

Managing Data with Microsoft Excel

# 巧学巧用

# Excel 2003

# 数据管理

(中文版)

[美] Conrad Carlberg 著

王 军 等译

que®



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Managing Data with Microsoft Excel

# 巧学巧用 Excel 2003 数据管理 (中文版)

[美] Conrad Carlberg 著

王 军 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书面向对 Excel 和数据库操作有一定基础的中级读者,目的是为用户提供所需的工具,帮助决定如何存储自己的数据——使用 Excel,还是使用像 Access 那样的数据库管理系统,以及怎样最佳地进行这一选择。全书文字通俗易懂,并提供有大量示例和案例研究。通过本书的学习,读者将能够更加有效地管理自己的数据。

**QUE®**

Authorized translation from the English language edition, entitled Managing Data with Microsoft Excel, 1<sup>st</sup> Edition, 0789731002, by Conrad Carlberg, published by Pearson Education Inc., publishing as Que. Copyright © 2004 by Que Publishing.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from Pearson Education Inc.. CHINESE SIMPLIFIED language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY&BEIJING MEDIA ELECTRONIC INFORMATION CO. LTD.

本书英文版由 Pearson Education Inc. (Que)出版,Pearson Education Inc.公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号 图字:01-2005-5214

### 图书在版编目(CIP)数据

巧学巧用Excel 2003数据管理(中文版)/(美)卡尔博格(Carlberg, C.)著;王军等译.一北京:电子工业出版社,2005.9

书名原文:Managing Data with Microsoft Excel

ISBN 7-121-01712-1

I. 巧… II. ①卡… ②王… III. 电子表格系统, Excel 2003 IV. TP391.13

中国版本图书馆CIP数据核字(2005)第106314号

责任编辑:朱 巍

特约编辑:李双庆

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:17.875 字数:450千字

印 次:2005年9月第1次印刷

定 价:29.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

## 关于作者

Conrad Carlberg 是“网络控制系统”股份有限公司的总裁,该公司为健康护理行业开发事件与统计分析软件。他具有美国科罗拉多大学的博士学位,并已在 Excel 领域编著过 9 本书籍,包括 Que 出版的“Business Analysis with Excel”。Carlberg 居住在圣地亚哥附近,经常出海航行。

## 题献词

再一次将本书献给 Toni Messer 和 Tabben——感谢上帝,他们还和我在一起。同时,敬将此书献给 Tigger 和 Button——他们过早地离开了我们,但将永远活在我们心中。

## 致谢

感谢本书的技术编辑 Debra Dalgleish、Melissa O'Malley 和 Jon Price,他们为我指点了不少迷津。感谢本书的开发编辑 Laura Norman,他巧妙地对我开放式的页面数量和其他编辑方面的问题进行了处理。同时感谢 Stephanie McComb 将本书的内容组合到一起。

还要感谢长期以来教育和鼓励我的所有人。Gene Glass 对我早期尝试利用 Excel 提取主要成分方面帮助非常大。Dave Derby 和 Bev Monigal 教会了我在一个非常难的 Excel 版本中(不是 Microsoft 的,而是 SBC 的)的相关数据库理论。永远记得那些来自 CompuServe(美国最大的在线信息服务机构之一)论坛日的我的 Microsoft MVP(微软最有价值专家)朋友们,Rob Bovey、David Hager、Tom Ogilvy、Jim Rech、Bob Umlas,以及其他告知我 Microsoft 在 Excel 中的各种优缺点的朋友们。最后,当然要感谢说服我们并为我们如此好地主持该论坛的 John LaTour。

## 联系方式

作为本书的读者,你是我们最重要的批评家和评论员。我们非常重视你的观点,并想知道我们有哪些做得正确,哪些可以做得更好,哪些方面的内容还没有涉及到,以及其他你想跟我们说的内容。

作为 Que 的一个发行人,我非常欢迎你的意见。你可以通过电子邮件或者直接写信给我,让我知道你喜欢这本书的哪些地方,不喜欢哪些地方——以及我们该怎么做才能使我们的书做得更好。

请注意,我无法帮助你解决与本书主题相关的技术问题。不过,我们有一个用户服务部门,我会把那些与本书有关的技术问题转发给他们。

在你写信时,请确保包含了本书的标题、作者以及你的名字、电子邮件地址和电话号码。我将仔细阅读你的意见,并和本书的作者以及编辑们共同分享有关的内容。

电子邮件:feedback@quepublishing.com

通信地址:Greg Wiegand

Associate Publisher

Que Publishing

800 East 96th Street

Indianapolis, IN 46240 USA

有关这本书或者 Que 出版的其他书目的更多信息,请访问我们的网站 [www. quepublishing. com](http://www.quepublishing.com)。在 Search 字段中输入本书的 ISBN(0789731002)或者标题,即可找到所需的内容。

## 简介

### 关于 Excel

在 20 世纪 90 年代后期,本书作者读到一本白皮书(一份市场研究报告),它发现大多数 Excel 工作表都没有使用公式。也就是说,大多数 Excel 用户更多的还是将 Excel 当做是一种存储数据的仓库,而不是一种分析工具。

这个发现令本书作者十分烦恼,因为 Excel 并不是设计来作为一种存储或者检索数据的工具的。当然,也可以那样使用 Excel,而且根据那份市场研究,许多人就是这么做的。

那可不是 Excel 开发人员所愿意看到的使用方式。他们原本打算让用户使用 Excel 分析数据,而不是管理它们。假如你使用过 Excel,那么他们会期望你使用像 AVERAGE()、RATE()和 PI()这样的函数,而不仅仅是向工作表单元格中填入字母和数字。

但是,是市场决定如何使用某一产品,而不是该产品的设计小组。而市场已经决定使用 Excel 来存储数据,根本不管该软件设计人员的初衷。

这一点可以理解。市场发现 Excel 是一种存储和排列数据的好方法,而市场通常都是正确的。本书作者同意市场的决定,而且猜想你也这么做的。Excel 工作表确实是一个存放数据的好地方。

因此,就有这样一个问题:假如我们想要使用 Excel 存储和管理数据,这样做的最好办法是什么呢?

对于这个问题,没有特别好的答案,至少没有哪一种方法简捷、清晰,并且能够提供所需的信息。这个问题的答案,部分取决于需要处理的数据量,部分取决于需要如何构建这些数据,而且部分取决于是否可以使用工作表公式汇总和分析数据。

正如大多数用户已经发现的那样,Excel 会提供极棒的数据存储方式。它非常灵活,允许用户定义想要如何存储自己的数据。用户不必遵照其他应用程序(例如 Access 和 SQL Server)强加于自己的数据结构。

举例来说,Excel 并不要求用户将不同的记录放入不同的行中;也不像其他应用程序那样,要求用户将不同的变量放入不同的列中。这一灵活性会非常方便,尤其是在数据显示方式比较重要的时候。

但是,这一灵活性也是有一定的代价的。假设想要如图 I.1 所示,将通信地址放入一张 Excel 工作表中。

图 I.1 显示的数据布局是一种很方便的样式。特别是,它与用户希望此信息在信封标签上的显示方式极其相似。

但是照例,用户需要为这种便利付出代价。举例来说,假如想要知道共有多少地址是在加利福尼亚州中、在旧金山市中、或者是在旧金山市的猎鹰大街上,那时该怎么办呢?图 I.1 中使用的数据布局可不是进行这种分析的好基础,而且,它也不适于进行任何一种分析。

尽管它不具有一个很好的基础,但是可以对其进行处理。举例来说,适当的公式可以根据任一指定的州名、城市名或者街道名称,统计出像图 I.1 中那样放置的地址数量。本书将演示

|    | A                       | B | C | D | E | F | G | H | I | J |
|----|-------------------------|---|---|---|---|---|---|---|---|---|
| 1  | John Smith              |   |   |   |   |   |   |   |   |   |
| 2  | 123 Elm Street          |   |   |   |   |   |   |   |   |   |
| 3  | San Diego CA 92001      |   |   |   |   |   |   |   |   |   |
| 4  |                         |   |   |   |   |   |   |   |   |   |
| 5  | Jane Jones              |   |   |   |   |   |   |   |   |   |
| 6  | 456 Maple Street        |   |   |   |   |   |   |   |   |   |
| 7  | San Francisco, CA 94101 |   |   |   |   |   |   |   |   |   |
| 8  |                         |   |   |   |   |   |   |   |   |   |
| 9  | Sam Spade               |   |   |   |   |   |   |   |   |   |
| 10 | 789 Falcon Avenue       |   |   |   |   |   |   |   |   |   |
| 11 | San Francisco, CA 94107 |   |   |   |   |   |   |   |   |   |
| 12 |                         |   |   |   |   |   |   |   |   |   |
| 13 | Peter Greenstreet       |   |   |   |   |   |   |   |   |   |
| 14 | 1011 Maltese Circle     |   |   |   |   |   |   |   |   |   |
| 15 | Claremont, CA 91711     |   |   |   |   |   |   |   |   |   |
| 16 |                         |   |   |   |   |   |   |   |   |   |
| 17 |                         |   |   |   |   |   |   |   |   |   |
| 18 |                         |   |   |   |   |   |   |   |   |   |
| 19 |                         |   |   |   |   |   |   |   |   |   |
| 20 |                         |   |   |   |   |   |   |   |   |   |
| 21 |                         |   |   |   |   |   |   |   |   |   |
| 22 |                         |   |   |   |   |   |   |   |   |   |
| 23 |                         |   |   |   |   |   |   |   |   |   |
| 24 |                         |   |   |   |   |   |   |   |   |   |
| 25 |                         |   |   |   |   |   |   |   |   |   |
| 26 |                         |   |   |   |   |   |   |   |   |   |
| 27 |                         |   |   |   |   |   |   |   |   |   |

图 I.1 这种排列方式特别像邮件标签上的样子

如何创建那些公式。

更为重要的是,本书还将演示如何以某些方式管理自己的数据,使其能够更加易于分析和汇总所需的信息。例如,对于上述 Excel 工作表中的数据来说,另一种可能的布局将如图 I.2 所示。

|    | A                    | B                   | C                       | D | E | F | G | H |
|----|----------------------|---------------------|-------------------------|---|---|---|---|---|
| 1  | Name                 | Street Address      | City, State, Zip        |   |   |   |   |   |
| 2  | John Smith           | 123 Elm Street      | San Diego CA 92001      |   |   |   |   |   |
| 3  | Bridget O'Shaunessey | 456 Maple Street    | San Francisco, CA 94101 |   |   |   |   |   |
| 4  | Sam Spade            | 789 Falcon Avenue   | San Francisco, CA 94107 |   |   |   |   |   |
| 5  | Peter Greenstreet    | 1011 Maltese Circle | Claremont, CA 91711     |   |   |   |   |   |
| 6  |                      |                     |                         |   |   |   |   |   |
| 7  |                      |                     |                         |   |   |   |   |   |
| 8  |                      |                     |                         |   |   |   |   |   |
| 9  |                      |                     |                         |   |   |   |   |   |
| 10 |                      |                     |                         |   |   |   |   |   |
| 11 |                      |                     |                         |   |   |   |   |   |
| 12 |                      |                     |                         |   |   |   |   |   |
| 13 |                      |                     |                         |   |   |   |   |   |
| 14 |                      |                     |                         |   |   |   |   |   |
| 15 |                      |                     |                         |   |   |   |   |   |
| 16 |                      |                     |                         |   |   |   |   |   |
| 17 |                      |                     |                         |   |   |   |   |   |
| 18 |                      |                     |                         |   |   |   |   |   |
| 19 |                      |                     |                         |   |   |   |   |   |
| 20 |                      |                     |                         |   |   |   |   |   |
| 21 |                      |                     |                         |   |   |   |   |   |
| 22 |                      |                     |                         |   |   |   |   |   |
| 23 |                      |                     |                         |   |   |   |   |   |
| 24 |                      |                     |                         |   |   |   |   |   |
| 25 |                      |                     |                         |   |   |   |   |   |
| 26 |                      |                     |                         |   |   |   |   |   |
| 27 |                      |                     |                         |   |   |   |   |   |

图 I.2 这种排列方式最适于汇总或者管理数据

图 I.1 和图 I.2 之间的差异,强调了 Excel 提供给用户的灵活性。用户既可以决定将收件人的姓名放在第 1、5、9、……行(如图 I.1 所示),也可以将它们放在同一列中(如图 I.2 所示)。Excel 将此决定权交给了用户。而其他应用程序,特别是数据库管理系统,并不允许做出那样的设计决策。

数据库管理系统如此注意琐碎细节是有很好的理由的,而这些理由同样适用于 Excel。在想要分析数据时,像图 I.2 中那样排列数据会比图 I.1 中更为有效。

但是,在想要像单独的记录那样查看这些数据时,图 I.1 中的布局便会更加有效。那么,如何在这两者之间做出抉择呢?

其实可能并不必做出抉择。正确的方法可以同时提供这两种布局的优点。如果正确地管

理这些数据,就可以既获得图 I.1 中所示布局的便利性,又获得图 I.2 中所示布局的高效性。

但是,首先需要对数据进行正确的排列。用户将需要知道如何使用 Excel 的数组公式,并需要了解 Excel 的数据管理功能。在许多情况下,都会使用到 Visual Basic for Applications (VBA)以及它的近邻——数据存取对象(DAO)和 ActiveX 数据对象(ADO)。

这本书的目的就是为用户提供所需的工具,帮助用户决定如何存储自己的数据——使用 Excel,还是使用像 Access 那样的数据库管理系统,以及怎样最佳地进行这一选择。那将使用户处于管理数据的最佳位置。

## 如何使用本书

错误的选择通常是一个很好的起点。第 1 章将详细介绍几个极糟糕的示例。

第 2 章将讨论利用 Excel 工作表函数定位以及重新整理工作表上数据的方法。

建立工作表(即,展示它的表目)有多种方式,从而使得用户可以非常轻松地管理其中的数据。第 3 章将演示如何有效地排列数据。

Excel 提供了许多种方法,可以从其他数据源将数据导入工作表。第 4 章将介绍其中最有用的一种:外部数据区域。这一章还将介绍 Microsoft 查询,并基于导入的数据建立数据透视表。

第 5 章将进一步讲解使用 Microsoft 查询获取外部数据的方法。包括如何将外部数据表连接在一起,以及如何利用选择标准来设计在此之后要导入的数据。这一章还将介绍如何管理外部数据区域,从而使其自动刷新自己、向下填充相邻的公式、进行密码保护等等。

和 Excel 的数据导入功能一样,还应该知道一些技巧和陷阱。第 6 章将讨论如何避免空值,在数据透视表中组合字段,更改每次运行某一查询时的条件以及如何建立和刷新 Web 查询。

“利用 Excel 管理数据”的剩余部分将关注在 Excel 和实际关系数据库之间交换数据的自动操作。尤其是在将数据从 Excel 移动到别的容器中时,功能最强大的方法包括 Visual Basic for Applications(VBA)。第 7 章将通过大量示例代码演示如何使用循环,理解对象模型,声明变量,建立 With 结构以及使用宏记录器。所有这些技巧都将根据它们和管理数据方面的应用来讨论。

第 8 章将介绍两个重要的库,它们可以在 VBA 中使用,使得代码能够直接访问数据库中的结构体——表、字段、记录、查询等等。通过在 VBA 中使用这两个库,可以在 Excel 环境中管理整个数据库。

假如正打算从 Excel 内管理数据库中的数据,那么知道如何从数据库管理系统实现这一点会很有帮助。第 9 章将演示如何利用数据库管理系统的用户界面直接创建表、字段和查询。

有了第 9 章的从数据库内部管理数据结构体的回顾基础,接下来的一章将演示如何从 Excel 内部实现相同的功能。第 10 章含有大量使用 VBA 和 DAO 组合以及 VBA 和 ADO 组合处理数据的示例。

第 11 章将分析使用 VBA、ADO 以及 DAO,将数据从数据库移动到 Excel 工作表中的最有效方法。这些技巧在数据不能被一下子全部取回来(如同外部数据区域的情况一样)以及代码需要对那些数据进行其他处理时,显得尤为重要。这一章将介绍一个非常长的 VBA 代码块的开发过程,它将把从数据库检索到的数据放到工作表上的精确位置。

第 11 章将注意力集中在将数据从数据库获取到 Excel 中。第 12 章将关注数据流的另一个方向——从 Excel 到数据库。它将演示如何向数据库表添加新的记录、编辑现有记录以及删除那些不再需要的记录——这一切都通过使用 VBA 和 DAO 或者是 VBA 和 ADO 的组合。

## 特殊元素

本书包含有一些不同类型的信息,用于帮助读者理解所介绍的内容。

### 案例研究

案例研究是用户在使用 Excel 期间可能会遇到的一个问题或者是一种情况——通常,都需要一些额外的独创性来进行处理。本书中的所有案例研究都直接源自于本书作者在其公司的客户站点所遇到的情况。这里隐去了客户的真实姓名。

### 说明、提示和警告

#### 说明

说明就是有关所讨论主题的一点花絮。假如轻易跳过这些说明,就可能会错过一个“金块”!

#### 提示

提示是本书作者得以传递自己在现实生活中使用 Excel 的经验,以及为帮助用户更有效地使用 Excel 而提供建议和技巧的地方。

#### 警告

这些内容是不应该跳过的。警告将帮助读者远离普遍容易犯的错误,或者警示潜在的问题。

### 交叉引用

当某一主题涉及其他章节中的更多方法时,或者是在其他章节中有与当前讨论相关的信息时,你会发现一些指向本书其他部分的有帮助的参考。

→ 交叉引用的格式就是这样子,它指向本书中其他有用的位置。

# 目 录

## 第一部分 在数据管理中易犯的错误

|                                |    |
|--------------------------------|----|
| 第 1 章 误把 Excel 当做数据库管理工具 ..... | 1  |
| 将数据管理看做上下文 .....               | 1  |
| 转置行与列 .....                    | 5  |
| 使用标签代替代号 .....                 | 7  |
| 处理每个类别中记录的变量数 .....            | 10 |
| 悬崖勒马 .....                     | 12 |

## 第二部分 从 Excel 内部管理数据

|                                |    |
|--------------------------------|----|
| 第 2 章 Excel 数据管理特性 .....       | 13 |
| 工作表函数使用综述 .....                | 13 |
| 使用 OFFSET 函数定位数据 .....         | 15 |
| 使用 INDEX 函数查找数据 .....          | 17 |
| 使用 MATCH 函数查找数据 .....          | 18 |
| MATCH 函数和其他函数的组合使用 .....       | 19 |
| 使用 VLOOKUP 函数 .....            | 26 |
| 使用 TRANSPOSE 函数重新部署数据 .....    | 29 |
| 从数据透视表中获取数据:GETPIVOTDATA ..... | 32 |
| 展望 .....                       | 35 |
| 第 3 章 Excel 列表、名称和筛选 .....     | 36 |
| 创建列表 .....                     | 36 |
| 排序列表 .....                     | 41 |
| 使用名称 .....                     | 43 |
| 使用“自动筛选”筛选数据 .....             | 51 |
| 使用“高级筛选” .....                 | 54 |
| 展望 .....                       | 56 |

## 第三部分 从 Excel 内部管理外部数据

|                                |    |
|--------------------------------|----|
| 第 4 章 导入数据:综述 .....            | 59 |
| 将外部数据获取到工作簿中 .....             | 59 |
| 将数据导入数据透视表 .....               | 74 |
| 展望 .....                       | 80 |
| 第 5 章 使用 Microsoft Query ..... | 81 |
| 了解“查询” .....                   | 81 |

|                               |            |
|-------------------------------|------------|
| 对多个表进行查询 .....                | 83         |
| 连接父记录和子记录 .....               | 85         |
| 使用数据库查询 .....                 | 90         |
| 处理数据区域 .....                  | 95         |
| 管理 Boolean(布尔型)字段和复选框 .....   | 97         |
| 展望 .....                      | 100        |
| <b>第 6 章 导入数据:深入探讨 .....</b>  | <b>101</b> |
| 了解数据透视表 .....                 | 101        |
| 为数据透视表准备数据 .....              | 108        |
| 使用附带条件的 Microsoft Query ..... | 112        |
| 查询 Web 站点 .....               | 113        |
| 展望 .....                      | 116        |

#### 第四部分 从 Excel 内管理数据库

|                                          |            |
|------------------------------------------|------------|
| <b>第 7 章 VBA 要点回顾 .....</b>              | <b>117</b> |
| 使用 VBA 管理数据 .....                        | 117        |
| 建立子例程 .....                              | 123        |
| 理解圆点记法 .....                             | 128        |
| 使用循环 .....                               | 129        |
| 使用 With 程序块 .....                        | 135        |
| 理解宏记录器的代码 .....                          | 137        |
| 使用对象变量 .....                             | 141        |
| 展望 .....                                 | 143        |
| <b>第 8 章 打开数据库 .....</b>                 | <b>144</b> |
| 连接到数据库 .....                             | 144        |
| 使用 ADO 进行连接 .....                        | 144        |
| 指定数据源 .....                              | 148        |
| 打开连接 .....                               | 152        |
| 使用 DAO 打开数据库 .....                       | 154        |
| 处理受保护的 Jet 数据库 .....                     | 159        |
| 打开受保护的 Access 数据库 .....                  | 170        |
| 展望 .....                                 | 171        |
| <b>第 9 章 管理数据库对象 .....</b>               | <b>173</b> |
| 创建表 .....                                | 173        |
| 定义字段 .....                               | 173        |
| 创建查询 .....                               | 186        |
| 展望 .....                                 | 197        |
| <b>第 10 章 使用 ADO 和 DAO 定义字段与记录 .....</b> | <b>198</b> |
| 从 Excel 创建数据库 .....                      | 198        |
| 使用 DAO 创建新的数据库 .....                     | 199        |

---

|                                                 |            |
|-------------------------------------------------|------------|
| 使用 ADO 创建表和字段 .....                             | 211        |
| 声明和使用记录集 .....                                  | 215        |
| 理解 DAO 记录集类型 .....                              | 220        |
| 理解 ADO 记录集类型 .....                              | 223        |
| 展望 .....                                        | 225        |
| <b>第 11 章 使用 ADO 和 DAO 将数据获取到 Excel 中 .....</b> | <b>226</b> |
| 使用 CopyFromRecordset .....                      | 226        |
| 构建应用程序 .....                                    | 228        |
| 从参数查询中取回数据 .....                                | 229        |
| 将数据从数据库返回至用户表单 .....                            | 237        |
| 展望 .....                                        | 255        |
| <b>第 12 章 使用 ADO 和 DAO 从 Excel 控制数据库 .....</b>  | <b>257</b> |
| 利用 DAO 执行操作查询 .....                             | 257        |
| 利用 ADO 执行操作查询和存储过程 .....                        | 263        |
| 利用 DAO 记录集将数据从 Excel 移至 Jet 数据库 .....           | 267        |
| 利用 ADO 将数据从 Excel 移至数据库 .....                   | 270        |
| 展望 .....                                        | 273        |

# 第一部分 在数据管理中易犯的错误

这一部分包括：

第 1 章：误把 Excel 当做数据库管理工具

## 第 1 章 误把 Excel 当做数据库管理工具

本章包括：

将数据管理看做上下文

过度使用函数

转置行与列

使用标签代替代号

处理每个类别中记录的变量数

悬崖勒马

### 将数据管理看做上下文

Excel 是一个极好的应用软件，其用户界面使用方便且有效——至少可以使它符合用户自己的舒适度和有效性。它的工作表布局灵活，行、列可以任意安排；记录和变量、星期和日子、账目和借贷或信贷等等。

Excel 提供了大量的函数选择，并且为用户制作了大量的预定义公式。图表能够帮助直观显示数据情况。数据透视表是对大量记录集合进行合成和分析的快照，甚至可以控制它的外观，使其他用户无法辨认出这是一个 Excel 工作表。

同时 Excel 还提供了许多管理和操作数据的方法。用户可以对数据进行筛选而只关心其细节，将数据排序成多个子集，甚至可以利用变量的查找函数和引用函数使 Excel 更像一个真正的数据管理系统。

多年以来，Excel 的应用是如此成功，以致许多人从来没有都想过要使用其他软件。在许多情况下，他们也没有必要使用其他软件。对于从小规模到中等规模的数据集合，Excel 提供了几乎所有用户需要管理数据和分析数据的方法。但是当数据集合变得很大的时候，事情就开始变得棘手了。

第 2 章展示了如何使用多个不同的 Excel 特性——特别是工作表函数——使 Excel 更像一个数据库管理系统。第三部分和第四部分分别展示了当涉及到的数据超出 Excel 自身能够有效处理的数据量时，应该如何进行处理。

本章的意图是让用户确信一点，当数据量变得很大的时候其实是一件好事。当意识到自己的某个项目遇到这种情况的时候，就可以自如地运用本书中的一个和多个数据管理解

决方案。

## 案例研究

### 过度使用函数

假设 The Sisyphus Corporation 公司已经成为你的一位客户,你的工作是帮助他们解决数据跟踪问题。Sisyphus 希望跟踪为其销售方准备的定购单(PO)信息。当销售方发送了发票做为付款凭证时,Sisyphus 职员将这些信息输入到一个 Excel 工作簿中。信息包括 PO 编号、PO 日期、发票编号、发票金额、发票日期等等数据。

随着时间的推移,Sisyphus 拥有了相当可观的信息量,其中一些如图 1.1 所示。

| 1  | A      | B          | C      | D            | E                           | F          | G          | H           |
|----|--------|------------|--------|--------------|-----------------------------|------------|------------|-------------|
| 1  | PO #   | DATE       | MGR ID | Amount       | Vendor                      | INV REC    | INV DATE   | INV AMT     |
| 2  | 020022 | 1/10/2002  | 6      | \$ 18,952.00 | A1 WINDOW CLEANERS          | 7/5/2000   | 6/29/2002  | \$ 3,502.00 |
| 3  | 020155 | 1/2/2002   | 7      | \$ 1,095.61  | TELECO                      | 1/2/2001   | 12/25/2001 | \$ 1,095.61 |
| 4  | 020010 | 1/15/2002  | 7      |              | SPECIALTY UNIFORM           | 1/15/2001  | 12/27/2001 | \$ 212.42   |
| 5  | 021189 | 5/2/2002   | 3      | \$ 1,050.00  | CONSUMERS OPTIONS           | 5/2/2001   | 5/2/2002   | \$ 1,050.00 |
| 6  | 021225 | 5/7/2002   | 1      | \$ 200.00    | TIRCO INDUSTRIES            | 6/2/2001   | 5/14/2002  | \$ 172.62   |
| 7  | 020278 | 1/17/2002  | 3      | \$ 1,560.00  | MOTTEL VOICE & DATA         | 12/17/2001 | 12/3/2001  | \$ 1,560.00 |
| 8  | 020277 | 1/17/2002  | 3      | \$ 1,560.00  | MOTTEL VOICE & DATA         | 12/19/2001 | 12/10/2001 | \$ 1,560.00 |
| 9  | 020204 | 1/8/2002   | 7      |              | WEBERN PRINTING COMPANY     | 12/21/2001 | 12/12/2001 | \$ 1,063.80 |
| 10 | 020203 | 1/8/2002   | 3      | \$ 231.50    | UNAVATEC                    | 12/24/2001 | 12/14/2001 | \$ 231.50   |
| 11 | 020226 | 1/11/2002  | 7      | \$ 75.00     | INTL ASSOC. FOR FACILITIES  | 12/27/2001 | 12/19/2001 | \$ 75.00    |
| 12 | 020377 | 1/29/2002  | 3      | \$ 920.00    | TRI-CO SERVICES             | 12/27/2001 | 1/23/2002  | \$ 920.00   |
| 13 | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 12/28/2001 | 12/30/2001 | \$ 539.63   |
| 14 | 020002 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 12/28/2001 | 12/30/2001 | \$ 630.76   |
| 15 | 020003 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 12/28/2001 | 12/30/2001 | \$ 1,700.00 |
| 16 | 020003 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 12/28/2001 | 12/30/2001 | \$ 246.76   |
| 17 | 020004 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 12/28/2001 | 12/30/2001 | \$ 157.42   |
| 18 | 020263 | 1/15/2002  | 8      | \$ 250.00    | WESTERN GROUNDS MAINTENANCE | 12/28/2001 | 12/27/2001 | \$ 250.00   |
| 19 | 020264 | 1/15/2002  | 8      | \$ 224.00    | WESTERN GROUNDS MAINTENANCE | 12/28/2001 | 12/27/2001 | \$ 224.00   |
| 20 | 020007 | 12/25/2001 | 6      | \$ 11,440.00 | SHORE'S OF BELLFLOWER       | 12/31/2001 | 12/25/2001 | \$ 210.78   |
| 21 | 020151 | 1/2/2002   | 3      |              | TOM KEPLER                  | 1/2/2002   | 12/31/2001 | \$ 840.00   |
| 22 | 020280 | 1/17/2002  | 3      | \$ 1,560.00  | MOTTEL VOICE & DATA         | 1/2/2002   | 12/18/2001 | \$ 1,560.00 |
| 23 | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 1/3/2002   | 1/1/2002   | \$ 2,250.00 |
| 24 | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC.  | 1/3/2002   | 1/1/2002   | \$ 415.00   |

图 1.1 不是所有相同 PO 编号的记录分组在一起;是按 F 列收到发票的日期进行排序的

### 提示

尽管已经更换了销售方的名称,但是图 1.1 中显示的信息是真实的,它来自一个中等规模的公司(并不是被命名为 Sisyphus)所持有的一个工作簿。本书中所有的案例研究和绝大多数的插图都是真实的范例。

在图 1.1 中显示的数据实际上包括从第 1 行一直到第 6045 行的区域。做为部分客户管理和控制过程,Sisyphus 想定期把每个 PO 支付金额总计,与原来为 PO 提取的金额进行比较。为了实现这样的比较,Sisyphus 需要得到所有发票金额总计,这些发票包括为每个 PO 提交的发票。

创建工作表的职员在工作表中间插入了几条新行来保存 PO 总计。如果工作表中只含有少数几个 PO 和发票记录,这种方法是行得通的。当有新的发票到来时,他们采取了下面的几个步骤,使用的是与图 1.1 含有相同数据的工作表,但是又附加了几行来显示发票总计:

1. 按照编号 PO 对整个数据区域进行排序。
2. 选择菜单命令“编辑”►“查找”,找到第一个与当前发票匹配的 PO 编号记录。
3. 向下滚动,找到该 PO 的总计行。
4. 在总计行正上方插入一个空白行。
5. 在新行中输入关于当前发票的信息。

6. 在总计公式中更正区域地址,以便捕获新发票金额(如图 1.2 所示)。

|    | A      | B          | C      | D            | E                          | F          | G          | H           |
|----|--------|------------|--------|--------------|----------------------------|------------|------------|-------------|
|    | PO #   | DATE       | MGR ID | Amount       | Vendor                     | INV REC    | INV DATE   | INV AMT     |
| 1  | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 12/28/2001 | 12/30/2001 | \$ 539.63   |
| 2  | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 1/3/2002   | 1/1/2002   | \$ 2,250.00 |
| 3  | 020001 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 1/3/2002   | 1/1/2002   | \$ 415.00   |
| 4  | SUM    |            |        |              |                            |            |            | \$ 3,204.63 |
| 5  | 020002 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 12/28/2001 | 12/30/2001 | \$ 630.76   |
| 6  | 020002 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 1/3/2002   | 1/1/2002   | \$ 3,242.53 |
| 7  | 020002 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 1/3/2002   | 1/1/2002   | \$ 648.32   |
| 8  | SUM    |            |        |              |                            |            |            | \$ 4,521.61 |
| 9  | 020003 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 12/28/2001 | 12/30/2001 | \$ 1,700.00 |
| 10 | 020003 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 12/28/2001 | 12/30/2001 | \$ 245.78   |
| 11 | SUM    |            |        |              |                            |            |            | \$ 1,945.78 |
| 12 | 020004 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 12/28/2001 | 12/30/2001 | \$ 157.42   |
| 13 | 020004 | 1/15/2002  | 8      |              | PROPERTY INNOVATIONS, INC. | 1/3/2002   | 1/1/2002   | \$ 1,250.00 |
| 14 | SUM    |            |        |              |                            |            |            | \$ 1,407.42 |
| 15 | 020007 | 12/25/2001 | 6      | \$ 11,440.00 | SHORE'S OF BELLFLOWER      | 12/31/2001 | 12/25/2001 | \$ 210.78   |
| 16 | SUM    |            |        |              |                            |            |            | \$ 210.78   |
| 17 | 020010 | 1/15/2002  | 7      |              | SPECIALTY UNIFORM          | 1/15/2001  | 12/27/2001 | \$ 212.42   |
| 18 | SUM    |            |        |              |                            |            |            | \$ 212.42   |
| 19 | 020012 | 12/4/2001  | 7      | \$370,000.00 | PACIFIC EXTRUSION SYS      | 1/7/2002   | 1/4/2002   | \$ 9,468.75 |
| 20 | 020012 | 12/4/2001  | 7      | \$370,000.00 | PACIFIC EXTRUSION SYS      | 1/7/2002   | 1/4/2002   | \$10,233.04 |
| 21 | 020012 | 12/4/2001  | 7      | \$370,000.00 | PACIFIC EXTRUSION SYS      | 1/7/2002   | 1/4/2002   | \$ 1,090.04 |
| 22 | 020012 | 12/4/2001  | 7      | \$370,000.00 | PACIFIC EXTRUSION SYS      | 1/7/2002   | 1/4/2002   | \$ 514.42   |
| 23 | SUM    |            |        |              |                            |            |            | \$21,307.25 |
| 24 | 020013 | 12/4/2001  | 8      |              | PACIFIC EXTRUSION SYS      | 1/7/2002   | 12/20/2001 | \$ 235.00   |

图 1.2 当对数据进行更加全面地分析时,这种直观明了的方法会产生相当大的问题

图 1.2 中的部署将引起以下几个问题:

- ☐ 数据排序——例如按照月份而不是按照 PO 来获得分类汇总——变得很笨拙。这些附加的总计行成为了障碍。
- ☐ 放入新数据和把 SUM 函数指向正确的单元格,常受到按键错误的影响。
- ☐ 每次插入新行,输入新数据和编辑 SUM 公式,所有的 SUM 公式都要进行重新计算。现在 Sisyphus 有如此多的数据,会使计算过程慢得难以接受。
- ☐ 用户借助诸如=SUM(H1:H7000)公式将无法得到所有发票的总计,这是因为 PO 的分类汇总也被包含在内。

当工作表中含有大约 50 个 PO、100 到 200 张发票时,这种方法也不能说是一个不好的想法。它实现起来速度快,运行却不清晰,但它确实是快。

当到工作表中发展到含有超出 3 000 个 PO 和超出 6 000 张发票时,事情就变得很棘手, Sisyphus 便会来找你。这些情况是在不知不觉中发生的:它们逐渐介入了部门的事务中,人们排着长队地使用它们,然后又厌恶地放弃了它们,“这就是我们通常的做事方法”。

一种稍微复杂的方法最后也许会使事情变得简单。以下就是一种可能:

1. 利用数据透视表或“高级筛选”创建一个惟一编号的 PO 列表。图 1.3 显示了具有惟一 PO 编号的数据透视表;在第 3 章“Excel 列表、名称和筛选”中讲述了如何使用 Excel“高级筛选”。

2. 假设第一步创建的列表在 AB3:AB3540 中。在 AE1 单元格中(例如),以数组形式输入下面的公式:

```
=SUM(IF($A$3:$A$6045=AB3,$H$3:$H$6045,0))
```

然后将这个公式复制和粘贴到 AE3:AE3540 中。图 1.4 显示了结果。

这条数组公式的开始是一个条件:

```
IF($A$3:$A$6045=AB3,
```

当 A2:A6045 中的值等于单元格 AB3 的值时条件计算为 TRUE,公式将继续运行:

```
$H$3:$H$6045,0))
```

| Count of PO # | Total |
|---------------|-------|
| 020001        | 30    |
| 020002        | 30    |
| 020003        | 18    |
| 020004        | 8     |
| 020005        | 8     |
| 020006        | 12    |
| 020007        | 7     |
| 020008        | 1     |
| 020009        | 12    |
| 020010        | 46    |
| 020011        | 6     |
| 020012        | 57    |
| 020013        | 36    |
| 020014        | 36    |
| 020015        | 124   |
| 020016        | 4     |
| 020017        | 1     |
| 020018        | 68    |
| 020019        | 3     |
| 020020        | 12    |
| 020021        | 2     |
| 020022        | 10    |
| 020023        | 1     |

图 1.3 数据透视表计算与  
每个 PO 有关联的  
发票记录数

| Count of PO # | Total | AE3           |
|---------------|-------|---------------|
| 020001        | 30    | \$ 31,590.87  |
| 020002        | 30    | \$ 42,221.47  |
| 020003        | 18    | \$ 16,014.08  |
| 020004        | 8     | \$ 5,438.35   |
| 020005        | 8     | \$ 315.00     |
| 020006        | 12    | \$ 1,367.05   |
| 020007        | 7     | \$ 9,000.00   |
| 020008        | 1     | \$ 6,877.00   |
| 020009        | 12    | \$ 7,565.94   |
| 020010        | 46    | \$ 349.87     |
| 020011        | 6     | \$ 385,952.50 |
| 020012        | 57    | \$ 8,434.45   |
| 020013        | 36    | \$ 4,230.35   |
| 020014        | 36    | \$ 647,064.64 |
| 020015        | 124   | \$ 7,150.02   |
| 020016        | 4     | \$ 16,517.00  |
| 020017        | 1     | \$ 10,054.51  |
| 020018        | 68    | \$ 5,970.00   |
| 020019        | 3     | \$ 4,000.00   |
| 020020        | 12    | \$ 6,892.00   |
| 020021        | 2     | \$ 13,596.00  |
| 020022        | 10    |               |
| 020023        | 1     |               |

图 1.4 在“公式编辑栏”中圈在公式外边的大括号  
表明它是一个数组公式

总之,对于 A 列中的某一行,当条件是 TRUE 时,公式将从 H 列返回相应的数值,否则它将返回数值 0。最后将这个逻辑值提交给 SUM 函数:

```
=SUM(IF($A$3:$A$6045=AB3,$H$3:$H$6045,0))
```

当在 A 列中发现 AB3 单元格中的 PO 编号,将值加到 H 列,否则加 0。

#### 提示

在以数组形式输入公式时,使用组合按键 Ctrl+Shift+Enter,而不是仅仅按下 Enter 按键。在第 2 章中会看到更多数组公式的实例。

上述系列步骤,产生了 3538 个数组公式,每个公式为每个 PO 编号都提供了发票金额总和。可以将这些结果与提取的每个 PO 金额进行比较——通过比较就可以知道哪个 PO 的金额超支,以及那一个还没有被用完。

在短暂的考虑后,你就会意识到这种方法行不通。问题在于看上去所有这些公式将会永远计算下去。而且在 Sisyphus 对工作表的内容进行任何修改时,所有公式都不得不进行重新计算(除非早已预见到将设置改为“手动重算”,之后记得将其改回“自动重算”)。在一台 CPU 为 1.8GHz 的 Pentium 4 计算机上,对图 1.1 中显示的数据进行重新计算至少要花费一分钟的时间。

即使是一分钟的时间等起来也是太长了。你的客户已经获取了大量的数据等待 Excel 来方便有效地进行处理。一种更好的方法是使用数据透视表,将 PO 编号当做行字段,将发票金额当做数据字段(如图 1.5 所示)。较好的方法是在真正关系数据库中存储 PO 和发票数据,因为这样做可以将所有的明细记录保存在工作簿之外。对于这样的布局,可以

- ☐ 将数据返回到工作簿的数据透视表中,指定外部数据源为数据位置。
- ☐ 将数据返回到工作簿的外部数据区域中,将“选择查询”当做数据源,对 PO 编号进行

|    | AA | AB             | AC           |
|----|----|----------------|--------------|
| 1  |    | Sum of INV AMT |              |
| 2  |    | PO #           | Total        |
| 3  |    | 020001         | \$31,590.87  |
| 4  |    | 020002         | \$42,221.47  |
| 5  |    | 020003         | \$16,014.08  |
| 6  |    | 020004         | \$5,438.35   |
| 7  |    | 020005         |              |
| 8  |    | 020006         | \$315.00     |
| 9  |    | 020007         | \$1,367.05   |
| 10 |    | 020008         | \$9,000.00   |
| 11 |    | 020009         | \$5,877.00   |
| 12 |    | 020010         | \$7,565.94   |
| 13 |    | 020011         | \$349.87     |
| 14 |    | 020012         | \$365,952.50 |
| 15 |    | 020013         | \$8,434.45   |
| 16 |    | 020014         | \$4,230.35   |
| 17 |    | 020015         | \$647,064.64 |
| 18 |    | 020016         | \$7,150.02   |
| 19 |    | 020017         | \$16,517.00  |
| 20 |    | 020018         | \$10,054.61  |
| 21 |    | 020019         | \$5,970.00   |
| 22 |    | 020020         | \$4,000.00   |
| 23 |    | 020021         | \$6,692.00   |
| 24 |    | 020022         | \$13,596.00  |
| 25 |    | 020023         |              |

图 1.5 在数据透视表中显示的数字是数值,不是必须重新计算的公式

分组(参见第4章和第5章中关于“选择查询”的介绍)。

在上述任何一种情况下,最终使用的是数值,而不是公式,显示的是每个 PO 发票金额总计。这就意味着 Sisyphus 无论在任何时候改变了内部数据或者某个人编辑了工作表,都不必等待 3538 次的重新计算,因而取消了重新计算。即使没有使用重新计算公式,当有新的发票到来或创建了新的 PO 时,它能够方便快捷地更新总计;客户所有要做的工作就是刷新数据透视表或外部数据区域。

## 转置行与列

Excel 不是一个数据管理系统。例如,它不是被设计用来管理父记录和子记录之间关系的。若要扩展上一节的举例,那么 PO 就是父记录,发票就是子记录,这是因为每张发票(子记录)都从属于一个 PO(父记录)。

然而,Excel 还是具有一些基本的数据库特性,其中包括“自动筛选”和“高级筛选”。这两种筛选尤其依赖于数据的具体走向,称为列表。在 Excel 中,列表具有以下特点:

- ☐ 每个不同记录(每个人、每件产品、每张发票)占用不同的行。
- ☐ 每个不同变量(例如,名称和地址,或者型号和价格,或者发票日期和金额)占用不同的列。
- ☐ 每一列的开始都有这一列的变量名。

图 1.1 是一个列表的实例。在不同的行上有不同的记录——发票,在不同列上有不同的变量——日期、金额等等,每一列都以这一列的变量名为标题。

## 利用选择性粘贴改变走向

当指定的数据部署在列表中时,许多 Excel 工具功能都很强大,而有一些工具功能单一。