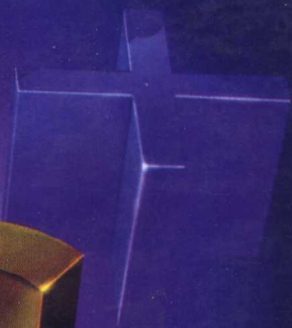


Jack Purdum

计算机程序设计语言系列丛书

- Create distributed accounting and finance applications through real-world examples
- Find out how to effectively use SQL and the Access™ database engine
- Master the tools and libraries you need to write successful database programs



# Accounting & Finance DEVELOPER'S GUIDE WITH VISUAL BASIC™

# Visual Basic 3

# 财务软件开发指南

SAMS  
PUBLISHING

希望

学苑出版社

计算机程序设计语言系列丛书

*Accounting & Finance Developer's Guide with Visual Basic 3*

## Visual Basic 3 财务软件开发指南

Jack Purdum 著

于春燕 译

谢小兵 审校

学苑出版社

(京)新登字 151 号

### 内 容 简 介

本书系统介绍了使用 Visual Basic 3.0 开发财务应用程序的基本方法和原理,并通过详细的程序实例把复杂的财会过程清晰表达出来。对于有志于财务电算化的计算机专业人员和财会人员,本书是一本最为优秀的专业书籍。

需要本书的读者,请直接与北京市海淀区 8721 信箱书刊部联系,邮编:100080,电话:2562329。

### 版 权 声 明

Authorized translation from the English language edition published by Sams Publishing Copyright © 1994.

Chinese language edition published by Beijing Hope Computer Company & Xue Yuan Press/Simon & Schuster (Asia) Pte Ltd Copyright © 1994.

本书英文版名为《Accounting & Finance Developer's Guide with Visual Basic 3》由 Sams Publishing 出版,版权归 Sams Publishing 所有。本书中文版由 Simon & Schuster (Asia) Pte Ltd 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。

### 计算机程序设计语言系列丛书 Visual Basic 3 财务软件开发指南

著 者:Jack Purdum

译 者:于春燕

审 校:谢小兵

责任编辑:汪亚文

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:北京市朝阳区展望印刷厂印刷

开 本:787×1092 1/16

印 张:19 字 数:425 千字

印 数:1~5000 册

版 次:1994 年 10 月北京第 1 版第 1 次

ISBN7-5077-0807-1/TP·18

本册定价:26.80 元

学苑版图书印、装错误可随时退换

## 致 谢

大家都知道,没有人能自己从头到尾事无巨细地编写出版一本书;书中的每一页都凝聚了许多人的辛勤劳动,因此在这里我们要向他们致谢。首先,我要感谢 Phil Paxton,每当我需要时他总是在我身边帮助我。其次,我要感谢 Mary Inderstrodt 和那些帮助她把复杂的语句变得更加易读的人们。接着,我要感谢 Laurie Pitman,她是本书的技术编辑,帮助我修改了许多我忽略的细节。我和他们工作得非常愉快,并且我很欣赏他们的工作。

还有一些人为我的工作提出了许多意见、建议和非常绝妙的想法,他们是 Phil Barth, Lonnie Blevins, Dave Cooper, Don Dudine, Dave Fegnolio, Jerry Franco, Skip Hendricks, Dan Kraft, Jim Merten, Jim Rheude, Mike White 和 Bart York。他们在我的写作过程中起到了很重要的作用。

最后,也是最重要的,我要感谢我的家人。在我的创作过程中,他们给了我很大的支持。尤其要感谢我妻子 Karol 的鼓励和关心!

## 作者简介

Purdum 博士自 1977 年开始就从事计算机编程工作并作了 18 年的大学教授。1985 年他辞去了大学的工作来到了擅长开发 C 编译和统计软件的软件所工作。现在他正参与为一个医学研究机构开发 PC—大型机的数据库。Purdum 博士已经编写了 10 余本有关编程的书籍并撰写了许多文章。

# 前 言

## 本书与其它 Visual Basic 编程书籍的区别

几乎每本书都要介绍它与同类书籍的不同之处,或者说它介绍了某个相同(相似)主题的不同方面。本书则不是这样,这是因为本书所涉及的主题本身就很独特。据我所知,除此之外没有任何一本书涉及到这个主题。

一个很受欢迎的计算机杂志最近发布说:现在数据库应用程序在小商务环境中发展很快。通过向 Microsoft 的工作人员咨询了解得知现在几乎 80% 的数据库的编程工作由 Visual Basic 完成。如果你咨询编程人员,我想他们会支持这种说法。

如果你进一步调查,你会发现现在最需要的数据库应用程序是关于总帐,应收帐,应付帐和存货控制等方面的程序。这些程序之所以受欢迎的原因是现成的软件都不符合顾客的需求。用户面对它们就好象遇到了“方钉,圆孔”问题。这是一个坏消息,但这也是一个好消息,因为那些具有编程特长的人就有了许多的工作机会。这也是本书的出版初衷。

## 本书的目的

本书的目的是为你提供编写满足个别商务的数据库应用程序的框架的代码体。具体说,本书提供了总帐,应收帐,应付帐和存货控制模块的代码。使每个模块都被详细介绍,尤其关注那些类似的地方。你将学会在编写代码前深思熟虑,从而使以后的工作更加轻松。

本书假定你正在为一个客户(即本书中的用户)编程。在某些实例中,你也许就是用户。但是,对于多数读者,你被假定成正为别人编程的程序员。对于首次承接任务的程序员,本书还带有一些提示来帮助你减少错误。一些错误是编程方面的,而另一些则是与客户的关系方面的。这些提示与具体的编程任务无关,但又非常有用。

如果你买过软件,你会知道一个会计软件包中的一个模块至少要花费 1000 美元。而且,许多软件包还需要运行版本。事实上,正是这些花费促使我们自己去编写程序。本书将使编程任务更加简单并且安全。

本书假设你非常熟悉 Visual Basic(VB)编程,并且对 VB 的数据库特性也非常了解。但是,本书仍使用了大量篇幅讲解了 SQL 和 Access 数据库的使用。虽然很多书都使用 Access 引擎作为 flat\_file 系统,但是没考虑它的执行。在多数情况下,利用 Access 引擎的 SQL 特性可减少普通编码方案一半的执行时间。

在本书的编写期间,Microsoft 推出了 Access 2.0。新版本的 Access 的最大的改进也许是结合了 Microsoft 从 Fox 软件公司购买的 Rushmore 技术。基于 Rushmore 技术的索引方案速度很快。的确,Microsoft 已宣称这些 Access 程序将比原来的版本速度提高 100 倍。

从我写这本书开始,Microsoft 已推出了 VB 4.0 的 Beta 版,该版本具有改进的 Access 引擎和 32 位的 VB。但是 VB 3.0 中的 Access 引擎也很不错,它和新版本差别不大。换句话说,VB 是数据库和客户/服务器编程的最佳选择。

## 本书的组织

第一章是使用 VB 编写数据库程序的介绍。这章可以使你重新认识 VB 对数据库的控制能力。第一章的例子是一个可以读取数据库并显示数据库内容的程序。虽然程序不很复杂,但在处理数据库时常常要用到它。本章还介绍了一些贯穿于全文中的编程风格和约定。

第二章展示了整个管理信息系统的总体框架。它详细介绍了菜单的样子和系统产生必要的商业数据的原理。本章展示了如何创建一个新的 Access 数据库和如何在 VB 中创建表格和字段。然后讨论什么是会计科目表和如何用它产生不同的报告。本章的剩余部分集中说明了会计科目表的组成。

第三章讨论了含有公司信息的数据库表格。这个表格不仅含有标准的名称和地址信息,还含有所有一起工作的模块需要的不同的数据结构。本书后面开发的所有模块所需的数据都存储在 Information 表中。

本章还解释了在 VB 中数据控件如何工作。尤其是,本章说明了如何在 VB 中使用结构化查询语言(SQL)。本书中的代码使用了多种 SQL 查询来从数据库访问数据。

第四章开始创建总帐模块。该模块是一个会计系统的核心。通过它提供的信息可跟踪公司的财会数据。利用这些帐目可把原始数据转变成公司的财政报告,比如损益报表和资产负债表。这章提供了向 GL 数据库输入数据的代码。还解释了向 GL 数据库输入交易和过帐的区别。

第五章展示了如何创建和显示财会报表。在这章中你将看到怎样创建试算表(Trial Balance),资产负债表(Balance Sheet)和损益报表(Income Statement)。由于会计科目表的编码方案会影响财会报表的展示,所以最好在阅读完第二章后立即阅读本章。

第六章开始系统的应收帐(AR)部分。许多书籍都使用应收帐程序作为主要的例子。但是,它们多数都采用 flat\_file 方式做 AR。本书中的 AR 使用 SQL 技术来创建和提取 AR 数据。这章提供了输入和保存顾客列表的代码。

第七章展示了一个使用 AR 顾客列表的发票系统。这个例子假设顾客在公司的存货(Inventory)系统订货。这就产生了一个鸡生蛋,蛋生鸡的问题,使得 AR 系统松散地与存货系统相连。本书把 AR 和存货系统同等对待,将来可根据你的具体需求决定谁更重要。通过建议用户在一天的结束时批处理全天的定单,可使 AR 发票系统不直接依赖于存货系统。

这种方法的缺点是:直到一天工作结束时才能知道准确的货存量。因此,也许有些人会想把定单输入和存货系统更加紧密地联系起来。

第八章展示了执行向 AR 系统过帐全天交易的代码。过帐自动在 GL 中创建汇总交易。在这个过程中顾客的帐目(例如,结欠)被修改。这章还展示直接向客户帐号付款和提供 AR 帐目月龄的程序。另一个程序提供了到期 AR 帐目的状态图。

第九章开始代码的应付帐(AP)部分。这章主要讨论如何创建和维护销售商文件。销售商文件的结构根据购货定单设计。本书尽量重用 AR 代码从而简化 AP 模块。由于一个人的 AP 是另一个人的 AR,所以这样做也很合理。

第十章提供了为公司创建购货定单(POs)的代码。POs 与销售商文件相连并在数据库的 Purchases 表格记录了每个 PO。第十一章提供了日 PO 过到总帐,查找 POs,绘制 AR 和 AP 对比图和开支票 4 个工具程序。

第十二章开始存货模块的讨论。本章展示了创建和维护存货项目的代码。还提供了向库存添加项目的程序(不需要出库程序,这是因为这由 AR 处理完成)。

第十三章展示了整个系统的期末过帐。月末过帐主要负责更新留存收益帐。年末过帐集中每月的货存收益数值并把它移到权益帐中。这个程序还负责为下一个会计年度做好准备。这章还标注了系统的会计元素。

第十四章提供了几个很有用的工具程序。第一个是万年历。只需稍作工作,就可变成一个时间表。第二个程序可被用于比较不同的折旧方法。最后一个程序是借款付款,年金付款计算程序。把它们结合到财会系统中会很有用(如果愿意,他们可很容易地从系统中去掉)。

最后,第十五章介绍安全性,详细说明了口令的设置方法。

## **你也能做些工作**

我意识到没有哪本书能够完全满足每位读者的特定需要,我最希望的是,提供一个框架,然后让读者去完善他所需要的特定细节。每位读者都会对本书所提及的主要模块有不同的需求。这本书的目的是尽可能地给读者提供解释和细节以便读者能够做所需的调整。

还有一点是本书的独到之处。你可以修改本书中的代码。显然,本书中所提及的代码只反映了我的观点,而不是事情的真实原貌。因为任何事情都不是一成不变的,如果你有更好的或更有效的方法来完成某些工作,请一定让我知道。如果你能够提供更好的办法,我们将在本书的下一版中增加进去,我对那些有效的方法特别感兴趣。总之,美好的东西应该让大家一起分享。只有得到大家的帮助,书中的代码才会日益完美。

好啦,前言已经够多了,让我们开始编写能帮助你和你的同事管理商务的代码吧!



# 目 录

<b>第一章</b>	<b>Access 和 Visual Basic 的编程介绍</b>	(1)
1.1	介绍	(1)
1.2	数据控件	(21)
1.3	总结	(22)
<b>第二章</b>	<b>菜单系统和会计科目表</b>	(23)
2.1	开始显示屏幕	(23)
2.2	创建数据库	(24)
2.3	什么是会计科目表	(46)
2.4	向 Chart of Accounts 添加数据	(49)
2.5	Chart of Accounts 代码	(51)
2.6	总结	(60)
<b>第三章</b>	<b>公司信息表格</b>	(61)
3.1	信息表格	(61)
3.2	检查用户输入	(65)
3.3	保存公司信息	(66)
3.4	设置会计科目范围	(71)
3.5	总结	(74)
<b>第四章</b>	<b>编写总帐输入</b>	(75)
4.1	总帐数据输入窗体	(75)
4.2	把数据交易保存到数据库中	(78)
4.3	编辑交易	(81)
4.4	登入交易	(89)
4.5	进度尺	(93)
4.6	总结	(94)
<b>第五章</b>	<b>总帐财务报告</b>	(95)
5.1	试算表	(95)
5.2	打印试算表	(104)
5.3	资产负债表	(106)
5.4	损益表	(117)
5.5	总结	(127)
<b>第六章</b>	<b>应收帐</b>	(128)
6.1	设计应收帐(AR)系统	(128)
6.2	我们的 AR 系统	(131)
6.3	Form_Load 事件	(132)
6.4	查看数据库中的顾客信息	(136)
6.5	把 AR 顾客数据写到数据库	(137)

6.6	删除或者查找记录 .....	(140)
6.7	总结 .....	(142)
<b>第七章</b>	<b>应收帐发票</b> .....	<b>(143)</b>
7.1	AR 发票概述 .....	(143)
7.2	AR Form_Load 事件 .....	(146)
7.3	购货折扣 .....	(152)
7.4	Add Item 按钮 .....	(153)
7.5	合计定单 .....	(155)
7.6	把发票信息保存到数据库中 .....	(156)
7.7	总结 .....	(160)
<b>第八章</b>	<b>AR 工具程序</b> .....	<b>(161)</b>
8.1	AR 付款程序 .....	(161)
8.2	把 AR 付款过到总帐 .....	(168)
8.3	查找 AR 发票 .....	(174)
8.4	AR 帐目状态图 .....	(180)
8.5	按期分类 AR 帐 .....	(184)
8.6	总结 .....	(186)
<b>第九章</b>	<b>应付帐</b> .....	<b>(187)</b>
9.1	AP 和购货定单 .....	(187)
9.2	AP 销售商表格 .....	(187)
9.3	销售商表格的 Form_Load 事件 .....	(188)
9.4	Find 按钮 .....	(189)
9.5	AcctNumber_LostFocus 事件 .....	(190)
9.6	btnSave_Click 事件 .....	(192)
9.7	总结 .....	(194)
<b>第十章</b>	<b>应付帐购货定单</b> .....	<b>(195)</b>
10.1	填写购货定单 .....	(195)
10.2	Form_Load 事件 .....	(198)
10.3	选定销售商 .....	(200)
10.4	输入 Item 数据 .....	(202)
10.5	btnAddItem 事件 .....	(204)
10.6	保存购货定单 .....	(205)
10.7	打印 PO .....	(207)
10.8	总结 .....	(210)
<b>第十一章</b>	<b>应付帐工具程序</b> .....	<b>(211)</b>
11.1	把 AP 交易登入到总帐 .....	(211)
11.2	查找 AP 购货定单 .....	(215)
11.3	比较 AR 和 AP .....	(224)
11.4	AP 支票填写程序 .....	(227)

11.5	总结	(236)
<b>第十二章</b>	<b>存货</b>	<b>(237)</b>
12.1	向 Items 表格添加存货项目	(237)
12.2	向存货添加一项	(240)
12.3	存货报告	(244)
12.4	查找存货	(249)
12.5	再订购状态	(250)
12.6	总结	(253)
<b>第十三章</b>	<b>期末过帐</b>	<b>(254)</b>
13.1	月末过帐	(254)
13.2	年末过帐	(260)
13.3	总结	(264)
<b>第十四章</b>	<b>工具</b>	<b>(265)</b>
14.1	日历程序	(265)
14.2	折旧程序	(268)
14.3	借款和年金程序	(271)
14.4	总结	(273)
<b>第十五章</b>	<b>安全性</b>	<b>(274)</b>
15.1	口令的选择	(274)
15.2	Passwords 表格	(274)
15.3	使用口令	(276)
15.4	口令编程的知识	(278)
15.5	向菜单选项添加口令保护	(279)
15.6	输入和编辑口令	(282)
15.7	frmPassword. Form_Load 事件	(283)
15.8	编辑或者增加新的用户名字	(284)
15.9	编辑或者增加新口令	(284)
15.10	设置安全级别	(285)
15.11	保存新的口令数据	(286)
15.12	总结	(287)

# 第一章 Access 和 Visual Basic 的编程介绍

## 1.1 介绍

在这章中,我们将开发一个能够查看数据库内容的工具程序。本书中使用的所有的示例数据库都用 Access 数据库引擎创建。Visual Basic 并不仅仅限于使用 Access 数据库引擎。其它受欢迎的数据库也可以使用,比如 FoxPro, Paradox, dBASE, Btrieve 和支持开放数据库连接(ODBC)的其它数据库。但是,在这里我们使用 Access。如果一定要用其它数据库代替,那么只需做很少的改变。

本章展示了一些编程习惯,它们自始至终贯穿于全文。有些习惯仅仅反映了我的编程风格,而有些习惯是为了使代码更加易读易理解。编程风格是一种非常个性化的东西,只要有利于工作就行。我感到它能使我记住一直以来的工作进展,也能使别人将来看得懂我的代码。因此,你的编程风格应尽量使代码更加易读,尤其让不参与你的编程的人读懂。

如果你正和一组程序员一起开发应用程序,那么这个小组应该规定一种编程风格。一旦风格定下来,小组成员就应在以后的工作中统一使用这种风格。小组的新成员也应该尽快尽可能地习惯这种风格,统一的风格有利于代码的连接,调试和维护。

本章讨论的一些编程习惯是很实用的,应该把它结合到代码中。我将尽量减少我个人的风格而尽量结合使代码易读易理解的习惯。

在进入第一个程序实例之前,需要先讨论一下数据库的基本元素及其构成。对数据库的黑盒子理解的越深,开发的程序就越好。

### 1.1.1 数据库基础

对多数用户而言,数据库仅仅是一个信息的集合。虽然这个观点很正确,但是数据库中的很多信息最终用户从未直接看到过。作为一名程序员,应该熟悉数据库的所有元素,而不仅仅是用户可以看见的那些。

#### 数据库属性和方法

表 1-1 展示了一组与数据库相关的属性和方法。

如表 1-1 所示,VB 中的所有的数据库都被定义成应用到数据库上的属性和方法。

**注意:**在 VB 中,属性定义了一个对象的特征,特性或行为。

QueryTimeout 属性决定一个数据库上的查询要等待多长时间后才能放弃(当网络很忙时,延迟可以增加)。因此,QueryTimeout 属性定义了某些情况下数据库如何工作。另外,Name 属性仅仅定义了数据库在系统上如何保存,或者说用一组 ASCII 字符来表示其特性。

**注意:**在 VB 中,方法定义了用一个对象可以做什么,和可以对该对象做什么。

CreateDynaset 方法允许你在程序中创建一个来自数据库的动态集合。在 VB 中执行的方法类似于子例程或者函数(如果了解表 1-1 中列出的属性和方法的更详细内容,可使用 VB 中的联机帮助特性)。

在继续之前,你应该知道表 1-1 中的有些方法只适用于 VB 的专业版本。例如,标准版本的 VB 不允许你创建 Snapshot 对象。标准版的 VB 的限制是:它不适用于大型数据库应用程序。由于这些限制,我假设你拥有 VB 的专业版本。如果没有,应和 Microsoft 联系争取升级成专业版本。

表 1-1 数据库对象

属性	方法
CollatingOrder	BeginTrans
Connect	Close
Name	CommitTrans
QueryTimeout	CreateDynaset
Transactions	CreateQueryDef
Updatable	CreateSnapshot
	DeleteQueryDef
	Execute
	ExecuteSQL
	ListFields
	ListTables
	OpenQueryDef
	OpenTable
	Rollback

表 1-2 列出了可以和数据库表格,动态集合或 Snapshot 一起使用的属性和方法。

表 1-2 数据库表格,动态集合 Snapshot

属性	方法
BOF	AddNew
Bookmark	Clone
Bookmarkable	Close
DateCreated	CreateDynaset
EOF	CreateSnapshot
Filter	Delete
Index	Edit
LastModified	FindFirst
LastUpdated	FindLast
LockEdits	FindNext
Name	FindPrevious
NoMatch	ListFields
RecordCount	ListIndexes
Sort	MoveFirst
Transactions	MoveLast
Updatable	MoveNext
	MovePrevious
	Seek
	Update

### 数据库表格, 动态集合和 Snapshots

数据库是一个或者多个表格加上处理这些表格需要的信息所组成的集合。表格含有关于某个主题的数据。通常, 表格中的每个主题都由一个或多个提供关于这个主题的详细数据的列组成。例如, 某个表格可能含有关于顾客的信息, 比如姓名、地址、年龄等类似的内容。表格中的所有列提供了定义表格主题的信息。由于通常顾客要多于一个, 因此表格的每行都描述了一个不同的顾客。

多数数据库都有多个表格。一个表格可能描述公司的顾客, 而另一个则可能描述公司出售的产品(产品数量、价格、描述等)。还有一些表格可能含有来自几个表格的信息。例如, 一个“定单”表应该含有谁买了什么产品的信息。所以定单表也就含有来自顾客表格和产品表格的信息。

当使用 Access 数据库引擎时, 数据库中的所有表格都被保存在一个数据库文件中。缺省文件扩展名是 MDB。因此, 如果创建了一个 Access 数据库并称其为 EMPLOYEE, 那么在磁盘上它被保存成 EMPLOYEE.MDB。

动态集合是一个来自数据库中的一个或多个表格的数据的集合。例如, 即使一个表格已含有关于某个职员的信息, 你也可能要创建一个只从该表格抽取该职员的姓名和年龄的动态集合。在这个例子中, 动态集合是该表格中可用信息的子集。另外一个动态集合也许返回职员的姓名和薪水级别。但是, 薪水级别的实际数值也许保存在这个数据库的另一个表格中。你很有可能要创建一个由来自两个或多个表格的信息组成的动态集合。一旦信息在动态集合中可用, 你就可以添加, 编辑或者删除动态集合中的信息(假设动态集合可被改变, 但产生它们的表格可能不能改变)。

简单地说, snapshot 是只读的动态集合。即, 你可以把来自一个或多个表格的信息抽取到 Snapshot 中, 但你不能添加, 编辑或删除 Snapshot 中的数据。当数据只被显示, 不被编辑时, 可利用 Snapshot。

### 数据库字段

前面曾提到过, 数据库中的每个表格都是一列或者多列的集合。在数据库表格中的每列都被称为一个字段。数据库中的一个字段通常用来描述数据库表格的主题的一个方面, 属性或者特性。多数数据库含有许多描述主题的字段。典型的字段项包括人的姓名、年龄、性别、地址、电话号码, 工资级别等等。所有的字段构成了对一个主题的描述。

表 1-3 列出了可用于数据库字段的属性和方法。

表 1-3 数据库字段

属性	方法
Attributes	AppendChunk
CollatingOrder	FieldSize
OrdinalPosition	GetChunk
Name	
Size	
SourceField	
SourceTable	
Type	
Value	

多数数据库程序都被设计成以某些方式从数据库抽取信息。例如,在数据库编程时, Value 属性经常被使用。由于每个字段都有 Name 属性,所以字段的名称就被用于抽取与其相连的 Value。在本章的后面部分你将看到实例。

#### 数据库记录

只有一个主题的数据库用处不大。也就是说,即使数据库中含有几打儿描述职员的信息,但是表中若只有一个职员的信息就没什么意义啦!几乎每个数据库在其表格中都有多行数据,在这里每一行描述了一个不同的主题。数据库中的一个记录是描述表中的某个主题的字段的集合。例如,如果每个职员有 25 个字段,并且公司中有 100 个职员,那么表格应含有 25 列(字段)和 100 行(记录)。其关系见表 1-4。

表 1-4 数据库字段和记录

FirstName	LastName	(其它字段)	Sex
Smith	James		M
Jones	Barbara		F
Kennedy	Bill		M
Wilson	Tom		M
Bryant	Lynn		F
Gavin	Delores		F

如表 1-4,表中的每个记录(或者行)描述了一名职员。职员的描述由表格中的字段(或者列)来提供。

最终用户最关心的是能否用一种简便易用的方法取得表格的信息。我的想法和他们一样,即使他们不是程序员,也应该了解数据库表格,记录和字段的概念。但是,他们最关心的还是信息。他们不关心如何得到,只想很容易地得到。

即使数据库中的信息被组织成结构固定的表格形式,你仍可以用多种方法从数据库抽取信息。有些抽取方法比较快,但是他们可能含有许多复杂的代码。在后面的章节中我们将进一步介绍。

#### 数据库索引和表格定义

通常用来加快数据库查找的方法是使用索引。索引被用于排列表格中的记录。通常,每个表格都把一个字段用作“主关键字”索引。主关键字字段是一个其值在表格中的所有记录均不相同的字段。如果正在为职员数据库表格选择主关键字字段,那么有些字段则不适合做主关键字字段。因为有些人的姓相同,所以姓就不适合做主关键字字段——它不是唯一的。实际上,你也许应创建一个职员 ID 做一个字段,并把它用作主关键字字段。以前,曾用职员的社会保险号码(SSN)做该字段,但是几年以后它会改变因而不合适。

一旦选定了主关键字字段,该字段就被用来排列表格中的记录。在 VB 中,一旦选定了主关键字字段,表格中的所有记录就根据主关键字字段值的升序排列。Access 数据库引擎可以使用主关键字字段索引快速地根据其主关键字字段值定位一个指定的记录。使用索引和不使用索引查找记录的区别很大。使用索引的查找要比不使用快 2 倍以上。

表格定义(TableDefs)定义了一个数据库中表格的结构。它提供了表格的构成,何时创建,何时最后更新等信息。本章结束时讨论的那个简单的程序使用了一些 TableDef 信息来告诉你数据库中的每个表格都含有什么内容。

## 查询定义(QueryDefs)和 SQL

在数据库术语中,从数据库取得信息被称为查询。把查询翻译成数据库可以识别的方式是程序员的任务。多年以前,数据库编程专家们开发了一种专门用于编写数据库查询的语言,叫做结构化查询语言或 SQL。因此,用户的查询通常被翻译成一个 SQL 语句,然后数据库引擎使用这个 SQL 语句来回答查询。正是这种 SQL 语句被保存在 QueryDef 中。

查询定义是关于数据库中内容的询问并且是数据库的一部分。例如,作为一名店主,你也许要每周查询数据库从而找出卖得最好的 10 种商品。程序员就应该编写满足这个要求的代码并把它保存在数据库中。

由于每个查询定义都被命名,因而你可以使用其名字而不是 SQL 语句来调用查询。因此,任何想要使用查询的程序只需知道查询的名字即可,不必了解 SQL 语句的具体内容。

虽然用户不需要知道索引,表格定义,查询定义和 SQL,但是这些概念对程序员来讲就十分重要了。即使使用了一些非常简单的 SQL 语句,你也要向处理 SQL 语句的引擎压入大量的代码。向 SQL 引擎压入的代码越多,要自己编写的代码就越少。即便最好的情况,留给你的代码也要自己调试和维护。虽然关于 SQL 的知识已能自成一本书,但是通过使用 SQL 语句仍可以简化工作。

使用 SQL 语句的好处很多。由于 SQL 位于数据库的中心,所以数据库的性能往往要用它处理 SQL 语言的好坏来检验。结果,编写数据库的人们要花费大量的时间来优化 SQL 性能。这通常意味着要用汇编来编写 SQL 引擎从而优化速度。因此,使用 SQL 语句最终意味着执行汇编语言代码。这就是为什么 SQL 查询往往比在 VB 中编写的类似代码快 2 倍以上的一个原因。

在这本书中你将学习许多关于 SQL 的知识,从而使你的数据库应用程序具有较快的执行速度。这是一条好消息。坏消息是你可能要学习大量的 SQL 知识才能提高一点性能。的确,这也许就是你可以改进代码的一个地方。如果你的确找到一个非常绝妙的 SQL 替代品,请告诉我们,让我们共同分享!

现在你知道了数据库中都有什么,下面我将告诉你如何编写一个程序来看看那个叫数据库的黑盒子里真的有信息吗!

### 1.1.2 READATA 程序

当你开发自己的数据库应用程序时,在整个开发过程中你将需要一个列有字段名称,类型和大小的列表。这正是 READATA 程序所要做的。它显示数据库中所有表格上的字段名称及其大小。每个表格的内容可被显示在屏幕上,也可被直接输出到打印机。

图 1-1 显示了一个数据库被选定并被显示后,READATA 程序的外观。

窗口上部的平板说明该数据库名为 CHART.MDB,它支持交易,并且是一个可更新的 Access 数据库。该数据库不是只读数据库,并且不能以独占形式打开。由于该数据库不是独占的,因此它可在网络上被多个用户使用。由此可见,上部的平板上仅仅显示了一些普通信息。

网格控件被用于传达关于数据库的更详细的信息。数据库中的第一个表格名为 APVendors。其创建和最后更新日期被显示,同时你可以看出该表格是可更新的。表格定义的属性位提供了该表格是系统表格还是附属表格的细节。通常其值为 0,表示它是普通的



Access 表格。

网格中的其余行提供了表格的详细说明。在图 1-1 中,第一个字段名为 APNumber,是正文(text)字段,最多有 10 个字符组成。第二个字段名为 VendorName,最多由 50 个字符组成。图 1-1 仅显示了 APVendors 表格的一部分,使用滚动条可以看到该表格的其余部分。

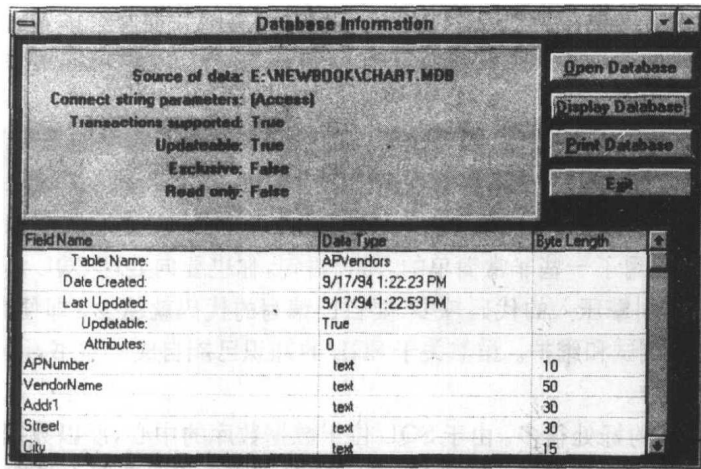


图 1-1 READDATA 程序的输出

### 风格考虑

当你运行 READDATE 程序时,会注意到我的风格。在图 1-1 中,我使用三维平板来放置数据库的普通信息。三维平板可被用于组合相关的信息,它可替代图文框控件(而且更加吸引人)。

**提示:** 在一个三维平板或图文框控件中放置一个控件的方法有两个。第一个方法是在工具栏上双击想创建的控件。这将自动地把该控件的备份放到窗体的中央。然后可在三维平板或图文框控件中拖动它。第二个方法是单击想创建的控件,移动光标到适当的位置,然后把该控件画成适当的大小。

当把控件放到平板或图文框控件上时,第二种方法更加适合。这是因为用第二种方法创建的控件可随窗体、平板的移动而移动。如果用第一种方法,你就必须自己动手移动每个控件。

第二个要注意的是颜色的使用。我喜欢用标准的灰色做按钮、平板和图文框控件的背景颜色。但是窗体的背景颜色我喜欢栗色。遗憾的是,这没什么科学依据——仅仅是喜欢而已。

对于网格控件,我使用了 8.25 点的 MS Sans Serif 字体并关闭了黑体属性。同样,这也没有科学依据。但是,我认为当字体较小时,不用黑体更加易读。虽然我可以使用大字体,但是为了不使用水平滚动条我减小了字体。

如图 1-1 所示,在三维平板上并没有用正文框来显示信息。而是用了标签,并在运行时填充其内容。如果有人编过这个程序,他可能在平板的左边用标签,而右边用正文框。然后,当程序运行时,再用适当的信息填充正文框。很虽然,这样做的结果和我一样。

但是,除非使用无边框,灰背景的正文框,否则就会让人觉得要往里面键入些什么。图