

计算机应用实用教程丛书

PowerBuilder 实用教程

宋 晔 编

北京理工大学出版社

图书在版编目(CIP)数据

ADAMS 实例教程/李军等编. —北京:北京理工大学出版社,
2002.7

ISBN 7-81045-890-6

. A... . 李... . 机械制造-计算机仿真-应用软件,
ADAMS-高等学校-教材 . TH164

中国版本图书馆 CIP 数据核字(2002)第 023013 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68912824(发行部)

网 址 / <http://www.bitpress.com.cn>

电子邮箱 / chiefedit@bitpress.com.cn

经 销 / 全国各地新华书店

印 刷 / 北京房山先锋印刷厂

装 订 / 天津武清高村印装厂

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 15

字 数 / 359 千字

版 次 / 2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

印 数 / 1~4000 册

责任校对 / 陈玉梅

定 价 / 24.00 元

责任印制 / 母长新

前 言

PowerBuilder 是美国 PowerSoft 公司开发的运行于 Windows 操作系统下的客户/服务器图形界面的分布式数据库前端开发工具，它可以快速开发出面向对象的大型数据库应用系统。本教程不仅涵盖了利用 PowerBuilder 进行数据库应用系统开发的基本方法和技术，还概述了 PowerBuilder 应用系统开发所应掌握和了解的基本理论知识和基本概念。本教程的一个重要特色就是将一个完整的应用开发实例贯穿于全书各章和附录之中，并配有大量紧密结合操作步骤的图示，通过开发实例较为全面及系统地介绍了如何利用可视化技术和面向对象等技术开发数据库应用系统的基本方法。每章和附录后均附有习题，便于读者检查及巩固所学知识。学习完本教程后，读者不仅掌握了数据库开发的基本方法，同时也基本上掌握了开发一个实际应用项目的全过程，非常适合于初学者快速掌握 PowerBuilder 的基本开发技术。本教程的论述以 PowerBuilder 7.0 为开发环境，但是所论述的基本理论与开发方法和 PowerBuilder 的版本无本质上的差异。

本书既可作为大中专院校计算机专业的教材，也可作为非计算机专业学习 PowerBuilder 的培训教材，还可供广大数据库应用系统开发人员进行参考。

编者多年来从事 PowerBuilder 的教学工作。本书就是编者在自己制作的教学电子课件的基础上，参考了国内外有关资料，对近几年教学实践进行了全面总结，充实而成的。

本书由北京理工大学宋晔、张弘编著，在编写过程中，曾得到了许多同志的热情帮助，郝彬、王振波、刘蕊等同学参加了文稿的录入及整理工作，北京理工大学计算机系的童明生教授在百忙之中审阅了全稿，并提出了宝贵意见，在此一并表示衷心的感谢。

限于编者的水平，书中难免有不妥之处，恳请读者批评指正。

编著者

2002 年 8 月于北京理工大学

目 录

第 1 章 数据库应用的基础知识

1.1 数据库应用的基础的发展和趋势	1
1.1.1 数据库应用模式的演变过程	1
1.1.2 主机应用模式	1
1.1.3 文件服务器应用模式	2
1.1.4 客户 / 服务器应用模式	2
1.1.5 文件服务器与客户 / 服务器的数据库操作	3
1.1.6 分布式计算应用模式	4
1.1.7 Web 网络应用模式	5
1.2 如何开发数据库应用系统	6
1.2.1 结构化生命周期法	7
1.2.2 快速原型法	8
1.2.3 面向对象方法	9
1.3 数据库结构的设计与实现	12
1.3.1 数据库结构设计	12
1.3.2 数据库结构的实现	15
习题	15

第 2 章 PowerBuilder 概论

2.1 PowerBuilder 7.0 简介	16
2.1.1 PowerBuilder 特性	16
2.1.2 PowerBuilder 的基本概念	17
2.2 PowerBuilder 7.0 的新特性	19
2.2.1 全新的用户接口和开发界面	19
2.2.2 数据库连接新特性	21
2.3 PowerBuilder 7.0 的开发环境	21
2.3.1 工具栏	21
2.3.2 画板	22
2.3.3 画板工作区的操作	26
习题	28

第3章 应用程序

3.1 创建应用程序对象	29
3.1.1 应用程序的基本知识	29
3.1.2 创建应用程序对象	30
3.2 设置应用程序的属性	31
3.2.1 设置应用程序的图标	33
3.2.2 指定应用程序的默认文本属性	33
3.2.3 设置应用程序的默认全局对象	33
3.3 应用程序对象的事件	33
3.3.1 Open 事件	35
3.3.2 Close 事件	35
3.3.3 SystemError 事件	35
3.4 对象库搜索表	36
3.5 运行应用程序	37
3.6 创建应用程序的开发实例	37
习题	39

第4章 窗口设计

4.1 创建窗口	40
4.1.1 窗口的创建	40
4.1.2 窗口的类型	42
4.1.3 窗口的属性	43
4.1.4 窗口的事件	45
4.1.5 窗口的函数	46
4.2 窗口控件的基本操作	48
4.2.1 添加控件	48
4.2.2 复制控件	50
4.2.3 布局控件	51
4.3 窗口控件的属性、事件和函数	52
4.3.1 PowerBuilder 控件	52
4.3.2 控件的公有特性	60
4.3.3 控件事件	61
4.3.4 控件函数	61
4.4 消息对话框	62
4.5 窗口设计的开发实例	63
习题	67

第 5 章 数据库管理

5.1 数据库的创建和删除	68
5.1.1 创建和删除 ASA 数据库	68
5.1.2 在 Sybase Central 中管理数据库	69
5.1.3 创建 ASA 数据库的开发实例	69
5.2 PowerBuilder 的数据库接口	73
5.2.1 ODBC 接口概述	73
5.2.2 ODBC 的软件层	74
5.2.3 ODBC 的配置	74
5.2.4 通过 ODBC 连接数据库	74
5.2.5 创建数据源和数据库描述文件的开发实例	75
5.2.6 连接数据库的操作	79
5.3 数据库画板及其应用	79
5.3.1 数据库画板的工具栏	80
5.3.2 数据库画板的工作区	80
5.4 数据库的操作	82
5.4.1 连接数据库	82
5.4.2 向数据库添加表	82
5.5 表的定义和数据操作	83
5.5.1 定义列属性	83
5.5.2 定义表属性	84
5.5.3 创建及设置关键字和索引	85
5.5.4 列的扩展属性	88
5.5.5 表数据的处理	91
5.5.6 创建数据库表的开发实例	93
习题	99

第 6 章 PowerBuilder 编程

6.1 PowerScript 语言的基本概念	100
6.1.1 PowerScript 书写基础	100
6.1.2 标识符	101
6.1.3 数据类型	102
6.1.4 变量、常量和数组的声明	104
6.1.5 代词	106
6.1.6 操作符和表达式	108
6.2 PowerScript 语句	109
6.2.1 赋值语句	110

6.2.2	条件控制语句	110
6.2.3	循环控制语句	113
6.2.4	其他语句	115
6.3	结构	117
6.3.1	定义结构	117
6.3.2	使用结构	118
6.4	调用函数和事件	119
6.4.1	常用的标准函数	119
6.4.2	自定义函数	120
6.4.3	编译和保存函数	122
6.4.4	调用函数和事件	123
6.4.5	调用祖先对象中的函数和事件	124
6.5	嵌入 SQL 语句与数据库操作	125
6.5.1	连接数据库的语句	125
6.5.2	SQL SELECT 查询语句	126
6.5.3	SQL 操作语句	127
6.5.4	事务提交与撤消	129
6.5.5	使用游标的语句	129
6.6	编写脚本	131
6.6.1	打开脚本视窗	131
6.6.2	脚本编辑	132
6.6.3	编译脚本	132
	习题	132

第 7 章 数据窗口

7.1	数据窗口对象	133
7.1.1	创建数据窗口	133
7.1.2	创建数据窗口对象的开发实例	133
7.1.3	数据窗口的数据源	138
7.1.4	数据窗口的显示风格	141
7.2	数据窗口画板	145
7.2.1	数据窗口画板环境	146
7.2.2	设计视窗与预览视窗	146
7.3	设计数据窗口对象	150
7.3.1	为数据窗口对象设置属性	150
7.3.2	为数据窗口对象添加控件	152
7.3.3	为数据窗口对象增加计算列或计算域	154
7.3.4	设计数据窗口对象的开发实例	156

7.4 数据窗口控件.....	159
7.4.1 数据窗口控件的属性.....	160
7.4.2 应用程序与数据库的连接.....	161
7.4.3 数据窗口控件的操作.....	164
7.5 数据窗口的开发实例.....	172
7.5.1 在窗口中放置数据窗口控件的开发实例.....	172
7.5.2 连接数据库和数据显示的开发实例.....	174
7.5.3 数据维护的开发实例.....	177
7.5.4 数据排序与过滤的开发实例.....	181
习题.....	186

第8章 菜单设计

8.1 菜单.....	187
8.1.1 菜单类型.....	187
8.1.2 菜单的组成.....	188
8.2 菜单的创建.....	188
8.2.1 菜单画板.....	189
8.2.2 添加菜单项.....	189
8.2.3 设置菜单的属性.....	193
8.2.4 保存菜单.....	194
8.2.5 通过继承创建菜单.....	194
8.3 为菜单项编写脚本.....	194
8.3.1 菜单项的事件.....	194
8.3.2 菜单项的脚本.....	195
8.4 菜单的使用.....	196
8.5 菜单设计的开发实例.....	197
习题.....	202

第9章 MDI (多文档界面) 应用程序

9.1 创建 MDI 应用程序对象.....	203
9.1.1 MDI 应用程序的组成.....	203
9.1.2 MDI 与 SDI 应用程序的比较.....	203
9.1.3 创建 MDI 应用程序的开发实例.....	204
9.2 应用继承创建 MDI 应用程序底稿.....	208
9.2.1 窗口的继承.....	209
9.2.2 继承的优点.....	209
9.2.3 在框架窗口中打开底稿的开发实例.....	209
9.3 利用窗口间传递消息打开底稿窗口.....	216

9.3.1	利用函数在窗口间传递参数	216
9.3.2	Message 对象	217
9.3.3	设置数据检索参数	217
9.3.4	窗口间传递消息的开发实例	218
9.4	应用类和实例的概念同时打开多个相同的底稿	223
9.4.1	类和实例	223
9.4.2	声明实例	223
9.4.3	同时打开多个相同底稿的开发实例	223
9.5	应用自定义函数设计菜单功能	226
9.5.1	自定义对象层函数	226
9.5.2	自定义对象层函数的开发实例	227
	习题	231

第 10 章 应用程序的调试与运行

10.1	应用程序的调试环境	232
10.1.1	调试画板	232
10.1.2	调试窗口菜单	235
10.2	调试过程	236
10.2.1	设置断点	237
10.2.2	在调试模式下执行程序	240
10.3	可执行程序的生成	243
10.4	分发应用程序	246
	习题	247
	附录 1 权限管理与密码修改的实例	248
	附录 2 应用多表的相同数据源实现同步检索的实例	262
	附录 3 应用多表的不同数据源实现同步检索的实例	275
	附录 4 生成 HTML 数据文件的实例	284
	附录 5 报表设计的实例	288
	附录 6 数据库维护的实例	298

第 1 章 数据库应用的基础知识

本章概述了应用 PowerBuilder 进行数据库应用系统开发所应该掌握和了解的有关基础知识和基本概念。通过这章的学习可以了解数据库应用模式各阶段的发展和变化、数据库应用系统的开发方法以及数据库结构的设计与实现的基本方法。

本章主要内容：

- 数据库应用的发展和趋势
- 如何开发数据库应用系统
- 数据库结构的设计与实现

1.1 数据库应用的发展和趋势

1.1.1 数据库应用模式的演变过程

PowerBuilder 是一个可视化图形界面的数据库应用系统开发工具，它支持面向对象技术、客户 / 服务器机制、分布式组件开发与因特网应用。在系统化学习这种客户 / 服务器结构的应用程序开发工具之前，首先介绍关于数据库应用的模式和演变过程。

计算机的应用结构经历了以下几个发展阶段：集中式结构、文件服务器的网络结构、客户 / 服务器网络结构、分布式客户 / 服务器网络结构和 Web 网络结构。

1.1.2 主机应用模式

在 20 世纪六七十年代，企业使用的计算机一般以大型机为主，大型机代表一种“集中式”的系统结构，由主机和客户终端组成，其主机相当于现在的服务器，如图 1-1 所示。

在集中式结构中，客户终端和主机之间传递数据的方式是用户从客户终端键盘键入的信息到主机和从主机返回到终端上的字符。

计算机的所有资源都在主机上，所有处理也在主机上完成。这种结构的优点是可以实

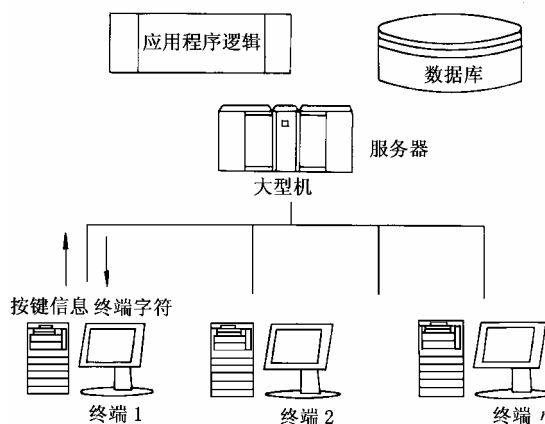


图 1-1 集中式主机结构

现“集中式”管理，确保安全性。它的缺点是应用程序和数据库都存放在主机中，难以划分应用程序的逻辑，并且计算机的费用非常昂贵。

1.1.3 文件服务器应用模式

到 20 世纪 80 年代，计算机应用的范围和领域日趋广泛，局域网和文件服务器技术相继问世。在文件服务器系统结构中，应用程序是在客户工作站上运行的，文件服务器只提供了资源的集中管理和访问途径，如图 1-2 所示。

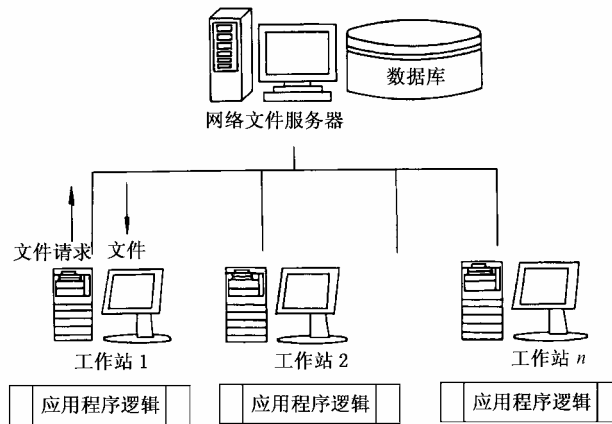


图 1-2 文件服务器结构

文件服务器结构的特点是集中管理共享数据资源，而将应用程序分散在各客户工作站上。它的优点是实现的费用比较低廉和配置灵活，在局域网中可以方便增减客户端工作站。它的缺点是：

(1) 要求客户端的计算机要有较高的性能，因为所有的应用处理都必须在客户端完成。

(2) 应用程序可以存放在网络文件服务器的硬盘上，但是在处理应用程序之前还要把它再传送到客户端计算机的内存中去执行，处理全部在客户端完成，因此网络上要经常传送大量无用的数据。

1.1.4 客户 / 服务器应用模式

由于在文件服务器系统结构中应用程序都是在客户端运行，所以与大型机的“集中式”相比虽然费用低廉，但它缺少足够的计算和处理能力。客户 / 服务器结构的问世解决了费用和性能之间的矛盾，在这种结构中应用程序分别在客户工作站和服务器的服务器上执行，既可以合理划分应用逻辑，又可以充分发挥客户工作站和服务器的性能。在客户 / 服务器结构中，我们常把客户端称作前台或前端客户，把服务器称作后台或后端服务器。客户 / 服务器结构如图 1-3 所示。

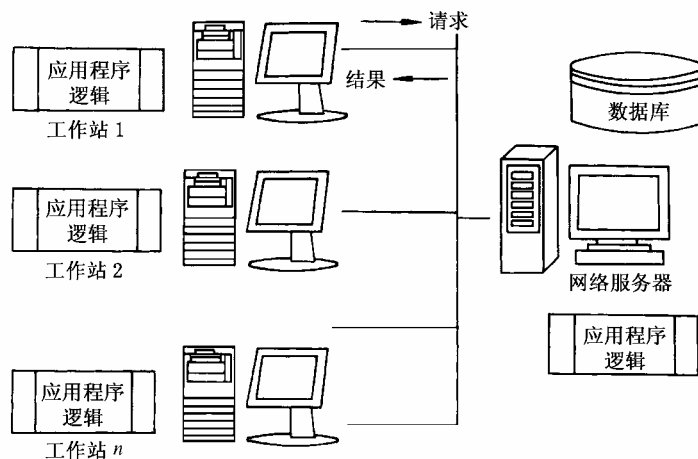


图 1-3 客户 / 服务器结构

客户 / 服务器结构和文件服务器结构两者之间的区别是：前者的客户工作站向服务器发送的是处理请求，而不是文件请求；服务器返回的是处理结果，而不是文件。

综述以上几种模式可知，大型机集中式结构的所有程序都在主机内执行，而文件服务器局域网结构的所有程序都在客户端执行，这两种结构都不能提供真正的可伸缩应用系统框架。而客户 / 服务器结构则可以将应用逻辑分布在客户工作站和服务器之间，该结构是当今较为流行的结构，其普遍应用的领域是数据库应用领域。

1.1.5 文件服务器与客户 / 服务器的数据库操作

文件服务器与客户 / 服务器是两种不同的结构，在数据库操作方面有着根本的区别。图 1-4 和图 1-5 显示了这两种结构在数据库操作上的区别。例如在数据库上都有一个含有 45 000 条历届学生记录的数据库表格 student，并且要求查询学号为 9909221 的学生的信息。

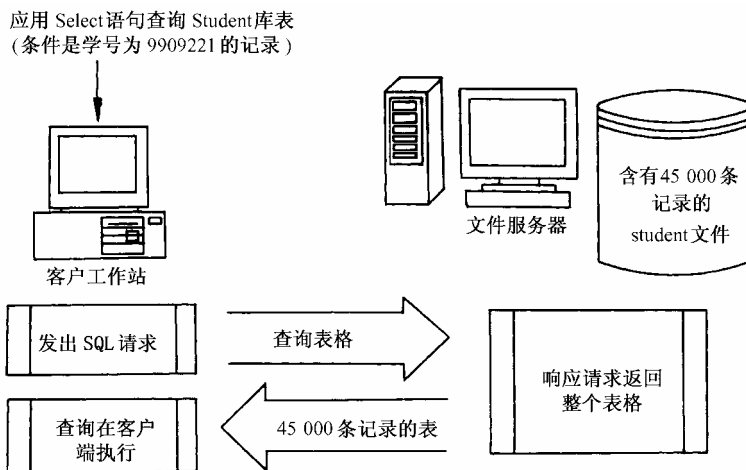


图 1-4 访问文件服务器的数据库

图 1-4 是文件服务器结构，查询是在客户端完成的。在执行 SQL 查询语句前，首先查询逻辑请求网络系统将带有 45 000 行记录的表格传送到本地的客户工作站上，然后在客户端进行处理并完成查询。如果不存在所需要查找的记录，则网络传输将做无用功。该处理方式的缺点是增加了网络线路的传输负荷，容易造成网络阻塞，增加系统的响应时间。

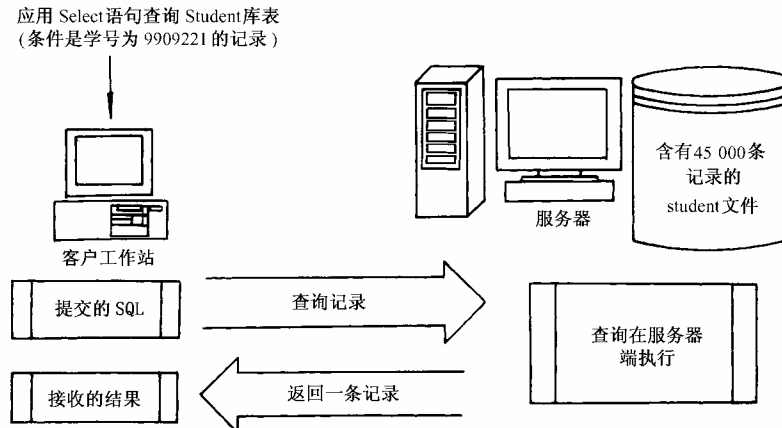


图 1-5 访问客户 / 服务器的数据库

图 1-5 是客户 / 服务器结构，SQL 查询语句将在服务器端执行，服务器发送给客户工作站的只是查询的结果。客户 / 服务器处理方式的优点是降低了网络线路的负荷，提高了系统传输的效率，避免了不必要的网络阻塞。

客户 / 服务器结构的核心是当前端用户需要后台服务器的服务时，仅需要发出请求，而服务器接受该请求后，执行相应的功能，并把满足条件的那部分数据反馈给前台客户端。

客户 / 服务器的主要特点有两个：一是由于查询和操作可以在数据库服务器上完成，因此需要配置功能强大的服务器，使所有客户工作站都可以共享服务器资源；二是数据库服务器的平台与客户端无关。数据库服务器上的数据库管理系统集中负责管理数据库服务器上的数据和资源，它向客户端提供一个开放的使用环境，客户端的用户通过数据库接口和 SQL 语言访问数据库，即客户端的用户只要能够通过网络协议和数据库接口程序连接到服务器就可以实现对数据库的访问。

1.1.6 分布式计算应用模式

传统的客户 / 服务器结构均是基于双层结构的。双层结构的物理实现方法为：个人计算机作为客户工作站，网络服务器作为后台数据库服务器。程序逻辑分布在客户机和服务器内。商业或企业规则存放在应用程序中，驻留在客户工作站或在后台服务器内。因为在这种应用模式中商业或企业规则不能被封装，而且也不能进行集中配置和管理，所以有时受到一些限制。

传统的客户 / 服务器结构存在的一个重要问题就是在网络计算机系统的中心位置不可能插接应用程序组件。而在很多企业应用中，由于业务逻辑的要求，经常需要为不同的客户机插接不同的应用程序。为了满足企业应用的需求，出现了一种三层客户 / 服务器模型，如图 1-6 所示。这种模型中有三个逻辑层，客户层是面向用户服务的，服务器层是面向数据服务

的，中间层是面向商业或企业规则的（满足企业业务逻辑要求的应用程序组件）。

这种多层客户 / 服务器结构又称做分布式客户 / 服务器结构。它强调组件开发，将原来很多客户端的处理逻辑剥离出来，形成相对独立的组件模块，它们被存放在服务器上，供所有的客户端应用程序访问。这个方案的优点有如下几点：

- (1) 组件是不同的应用程序重复使用的，即组件共享。
- (2) 组件的配置和设计的灵活性，使计算的性能得以改善。
- (3) 组件开发使复杂的应用程序变为易于管理的模块。

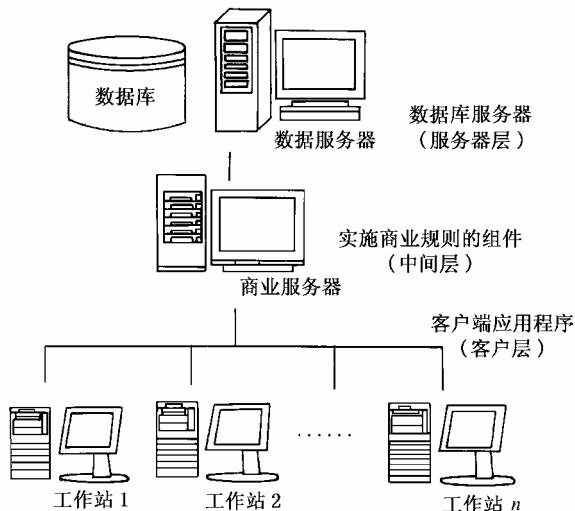


图 1-6 三层客户 / 服务器结构

1.1.7 Web 网络应用模式

Internet 是个全球性的计算机网络系统,它通过采用共同的网络通信协议使分布在世界各地的计算机系统及网络用户可以在不同的网络和操作系统间进行数据交换。World Wide Web (WWW) 是 Internet 上的最为流行的信息服务。通常也将 WWW 称为 Web, 用户通过计算机上的 Web 浏览器 (IE 等) 软件来浏览 Web 服务器上的信息。

在传统的 Web 服务器中,文本和其他多媒体信息都是以文件的形式来进行存储和管理的,随着信息量的不断增加,系统的速度等性能受到越来越多的影响。另一方面,WWW 的应用领域在不断拓展,静态的 Web 页面也越来越不能满足对信息服务的动态性、实时性和交互性的要求。因此,将 Web 技术与数据库技术相结合,开发动态的 Web 数据库应用势在必行。

实现 Web 数据库的最常用方法与分布式计算应用模式中介绍的组件方法非常类似,它是在 Web 服务器端提供中间件来连接 Web 服务器和数据库服务器,其客户端浏览 Web 服务器信息的过程如图 1-7 所示。

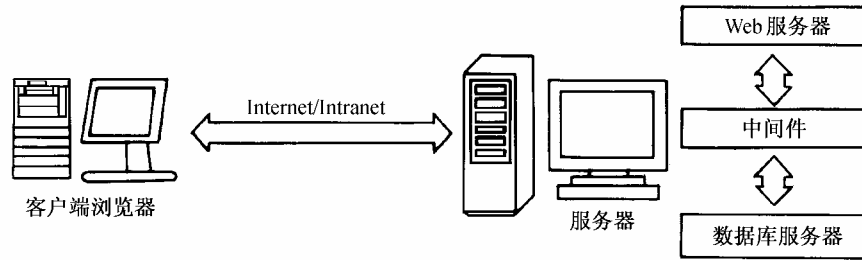


图 1-7 使用中间件间接访问的 Web 数据库模式

图中：

- 客户端通过因特网(Internet)的 Web 浏览器(如 IE)来浏览 Web 服务器上的信息。
- 由中间件把 Web 服务器所接收到的请求转化为数据库服务器所能识别的请求信息。
- 通过中间件将数据库中的数据信息转换为客户端能够访问的信息。

中间件负责管理 Web 服务器和数据库服务器之间的通信并提供应用程序服务,它能够直接访问数据库、调用外部程序或利用程序代码来访问数据库。因此它可以提供与数据库相关的动态 HTML 页面或执行用户查询,并将查询结果格式化成 HTML 页面,然后通过 Web 服务器返回给用户浏览器。

在 Web 服务器端除了采用中间件以外,还可以通过 Web 浏览器把应用下载到客户端运行,在客户端直接访问数据库。客户端应用包括:Java Applet、ActiveX 等等,其中最典型的的就是 Java Applet。利用 Java Applet 不仅可以方便地实现与用户的交互,还能提供丰富的图形功能、声音、视频等多媒体功能。特别是由于 Java 是一种与平台无关的编程语言,因而具有较强的可移植性。如图 1-8 所示。

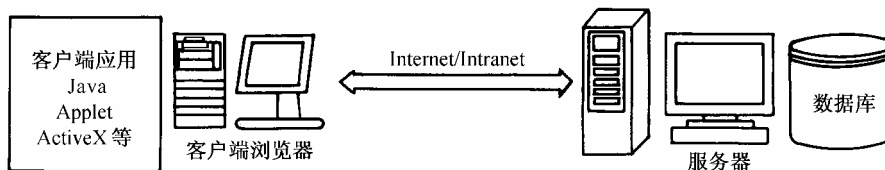


图 1-8 直接访问的 Web 数据库模式

1.2 如何开发数据库应用系统

本节将介绍有关软件工程知识范畴内的一些数据库应用系统的开发方法;使用 PowerBuilder 进行数据库应用系统快速开发的特征及方法。信息系统的开发方法主要可以分为结构化生命周期法、快速原型法和面向对象法。

通常软件系统根据需求不同被分为两类。一类需求是需要比较稳定而且能够预先指定的系统,称之为预先指定的系统。例如,传统工业生产过程的计算机控制系统、卫星图像处理系统、火箭发射跟踪控制、数据库管理系统之类的系统软件,通常都可归类为预先指定的系统。开发这类系统应该预先进行严格的形式化的需求分析,制定出精确的需求规格说明,并

在严格的管理下采用传统的生命周期法进行系统开发。另一类需求是模糊的或随时间变化的系统,通常在系统安装运行后,还会由用户根据需求进行动态修改,称之为用户驱动的系统。例如,一般商业和行政的数据处理系统、决策支持等系统通常都属于用户驱动的系统。开发这类系统需要应用一种适合进行反复试探的技术。这类系统必须具有能够快捷并简便地进行调整的特性,便于在运行使用的过程中能够及时根据用户需求的变化相应地修改系统。若应该归类为用户驱动的系统而使用传统方法进行开发,即使花费大量人力和物力去分析确定的需求,其结果也不能反映用户的真实需求或所开发出来的系统早已过时。因此,对于用户驱动的系统适合采用快速原型法进行开发。

1.2.1 结构化生命周期法

这是基于软件工程开发概念的一种成熟的信息系统的系统化开发方法,它的基本开发准则是把系统开发看做是工程项目,也就是说需要经过用户需求、可行性分析、立项批准、设计、施工(编程)、验收和交付使用等过程。信息系统的生命周期可以分为如下五个阶段:

1. 系统分析阶段

(1) 系统调查及可行性分析。了解信息系统用户的组织机构、业务范畴以及新系统的目标,并且做出可行性(经济可行性、技术可行性和使用可行性)分析。

(2) 需求分析。了解企业业务需求是设计数据库和信息系统的基礎。数据需求是要明确用户要从数据库中获取哪些数据,什么样的数据是基本存储数据,什么样的数据是导出的数据。同时还要了解用户对数据安全性和完整性等方面的需求;处理需求是要明确用户要完成什么样的处理功能和对处理的响应时间和处理方式有什么特殊的要求。

2. 系统设计阶段

(1) 概要设计(总体设计)。总体设计是把用户的信息要求统一到一个整体的逻辑结构中,该结构既能表达用户的要求又能独立于任何硬件和数据库管理系统。该阶段在应用程序方面要完成子系统的划分和功能模块的划分,在从数据库设计方面要完成概念模式的设计。

(2) 详细设计(模块设计)。这一步也包括数据库设计和应用程序设计两大部分。

3. 系统实现(编程)阶段

建立数据库并装入原始数据,建立存储过程,编写和调试应用程序代码等。

4. 系统调试与运行(测试)阶段

在局部编码和测试后,把各子系统与各模块进行联合调试、调试与试运行以听取用户意见为主,对运行效果进行评估,修改系统错误和改进系统性能。

5. 系统运行、评价与维护(运行)阶段

在交给用户使用的过程中可能还会出现新的问题和提出新的需求,此阶段要不断对系统进行评价和维护。只要数据库系统存在,就要不断进行评价、调整、修改,直至数据库生命

周期结束。

综上所述，结构化生命周期法这种信息系统开发方法的最大优点是：每一步的开发目标都很明确，易于项目管理。但是，该方法的最大缺点是用户只参加了需求分析阶段的工作，对即将建立的新系统缺乏直观的效果确认，有可能导致开发的系统偏离用户最初的需求目标。

1.2.2 快速原型法

利用传统的生命周期法开发某种类型的应用系统时并不成功，所开发出的系统往往不能真正满足用户的需要。为了克服传统开发方法的缺点，人们在实践中逐渐创建出了快速原型法和面向对象的方法的这些软件工程的新途径。下面来介绍快速原型法。

要开发用户驱动的系统，就要从常规的程序设计方法和不易修改的生命周期法飞跃到一种快速、灵活、交互式的软件开发方法上，它能够使所建立的应用系统符合用户的需要。用户通过在计算机上实际运行和试用原型系统而向开发者提供真实的反馈意见。这种新的软件开发方法就是快速原型法，简称为原型法。原型法打破了传统的自顶向下的开发模式，是目前较为流行的开发模式。

快速原型法的基本思想是依据原型进行快速开发，对存在的问题进行反复的修正，直至形成用户满意的系统，如图 1-9 所示。

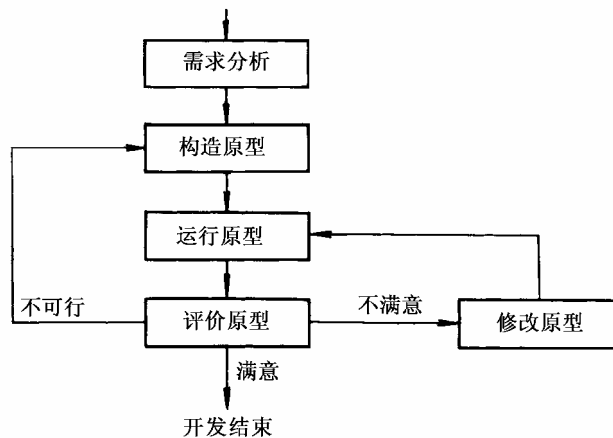


图 1-9 快速原型法的开发过程

快速原型的开发过程为：

- (1) 系统基本需求的确定。
- (2) 对原型的功能选择。
- (3) 原型的构造与试用。
- (4) 原型的修改和完善。

若构造的原型存在性质上的问题根本不可行，则重新构造原型；若构造的原型可行，但是存在质量上的问题，则必须修改原型并重新运行原型。

原型法基本反映了最终系统的基本功能和基本特征，它适用于快速开发一个演示系统。