

计算机软件开发系列丛书

L. John Ribar and Steven Nameroff

PowerBuilder 4.0 软件开发工具

THE POWERBUILDER CONSTRUCTION KIT

Using PowerBuilder to Design and Customize Applications

COVERS

Version 4



希望

McGRAW-HILL
学苑出版社

计算机软件开发系列丛书

PowerBuilder Construction Kit

PowerBuilder 4.0 软件开发工具

L. John Ribar

Steven Nameroff

章立生

燕卫华

著

译

审校

学苑出版社

(京)新登字 151 号

内 容 提 要

本书简要说明了 PowerBuilder 的功能,详细介绍了如何使用 PowerBuilder 构造数据库应用程序、增加帮助文件、进行测试和调试、建立函数库等,并给出了脚本语言 PowerScript 中所有函数的快速参考。

本书内容全面、叙述清楚、图文并茂,是一本 PowerBuilder 数据库应用软件设计的优秀参考书,可供数据库应用软件设计者和软件爱好者参考。

需要本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

版 权 声 明

本书英文版名为《PowerBuilder Construction Kit》,由 McGraw-Hill 公司出版,版权归 McGraw-Hill 公司所有。本书中文版由 McGraw-Hill 公司授权出版,未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

计算机软件开发系列丛书

PowerBuilder 4.0 软件开发工具

著 者: L. John Ribar Steven Nameroff
译 者: 章立生
审 校: 燕卫华
责任编辑: 甄国宪
出版发行: 学苑出版社 邮政编码: 100036
社 址: 北京市海淀区万寿路西街 11 号
印 刷: 双青印刷厂
开 本: 787×1092 1/16
印 张: 12.75 字 数: 279 千字
印 数: 1~5000 册
版 次: 1994 年 7 月北京第 1 版第 1 次
ISBN7-5077-0779-2/TP·11
本册定价: 23.00 元

学苑版图书印、装错误可随时退换

作者简介

L. John Ribar 是一名程序员,他编写了几本 C 语言程序设计方面的畅销书,包括《BYTE's Windows Programmer's Cookbook》,《C DiskTutor》和《FORTRAN Programming for Windows》。Ribar 是 Picasso Software Group 的董事长,该软件开发公司位于宾夕法尼亚州约克郡,专门从事客户应用程序和编程工具的开发。

Steven Nameroff 是《Quick BASIC: The Complete Reference》的著者和《WordPerfect 6: The Complete Reference》及《Turbo Pascal 7: The Complete Reference》的著者之一。Steven 是 Rational Software 公司的顾问,该公司使用面向对象的技术专门从事复杂的软件开发。

致 谢

这确实是个有趣的旅行。在十几年的兼职工作并坚持软件设计和开发的专职工作之后,一年前我开始自己在软件开发方面投资。这完全改变了我的整个前途和实际生活!

感谢我的好朋友和编辑 Jeff Pepper,正是他的帮助使我有机会不断为 Osborne/McGraw—Hill 编写著作,事实上他连续给了我几个选题。这是我要完成的几本书中的最后一本,也是我最费心写的一本不论是从体力(脑力还是语言上都是如此)。为什么?因为在本书出版之时,业务也在发展。

Jeff 与 Ann Wilson 一起组成了一个编辑组,为这本书的完成做了大量工作。我还特别感谢 Steven Nameroff,他及时给我很大的帮助,我们的合作非常愉快!

感谢 Greg Keesey 和 Mark Lindsay,他们以超乎寻常的耐心等待直到本书的完成。

我一如既往地衷心感谢我的妻子 Deborah 以及我的孩子们(Louis, Jamie, Michael 和 Leah),他们忍受了我这么长时间的离开。我想 Deb Drah 比我为这本书付出了更多的心血!

目 录

第零章 引言	1
0.1 本书的对象	1
0.2 本书内容	1
第一章 工具的基础知识	3
1.1 PowerBuilder Enterprise	3
1.2 Enterprise 绘写工具(Painter)	3
1.3 成功的蓝图	6
1.4 长远目标:客户-服务器模型	7
第二章 数据基础	8
2.1 早期的数据库	8
2.2 自动化数据库	9
2.3 设计数据库	10
2.4 小结	13
第三章 工作环境	14
3.1 Windows 基础——可视化外观	14
3.2 面向对象的程序设计	19
3.3 小结	21
第四章 基础	22
4.1 程序设计简介	22
4.2 Application Painter	23
4.3 小结	27
第五章 构造数据库	29
5.1 数据库设计的基础知识	29
5.2 连接到一个已存在的数据库	36
5.3 小结	37
第六章 用菜单和工具条组织	38
6.1 从生成的菜单开始	38
6.2 修改菜单	40
6.3 小结	46
第七章 与数据交互	47
7.1 建立 DataWindow	47
7.2 设计自由格式 DataWindow	49
7.3 设计一个网格 DataWindow	53
7.4 其他信息	55

第八章 设计窗口和对话框	56
8.1 从最简单的窗口开始.....	56
8.2 从头开始建立一个简单窗口.....	59
8.3 使用 DataWindow	61
8.4 MDI 窗口	63
8.5 合并.....	64
第九章 定义程序如何动作	65
9.1 理解术语.....	65
9.2 格式化代码.....	68
9.3 声明变量和常量.....	73
9.4 流控制.....	74
9.5 重复处理.....	76
9.6 使用 PowerScript Painter	79
9.7 实现应用.....	79
9.8 小结.....	85
第十章 增加一些复杂性	87
10.1 报表	87
10.2 漂亮的窗口	91
10.3 小结	96
第十一章 为用户增加帮助	97
11.1 帮助机制	97
11.2 项目计划.....	101
11.3 使用 PowerBuilder 中的帮助文件	101
11.4 RoboHelp	102
11.5 小结.....	103
第十二章 使用和共享库	104
12.1 什么是库.....	104
12.2 使用 Library Painter	105
12.3 使用其他人的库.....	109
12.4 小结.....	109
第十三章 建立最终发行版本	110
13.1 为自己或办公室创建发行版本.....	110
13.2 建立应用进行分发.....	112
13.3 使用资源文件.....	114
13.4 小结.....	114
第十四章 用 ERwin/ERX for PowerBuilder 设计数据库	115
14.1 实体与关系.....	115
14.2 给现有数据库增加新的生命.....	117

14.3	小结.....	118
第十五章	测试与调试	119
15.1	测试.....	119
15.2	调试.....	121
15.3	小结.....	123
附录 A	PowerScript 函数快速参考	124

第零章 引 言

欢迎使用新一代的应用软件开发工具 PowerBuilder。你是程序员吗？你不必是程序员就可以使用 PowerBuilder；你是程序员的管理者吗？或许你现在就可以开始自己的应用。但是这本书是为什么人写的呢？这本书能给你提供些什么呢？

0.1 本书的对象

你可能属于下面的几个类别，从而这本书是你的最佳选择。

- 想得到一个好的客户-服务器数据库工具，但不能肯定使用哪个工具。本书介绍有关 PowerBuilder 的基础知识，说明构造应用程序的简单性及通过脚本语言得到的功能。
- 刚购买了 PowerBuilder，但不知道何从下手。从这里开始，首先得到概貌性的知识，然后如果需要更详细的信息，阅读随软件的手册。
- 公司使用 PowerBuilder，但你还没有使用。你不想从软件手册来学习 PowerBuilder，但认为好的背景会有助于将来的工作，或者有助于与频繁使用该软件的其他人打交道。
- 编程管理人员，寻找一种快速的方法来生成客户-服务器数据库应用程序而不用一直与工时昂贵的程序员联系。
- 程序员，正寻找说明这些新的工具如何工作的一个概述；想学习有关脚本语言的一些知识，或者刚开始一个新的项目，且想在开始实际编程之前有一个好的产品概述。

0.2 本书内容

首先，本书提供最新的 Powersoft 应用软件开发环境 PowerBuilder 的功能概述。介绍如何设计应用软件，如何开发数据库，如何建立实际的应用软件，包括菜单、对话框和报表。

其次，说明作为 PowerBuilder Enterprise 系列的组成部分有哪些其他的支持。这些是完成项目所需要的，包括帮助文件、调试、建立项目文件和使用函数库以允许使用同事已做的工作。

第三，学习 PowerScript 语言，它是一种可以用来增强 PowerBuilder 应用的简单语言。附录提供可以在脚本中使用的所有函数的快速参考，但是不要性急，可以逐步学习，在以后需求和能力增长时积累知识。

我们仍需要程序员，他们是带给 PowerBuilder 魔力的人。但是使用这样的工具，你可以构造程序的第一个版本，看菜单和对话框如何工作，数据如何输入和显示及给最终用户提供什么功能。

然后，如果需要更多的帮助，可以使用 PowerBuilder 系列的工具，图形化地说明你的期

望,甚至启动一个运行的程序。

下面是本书内容的提要。

在第一章中,学习 Powersoft 的产品及客户-服务器技术的概念。在第二章中进一步学习客户-服务器数据库概念。

第三章继续讨论 Windows 程序设计和面向对象技术的使用。这些主题对充分利用 PowerBuilder 是十分重要的。在第四章中,通过构造第一个应用对象开始学习 PowerBuilder 的实质。

在第五章至第八章中,开始学习设计和开发自己的应用数据库,用菜单、工具条、Data Windows、单选按钮及列表和对话框组织它。

第九章介绍 PowerScript 的术语和语言,并说明脚本、函数、变量和属性如何一起工作来实现一个应用程序。

第十章使用更先进的数据窗口、窗口对象和窗口控件来给应用软件的外观增加一些复杂性。

在第十一章中学习为应用的用户增加帮助;第十二章讨论函数库的使用和创建,这是再利用在每个项目中开发的对象和代码的两个非常重要的主题。在第十三章,学习建立最终的应用及如何准备将它分发给用户;在第十四章中学习应用程序的调试(消除任何可能的缺陷)。

第十五章介绍另一个产品,它与 PowerBuilder 一起工作。ERwin/ERx for PowerBuilder 给开发环境增加扩展数据库设计能力,并完全与 PowerBuilder 环境结合在一起。

最后,附录提供有关在 PowerScript 中可用的全部函数的信息。这些脚本是如何给应用程序增加更多功能的手段,且通常不需要程序员就可以使用。

第一章 工具的基础知识

欢迎阅读这本介绍用 PowerBuilder 构造应用程序的专著。你要学习如何建立自己的数据库应用和 Windows 程序。想象仅在几年前,程序设计完全由专门培训过的 C 语言程序员在 Windows 中进行,他们必须非常努力地工作来设计甚至最简单的程序。

时代已经变化,PowerBuilder 是未来浪潮的一部分。在这一章中介绍在 Powersoft Enterprise 中提供的工具集;第二章“数据基础”中介绍数据库设计。在后面几章中,通过构造一两个实际应用程序进一步讨论这些工具。

1.1 PowerBuilder Enterprise

Powersoft 公司的 PowerBuilder 是新一代的应用开发工具之一。这些工具使非程序员能够以一种简化的方法建立他们自己的应用程序,通常不需要编程能力。这是程序如何构造的一个重大进步。以前,通常需要等待数月或数年才能得到需要的程序或现有软件需要的特定功能。

Powersoft 提供大量的工具帮助你的应用构造任务。有几个不同层次的软件:

- PowerBuilder 软件包在建立新应用程序中很有用。它是生成客户-服务器数据库应用的面向对象开发工具,形成 Enterprise 软件包的基础。该软件包要在本书中进行全面讨论,并含有这里提到的所有其他工具。
- PowerMaker 是一个个人工具,使用一种基于表格的方法,对使用现有数据库建立报表、查询和图形来观看商务信息是十分有用的。
- PowerViewer 是一个个人工具,用于通过查询和报表观看信息,利用由其他人创建的数据库和表格。
- WATCOM SQL Database 是一个局部化的单用户数据库管理软件包,集成到 Powersoft Enterprise 中,允许用户开发自己的应用程序和测试客户-服务器应用,而不必访问实际的数据源。WATCOM SQL 也有多用户版本。
- PowerBuilder Enterprise 是完整的工具包,它包括上面列出的所有工具的功能,是本书的基础。

首先我们讨论 Enterprise 的各个工具,用户可以使用这些工具用图形建立新应用程序的许多部分。

1.2 Enterprise 绘写工具(Painter)

绘写工具是 PowerBuilder 中独立的程序,允许用户创建(或绘制)应用程序的每个部分。之所以称它们为绘写工具,是因为它们的可视性和界面(见图 1.1)。这些绘写工具要在以后几章作深入讨论,这里只给出它们的一般介绍,使你对它们有个初步印象。

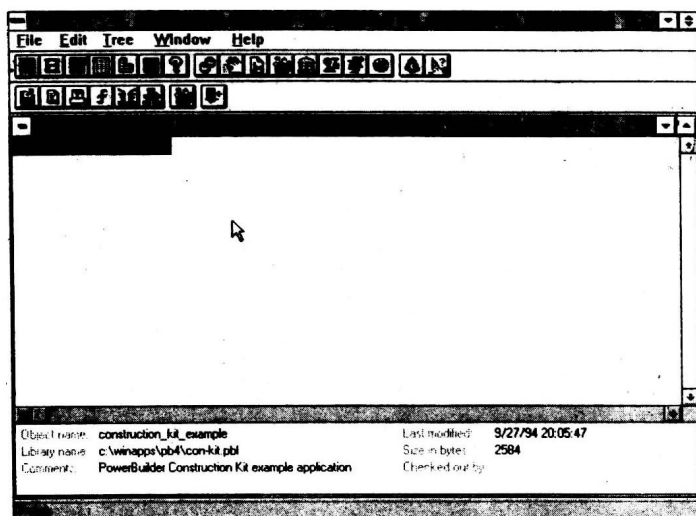
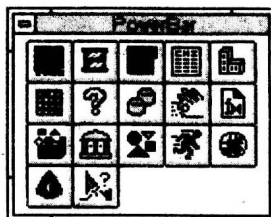


图 1.1 Application Painter 允许用图形生成一个新程序

通过简单地选择某个 PowerBar 按钮(下面给出),可以从打开的 PowerBuilder 屏幕使用所有的绘写工具。每个按钮对应一个特定的工具。



Application Painter

Application Painter 用于整体应用,它包括应用软件的层次结构,显示从使用的图标到用于实现每个特定功能的脚本(编程代码)。不用担心,不必是个程序员才能写脚本,读者在这本书中就会学习如何写脚本。

Window Painter

一旦在 Application Painter 中为应用定义了一个提纲,就可在 Window Painter 中定义特定的窗口。这些窗口是用于与程序的用户接口的方法,可以包括按钮、滚动列表、单选按钮和复选框及 DataWindows(在下一节“DataWindow Painter”中说明)。

Menu Painter

要新的应用内移动,定义一个或多个菜单。菜单中的每一项用于调用一个脚本或在定义的窗口和其他对象内移动。Menu Painter 还允许建立工具条,它是一种使用小的按钮(在

每个按钮上有一个图标定义按钮要做什么)来与用户接口的图形菜单。这允许更常用的功能按图形来选择,而不是通过多次菜单选择来进行。

DataWindow Painter

DataWindow 是在 PowerBuilder 应用软件中使用的一种特殊对象。DataWindow 是有智能的,它们知道如何做显示数据库信息所必需的动作。根据应用的不同,可能从来不用写任何代码就可访问、显示和报告数据库中的信息。

Structure Painter

如果计划写脚本来处理数据,有时候将数据一起包装到一个结构中会更简单些。结构含有几个变量,常常是不同类型的,并允许同时使用全部数据并使之组织好,而不必独立地指定每个变量。Structure Painter 允许建立这些结构,且用户可以在 PowerScript 中使用它们。

Preferences Painter

Preferences Painter 用于定义用户在新的用户软件执行过程中想要的优先项。这些优先项包括数据源(应用程序在哪里查找数据),缺省函数库和窗口设置优先项。

Database Painter

Database Painter 用于定义数据要求,它是定义整个应用的基础。在 Database Painter 中,设计数据文件、关键字(定义数据的排序顺序)及数据文件之间的关系。一旦定义了这些,其他对象就使用这个设计。

在第二章“数据基础”中,要进一步学习如何定义数据库。以后在第十四章“用 ERwin/ERX for PowerBuilder 设计数据库”中学习另一个由 Logic Works 公司设计的另一个称为 ERwin/ERX 的工具,它与 Enterprise 系列一起工作来图形式地设计数据库。

Query Painter

因为并不总是观看全部数据,因此要定义一些查询从数据库得到特定的信息,使用 Query Painter 定义这些查询就十分简单。只要选择想观看的信息的字段和重要的记录,Query Painter 生成必需的 SQL 代码。

Report Painter

Report Painter 允许图形式定义数据的报告,可以容易地布局页标题、表格头和数据及构造总和与子和。

PowerScript Painter

在有了 PowerBuilder 的工作的进一步了解之后,可能要给应用增加更多的功能。对于找不到的函数,可以使用 PowerScript Painter。这个工具允许你写可以在整个应用中使用的脚本(类似于编程代码)。

Function Painter

在开始为应用写脚本时,可能发现要为每个应用写一些类似的函数。Function Painter 允许将这些函数进行包装以便再使用,从而可以使它们用于每个新程序而不必重写任何东西。

Library Painter

Library Painter 是 PowerBuilder 环境中对代码管理的开发小组的回答。库的概念涉及一次只有一个开发人员能够改变原代码的要求。Library Painter 允许代码由一个开发人员签出,然后在修改完成时再签入回来。另外,为你设计的每个应用维持一个特定对象或脚本的版本。

另外,库可以不只是含有脚本。你设计的觉得在其他应用中有用的任何对象都可以存放在一个库中,这些对象可以包括 DataWindows 菜单和结构。

User Object Painter

User Object Painter 从现有的 Windows 控件对象创建用于 PowerBuilder 的对象,它包括使用某些 VBX 控件的能力(像在 Visual Basic 应用中使用的一样)。

1.3 成功的蓝图

现在有了可以使用 PowerBuilder Enterprise 中有哪些类型的工具的认识,但是如何创建第一个应用呢?它并不是想象的那样困难。事实上,按 Powersoft 的观点,只需十步就可完成这个工作。

1. 建立应用对象。使用 Application Painter 建立应用对象,这是所有其他对象和设计要放入的容器。
2. 设计用户界面。对于这一步,要利用 Window Painter 和 Menu Painter,在完成时将这些对象加入到应用中。另外,高级用户对非 Powersoft Windows 控件和 VBX 文件会使用 User Object Painter。
3. 定义创建的对象的行为。使用 PowerScript Painter,编写使每个按钮和菜单项按要求工作所需的脚本。
4. 将数据加到应用。对这个过程有两步。首先,需要使用 Database Painter 设计数据库,或者需要发现程序要利用的已存在数据库的更多信息。在任一种情况下,下一步都要使用 DataWindow Painter 定义希望如何观看和访问数据,可能要将最后的数据窗口放到用户界面窗口中。
5. 使用 Report Painter 生成报告(这一步仅在要求报告时才使用)。产生一些报告,用户可以在屏幕上或在打印机上观看这些报告。
6. 给应用增加帮助信息(这也是可选的,但如果加入帮助,则程序看起来更加专业了)。增加帮助还避免了用户在不知道程序该如何动作时不断给你打电话的情况。再次使用 PowerScript Painter 生成帮助。

7. 使用 Library Painter 以使用和记录你创建的或从以前应用使用的各个对象。使用 Library Painter 可以为所有其他人更好地管理这些可重利用的对象。
8. 使用 Debugger 调试应用程序。有错误吗？没有程序员？这可能决不会发生，但如果你发现一个功能或菜单项动作很奇怪，则可使用 Debugger 查找原因。或许这是一个简单的单词拼写错误，或是其他某个小故障。Debugger 是单步执行应用查找任何问题的工具。
9. 将应用软件提交给用户。使用 Application Painter，可以进行让应用最后发布所要求的所有最后调整。将已做的所有工作联系在一起。
10. 文档。你可能不是程序员，可能不习惯写作，可能不想为新的应用程序编辑文档。但是如果不写文档，就会有招来更多的电话求助，且有可能忘记在应用中为什么那样做。为了解决这些问题，编写文档描述应用程序如何工作，使用什么数据库文件，不同的菜单项应做什么，如何安装和运行程序等。

这不是应用开发的唯一蓝图，但它很简单，易于理解和遵守。在这本书中，你会看到这十步计划投入实施，并有使用自己的方法的地方。

在开始之前还有一件事情要理解：你究竟要创建什么？

1.4 长远目标：客户-服务器模型

许多年来，数据库一直在大型计算机上运行，没有哪个人实际知道数据库是如何工作的，且灵活性也不够。

近年来这种情况正在发生变化，更大的数据库系统已变得更灵活，许多数据库系统已到了个人计算机级。问题是如何一起使用这些工具。

原来，PC 数据库工具只允许单个用户使用一个数据库文件。如果你需要一个报表或要查找某些信息，必须呼叫有该数据的人。当有网络时，数据库文件可以在几个用户之间共享。

下一步发展是客户-服务器数据库。在客户-服务器环境中，一台计算机作为数据库服务器，所有那些希望访问信息的计算机是客户。服务器只管理数据库，而客户给服务器发送对信息的请求（查询），然后接收具有合适数据的回答。这是一种功能很强的解决方案，允许专门的一部分软件维护数据，同时允许任何人通过远程命令访问。

PowerBuilder Enterprise 是一个客户-服务器开发工具。你要建立一个数据库（服务器），但几乎总是生成用户（客户）应用访问该数据库。数据库可以是随 Enterprise 软件包所含的 WATCOM SQL 数据，或是其他数据库。

现在已有了足够的概括知识，接下来我们开始讨论数据库。

第二章 数据基础

在我们开始构造 PowerBuilder 系统之前,有几个需要学习的技术。在这一章中学习有关数据库技术,在下一章中学习面向对象的技术。如果你已经熟悉这些技术,可以直接跳到第四章“基础”。

2.1 早期的数据库

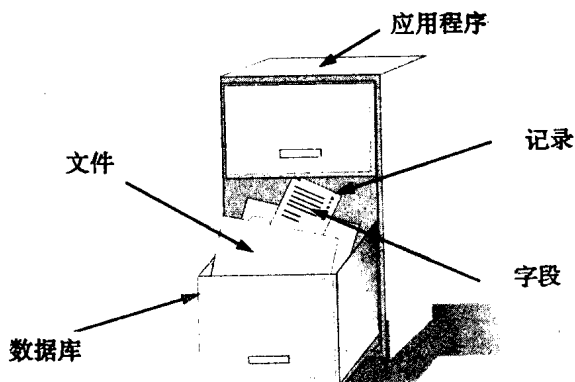
你可能在非常年轻时就了解了有关数据库技术。如果你(或一个朋友、兄弟或姐妹)有一组棒球卡,你的卡集可以看作是个数据库,你的卡如何组织?或许它们是按卡片背面的卡片号排序,或者按队名然后按每一队中球员的名字的字母排序。

当然,你收集的第二年发生什么呢?现在对大部分球员有两张卡片。卡片编号也重复了。你是按年、按球队、还是按球员的名字将这些卡片放在一起?

不管是否相信,所有这些技术都与数据库技术有关。在开始更详细的解释之前,我们首先介绍一些基本的数据库术语和概念。

2.1.1 术语

应用程序是你负责的任何工程或系统,它可以是个记帐系统、是建筑成本跟踪系统,或是个商店的销售点系统。应用也可以指为你处理数据的计算机程序。如果你现在在一个办公环境中,你的应用可能存放在一个文件框中,如下图所示:



数据库是信息的集合。它是对一个指定的应用维护所需的全部信息。例如,对一个记帐应用,需要保持有关可支付的帐号(你欠钱的人)。可收付的帐号(欠你钱的人)和帐号(你如何组织钱款在你的业务中流进流出)的信息,所有这些信息放在一起就是个数据库。在你目前的办公室中,一个数据库可以看作是你的文件柜中的一个抽屉,如上图所示。

继续这种比拟,你的数据库由文件组成。文件,在某些领域也称为表,是数据库包含的信息组。如果我们再次谈论记帐系统,你可能对可收付帐号信息有一个文件,对可支付帐号信

息有另一个文件,第三个文件用于客户信息(当然,在一个记帐系统中还涉及许多文件,但这只是个图示)。使用文件柜例子,文件可以是每个抽屉中的特定文件夹(即可支付帐号文件夹、客户文件夹等)。

在每个文件夹中是一个或多个记录。在同一文件中所有记录有相同的格式,但每个记录中的信息是不同的。例如,一个人事记录包含关于一个雇员的信息:名字、地址、社会保险号、工资、预扣税等。但是,尽管每个雇员的信息的格式是相同的,每个记录保持的信息却是不同的(即不同的名字,不同的地址等)。在文件柜中,一个记录是文件夹中的单张纸。

文件记录中的每一块信息称为一个字段(在某些领域中称为表栏)。如果你的数据库系统使用文件,它可能使用字段。如果它使用术语“表”,则字段可能称为“表栏”。每个字段是个特定信息(雇员名、地址、工资等)。

在了解一些数据库术语之后,让我们看一下棒球卡片集如何实际是个数据库应用。

该数据库是你拥有的所有卡片的集合。你可能为每个队,每一年保持文件,或者使用其他的方法进行划分和排序卡片。

文件中的每个记录是单个卡片。字段当然是存放在每张卡片上的信息(球员的名字、球队、统计值等)。

有了这个信息及这么多年的数据库经验,你就可以开始学习关于数据库设计及数据库如何在计算机上管理的更多知识。

2.2 自动化数据库

棒球队卡片集的一个问题是,如果你想重新组织卡片要花费许多时间。如果所有卡片按字母顺序排列,没有什么方法来说明你是否在该集合中有全部卡片。因此你有两种选择。

首先,你可以物理地将这些卡片按数字顺序排放。如果丢失任何卡片,你可以见到这些卡片从堆中丢失。这样你可能保持一个丢失卡片的清单,并在以后某个时候需要时检查。

第二种方法是使用一个检查表。许多棒球卡片系列带有一个或多个检查表卡,它列出所有可用的卡片。这里,你可以按保持它们的方式检查卡片集中的全部卡片,并在发现它们时在检查表上标出。

在计算机中,这个过程可通过使用索引文件极大地简化。在一个数据文件中,信息被存放一次,通常按它放到数据库中的顺序存放,然后使用索引文件找到不同排序中的东西。下面是它如何工作的说明(请见图 2.1)。

- 数据按键入的顺序存放到一个数据文件中。
- 建立若干索引文件维持数据的一个或多个排序的文件。
- 每个索引文件只含有索引值(也称为键值)和一个指向数据文件中相关记录的指针。索引值是被排序的字段。
- 当一个新记录加到文件中时,索引文件中的每个索引值被更新,甚至在必要时被排序。数据记录自身只是加到文件的最后。

索引文件在数据库系统的设计中是非常重要的。如果在文件中只有几个记录,一个索引似乎有许多开销。但是如果有几千个记录,则索引是快速找到信息的唯一方法。没有索引,就必须一个个地检索来找到特定的记录,有了索引,可以在索引中快速找到合适的值(排序