发挥越来越重要的作用。

Delphi 是 Inprise（以前的 Borland）公司推出的第四代编程语言。Delphi 在数据库编程方面具有强大的功能，在数据库软件开发工具里占有巨大的市场，从而，Delphi 本身也得到了快速的发展。至今，Delphi 已经发展到 6.0 版本。本章先介绍 Delphi 6.0 数据库的新特点，还有 Delphi 数据库应用程序的体系结构，并介绍一个简单的 Delphi 数据库程序实例，让读者对开发 Delphi 数据库应用程序有一个感性的认识和宏观的了解。

2.1 Delphi 6.0 数据库功能新特点

较之 Delphi 5，Delphi 6.0 在数据库方面有以下的功能新特点。

2.1.1 新增数据库访问机制 dbExpress

Delphi 6.0 新增了一个数据库访问机制 dbExpress，它提供非常快速、易于配置访问 SQL 数据库服务器的功能，使用的是一种很方便写的第三方数据库驱动机制。

dbExpress 是一个数据库驱动的集合，提供对 SQL 数据库服务器的快速访问。对于其所支持的每一个数据库，dbExpress 都提供一个驱动，用于使得特定的服务器软件能适应统一的 dbExpress 接口。在配置应用程序的时候，要么需要包含一个单一的 DLL，其实就是一个特定的服务驱动，它包括一些你随意创建的应用文件和一些在运行期加载连接信息的文件；要么需要将应用程序编译成一个独立的 EXE。

一般，dbExpress 包括如表 2-1 中的驱动。

表 2-1 dbExpress 包括的驱动及相应的文件名

驱动	文件名
InterBase 驱动	DBEXPINT.DLL
DB2 驱动	DBEXPDB2.DLL
Oracle 驱动	DBEXPORA.DLL
MySQL 驱动	DBEXPMYS.DLL

在元件板 dbExpress 页下, 主要有 TSQLConnection、TSQLDataSet、TSQLQuery、TSQLStoredProc 和 TSQLTable 等控件。

dbExpress 的数据集控件是单向的数据集控件, 因为 dbExpress 实现的是访问记录的单向指针。单向数据集控件并不在内存中缓冲数据, 从而比其他的数据集更快更节省资源, 不过它也有它的一些局限性。比如, 不能将单向的数据集控件连接到数据显示控件, 因为它没有缓冲的记录。在单向的数据集控件中, TDataSet 介绍的很多功能或者是没实现, 或者是将触发异常。尽管有这些局限, 但是单向的数据集控件是访问数据的一种很好的方式, 它访问速度快、使用和配置简单。

如果要编辑单向数据集中的数据的话, 需要将它们连接到一个客户端数据集。由一个数据集 Provider 提供客户端数据集和单向数据集的连接。

2.1.2 新增两个字段类型

在 Delphi 6.0 中, 增加了 TFMTBCDField 和 TSQLTimeStampField 两个字段类型。

TFMTBCDField 代表一个真正二进制编码的十进制类型的字段, 并不像 TBCDField 字段类型代表的是货币类型转换后的 BCD 码。用户可以为任何数据集中的 BCD 字段指定 TFMTBCDField 类型的永久字段。在使用 dbExpress 数据集的情况下, 利用 TBCDField 创建 TFMTBCDField 类型的动态字段控件将降低精度, 其实在使用 BDE、ADO 以及 IBX 数据集的情况下也一样。

注意:

为了支持 TFMTBCDField 类型, 原来在 db 单元中定义的 CurrToBCD 函数和 BCDToCurr 函数在 FMTBCD 单元中定义。

TSQLTimeStampField 支持 dbExpress 驱动的时间/日期类型。

2.1.3 客户端数据集控件的增强

在 Delphi 6.0 中对客户端数据集有所增强, 增加了 TBDEClientDataSet、TSQLClientDataSet 以及 TIBClientDataSet 三个客户端数据集控件, 它们包含内置的 Provider

和 Source 属性。它们是用来在简单的应用程序中简化使用 Provider 和 ClientDataSet 缓存更新的处理过程。于是，设立了一个新的类 TCustomClientDataSet，新增的三个数据集控件和 TClientDataSet 都是它的子类。

注意：

TBDEClientDataSet、TSQLClientDataSet 以及 TIBClientDataSet 控件是应用于简单的应用程序的。如果应用程序使用 master/detail 关系或者需要频繁地从服务器存取数据，那么应当使用外部的 TDataSetProvider 和 TclientDataSet 控制。

另外，TClientDataSet 控件增加了几个新属性：

1. ConnectionBroker

增加一个额外的间接层来描述连接控件。在设计期和运行期使用不同的连接控件的时候该属性显得特别地有用。在应用程序中，如果几个客户端数据集使用的是同一个连接代理，那么想改变这些客户端数据集所使用的连接的时候，只需要改变连接代理的 connection 属性即可，而不需要改变每一个客户端数据集的 RemoteServer 属性。

2. DisableStringTrim

控制客户端数据集是如何处理用户在字段中输入的空格。

3. XMLData

提供在 XML 形式下对客户端数据集的数据包的访问。

另外，TUpdateSQL 控件的功能也得到了加强，在使用客户端数据集进行缓存更新的时候可以使用多种更新对象。它的 DataSet 属性也从原来的需要一个 TBDEDataSet 控件到需要一个 TDataSet 控件即可。当使用一个客户端数据集和多种更新对象的时候，必须在 Provider 的 BeforeUpdateRecord 事件句柄的 DeltaDS 参数中进行设置。在这种情况下，单凭 DataSet 属性，TUpdateSQL 控件不能确定数据库会话期，而是必须结合设置 DatabaseName 属性和 SessionName 属性才能确定。

在使用单一的更新对象或基于 BDE 的数据集进行缓存更新的时候，TUpdateSQL 控件的使用方式和以前一样。

2.1.4 对多层数据库应用程序支持的增强

在 Delphi 6.0 中增加了 TSharedConnection 和 TLocalConnection 两个连接控件。

对于同一个应用服务器中的多种数据模块，TSharedConnection 控件可以让客户端应用程序建立几个连接。通过使用 TSharedConnection 控件，可以使用一个共享的连接来连接