经典实例精解



PowerBuilder
数据库开发
经典实例精解

陈永强 李 茜 詹红霞 等编著

★ 国内资深数据库专家精心打造
★ 从基础到实战开发全过程精解
★ 9个典型数据库的完整开发过程
★ 开发过程中重点讲解相关理论知识
★ 举一反三，类似数据库系统开发的思路和方法
★ 超值赠送书中实例的完整代码和练习文件



机械工业出版社
CHINA MACHINE PRESS

需要全本请访问 www.ertongbook.com

PowerBuider 数据库开发

经典实例精解

陈永强
李　茜
詹红霞　等编著

机 械 工 业 出 版 社

本书主要介绍如何使用 PowerBuilder 进行数据库管理系统开发。全书所有示例都是以 Windows XP Professional+PowerBuilder 10.0+Adaptive Server Anywhere 9.0 构成的开发环境为基础进行讲解的。全书分为 3 篇：第 0 篇，主要介绍 PowerBuilder 编程的基础知识；第 1 篇包括实例 1 至实例 3，着重介绍 PowerBuilder 基本编程方法；第 2 篇包括实例 4 至实例 8，主要介绍完整的 PowerBuilder 数据库应用程序的开发。

本书定位于已有一定基础的 PowerBuilder 初中级用户、数据库管理系统开发相关人员以及电脑爱好者学习使用，还可作为大中专院校、社会相关办学的辅助教材使用。

为了方便读者学习和操作，书中实例所涉及到的源代码都收录到了本书的配套光盘中，供读者在学习过程中调用。

图书在版编目（CIP）数据

PowerBuilder 数据库开发经典实例精解/陈永强，李茜，詹红霞等编著.
-北京：机械工业出版社，2006.6
（经典实例精解）
ISBN 7-111-19311-3

Ⅰ.P… Ⅱ.①陈… ②李… ③詹… Ⅲ.数据库系统-软件工具，PowerBuilder-程序设计
Ⅳ.TP311.56
中国版本图书馆 CIP 数据核字（2006）第 061475 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：吴宏伟 责任编辑：刘欢欢 版式设计：崔俊利
北京市朝阳展望印刷厂印刷
2007 年 2 月第 1 版第 1 次印刷
184mm×260mm · 22.25 印张 · 519 千字
0001-5000 册
定价：36.00 元（含 1CD）

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
封面无防伪标均为盗版

第0篇

PowerBuilder 程序设计基础

PowerBuilder 作为一款优秀的企业级数据库前端应用开发工具，以其高效快捷的集成开发环境、简洁友好的用户界面、功能强大的数据窗口技术，以及性能优越的数据库访问能力，越来越受到众多数据库开发人员的青睐，成为当前最为流行的数据库前端开发工具之一。对于初学数据库应用程序制作的朋友来说，使用 PowerBuilder 作为开发工具，无疑是一个较为理想的选择。本书是笔者几年来从事 PowerBuilder 数据库应用程序学习和开发的心血结晶，希望它能为您更快更好地掌握 ASP 网页设计插上一双乘风破浪的翅膀！



- *PowerBuilder 简介*
- *数据库应用系统的开发方法*
- *PowerScript 概述*
- *嵌入式 SQL 语句*

0.1 PowerBuilder简介

PowerBuilder 最早是由 PowerSoft 公司开发并发布销售的，并出品了 PowerBuilder5.0 为止的各个版本，初期在广大开发人员中反映并不强烈，到 1995 年左右，由 Sybase 收购 PowerSoft 公司开始并成功推出了 PowerBuilder6.0 版本，由此开始，由于它的易用性、数据窗专利技术、对各种流行数据库系统的广泛支持、独特的开发方法及理念、独特的对象编程技术等等，成为现在为止可能是世界上用于数据库前端开发的最快捷、最方便、最强大的工具。当然，PowerBuilder 也越来越受到众多数据库开发人员的喜爱。

最近几年以来，PowerBuilder 的功能不断增强，特别是加强了网络应用开发功能与 C/S 模式开发功能、分布式应用开发功能的整合，使得用户可以充分利用这些强大的网络应用开发功能，将应用扩展到广阔的 Web 世界。相应的 PowerBuilder 的版本也在不断更新，当前其最新的版本为 10.0。

0.1.1 PowerBuilder 的应用范围

PowerBuilder 的主要应用范围如下：

1．大型数据库系统开发

大型数据库系统 C/S 模式前端开发工具或 B/S 模式应用服务器开发工具。一般应用于需要大型或中型数据库系统支持（如 ORACLE、Sybase、MS SQL Server、DB2、Informix）作为后台数据支撑平台的管理类系统，一般可以承受千万级纪录量的系统，如：

- MRPII 系统 － 物料资源计划
- ERP 系统 － 企业资源计划
- CRM － Customer Relation Management（客户关系管理）
- Call Center － 呼叫中心（Haier）

2．中、小型数据库系统开发

中、小型数据库系统开发，一般数据量为 10 万左右，支持并发或仅单机用户，如中、小型企业的内部信息管理、各种小型软件、共享软件等；

3．电子商务

PowerBuilder 与 Sybase 的其他产品——PowerDynamo、Jaguar CTS、PowerSite 等集成，可以建立起电子商务应用服务的支撑平台。

4．Web 应用的后台维护

PowerBuilder 与其他 Web 应用程序开发工具——PHP、Java、ASP 等集成，可以作为 Intranet 或 Internet 的后台数据库系统维护工具，如异种平台（Unix、Linux、Solaris、NT）上的 ORACLE、MySQL、DB2 等的 Client 方式维护。

0.1.2 PowerBuilder 的主要特点

作为一款优秀的企业级数据库前端应用开发工具，PowerBuilder 主要具有以下一些特

点：

1．可视化的开发环境

PowerBuilder 是一个集成的完全可视化的开发工具。使用 PowerBuilder 可以用一种可视、直观的方式来创建应用程序所需的用户界面和数据库接口。

2．面向对象的开发工具

PowerBuilder 是一种面向对象编程的快速开发工具。同绝大多数的 Windows 应用程序一样，PowerBuilder 采用事件驱动工作方式。在这种方式下，程序的执行主要是由对象的各种事件来触发的——事件代码只有当与之关联的事件发生时才会被执行。PowerBuilder 的窗口、菜单、数据窗口等都是对象。用户也可以构造自己的对象。

3．功能强大的编程语言

PowerScript 语言是 PowerBuilder 专用的编程语言。这种语言是一种增强型的结构化编程语言，其基本语法要素与 Pascal、C 等传统结构化语言类似，而且如同 BASIC 语言一样简单易学。在 PowerScript 中除了可以直接引用 PowerBuilder 提供的大量函数和系统对象外，还支持大二进制对象类型和嵌入式 SQL 语句。

4．独一无二的数据窗口技术

PowerBuilder 最大的特点就是采用了其独一无二的数据窗口技术。这种技术对定义和开发数据库连接过程进行了抽象，使得程序开发人员能够完全可视化地完成应用系统中有关数据库的编程工作。通过数据窗口，用户可以轻松地完成客户端与服务器端之间的数据交互。利用数据窗口控件，则可以方便地显示、修改和更新数据库中的数据。

5．开放的系统

PowerBuilder 可以访问任何一个常用的后台数据库系统，比如 Oracle、Sybase、Informix、DB2 和 SQL Server 等大型数据库，以及 Xbase、FoxPro 和 Paradox 等小型数据库。它本身也带一个小型的个人数据库 ASA（Adaptive Server Anywhere），以方便编程人员学习和使用。PowerBuilder 提供了通用的前台用户界面和相关的数据库接口，开发人员无需关心每个数据库的特点和特性，只需要集中精力进行前台应用程序的开发。

6．多平台的开发环境

PowerBuilder 的开发环境现在有 Windows，Mactintosh 和 Unix 等版本。在不同操作系统开发的应用程序只需要在另一操作系统下重新编译，即可直接运行。

7．网络应用程序的开发

使用 PowerBuilder，除了可以方便快捷地创建传统的基于客户/服务器（C/S）模式的数据库应用系统外，同时还支持基于 N 层体系结构分布式的应用程序和基于 Web 的应用程序的开发。完全具备开发企业级商务应用系统的能力。

0.1.3 PowerBuilder 的集成开发环境

下面我们以 PowerBuilder10.0 为例，简单介绍 PowerBuilder 的集成开发环境。图 0-1 所示为 PowerBuilder10.0 的集成开发界面。从整体来看，集成开发环境可以分成 4 部分：

菜单栏、工具栏、工作区、状态栏。其中，工作区又可分为系统树窗口区、剪贴窗口区、输出窗口区、画板工作区等。实际使用时，这 4 部分的具体内容会随着当前所操作对象的不同而发生变化。

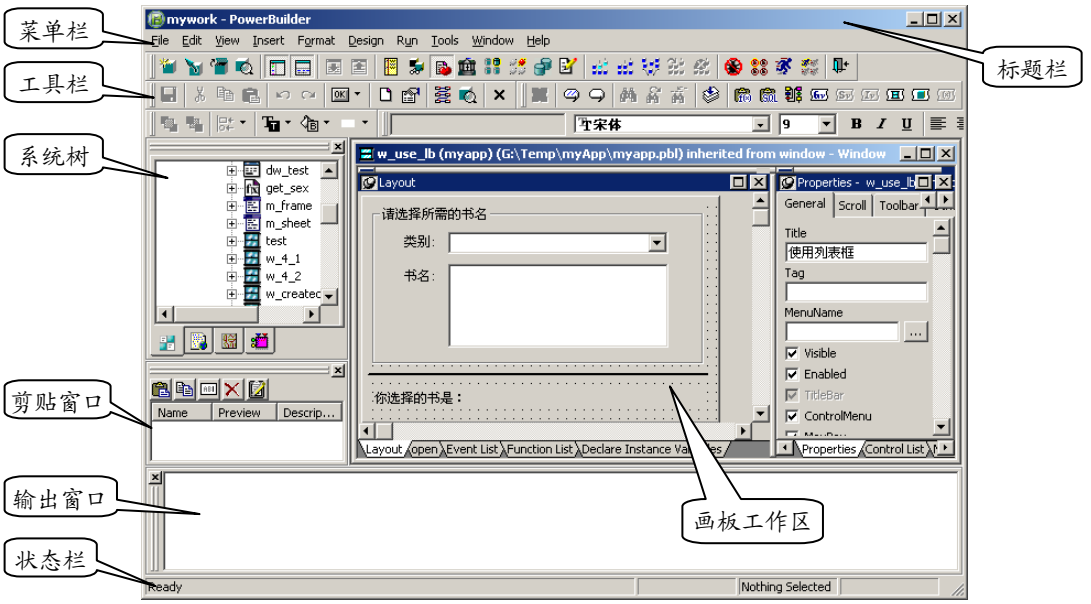
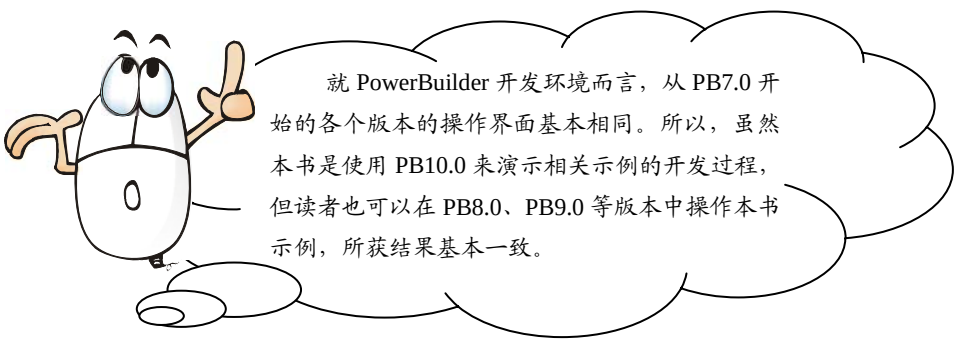


图 0-1　PowerBuilder 的集成开发环境



0.1.4 PowerBuilder 的工作画板

使用 PowerBuilder 开发应用程序时，各种对象的创建和编辑工作主要是在与对象相应的画板中进行的。要学习 PowerBuilder 编程，就必须掌握各种工作画板的功能和使用方法。这里我们简单介绍一下 PowerBuilder 的各种画板，至于其使用方法，我们将在后面的开发示例中进行介绍。

1．应用程序对象画板（Application Painter）

应用程序对象画板用于创建应用程序对象以及保存应用程序所有对象的应用库。应用对象定义应用程序的执行环境，包括三方面的内容：应用对象名称和相应的图标；应用程序中其它对象使用的缺省文本颜色、字体以及字型；应用程序用到的各种对象库等。

2．窗口画板（Window Painter）

窗口画板用于设计应用程序中使用的窗口以及定义窗口中的控件，实际上也就是定义应用程序与用户交互的人机接口。PowerBuilder 提供了大量控件供开发人员直接使用，它既包括常见的命令按钮、单选钮、复选框、编辑框、滚动条、列表框，也包括功能不凡的数据窗口控件、TreeView（树形列表）控件、ListView（列表视图）控件、标签控件、各种绘图控件等。

3．菜单画板（Menu Painter）

在标准的 Windows 应用程序中，菜单是一个不可或缺的界面组成部分。菜单画板用于生成菜单和工具栏对象。菜单对象既可以直接运用在窗口的菜单栏上，也能够作为应用程序的弹出菜单使用。

4．函数画板（Function Painter）

开发应用程序时，经常会发生这样的情况：某段程序需要在多个地方使用，这时就可以用函数画板生成函数，避免重复编码并提高代码的可重用性。函数画板创建的函数是全局函数，在整个应用程序中都能够使用。

5．结构画板（Structure Painter）

结构画板用于定义应用程序中使用的全局结构。结构使得开发人员能够更为有效地组织复杂的变量。定义好的结构可以作为一种新的数据类型在程序中的任何地方进行声明和使用该结构类型的变量。

6．用户对象画板（User Object Painter）

用户对象画板用于生产用户对象，包括可视用户对象和非可视用户对象（即类用户对象）。用户对象一般用于完成某些通用的功能，它可有效提高程序的开发效率和代码质量。另外，用户对象即可在应用程序内部反复使用，也可以单独发布而供其他程序调用。

7．数据库画板（Database Painter）

数据库画板提供了生成和维护服务器端数据库的良好环境，开发人员无须查阅 SQL 语句手册，就能够创建新表、修改老表以及操纵数据。用户可以同时建立多个数据库连接并对之进行测试和修改，但当前处于激活状态的数据库连接只能有一个。

8．数据窗口画板（DataWindow Painter）

数据窗口画板用于生成能够智能操纵数据库的数据窗口对象，该对象可以以多种显示风格查询、修改数据库，且不需要复杂的 SQL 编程。数据窗口对象能够访问多种数据源，包括表、视图、查询、存储过程以及外部数据源等，同时，它有十几种表现风格，极大地丰富了数据的表达能力。数据窗口对象对 PowerBuilder 的应用程序十分重要，只有做到灵活自如地运用数据窗口，才能说学会了 PowerBuilder。

9．查询画板（Qurey Painter）

查询画板可使开发人员在图形界面下以交互方式生成数据库查询语句，并将其作为一个对象进行保存，该对象可为数据窗口对象所用。这样，即使开发人员不十分熟悉数据库操作语句，利用该画板也同样能够生成复杂的数据查询。

10．调试画板（Debug Painter）

调试画板可单步跟踪、断点跟踪应用程序，并可在程序执行过程中手工修改变量的取值，该画板用于程序调试及排错。

11．工程画板（Project Painter）

工程画板用于建立工程文件、编译应用程序、生成可执行文件、生成代理对象、产生类用户对象的 C++头文件/源代码或目标库。

12．数据管道画板（Data Pipeline Painter）

数据管道画板用于把一个数据表或者数据库的结构和数据复制到另一个数据库中，从而实现数据库之间的数据转换。在数据管道画板中定义的这种数据传递方式将作为一个数据管道对象被保存。

13．库画板（Library Painter）

库画板提供了管理应用程序各个组成部分的有效手段。利用库画板可以删除、修改、移出、复制、重新生成应用程序库中的各种对象（如窗口、菜单、数据窗口等等）。当然，用户也可通过库画板的树状列表视图来搜索具体对象。

0.2 数据库应用系统的开发方法

在进行数据库应用系统开发之前，我们有必要先了解一下数据库应用系统开发的一般方法及其相关概念。下面我们就关系数据库系统、数据库应用系统的模式、面向对象的程序设计方法以及应用程序开发的一般步骤等内容作一个简单介绍。

0.2.1 关系数据库系统的概念

数据库系统包括以数据存储为主体的数据库（Database）和以数据管理为主体的数据库管理系统（DBMS）。数据库系统是在文件系统的基础上演变而来的，它的发展经历了层次数据库、网状数据库、关系数据库 3 个阶段。而关系数据库技术经过了几十年的发展，在理论上已经相当成熟，目前也是数据库技术领域的主流。

关系数据库系统的基本概念如下：

- 表单（Table）：由属性和记录构成的二维关系表。
- 视图（View）：基于单表或多表的查询。
- 存储过程（Store procedure）：存储于数据库服务器上的一系列操作的合集，由服务器执行，可以增强可管理性和运行效率。
- 触发器（Triggers）：当对表的某种操作发生时，就引发其他的一些逻辑处理。
- 索引（Index）：服务于快速查询，寄生于表的一种服务。
- 函数，包（function,package）：类似于存储过程。
- 数据库（database）：上述各对象的集合。
- 设备（device、tablespace、datafile）：存放数据库的物理定义。
- 日志（log）：记录对数据库的任何操作，用于备份和恢复。

- 用户（User）：可以使用数据库中具体内容的人员定义。
- 角色（Role）：给同一用户赋予不同的角色，享有不同的权限。
- 权限（Privilege）：赋予用户操作数据的权限。

0.2.2 数据库应用系统的模式

数据库应用系统是一种基于数据库的计算机应用系统。这样的系统大体可以分为两部分：数据库系统和数据库应用程序。其中数据库应用程序是基于数据库的、通过数据库管理系统与数据库进行数据交互的应用软件。现在，基于关系数据库的应用系统大体上分为 3 种模式，即 C/S 模式、B/S 模式和分布式模式。

1．C/S 模式

C/S（Client/Server）模式即我们常说的客户/服务器模式，其结构示意图如图 0-2 所示。C/S 模式使用广泛、技术成熟，并且目前依然有大量的新的实际应用系统采用这种模式。一般情况下，客户端应用程序运行在工作站上，负责提供用户界面和商务逻辑；服务器端应用程序运行在服务器上（对于小规模的系统来说，可以将客户端应用程序和服务器端应用程序放置在同一台计算机上运行）。客户端与服务器端通过网络进行相互通信——客户端程序向服务器发送 SQL 请求并取回结果；服务器端程序则侦听网络，响应客户端的请求并返回相关的处理结果。

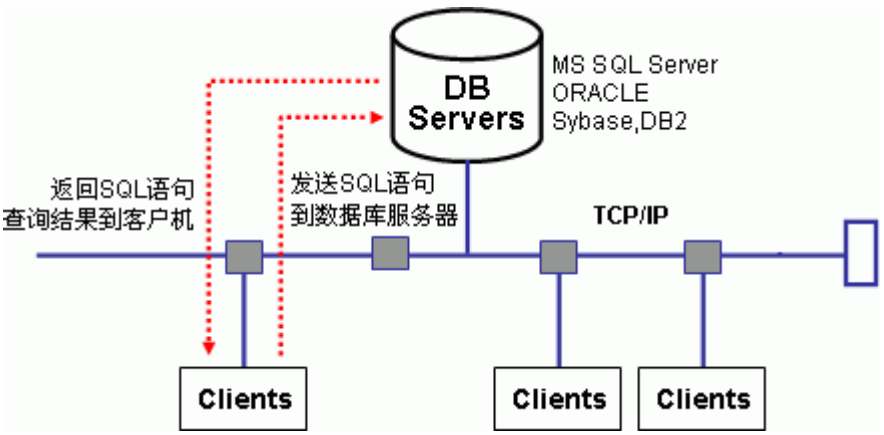


图 0-2　C/S 模式的结构示意图

上述模式也称为双层结构的客户/服务器模式。如果在客户端与服务器端之间添加一些中间件（商业规则），则可以构成 3 层或者 N 层结构的客户/服务器模式。

2．B/S 模式

B/S 模式即 Browser/Server 模式，也称为 Web 模式或者 Internet 模式，其结构示意图如图 0-3 所示。这种模式是随着 Web 技术在 Internet 上的日益广泛应用而逐步发展起来的。B/S 模式中的客户端使用一个通用的 Web 浏览器，以取代 C/S 模式中的客户端应用软件，其数据处理方式则类似于 C/S 模式。这种模式下的客户端也被称之为“瘦客户端”。

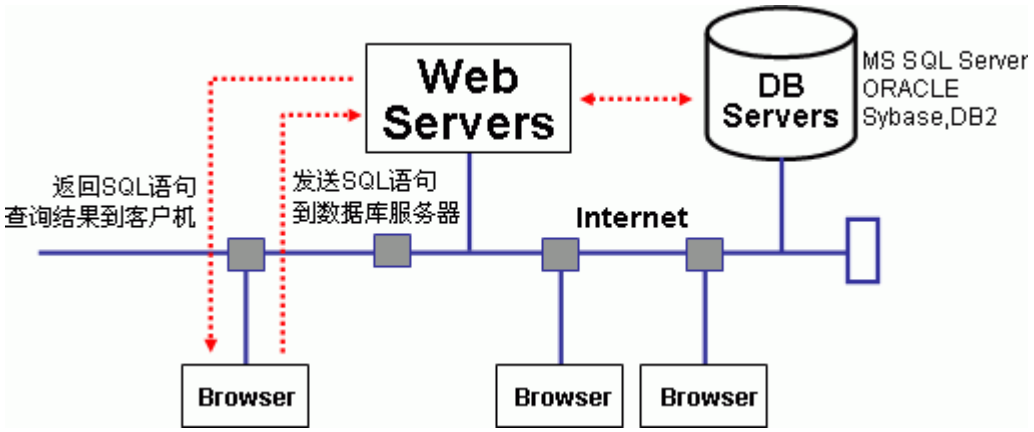


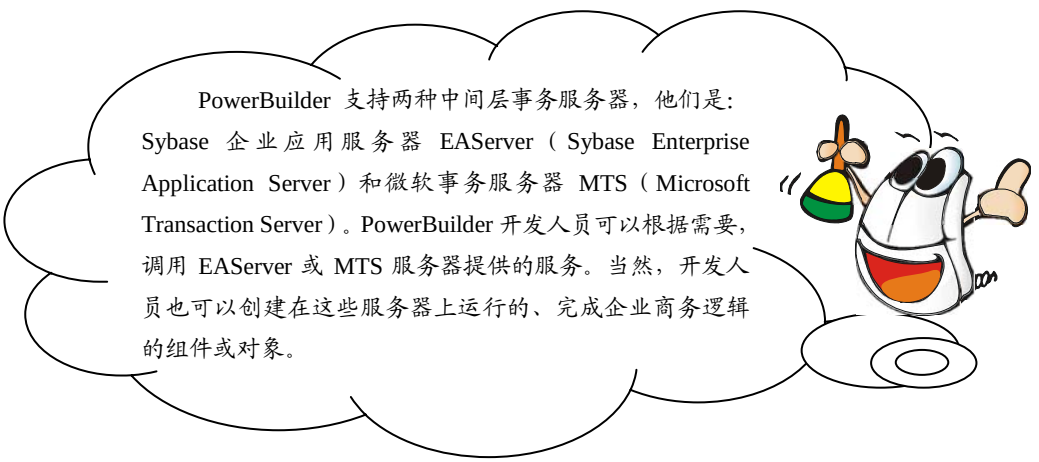
图 0-3　B/S 模式的结构示意图

3．分布式模式

分布式模式提供了一种将应用程序的用户界面部件与应用程序所需的业务逻辑进行分离的自然途径。通过把业务逻辑集中在中间层服务器上，我们可以降低客户端的工作负载，从而控制对敏感信息的访问。

在分布式应用环境中，客户端和服务器端协同工作，完成商务用户所需的任务。客户端处理与用户交互的所有操作，中间层服务器为客户端提供后台服务。典型情况下，中间层服务器完成绝大多数的数据处理和数据库访问。客户端通过调用运行在服务器上的组件或对象的方法，来使用服务器提供的服务。

在分布式应用程序体系结构中，应用程序的划分是一件十分重要的工作。在企业级应用程序中，客户端的应用程序应该尽可能地简单高效，以便保留尽可能大的网络带宽。为了到达这样的目标，应用程序被划分为三个部分：操作界面展示部分、商务逻辑和数据库访问。数据库系统位于企业应用系统的底层，他的目标是存储企业和机构的信息，并保证这些信息的安全性。商务逻辑位于企业应用系统的中间层，他可以运行在组件事务服务器上。操作界面展示部分运行在用户的个人计算机上，这一部分位于企业应用系统的顶层。



0.2.3 面向对象的程序设计

一般说来，面向对象的系统至少需要具备 3 大特性：封装性、继承性和多态习性。将封装、继承、多态（包括重载）等面向对象方法应用于程序的开发和开发过程中，不仅可以加快开发速度，还可以极大地增强应用程序的可维护性和可扩展性，提供代码的重用率。下面对上述 3 种特性进行简单说明。

1．封装性

封装的目的是为了实现数据库隐藏和数据保护，为对象提供一个对外操作的接口，外界只能从对象所提供的操作接口来认识和操作该对象。在 PowerBuilder 中，对象的访问类型有 3 种：Public，Protected，和 Private，对象中的变量、函数均可用这 3 种类型来进行访问控制。其中，实例变量和对象函数的默认访问类型均为 Public 类型，它允许外部对象对其进行任意访问。对于 Protected 类型的对象元素，可以接受其派生对象和对象内部元素的访问；而对于 Private 类型的对象元素，这仅允许对象内部的成员对其进行访问。为了维护数据安全，建议用户在保证数据一致性的前提下尽可能多并正确地使用 Protected 和 Private 访问类型。

2．继承性

继承是现实生活中一个非常容易理解的概念。在面向对象的程序设计方法中，引入继承机制的目的在于：其一，避免可公用代码的重复开发，减少数据冗余；其二，增强数据一致性，尽量降低模块间的耦合程度。在 PowerBuilder 中，可被继承的对象包括窗口、菜单和用户对象 3 种，其余的对象则不具备这种特性。当一个派生对象继承了某个父对象后，它几乎将得到父对象的一切，包括对象的属性、事件、函数、控件、各种变量、用户自定义事件和函数、有关代码等。但需要注意的事，派生对象所继承到的这些构成元素不能被随意作改动。例如，在派生对象中不可以删除从父对象继承而来的实例变量和共享变量、不可以改变从父对象继承而来的各种控件的名称等。

3．多态性

所谓多态性就是当不同的对象收到相同的消息时，产生不同的动作。这里所说的消息可以理解为函数名或事件名。通俗地讲，多态性就是使用一个名称来定义不同的函数，这些函数执行类似的但又不同的操作，，即以相同的接口来访问功能不同的函数，从而实现“一个接口，多种方法”。编译时的多态性（动态多态性）可以通过函数重载来实现。函数重载是面向对象程序设计的一个重要特性，即同一函数名却有着不同的参数和返回值。使用重载函数减少了程序员的记忆量，它使得程序员在只知道操作的一般含义而不知道操作的具体细节的情况下，仍然能够进行正确地完成有关操作。

0.2.4 PowerBuilder 应用程序的开发流程

使用 PowerBuilder 开发一个数据库应用系统，一般都要经过系统分析、系统设计、系统开发、代码调试、系统测试、生成 EXE、发布应用等几个阶段。图 0-4 显示了应用程序的开发流程，而图 0-5 则显示了应用程序开发过程中各阶段工作量的分配情况。

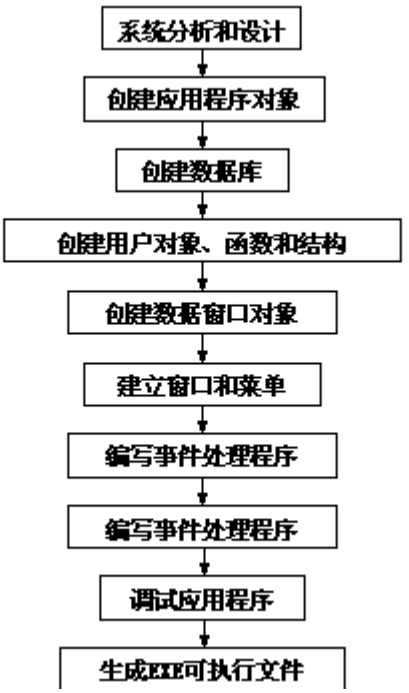


图 0-4　PowerBuilder 应用程序的开发流程

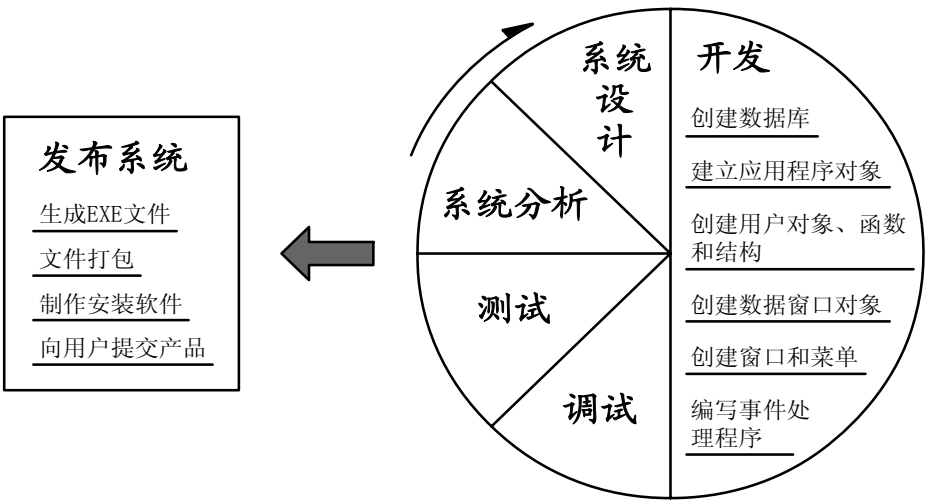


图 0-5　PowerBuilder 应用程序开发过程及工作量分配

图 0-4 和图 0-5 中主要步骤的任务和基本要求如下：

- 系统分析和系统设计

实际上，在开发任何应用系统之前，开发人员都必须充分了解系统需求，理解系统应该完成的真正任务，千万不能因为赶进度而过分缩减或忽略该阶段的工作，这一阶段的成果很大程度上决定了项目的成败。理解了系统需求后，就进入系统设计阶段，此时考虑用户界面的风格、使用何种数据库、如何组织数据等。

- 系统开发

在 PowerBuilder 中，建立应用程序需做的大量工作是创建对象、描述对象、设置属性，而不是编写大量的代码。系统开发过程可分为：创建数据库；建立应用对象；生成用户对象、编写函数和创建结构；建立窗口和菜单；创建数据窗口对象；将数据窗口对象连接到窗口上的数据窗口控件上；编写各对象的事件处理程序。除了建立应用对象需要在应用开发的一开始就要完成外，其它步骤可按任意次序进行。

- 调试

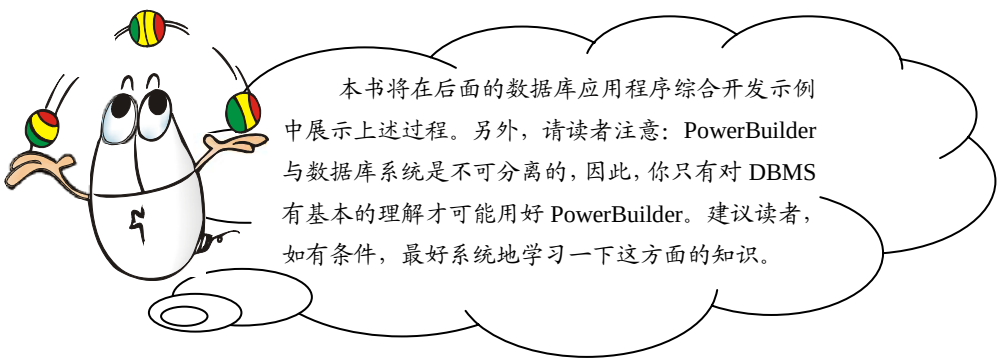
PowerBuilder 的调试画笔提供了单步跟踪、断点跟踪、条件断点、变量查看等多种调试功能，可以帮助开发人员快速地排除程序错误，而且，它的自动编译功能在保存事件处理程序以及函数时能够找出所有的语法错误。另外，使用 PowerBuilder 的 PBDEBUG 功能可以跟踪和记录应用程序的运行情况，使用数据库调试功能可以查出数据访问方面可能存在的瓶颈。开发人员应该清晰地认识到，调试应用程序不是为了证明程序不存在错误，而是为了尽可能多地找出所存在的问题，没有找到任何错误的调试是次不成功的调试。同时，调试也是件费时、费力的创造性工作。

- 测试

测试用于排除系统设计缺陷，比如是否实现了某些功能。

- 生成可执行文件

系统测试无误后，就可以将应用编译成可执行文件用于发行了。



0.3 PowerScript概述

PowerScript 是 PowerBuilder 的专业编程语言。它并不仅仅是一种简单的第四代语言，而是一种高级的、结构化的编程语言。我们编写的事件处理程序、用户自定义的函数等都是使用 PowerScript 语言编写的。如果你以前使用过其他高级语言，就会发现 PowerScript 十分简单易学，他的许多语言成分与大多数编程语言很相似，比如，PowerScript 和 C 均使用（//）和（/*　*/）来注释程序行。

0.3.1 语言基础

PowerScript 是一种自由格式的语言，语句行中的空格、缩进等格式编排信息完全被编译器忽略，为了阅读方便，我们可以随意安排语句行的位置。与其它大多数编程语言相似，

PowerScript 提供了断行、续行的方法，如果你愿意，在一行中写上几条语句也毫无问题，另外，该语言提供了两种注释方法，通过注释进一步提高程序的可读性。

1．断行、续行与多条语句

通常情况下，PowerScript 的一条语句书应写在一行上，语句书写完毕后，按 Enter 键（回车键）转到下一行，开始下一条语句。有时候，为了阅读方便等原因，需要把一条语句书写在几行上，这时就需要使用续行符（&），它放在一行的末尾指示下一行是当前行的继续。例如可以将语句：

```
messagebox("操作提示", "当前操作将删除选中的记录！是否继续？...", Question!, YesNo!, 2)
```

书写在两行上，即：

```
messagebox("操作提示", "当前操作将删除选中的记录！是否继续？...", + &
Question!, YesNo!, 2)
```



不能在标识符或保留字的中间续行，这样会将标识符分成了两行。另外在续行符后面不能添加注释信息，否则系统将报错。

如果想在一行中书写多条语句，那么在语句之间用分号（；）进行分隔，例如：

```
b = 1;   c = 2;   a = b + c
```



一般情况下，建议不要在一行中书写多条语句，因为一条语句占据一行既直观又清晰，便于程序的阅读。

2．注释

注释是添加在程序中的一些说明性文字，以便于用户阅读程序。PowerBuilder 并不执行任何注释，而是简单地跳过它。PowerScript 的注释有两种方式：行注释和块注释。行注释使用双斜线（//）作为注释符，用于注释单行说明文字或者一条暂时不用的语句；块注释需要将语句放在注释符（/*　*/）之间，用于注释多行说明文字或语句。语句一旦被注释将以蓝色字体显示。注释符的使用示例如下：

```
// 注释的使用
/*-----------------------------------------------
  *    用户自定义函数  my_get()
  *-----------------------------------------------*/

// return -1
if a > b then return 1
```



在 PowerBuilder 的脚本编辑器中，可以单击工具栏上的注释（Comment）图标快速将选中的语句行转换为注释行，单击（Uncomment）图标可以取消选中语句行的注释。另外，对于暂时不用的语句，不要将其删除，只需将它们设为注释语句，在以后需要使用时再取消其注释便可。

3．标识符

标识符是程序中用来代表变量、标号、函数、窗口、菜单、控件、对象等名称的符号，在 PowerBuilder 中，定义标识符必须遵从下述规则：

- 标识符必须以字母或下划线开头。
- 标识符的非开头字符可以是字母、数字或者五中特殊符号（-、_、$、#、%）。
- 标识符不区分大小写（比如，PART、Part、part 是一个标识符）。
- 标识符的书写应保持连续，中间不能留空格。
- 标识符的最大长度应控制在 100 个字符内。
- 保留字不能用做标识符，因为它们已有特殊的含义。

标识符的使用示例如下：

```
// 下面是一组正确的标识符
rc                  // 返回值
r_code              // 代码
Large_Button#1      // 第一个大按钮
Child-ID            // 使用短横线（不建议使用）
_SpecialID          // 以下划线开头
//    下面是一组错误的标识符
Abc def             // 标识符中间不能有空格
abc>def             // 标识符中间有非法字符
This                // 误用保留字
2th                 //以数字开头
```



短横线与减号是同一个字符，因此，表达式中使用减法运算符时，必须在减号的两边加上两个空格，否则可能产生语法甚至语义错误。通常情况下，尽量不要在标识符中使用短横线（-），以避免与减法运算混淆，用下划线分隔单词是较好的替代方法。

4．特殊 ASCII 码字符

字符串中可以包括特殊的 ASCII 码字符，它们不能使用常规的输入方法直接输入，需要使用其它字符来代替，表 0-1 列出了常用特殊字符的表示方法。

表 0-1　　常用特殊字符的表示方法

特殊字符	表示方法
新行（Newline）	~n
制表符（Tab）	~t
垂直制表（Vertical Tab）	~v
回车（Carrige Return）	~r
换行（Formfeed）	~f
退格（Backspace）	~b
双引号（"）	~"
单引号（'）	~'
波浪号（~）	~~
十进制形式的 ASCII 码	~o00 到~255
十六进制形式的 ASCII 码	~h00 到~hFF
八进制形式的 ASCII 码	~o00 到~o377



波浪号（~）用于引用特殊 ASCII 字符，八进制形式的 ACSII 码表示方式中，波浪号后的第一个符号是英文字母 O 而不是数字 0。另外，在编写字符型表达式时可能要大量用到这些转义符。

特殊的 ASCII 码字符的使用示例如下：

```
// 显示一用户的相关信息
messagebox ("用户信息", "当前用户的基本信息如下：" + &
                        "~n~n 姓名~t 密码~t 身份" + &
                        "~n~n 陈××~t123456~t 系统管理员")
```

其对应的运行界面如图 0-6 所示。

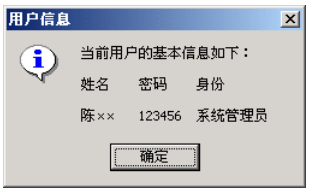


图 0-6 使用转义符

5．代词

日常生活中我们用我、你、他之类的代词指代特定的人物，PowerBuilder 也提供了几个代词指代特定的对象，程序中使用代词的好处是避免硬代码，做到通用化。PowerBuilder 提供了四个代词，分别为：This、Parent、ParentWindow 和 Super，其中前三个代词应用广泛，下面简单介绍一下这三个代词的意义和用法。

- This：代表窗口、用户对象、菜单、应用对象或控件本身，即代表正在为之编写事件处理程序的对象。例如窗口中有一个名称为“cb_button”的按钮，其上显示的文本为“请单击”，在该按钮的“Clicked”事件中可以添加如下代码：

```
// 设置按钮文本
this.text = "再单击一次"
// 语句执行后将把当前按钮的文本改为"再单击一次"
// 上面的语句等同于 cb_button.text = "再单击一次"
```

- Parent：指当前控件所在的窗口。例如，窗口“w_win”中有个文本为“添加”的按钮，当用户单击该按钮时将窗口的标题改为“用户选择了〖添加〗按钮”，我们可以在该按钮的“Clicked”事件中添加如下代码：

```
// 设置当前窗口的标题
parent.title = "用户选择了〖" + this.text + "〗按钮"
// 上面的语句等同于 w_win.title = "用户选择了〖添加〗按钮"
```

- ParentWindow：代表运行时菜单所在的窗口，该代词只能在菜单的事件处理程序中使用。例如，窗口“w_win”关联了一个菜单，该菜单的“关闭”菜单项用于关闭窗口“w_win”，我们可以在该菜单项的“Clicked”事件中添加如下代码：

```
// 关闭关联窗口
close(parentwindow)
// 上面的语句等同于 close(w_win)
```

- Super：在编写控件或对象的子对象时，子对象中可以调用父对象的事件处理程序，程序中既可直接通过父对象的名称调用他们，也可以使用 Super 代词来引用。例如，想调用父对象的 Clicked 事件处理程序时，子对象中可以这样编写：

```
CALL Super::Clicked
```

当调用在子类中被重载的父对象的函数时，也可以使用 Super 来指代父对象。例如，子对象重载了父对象的函数 f_func()，在子对象中调用父对象的 f_func()函数的语句可以这