

数据库应用开发技术丛书

PowerBuilder

数据库高级教程

陈磊 等编著



清华大学出版社

数据库应用开发技术丛书

PowerBuilder 数据库 高级教程

陈 磊 等编著

清华大学出版社

北 京

内 容 简 介

本书提供利用 PowerBuilder 开发高级数据库应用程序的实际解决方案,并不一味地追求各种知识的系统介绍,而是以解决问题为基础,按照各个专题讲解 PowerBuilder 程序员在开发中经常使用的技术。本书主要内容包括各种大中小数据库的介绍、高级 SQL 编程、各种数据库连接技巧、数据处理技术、存储过程高级技术、触发器设计、二进制数据处理、打印报表高级技术、数据备份/恢复技术、数据格式转换技术、数据库安全、高级界面编程、三层数据库开发、帮助文件和安装文件制作等一系列读者在数据库编程过程中需要解决的各种技术。本书除了解决各种技术问题外,还着重分析了功能强大的数据窗口的使用方法和技巧。

本书内容全面,覆盖面广,对于各种数据库技术均有详细介绍;示例丰富,对于书中涉及的所有知识点均用实例加以演示说明,可以强化读者对知识点的理解。本书适用于使用 PowerBuilder 进行数据库应用系统开发的读者,是数据库程序员的理想参考书。

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。

图书在版编目(CIP)数据

PowerBuilder 数据库高级教程/陈磊等编著. —北京:清华大学出版社, 2004

(数据库应用开发技术丛书)

ISBN 7-302-08333-9

I. P… II. 陈… III. 数据库系统—软件工具, PowerBuilder IV. TP311.56

中国版本图书馆 CIP 数据核字(2004)第 023815 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

组稿编辑: 孟毅新

文稿编辑: 刘金喜

封面设计: 久久度企划

版式设计: 康 博

印 刷 者: 北京市世界知识印刷厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 30 字数: 711 千字

版 次: 2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号: ISBN 7-302-08333-9/TP · 6006

印 数: 1 ~ 4000

定 价: 42.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704

前 言

PowerBuilder 是一种可视化的、面向对象的应用程序开发环境，它已经被许多数据库应用开发人员熟悉和使用，并获得广泛好评。与其他应用程序开发工具相比，PowerBuilder 是最容易学习的数据库应用系统开发工具之一，初学者完全可以在很短的时间内利用其简单的开发语言和大量实用的控件编写一个颇具规模的应用程序。这也是 PowerBuilder 在诞生后迅速流行的原因。

但是，初学者在经历一段学习后往往会发现，自己的编程水平开始停滞不前，特别是在编写了大量初级代码，不再满足仅仅编写简单的数据库操作功能时。本书就致力于解决这个问题，帮助读者进一步提高开发 PowerBuilder 应用程序的水平。

考虑到本书的读者对 PowerBuilder 已经有一定的认识，因此本书没有对 PowerBuilder 的操作方法和 PowerBuilder 的各种函数、对象做系统的介绍（这些内容都可以在 PowerBuilder 的帮助中查到），而是针对 PowerBuilder 的初级程序员，以各种专题的形式介绍了进一步提高 PowerBuilder 应用程序开发水平所必备的知识，并在此基础上提供数据库高级程序开发的实际解决方案，可以说本书全部是以解决问题为基础的。

本书共分为 18 章。前两章介绍了 PowerBuilder 和数据库的基础知识；第 3 章系统地介绍了如何编写 SQL 语言；第 4 章通过实例介绍了 PowerBuilder 和各种数据库的连接技术；第 5 章和第 6 章介绍了 PowerBuilder 对数据库以及数据库中各种对象的操作技术；第 7 章系统讲解了 PowerBuilder 对二进制数据的处理；第 8 章和第 9 章介绍了存储过程和触发器的设计方法；第 10 章和第 11 章详细讲解了数据窗口的操作和打印；第 12 章介绍了数据库的备份和恢复；第 13 章介绍了如何利用 PowerBuilder 转换数据格式；第 14 章介绍了如何保护数据的安全；第 15 章讲解了如何开发现在流行的三层数据库应用系统；第 16 章以大量实例和代码讲解了如何利用 API 和现有控件开发高级界面；最后在第 17 和第 18 章介绍了帮助文件和安装程序的制作方法。

自 1997 年开始使用 PowerBuilder 开发数据库应用程序以来，本书作者已经完成了多个大型实用系统的研发，积累了丰富的经验，本书中的大量源码都是作者在实际开发中用到的。在编写本书时，作者融入了自己多年运用 PowerBuilder 开发应用系统的经验、教训和体会，并站在读者的立场上，充分考虑读者的需求，尽可能地为读者提供最为有用的信息。

本书由陈磊编写，另外，张凡、李程、曹峰、廖爽、刘维佳、周荣华、曾国华、邵晓慧、甄伟、缪辉、仲新辉、付海燕、陈峰、周勇、周峻峰、陈军、孙华、裴清、顾晔、姜海、朱有章、张伟峰、关政、刘信义、祝磊、赵刚、徐娟、徐闻闻和黄庭等同志在整理材料方面给予了作者很大的帮助。在此，对他们表示感谢。

由于作者水平有限，书中难免有疏漏和不足之处，恳请广大读者提出宝贵意见。

编 者

目 录

第 1 章 PowerBuilder与数据库开发.....	1
1.1 数据库应用的两种模式	1
1.2 PowerBuilder 在数据库应用系统开发中的优异表现.....	2
1.2.1 专业的客户机/服务器开发工具	2
1.2.2 面向对象的程序设计	3
1.2.3 与 EAServer 共筑多层应用架构.....	7
1.3 小结.....	8
第 2 章 大型关系型数据库	9
2.1 关系型数据库系统概述	9
2.2 数据库的发展.....	11
2.3 数据库的分类.....	13
2.4 Oracle 数据库.....	15
2.5 SQL Server 数据库	17
2.6 Sybase 数据库	19
2.7 Access 数据库	20
2.8 关系型数据库的设计原则	21
2.8.1 数据库设计的基本任务	21
2.8.2 数据库设计的基本流程	23
2.8.3 数据库设计的几个原则	24
2.9 数据库设计实例.....	26
2.10 小结.....	29
第 3 章 高级SQL编程	30
3.1 概述.....	30
3.2 利用 SQL Server 学习 SQL 语言.....	31
3.2.1 SQL Server 2000 的组件	31
3.2.2 如何在企业管理器中使用 SQL 语言.....	34
3.2.3 利用查询分析器更深入地学习 SQL 语言.....	35
3.3 基本 DDL 语言	36
3.4 基本 DML 和 DQL 语言	38
3.5 超越基本的数据操纵语言	43

3.5.1	聚合函数在选择列表中的使用	43
3.5.2	子查询基础知识	45
3.5.3	用 CASE 函数处理条件数据	47
3.6	查询优化	49
3.6.1	优化概述	49
3.6.2	优化分类	50
3.6.3	优化建议	51
3.7	高级 TSQL 语言	54
3.7.1	TSQL 变量的声明	54
3.7.2	运算符和函数	56
3.7.3	流程控制语句	57
3.8	小结	58
第 4 章	PowerBuiler与大型数据库的连接	59
4.1	概述	59
4.2	网络数据库的网络拓扑结构	61
4.3	ODBC 连接技术	62
4.3.1	利用 ODBC 技术访问单一数据库	62
4.3.2	利用 ODBC 在多个数据库间动态切换	66
4.3.3	利用 ODBC 同时访问多个数据库	67
4.3.4	在 PowerBuilder 程序中添加 ODBC 数据源	68
4.4	OLE DB 连接技术	70
4.4.1	安装 OLE DB 接口和 MDAC 组件	70
4.4.2	OLE DB 数据源供应者(Provider)的定义	71
4.4.3	OLE DB 数据源接口的定义	73
4.5	JDBC 连接技术	73
4.5.1	安装 JDBC 接口	74
4.5.2	PATH 和 CLASSPATH 环境变量的设置	75
4.5.3	为 PowerBuilder 定义数据库连接参数	77
4.6	专用接口连接技术	80
4.7	PowerBuilder 同 Access 数据库连接实例	81
4.8	PowerBuilder 同 SQL Server 数据库连接实例	82
4.9	PowerBuilder 同 Oracle 数据库连接实例	84
4.10	小结	86
第 5 章	PowerBuilder数据操作技术	87
5.1	概述	87

5.2 增加数据	89
5.2.1 在数据库画笔中增加数据	89
5.2.2 在数据窗口中增加数据	90
5.2.3 利用脚本语言中嵌入的 SQL 语句增加数据	90
5.3 删除数据	91
5.3.1 在数据库画笔中删除数据	91
5.3.2 在数据窗口中删除数据	91
5.3.3 利用脚本语言的嵌入 SQL 语句删除数据	92
5.4 修改数据	92
5.4.1 在数据库画笔中修改数据	92
5.4.2 在数据窗口中修改数据	93
5.4.3 利用脚本语言的嵌入 SQL 语句修改数据	93
5.5 查找数据	94
5.5.1 在数据库画笔中查找数据	94
5.5.2 在数据窗口中查找数据	95
5.5.3 利用脚本语言嵌入 SQL 语句查找数据	95
5.6 过滤数据	96
5.6.1 在数据库画笔中过滤数据	96
5.6.2 在数据窗口中过滤数据	96
5.7 数据排序	97
5.7.1 数据库画笔中的数据排序	98
5.7.2 在数据窗口中排序数据	99
5.8 主从表处理	99
5.9 事务处理	102
5.9.1 事务概述	102
5.9.2 PowerScript 的事务管理语句	103
5.9.3 事务对象的 AutoCommit 属性	104
5.10 提高数据处理效率的技术	104
5.11 小结	105
第 6 章 PowerBuilder 数据库操作技术	106
6.1 概述	106
6.2 创建数据库	107
6.2.1 利用 Sybase Central 工具创建数据库	107
6.2.2 通过 SQL 语句创建数据库	110
6.3 创建表	112
6.3.1 利用 PowerBuilder 数据库画笔创建表	112

6.3.2 利用 SQL 语句创建表	115
6.4 创建视图	115
6.4.1 视图概述	116
6.4.2 利用 PowerBuilder 数据库画笔建立视图	117
6.4.3 利用 SQL 语句创建视图	119
6.5 创建字段	120
6.5.1 利用 PowerBuilder 的数据库画笔创建字段	120
6.5.2 利用 SQL 语句创建新字段	120
6.6 创建索引	121
6.6.1 索引概述	121
6.6.2 利用 PowerBuilder 数据库画笔建立索引	123
6.6.3 利用 SQL 语句创建索引	124
6.7 获取数据表信息	124
6.8 获取字段信息	126
6.9 小结	128
第 7 章 二进制数据的处理	129
7.1 概述	129
7.2 普通二进制数据处理	130
7.2.1 在数据窗口中使用二进制数据	130
7.2.2 不使用 OLE 方式对二进制数据进行操作	132
7.3 图像文件处理	136
7.3.1 图片控件的使用	136
7.3.2 使用 OLE 对象处理图像	136
7.4 声音文件处理	139
7.4.1 调用 API 函数播放声音	139
7.4.2 使用 OLE 对象处理声音	139
7.5 视频文件处理	140
7.6 二进制数据的查询	141
7.7 Office 文件处理	142
7.7.1 Word 文件和 PowerPoint 文件的处理	142
7.7.2 Excel 文件的处理	149
7.8 小结	150
第 8 章 存储过程高级设计技术	151
8.1 概述	151
8.2 游标使用技术	151

8.2.1 基本游标语句	152
8.2.2 高级游标语句	154
8.2.3 PowerBuilder 中的游标语句实例	155
8.3 动态执行	156
8.3.1 直接执行的动态 SQL	156
8.3.2 带动态参数的动态 SQL	157
8.4 创建存储过程	157
8.4.1 创建存储过程的语法	158
8.4.2 使用 cursor 数据类型的参数	159
8.4.3 临时存储过程的作用域	160
8.4.4 创建存储过程实例	160
8.5 将消息集成到存储过程	161
8.6 在 PowerBuilder 中调用存储过程	162
8.6.1 存储过程的定义和执行	162
8.6.2 利用动态 SQL 语句定义和调用存储过程	163
8.6.3 通过创建事务对象调用存储过程	167
8.7 小结	170
第 9 章 触发器高级设计技术	171
9.1 概述	171
9.2 嵌套、递归和触发器基础	171
9.2.1 触发器的创建	172
9.2.2 嵌套触发器	175
9.2.3 递归触发器	176
9.2.4 Inserted 和 Deleted 虚拟表	176
9.3 使用触发器加强业务规则	176
9.4 触发器检查	178
9.4.1 查看表中的触发器类型	178
9.4.2 查看触发器的文本	179
9.4.3 查看触发器的相关性	179
9.5 存储系统信息	180
9.6 用触发器维护引用完整性	181
9.7 级联删除和级联更新触发器	182
9.8 触发器设计实例	183
9.9 小结	186

第 10 章 数据窗口高级技术	187
10.1 概述	187
10.2 数据窗口控件的重要属性、事件和函数	188
10.2.1 数据窗口控件的重要属性	188
10.2.2 数据窗口控件的重要事件	189
10.2.3 数据窗口控件的重要函数	197
10.3 数据窗口的数据连接	209
10.3.1 数据源选择	209
10.3.2 数据窗口与多个数据库表的连接与更新	214
10.4 数据窗口的各种数据处理技术	216
10.4.1 数据窗口中的数据访问	217
10.4.2 数据存储对象(Data Store)的使用	220
10.4.3 OLE 对象在数据窗口中的实现	221
10.5 数据窗口共享数据技术	223
10.5.1 共享数据窗口在下拉数据窗口中的应用	223
10.5.2 在查询式数据窗口中使用共享数据窗口	224
10.5.3 共享数据技术在数据窗口打印中的应用	224
10.5.4 共享数据窗口在图形显示中的应用	225
10.6 数据窗口事件处理技术	225
10.6.1 数据窗口消息的含义	226
10.6.2 用户自定义事件的使用	226
10.7 动态数据窗口技术	227
10.7.1 动态创建数据窗口的相关函数	228
10.7.2 动态创建数据窗口的一般方法	230
10.8 数据窗口的界面显示技术	232
10.8.1 数据窗口的表现风格	232
10.8.2 数据窗口的布局	240
10.8.3 列对象的编辑与显示	242
10.8.4 数据窗口增强对象的使用	249
10.8.5 依据条件动态设置界面显示的示例	257
10.9 数据窗口错误处理技术	257
10.9.1 数据窗口的有效性验证	258
10.9.2 数据窗口的错误处理	261
10.10 小结	263
第 11 章 报表/图表高级技术	264
11.1 概述	264

11.2	单表打印	267
11.3	多表打印	270
11.4	嵌套报表	279
11.5	复合报表	282
11.6	自定义报表	284
11.7	各种数据统计图表显示	285
11.8	小结	288
第 12 章	数据备份与恢复	289
12.1	概述	289
12.2	数据库的复制	290
12.3	数据表的复制	291
12.4	复制指定的数据库对象	298
12.5	数据上报/备份和数据的接收/恢复	299
12.6	小结	305
第 13 章	数据格式转换	306
13.1	概述	306
13.1.1	在数据库画笔中转换数据	306
13.1.2	在数据窗口画笔中转换数据格式	308
13.1.3	在应用程序中导出数据	309
13.2	将数据转化成 CSV 格式	311
13.3	将数据转化成 HTML 格式	312
13.3.1	HTML Table 选项卡中的参数设置	312
13.3.2	应用程序中 HTML 格式文件的创建	315
13.4	将数据转化成 XML 格式	316
13.4.1	XML 格式简介	316
13.4.2	在数据库画笔和数据窗口画笔中产生 XML 格式数据	319
13.4.3	创建 XML 模板	320
13.4.4	在应用程序中以 XML 格式导出数据	322
13.5	将数据转化成 TXT 文件	322
13.6	将 DBF 转化成 MDB	323
13.7	将 XLS 转化成 MDB	324
13.8	小结	324
第 14 章	数据库安全	325
14.1	概述	325
14.2	数据库自身安全机制	326

14.2.1	数据的安全性	326
14.2.2	保证数据库的完整性	338
14.2.3	保证数据库的一致性	339
14.3	程序实现安全机制	340
14.3.1	数据安全性的实现	340
14.3.2	数据完整性规则的实现	342
14.4	加密存储数据	344
14.5	SQL 的加密机制	345
14.6	小结	347
第 15 章	三层数据库应用系统开发	348
15.1	概述	348
15.1.1	三层模式的主要优点	348
15.1.2	PowerBuilder 对三层模式的支持	349
15.2	Jaguar CTS 应用技术	353
15.2.1	Jaguar 应用程序的结构	353
15.2.2	设计 Jaguar 应用程序	354
15.2.3	Jaguar 组件和客户端程序的开发	358
15.2.4	Jaguar 应用程序的提交	359
15.2.5	利用 PowerBuilder 开发 Jaguar 组件	360
15.3	PowerDynamo 应用技术	360
15.3.1	PowerDynamo Web 应用开发	361
15.3.2	构造 PowerDynamo Web 站点	361
15.4	三层数据库开发实例	365
15.4.1	定义 EAServer 连接配置参数	366
15.4.2	创建服务器端组件	367
15.4.3	创建客户端程序	369
15.5	小结	375
第 16 章	高级界面开发	376
16.1	概述	376
16.2	类 QQ 界面开发	376
16.3	托盘程序开发	388
16.4	全屏程序开发	393
16.5	COOL 菜单、工具栏开发(BCG)	394
16.6	多文档程序开发	395
16.7	其他高级界面开发	398

16.7.1 利用 API 实现动画特效	399
16.7.2 COM 组件的使用	404
16.7.3 移动没有标题条的窗口	404
16.7.4 控制窗口的自动重绘	405
16.8 小结	405
第 17 章 帮助文件的开发	406
17.1 概述	406
17.2 帮助文件介绍	407
17.2.1 帮助文件的组成	407
17.2.2 帮助文件制作工具简介	410
17.3 利用 Microsoft HTML Help Workshop 制作帮助文件	410
17.3.1 创建项目文件	411
17.3.2 创建目录文件	414
17.3.3 创建索引文件	419
17.3.4 创建搜索	420
17.3.5 使用 HTML Help Workshop 的两点技巧	421
17.4 应用程序同帮助文件关联	422
17.4.1 使用 ShowHelp 和 ShowPopupHelp 函数调用帮助文件	422
17.4.2 定义 WinHelp 的 API 函数调用帮助	423
17.5 小结	423
第 18 章 安装程序制作	425
18.1 InstallShield 概述	425
18.1.1 InstallShield 和安装程序简介	425
18.1.2 InstallShield 的集成开发环境	426
18.2 利用 Install Shield 制作高级的数据库安装程序	428
18.2.1 使用向导制作数据库安装程序	428
18.2.2 InstallScript 语言	433
18.2.3 制作多个安装程序	453
18.3 小结	455
附录	456

第1章 PowerBuilder与数据库开发

PowerBuilder 是 Sybase 公司的子公司 PowerSoft 推出的数据库应用开发工具,利用它,不仅可以设计传统的基于客户机/服务器(Client/Server)体系结构的应用系统,也能开发基于 Internet 的应用系统。

PowerBuilder 支持应用系统访问多种数据库,包括 Oracle、Sybase 之类的大型数据库,以及小至 FoxPro 之类的支持 ODBC 接口的数据库,而且 PowerBuilder 可以根据应用程序的需要同时访问多个数据库。此外,PowerBuilder 开发环境随身携带的 Sybase SQL Anywhere 本身就是一个功能强大的 DBMS,对小型应用来说,可以直接使用这个数据库。

本章将综述 PowerBuilder 与数据库开发之间的关系,以及 PowerBuilder 在数据库应用开发中的优势。

1.1 数据库应用的两种模式

计算机应用的一个非常重要的方面就是数据库应用,而客户机/服务器模式和浏览器/服务器模式则是当前数据库应用技术中普遍采用的两种体系结构。这两种体系结构中,所有的数据和数据库管理系统都在服务器上。采用客户机/服务器模式时,客户机通过采用标准的 SQL 语句等方式来访问服务器上数据库中的数据;而浏览器/服务器模式则更加精简了客户机的功能,把 SQL 语句的查询工作也放在了服务器上。由于这两种体系结构都把数据和对数据的管理放在了服务器上,因此很好地保证了数据的安全性和完整性,同时也可以充分利用服务器的资源。正因为客户机/服务器体系结构和浏览器/服务器体系结构具有这些优点,因而它们得到了非常广泛的应用。

一个典型的客户机/服务器模式的数据库应用系统的结构如图 1-1 所示。

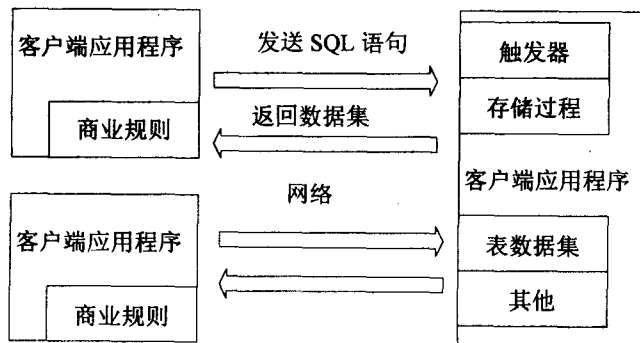


图 1-1 客户机/服务器模式

如图 1-1 所示, 数据库服务器和客户端应用程序之间通过基础网络连接, 客户端应用程序向数据库服务器请求数据, 数据库服务器则根据客户端的请求将数据返回给客户端应用程序。这种模式中, 实现商业规则的数据处理和显示功能主要放在客户端应用程序中实现, 数据库服务器仅负责数据的存储和维护以及响应客户端的数据请求。

一个典型的浏览器/服务器模式的数据库应用系统的结构如图 1-2 所示。

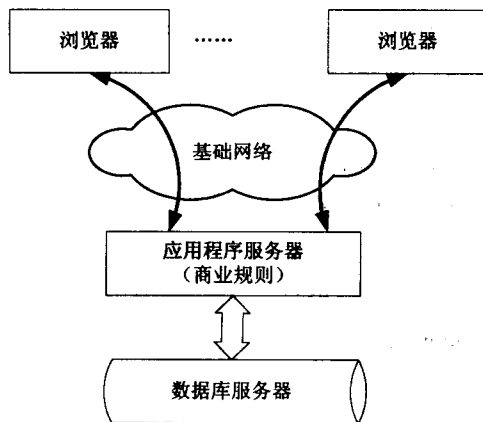


图 1-2 浏览器/服务器模式

如图 1-2 所示, 浏览器/服务器模式与客户端服务器模式最大的不同在于增加了应用程序服务器这一层, 因此该模式又称为三层模式。此外, 该模式中, 客户端使用浏览器替代了客户端应用程序, 商业规则放在应用程序服务器中实现。这种模式中, 客户端直接访问应用程序服务器并从中取得格式化数据, 客户端不再对数据进行再加工。浏览器向应用程序服务器发出服务请求后, 应用程序服务器根据该请求从数据库服务器中获得数据, 并对数据进行处理, 处理后将处理结果发回客户端。

1.2 PowerBuilder 在数据库应用系统开发中的优异表现

PowerBuilder 最初是数据库应用开发工具生产商 PowerSoft 公司推出的产品, 但在 PowerBuilder 5.0 推出后不久, PowerSoft 公司就被数据库厂商 Sybase 收购。起初 PowerBuilder 完全按照客户机/服务器体系结构研制设计, 但随着 Internet 的发展, PowerBuilder 开始提供基于 Web 的开发环境, 并在这方面做得非常优秀。PowerBuilder 采用了面向对象和可视化技术, 提供了可视化的应用开发环境。利用 PowerBuilder, 不管该应用程序是基于客户机/服务器模式还是基于浏览器/服务器模式, 都可方便快捷地开发出数据库应用程序。

1.2.1 专业的客户机/服务器开发工具

利用 PowerBuilder 可以开发出独立于服务器和数据库管理系统的应用程序。这是因为

PowerBuilder 的应用程序中对数据库访问的部分一般都采用国际化标准数据库查询语言 SQL, 这就使得用 PowerBuilder 开发的应用程序几乎可以不做修改或者只做少量的修改就能在不同的后台数据库管理系统上使用。PowerBuilder 也提供了对目前流行的大多数关系型数据库管理系统的支持, 如 Oracle、Sybase、DB2 等。

PowerBuilder 从 6.0 版开始支持所谓三层或 N 层结构。三层模式以构建分割式应用程序的能力为基础, 应用程序可分割为不同的逻辑组件, 使用这种模型和实现技术开发的应用程序, 每个组件可分配给不同的计算机以发挥最优性能。相比于传统的 C/S 模式, 这种方式具有如下优点:

- 可复用: 由于可复用, 每次开发过程中产生的组件可以在不同的应用程序间共享。
- 提高性能: 组件可配置在不同的功能强大的服务器上, 使功能不强的客户端计算机的负担降低。这样设计和配置就非常灵活, 开发人员可以充分利用服务器的计算资源, 使系统获得更好的性能。
- 易于维护管理: 组件的集中, 更便于重复利用。而且一旦需要修改, 可以更容易地进行重新配置, 这样就可以更快速的适应应用需求的变化。此外, 组件的封装和集中, 也更易于管理。

随着 Internet/Intranet 越来越广泛的使用, 用户对 Web 服务器所查询的信息不再仅仅局限于以文件方式存放在服务器端的静态超文本文件。这时就需要借助关系型数据库来存放变化的数据, 并在 Web 服务器与数据库服务器之间以 JDBC、CGI 等协议建立连接, 使 Web 服务器能够实现对数据库的实时查询, 并将结果返回浏览器。PowerBuilder 对三层结构的支持满足了开发 Internet/Intranet 应用程序的需要。在这样一个结构中, 浏览器、Web 服务器、数据库服务器就分别对应于客户端、应用服务器、数据库服务器三层体系结构中的三个环节。

1.2.2 面向对象的程序设计

软件工程中对应用程序所提出的重要目标就是软件的可重用性和可扩展性。正是基于这样一个目标, 面向对象技术才成为当前软件开发的主流。尽管在 PowerBuilder 应用程序的开发中, 面向对象技术不是必需的(这对于 PowerBuilder 的初学者来说是件好事), 但作为一种面向对象的开发工具, PowerBuilder 全面支持面向对象方法中的各种技术。面向对象方法中对象的封装性、继承性、多态性等特点使得利用 PowerBuilder 开发的应用程序具有极大的可重用性和可扩展性。

尽管理解面向对象的思想对开发应用程序很重要, 但因为在 PowerBuilder 中大量使用了组件封装技术, 因此可以这样认为, 面向对象的方法实际上就是基于组件的开发方法。当需要开发一个数据库应用时, 程序员并不需要从头开始, 而只需要利用已有的或者前面使用过的组件就可以完成应用系统的开发。用 PowerBuilder 开发的好处是, 即使程序员不懂得什么是面向对象技术, 也可能会下意识地用到面向对象技术的一些特性, 但是如果对这一技术有了充分的了解, 那么所设计程序的可重用性, 可维护性和其他各方面的质量都

会有大幅度提高。

PowerBuilder 不仅拥有在 Windows 应用程序中常见的窗口、菜单、控件等对象，还允许用户创建自己的对象。

PowerBuilder 对象有 3 部分元素：属性、函数和事件。

属性也就是数据。包括系统属性和用户定义属性两类，用于描述该对象的各种特性，如窗口对象的系统属性包括标题、高度、宽度等。用户定义的属性可以是实例变量或共享变量，对这种属性的访问与系统属性相同，这两种属性只是在封装性上有所不同。

PowerBuilder 中的事件与 Windows 的事件存在着映射关系。当用户的操作或系统本身产生 Windows 的标准事件时，系统会将其转化成 PowerBuilder 中的事件。例如命令按钮有一个 DoubleClicked 事件，该事件在用户单击按钮时触发。用户单击按钮时，操作系统首先检测到鼠标在按钮位置被单击，然后再把 DoubleClicked 这一消息转给该按钮，PowerBuilder 运行时的工作引擎翻译这一事件，并执行适当的 PowerScript 语句。PowerBuilder 中大部分的事件都可以映射成由用户激发的 Windows 事件。另一类事件是用户定义事件，这种事件由用户声明并可以在任何时刻由用户触发。

用户通过函数来对对象完成某些操作。为满足对象封装的要求，PowerSoft 公司建议用户通过调用对象事件和函数来修改对象而不要直接修改对象属性。例如采用 Window.hide() 而不要用 Window.visible=False 来使窗口不可见。

如果说一门语言具备面向对象特性，它就应当具有继承性、多态性和封装性，如果缺少其中的一个，则只能称其为基于对象的系统，而不是面向对象的系统。PowerBuilder 则很好地具备了全部 3 个特性，下面分别讲述。

(1) 继承性

在 PowerBuilder 中，窗口、菜单和用户对象是可以继承的，而其他对象则不能。如果继承了一个对象，得到的子类将具有父类的属性、实例变量、共享变量、控件、用户自定义事件、对象级函数、事件和代码(script)。也就是说当继承了一个类，就几乎得到了这个类的全部。不过要注意，不能在子类中删除任何一个继承到的特性。

在继承了祖先类后，可在子类中扩展或覆盖祖先的元素。例如代码的继承在 PowerBuilder 中默认采用扩展方式，也就是说，先执行继承到的祖先代码，后执行子类的扩展代码。当然，也可以选择覆盖祖先代码的选项，只执行子类的代码，并在代码中的任意位置调用祖先代码。例如调用祖先窗口 w_1 的关闭事件，代码如下：

```
CALL W_1::close
```

重载函数是面向对象程序设计的重要特性，即同一函数名却有着不同的参数和返回值。在运行时，系统自动寻找与执行参数相匹配的那个函数，例如常用的 MessageBox() 这个函数就有 12 个不同参数的重载函数。

对于对象的继承，PowerBuilder 中所有对象都有其共同的基类 PowerObject，从这一个基类下面分别继承图形对象(GraphicObject)和不可视对象(NonVisualObject)等。在 PowerBuilder 9.0 中，可以通过如下步骤显示 PowerBuilder 中系统对象的继承关系。