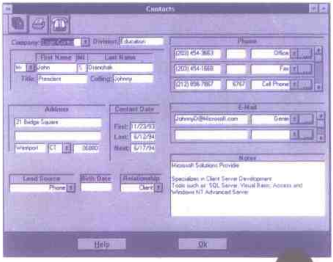
〔美〕John S. Dranchak Joseph R. La Croce 著
郭勇 郭小莉 施思 王谨 译
陈隆 审校



构建 ACCESS™ 2 应用程序

WILEY
電子工業出版社 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

构建 ACCESS 2 应用程序

[美] John S. Dranchak Joseph R. LaCroce 著
郭 勇 郭小莉 施 思 王 谨 译
陈 隆 审校

電子工業出版社

内 容 提 要

Microsoft Access 2.0 是一个强有力且易于使用的数据库管理系统。《构建 Access 2 应用程序》一书将告诉读者如何将 Access 的各种部件组装在一起。本书是为利用 Access 2.0 构建应用程序的人员而编写的，即适用于专业开发人员也适用于最终用户。本书面向那些曾经使用过 Access 但不知道如何将各种部件组装在一起来构建完整应用程序的用户。本书假定读者具备表、查询、表单和报表的基本知识。本书不是 Access 的另一参考手册，本书通过一个构建联系管理应用程序(WinContact)的实例指导读者构建自己的 Access 应用程序。在第一部分过后，将以 WinContact 为例，详细介绍构建 Access 应用程序的各个步骤。

构建 ACCESS 2 应用程序

[美] John S. Dranchak　Joseph R. LaCroce　著
郭 勇　郭小莉　施 恩　王 谦　译
陈 隆　审校

特约编辑　辛再甫
责任编辑　赵 平

*

电子工业出版社出版
北京市海淀区 173 信箱（100036）
电子工业出版社发行　各地新华书店经销
北京市顺义县振华印刷厂印刷

*

开本：787 × 1092　1/16　印张：18.75　字数：476.8 千字
1996 年 4 月第 1 版　1996 年 4 月第 1 次印刷
印数：4000 册　定价：32.00 元
ISBN 7-5053-2980-4/TP・1027
著作权合同登记号
图字：01-95-918

前　言

本书的读者对象

Microsoft Access 2.0 是一个强有力且易于使用的数据库管理系统。许多 Access 用户都未能充分利用其功能来方便地构建图形数据库应用程序。他们虽然掌握了 Access 中使用的某些部件，却不能够将这些部件组织在一起，以构造完整的应用程序。这并不是由于这是一个较困难的过程，而是因为缺少有关如何将这些部件组装在一起的指导，或是因为 Access 用户对 Access 采用的基于对象的事件驱动机制还不太了解。《构建 Access 2 应用程序》一书是使用这种机制的指导，它将告诉读者如何将 Access 的各种部件组装在一起。

本书是为利用 Access 2.0 构建应用程序的人员而编写的，即适用于专业开发人员也适用于最终用户。本书面向那些曾经使用过 Access 但不知道如何将各种部件组装在一起来构建完整应用程序的用户。本书假定读者具备表、查询、表单和报表的基本知识。

本书的组织方式

本书分为三个部分，共十章。本书不是 Access 的另一参考手册，本书通过一个构建联系管理应用程序(WinContact)的实例指导读者构建自己的 Access 应用程序。在第一部分过后，将以 WinContact 为例，详细介绍构建 Access 应用程序的各个步骤。

第一部分——"应用程序设计"由四章组成，介绍应用程序的基本概念和开发方案。第一章"什么是应用程序"给出了数据库管理系统的定义，介绍 Microsoft Access，并定义了什么是 Access 应用程序。第二章"Dranchak 开发方案"向读者介绍什么是开发方案，并详细讲述构造以数据为中心的应用程序的简单有效的方案。这一章还介绍了本书使用的实例——WinContact 应用程序。第三章"Windows 设计基础"，阐述了 Microsoft Windows 界面是用户和开发人员的首选界面的原因，并讲述了如何设计和实现好的用户界面。第四章"信息建模"，介绍信息建模技术，并说明如何建立 WinContact 应用程序的逻辑数据模型。

第二部分——"没有代码的应用程序"由四章组成，讨论用标准 Access 对象生成应用程序的过程，但不涉及宏和模块。第五章"表和数据库"介绍如何生成 WinContact 应用程序所需的表，并概要介绍了将被以后生成的对象继承的一些特性。第六章"查询"讨论表与查询的关系。第七章"表单"解释什么时候使用表单，以及如何生成表单。第八章"报表"讨论向最终用户提供信息的报表。

第三部分——"用宏改进应用程序"由两章组成，是本书的重点，详细介绍如何将数据库中的各种对象组装在一起，构建一个完整的应用程序。第九章"宏"介绍什么是宏，以及如何生成、编辑和存储宏。第十章"用宏修改用户界面"教读者如何用已经建立的其它一些对象生成定制用户界面，从而结束 WinContact 应用程序的构建。

目　录

第一部分

第一章　什么是应用程序 …… (1)
1.1　本章内容 …… (1)
1.2　信息和数据 …… (1)
1.2.1　数据 …… (1)
1.2.2　信息 …… (2)
1.3　什么是数据库 …… (2)
1.4　什么是数据库管理系统 …… (3)
1.4.1　数据库简史 …… (4)
1.4.2　第一次最终用户革命 …… (4)
1.4.3　第二次革命的曙光 …… (4)
1.4.4　第三次革命 …… (5)
1.5　什么是 Access …… (7)
1.6　什么是 Access 应用程序 …… (7)
1.6.1　Access 构造块 …… (8)

第二章　Dranchak 开发方案 …… (11)
2.1　实例研究:Saugatuck Marketing Group …… (11)
2.1.1　SMG 技术剖析 …… (11)
2.2　方案 …… (12)
2.2.1　第一步:认识问题 …… (13)
2.2.2　第二步:建立信息模型 …… (18)
2.2.3　第三步:创建界面 …… (20)
2.2.4　建立自己的报表 …… (23)
2.2.5　设计和实现用户界面 …… (23)
2.2.6　第四步:测试应用程序 …… (27)
2.2.7　第五步:建立文档 …… (30)
2.2.8　印刷文档 …… (31)
2.2.9　第六步:发行应用程序 …… (31)

第三章 Windows 设计基础 …… (33)
3.1 Windows 界面概述 …… (33)
3.2 窗口和表单 …… (34)
3.2.1 父、子窗口和表单 …… (34)
3.2.2 窗口部件 …… (36)
3.2.3 菜单 …… (41)
3.2.4 窗口模式 …… (44)
3.3 窗口控件概述 …… (45)
3.3.1 Access 支持的控件 …… (46)
3.3.2 Access 不支持的控件 …… (55)
3.4 什么是优秀的界面 …… (56)

第四章 信息建模 …… (59)
4.1 什么是信息建模 …… (59)
4.2 关系数据库 …… (60)
4.2.1 关系理论 …… (60)
4.2.2 实体、属性和关系 …… (62)
4.2.3 关键字 …… (65)
4.2.4 基数/约束 …… (70)
4.2.5 范式化 …… (72)
4.3 CASE 工具 …… (83)

第二部分

第五章 表和数据库：把逻辑设计转换为物理模式 …… (85)
5.1 数据库 …… (85)
5.2 表 …… (86)
5.2.1 数据类型 …… (86)
5.2.2 表层特性 …… (87)
5.2.3 建立表 …… (96)
5.2.4 建表工具 …… (104)
5.3 定义主关键字 …… (129)
5.3.1 Set Primary Key 按钮 …… (130)
5.3.2 索引窗口 …… (130)
5.4 创建索引 …… (132)
5.4.1 使用 Indexed 特性 …… (133)
5.4.2 索引窗口 …… (133)
5.5 定义关系 …… (135)
5.6 测试表 …… (140)

第六章 查 询 …… (143)
6.1 什么是查询 …… (143)
6.1.1 查询的类型 …… (143)
6.2 构建查询 …… (144)
6.2.1 查询设计窗口 …… (144)
6.2.2 构建 WinContact 表单查询 …… (145)
6.2.3 构建 WinContact 报表查询 …… (150)
6.2.4 使用来自表单的值作为一个宏的准则 …… (160)

第七章 表 单 …… (163)
7.1 什么是表单 …… (163)
7.2 构造表单 …… (163)
7.2.1 表单的段 …… (165)
7.2.2 特性 …… (166)
7.2.3 事件处理 …… (167)
7.2.4 给表单添加控件 …… (168)
7.2.5 往表单中添加命令按钮 …… (174)
7.2.6 调色板 …… (175)
7.2.7 在表单中使用子表 …… (183)
7.2.8 选项组 …… (194)
7.2.9 列表框 …… (195)
7.2.10 开关按钮 …… (199)
7.2.11 使用组合框 …… (204)

第八章 报 表 …… (213)
8.1 什么是报表 …… (213)
8.1.1 报表的类型 …… (214)
8.2 构造报表 …… (217)
8.2.1 Report Design 窗口 …… (219)
8.2.2 用 Report Wizards 来构造 WinContact 报表 …… (222)
8.2.3 从头开始构造报表 …… (238)

第三部分

第九章 宏 …… (247)
9.1 什么是宏 …… (247)
9.2 Macro 窗口 …… (247)
9.2.1 了解窗格 …… (247)

9.2.2 条件表达式 …… (248)
9.3 创建宏 …… (249)
9.3.1 应用程序的启动 …… (250)
9.3.2 宏动作 …… (252)
9.3.3 宏组 …… (253)
9.3.4 测试宏 …… (273)

第十章 用宏修改用户界面 …… (275)
10.1 Menu Builder Add-In …… (275)
10.2 手工创建菜单宏 …… (279)
10.3 创建 WinContact 的菜单 …… (280)

第一章

什么是应用程序

1.1 本章内容

Microsoft Access 是一个数据库管理系统，它可以用来建立定制数据库应用程序，以满足用户的需求。本章讲述数据库管理系统的基本知识，以及 Access 应用程序的组成。本章的具体内容有：

- 数据和信息的区别。
- 什么是数据库。
- 什么是数据库管理系统。
- 形成 Access 市场的一系列事件。
- 为什么对 Windows 数据库管理系统具有强烈的需求。
- Access 是如何利用 Windows 环境的。
- Access 应用程序的组成。

1.2 信息和数据

1.2.1 数据

数据是广义上的原始输入，除非在特定的环境中，数据可以有多种解释，以下列字符串为例：

201
203
212
516
718
914

除了以升序排列和具有三位数字外，它们还意味着什么呢？它们是某项试验的输出结果呢，还是某个城市的某条街道上的门牌号码？也许，它们是昨晚的彩票中奖号码。以下列出了另一组数据：

Connecticut
New Jersey
Manhattan
Long Island
Queens
Westchester

这组数据的意义也许要比前一组明显些。毕竟，Connecticut 和 New Jersey 是州名。但其它数据代表什么呢？居住在纽约城(New York City)的人会告诉你，Queens 和 Manhattan 是纽约城的两个区，Westchester 则是纽约城北部的一个县，而 Long Island 则是纽约城东部 Connecticut 南部的一个大岛。但是，如果不熟悉纽约地区的地理情况，就不能得出这些结论。无论如何，问题仍未解决：这些数据到底代表什么？它不仅仅是个州名的列表，也不仅仅是纽约城的一些城区的列表。也就是说，你不知道这些数据意味着什么，因为你不知道解释这些数据的具体环境。从另一个角度说，上面列出的六个词之间没有明确的关系。为了定义一个关系，你也许要把前面两组数据列在一起，如表 1.1 所示，不论怎样解释，这两组数据都有六个元素。现在，将左列的地名和右列的数字联系在一起，就可能得出新的解释。但该表不会告诉你这种关系是什么。右边的数字是位于同一行上的地址的 Microsoft Access 用户呢还是由联邦政府维护和管理的州际高速公路的里程呢？

表 1.1 地名和数字

地名	区号
Connecticut	203
New Jersey	201
Manhattan	212
Long Island	516
Queens	718
Westchester	914

1.2.2 信息

实际上，右边的数字是左边地名的区号。但是，在你把这两组数据联系在一起之前，你所拥有的仅仅是一些数据而已。将这两组数据组合在一起，给每一列加一个标签，就可以推断出一个关系，从而获得信息。信息是特定环境中的数据。

1.3 什么是数据库

数据库是数据的有组织的集合，合理地使用这些数据就可以得到信息。虽然本书侧重于用计算机管理用户的数据，不过数据库却并不一定是基于计算机的。一本包含有商务联系人和朋友的姓名、地址和电话号码的地址簿是一个数据库；放置你的所有的供应商的产品目录的文件柜中的抽屉也是一个数据库。尽管你不一定知道，在你的地址簿和文件柜中的抽屉之

间甚至也存在着某种关系，因为你已经将每一个供应商都列入了你的地址簿。因此，如果你拥有某位供应商的产品目录，在你的地址簿中也就有了有关该供应商的一项。有时，你可能查找产品目录，并决定给供应商打电话，订购其产品。为此，你再次查找产品目录，从中找出该供应商的电话号码。在另一些时候，当你想给供应商打电话时，你不必从产品目录中查找其电话号码，而是打开桌上的地址簿，从中查找所需电话号码。换言之，通过决定需要什么信息以及从何处获得信息，你在管理数据库。

表 1.2　纽约地区的地名及其区号

地名	K号
Connecticut	203
New Jersey	201
Manhattan	212
Long Island	516
Queens	718
Westchester	914

你也许按具有特定意义的方式对数据进行过组织。当你在地址簿中添加供应商时，可能在该供应商的名下建立一个项，然后写入联系人的姓名。或者，也可能在联系人的名下添加一项，经过后来的考虑，纳入公司的名称。你按某种次序排放文件柜中的产品目录。也许，它们是根据供应商的姓名，按字母顺序插入的。也许，你根据产品类型的字母顺序排放它们。你还可能按先进先出(FIFO)的方式，将最近的产品目录放在抽屉的最前面，而不管其来自哪位供应商或属于哪一类产品。这里，由你决定这些目录的物理排放次序，你再次成为数据库管理员。

1.4　什么是数据库管理系统

虽然，人脑能够管理大量数据(和信息)，但这是有限的。在信息时代，越来越多的数据需要吸收和分析。另外，也许还有其它用户需要你的数据。也许你正在休假，而你的同事 Bob 需要给供应商打电话。虽然 Bob 知道所需产品目录在哪个文件柜的哪个抽屉中，但他也许不知道你是按供应商代理的姓氏的字母顺序排放产品目录的。虽然，他最终能够找到所需目录，但如果他事先知道数据是怎样组织的，会化少得多的时间。如果他可以向数据库管理员提出请求，而不必处理潜在的数据，则会化更少的时间，因此，不仅数据库管理员要处理数据的物理事务和数据存储，而且他还要向需要使用数据的其他成员提供一致的访问数据的方式。这就使得可以用数据来帮助他们进行工作，而不致于使他们做数据库管理员所做的工作。

这就提出了一个问题——如何有效地管理数据及其关系，并为使用数据的所有人员提供公共的访问方式。解决的方法是使用专门设计用来完成这种任务的工具，即数据库管理系统。数据库管理系统(DBMS)是用来操作数据库的应用软件。数据库管理系统的唯一目的是将数据转化为信息，并提供对这些信息的访问手段。小至用于便携式个人信息管理的 Sharp Wizard，大到在 IBM 3090 系列主机上运行的具有太字节数据的 DB2，数据库管理系统的形

式和规模不尽相同。

1.4.1 数据库简史

计算机数据库管理系统最初只在大型主机上运行。如果没有大型主机，就不能拥有DBMS。因此，由于经济原因，DBMS 实际上仅仅为大公司、研究机构和政府部门所有。随着小型机的引入，许多买不起大型主机的小公司和研究所也能够获得较便宜的硬件以及在这种平台上运行的 DBMS 了。

传统上，数据库管理系统是由信息系统部门(ISD)合作实现和维护的。最终用户很少或不与物理 DBMS 打交道。虽然在 DBMS 的设计和实现阶段要向用户咨询，但用户最终仅获许请求报表或在有限的屏幕空间内输入数据。虽然 DBMS 是设计用来帮助最终用户的，但由于早期系统的复杂性及其对 ISD 的依赖性，用户经常不能获得他们所需的信息。ISD 也理解这些用户的需求，但在过去由于可用资源的耗尽，DBMS 不能处理用户的请求，从而导致事务积压。好信息通常具有较强的时间性，当用户花数月乃至数年时间等待某个系统被开发出来后，所需信息也已经贬值了。这常使用户面临困境，便得他们称 ISD 为"象牙塔"，因为他们不能控制这种情况。虽然，这对最终用户来说是不公平的，但也不是 ISD 之过，至少最初不是。

主机和小型机较为复杂和神秘，需要专门的技术来操作。除了操作硬件本身外，分析员、设计员和程序员还需要编制程序使硬件完成所需的任务。虽然近来越来越多的入具有上述技能，但计算机系统的专业人员仍显得供不应求。而最终用户往往不具备这些技能。另外，大多数最终用户对数据库和计算机编程本身并不感兴趣，他们只是想获得完成他们的工作所需的信息。虽然许多最终用户面临困境，但数据库的这种模式在当时是必须的(现在某些时候仍然如此)。实际上，最终用户没有对数据库进行操作的权力。

1.4.2 第一次最终用户革命

后来，情况发生了变化，引入了一系列称为 4GL 的(第四代语言)新的计算机编程工具。第四代工具，如 RAMIS 和 NOMAD，允许用类似英语语句的方式向计算机输入数据请求，不象传统的编程语言(如 COBOL、FORTRANT 和 APL)的晦涩难懂的语法。这使得最终用户可以建立自己的查询，访问数据库中的数据。最终用户可以自己建立报表，而不必向程序员请求一份新的报表格式。虽然许多程序员对最终用户数据操作有所顾虑，但结果往往都很好，因为程序员可以从建立报表的事务中解脱出来，去完成更重要的项目。由于 ISD 仍然可以控制数据库的输入和结构以及和保密数据有关的其它系统程序，即使最终用户可以对数据库进行操作，数据库也还是能够良好运行的。从建立格式化报表的事务中节省下来的时间可以用于开发设计更好的应用程序。

但是，一旦给最终用户一些权力，告诉他们如何去完成在 4GL 出现前只有 ISD 才能完成的任务，就打开了潘多拉魔盒。用户要求对数据库和个人应用程序具有更多的控制权。然而，直到 1982 年之前，他们是无法获得这种控制权的。

1.4.3 第二次革命的曙光

虽然个人计算机曾一度被人们不屑一顾，但当 1982 年 IBM 推出其 PC 机之后，个人计

算机便普及开来了。专眼之间，世界上所有面临困境的最终用户都获得了他们自己可以理解和控制的硬件，他们能够建立、编辑和操纵自己的数据，而不必与 ISD 打交道了。重要的是，PC 机的价格及其软件使得小公司和个人也能够承受得起。因此，每一个人都似乎成了赢者；最终用户获得了对其数据的更多的控制、节省了获取所需信息所化的时间；小公司和个人也买得起以往大公司才能负担的计算工具了。

但 PC 机仍然存在着问题。从合作共事的角度来看，PC 机只是一座座信息孤岛而已。虽然现在网络已经随处可见，但这毕竟历经了许多年头。在此之前，当用户需要大型主机产生的数据时，不得不打印在纸上，然后手工输入 PC 机。如果用户用 PC 机生成了一个报表，其中的大多数数据最终仍要重新输入到大型主机或小型机上的大系统（如大的会计系统）中。虽然曾经出现过许多针对这些问题的有创意性的解决方案，但是管理和维护孤立 PC 机上的数据的概念要比管理中央数据源复杂得多。因而许多 ISD 职员反对使用 PC 机。

即使到了今天，对于用网络连接在一起的 PC 机而言，数据孤立和冗余的问题仍非常严重。如果两个用户各自拥有一个数据库，分别存有某公司的总嬴余图，那么到底哪一个的数据是准确的呢？但愿他们具有相同的数据。不过，通常并不这样理想，某一个的数据要比另一个稍稍正确些，这样就造成了数据的歧义生，最终导致基于错误假设的商务决策。另外，在 PC 机应用的早期，应用程序在结构、语法和用户界面的一致性方面做得很不好。这使得学用新的应用程序成为一项烦锁而废时的事情。学用某个应用程序获得的知识大多数不能适用于后来使用的其它应用程序。因而，大多数用户只能使用两到二个应用程序，PC 机应用程序很丰富，但它们并不总是直观的、不易掌握。更糟糕的是，用户需要经常重新开始使用单个软件包，如 Lotus1-2-3。当发现某个软件包不能满足其需要时，用户不能更新或重新开发，只得去购买新的软件包，以满足其新的需求。他们所学的操作先前应用程序的知识根本无法用来操作新的软件，因为新的软件包通常是由其它公司设计的，而且几乎不遵循什么设计标准。类似 WordStar 的字处理程序的外观和行为与一个电子报表软件大不相同。

从另一个角度来说，用户的工作效率还是得到了提高，他们可以用类似电子报表软件的工具去完成以前用计算器甚至用笔和纸完成的工作。字处理程序使建立文档的工作效率得以大幅度提高，而图形表示程序则大大改善了数据表示方式。

然而，这些收获大多数仍局限在个人工作效率的提高。用户的工作效率虽然提高了，他们仍然缺少建立定制应用程序的手段，因为 PC 机应用程序仍然要用传统的编程语言开发。随着 Ashton-Tate 的 dBASE 产品系列的出现，用户可以用一系列新的工具在 PC 机上进行定制开发，而不必使用传统的开发语言了。dBASE 还提供"以数据为中心"的应用程序视图，强迫用户和开发人员按数据库的方式进行思考，输入表单和报表。虽然，大多数最终用户还未能掌握这些语言，但它毕竟为新一类开发人员打开了方便之门，使他们不再需要用传统语言进行数据库开发。这使得从大公司到小公司的许多用户能够开发他们自己的定制应用程序。这些工具同样受到许多有眼光的 ISD 部门的欢迎。转眼之间，最终用户和系统专业人员都在进行 PC 机开发工作了。

1.4.4　第三次革命

1985 年 Microsoft 公司引入了用于 DOS 的图形用户界面（GUI）Windows，不过直到 1990 年，当得到巨大改进的 Windows 3.0 版问世后，Windows 才普及开来。虽然 Windows

是基于 DOS 的，并且仍然具有 DOS 的一些限制（如 16 位体系结构、640 基本内存限制等等），不过它具有解决这些限制和许多其它问题的措施，并提供了全新的外观。Windows 比 DOS 好的主要优点也许在于：

- 一致的图形用户界面。
- 应用程序间共享数据的功能（通过 DDE 和 OLE）。
- 任务切换。
- 有限多任务。
- 设备独立性。
- 对 640k 以上内存的使用。
- 由操作系统而不是应用程序处理外设（打印机等）。
- 运行 DOS 应用程序的功能（在一个窗口中或在全屏幕）。
- 一种对象/动作机制（它的确是一个面向对象的界面，但却不是用面向对象的编程工具开发的）。
- 即包含传统工具和语言（如 Basic，C 和 Pascal），又包含可视化编程工具（如 Visual Basic，Visual C++和 ObjectVision）的开发环境。

未来的 Windows 版本将克服前面所述的 DOS 限制（由其对成熟的、32 位 Windows 操作系统而言），并增加以下优点：

- 规模可变的 32 位体系结构。
- 集成的网络支持。
- 对称多处理。
- 运行为其它操作系统（UNIX，OS/2 等）所编写的应用程序的功能。
- 标准化的用于所有 Windows 应用程序的宏语言，可以通过它轻而一举地开发紧密集成的定制套装软件（Visual Basic，Applications Edition）。

注释：

因为 Windows 系列不再是单一的产品，并不是所有的 Windows 成员都具有上述特性。

在 Microsoft 自己的 Excel 和 Word for Windows 的引导下，Windows 应用程序（如电子报表程序和字处理程序）的销售势头很旺，然而，在 1992 年秋季，当 Access 刚刚问世时，少数几个可以选用的 Windows 数据库管理系统没有一个能够在市场上站稳脚跟。用 Borland 的 dBASE 和 Paradox 开发过独立应用程序的专业开发人员，以及不满足于电子报表软件提供的非关系数据库功能的最终用户都迫切需要一个好的数据库管理系统。Access 就是在这种情况下进入市场的。

1.5　什么是 Access

Microsoft Access 是一个基于 Windows 的关系数据库管理系统，它真正发挥了 Windows 的长处。它不是现有 DOS 数据库的翻版，而是纯粹为 Windows 环境设计的。它具有以下特性：

- 拖动和放置的建立，以及对查询、报表、宏和表单的编辑。
- 支持 Windows 多文档界面(MDI，Multiple Document Interface)。
- 能够包含 OLE 对象(如声音、视频或者硬盘上存储的其它内容)，将其作为数据库中的记录或表单和报表中的对象。
- 指导用户进行复杂操作的 Wizard(指导程序)。
- 强有力的宏语言，可以用来自动完成日常任务，而无需编程。
- Access Basic 编程语言，它是根据被广泛应用的 Visual Basic 设计的，并具有附加数据库扩展。
- 可以用 Access 开发工具箱(ADT，Access Developers Toolkit)生成可发行的、免版税独立应用程序。
- 可以同时附连到或者交叉连接其它格式的数据库(如 dBASE，Paradox 和 Btrieve)。
- 可以按几种标准格式，如 Excel，Lotus 1-2-3，Fox Pro 和 ASCII，引入和引出文件。
- 规模可变的体系结构，这使得 Access 可以作为独立的数据库、多用户文件服务器数据库以及任何支持 ODBC 的客户机-服务器的核心使用。
- 在数据库核心层支持引用完整性。
- 数据校检如果在表层建立，可以自动迁移到表单和报表中。
- 可以链接任何 Windows 动态链接库(DLL)的可扩展的体系结构。
- 可以建立和使用可重复使用代码库。
- 可以建立和使用第三方 Wizard。

如果读者熟悉 Windows 产品，如 Excel 和 Word for Windows，就会觉得 Access 很面熟。它提供工具条、一个浮现工具框、标尺(如有必要)和熟悉的打印预览功能(Print Preview)。它还有广泛的联机帮助。

1.6　什么是 Access 应用程序

Access 应用程序是一些为了解决某个问题而共同工作的 Access 对象集。问题和解决方案可简可繁。例如，如果要显示与邮政编码相对应的城市，则相应的应用程序就有可能相对

简单些。事实上，你可能并不认为这是一个应用程序。相反，如果你在管理一个小航空公司，要使订票过程、飞行调度、职员管理和飞机维护工作实现自动化，实际面临的问题就要复杂得多。它实际上是几个问题，可以用集成的方式解决，相应的应用程序会很复杂。有时会出现并不常见的情况，即问题的复杂程度和应用程序的复杂程度呈反比。也就是说，简单的问题会导致复杂的应用程序(通常是由于应用程序设计得极不合理)，也可能是复杂的问题却可用简单的应用程序(由于好的应用程序设计)解决。

1.6.1 Access 构造块

Access 有七类对象，它们是所有 Access 应用程序的基本构造块：

- 数据库是存放所有其它对象的容器。Access 数据库对象不同于本章前面描述的数据库。传统的数据库只存放原始数据和有关数据间关系的信息。Access 数据库对象也存储原始数据和关系信息，但它们还存放和数据相联的任何其它东西，包括表单、报表、宏和程序。图 1.1 说明了这种概念体系。一个 Access 应用程序至少含有一个数据库，即使是其中没有表。

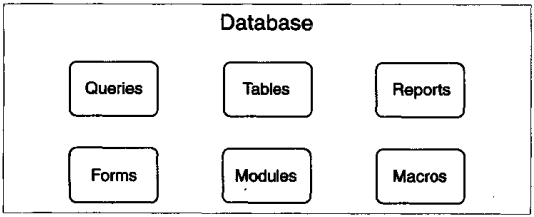


图 1.1 概念体系

- 表是存放实际数据的对象。
- 查询是用来从一个或多个表中提取信息(用类似于表的格式)的对象。查询还用来在表上进行各种修改数据的操作。选择查询(view)是与 SQL Server 中的察看(view)相并列的。
- 表单用于输入、编辑和察看数据。表单是用户用来同数据打交道的界面。
- 报表用来按特殊的格式提取和打印数据，还具有对成组或相关数据进行汇总和其它计算的功能。
- 宏是预定义的动作链，它通过操纵其它对象自动完成重复任务和修改用户界面。
- 模块是用 Access Basic 代码(ABC，Access Basic Code)生成的，用于高级应用程序的定制过程。当宏和其它对象不能满足应用程序的需要时就使用模块。

除了宏和模块外，本书假设读者至少熟悉上述基本概念。如果读者不了解上述概念，也可以阅读本书，但应该花时间看一下 Access 手册和教程。

有了这七类构造块，任何人都能用 Access 生成应用程序。实际上，这七类对象不一定都用得到。仅用一数据库、表和表单就能构建一个具有有限功能的应用程序。另外，还有三种与上述七类对象打交道的对象：

- 动态集是查询返回的结果数据集。它们完全是可编辑(除非使用右连接 right joins 或应用程序被设计成不允许编辑)和更新的。
- 快像是动态集的静态图片，不能更新和编辑。
- 控件是表单和报表上使用的对象，用以接受用户输入以及在屏幕或硬拷贝上显示程序输出。

如果你有兴趣创建免版税、可发行应用程序，不论在本组织内发行还是用于再次销售，都需要 Access 开发工具箱(ADT)。ADT 提供了几个工具，用以帮助用户将各种构造块组装在一起，以建立一种自由结构，这些工具有：

- Windows Help Compiler(Windows 联机帮助编译程序)。
- 几个帮助用户建立定制帮助文件的实用程序。
- 一个让用户用来为其应用程序建立定制安装(setup)程序的程序。
- 发行应用程序所需的所有运行时文件。
- 几个 OLE 定制控件。

本书的目的就是指导读者将这些对象组装在一起，以生成有用的应用程序，从而提高读者及其用户的工作效率。