

目 录

第一章 Access 97 快速入门

1.1 安装 Access 97	(1)
1.1.1 运行 Microsoft Access 所需配置	(1)
1.1.2 安装 Microsoft Access 97	(1)
1.2 Microsoft Access 97 的新特点	(2)
1.2.1 Visual Basic for Application——Microsoft Access 97 的编程语言	(2)
1.2.2 新的开发环境	(2)
1.2.3 Internet 特性	(6)
1.3 Access 97 的系统组成	(8)
1.4 浏览 Access 97	(9)
1.4.1 运行 Microsoft Access	(9)
1.4.2 打开数据库	(11)
1.4.3 Access 97 的界面	(13)
1.5 表的使用	(14)
1.5.1 数据表视图下的表窗口	(14)
1.5.2 设计视图下的表窗口	(15)
1.6 查询的使用	(16)
1.6.1 数据表视图下的查询窗口	(16)
1.6.2 设计视图下的查询窗口	(16)
1.7 窗体的使用	(17)
1.7.1 窗体视图下的窗体	(17)
1.7.2 设计视图下的窗体	(18)
1.7.3 数据表视图	(19)
1.8 报表的使用	(19)
1.9 宏的使用	(21)
1.10 模块的使用	(22)
1.11 帮 助	(24)
1.12 退出 Access 97	(25)

第二章 建立 Microsoft Access 中的数据库

2.1 建立新的数据库	(26)
2.1.1 创建新的数据库	(26)
2.1.2 用“数据库向导”创建新数据库	(29)

2.2 数据库窗口与工具条	(30)
2.3 创建表	(32)
2.3.1 设计画面	(33)
2.3.2 设计画面的工具条	(33)
2.4 定义字段	(35)
2.4.1 字段数据类型	(36)
2.4.2 字段特性	(37)
2.4.3 完成表中字段的创建	(39)
2.4.4 定义字段有效性规则	(40)
2.4.5 定义输入掩码	(43)
2.5 定义主关键字	(45)
2.6 定义表属性	(46)
2.7 表的存储	(47)
2.8 使用表向导创建表	(47)
2.9 视图	(49)
2.10 定义关系	(50)
2.11 增加索引	(52)
第三章 修改数据库设计	
3.1 准备工作	(56)
3.1.1 备份表	(57)
3.1.2 复原	(58)
3.1.3 将数据库升级至 Microsoft Access 97	(58)
3.1.4 打开未转换的旧版本数据库	(59)
3.2 删除表	(60)
3.3 更改表名	(61)
3.4 修改字段名	(62)
3.5 插入新字段	(62)
3.6 移动字段	(64)
3.7 拷贝字段	(67)
3.8 删除字段	(68)
3.9 更改字段属性	(70)
3.9.1 修改数据类型	(70)
3.9.2 更改字段大小	(71)
3.9.3 更改字段属性的后果	(71)
3.10 更改主关键字	(73)
3.11 修改表之间的关系	(74)
3.11.1 删除表中已有的关系	(74)
3.11.2 编辑表中已有的关系	(74)
3.11.3 添加新的关系	(75)

第四章 使用数据表

4.1 数据表视图	(76)
4.2 设置数据表格式	(78)
4.2.1 改变字体	(78)
4.2.2 单元格效果	(79)
4.2.3 改变行高和列宽	(80)
4.2.4 隐藏和显示列	(81)
4.2.5 移动列	(82)
4.2.6 重命名列	(82)
4.2.7 冻结列	(82)
4.3 修改数据	(83)
4.3.1 记录指示符	(83)
4.3.2 增加一个新的记录	(84)
4.3.3 选择和修改数据	(84)
4.3.4 替换数据	(86)
4.3.5 复制数据	(86)
4.3.6 删除记录	(87)
4.4 数据的排序与查找	(88)
4.4.1 排 序	(88)
4.4.2 查找数据	(89)
4.4.3 筛选数据	(90)
4.5 打印数据	(93)
4.5.1 打印预览	(94)
4.5.2 打印设置	(95)
4.5.3 打 印	(96)

第五章 选择查询

5.1 查询的“设计”视图	(97)
5.1.1 “设计”视图	(97)
5.1.2 “设计”视图的工具栏	(99)
5.1.3 查询属性设置	(100)
5.2 创建选择查询	(101)
5.2.1 添加表/查询	(102)
5.2.2 更改联接	(102)
5.2.3 删除表	(103)
5.2.4 添加字段	(103)
5.2.5 设置字段特性	(105)
5.2.6 排序次序	(105)
5.2.7 显示字段	(106)
5.2.8 输入准则	(106)

5.2.9 计算值	(109)
5.2.10 使用表达式生成器	(110)
5.2.11 指定字段名	(113)
5.2.12 合计查询	(113)
5.2.13 参数查询	(115)
5.3 交叉查询	(117)
5.4 使用查询向导	(119)
5.4.1 使用简单查询向导	(119)
5.4.2 交叉表查询向导	(120)
5.4.3 “查找重复项查询向导”与“查找不匹配项查询向导”	(122)
第六章 操作查询	
6.1 删除查询	(123)
6.2 更新查询	(124)
6.3 追加查询	(125)
6.4 生成表查询	(127)
第七章 窗体基础	
7.1 窗体的使用	(129)
7.2 窗体视图	(129)
7.3 窗体简介	(131)
7.3.1 窗体组成	(131)
7.3.2 多页窗体	(132)
7.3.3 连续窗体	(133)
7.3.4 子窗体	(133)
7.4 在窗体中操作数据	(134)
7.4.1 观察数据	(134)
7.4.2 增加新记录	(135)
7.4.3 修改与删除数据	(137)
7.5 查找和排序数据	(137)
7.5.1 执行一个简单查找	(138)
7.5.2 排 序	(138)
7.5.3 筛选数据	(138)
7.6 从窗体中打印数据	(144)
第八章 创建窗体	
8.1 窗体的设计视图	(145)
8.1.1 设计视图下的工具栏	(146)
8.1.2 工作区	(147)
8.1.3 工具箱	(147)
8.1.4 字段列表	(150)
8.1.5 属性窗口	(150)

8.2 创建窗体	(150)
8.2.1 向窗体内添加控件	(151)
8.2.2 更改控件尺寸与移动控件	(152)
8.2.3 设置属性	(153)
8.3 使用窗体向导	(155)
8.4 修饰窗体	(157)
8.4.1 控件位置	(157)
8.4.2 控件中的文本	(158)
8.4.3 控件的特殊效果	(159)
8.5 使用列表框与组合框	(162)
第九章 报表基础	
9.1 报表的作用	(165)
9.2 浏览报表	(165)
9.2.1 工具条	(165)
9.2.2 报表的滚动	(167)
9.2.3 打印预览页面	(167)
9.2.4 子报表	(167)
9.2.5 以不同的缩放比例预览报表	(168)
9.2.6 报表中的对象	(169)
9.3 打印报表	(169)
第十章 创建报表	
10.1 报表的设计视图	(171)
10.1.1 工具栏	(172)
10.1.2 工具箱	(173)
10.1.3 工作区	(173)
10.1.4 其他窗口	(173)
10.2 创建报表	(174)
10.2.1 向报表工作区内添加控件	(175)
10.2.2 更改控件尺寸及位置	(177)
10.2.3 设置属性	(177)
10.2.4 保 存	(177)
10.3 使用报表向导	(177)
10.4 报表类型	(180)
10.5 排序与分组	(184)
第十一章 宏	
11.1 宏的作用	(189)
11.2 宏中的工具条	(190)
11.3 建立宏	(190)
11.3.1 操 作	(191)

11.3.2	操作参数	(192)
11.3.3	向宏中添加操作	(192)
11.3.4	在宏中设置操作参数	(193)
11.3.5	编辑宏	(193)
11.4	运行宏	(194)
11.4.1	直接执行宏	(194)
11.4.2	从窗体或报表中运行	(194)
11.4.3	执行在另一个宏或 Visual Basic 程序中的宏	(196)
11.4.4	存储宏	(197)
11.4.5	检验宏	(197)
11.5	宏的高级技术	(198)
11.5.1	建立多操作宏	(198)
11.5.2	创建宏组	(199)
11.5.3	使用宏中的条件操作	(200)
第十二章 Visual Basic 基础		
12.1	用 Visual Basic 代替宏	(203)
12.1.1	什么时候使用宏	(203)
12.1.2	什么时候使用 Visual Basic	(203)
12.2	基础知识	(204)
12.2.1	事件	(204)
12.2.2	事件驱动的应用	(204)
12.2.3	模块	(205)
12.3	模块窗口新特点	(208)
第十三章 Visual Basic 编程		
13.1	Visual Basic 中的数据类型	(212)
13.1.1	使用数据类型	(212)
13.1.2	数据类型转换	(217)
13.1.3	数据类型对比	(219)
13.2	变量	(220)
13.2.1	变量的声明	(220)
13.2.2	变量的范围与可见性	(223)
13.3	范围和变量的名称	(227)
13.3.1	Visual Basic 的命名规则	(227)
13.3.2	命名冲突	(227)
13.3.3	命名冲突时的范围	(228)
13.3.4	变量的生存周期	(230)
13.4	对象变量	(231)
13.4.1	声明一个对象变量	(232)
13.4.2	赋值对象变量给对象	(232)

13.5 创建自定义数据类型	(233)
13.5.1 声明自定义数据类型	(233)
13.5.2 变量赋值	(234)
13.5.3 用户自定义类型中含有数组	(234)
13.5.4 自定义数据类型嵌套	(236)
13.6 使用常数	(236)
13.6.1 符号常量	(236)
13.6.2 固有常量	(238)
13.6.3 操作常量	(242)
13.6.4 系统定义常量	(244)
13.7 数 组	(244)
13.7.1 固定大小数组的声明	(244)
13.7.2 动态数组	(247)
第十四章 创建 Visual Basic 程序	
14.1 Visual Basic 编程环境	(249)
14.2 编写 Visual Basic 语句	(251)
14.2.1 声明语句	(251)
14.2.2 赋值语句	(251)
14.2.3 可执行语句	(252)
14.3 创建事件过程	(254)
14.4 窗体模块和报表模块	(257)
14.4.1 编写和修改代码	(258)
14.4.2 在过程之间切换	(260)
14.5 创建过程	(262)
14.5.1 Function 过程	(262)
14.5.2 Sub 过程	(266)
第十五章 调试 Visual Basic 程序代码	
15.1 避免错误	(270)
15.2 调试工具及其功能	(271)
15.3 启动 Visual Basic 程序代码	(272)
15.4 重新设置代码	(273)
15.5 暂停 Visual Basic 程序代码的执行	(273)
15.6 执行 Visual Basic 程序代码	(275)
15.6.1 跟 踪	(275)
15.6.2 单步执行 (跳过过程)	(275)
15.6.3 移 出	(275)
15.6.4 运行到光标	(276)
15.6.5 设置下一语句	(276)
15.7 查看变量值	(276)

15.7.1	在模块窗口中查看数据	(276)
15.7.2	“调试窗口”的新功能	(277)
15.7.3	使用即时窗格	(278)
15.7.4	在“调试窗口”中监视数据	(280)
15.7.5	在“调试窗口”中查看对象	(281)
15.7.6	使用监视表达式监视数据	(282)
15.7.7	设置控件、属性、变量	(283)
15.7.8	跟踪 Visual Basic 的过程调用	(284)
15.8	优化 Visual Basic 程序代码	(284)
15.9	错误和错误处理	(285)
15.9.1	出现错误时选择执行路径	(285)
15.9.2	退出过程	(286)
15.9.3	处理复杂过程中的错误	(287)
15.9.4	获得有关错误的信息	(288)
15.9.5	Err 对象	(288)
15.9.6	Error 对象和 Errors 集合	(289)
第十六章 形成整体数据库		
16.1	使用外部数据	(290)
16.1.1	链接外部数据	(290)
16.1.2	导入数据	(293)
16.1.3	导入与链接	(294)
16.1.4	导入与链接另一个 Access 数据库	(294)
16.2	创建一个启动窗体	(296)
16.3	工具栏、菜单栏及快捷菜单	(298)
16.3.1	利用宏创建工具栏、菜单栏及快捷菜单	(298)
16.3.2	为当前数据库创建自定义工具栏	(300)
16.3.3	为当前数据库创建自定义菜单栏	(301)
16.3.4	为当前数据库创建自定义快捷菜单	(301)
16.3.5	将按钮添加到“菜单栏”、“快捷菜单”工具栏或其他工具栏上	(302)
16.3.6	移动或复制工具栏按钮	(303)
16.3.7	将自定义工具栏、菜单栏、快捷菜单附加于窗体或报表	(304)
16.3.8	显示或隐藏工具栏、菜单栏或快捷菜单	(304)
16.4	设置数据库安全	(304)
16.4.1	设置密码	(304)
16.4.2	用户级安全	(305)
16.4.3	MDE 文件	(305)
16.4.4	设置启动选项	(307)

第一章

Access 97 快速入门

现代社会是信息社会，人们无时无刻不在与数据打交道，为了保存与处理这些数据，数据库是当然的选择。数据库的应用一直在个人计算机领域占有十分重要的地位。Access 是美国微软公司推出的数据库应用系统。它是基于 Microsoft Windows 系统，由于继承了 Windows 的特性，不仅易于使用，而且界面友好，在世界各地广泛流行。特别是于 1997 年推出的 Microsoft Access 97 提供了许多新增和改进的功能，可以帮助用户创建功能强大的数据库应用程序。可以预见，在不久的将来，Microsoft Access 97 会成为中小型数据库的主流系统。

本章对 Microsoft Access 97（简称为 Microsoft Access，如果有必要将当前版本的 Microsoft Access 与先前的版本区别开来，本书将会给出先前版本的版本号）作一简介。

1.1 安装 Access 97

1.1.1 运行 Microsoft Access 所需配置

运行 Access 97 计算机至少需要有以下配置：

- 系统必须装有 Microsoft Windows 95 或更高版本，或者 Microsoft NT 3.51 或更高版本的操作系统；
- 必须具有足够的硬盘空间。

若用户的系统不具备以上要求，在安装过程中，安装程序会给出错误提示信息。

1.1.2 安装 Microsoft Access 97

安装 Access 97 是十分简单的，用户只需按如下步骤进行。

(1) 将 Access 软盘或 Office 软盘的第一张盘插入软驱（如果购买的是 Access 或 Office 光盘版，请将光盘插入光驱中）。

(2) 请单击 Microsoft Windows 或 Microsoft NT 的“开始”按钮，然后单击“运行”命令，请输入 drive: setup.exe，再单击“确定”按钮。例如，如果在 A 驱动器中插入第一张盘，请输入 a: setup.exe。或在 Windows 与 NT 的资源管理器中，用鼠标单击 setup.exe，文件名加亮，然后在文件菜单中单击“打开”命令；也可以直接用鼠标双击 setup.exe。

(3) 如果是从网络驱动器中安装，请单击“浏览”按钮，然后使用“查找范围”框中查找“安装”文件。

(4) 在“安装”程序中，请单击“添加/删除”按钮。

(5) 在“选项”列表中，请单击要添加或删除的一个或多个组件。如果复选框中的“对号”暗淡显示，表示该组件只安装一部分，或者即将安装。如果要看到组件的全部元素，请

选择相应的选项，然后单击“更改选项”按钮。

(6) 根据安装程序提示即可完成安装。

(7) 重新启动计算机，安装结束。

若原来计算机中已经安装了老版本的 Access，则安装程序会询问是否保留原来的 Access，或是用新版本替代它。在一般情况下，安装时请不要用 Access 97 新版本替代以前的 Access 7.0 或更早的版本，除非用户愿意将已有的数据库应用进行转换，并且将来在 Access 97 下使用。用户最好先保留原来的数据库不进行转换，直到已熟练掌握 Access 97 为止。有关数据库转换的内容在第二章的“将数据库升级至 Access 97”中将作详细介绍。

1.2 Microsoft Access 97 的新特点

Microsoft Access 97 与 Microsoft Access for Windows 95，即 Microsoft Access 7.0 相比，又增加了一些新的功能，其中包括对 Microsoft Access 7.0 中已有功能的升级。

1.2.1 Visual Basic for Application——Microsoft Access 97 的编程语言

在以前的 Microsoft Access 2.0 及更早的版本中，Microsoft Access 是以 Microsoft Access Basic 为编程语言的；在 Microsoft Access for Windows 95 中，虽然也采用了 Visual Basic 作为编程语言，但是保留了原有编程语言的一些特点；而在 Microsoft Access 97 中，完全用 Visual Basic for Application 替代了 Microsoft Access Basic，成为 Microsoft Access 新的编程语言。需要说明的是，在 Office 97 中，包括 Word、Excel、PowerPoint，均是以 Visual Basic for Application 为编程语言的。

在 Microsoft Access 97 中，由于 Visual Basic 只是调用应用所需要的代码，比以前提高了速度，而且 Microsoft Access 97 中包括了新的 32 位的 Microsoft Jet 数据库工程，因而能使用户的应用快速调用及运行。

在 Microsoft Access 97 中，可以用类模块创建自定义对象，并对它定义自身的方法 (method) 和属性 (property)。

1.2.2 新的开发环境

在 Microsoft Access 97 中，由于引入了 Visual Basic for Application，引入与改进了如下开发工具。

一、改进了“模块窗口”

Microsoft Access 97 中的模块窗口比以前更易使用。由于使用了彩色代码标记，使代码更易阅读，而且自动创建清单功能和数据提示功能使输入代码更便捷、准确。

二、“调试窗口”的新功能

Microsoft Access 97 的“调试窗口”由三个窗格组成：功能很像 Microsoft Access 旧版本中的 Immediate 窗口的“即时调试”窗格、显示当前堆栈框架中全部变量和它们的值的“局部”窗格和“监视”窗格。单击“调试窗口”中的“局部”标签或“监视”标签，可以在“局部”窗格和“监视”窗格之间进行切换。而无论单击哪一个标签，“即时调试”窗格都保持可见，如图 1.1 所示。

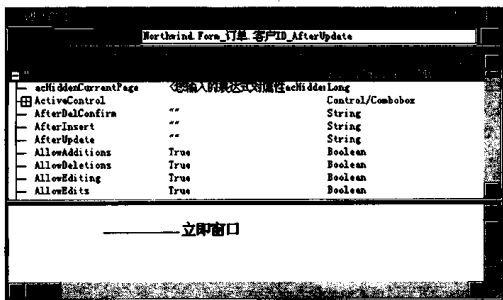


图 1.1

要查看“调试窗口”，可以打开一个模块，然后单击工具栏上的“调试窗口”按钮；也可以通过单击“视图”菜单上的“调试窗口”命令或在任何时候按 Ctrl + G，来打开“调试窗口”。

三、程序工具栏和菜单栏

Microsoft Access 97 中包含命令栏，即可用于编程的工具栏和菜单栏。通过命令栏可以为应用程序创建自定义工具栏和菜单。

只有设置对 Microsoft Office 8.0 对象库的引用后，才能使用命令栏来编写程序。要设置对 Microsoft Office 8.0 对象库的引用，可在模块“设计”视图中，单击“工具”菜单上的“引用”命令，并选中“Microsoft Office 8.0 Object Library”复选框，如图 1.2 所示。

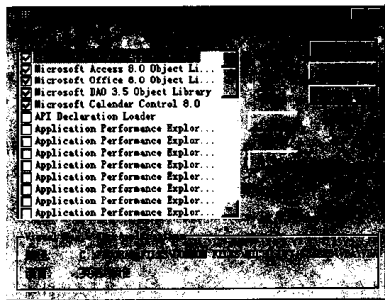


图 1.2

四、创建部分复制

部分复制是包含完整副本记录子集的副本。要创建部分复制，需要指定一个筛选，用于限制与整个数据库的子集同步的数据。使用部分复制，可以仅针对需要的数据来同步复制，而不必使用整个数据库。例如，某个企业可能在总部办公室保存了它的整个业务人员数据库，但是仅将地区数据复制到全国各地的办公室。对于每一个地区的办公室，只需创建仅包含该地区数据的独立副本，而在总部办公室的数据库则是完整的副本，以便使每个部分复制与之同步。

1. 使用部分复制的优点

部分复制能有效地缩小数据库，因为只需复制经常处理的部分即可。使部分复制同步只需花费较少的时间，并且副本本身需要的磁盘空间也较少。

部分复制能帮助限制对数据的访问。在业务人员数据库中，部分复制能帮助确保各地区办公室的业务人员不能查看其他地区办公室的业务人员数据。不过，虽然可以使用部分复制来限制记录访问，但它们仍不能替代安全系统。

部分复制还具有能够通过局域网（LANs）和广域网（WANs）复制数据的优点。通过限制需要复制的数据，可以减少在 LAN 或长距离载波中传送的数据量以减轻网络负担，并降低电子通信成本。

2. 使用数据访问对象（DAO）创建部分复制

要创建部分复制，可以按照下列步骤进行操作：

(1) 使用 Database 对象的 MakeReplica 方法，并指定 options 参数中的 dbRepMakePartial 常量；

(2) 使用 ReplicaFilter 和 PartialReplica 属性来设置所需的筛选和关系，以决定从完整副本中复制何种数据；

(3) 使用 PopulatePartial 方法从完整副本中传送与新的副本筛选准则相匹配的所有记录。

当更改部分复制中的数据或筛选准则时，应该同时使用 Synchronize 和 PopulatePartial 方法来确保所有数据反馈回完整副本，并且根据当前的筛选准则重新发布部分复制。

创建了部分复制之后，就不能再将其转换为完整副本。如果删除部分复制中的所有筛选和关系，它将包含完整副本的所有记录，但是仍然具有与部分复制相同的限制。

当部分复制中具有实施引用完整性的关系时，数据库中可能有许多表与筛选的基表相关。因此在创建部分复制时，必须确保包括所有与包含所需数据的表相关的表。例如，在使用 Northwind 示范数据库时，可以在“雇员”表上放置类似 [EmployeeID] = 1 这样的筛选，来仅仅同步一个员工的记录。因为“雇员”表和“订单”及“订单明细”表有关，因此也必须同时包含这些表，这样在部分复制中就会包含该员工在“订单”及“订单明细”表中的对应记录。

在某些情况下，也可能只希望包含与所用筛选间接相关的表。例如，可能想要在部分复制中包含所有产品的列表，这样新订单就能包含任何可得到的产品，这时就必须在部分复制中包含“产品”表。

对于数据库中不存在关系的表（例如查询表）或其关系不实施引用完整性的表，例如一个用于产生组合框的表，可能不参与任何关系。如果选择了在部分复制中包含此种表，那么所有记录将被捕获；否则没有记录会被捕获。

五、对象浏览器

“对象浏览器”是用于查看 ActiveX 组件这一类型库的工具。要查看“对象浏览器”，可打开一个模块，并单击工具栏上的“对象浏览器”按钮，或“视图”菜单中的“对象浏览器”命令。在模块“设计”视图中，还可以按 F2 来打开“对象浏览器”，如图 1.3 所示。

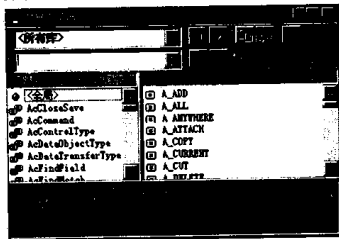


图 1.3

类型库中的信息包括对象及其属性、方法、事件、常量以及有关的函数和语句。这些信息组织在类模块和标准模块中。类模块和标准模块都列在“类”框中，而有关的方法、属性、事件及常量都列在“成员”框中。

Microsoft Access 97 的“对象浏览器”包含下列新功能。

- 特殊的<全局>项目是“类”框中的第一个输入项。这是“对象浏览器”中所有库的最上层输入项。当在“类”框中选择<全局>时，“成员”框中将显示所有能够全局访问的成员，包括所有常量。
- 在“搜索文本”框中键入搜索术语，然后单击“对象浏览器”工具栏上的“搜索”按钮，可以在引用类型库中搜索任何类或成员。
- 在“对象浏览器”中，选择某个方法或属性，单击“对象浏览器”工具栏上的“复制到剪贴板”按钮，然后切换到“模块”窗口，单击 Visual Basic 工具栏上的“粘贴”按钮，可以将程序代码粘贴到模块中。
- 在“成员”框中选择任何对象、方法、属性或事件，然后单击“对象浏览器”工具栏上的“帮助”按钮，或按 F1 键，可获得有关帮助信息。
- 可以查看用户自定义过程的定义。从“工程库”列表中选择包含过程的项目名称，然后在“类”列表中选择包含过程的标准模块或类模块名称，在“成员”列表中选择过程名称，然后单击工具栏上的“查看定义”按钮，Microsoft Access 就会打开包含所选过程的模块并在过程中放置插入点。
- 在工具栏上单击“向后”按钮，可以返回前面查看过的元素；在单击“向后”按钮以后，再单击“向前”按钮又可以回到开始的位置。
- 现在，某些固有常量按照使用它们的方法，也被集成在“对象浏览器”中。例如，单击“工程库”框中的 Microsoft Access 按钮，然后在“类”列表中滚动所有的对象类，将看到可用在 Microsoft Access 中的所有常量组。单击“AcCommand”可以查看能够与

RunCommand 方法一起使用的所有常量。

六、标签控件

Microsoft Access 97 包含可以用来在一个控件中将控件集合组合在一起的新的标签控件。例如，可以创建一个标签对话框，将不同类型的选项组织在不同的标签上。

标签控件的每个页对应于标签控件的 Pages 集合中的一个 Page 对象。每个 Page 对象都有它自己的 Controls 集合，该集合包含这一页上的所有 Control 对象。

要向标签控件添加新页，可使用 Pages 集合的 Add 方法；而要从标签控件删除一页，可使用 Remove 方法。

标签控件的 Value 属性指示当前被选择的页。它返回一个整型值，指出被选择的页在 Pages 集合中的位置。例如，如果选择了标签控件的第一页，Value 属性返回 0，即 Pages 集合中第一页的索引编号；如果选择了第二页，Value 属性返回 1 等等。在集合中页的位置对应于该页的 PageIndex 属性值。

七、编译性能的改善

Microsoft Access 改善了模块加载和编译的性能，使程序代码能够以更快的速度编译并执行。

1.2.3 Internet 特性

Microsoft Access 97 提供了广泛的新增功能，可以使用户更方便地使用 Internet 和开发全球广域网应用程序。如果要使用这些新功能来访问 Internet，需要具备 Web 浏览器（如 Microsoft Internet Explorer）以及调制解调器、intranet 连接或其他网络连接。

一、导入、导出、链接

1. 导入或链接 HTML 文件

通过“文件”菜单上的“获取外部数据”命令，可以从 HTML 文件导入或链接（只读）表或列表。“导入 HTML 向导”和“链接 HTML 向导”可以将 HTML 链接转换为 Hyperlink 数据类型的字段。

2. 将对象导出为 HTML 格式

通过“文件”菜单上的“另存为/导出”命令，可以将报表导出到静态的 HTML 格式，或将数据表和窗体导出到静态或动态的 HTML 格式。使用 HTML 模板文件，可以增强 Web 主页的外观、一致性及浏览性。

3. Web 发布向导

使用“文件”菜单上的“另存为 HTML”命令，可以通过“Web 发布向导”来创建 Web 应用程序。可以使用一个或多个 HTML 模板文件，将数据表、窗体或报表输出到静态或动态的 HTML 格式；创建主页；将所有文件作为 Web 发布程序存储到指定的文件夹；使用“Web 发布向导”将文件复制到 Web 服务器；以及保存有关 Web 发布程序的说明，以便以后使用。

4. 访问 FTP 和 HTTP 服务器

导入或链接（只读）FTP 或 HTTP 服务器上的数据，将数据库对象导出到 FTP 服务器，添加或修改“打开”对话框“查询”框中的 FTP 服务器位置列表。

二、超级链接

Microsoft Access 97 中包含超级链接，帮助用户方便地将应用程序连接到 Internet 或 intranet 上。超级链接可以跳转到 Internet 或 intranet 上的某处，或跳转到当前数据库或其他数据库中的对象以及本地计算机或联网的其他计算机上的文档中。

1. 超级链接数据类型

Microsoft Access 提供了一种新的 Hyperlink 数据类型，它可以包含超级链接地址。使用这种数据类型来定义表字段，可以将超级链接保存在表中。使用“插入超级链接”按钮，可以在数据表或窗体中添加超级链接地址。超级链接可以是硬盘驱动器上的文件路径、UNC 路径或 URL。当单击超级链接字段时，Microsoft Access 将跳转到相应的对象、文档、Web 页或其他目标。

2. 超级链接对象

从数据表、窗体或报表中的超级链接，或者其他 Office 文档，如 Word 文档或 Excel 工作表，跳转到 Microsoft Access 数据库或对象。从数据表、窗体或报表中的超级链接跳转到 Office 文档。

3. 窗体和报表上的超级链接

超级链接也可以包含在窗体或报表中。命令按钮、标签和图像等三种类型的控件可以包含超级链接。这些控件都有 Hyperlink 属性，以返回代表控件中的超级链接的 Hyperlink 对象；这些控件还有 HyperlinkAddress 和 HyperlinkSubAddress 属性，以设置或返回超级链接的地址和子地址。

单击包含超级链接的控件，将触发控件的 Click 事件。当打开超级链接时，由 Hyperlink - Address 和 HyperlinkSubAddress 属性指定的文档或对象就会打开。例如，如果 Hyperlink - Address 属性引用 Microsoft Access 数据库中的某个窗体，该窗体就会打开；如果引用网络上另一台计算机上的 Word 文档，那么该文档会在 Word 中打开；如果引用全球广域网上的一个地址，那么会打开 Web 浏览器（例如 Microsoft Internet Explorer）并显示 Web 页。

三、ActiveX 控制

Microsoft “Web 浏览器”控件是一个 ActiveX 控件，通过它可以从 Microsoft Access 窗体中查看 Internet 或 intranet 上的 Web 页或其他文档。“Web 浏览器”控件由 Office 97 中的 Microsoft Internet Explorer 3.0 提供。也可以免费从公司的 Web 节点 (<http://www.microsoft.com/china>) 下载 Microsoft Internet Explorer，并从 <http://www.microsoft.com/intdev/sdk/docs/iexplore/> 获得关于“Web 浏览器”控件的文档。

当安装 Internet Explorer 时，“Web 浏览器”控件将随着操作系统自动地登录，所以不用经过登录就可以从 Microsoft Access 使用它。如果要将“Web 浏览器”控件添加到窗体上，可以单击“插入”菜单中的“ActiveX 控件”命令，然后单击 ActiveX 控件列表中的“Web Browser Control”。

将“Web 浏览器”控件添加到窗体之后，就可以使用控件的 Navigate 方法在“Web 浏览器”窗口中打开一个 Web 页。例如，假设已经向窗体中添加了一个名为 ActiveXCtl0 的“Web 浏览器”控件，就可以为窗体创建一个如下所示的 Load 事件过程。

```
Private Sub Form_Load
```

```
Me! ActiveXCtl0.Navigate "http://www.microsoft.com/china"
```

End Sub

四、Office 工具

1. Web 工具栏

使用类似于 Microsoft Internet 资源管理器工具栏中的功能可以方便地浏览 Microsoft Access 数据库、对象或 Office 文档。

2. Web 快速查找

通过数据库和文件属性索引查找 Microsoft Access 数据库和其他 Office 文档。

五、其他功能

在使用 Synchronize 方法时,通过设置 dbRepSyncInternet 标志,可以使数据库副本与 Internet 服务器上的副本或设计主副本同步。

1.3 Access 97 的系统组成

Access 97 体现了“面向对象”的思想,将存储数据与操作数据的各种对象组合在一起,成为一套完整的数据库应用系统。Access 97 与以前的 Access 一样,提供了六种主要的对象:表 (Table)、窗体 (Form)、查询 (Query)、报表 (Report)、模块 (Module)、宏 (Macro)。下面对这些对象作简要的介绍。

● 表 (Table)

表是 Access 97 中最重要的对象,它是用户定义用于存储数据的对象,是一种特定实体的有关数据的集合,例如产品或供应商。对每种实体分别使用不同的表,意味着每种数据只需存储一次,这将提高数据库的效率,并且减少数据的冗余,充分利用存储空间。表是以行(称为记录)列(称为字段)格式组织数据的。

表中包括了不同种类数据的字段,例如,订单的日期、客户姓名。表中每一行称为记录。记录包含了实体的信息。必须对每个表定义主关键字。主关键字是指对每个记录有唯一值的一个或多个字段。还可以对表中的某些字段建立索引以加快搜索速度。

● 查询 (Query)

查询是用来操作数据的对象,它的主要目的是将一个或多个表中需要操作的字段集中起来,形成全局性的数据集合。由于考虑到存储时的冗余,一般在设计表时将实体数据分解为多个表。查询实际上就是将这些分散的数据集中起来。用户可以为每个表建立查询,也可以将多个表中的字段集中建立查询。用户既可以利用实例工具提供的表查询,也可以用语句建立查询。可以这样说,只要需要,用户均应该建立专门的查询。

● 窗体 (Form)

窗体主要用于控制应用程序的执行或数据的输出或显示。可以说窗体是数据库与用户进行交互的最好界面,利用窗体,能从表中查询提取所需的数据,并将数据显示。

● 报表 (Report)

如果要以打印格式来显示数据,使用报表是很有效的方法。报表是为格式化、计算、打印、总结数据而设计的对象。用户可以将表内的数据有选择的集中起来,建立报表。在报表中,可以控制每个对象的大小和显示方式,并可以按照所需的方式来显示相应的内容;报表还能进行计算,如求和,求平均;此外还有在打印报表前进行预览的功能。

● 宏 (Macro)

宏是指一个或多个操作的集合, 其中每个操作实现特定的功能, 例如打开某个窗体或打印某个报表。宏可以使某些普通的任务自动完成。例如, 可设置某个宏, 在用户单击某个命令按钮时运行该宏, 以打印某个报表。宏就是把一些动作集合起来, 让它们在某些情况下执行。宏可以简化一些“经常性”的操作。

● 模块 (Module)

模块是将 Visual Basic for Applications 声明和过程作为一个单元进行保存的集合。模块有两个基本类型: 类模块和标准模块。模块中的每一个过程都可以是一个函数过程或一个子程序。模块可以与报表、窗体事件相连, 建立完整的应用。

总的来说, 表是用来存储数据的。这些数据, 可以用查询来提取, 并且在报表中显示, 或者在窗体中显示和修改。窗体和报表可以直接使用来自目录或查询过滤的数据视图中的数据, 在模块中能使用 Visual Basic 提供的函数来进行数据计算。Microsoft Access 有许多内部功能来总结和格式化查询中的数据。

窗体和报表上的事件能够触发宏和用 Visual Basic 编写的程序。从宏和模块中, 能够改变应用软件流程; 能够打开、过滤以及改变窗体和报表中的数据; 能够运行查询和建立新对象。在 Visual Basic 中, 可以建立、修饰和删除任何 Microsoft Access 对象, 一行一行、一列一列地操作数据库中的数据以及处理意外情况。

1.4 浏览 Access 97

在 Office 97 中, 所有部件, 包括 Access、Word、Excel、PowerPoint、Outlook 均具有十分友好的界面。下面就以数据库 Northwind 为例浏览一下 Access 97, 体会一下 Access 97 的精彩之处。

1.4.1 运行 Microsoft Access

要使用 Microsoft Access, 首先必须进入 Microsoft Access 系统。以 Windows 95 为例, 在 Windows 95 中用鼠标单击“开始”按钮, 移动鼠标指到程序项, 停留一会儿, 屏幕上会显示出已安装的所有应用程序, 如图 1.4 所示。移动鼠标指到 Microsoft Access, 单击, 进入 Access 应用系统。

当用户运行 Microsoft Access 时, 会看到平面上显示的版权信息。在此之后, 将会出现如图 1.5 所示的 Microsoft Access 的界面, 此窗口可分为上、下两个部分。上半部分是用于创建新数据库的, 有两种方法, 一种是创建空数据库; 另一种是使用数据库向导创建新数据库 (这部分将在第三章中作详细介绍)。下半部分用于打开已经存在的数据库, 窗口中显示了五个选项, 其中四个是已经存在带有详细路径的数据库名, 这些数据库是最近被用户打开过的, 用户可以选择所要打开的数据库。如果要打开其他数据库, 则选中更多的文件选项, 在选择好以后单击确定按钮, 系统进入下一个画面; 如果不想打开任何数据库, 则选中“取消”按钮。

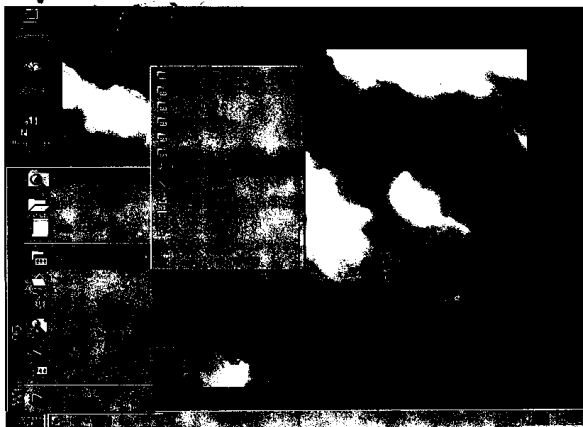


图 1.4



图 1.5



1.4.2 打开数据库

一、打开数据库

用户要打开已经存在的数据库，前面已经介绍过可以在刚刚进入 Microsoft Access 时进行。但是许多情况下，需要在进入系统后再打开数据库，这时用户可以按下列步骤进行。

(1) 用鼠标“打开”文件菜单，选择打开数据库命令，用鼠标单击，进入打开数据库对话框，如图 1.6 所示。

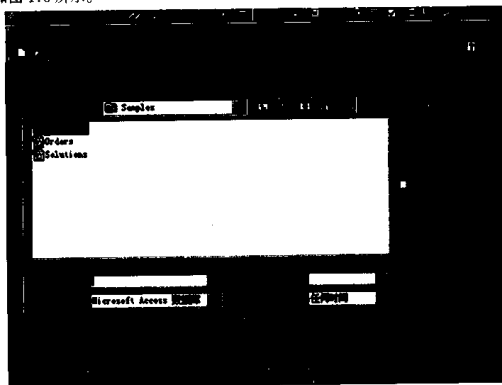


图 1.6

按 Alt + F 或 Ctrl + O 键也可打开数据库对话框，或单击工具栏上的“打开数据库”按钮。

(2) 在“查找范围”框中，单击包含所需文档的驱动器。设 Access 装在 D 盘中，则选择 D 盘驱动器。

(3) 在文件夹列表中双击相应的文件夹，直到打开包含所需数据库的文件夹。Northwind 数据库一般装在 Samples 子目录中。选择 Samples 子目录。

(4) 如果找不到要打开的数据库，可以对其进行搜索。只需在对话框的下部的各框中输入关于文件的特定信息，然后单击“开始查找”按钮即可。如果要使用其他高级搜索准则，请单击“高级”按钮。有关“高级查找”对话框的详细内容，将在后面介绍。

如果要在多用户环境下以共享方式打开数据库，请清除“独占”复选框。如果选择“独占”复选框，则可以独占方式打开数据库。

如果要以只读访问方式打开数据库，请单击“命令和设置”按钮，并单击“以只读方式打开”命令。

(5) 单击打开命令, 打开数据库。

有关对话框中各项目的详细内容, 请单击对话框右上方的问号标记, 然后单击要查询的项目。

二、高级查找

如果要使用其他高级搜索准则, 则需要进行高级查找。在打开数据库对话框中单击“高级”按钮, 屏幕上会显示“高级查找”对话框, 如图 1.7 所示。



图 1.7

“高级查找”对话框列出了用于定义要搜索的文件属性的准则语句。在默认情况下, Access 将搜索所有具有 mdb 扩展名的文件。可以在“高级查找”对话框中添加其他的准则语句来改善搜索条件。

- (1) 在“属性”框中单击要指定准则的属性。
 - (2) 在“条件”框中单击所需的条件。
 - (3) 在“值”框中输入数值, 完成准则语句。
 - (4) 回到左边的“属性”框, 执行下列操作之一:
 - 如果要求文件必须同时符合上述准则以及“按下列条件查找文件”框列表中的最后一行, 单击“与”;
 - 如果要求文件符合上述的准则或者列表中的最后一行, 单击“或”。
 - (5) 单击“添至列表”。
 - (6) 如果要限制搜索特定的文件夹, 可在“查找范围”框输入相应的路径。
 - (7) 如果要保存当前的搜索准则, 可单击“保存搜索条件”, 并输入搜索名称。如果要再次使用该搜索准则, 可单击“打开搜索条件”, 并单击所需的搜索名称。
 - (8) 单击“开始查找”, 开始搜索。
- 搜索完成后, 自动返回打开数据库对话框, 单击“打开”按钮完成打开数据库操作。

1.4.3 Access 97 的界面

根据上一部分的步骤打开 Samples 子目录中的 Northwind 数据库后，屏幕上出现的是“Northwind: 数据库”窗口，如图 1.8 所示。

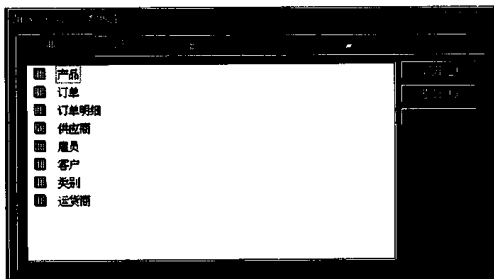


图 1.8

由图 1.8 可以看出，数据库窗口的标题条中显示了已打开的数据库的名称，并且以分页的方式显示数据库中的各种对象类型所包括的对象列表，共有前面介绍的六类对象：表、查询、窗体、报表、宏和模块。选中某一类对象类型，在窗体中就会显示该对象的内容，图中显示的是表的内容。

从图 1.8 中可以发现，在数据库窗口的右边有三个命令按钮，用户可以通过这三个命令按钮以三种方式对对象列表中的对象进行操作。

- 选择“新建”按钮，可以创建当前对象类型的新对象。
- 单击窗口中的对象，对象名加亮，选择“打开”按钮就能查看与修改数据。也可以直接在窗口内双击对象名打开相应的对象。
- 单击窗口中的对象，对象名加亮，选择“设计”按钮就可以修改此对象的设计。

在选择不同的对象类型时，用户会发现：报表、宏、模块与其他三种对象相比，没有“打开”按钮，而是用其他按钮代替：报表是“预览”按钮，宏是“运行”按钮，模块没有“打开”这个按钮。但是六种对象都有“新建”按钮与“设计”按钮。

对一些简单常用操作，Access 97 提供了快捷菜单，如图 1.9 所示。按鼠标的右键即可弹出快捷菜单。



图 1.9

在 Access 97 窗体上部是菜单与工具条。Access 97 的菜单与 Access 7.0 的不一样，在 Access 97 以前的菜单是固定的，与工具条有明显的区别；而在 Access 97 中，菜单同工具条

一样，是可以随意移动的，而且可以同时显示两个菜单，菜单与工具条几乎没有区别，只是菜单比工具条系统，是下拉式的，而工具条比菜单形象直观，如图 1.10 所示。

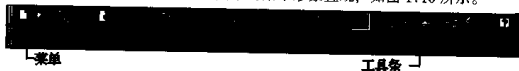


图 1.10

Access 97 带有十几个内部工具条。当 Access 的主窗体变化时，Access 97 会自动更改工具条，增加显示某些工具和工具条，同时又隐藏另外一些工具和工具条。通常情况下，Access 97 自动给出的工具条是对正在进行的工作最有价值的。用户能自主的控制显示和隐藏工具条，也可以在代码中控制它。

Access 97 具有“工具提示”功能。若用户要得到某个工具按钮的简短描述，只需将鼠标指针放在该工具按钮上，但不要单击它，过一段时间后，Access 会显示工具按钮的简短描述。

下面将分别介绍 Microsoft Access 中的各种类型的最简单的使用。

1.5 表的使用

当用户刚打开数据库窗口时，系统会自动地按下表按钮，窗口中显示的是这个数据库所有的表。

表是 Access 中最基本的对象，也是用户必须设计的第一个对象，下面以数据库中的产品表作为例子，说明表的使用。

1.5.1 数据表视图下的表窗口

用鼠标单击“产品”表名，再按“打开”按钮（或用鼠标双击“产品”表名），就打开了“产品：表”，如图 1.11 所示。

由图中可以看出，表由若干条记录组成，每个记录均由若干字段组成。表中的每一行为一个记录，每一列为一个字段。此产品表中共有 77 条记录、11 个字段。在表的底部，有记录选定器，如图 1.12 所示。

通过记录选定器，用户可以很方便地了解表中的数据量，进行数据的定位和浏览。表的操作有两种视图：“数据表”视图与“设计”视图。图 1.11 所示就是“数据表”视图。在“数据表”视图中，可以添加、编辑或查看表中的数据，也可以检查拼写及打印表中的数据，还可以筛选或排序记录、更改表的外观或通过添加或删除列来更改表的结构。

在表能存储数据以前，必须对表的结构进行设计，表的结构设计是在表的“设计”视图中完成的。在表“设计”视图中，可以创建整个表，或添加、删除或自定义一个已有的表的字段。

1	Chai	Exotic Liquids
2	Chang	Exotic Liquids
3	Anisseed Syrup	Exotic Liquids
4	Chef Anton's Cajun Seasoning	New Orleans Cajun Delights
5	Chef Anton's Gumbo Mix	New Orleans Cajun Delights
6	Grandma's Boysenberry Spread	Grandma Kelly's Homestead
7	Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead
8	Northwoods Cranberry Sauce	Grandma Kelly's Homestead
9	Nishi Koba Wika	Tokyo Traders
10	Dhura	Tokyo Traders
11	Queso Cabrales	Cooperativa de Quesos 'Las Cabras'
12	Queso Manchego La Pastora	Cooperativa de Quesos 'Las Cabras'
13	Konbu	Mayumi's
14	Tofu	Mayumi's
15	Genen Shoyu	Mayumi's
16	Bunches of Bunches	Bunches of Bunches

图 1.11

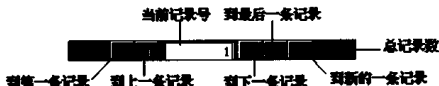


图 1.12

1.5.2 设计视图下的表窗口

用户可用工具条中的工具在两种视图间进行切换。单击视图即可实现由“数据表”视图向“设计”视图的切换，从而显示表的“设计”视图（“产品：表”），如图 1.13 所示。

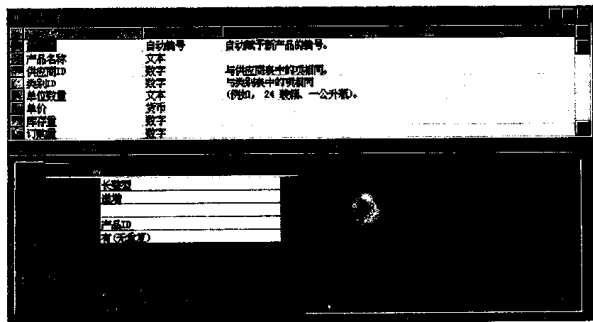


图 1.13

象，而查询则是基本的数据操作对象。查询可以替代表而成为窗体与报表的数据基础。一般情况下，“选择查询”是建立在表基础上的。当然“选择查询”也能建立在其他“选择查询”的基础之上。

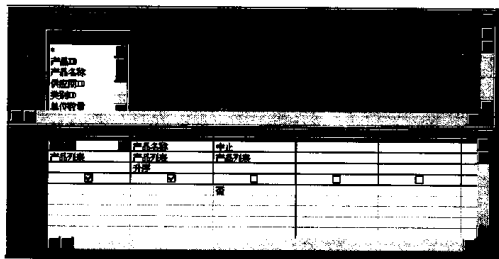


图 1.15

1.7 窗体的使用

由前面的叙述可以看出，表与查询能够提供许多有用的操作，但它们最大的缺点是使用上不方便，并且与用户交互困难、用户界面不友好。窗体就是用来完善与用户交互的，是 Access 中十分重要的部分。与数据库用户打交道的只有窗体，而窗体的美观与否，直接决定了数据库用户界面的友好性。在 Access 97 中，窗体与 Visual Basic 中的窗体在外表上可以说没有任何差别，在 Access 97 中也可嵌入图片、控制颜色与显示方式，在画面上放置按钮与列表框以及其他各种控件。

一个窗体有三种显示视图：设计视图、窗体视图和数据表视图。窗体比表和查询多了一种窗体视图，当然这三种视图也可由工具条中的按钮进行切换。

下面以 Northwind 数据库中的客户电话列表窗体为例介绍窗体。

1.7.1 窗体视图下的窗体

首先要打开“客户电话列表”窗体。

在 Northwind 数据库窗口中单击窗体页，窗体里会显示出所有窗体对象，选中“客户电话列表”，再单击“打开”按钮，或是双击窗体名称，Access 97 就以窗体视图的形式显示“客户电话列表”窗体，如图 1.16 所示。

在窗体视图方式下，可以观察窗体中的数据，但显示数据的形式与表和查询完全不同。它是以十分形象的方式显示这些数据的。可以将数据分为几类，进行分区显示，而且在窗体中含有子窗体，子窗体中的数据是与主窗体中的数据相互关联的，它的数据会随着主窗体中显示数据的变化而不同，例如 Orders 数据库中的定单窗体。窗体是提供友好用户界面的基



图 1.16

础，也是显示与组合数据的工具。

在窗体的底部，也有一个记录选定器，它的作用与表中的记录选定器相同，通过操作此按钮，用户可以浏览窗体中的数据。

1.7.2 设计视图下的窗体

在工具栏中单击视图按钮，窗体将转换至“设计”视图显示，如图 1.17 所示。用户可以在此视图方式下修改窗体的设计。

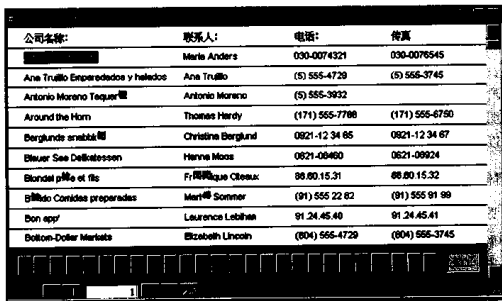


图 1.17

1.7.3 数据表视图

在工具栏中单击视图按钮旁边的下拉箭头，在下拉列表选中数据表视图，窗体将在“数据表”视图方式下显示，如图 1.18 所示。这种视图方式下的窗体与表的“数据表”视图十分相似。

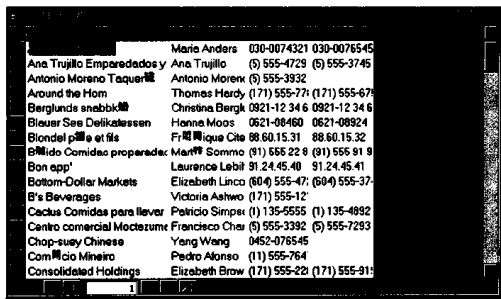


图 1.18

1.8 报表的使用

表、查询、窗体对象一般用于屏幕上的数据，虽然也可以用于打印，但只能做一些简单的工作。由于某些字段要自动求和计算，某些字段要分类打印，要想打印一份比较复杂的报表，这时就必须用到报表。

当要以打印格式来显示数据，使用报表是很有效的方法。因为可以在报表中控制每个对象的大小和显示方式，并可以按照所需的方式来显示相应的内容。报表也有“设计”视图，并且它的“设计”视图与窗体的“设计”视图几乎一样。如果要修改报表的设计，就必须进入设计视图。

报表的设计十分灵活，可以加入报表的页眉与页角，可以将记录进行分组和排序，并且可以进行数据的分层求和计算。

在数据库窗口中单击报表页，窗口内显示数据库内的报表列表。单击“客户”报表，然后按“打开”按钮或双击“客户”列表，进入报表的打印预览，如图 1.19 所示。

此时用户可以清楚地看到报表中的格式。如果要看报表的全页，只须将鼠标指针移到预览转换窗口，此时指针形状变成放大镜，单击，用户就可以看到报表的全页；再次单击，报表又回到缩小状态下。单击工具栏中的“关闭”按钮，回到数据库窗口中。单击工具栏中的“视图”按钮，可以将报表切换到“设计”视图方式下，如图 1.20 所示。在“设计”视图方式下，用户可以对报表进行修改和设计。一般情况下，报表均有报表页眉、报表页脚、页面页眉和页面页脚。

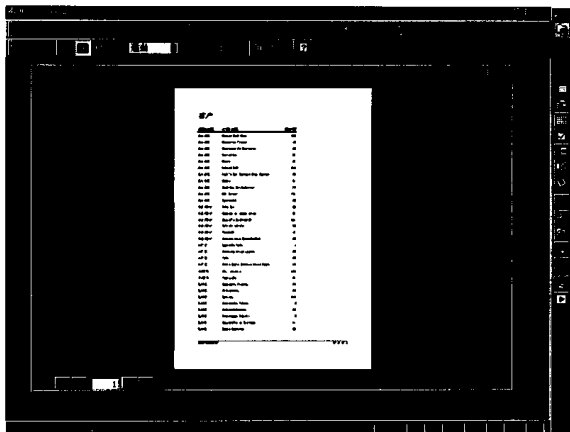


图 1.19

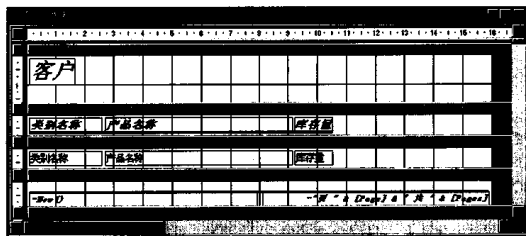
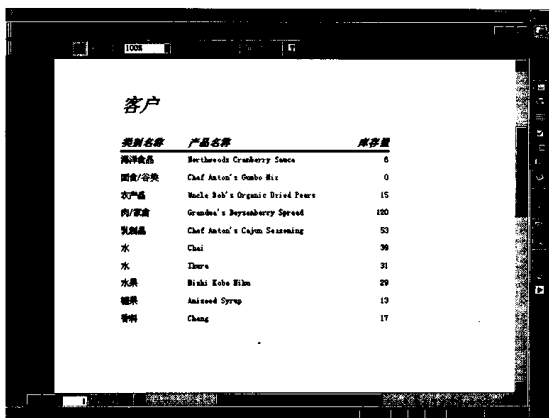


图 1.20

报表还有一种版式预览，使用“版式预览”可以查看报表的版面设置，其中只包括报表中数据的示例。图 1.21 所示，即为“客户”报表的版式预览。



类别名称	产品名称	库存量
西洋食品	Northwoods Cranberry Sauce	6
面食/谷类	Chef Anton's Gumbo Mix	0
农产品	Wackie Bob's Organic Dried Peas	15
肉/家禽	Grandma's Boysenberry Spread	100
乳制品	Chef Anton's Cajun Seasoning	53
水	Chai	39
水	Tea	31
水果	Bishi Koko Nibs	29
糖果	Amiseed Syrup	19
香料	Chang	17

图 1.21

1.9 宏的使用

宏是指一个或多个操作的集合，其中每个操作实现特定的功能，例如打开某个窗体或打印某个报表。宏可以使某些普通的任务自动完成，例如，可设置某个宏，在用户单击某个命令按钮时运行该宏，以打印某个报表。

宏可以是包含操作序列的一个宏，也可以是某个宏组，使用条件表达式可以决定在某些情况下运行宏时，某个操作是否进行，图 1.22 所示就是一个宏的“设计”视图。

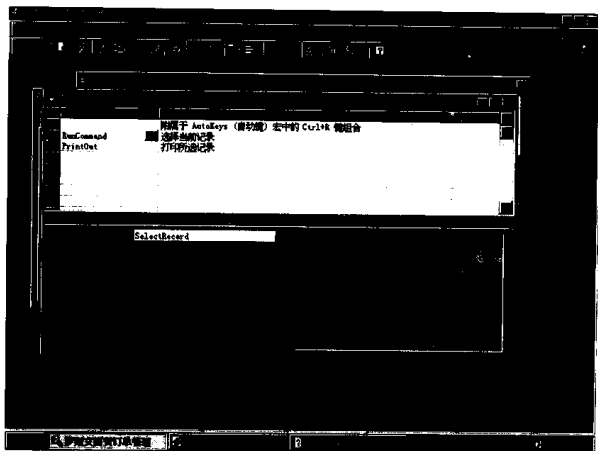


图 1.22

1.10 模块的使用

在数据库中，仅仅有表、查询、窗体、报表和宏是不够的，虽然它们能够完成大部分的工作，但是这些工作只是那些在 Microsoft Access 的界面中能够完成的，而对于那些超出 Microsoft Access 界面功能的就不能实现。这些功能就需要由模块来完成，当然是依靠在模块中的代码来完成。

模块是将 Visual Basic for Applications 声明和过程作为一个单元进行保存的集合。模块有两个基本类型：类模块和标准模块。模块中的每一个过程都可以是一个函数过程或一个子程序。

窗体和报表模块都是类模块，而且它们各自与某一窗体或报表相关联。窗体和报表模块通常都含有事件过程，该过程用于响应窗体或报表中的事件。可以使用事件过程来控制窗体或报表的行为以及它们对用户操作的响应，例如用鼠标单击某个命令按钮。

为窗体或报表创建第一个事件过程时，Access 将自动创建与之关联的窗体或报表模块。如果要查看窗体或报表的模块，只要单击窗体或报表“设计”视图中工具栏上的“代码”命令。图 1.23 所示就是窗体的“模块”窗口。

标准模块包含的是通用过程和常用过程。这些通用过程不与任何对象相关联，常用过程可以在数据库中的任何位置执行。单击“数据库”窗口中的“模块”标签，可以查看数据库中标准模块的列表过程，包含 Visual Basic for Applications 代码。图 1.24 所示就是订单数据库中的一个标准模块。

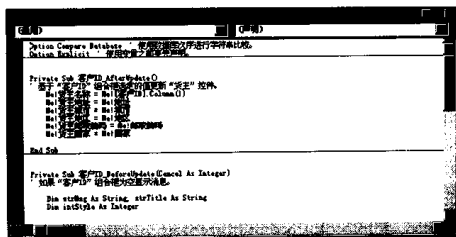


图 1.23

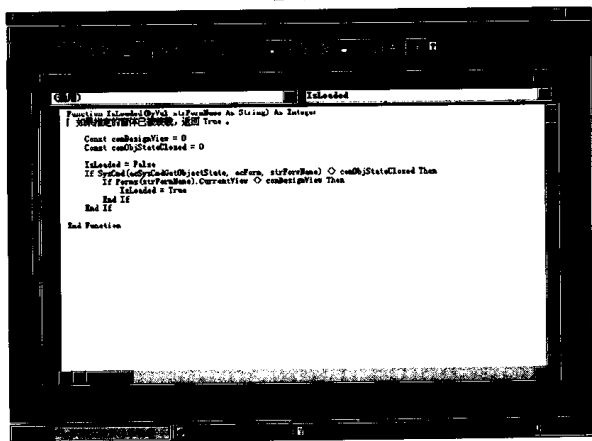


图 1.24

Access 中有两种过程。

- Sub 过程，执行一个操作或一系列的运算，但是不返回值。可以自己创建子程序或使用 Access 所创建的事件过程模板。

- 函数过程（通常简单地称为函数），将返回一个值，例如计算结果。Visual Basic 包含许多内置函数，例如 Now 函数可返回当前的日期与时间。除了这些内置函数外，也可以创建自己的自定义函数。

因为函数有返回值，所以可以在表达式中使用。可以在 Access 表达式中使用函数，例

如在 Visual Basic 语句或方法中，在一些属性设置中，或在筛选和查询的准则表达式中。

1.11 帮 助

在 Microsoft Access 97 中，用户可以通过 Office 助手获取帮助、提示及信息。Office 助手能回答用户的问题，提供提示，并可提供正在使用的 Office 程序的多种特定功能的帮助。按 F1 键就可以看到 Microsoft Access 97 中的 Office 助手，如图 1.25 所示。Office 助手还能显示下列内容：

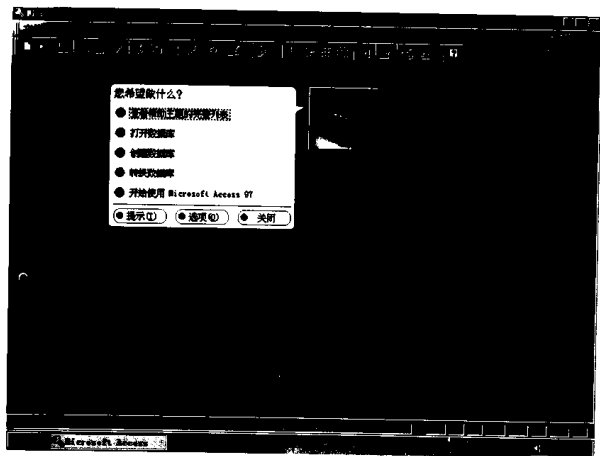


图 1.25

- 建议帮助，这与正在执行的特定任务有关，并且会在需要帮助之前显示出来。例如，如果选中了该选项，在对表进行更改时单击“助手”按钮，则返回关于如何使用表的帮助主题；

- 提示，指出如何在程序中更高效地使用其功能或键盘快捷方式；

- 信息，在打开 Office 助手时将显示有关信息，可以选择要在 Office 助手显示信息，或是在其他的对话框中显示。

在 Microsoft Access 97 中，用户也可以不使用 Office 助手来获取帮助。用户只须在“帮助”菜单中单击“目录和索引”命令，Access 会显示如图 1.26 所示的帮助主题窗口，用户可以直接键入所要查询的内容，Access 会自动显示内容，选中后，单击确定按钮，在屏幕中会显示帮助信息。

第二章

建立 Microsoft Access 中的数据库

本章将介绍如何建立 Microsoft Access 中的数据库，而表是数据库中用来存储数据的对象，它设计得好坏，将直接关系到数据库的性能，因此可以说表的设计在 Access 中是首要的。

2.1 建立新的数据库

数据库是用来执行数据管理，例如保存数据，对数据进行分析。Microsoft Access 数据库是用户管理数据库的所有相关对象的集合，包括表、查询、窗体、报表、宏、模块。每个对象都是数据库的一个组成部分，并且将这些对象组合成统一的系统。

要创建 Microsoft Access 数据库，可以有两种方法：一种是先创建一个空数据库，然后再添加表、窗体、报表及其他对象，这是最灵活的方法，但是必须分别定义每一个数据库组件；另一种方法是使用 Microsoft Access 97 提供的“数据库向导”，仅一次操作就可以为所选择的数据库类型创建所需的表、窗体和报表，这是创建数据库最简便的方法，但是它的应用范围比较小。无论哪一种方法，在数据库创建之后，都可以在任何时候修改或扩展数据库。

2.1.1 创建新的数据库

要在 Microsoft Access 97 中创建数据库，可以按以下步骤进行。

(1) 在第一章中已经写到，在启动 Microsoft Access 时，会自动显示对话框，上面有新建数据库或打开已有数据库的选项。如果此对话框显示，请单击“空数据库”，然后单击“确定”按钮。

如果已经打开了数据库或当 Microsoft Access 打开时显示的对话框已经关闭，则在“文件”菜单（如图 2.1 所示）中，用鼠标单击“新建数据库”命令，或单击工具条左端的“新建数据库”按钮。

(2) 此时，Microsoft Access 将弹出“新建”窗口，如图 2.2 所示。在此窗口中，用户可选择上述两种方法创建新数据库。若要用第一种方法，即自主创建数据库，则选中“常用”标签上的空白数据库图标，然后再按“确定”按钮，或双击空白数据库图标；若要用“数据库向导”创建数据库，则单击“数据库”标签，在窗口中会显示可用的数据库模板，如图 2.3 所示。在这些模板中选中要创建的数据库类型的图标，再单击“确定”按钮，或双击该图标。

(3) 此时，Microsoft Access 弹出“保存新数据库为”对话框，如图 2.4 所示。在文件名域中，输入新建数据库名。在 Microsoft Access 97 中，与以前的 Microsoft Access 不同，数

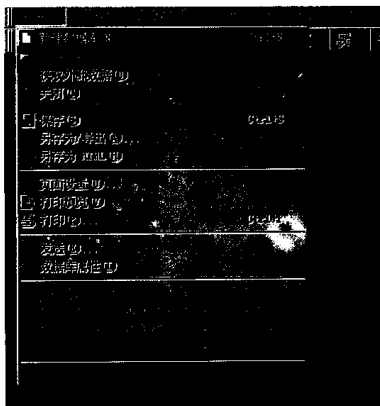


图 2.1

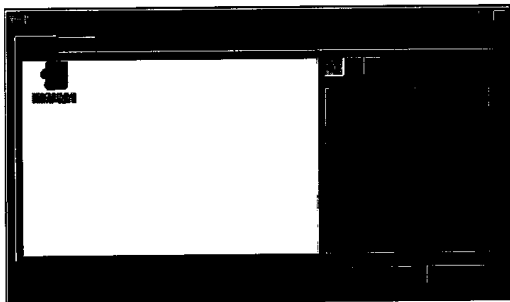


图 2.2

数据库名长度不受限制,而且可以用中文,这就十分便于中文用户存储数据库,使数据库的名称十分直观。在选择保存类型时,一般选择 Microsoft Access 数据库。在输入文件名时,可以输入扩展名,也可以不输入。当选择 Microsoft Access 数据库作为保存类型时,扩展名为 MDB (Microsoft Database)。当然,用户也可以用其他扩展名,但是使用 MDB 作为扩展名最方便,因为在打开数据库时,数据库类型缺省值是 Microsoft Access 数据库。

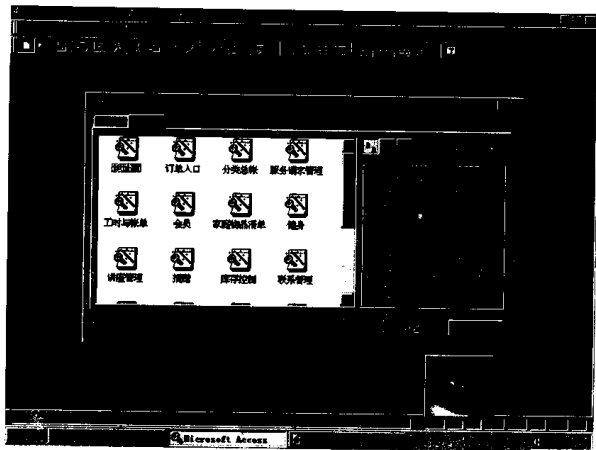


图 2.3

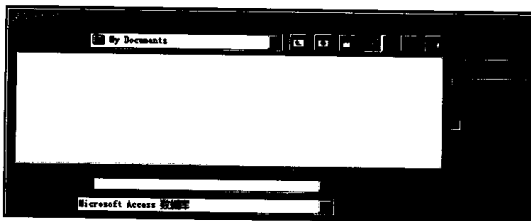


图 2.4

对于初学者来说，在对数据库进行命名时，也许会按照习惯将数据库命名为 db1、db2，实际上应避免使用这样的数据库名，而应该充分利用 Microsoft Access 97 提供的强大的命名功能，使数据库名称更加形象，方便用户日后管理与使用。

在命名时，要是新数据库名与该数据库所在目录中的数据库有相同的名称，那么旧数据库就将被覆盖。当然，在执行覆盖操作之前，Microsoft Access 97 会给出提示信息。因此，在对数据库进行命名时，除非用户要覆盖原有的数据库，应注意不要与原来已有的数据库重名。

在保存位置域中,选择该数据库存放的路径。

(4) 在完成这些选择后,用鼠标单击“创建”按钮,完成数据库的创建。

2.1.2 用“数据库向导”创建新数据库

利用“数据库向导”创建数据库时,开始步骤与自主创建数据库相同,只是在单击“创建”按钮以后,用“数据库向导”创建新数据库会在单击“数据库”窗体之后,立刻弹出如图 2.5 的窗体。

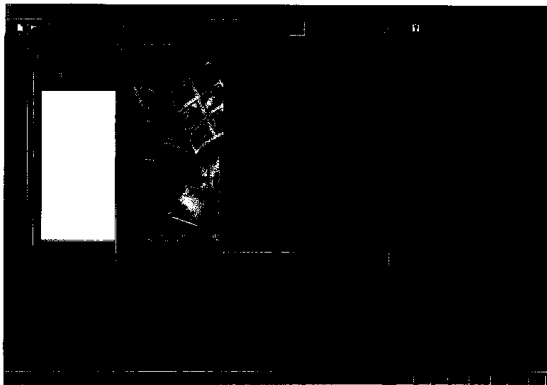


图 2.5

在图 2.5 所示的画面中,单击“下一步”按钮继续数据库的创建。用户可以根据窗体显示的信息完成创建。

在利用“数据库向导”创建数据库时,主要是要选择用户所需的表及表中的字段。以地址为例,在得到图 2.5 后,单击“下一步”,其显示如图 2.6。从图中可以看出,用户可以从“数据库中的表”列表框中选择表,在“表中的字段”列表框中会显示相应表的字段,字段包括必选字段与附加的可选字段,用户可根据需要选择相应的附加可选字段。选择完毕,再单击“下一步”,选择背景样式。

如果用户对上一步选择不满意,可单击“上一步”回到原来状态,重新选择。

最后单击“完成”按钮,此时“下一步”按钮已被隐藏。

在所有这些过程中,如果用户想取消创建数据库操作,则可以按“取消”按钮;如果想立刻完成用“数据库向导”创建数据库的操作,则可按“完成”按钮,当然后面的选项就以缺省的操作为标准。

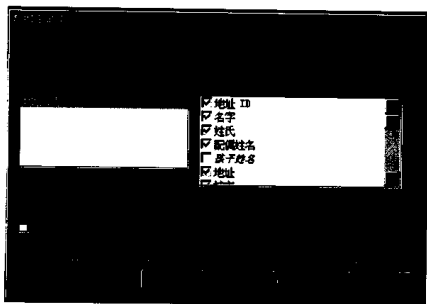


图 2.6

2.2 数据库窗口与工具条

第一章已经介绍过，在创建数据库完成后，在屏幕左上方将会出现“数据库”窗口，而且窗口的标题是用户新创建的数据库的名称，如图 2.7 所示。

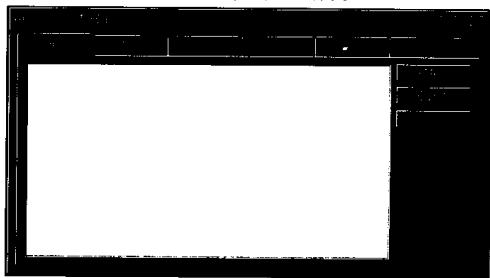


图 2.7

“数据库”窗口在 Microsoft Access 数据库的建立过程中，它的作用就像文件管理器在 Windows 中的作用，用户只有通过它来显示该数据库中的所有对象，而且在以后所需要创建和修改的各种表、窗体、报表、宏、模块，也都需要通过“数据库”窗口来操作。

当用户打开数据库时，Microsoft Access 就缺省地显示窗口内表标签，并且在窗体内显示该数据库中所有的表的名称。由于这是个自主创建的新数据库，在创建完以后，数据库内

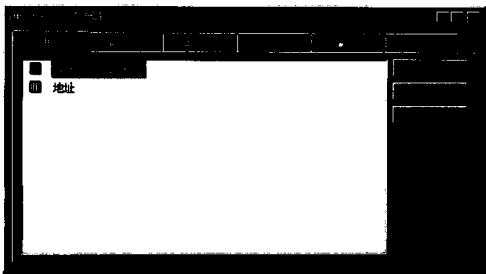


图 2.8

没有任何对象,因此窗体内是空白的。如果该数据库是用户利用“数据库向导”创建的,那么窗口内就会显示用户在创建过程中选中的表,如图 2.8 所示。图中的 Address 数据库就是刚才用“数据库向导”创建的,而且刚才就选中了两个表,因此在图中就显示了这两个表。

在 Microsoft Access 中,当当前窗口变化时,工具栏区的工具条就会发生变化。这个工具条是最适合于当前窗口使用的。由于此时的当前主窗口为数据库窗口,在 Microsoft Access 工具栏区中就显示了最适合于数据库窗口操作的工具,如图 2.9 所示。图中说明了这些工具的名称。



图 2.9

下面分别介绍这些工具按钮的功能。

- 新建数据库按钮——单击此按钮,建立新数据库。
- 打开数据库按钮——单击此按钮,打开数据库。如果用户已经打开了一个数据库,那么 Microsoft Access 会在打开另一个数据库之前关闭原有数据库。
- 保存按钮——单击此按钮,保存已作的修改。
- 打印按钮——选择任何表、查询、报表、窗体或模块,然后单击此按钮,打印用户的对象。它可以打印出表与查询的数据表视图,窗体的窗体视图、报表的报表视图以及代码。
- 打印预览按钮——选择一个表、查询、报表、窗体,单击此按钮,可以在打印预览窗口中看到所选对象在打印中的显示。

- **拼写按钮**——从表、查询或窗体中选定数据，单击此按钮，可对该对象进行拼写检查。
- **剪切按钮**——选中“数据库窗口”中的对象，单击此按钮，把这个对象从数据库中删除。
- **复制按钮**——在“数据库窗口”中选择一个对象，单击此按钮，把这个对象拷贝到剪贴板上。
- **粘贴按钮**——单击此按钮，将存储在剪贴板上的一个对象拼接到数据库中。Microsoft Access 会提示用户为此对象提供一个新名称。
- **格式刷按钮**——单击此按钮，可以从一个控件将格式复制到另一个控件。
- **撤消按钮**——单击此按钮，撤消前一步操作。
- **Office 连接**——将数据库与 Office 97 中其他套件相连，方便操作。有三种：用 Word 合并，用 Word 发布，用 Excel 分析。
- **分析按钮**——可用此按钮对数据库进行优化。
- **大图标、小图标、列表、详细信息按钮**——这四个按钮是用来确定“数据库窗口”内列表的显示方式的。
- **代码按钮**——选择一个数据库窗体或报表，单击此按钮，可以看到用户编写的 Visual Basic for Application 代码。
- **属性按钮**——选择一个对象，单击此按钮，Microsoft Access 就会给出此对象的属性信息。
- **关系按钮**——用此按钮建立表、查询之间的联系。
- **新对象按钮**——要在数据库中加入新对象时，单击此按钮。共有如下几种新对象：自动窗体、自动报表、表、查询、窗体、报表、宏、模块和类模块。
- **Office 助手按钮**——单击此按钮，可显示 Office 助手，它能回答用户的问题，提供提示。

2.3 创建表

用户只有通过“数据库窗口”，才可以创建新表。

在“数据库窗口”中单击表标签，Microsoft Access 会在窗口中显示已有的表的信息。再单击新建按钮，Microsoft Access 会弹出新建表窗口。在此窗口中，提供了五种创建表的方法：在“数据表”视图中新建表，在“设计”视图中新建表，利用“表向导”创建新表，从外部文件中导入表，从外部文件中链接表。“新建表”窗口如图 2.10 所示。

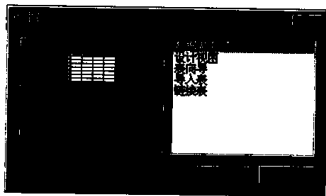


图 2.10

一般情况下,用户都是在设计视图方式下设计表,下面就以“设计”视图为例,详细说明表的创建过程。

首先在新建表窗口中选“设计”视图,再单击确定按钮,屏幕上将出现表在“设计”视图下的窗口,如图 2.11 所示。这就是表的设计画面。

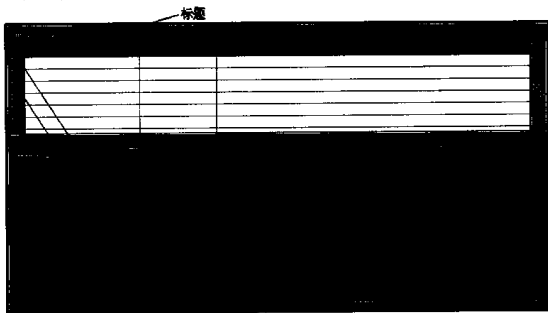


图 2.11

2.3.1 设计画面

由图 2.11 可以看出,整个设计画面分为以下几个部分。

在设计画面的最上方是表窗口的标题区,这里显示的是打开表的名称。若当前被打开的表是新建的,则在标题区显示的是缺省名。一般是表 1,如图 2.11 所示。

设计画面的上半部分是用户设计表中的字段用的表格。表格中每一行代表一个字段,表格的每一行均由四部分组成。在最左边有一灰色的小方块,它是“行选择区”,当用户移动光标到某一行时,对行选择区内会出现一个三角形符号——行指示器,用它来指明当前焦点的位置。每行都有三列,分别为字段名称、数据类型和说明。在字段名称列中输入要创建的字段的名称;在数据类型列中输入该字段的数据类型;在定义完名称与数据类型之后,最好在说明部分输入相应的字段说明文字,可包括字段的用途等,来增加可读性,当然,说明部分可写也可不写。

左下部是字段特性参数区。当定义了一个字段后,在此区域内,会显示对应字段的特性参数。这些参数将在后面作详细介绍。

右下角是一个信息框。在这个区域内,Microsoft Access 向用户显示有关字段或特性信息。当用户移动焦点时,框内的内容会自动改变,向用户提供相关的信息,实际起到一个帮助作用。

2.3.2 设计画面的工具条

当打开表的设计视图时,Microsoft Access 工具条会发生变化,在工具条中出现了用于

设计表的几个专门工具，此时的工具条如图 2.12 所示。



图 2.12

下面分别介绍这些工具按钮的功能。

- 视图按钮——单击此按钮，将显示设计视图与数据表视图两个选项。此按钮用于实现两种视图之间的转换。

- 保存、打印、打印预览、拼写、剪切、复制、粘贴、格式刷、撤消按钮的功能与数据库窗口相同，这里不作叙述。

- 主关键字按钮——选中一个或几个字段，再单击此按钮，就可将这些字段设为该表的主关键字。

- 索引按钮——单击此按钮，为此表打开索引窗口。

- 插入行按钮——单击此按钮，在当前行上方插入一个空行。为了使用该按钮，用户必须将光标放在设计画面上半部分的一行上。

- 删除行按钮——单击此按钮，删除当前行。要使用该按钮，也必须将光标放在设计画面上半部的一行上。

- 属性按钮——单击此按钮，Microsoft Access 会弹出表属性窗口。此窗口的作用将在后面叙述。

- 生成器按钮——单击此按钮，Microsoft Access 给出“字段生成器”，如图 2.13。在“字段生成器”窗口中，用户可根据需要选择示例表中的字段，将这些字段作为新表的字段。

- 数据库窗体——要回到“数据库”窗口状态，单击此按钮。



图 2.13

2.4 定义字段

表是由许多个字段组成的，因此在显示了创建表的设计画面以后，就该准备为 Northwind 数据库中的客户表设计定义字段。

首先要确定光标在字段名称的第一行，然后键入字段名称“客户 ID”。单击 Tab 键或用鼠标将光标移至数据类型列。此列是一个组合框，单击此列中右侧箭头，或按键盘的 Alt + ↓ 键打开数据类型选择列，如图 2.14 所示。在数据类型列中，既可以在框中键入一个有效类型，也可在下拉列表框的有效类型中选择。

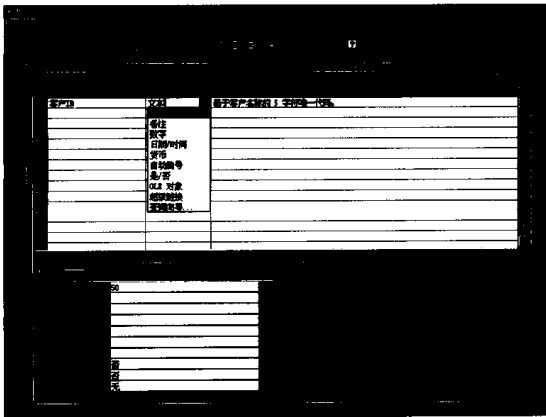


图 2.14

对字段命名时应注意，在 Microsoft Access 中允许使用汉字，最长为 64 个字。

在选择数据类型之后，在窗口的左下部字段特性参数区中，会显示一些特性参数。在特性参数框中，允许用户设置特性，并且能够定制字段。依照所选的数据类型，Microsoft Access 会显示不同的特性参数，如图 2.15 所示。对特性参数的不同设置，将在稍后详细介绍。

在每个字段的说明列中，用户可输入每个字段的描述性语句。当用户在数据表视图的查询中或在窗体视图或数据表视图的窗体中选择此字段时，Microsoft Access 在状态条显示出这个说明，状态条在 Microsoft Access 底部。

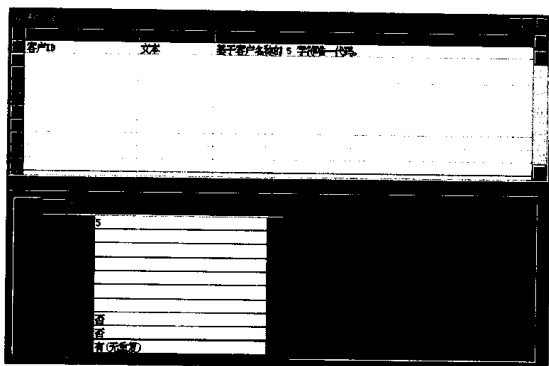


图 2.15

2.4.1 字段数据类型

与 Microsoft Access 以前的版本相比, Microsoft Access 97 中的数据类型有所增加, 共有十种, 它们分别是: 文本型 (Text)、备注型 (Memo)、数字型 (Number)、日期/时间型 (Date/Time)、货币型 (Currency)、自动编码型 (Auto Number)、是/否型 (Yes/No)、OLE 对象 (OLE Object)、超级链接 (Hyperlink) 和查阅向导 (Lookup Wizard)。每个数据类型都有一个特殊目的, 在表 2.1 中详细描述了这些数据类型。

表 2.1 字段数据类型

数据类型	使用对象	大 小
文本型 (Text)	文本或文本与数字的组合, 例如地址; 也可以是不需要计算的数字, 例如电话号码、零件编号或邮编	最长为 255 个字符
备注型 (Memo)	保存长度较长的文本及数字, 例如备注或说明	最长为 64 000 个字符
数字型 (Number)	可用来进行算术计算的数字数据, 计算中包含有关金钱的除外 (使用货币类型)。设置“字段大小”属性定义一个特定的数字类型	1、2、4 或 8 个字节, 16 个字节只提供同步复制标志号 (GUID)
日期/时间型 (Date/Time)	日期及时间	8 个字节
货币型 (Currency)	货币值。使用货币数据类型可以避免计算时四舍五入, 精确度为小数点左方 15 位数及右方 4 位数	8 个字节
自动编码型 (AutoNumber)	在添加记录时自动插入的唯一顺序 (每次递增 1) 或随机编号	4 个字节, 16 个字节只提供给同步复制标志 (GUID)
是/否型 (Yes/No)	这些字段只包含两种值中的一种, 例如 Yes/No、True/False、On/Off	1 位

续表 2.1

数据类型	使用对象	大 小
OLE 对象 (OLE Object)	在其他使用 OLE 协议程序创建的对象 (例如 Microsoft Word 文档、Microsoft EXCEL 电子表格、图像、声音或其他二进制数据), 可以将这些对象链接或嵌入 Microsoft Access 表中, 必须在窗体或报表中使用结合对象框来显示 OLE 对象	最大可为 1 GB (受磁盘空间限制)
超级链接 (Hyperlink)	保存超级链接的字段。超级链接可以是 m 某个 UNC 路径或 URL	最长为 64 000 个字符
查阅向导 (Lookup Wizard)	Wizard 创建字段, 该字段将允许使用组合框来选择另一个表或一个列表中的值。从数据类型列表中选择此选项, 将打开向导以进行定义	与主关键字字段的长度相同, 且该字段也是“查阅”字段, 通常为 4 个字节

有时对于字段而言, 可以选取两种甚至更多种数据类型, 但总有一种是比较合适的, 例如: 在一个处理学生成绩的数据库中, 对于学生的学号字段可以使用数字型中的长整型 (Long), 也可以使用数字型中的整型 (Integer), 还可以使用文本型 (Text), 在这种情况下, 就需要进行选择。在选择字段应该使用的数据类型时, 应考虑如下几方面。

- 在字段中允许什么类型的值, 例如不能在“数值”数据类型的字段中保存文本数据。
- 要用多少存储空间来保存字段中的值。
- 要对字段中的值执行什么类型的运算, 例如 Microsoft Access 能够将数字型 (Number) 或货币型 (Currency) 字段中的值求和, 但不能对文本型 (Text) 或 OLE 对象型 (OLE Object) 字段中的值进行求和操作。
- 是否需要排序或索引字段。备注型 (Memo)、超文本链接型 (Hyperlink) 及 OLE 对象型 (OLE Object) 字段不能作排序与索引。

● 是否需要在查询或报表中对记录进行分组。Memo、Hyperlink 以及 OLE Object 均不能用于分组记录。

● 如何对字段中的值进行排序。在 Text 字段中, 将数字以字符串形式进行排序, 例如 1、10、100、2、20、200 等, 不是作为数值来进行排序。如果将各种日期数据输入到 Text 字段中, 将不能正确排序, 只有使用 Date/Time 字段才能正确排序。

当选择数据类型时, 还需仔细考虑输入什么作为字段大小特性值, 因为这些选择将影响到精度与长度。要使用货币型来存储货币, 货币有整数精确度, 但也有固定的小数位。日期/时间型对于日历和时间数据是十分有用的, 并且在日期/时间的加减计算中会有更大的好处。自动编码型是一种特殊的整型字段, 主要是为表中关键字的产生而设计的。一个表中只能有一个字段的数据类型是自动编码型的。OLE 对象型允许用户存储复杂的数据, 例如图、声音、画面与 Microsoft Windows 操作系统的其他应用程序的动态连接支持这个类型。超级链接型的字段可以保存超级链接地址。查阅向导型字段可以使用列表框或组合框从另一个表或值列表中选择一值。

2.4.2 字段特性

在确定好数据类型后, 并不是已经确定好了字段, 要详细确定表中字段的存储, 还必须设定字段特性。这些特性根据用户所选择的数据类型的不同而不同。下面是一个表中的一字段的特性。

1. 字段大小

使用这个属性，用户可以为 Text 数据类型确定长度，还可以为 Number 数据类型确定数值范围和种类。Text 的长度可以从 1 到 255 个字符，缺省长度为 50 个字符。对 Number，字段长度如下：

- Byte 一个单字节整数，包括 0~255 的值
- Integer 一个两字节整数，包括 -32768 ~ +32768 的值
- Long Integer 一个四字节整数，包括 -2 147 483 648 ~ +2 147 483 647 的值
- Single 一个四字节浮点数，包括 $-3.4 \times 10^{38} \sim +3.4 \times 10^{38}$ 的值
- Double 一个八字节浮点数，包括 $-1.797 \times 10^{308} \sim +1.797 \times 10^{308}$ 的值

2. 格式

用户可以控制怎样显示和打印数据。格式选择根据数据类型的不同而不同。对于 Text 和 Number 数据类型，可以指明一制定格式来控制 Microsoft Access 怎样显示数据；对于 Number（数字）、Currency（货币）和 Counter（计数器）的格式选择如下：

- 一般数字 缺省（无逗号或货币符号，小数位根据数据精确度显示）
- 货币 货币符号和两位小数
- 固定的 至少一位数字和两位小数
- 标准 两个小数位和分隔逗号
- 百分率 百分数
- 科学计数 科学计数法（如 1.05×10^3 形式）
- 对于 Date/Time 数据类型，格式选择应依照以下模式：

一般日期（缺省）	05/15/97 05:30:10 PM (US) 15/05/97 17:30:10 (UK)
长日期	Wednesday, May 15, 1997 (US) 15 May 1997 (UK)
中等日期	15 - May - 97
短日期	5/15/97
长时间	5:30:10 PM
中等时间	5:30 PM
短时间	17:30

- 对于 Yes/No 数据类型，选择如下：

在缺省情况下为 Yes/No

其他还可以是：

True/False

On/Off

3. 小数位

对于 Number 和 Currency 数据类型，可以指定 Microsoft Access 显示的小数位数目。缺

省指定是 Auto。Auto 能使 Microsoft Access 显示 Currency、Fixed、Standard 和 Percent 格式中的两个小数位,并显示出 General Number 格式中的数字值当前精度显示所必需的小数位的个数。

4. 输入标记

对于 Text、Number、Currency 和 Date/Time 数据类型,可以指定一个编辑标记,当用户输入表中数据时能看到它。例如,可以使 Microsoft Access 提供一个在日期表中的界线(_/_/_/_);或可以使 Microsoft Access 提供一个 U.S 电话号码格式(# # #) 000 - 0000。

5. 标题

用户可以输入一个 Microsoft Access 显示在表格标号上或报表标题中的更具有描述性的字段名字。

如果用户建立了一字段名,它不含有空格,那么,用户可以用 Caption 特性指定一名字,它有空格,这样,Microsoft Access 便可以在查询、窗体、报表和报表所连接的标志和表头中,使用这个名字。

6. 缺省值

用户可以为除了 Counter、Memo 和 OLE Object 之外的所有数据类型指定一个缺省值。对于数字,缺省值是 0。Microsoft Access 为 Text 和 Memo 数据类型提供了 Null (空)缺省值。

7. 检验规则

每当用户在这个字段中输入和改变数据时,用户可以提供一个必须是真的表达式,例如, <100 指定一个数字必须小于 100。也可以核查一系列值之一,例如,用户可以核查一个有效城市列表,方法是指定“Chicago”或“New York”或“San Francisco”。另外,用户可以指定一个复杂的包括 Microsoft Access 任一内部功能的表达式。

8. 检验文本

每当输入的数据没有通过用户的检验规则时,用户可以让 Microsoft Access 显示文本本来给出提示信息。

9. 要求

如果用户不允许在字段中有 Null 值,设置该特性为“Yes”。

10. 允许零长度

对于 Text 和 Memo 字段,用户可以设置字段等于零长度字符串(“”)。

11. 索引

用户可以要求建立一个索引,以更快地存储 Text、Number、Date/Time、Currency 和 Counter 数据类型数据值。也可以要求索引字段中的字对整个表是唯一的。

2.4.3 完成表中字段的创建

了解以上叙述的内容之后,用户就可以在“客户:表”中加入相应的字段。表 2.2 中定义了这些字段,图 2.16 给出了完成字段定义后的“客户:表”设计视图。

表 2.2 定义字段

字段名称	数据类型	说 明	字段大小
客户 ID	文本	基于客户名称的 5 字符唯一代码	5
公司名称	文本		40
联系人姓名	文本		30
联系人头衔	文本		30
地址	文本	街道或邮政信箱	60
城市	文本		15
地区	文本	州或省	15
邮政编码	文本		10
国家	文本		15
电话	文本	电话号码包括国家代号或区号	24
传真	文本	电话号码包括国家代号或区号	24

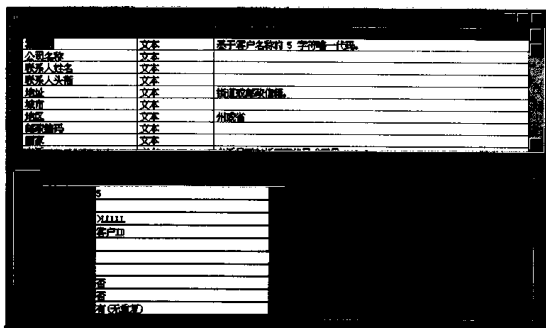


图 2.16

2.4.4 定义字段有效性规则

向字段有效性规则特性框中输入一个表达式可以定义一个字段中的值的简单核查。用户在该字段中输入值后，Microsoft Access 会自动检查这个值是否符合有效性规则。如果用户输入的字段值不符合该字段的有效性规则，Microsoft Access 将显示一条消息告诉用户数据无效。消息的来源有两种，一种是来自用户给字段有效性文本特性框输入的提示信息；另一种，若用户没有输入有效性文本，则 Microsoft Access 将显示标准错误信息。只有编辑数据及焦点离开字段时，字段的有效性规则才会起作用。定义了有效性规则，Microsoft Access 就不允许用户输入不符合该规则的非法数据，达到初步检查数据是否有效的效果。

一般情况下,一个字段的有效期规则表达式中包括一个运算符和一个比较值。运算符见表 2.3。如果用户未输入运算符,Microsoft Access 默认认为是等号比较。在表达式中除了比较操作符外,还可包含算述运算符与逻辑运算符。算术运算符包括加、减、乘、除、乘方、取模等,逻辑运算符包括与、非、或、异或、等于等。

表 2.3 比较运算符

操作符	意 义
<	小于
<=	小于或等于
>	大于
>=	大于或等于
=	等于
<>	不等于
IN	核查等于一个列表中的任何数;比较值必须是一个包括圆括号的列表
BETWEEN	核查值的一个范围,比较值必须是关键字 AND 分隔的两个值(一低一高)
LIKE	检查一个文本或备注字段是否匹配一个模式字符串

用户可以利用比较符号来把字段的值与一个或多个用户的有效性规则中的值进行比较,例如用户想核查数字的值是否总小于 100,则用户可在有效性规则中输入<100。用户也可用一对或多对比较来核查一个值是否在某个范围之内,例如要判断一个数是否在 10 与 20 之间,用户可以输入>10 AND <20 或 BETWEEN 50 AND 100。

IN 运算符是用来核查是否等于列表中的数。例如,用户要核查字段值是否等于 1、2、5、7、9 中的值,则可在有效性规则框中输入 IN (1、2、5、7、9)。

如果需要检查一个文本或备注字段是否是一个匹配模式,例如邮政编码、电话号码,则可用 LIKE 运算符。Microsoft Access 提供了一个文本字符串作为一个比较值来定义哪一个字符在哪个位置有效。

Microsoft Access 提供了如下的通配字符,可以用这些通配字符在某个位置上代替包括单个字符、零或多字符或任意单一数字,详细如下:

- ? 任何单一字符
- * 零或多字符,用来定义标题、结尾
- # 任何单一数字

也可以指定,在字段的任何位置只包括来自用户提供的列表的字符。为了定义一个有效的字符序列,可将此序列放在方括号([])内,可用格式[低值—高值]来指定字符范围,如[A—H]、[1—10]等。表 2.4 给出了 LIKE 运算符的一部实例及其意义。

给字段加有效期规则,除直接在框内输入表达式外,还可用表达式生成器完成。置焦点于有效期规则框内,此时用户会看到带有三个点的按钮出现在特性框的右边,即为生成器按钮,见图 2.16。单击生成器按钮,Microsoft Access 会弹出“表达式生成器”窗口,如图 2.17 所示,可以将窗口分为三个部分。

表 2.4 LIKE 运算符

字 段	表达式	显 示
公司名称	Like " A*"	已运货到客户的订单, 订单中的客户名称以字母 A 开头
公司名称	Like " *Imports"	已运货到客户的订单, 订单中的客户名称以 "Imports" 结尾
公司名称	Like " [A-H]*"	已运货到客户的订单, 订单中的客户名称以字母 A 到 H 开头
公司名称	Like " *tr"	已运货到客户的订单, 订单中的客户名称包含字母串 "tr"
公司名称	Like " Maison Dewe?"	已运货到客户的订单, 订单中的客户名称以 "Maison" 作为名称的第一部分, 并具有五个字母长的第二名称, 且其中前四个字母是 "Dewe", 而最后的字母为未知的

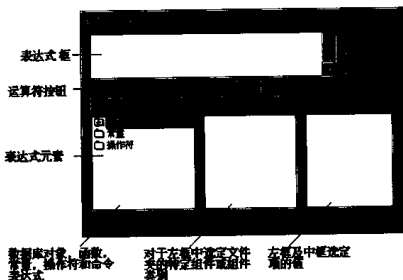


图 2.17

在生成器窗口的上部是一个表达式框, 用于创建表达式, 创建的表达式就显示在这里。

在生成器窗口的下部是用于创建表达式的元素, 用户可将这些元素粘贴到表达式框中以形成表达式, 也可以直接在表达式框中键入表达式的某一部分。

在生成器中部是常用运算符按钮。如果单击运算符的某个按钮, “表达式生成器”将在表达式框中的插入点位置插入相应的运算符。有关表达式可用运算符的完整列表, 可单击左下框的“运算符”文件类和中部框中相应的运算符类。右边的框将列出选定分类中的所有运算符。

生成器下部含有三个框:

- 左边的框包含文件类, 该文件类列出表、查询、窗体和报表等数据库对象以及内置和用户定义的函数、常量、运算符和常用表达式。
- 中间的框列出左边框中选定文件类中的元素或指定元素的类别。例如, 如果在左边的框中单击“内置函数”, 中间的框便列出 Microsoft Access 函数的类别。
- 右边的框列出了在左边和中间框中选定元素的值。例如, 如果在左边的框中单击“内置函数”, 并在中间框中选定了一种类别, 右边的框将列出选定类别中所有的内置函数。

在完成表达式以后, 单击确定按钮, 完成表达式的创建。

2.4.5 定义输入掩码

为了帮助用户输入格式化数据, Microsoft Access 允许用户为文本型、数值型、货币型、日期型、数据设置输入掩码。用户可用输入掩码, 将输入字母变为大写或向电话号码中加入括号等。输入掩码主要用于文本与日期/时间型。

在输入掩码表达式中, 需要使用某种标记定义字符, 建立输入标记。这些标记字符及其作用见表 2.5 所示。

表 2.5 掩码表达式标记字符

字 符	说 明
0	数字 (0 到 9, 必选项; 不允许使用加号 “+” 和减号 “-”)
9	数字或空格 (非必选项; 不允许使用加号和减号)
#	数字或空格 (非必选项; 空白将转换为空格, 允许使用加号和减号)
L	字母 (A 到 Z, 必选项)
?	字母 (A 到 Z, 可选项)
A	字母或数字 (必选项)
a	字母或数字 (可选项)
&	任一字符或空格 (必选项)
C	任一字符或空格 (可选项)
.,:;- /	十进制占位符和千位、日期、时间分隔符 (实际使用的字符取决于 Windows “控制面板” 的 “区域设置” 中指定的区域设置)
<	使其后所有的字符转换为小写
>	使其后所有的字符转换为大写
\	使其后的字符显示为原义字符, 可用于将该表中的任何字符显示为原义字符 (例如, \A 显示为 A)
密码	为 “密码” 设置 “输入掩码” 属性, 以创建密码输入项文本框, 文本框中键入的任何字符都按原字符保存, 但显示为星号 (*)

一个输入掩码表达式由三部分组成, 中间有分号进行分隔, 如: (999) 000-0000!; 0; “”。第一部分输入掩码本身。第二部分决定是否保存原义显示字符。0 = 以输入的值保存原义字符, 1 或空白 = 只保存输入的非空格字符。第三部分显示在输入掩码处的非空格字符, 可以使用任何字符。“” (双引号、空格、双引号) 代表一个空格。如果省略该节, 将显示下划线 (_)。

输入掩码表达式最简单的方法是使用输入掩码向导, 但只有在字段类型是文本型与日期/时间型时才能使用; 对于数值型与货币型, 只能手写输入, 不能利用输入掩码向导。

在客户表中, 邮政编码、电话与传真两字段适合使用输入掩码属性。在 “客户: 表” 的设计视图中, 将焦点设在字段中, 然后用鼠标在输入掩码属性框中单击, 在框的右边出现生成器按钮, 如图 2.18 所示。单击生成器按钮激活 “输入掩码向导”, Microsoft Access 弹出如图 2.19 的窗口。在图中可以看到, 向导已产生一些标准输入掩码, 单击邮政编码, 同时用户可在尝试域中单击鼠标以试验掩码。

客户ID	文本	基于客户名称的 5 字拼音代码。
公司名称	文本	
联系人姓名	文本	
联系人头衔	文本	
地址	文本	街道或邮政信箱。
城市	文本	
地区	文本	州/国家。
邮政编码	文本	
国家	文本	
电话	文本	电话号码包括国家代码或区号。
传真	文本	电话号码包括国家代码或区号。

图 2.18

身份证号码	110107680603121
Password	*****
中文日期	1996年4月24日
日期	1996/4/24
创建日期/时间	06:05:10 AM 1996

图 2.19

单击“下一步”按钮，画面移至下一屏幕，如图 2.20 所示。

在这个窗口中，用户可以看到输入掩码名、推荐的标记字符串、一个用来挑选占位符的列表框以及另一个尝试文本框。邮政编码占位符可不用。单击“下一步”，移到下一屏幕。在这个屏幕上，用户可以选择如何存储数据，是否使用掩码中的符号。再单击“下一步”，出现最后一个画面，然后单击完成，将标记写入“输入掩码向导”属性框中，如图 2.21 所示。

身份证号码	110107680603121
Password	*****
中文日期	1996年4月24日
日期	1996/4/24

客户名称 (必填项)

图 2.20

客户名称 (必填项)

图 2.21

2.5 定义主关键字

Microsoft Access 数据库中的每一个表都应该有一个关键字。

在 Microsoft Access 中定义主关键字是十分简单的。首先选择要定义为主关键字的一个或多个字段。如果要选择一个字段，单击行选择器；如果要选择多个字段，按下 Ctrl 键，然后对每一个所需的字段单击行选择器。最后单击工具条中的“主关键字”按钮，或选择“编辑”菜单中的“主关键字”命令。

在设置完主关键字以后，在相应字段左侧会出现主关键字标记。“客户：表”中的主关键字为“客户 ID”，图 2.22 所示就是定义完输入掩码和主关键字的“客户：表”。

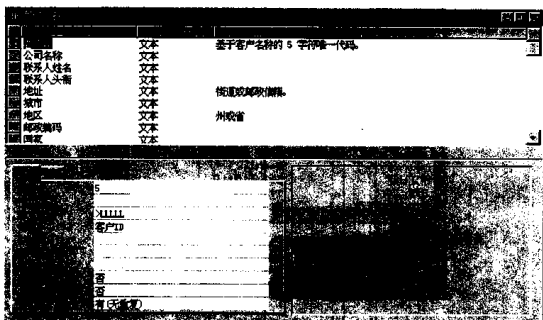


图 2.22

2.6 定义表属性

在“设计”视图中，单击工具栏“属性”按钮，Microsoft Access 会显示表属性窗口，如图 2.23 所示。表属性是指一些和整个表有关的参数，其中最常用的是设置表的有效性规

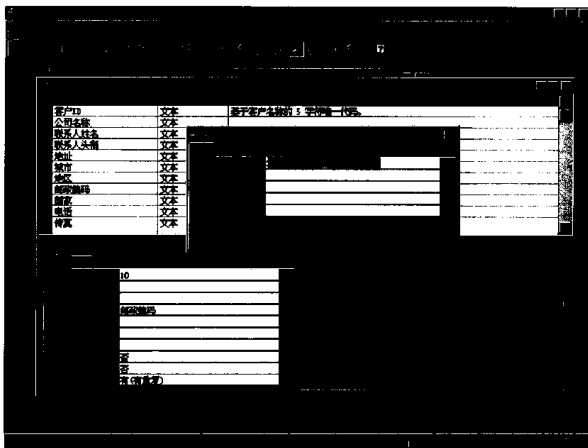


图 2.23

则与有效性文本。通过设置表属性，可以更改表的外观与特性。

在说明属性中，可以输入描述此表的说明文字，最长不超过 255 个字符。在有效性规则中，同字段的有效性规则一样，也可以输入表达式，来判断表中的内容是否有效。同字段的有效性规则属性不同，在这里比较的对象可以是字段的值，也可以在字段之间作比较。需要指出字段名称时，把该字段名放在方括号中；当表达式中有其他任何对象时，也需加方括号。

2.7 表的存储

在表设计中的任何时候，用户都可以从“文件”菜单中选择“保存”命令或按工具条中的保存按钮，来保存新表或对表所作的修改。如果用户是首次保存该表，则 Microsoft Access 会显示“另存为”对话框，如图 2.24 所示。在“表名称”文本框中键入用户为表起的名字，再按“确定”按钮，完成表的保存。



图 2.24

在对表进行命名时，应避免使用表 1、表 2 之类的表名，而应该给表起具有意义的名称，使用户能一目了然。表名可以使用汉字，可以有空格，但应尽量不使用空格。为方便引用，表名应尽量短，而且不要使用重复表名，因为表名重复将引起覆盖，当然用户要是想删除旧表，可用原有表的表名作为新表的名称。

表，以至于所有 Microsoft Access 对象的命名，应符合 Microsoft Access 命名规则。规则可以简述为：在 Microsoft Access 中，对象名最多可达 64 个字符长，可以包括字母、数字和空格以及除句号、惊叹号、重音符号和方括号外的所有特殊字符，同时，在对象名中还不能使用前导空格和控制字符（ASCII 值从 0 到 31 的字符）。

2.8 使用表向导创建表

数据库中表很多，要是每个表都必须由人工创建，将是一件十分令人厌烦的事，幸运的是，“表向导”能帮助用户创建表。

在图 2.10 中，选中“表向导”，然后单击“确定”按钮，Microsoft Access 就显示出表向导创建数据库的第一个画面，如图 2.25 所示。

在屏幕的左下方，有两个选择按钮：“商业”和“个人”。由于 Northwind 是一个商业性的，用户可能在商业样例中发现合适的表。仍以创建客户表为例，在样本表列表框中选中“客户”，向导在中间列表框中列出了所有的样本字段。

为建立客户表，需要从样本字段中选择所需字段。

为了选取一个字段，在中间列表框中选定它，并单击右箭头键（>），把它移到最右边

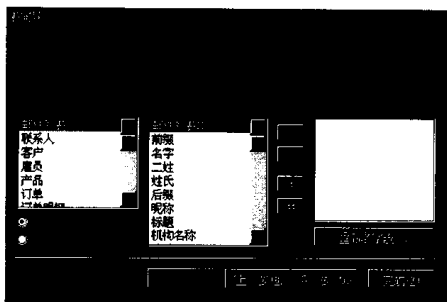


图 2.25

的列表框中；也可以双击所需字段来选中它。用户可以依据从样本列表中的字段顺序来定义客户表中的字段顺序。如果选择了不需要的字段，用户只需在右边列表框中选中它，再单击左箭头键（<），移走此字段。如果用户需要重新开始，则单击双左箭头（<<）按钮来移走全部字段。

由于示例“客户：表”中的几乎所有字段都是新表中所需要的，可以单击双右箭头（>>）按钮来全部选中，然后将相应不需要的字段移去。若是需要更改字段名称，则按“重命名字段”按钮，Microsoft Access 显示“更改字段名”对话框，在文本框中键入新字段名，单击“确定”，完成更改。

在完成新表字段的选择之后，单击“下一步”按钮，用户可以看到图 2.26 的画面。在

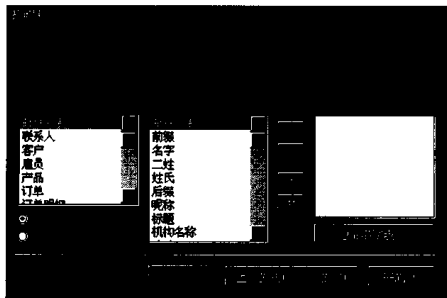


图 2.26

何种视图下打开表,要想转换显示视图,单击工具条中的视图按钮,再选中对应的视图即可,十分简单。数据表视图将在第五章中作详细说明。

2.10 定义关系

在 Microsoft Access 数据库中,不可能只有一个表,而且表与表之间不会是相互独立的,肯定是通过字段相互联结在一起,只有用户告诉 Microsoft Access 表是如何联系的,在表查询、窗体、报表中,才能知道如何连接所有的表,这就需要定义关系。

在定义关系之前,用户必须回到数据库窗口下,必须关闭所有的表,因为不能在已打开的表之间创建或修改关系。在关闭表之后,如果还没有切换到数据库窗口,可以按 F11 键从其他窗口切换到数据库窗口。

单击工具栏上的“关系”按钮,如果数据库没有定义任何关系,将会自动显示“添加表/查询”(“显示表”)对话框,如图 2.28 所示。如果需要添加一个关系表,而“添加表”对话框却没有显示,请单击工具栏上的显示表按钮。双击要作为相关表的名称,然后关闭“添加表/查询”对话框,此时,“关系”窗口中将显示这些表的所有的字段,如图 2.29 所示。再从某个表中将所要的相关字段拖动到其他表中的相关字段,如图 2.30 所示。如果要拖动多个字段,在拖动之前请按下 Ctrl 键并单击每一字段。在大多数的情况下,表中的主关键字段(以粗体文本显示)将拖动到其他表中的名为外部关键字的相似字段(经常具有相同的名称)。相关字段不需要有相同的名称,但它们必须有相同的数据类型(有两种例外的情况),得包含相同种类的内容。此外,当匹配的字段是 Number 字段时,它们必须有相同的“字段大小”属性配置。

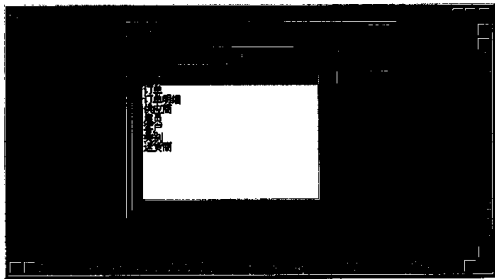


图 2.28

匹配数据类型的两种例外情况是:可以将 Auto Number 字段与“字段大小”属性设置为“长整型”数据类型的 Number 字段匹配;可以将 Auto Number 字段与“字段大小”属性设置为“同步复制标志号”数据类型的 Number 字段匹配。

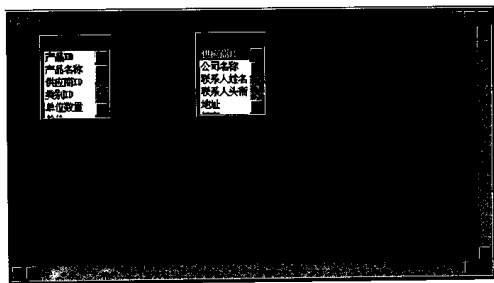


图 2.29

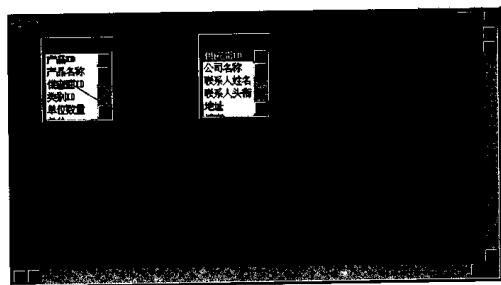


图 2.30

Microsoft Access 自动显示“关系”对话框，如图 2.31 所示，检查显示在两个列中的字段名称以确保正确性。必要时可以进行更改。单击“创建”按钮创建关系。对每一对要关联的表，请重复以上步骤。在关闭“关系”窗口时，Microsoft Access 将询问是否要保存此布局配置。不论是否保存此配置，所创建的关系都已保存在此数据库中。

图 2.32 显示了 NorthWind 数据库中所有主要表的“关系”窗口。

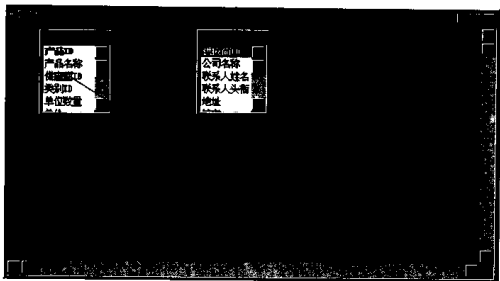


图 2.31

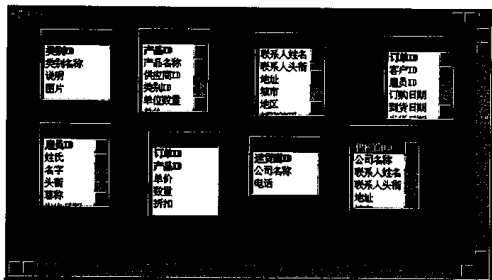


图 2.32

2.11 增加索引

用户的表中的数据越多，就越需要索引来帮助 Microsoft Access 更有效地搜寻数据。索引有助于 Microsoft Access 快速查找和排序记录。Microsoft Access 在表中使用索引来查找数据，就像在书中使用索引来查找数据一样方便。

假设经常要查询 Northwind 数据库中客户表，没有索引，当让 Microsoft Access 去找 Utah 州的所有客户时，Microsoft Access 不得不查寻表中的每一个记录。如果只有十几个客户，查寻将会很快；但是如果有一百个或几千个客户，查寻将很慢。如果用户在 State 字段上建立索引，Microsoft Access 就能利用索引直接在指定的州中找出客户的记录。

大多数情况下，都希望对经常搜索的字段、排序字段或查询中联接到其他表字段的这些字段设置索引。然而在执行这些操作时，因为有许多字段的索引需要更新，索引可能会减慢

一些操作查询的执行速度，如追加查询。

表的主关键字将自动设置索引，而对 Memo、Hyperlink、OLE Object 等数据类型的字段则不能设置索引。对于其他的字段，如果符合下列所有条件，可以考虑对字段设置索引。

- 字段的数据类型为 Text、Number、Currency 或 Date/Time。
- 搜索保存在字段中的值。
- 排序字段中的值。
- 在字段中保存许多不同的值。如果字段中许多值是相同的，则索引可能不会显著地加速查询。

索引可以分为单一字段索引与多字段索引两种。一般情况下，表中的索引为单一字段索引。在单一字段上建立索引是十分简单的。打开设计视图，在窗口上部，单击要创建索引的字段；在窗口下部，单击索引属性框，然后选择“有（有重复）”或“有（无重复）”。选择“有（无重复）”可以保证任何两个记录的这一字段没有重复值。以“客户：表”为例，为“客户：表”建立单一字的索引，以城市为索引，则如图 2.33 所示。

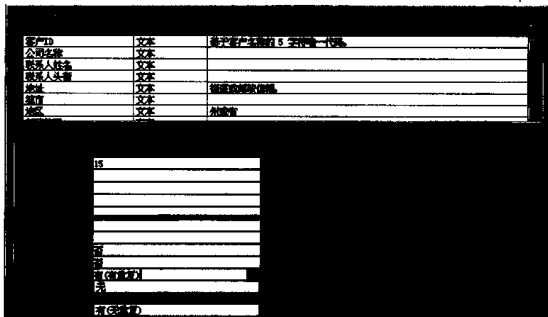


图 2.33

如果经常需要同时搜索或排序两个或更多的字段，可以为组合字段创建索引。例如，经常在同一查询中对“姓氏”与“名字”字段设置准则，就应该在这两个字段上创建多字段索引。

下面是建立多字段索引的方法。

首先在“设计”视图中打开表，单击工具条中的“索引”按钮，Microsoft Access 显示出如图 2.34 所示的“索引”窗口。窗口中已有几个单一字段索引。

在“索引名称”列的第一个空白行，键入索引名称。用户可以使用索引字段的名称之一来命名索引，或使用其他名称。在字段名称列中，单击向下的“箭头”，选择索引的第一字段，在“字段名称”列的下一行，选择索引的第二字段，该行的索引名称列为空。重复该步，直到选择了索引包含所有的字段，最多为 10 个字段。“排序次序”列的默认值是“升

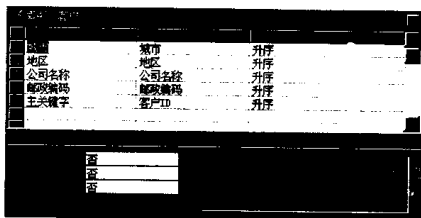


图 2.34

序”，选择为“降序”可按递减顺序排列相应的字段数据。

为客户设计一多字的索引，它由国家与城市、公司组成，定义名称为国家城市公司，最后状态如图 2.35 所示。

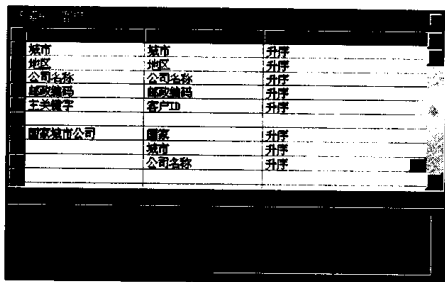


图 2.35

在使用多字段索引排序表时，Microsoft Access 将首先使用定义在索引第一字段进行排序。如果记录在第一字段中有重复值，则 Microsoft Access 再使用索引中的第二个字段进行排序，以此类推。

在查寻中使用多字段索引时，应注意：只有最后一个标准可以是不等式，例如 >、<、>=、<=、<>，因此用户要利用图 2.35 所示索引进行查询时，表达式可写为

国家 = CA

国家 > CA

国家 = GERMAN AND 城市 = BERLIN

国家 = GERMAN AND 城市 = BERLIN AND 公司名称 > Ma

但如果用户要求下列查询，则不能利用图 2.31 中的多字段索引。

国家 = GERMANY AND 公司名称 = Marble (不能跳过一个字段)

国家 > CA AND 城市 > San (只有最后一个字段可以是不等式)

城市 = BERLIN (一定要含有第一个字段)

第三章

修改数据库设计

由于在设计数据库时不可能做到毫无差错，所以用户在以后就会修改它，而表是修改的主要对象。

由于下列原因，用户就必须修改表的设计。

● 要执行一些新的任务，它不仅要求建立新表，而且还要求用户在现存表中插入一些字段。

- 由于表中的某些字段使用频率较高，须将它们提前以提高工作效率。
- 需要删除某些字段。
- 需要增加一些与已存在字段非常相似的新的字段。
- 需要更改字段的数据类型，方便存储。
- 由于原先的字段太小，存不下某些数据，需要改变字段的长度。
- 将现存的表行拆成两个表可以提高效能。
- 由于原先的关键字值不唯一，需更新主关键字。
- 需要增加索引，以改善查询效率。
- 原来是 Microsoft Access 97 以前版本的数据库，要转化为 Microsoft Access 97 下的数据库。

在学习本章之后，用户将学会如何修改表的设计，仍以 Northwind 数据库中的客户表为例。

3.1 准备工作

由于表是整个数据库的核心，虽然用户可以很方便地修改表的设计，但不管用户要增加一个表或删除一个表、删除表内的字段或更改字段名，这些操作都将影响到原有的其他对象，如查询、窗体、报表，因此，在对表作修改之前，应该先仔细考虑如下各方面。

首先，由于表是数据库的核心，其他对象包括查询、窗体、报表都是在表的基础上衍生出来的。如果用户修改了某个表的定义，那么必将影响到其他对象，一旦修改的内容与已有的查询、窗体、报表有所抵触，那势必造成数据的混乱。而且一个数据库内都有十几或几十个表、查询、窗体、报表，用户可能已忘记表与它们之间的关系，Microsoft Access 又不会自动修改，因此用户在设计数据库时，必须详细记录设计时的文档，以免日后在需要修改表时造成麻烦。

其次，用户在表被打开或使用的情况下，不能修改表的设计，必须先将与此表有关的其他对象关闭之后，才能进行修改。如果数据库是在网络上的，必须在所有用户都退出使用该表时，才能进行修改，因此修改表的时间是个十分重要的问题。

再次，用户不能修改关联字段，要修改表中的关键字段，必须将关联先去掉；否则

Microsoft Access 将阻止用户修改。

最后, Microsoft Access 在完成修改后,在存盘之前会给出提示信息,以提醒用户是否真要改变。

在进行修表之前,最好对数据库或相应的表进行备份,这样可以保证安全性。

3.1.1 备份表

如果用户需要修改多个表,最好将整个数据库进行备份,这只需进行文件拷贝即可。

在 Northwind 数据库中,要修改客户表,需要对客户表进行备份。按 F11 键打开“数据库窗口”,按下窗口中的表按钮,在列表框中显示所有表的名称,用鼠标选中要备份的对象——客户表,然后在编辑菜单中选择复制命令,或按 Ctrl + C 键。图 3.1 所示是利用菜单进行复制。

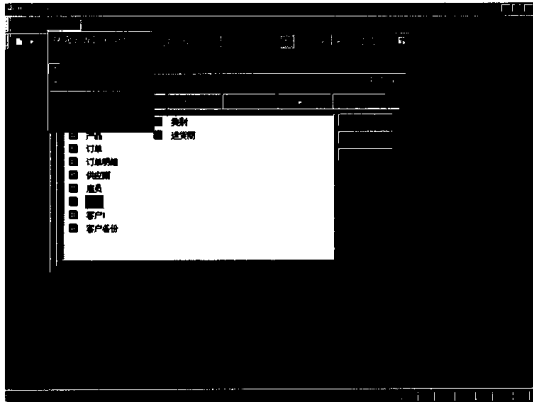


图 3.1

此时,客户表已复制到剪贴板上,再在“编辑”菜单中选择“粘贴”命令,或按 Ctrl + V 键,Microsoft Access 会弹出“粘贴表方式”对话框,如图 3.2 所示。用户可以在“表名称”项中输入备份的名称。用户还可在粘贴选项中选择备份表中的内容,是表的结构、数据,或是数据和结构一起,然后单击“确定”,完成备份。

备份的另一种方法是,在选中客户表后,单击鼠标右键,显示快捷菜单,如图 3.3 所示。在快捷菜单中选中“另存为/导出”命令,Microsoft Access 将显示如图 3.4 所示的“另存为”窗口。在窗口中有两个选项,可以将原表另存为一个外部文件或数据库,也可以另存为本数据库中的表。选择最上项,在新名称框中,Microsoft Access 缺省地给出新表名“客户的备份”。一般可以选用此名,按“确定”按钮,完成备份。

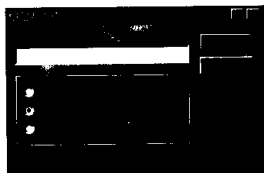


图 3.2



图 3.3



图 3.4

3.1.2 复原

用户在修改数据库过程中，如果发现修改错误，可以选用“编辑”菜单中的“撤消”命令来恢复到修改之前的状态，但撤消命令只能回到上一步状态。要想恢复到最先状态，有两种方法：一种是不保存修改后的表，只需关闭表，Microsoft Access 会弹出如图 3.5 所示的窗口，单击“否”按钮，放弃保存即可；另一种是用备份表代替当前表，回到最初的状态。



图 3.5

3.1.3 将数据库升级至 Microsoft Access 97

如果用户要修改 Microsoft Access 97 以前版本的数据库，首先必须将旧版本数据库转换为 Microsoft Access 97 数据库。

为了安全起见，在转换数据库之前，最好首先备份要转换的数据库，保留该副本直到已经能够熟练使用 Microsoft Access 97 的数据库。由于转换数据库时，该数据库必须是关闭

的,因此必须先关闭要转换的数据库;如果数据库位于服务器或位于共享文件夹中的多用户(共享)数据库中,应确定所有用户都已关闭该数据库。

在 Microsoft Access 97 中,指向“工具”菜单上的“数据库实用工具”子菜单,然后单击“转换数据库”命令。Microsoft Access 将显示如图 3.6 所示的“数据库转换来源”对话框。

在“转换数据库来源”对话框中,选择要转换的数据库,然后单击“转换”按钮,Microsoft Access 将显示“转换数据库到”对话框。在对话框中,输入 Microsoft Access 97 数据库的新文件名(不包括 mdb 扩展名),选择 Microsoft Access 97 数据库文件保存的位置(可以保留原来的名称,也可以进行更改),然后单击“保存”按钮,Microsoft Access 将把数据库转换为 Microsoft Access 97 格式。

在转换带有链接表的旧版本 Microsoft Access 数据库时,应确保外部表位于其原始文件夹中。如果 Microsoft Access 找不到链接到转换数据库的表,将不能使用转换数据库命令。可以在转换数据库以后移动这些表,并使用“链接表管理器”进行重新链接。转换数据库时并不转换链接到数据库的表。

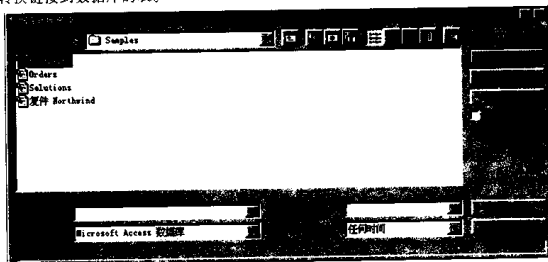


图 3.6

3.1.4 打开未转换的旧版本数据库

可能在所有用户不能同时升级到 Microsoft Access 97 的多用户环境下,需要使用数据库。在这种情况下,必须同时在 Microsoft Access 的不同版本中使用数据库,其中已经升级到 Microsoft Access 97 的用户可以以旧版的格式使用数据库。当 Microsoft Access 97 的用户使用数据库时,Microsoft Access 将保留原始格式,以使旧版本 Microsoft Access 的用户能够继续使用数据库。例如,在 Microsoft Access 2.0 版中创建的共享数据库,可以继续 Microsoft Access 2.0 版中使用,并在 Microsoft Access 97 中使用。

使用 Microsoft Access 95 和 Microsoft Access 97 的用户都可以使用 Microsoft Access 1.x 或 2.0 版的数据库,也可以在 Microsoft Access 97 中使用 Microsoft Access 95 的数据库。使用 Microsoft Access 97 的用户使用旧版本数据库时,可以查看数据库对象,添加、删除或修改记录,但是不能修改对象的设计。如果要修改已有对象的设计或添加新对象,必须使用创

建数据库的 Microsoft Access 版本打开数据库。Microsoft Access 97 的数据库不能在旧版本的 Microsoft Access 中打开。

除使用以外，也可以在 Microsoft Access 1.x 或 2.0 版、Microsoft Access 95 以及 Microsoft Access 97 之间共享前端/后端应用程序，并使后端数据库维持其最早的 Microsoft Access 版本。

在 Microsoft Access 97 和 Microsoft Access 95 中，表的基本结构是相同的，因而可以在 Microsoft Access 95 和 Microsoft Access 97 之间共享“前端/后端”应用程序，而不需要转换或使用前端或后端数据库。例如，仅包含表的后端数据库可能是 Microsoft Access 97 的数据库，而前端数据库可以驻留在使用 Microsoft Access 95 的计算机中，也可以驻留在使用 Microsoft Access 97 的计算机中。

Microsoft Access 97 支持新样式的工具栏和菜单栏。在 Microsoft Access 97 中使用 Microsoft Access 95 的数据库时，自定义工具栏将转换为新的样式，但是该转换并不保存。而自定义菜单栏将解释为新样式的菜单栏，但菜单栏宏并不转换而是继续使用。

对于安全数据库，可以使用“工作组管理器”加入安全数据库的 Microsoft Access 1.x 或 2.0 版工作组信息文件，以确保保留其所有的安全设置。只要在原始的工作组中使用数据库，就将保留所有的安全设置。

在使用数据库时，数据库可能会变大。如果数据库中包含项目，Microsoft Access 97 会转换项目并将其保存在名为 MSysModules2 的隐含系统信息表中，以便数据库能够在 Visual Basic for Applications 的当前版本下工作。根据项目的大小不同，MSysModules2 表的附加部分可能达到数据库大小的两倍。如果在 Microsoft Access 95 和 Microsoft Access 97 中使用同一个数据库，则项目可能变得更大。在很少的情况下，可以在 Microsoft Access 2.0 版中使用 Microsoft Access 1.x 版的数据库，然后在 Microsoft Access 95 中使用它，再在 Microsoft Access 97 中使用；在这种情况下，项目将分别存储在三个隐含系统文件中。

3.2 删除表

也许在数据库中含有用户不再需要的表，虽然不删除它并没有什么影响，但它占据了数据库的存储空间，最好删除它。

在删除表之前，必须先将该表关闭，而且，如果该表与其他表之间有关系存在，也应先删除关系。

回到“数据库”窗口中，删除表是十分简单的。只需在窗口中单击所要删除的表，然后按 Del 键，或在“编辑”菜单中选择“删除”命令，此时 Microsoft Access 会打开消息框，如图 3.7 所示。如果确定删除该表，按“是”按钮，则该表的名称就从“数据库”窗口内消



图 3.7

失；如果此时不想删除该表，则按“否”按钮。若在按了“是”按钮删除该表后，想恢复此表，只需立刻单击工具栏上的撤消按钮即可，前提是删除后没有进行其他操作。

3.3 更改表名

为了使数据库管理更方便，取一个有意义的表名是十分重要的。用户可以在 Microsoft Access 中很容易实现更换表名这一功能。

回到“数据库”窗体下，单击表按钮，在显示的表名列表中单击所要更名的表。然后用户可用两种方法更换表名，一是单击“编辑”菜单中的“重命名”命令，如图 3.8 所示；二是单击鼠标右键，Microsoft Access 弹出快捷菜单，在快速菜单中选中“重命名”命令，如图 3.9 所示。在“数据库”窗口中的被更名的表名处出现“复件 Northwind: 数据库”文本框，如图 3.10 所示，在文本框中输入新表名，然后用鼠标在其他地方单击，完成表名的更改。

在更名后，未进行其他操作之前，若用户要取消更名操作，只需单击工具条中的撤消按钮。

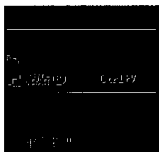


图 3.8

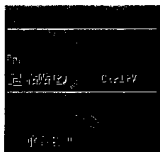


图 3.9

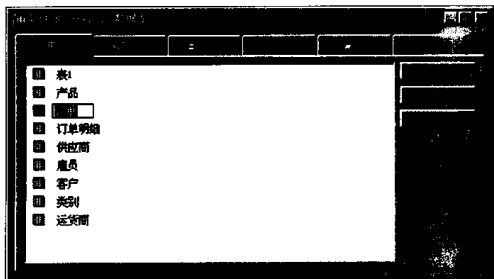


图 3.10

3.4 修改字段名

也许用户在第一次建立表时，拼错了一个字段的名字，或者觉得原来的字段名不合适，不具有充分的描述性，这时就需要将字段名修正过来。

在 Microsoft Access 中，修改字段名是十分容易的。用户可以随时更改字段名而不影响到字段中的数据；但是，更改字段名将影响数据库的其他部分。如果其他数据库对象，如查询、窗体、报表中包含对更改的字段名的引用时，需要更新相应的引用，使它们能够反应更改后新的名称。

用户可以在“设计”视图和“数据表”视图中更改字段名。

以“客户：表”为例，分别在两种视图下更改。将“客户：表”中公司名称字段改为公司。

在“设计”视图中打开“客户：表”，将焦点着于公司名称字段，双击字段名，如图 3.11 所示。键入新字段名，按工具栏上的“保存”按钮，即完成字段名的更改。

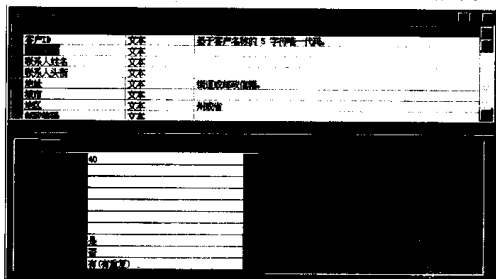


图 3.11

在“数据表”视图中更改字段名，只需在“数据表”视图中打开“客户：表”，再双击要更改字段——公司名称的“字段选定器”。然后键入新的名称，再按 Enter 键保存新名称，见图 3.12 所示。

如果已经为此字段设置“标题”属性，在“字段选定器”中显示的文本与实际的字段名称并不相同。在这种情况下，双击“字段选定器”将清除标题文本并且用字段名称替换。如果继续操作，可以编辑字段名称，当按 Enter 键保存此更改时，标题文本被删除。如果不想删除标题文本，在双击“字段选择器”之后，应立即按 Esc 来取消编辑。

需要注意的是，如果已经定义了一个使用已经修改了其名字的字段的查询、窗体、报表、模块或宏，用户就必须改变在那些对象中的字段名字。

3.5 插入新字段

对数据库中表的修改最普通的大概要属插入新的字段了。

ALPEK	Alfredo Patterkiste	Marie Anders	销售代表
AMATE	Ann Trujillo Esparedados y helados	Ann Trujillo	店主
AMTOS	Antonio Berano Tugueres	Antonio Berano	店主
AMOUT	Around the Horn	Thomas Hardy	销售代表
BERGOS	Berglund's snabbkött	Christina Berglund	采购员
BLAUS	Blauer See Bakikattessen	Hanna Noor	销售代表
BLORP	Blondal pils og fil	Pj. Pj. Miguel Citeaux	市场经理
BOLDI	Boldo Comidas preparadas	Walter Gomez	店主
BORAP	Bon app	Leopoldo Lehihan	销售代表
BUTTA	Butter-Baller Markets	Elizabeth Lincoln	销售经理
BSBEV	B's Beverage	Victoria Ashworth	销售代表
CACTU	Cactus Comidas para llevar	Pedroic Simpson	销售代表
CEBTE	Centro comercial Bectarus	Francisco Chang	市场经理
CHIPS	Chop-suey Chinese	Yang Yang	店主
COMBI	Comércio Mineiro	Pedro Afonso	销售员
CONSH	Consolidated Holdings	Elizabeth Brown	销售代表
DRACO	Drachmalt Bakikattessen	Sven Ottlieb	采购员
DURON	Du monde entier	Janine Labrousse	店主
EASTC	Eastern Connection	Ann Beron	销售代表
EMER	Emerald Market		

图 3.12

有时候用户可能需要在原表格中插入新城。例如，在“客户：”表中，由于公司可能不只一个地址，比如说有总部地址，还有营业部地址，因此需要再加一个地址字段，设为部门地址字段，将部门地址字段插在地址字段与城市字段之间。

插入新字段操作，也可在“设计”视图与“数据表”视图图中进行，但在“数据表”视图中插入新字段后，不能作进一步的修正、设置属性，因此，一般情况下，在“设计”视图下添加字段较好。

下面介绍在“设计”视图图中插入字段。

首先，在“设计”视图中打开相应的表，这里打开“客户：表”。单击要在其上插入字段的那一行，单击工具栏上的插入行按钮（如果要将字段添加到表的结尾，则单击第一个空白行即可），如图 3.13 所示，此时，在地址与城市字段间出现一空白行。

客户ID	文本	基于客户名称行 5 字符唯一代码
公司	文本	
联系人姓名	文本	
联系人头衔	文本	
地址	文本	街道或邮政信箱
城市	文本	
地区	文本	州/省/市

图 3.13

然后，单击“字段名称”列，并键入新字段名“部门地址”，同时按照前一章所述设置数据类型与说明以及其他属性。

最后按工具栏上的“保存”按钮，保存所作的修改。

加入新字段后的“客户：表”“设计”视图，如图 3.14 所示。

新加的字段的数据均为当前的缺省值。

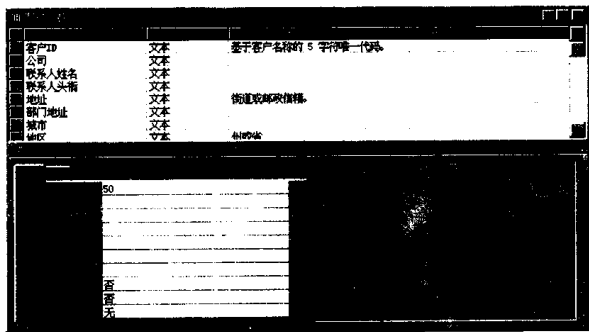


图 3.14

3.6 移动字段

用户也许会觉得移动字段没有多大的用处，因为不管表里的字段如何排列，用户均可以在使用查询、窗体或报表中来操作表格，重新进行排序。但是，在浏览数据时，还是希望数据以一定的顺序排列。

移动字段操作一般在表的“设计”视图中进行。以“客户：表”中为例，现在假设需要将公司字段移至联系人头衔字段后面。

在“设计”视图中打开“客户：表”，选中所要移动的字段——公司，只需单击“行选定器”即可选中此字段。假如要移动的字段有多个，且连结在一起，则按住鼠标左键拖动，经过所需字段的“行选定器”。图 3.15 中表示选中了公司字段。

再次用鼠标单击“行选定器”，并按住鼠标左键，此时 Microsoft Access 将在最后一条选择行的上方显示一条粗体的水平线。将此水平线拖至相应的行，该行的下方为所需移动字段的最终位置。此例中应将水平线移至联系人头衔字段中，如图 3.15 所示。图 3.16 显示了移动字段后的“客户：表”“设计”视图。

单击工具栏中的“保存”按钮保存所作的修改。

移动字段操作也可在表的“数据表”视图中进行，同样，也将“公司名称”字段移到“联系人头衔”字段后面。

首先，在“客户：表”的“数据表”视图中选中“公司名称”字段，只要用鼠标单击该字段的列选定器，此时“公司名称”列变成黑色，如图 3.17 所示。

然后再用鼠标单击该字段，并且按住不放，拖动鼠标指针，在拖动时，指针变成一个小方块，当拖动到某一个字段内时，在这个字段的左侧出现一条粗体的垂直线，图 3.18 所示就是在“联系人头衔”字段后面出现了一条粗体的垂直线。放开鼠标按钮，“公司名称”字段就移到“联系人头衔”字段的后面。图 3.19 所示就是更改后的“客户：表”的结果。

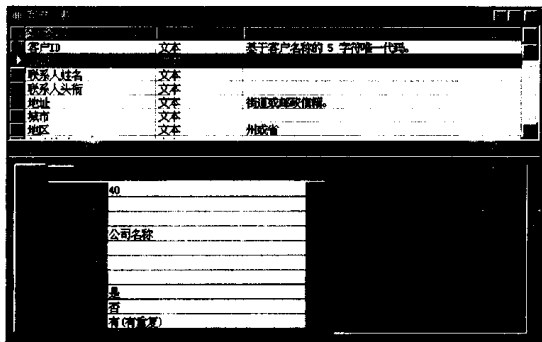


图 3.15

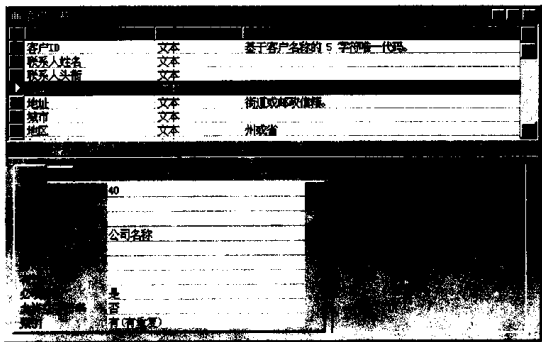


图 3.16

公司名錄		
ALPAC	Boris Anders	銷售代表
ARAB	Ann Truillo	業主
ANTON	Antonio Barano	業主
AROUT	Thomas Hardy	銷售代表
BERGS	Christina Berglund	不納稅
BELAS	Hanna Moss	銷售代表
BELOR	Frédérique Citeaux	市場經理
BOLID	Bertie Sommer	業主
BORAP	Lawrence Labihan	業主
BOTIN	Elizabeth Lincoln	經理經理
BSREV	Victoria Ashworth	銷售代表
CACTU	Patricia Simpson	銷售代表
CEUTC	Francisco Chang	市場經理
CHOPS	Yang Yang	業主
COMBI	Pedro Afonso	銷售員
COISH	Elizabeth Brown	銷售代表
DRACO	Sven Ottlieb	不納稅
EMMON	Janine Labruno	業主
RASC	Ann Devan	銷售代表
ERESH	Roland Mendel	銷售經理
TAMLA	Aria Cruz	市場經理
FLSSA	Diego Real	經理經理
POLIG	Martina Rame	經理銷售代表

图 3.17

公司名錄		
Boris Anders	銷售代表	Obere Str. 57
Ann Truillo	業主	Ard. de la Con
Antonio Barano	業主	Mataderos 2312
Thomas Hardy	銷售代表	120 Hamner Sq.
Christina Berglund	不納稅	Parguavien 6
Hanna Moss	銷售代表	Fensterstr. 57
Frédérique Citeaux	市場經理	24, place Kluge
Bertie Sommer	業主	C/ Aragón, 67
Lawrence Labihan	業主	12, rue des Bou
Elizabeth Lincoln	經理經理	23 Tawasson Bl
Victoria Ashworth	銷售代表	Fennelroy Circ
Patricia Simpson	銷售代表	Cerro 333
Francisco Chang	市場經理	Sierres de Gran
Yang Yang	業主	Hauptstr. 20
Pedro Afonso	銷售員	Av. des Jardins
Elizabeth Brown	銷售代表	Barley Garden
Sven Ottlieb	不納稅	Falkenberg 21
Janine Labruno	業主	87, rue des Cin
Ann Devan	銷售代表	35 King George
Roland Mendel	銷售經理	Kirchstrasse 8
Aria Cruz	市場經理	Rue Dr. 92
Diego Real	經理經理	C/ Barceloneta
Martina Rame	經理銷售代表	104, chemin 4

图 3.18

姓名/角色	地址
Ana Trujillo 店主	Ave. de la Con
Antonio Merano 店主	Mataderos 2312
Thomas Hardy 销售代表	120 Manover St.
Christina Berglund 采购员	Bergersvägen 8
Menna Roos 销售代表	Forststr. 57
Prudence Citoux 市场经理	24, place Kléber
Marthe Sommer 店主	C/ Arangiz, 67
Laurence Labihan 店主	12, rue des Bou
Elizabeth Lincoln 销售经理	23 Janssens Bl.
Victoria Ashworth 销售代表	Familleey Cir.
Patricio Simpson 销售代表	Carrito 333
Francisco Chang 市场经理	Sierras de Gran
Tang Yang 店主	Montate 29
Pedro Afonso 销售员	Av. des Lurda
Elizabeth Brown 销售代表	Berkley Garden
Sven Ottlieb 采购员	Falkenberg 21
Janine Labruna 店主	87, rue des Cin
Ann Deven 销售代表	35 King George
Helene Mandel 销售经理	Kirchgasse 6
Aria Cruz 市场经理	Rue Orpè 92
Diego Reel 销售经理	C/ Moralesm
Martine Ranc 销售经理	104, chemin d
Maria Larsson 店主	Bergsten 24

图 3.19

3.7 拷贝字段

当用户建立表时，会发现表中许多字段的定义十分类似，用户要依次键入是十分麻烦的，实际可以利用 Microsoft Access 提供的字段拷贝功能来实现。在拷贝后，再修改其中不同的地方就可以了。

拷贝字段十分简单，其实就是“剪贴板”的动作。

在“客户：表”中，电话、传真两字段除了字段名称不同外，其他属性均一样，这里就以此为说明拷贝字段的方法。

假设传真字段未输入，而只有电话字段，此时表的“设计”视图如图 3.20 所示。

在“设计”视图中选定所要复制的字段——电话字段。单击字段的“行选定器”或用箭头键把光标移到该字段行上并按 Shift + Space，来选择含有字段定义的完整行。如果要选择一组字段，按住鼠标左键拖动经过所需字段的“行选定器”；再单击工具栏上的“复制”按钮，或者从编辑菜单中选择复制命令拷贝这一行；然后单击第一个空行，再单击工具栏上的粘贴工具，或者选中编辑菜单中的粘贴命令来插入拷贝行，如图 3.21 所示。

在新复制的行中的字段名称列中，修改字段名称。此例中只需将电话改为传真就可以了。若是需要，用户还可以更改字段的其他属性。

最后，单击工具栏上的“保存”按钮保存所作的修改。

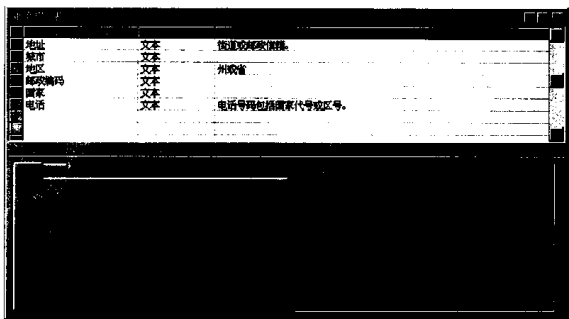


图 3.20

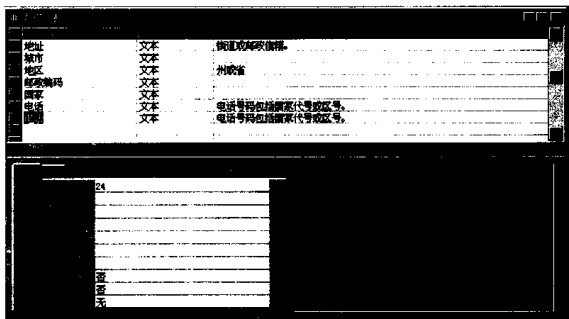


图 3.21

3.8 删除字段

也许用户有时会发现某些字段没有用了，虽然放在表中也没有什么妨碍，但影响到效率与查寻速度，最好将它删除。即使在表的该字段中已有数据存在，Microsoft Access 也允许用户删除该字段。

删除字段十分简单，同样，它也可以在两种视图中完成。下面以删除“客户：表”中的国家字段为例。

要在“设计”视图中删除字段，只要用户在“设计”视图下打开相应的表，在表中选择要删除的字段，在对应字段上单击一下，最后单击工具栏上的“删除行”按钮；也可以从编辑菜单中选择删除行命令，如图 3.22 所示。若是利用“行选定器”选中某一要删除的字段，那么除了以上两种方法以外，还可以直接按 Del 键或选中“编辑”菜单上的“删除”命令。

要在“数据表”视图中删除字段，是在“数据表”视图下打开相应的表，然后单击删除字段的“字段选定器”，再单击“编辑”菜单中的删除列命令。图 3.22 显示了如何在“客户：表”中删除国家字段。

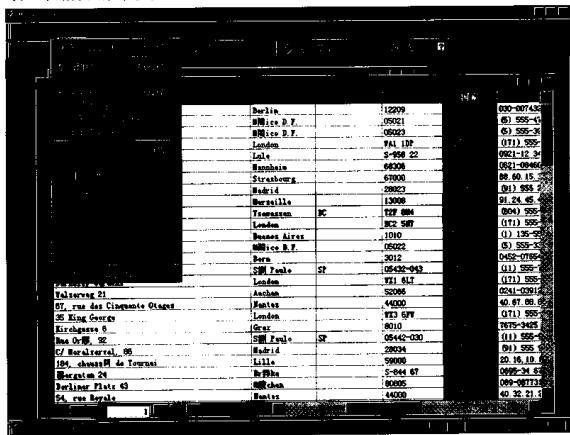


图 3.22

需要注意的是，要删除表的关系中的相联字段，首先必须先删除表之间的关系。

无论在何种视图下删除字段，如果其他数据库对象包含对删除字段的引用时，也需要删除这些引用，因为 Microsoft Access 由于找不到该字段的数据，会产生错误信息。

如果被删除的字段中含有数据，那么在运行删除动作时，Microsoft Access 会打开一个消息框，提示用户是否永久删除，如图 3.23 所示。如果确定删除，则单击“是”按钮，放



图 3.23

弃则单击“否”。

3.9 更改字段属性

在建表时，用户需要为每个字段建立数据类型，每一种数据类型的属性是不相同的。用户在刚建立表时，由于设想得不够周全，经过一段时间后，会发现原来设定的数据类型不够用，需要更换数据类型来适应工作的需要。Microsoft Access 为用户提供了这个功能，即便是该字段下已有数据。

在字段属性中，对数据影响最大的是字段的数据类型与字段的长度。

3.9.1 修改数据类型

改变字段的数据类型十分简单。与在创建表时定义字段数据类型操作一样，只需将鼠标置于所需更改字段的数据类型列中，单击列右侧的下拉按钮，即出现数据类型列表框，然后在列表中选择所需的数据类型，如图 3.24 所示。在更改完成后，单击工具栏内的“保存”按钮保存。

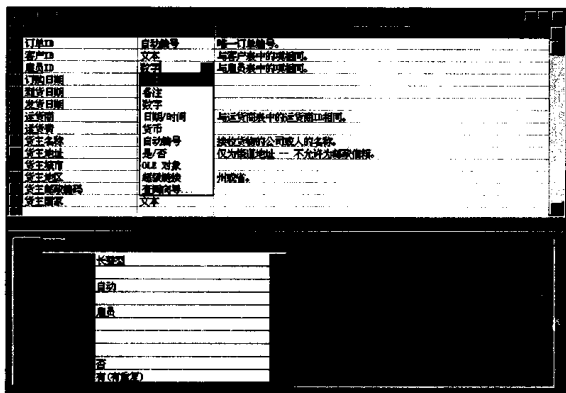


图 3.24

在更换数据类型以后，Microsoft Access 会自动对已有数据进行转换。如果因为数据类型转换造成数据丢失，Microsoft Access 在实际保存结果之前，将显示消息，告诉在转换期发生了错误。此时单击“否”按钮即可取消更改，单击“是”按钮继续并保存更改，如图 3.25 所示。

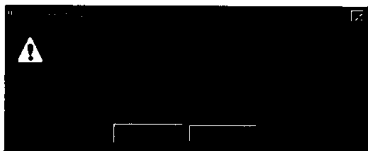


图 3.25

要提醒用户的是,在将字段数据类型转换为另一种类型之前,要事先考虑更改可能对整个数据库造成的影响。有哪些查询、窗体及报表使用了要转换的字段,可能需要更改基于更改字段的表达式。

3.9.2 更改字段大小

只有文本型、数值型字段有“字段大小”属性,因此只有字段的数据类型是文本型或数值型时,才考虑更改字段大小。

要更改字段大小,只需进入表的“设计”视图,单击需要更改字段大小的字段行,在窗口左下部的“字段大小”属性框中单击一下,输入所要的字段大小值。文本型字段,输入值最大为 255;而对于数值型,则单击箭头,选择所需的数值类型。数值类型包括字节、整型、长整型、单精度型、双精度型、同步复制 2D 六种。

3.9.3 更改字段属性的后果

最常更改的数据类型可分为下列四种类别:从其他数据类型转换为 Text 类型;从 Text 类型转换为 Number、Currency、Date/Time 或 Yes/No 类型;Currency 与 Number 之间的转换或 Text 与 Memo 之间的转换;更改 Number 字段的“字段大小”属性设置。

1. 从其他数据类型转换为 Text 类型

如果数字将用于计算,应该将这些数字保存在 Number 或 Currency 字段中,但是有时某个原来只保存数字的字段现在需要使用字母和其他非数值字符,例如某个保存代码数字的 Number 字段,如果稍后需要包含字母、连字符、括号或其他非数值字符作为代码的一部分时,需要将字段的数据类型更改为文本类型。

当 Microsoft Access 将数字值转换为文本时,使用“常规数字”格式;将日期值转换为文本时,使用“常规日期”格式。转换的值中不包含对此字段所设置的任何货币符号或其他特殊格式字符。

2. 从 Text 类型转换为 Number、Currency、Date/Time 或 Yes/No 类型

如果在 Text 字段中保存有数据,并且要将字段改变为另一种数据类型,Microsoft Access 能够将数据转换为适合新数据类型的值,例如将数字保存在 Text 字段中。如果要用这些数据做数学计算,必须将此字段转换为 Number 或 Currency 数据类型。如果字段中所保存的全部数据只为数字时,更改数据类型将不会丢失数据。

对于从 Text 到 Number 数据类型的转换,小数点及千位分隔符可以正确地转换。货币

符号将根据 Windows 控制面板中双击“区域设置”所指定的区域设置值来转换。

对于从 Text 到 Date/Time 的转换,大部分的日期和时间格式都可以正确转换。日期和时间格式将根据 Windows 控制面板中双击“区域设置”所指定的区域设置值来转换。

对于从 Text 到 Yes/No 类型的转换,如“是”、“真”或“开”的词将转换为“是”值,而“否”、“假”或“关”将转换为“否”值。如果要显示“是”或“否”,必须在表“设计”视图中单击“查阅”标签,并设置“显示控件”属性为“文本框”。也可以从“数字”数据类型转换为 Yes/No 数据类型;零或 Null 值将转换为“否”值,而非零值将转换为“是”值。

3. Currency 与 Number 之间的转换或 Text 与 Memo 之间的转换

随时都可以在 Currency 与 Number 之间转换数据类型。应该将有关钱的数据保存为 Currency 数据类型,以确保得到正确的计算结果。如果要在字段上执行很多计算,且该字段包含一到四位数的小数位,也应该使用 Currency 数据类型。Currency 字段使用整型计算方法,且该计算方法可避免四舍五入的错误。也随时可以在 Text 与 Memo 之间转换数据类型。例如,如果要保存比原有数据长的文本时,可以将字段更改为 Memo 数据类型,请记住不能对 Memo 字段索引或排序。但是,如果字段所保存的数据没有超过 255 个字符时,应该使用 Text 数据类型。

4. 更改 Number 字段的“字段大小”(field size)属性设置

对于使用 Number 数据类型的字段,“字段大小”属性的设置将决定数字的类型:“字节”、“整型”、“长整型”、“单精度”、“双精度”或“同步复制标志号”。如果要将字段转换为 Number 类型,请考虑是否需要改变保存在字段中数据的“字段大小”的设置。如果更改“字段大小”的设置,从较大的“双精度”更改为较小的“整型”,必须确保保存的数据能适合于新字段大小。如果更改后的“字段大小”设置不允许当前值中包含小数位,数字将四舍五入。例如,如果将“双精度”更改为“长整型”,小数位将会四舍五入为最接近的整数;如果保存的数据太大而不能存储在新字段中时,这些数据将删除并以 Null 值替换。

数据类型转换结果的总结如表 3.1 所示。

表 3.1 数据类型转换结果的总结

从	到	结 果	注 意
所有数据类型	AutoNumber	Microsoft Access 禁止	无
Text	Number、Currency、Date/Time、Yes/No	将 Text 转换为相应的值	确保这些值能适合于新的数据类型;不适合的值将删除
Memo	Text	简单的转换	比“字段大小”设置长的数据将会截断
Number	Text	将值转换为文本	数字采用“常规数字”格式
Number	Currency	将数字转换为货币	确保这些值能适合于新的数据类型;不适合的值将删除
Date/Time	Text	将值转换为文本	日期或时间采用“常规日期”格式
Currency	Number	简单的转换	确保这些值能适合于新的数据类型;不适合的值将删除

续表 3.1

从	到	结 果	注 意
Currency	Text	将值转换为文本	文本不包含货币符号, 例如 \$
AutoNumber	Text	将值转换为文本	根据“字段大小”的设置, 值可能截断
AutoNumber	Number	简单的转换	根据“字段大小”的设置, 值可能截断
Yes/No	Text	将值转换为文本	无

3.10 更改主关键字

在前面的章节中, 已介绍过主关键字的定义, 通常主关键字域只能存在单一而且独立的数据, 以便与其他表关联起来。如果某表中没有定义关键字字段, 那么用户就不能确立它与其他表之间的关系, Microsoft Access 也无法自动帮用户建立关系。如果有某个表在使用一段时间之后, 忽然发现主关键字设置不恰当, 也就是该字段中有可能存放多笔相同的值, 此时, 应该更改主关键字了。

这里以 Northwind 数据库中客户表为例, 假设原来的主关键字字段设为联系人姓名字段, 但后来在同公司人员接触中发现, 有可能有同名同姓的联系人, 因此就需要将旧的主关键字设置删除掉, 然后增加一个新的字段来当作主关键字。

这个过程可以分为两步: 首先删除旧的主关键字, 再重新设置主关键字。

如果主关键字在某个关系中, 用户在删除主关键字之前, 必须删除这个关系。然后在“设计”视图下打开该表, 单击该表的主关键字字段, 再单击工具栏中的“主关键字”按钮, 如图 3.26 所示。

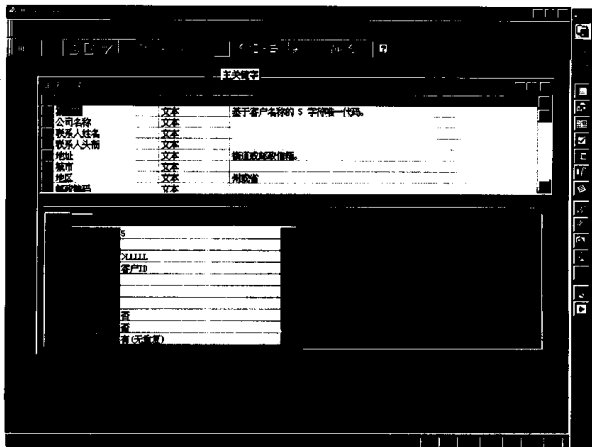


图 3.26

在删除主关键字之后，也许需要新增一个字段作为主关键字，这个字段一般数据类型为自动编码型，可用以前介绍的方法给表重复设置主关键字。

3.11 修改表之间的关系

由于表作了其他的改动，而且在修改表中字段之前，一般要求先将表与表之间的关系删除，那么，在表修改完成之后，就要求用户重新建立表之间的关系。即使表未作改动，有时用户也需要修改表之间的关系。

3.11.1 删除表中已有的关系

在删除关系之前，Microsoft Access 要求用户关闭所有的表，因为不能删除已打开的表之间的关系。在关闭表之后，如果还没有切换到“数据库”窗口，则按 F11 键或单击工具栏上的“数据库窗口”按钮，从其他窗口切换到“数据库”窗口。

单击工具栏上的关系按钮，Microsoft Access 将会显示数据库中已有的一些关系。如果要删除的表的关系没有显示，请单击工具栏上“显示表”按钮，Microsoft Access 弹出用于添加表与查询的“显示表”窗口，如图 3.27 所示。双击这个要添加的表，然后单击关闭按钮。

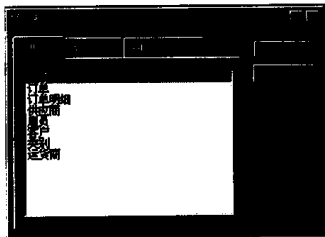


图 3.27

在“关系”窗口中选中要删除的关系，单击此关系的关系连线。选中时，关系连线会变粗黑，然后按 Del 键，即删除关系，如图 3.28 所示。

最后按工具栏中的“保存”按钮保存所作的修改。

3.11.2 编辑表中已有的关系

首先按前面删除关系中的叙述的方法显示出所需的表，然后用鼠标双击所要修改的关系的关系连线，Microsoft Access 会弹出如图 3.29 所示的“关系”对话框，在此对话框中可设置该关系的选项。

修改完成后，单击“创建”按钮，完成修改。

类别ID	产品ID	联系人姓名	订单ID	雇员ID	订单ID	运输商ID	供应商ID
类别名称	产品名称	联系人头衔	客户ID	姓氏	产品ID	公司名称	公司名称
说明	供应ID	地址	雇员ID	名字	单价	联系人姓名	联系人姓名
图片	类别ID	城市	购买日期	头衔	数量	电话	地址
	单位数量	地区	到货日期	职称	折扣		

图 3.28

供应ID	供应ID

☐ 是否关联到行数据
☐ 是否关联到列数据

图 3.29

3.11.3 添加新的关系

添加新的关系已在第2章中叙述过，这里不再重复。

第四章 使用数据表

在完成表的创建与修改之后，本章要介绍如何使用表。

使用表包括在“数据表”视图下观察数据、修改数据单格式、修改数据、对数据进行排序、查寻数据、打印数据等内容。

使用表时，都在“数据表”视图下进行。

4.1 数据表视图

仍以 Northwind 数据库中的“客户：表”为例，在“数据库”窗口中的表标签的列表选中“客户：表”，双击表名，或单击“打开”按钮，就进入了表的“数据表”视图，如图 4.1 所示，图中介绍了该视图下的一些关键术语。

10249	Vins et alcools Chevalier	Buchanan, Steven		94-08-04	9
10250	Toms Spezialitäten	Suyama, Michael		94-08-05	9
10251	Manari Curries	Pesock, Margaret		94-08-06	9
10252	Victuailles en stock	Leverling, Janet		94-08-08	9
10253	Supermerces d'Alsace	Pesock, Margaret		94-08-09	9
10254	Manari Curries	Leverling, Janet		94-08-10	9
10255	Chop-suey Chinese	Buchanan, Steven		94-08-11	9
10256	Richer's Supermarket	Bodsworth, Anne		94-08-12	9
10257	Hallington's Importadora	Leverling, Janet		94-08-15	9
10258	MILA-Peru-Abravos	Pesock, Margaret		94-08-16	9
10259	Ernst Handel	Barolio, Nancy		94-08-17	9
10260	Centro comercial Moctezuma	Pesock, Margaret		94-08-18	9
10261	Otilios KFFeleiden	Pesock, Margaret		94-08-19	9
10262	Que Pasa	Pesock, Margaret		94-08-19	9
10263	Rattlesnake Canyon Grocery	Callahan, Laura		94-08-22	9
10264	Ernst Handel	Bodsworth, Anne		94-08-23	9
10265	Folk och färg	Suyama, Michael		94-08-24	9
10266	Bleuvald pizzeria	Feller, Andrew		94-08-25	9

图 4.1

同“设计”视图一样，当 Microsoft Access 进入“数据表”视图后，工具条也相应变化，显示了一些在“数据表”视图下十分有用的工具，如图 4.2 所示。

- 视图按钮——单击此按钮，下拉显示“设计”视图与“数据表”视图两项，再选择两者之一，单击可实现两种视图之间的转换。

- 保存按钮——单击此按钮，Microsoft Access 保存用户对表所作的删除、修改数据等

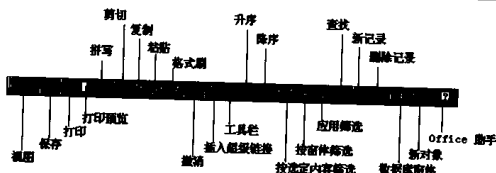


图 4.2

操作。

- 打印按钮——单击此按钮，打印数据单上的数据。
- 打印预览按钮——单击此按钮观察被打印页的预览图。
- 插入超级链接按钮——单击此按钮，可在表中插入一超级链接。
- 工具栏按钮——单击此按钮，在工具栏区内显示 Web 工具栏。

● 升序按钮——先用鼠标在用户要排序的字段列中点一下，然后按下此按钮，那么表内所有的记录会根据这个字段的内容升序排列。

● 降序按钮——选用鼠标在用户要排序的字段列中点一下，然后按下此按钮，那么表内所有的记录会按这个字段的内容降序排列。

● 按选定内容筛选按钮——直接在数据表内选择某个字段值的全部或部分，然后单击此按钮，Microsoft Access 自动为用户在此表内查找具有选定值的所有记录。

● 按窗体筛选按钮——在空白字段中指定要在数据表中搜索的值，可以键入该搜索值或从列表中选择。

● 应用筛选按钮——单击此按钮，进行筛选动作。

● 查找按钮——用鼠标按下此按钮，屏幕上将出现一个“查找”对话框，提供一个简单的查找方式。

● 新记录按钮——按下此按钮，将会跳到这个表的底端的空白横行，用户可以在此插入新的记录。

● 删除记录按钮——在表中选中要删除的记录行，再单击此按钮，Microsoft Access 将删除当前记录。

下面再介绍一下记录指示框。

在表的“数据表”视图中，除了正常窗口拥有的滚动条之外，在窗口左下方还有一个特殊的元件，叫做“记录指示框”，如图 4.3 所示。在记录指示框中，显示着当前的记录号以及表中数据的总数。用户可以不必要再用鼠标单击数据区内的“行选定器”来选择数据，而在记录指示框中简单完成。在图 4.3 中详细说明了各按钮的用途。

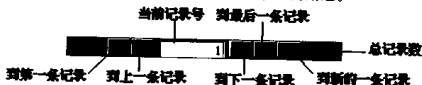


图 4.3

4.2 设置数据表格式

用户虽然可以利用表窗口的“数据表”视图来输入数据，但不能像窗体一样任意安排画面与位置。不过，用户还是可以对画面作一些简单的调整的，例如设置行高和列宽、隐藏或中断字段，还可以改变显示的字体、隐藏或显示格线。这些操作均可在 Microsoft Access 中简单实现，主要在“格式”菜单中完成，如图 4.4 所示。

4.2.1 改变字体

用户在图 4.4 中可以看见，格式菜单中的第一个选项是“字体”。用鼠标在“字体”命令上单击一下，Microsoft Access 将会打开一个“字体对话框”，如图 4.5 所示。这个对话框内分作以下几个部分。



图 4.4

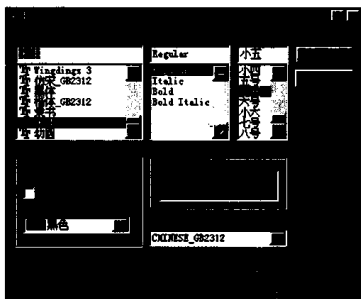


图 4.5

- 字体——这是一个组合框，用来选择使用的字体，包括显示或打印。用户可以用鼠标滚动列表框内的选项，当用户找到喜欢的字体时，再用鼠标单击它。有一点需要特别注意的是，如果该字体应用于打印机，该选项的左边会有一个打印机的符号；如果该字体是给屏幕显示用的，选项左边是空白的；如果该字体两者皆可使用，则屏幕上有两个重叠的 T 符号。

- 字型——当用户选定字体之后，Microsoft Access 会自动判断该字体有几种字型，并且列在字型的列表框中，一般有 Regular（正常）、Italic（斜体）、Bold（粗体）、Bold Italic（粗斜体）。

- 字号——当用户选定字体后，Microsoft Access 会自动判断该字体有几种大小以供选择，并且列在字号列表框中。字号大小的范围变化很大。

- 特殊效果——用户需要下划线或更多显示字体的颜色时，需要在此区域内工作。需要下划线时，只需在下划线选择框中单击一下，若该选择框中显示“√”时，则已选中；若

再单击，则选择框内空白。需要特殊的颜色时，可单击颜色列表框，然后在颜色列中选需要的颜色。

- 示范——当用户选定诸如字体、字型、大小、特殊效果等选项时，Microsoft Access 会根据用户当前的选择，在示范这个局部里显示该字体。用户可由此决定是否选择该选项。

- 语系——可以在此处选择语系，Microsoft Access 自动根据用户选定的语系显示其他选项，主要是字体选项。

在窗口的最下面是提示，Microsoft Access 将告诉用户字体的类型，以及字体是否可以用于屏幕显示或者打印机。

4.2.2 单元格效果

格式菜单的第二项是“单元格效果”，单击它，Microsoft Access 将显示如图 4.6 所示的“单元格效果”对话框。由图中可以看到，该对话框中可以选择以下各项的值。

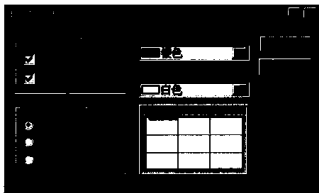


图 4.6

- 网格线显示——网格线是指屏幕上分隔每条记录和每个字段之间的隔线。通常来说，Microsoft Access 都会显示出隔线。如果用户希望在打印表时不打出隔线，就应该将网格线取消。网格线格线方向可分为垂直的与水平的两种，在“单元格效果”中也分别为两种格线做了选择框。在前头的选择框内单击一下，当选择框空白时，取消了网格线；再单击一下，网格线将显示。取消了水平与垂直网格线的“客户：表”如图 4.7 所示。

	Alfredo Patterkiste	Boris Anders	销售代表
AMATE	Ana Trajillo: Noparedados y halados	Ana Trajillo	销售
ANTON	Antonio Barona Tapar	Antonio Barona	销售
ARWY	Arend the Horn	Thomas Hardy	销售代表
BIRMG	Berglund's mabbl	Christine Berglund	采购员
BLANS	Blauer See Delikatessen	Hanna Blauer	销售代表
BLOPP	Bloedel y Söhne et fils	Pr. 周周: que Citizem	市场经理
BOLLE	Böhlke's Conidas preparadas	Burke's Towner	销售
BONAZ	Bon app'	Lawrence Labihan	销售
BOTTN	Bottor-Bellor Markets	Elizabeth Lincella	销售经理
BONNY	B's Beverages	Victoria Aswerth	销售代表
CACU	Cactus Conidas para llevar	Patricia Simpson	销售代理
CENYC	Centro comercial Beteruma	Francisco Chang	市场经理
CHOPY	Choprasay Chinese	Yang Wang	销售
CONKE	Compañia Mineira	Pedro Afonso	销售员
CONSH	Consolidated Holdings	Elizabeth Brown	销售代表
DRACD	Drachmbist Balintessen	Sven Ottlieb	采购员

图 4.7

● **网格线颜色**——用户需要改变网格线颜色时，可以在颜色列表中选一种，当显示数据表时，Microsoft Access 以此种颜色显示表中的网格线。

● **背景颜色**——对于背景颜色，用户有不同的喜好，Microsoft Access 一般以白色作为缺省的背景色。用户只需单击颜色列表中自己喜欢的颜色，以后表的背景色即为此种色彩。

● **单元格效果**——共有三种单元格效果，分别是平面、凸起、凹陷。Microsoft Access 的缺省效果是平面，其他两种使表看上去更加具有立体感。在使用中，用户会发现，即使取消了网格线，但由于单元格效果不是平面，表的视图中仍然有格线，而且背景色也改变了。

● **示范**——Microsoft Access 在示范区内显示了当前选项下的表的样式，便于用户参考。若用户不满意，可立即更换选项。

在选定后，按窗口中的“确定”按钮，Microsoft Access 便以更改后的单元格效果显示表。

4.2.3 改变行高和列宽

一般而言，当用户打开表的“数据表”视图时，Microsoft Access 会自动使用最紧密而又看得清楚的行高作为缺省值。但是，如果用户需要打印表格时，就可能需要更改行高的设置了；否则，打印出来后，可能看上去会太密。

要改变行高设置十分简单，用户只需单击“格式”菜单中的“行高”选项，Microsoft Access 会显示如图 4.8 所示的“行高”对话框。用户可以在对话框内直接输入行高，输入的单位是 Point。这单位的大小由显示器的分辨率决定。用户可以选择“标准高度”作为行高，只需单击“标准高度”选择框，当选择框如图 4.8 中所示时，即选中。



图 4.8

另外，用户也可直接在“数据表”视图中直接更改行高。将鼠标光标移动到表窗口最左边的记录选择区与每条记录的分格线处，此时鼠标光标会变成十字形，按住左键并拖动，直到用户满意的行高为止，松开左键。

图 4.9 显示了更改行高后的“客户：表”，行高为 20。

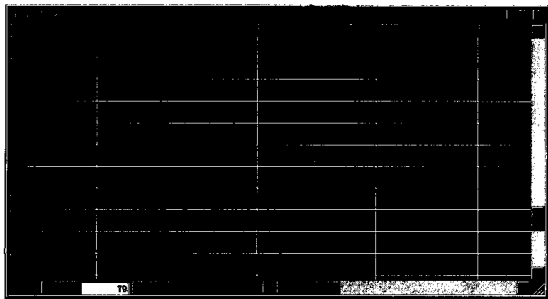


图 4.9

同样，用户有时也需要更改列宽。Microsoft Access 会根据用户所选字体的大小自动计算出缺省宽度，但有时用户会发现有的字段太宽，而有的又太窄了。此时，用户可利用 Microsoft Access 提供的设置列宽功能更改列宽。

更改列宽操作十分简单，先在视图中选中要更改列宽的列，然后在格式菜单中选择列宽命令，Microsoft Access 自动显示如图 4.10 所示的“列宽”对话框。在对话框中，用户可以直接输入列宽，或者选用标准列宽。同样，用户可以用鼠标直接在“数据表”视图中更改表的列宽。将鼠标的光标移至表的上端、“列选定器”的分格线处，此时鼠标的光标会变成十字形，这时按住左键并拖动光标，直至用户需要的宽度，松开左键，完成列宽的更改。

图 4.11 显示了更改列宽后的“客户：表”，列宽是最佳匹配。在“列宽”对话框中，用户单击“最佳匹配”按钮，Microsoft Access 将表的列宽自动置于最佳值。与图 4.9 相比，本图中联系人头衔列的宽度明显缩小。最佳匹配是指与列中数据的最大长度相匹配。

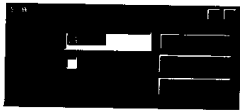


图 4.10



图 4.11

4.2.4 隐藏和显示列

有时候，用户并不想浏览全部字段，这时候，用户可以将某些字段隐藏起来。隐藏列的操作十分简单，有两种方法，一种是利用改变列宽的方法，用鼠标拖动列分隔线，缩小列宽直到该列消失为止；另一种方法更加简单，用户只需在所要隐藏的列中单击一下，选中此列，然后单击格式菜单中的隐列命令，选中的字段将被隐藏。

在表中的字段被隐藏之后，如果用户又想再看这些字段，就必须在格式菜单中选中取消隐藏列项，单击后，Microsoft Access 会弹出“撤消隐藏列”对话框，如图 4.12 所示。图中公司名称列被隐藏，因此它的选择框是空白。单击，选中公司名称，此时选择框中打上勾，表示选中，单击关闭按钮，取消隐藏完成。

当然，在“撤消隐藏列”对话框中，也可以完成隐藏列的功能，只需在相应列名的选择

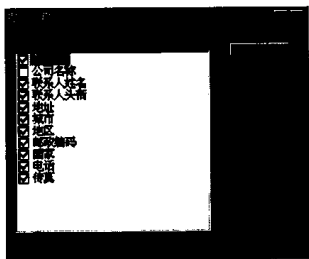


图 4.12

框中单击一下。

4.2.5 移动列

“数据表”视图中，字段从左到右的缺省顺序是用户在表中定义字段的顺序。可以改变列的顺序来观察和打印。用“字段选择器”来选择用户想要的列，Microsoft Access 加亮了整个列；也可以选择多个列，然后把字段拖到理想的新位置上即可。

4.2.6 重命名列

重命名列相当于是对表中的字段名进行更改，如何在“数据表”视图中更改字段名，已在第3章的更改字段名中介绍过了，不再重复。

4.2.7 冻结列

由于一般表的字段都比较多，当用户浏览整张表时，每个字段会在画面上轮流滚动，有时出现，有时消失。有的字段相对重要一些，用户也许希望它不要在屏幕上消失，那么就需要冻结列这一功能了。

在需要冻结的那一列中单击一下，选中此列，然后在“格式”菜单中单击“冻结列”命令，这个字段列将被固定在画面上，不随列的移动而消失。图4.13所示即为在“客户：表”中冻结公司名称列。在冻结以后，公司名称列就自动移至最左端，不随表中字段的滚动而移动。消除网格线后，由于有冻结的列，在第一列后，有一条线，如图4.14所示。

当用户不再需要冻结列时，可选用此命令取消对所有列的冻结。

以上过程均可用 Microsoft Access 中的快捷菜单完成。并且在完成更改之后，需单击工具栏上的“保存”按钮保存修改的结果。

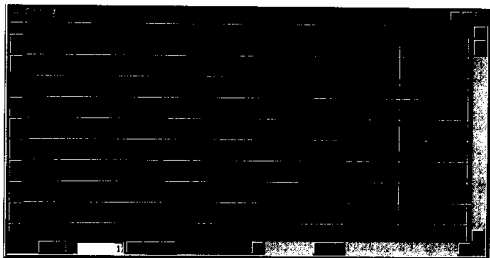


图 4.13

Alfreds Futterkiste	Obispo Str. 57	Berlin
Ana Trujillo Emparedados y helados	Avda. de la Constituci3n 2222	Medico D.F.
Antonio Moreno Taquer3a	Mataderos 2312	Medico D.F.
Around the Horn	120 Hanover Sq.	London
Berglunds snabbköp	Berguvavägen 8	Luleå
Bleser See Delikatessen	Forsterstr. 57	Munheim
Blondel y Blanco et fils	24, place Kléber	Strasbourg
Bonito Comidas preparadas	C/ Aragui3, 67	Madrid
Bon app'	12, rue des Bouchers	Marseille
Botten-Bollar Warkets	23 Torsvænns Blvd.	Torshavn

图 4.14

4.3 修改数据

用户不仅可能修改数据表格式，而且还可以修改表中的数据，包括插入新记录、修改数据和删除记录。

4.3.1 记录指示符

当用户在表的“数据表”视图中移动光标时，在表的最左端的“行选定器”中出现各种记录指示符。下面介绍这些指示符与它们的意义。

- 三角形——显示当前行，即现在操作的记录。
- 星花形——这个光标用来标示表的末端的空白记录，如果用户要建立新的记录，可以从这条记录开始。
- 铅笔形——这个光标用来提示用户，当前这条记录的内容正在修改。通常 Microsoft

Access 并不会马上修改真正的数据库的内容，等移动到另一条记录时，才会真正修改。所以在这个光标出现时，用户都有反悔的机会，只要按一下 Esc 键就可以把当前的字段值恢复到原来的值；接连两次按 Esc 键，是把所有字段的值恢复到原来的值。如果用户是在网络的环境操作，这个光标同时也代表这条记录已被“锁定”，在没有修改完毕前，别人无法修改这条记录。

● 锁形——此光标代表当前记录已被“锁定”，因此用户只能查看，而不能修改其内容。这个光标通常在网络环境下才会出现。

4.3.2 增加一个新的记录

在 Microsoft Access 中向表中增加新记录是十分简单的。当用户在设计完表后，第一次以“数据表”视图打开表，此时，表中没有任何记录，就需要用户向表中添加记录；还有当用户需要向表中增加记录时，也需要利用添加记录功能。

当用户以“数据表”视图打开表时，在表的最末端有一条空白记录，而其“行选定器”中有“*”型指示器，这表示用户可以从这里开始增加新的记录。

此时，用户只需单击工具栏上的“新记录”按钮，焦点转到空白记录行中，可以向第一个字段中输入数据，此时“行选定器”内的指示器变成“铅笔”形，表明正在输入数据。在输完第一字段后，按 Tab 键，或在其他字段列中单击，转到其他域中。

在输入数据时，Microsoft Access 自动检查输入值是否符合有效性规则，如果不符合，Microsoft Access 会将光标留在原处，并且显示提示信息，用户必须接着输入，直到输入值满足有效性规则为止。图 4.15 给出了由于在“产品：表”中增加新记录时，输入单价为负而引起的错误信息。

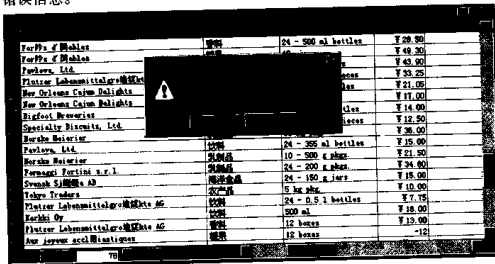


图 4.15

在输入完最后一个字段值后，再按 Tab 键，跳到下一条新记录。

4.3.3 选择和修改数据

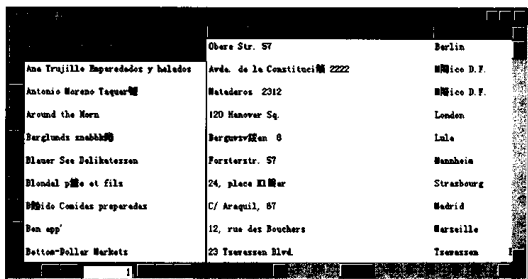
表中有数据之后，用户就可以很容易在“数据表”视图中编辑它，对数据进行修改。在

对记录进行修改之前，必须选择数据。

单击含有想更改数据的框，如果用户要替换整个字段的值，将鼠标光标移至框的最左边。当光标变成加号时，单击，整个数据框加黑，如果仅仅要修改几个字，可用键盘上的左移与右移键，将光标移到需要更改处。图 4.16 所示即选中了框中的全部内容。表 4.1 列出了用鼠标在“数据表”视图中选定数据或记录的技巧。

表 4.1 鼠标技巧

如果要选定	请单击
字段中的数据	选定开始处，并拖过数据
整个字段	数据表中字段的左边缘
相邻多个字段	字段左边缘，并拖过选定范围
一列	字段选定器
相邻多列	列项端的字段名，然后不要释放鼠标，拖过选定范围
一个记录	记录选定器
多个记录	第一个记录的记录选定器，然后拖过选定范围
所有记录	“编辑”菜单上的“选定所有记录”命令



	Obero Str. 57	Berlin
Ana Trujillo Emparedados y baldos	Arda. de la Constitucion 2222	Mexico D.F.
Antonio Moreno Taqueria	Mataderos 2312	Mexico D.F.
Around the Horn	120 Hanover Sq.	London
Berglunds snabbköp	Bergsgatan 8	Luleå
Blauser See Delikatessen	Fornsterstr. 57	Mannheim
Blondel y Söhne et fils	24, place Kléber	Strasbourg
Bólido Comidas preparadas	C/ Araquil, 67	Madrid
Bon app'	12, rue des Bouchers	Marseille
Bottou-Pollard Markets	23 Tsarassen Blvd.	Tsarsassen

图 4.16

表 4.2 列举了用键盘在“数据表”视图下选定数据式记录的技巧。

在选择好修改的对象之后，在框内输入新的数据。如果键入有错，可按 Backspace 键；如果要取消对当前字段的更改，按 Esc 键；如果要取消对整个记录的更改，按两次 Esc 键。在更改完一个字段后，Microsoft Access 并不立即保存，而是要等到由此记录移到另外一个记录时，才保存对该记录的修改。

表 4.2 键盘技巧

如果要选定	方 法
字段中的数据	将插入点移动到要选定文本的开始处, 然后按住 Shift 键不放, 并按箭头键直到选定的结束处
整个字段	将插入点移到字段中, 然后按 F2 键
相邻多个字段	选定一个字段后, 按住 Shift 键不放, 再按相应箭头键
当前列	按 Ctrl + 空格键
当前记录	按 Shift + 空格键
多个记录	按 Shift + 空格键, 然后按 Shift + 上箭头键或 Shift + 下箭头键

有些字段是不能修改的, 例如自动编码字段。自动编码型字段的值是由 Microsoft Access 在添加新记录时生成的, 是不可修改的, 可以被选中, 但如果要向其中输入数据却不会有任何作用。

4.3.4 替换数据

如果需要在—个以上的记录中做同样的改变, Microsoft Access 能使这种改变重复进行。

在表的“数据表”视图中, 选择要进行替换的字段, 单击该字段列, 然后在“编辑”菜单中单击“替换”命令, Microsoft Access 显示如图 4.17 的“在字段中替换”对话框。在对话框的标题中也显示了在哪个字段中进行替换。

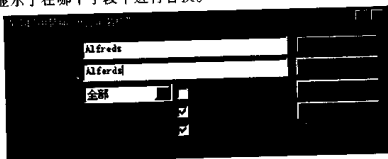


图 4.17

在“查找内容”框中输入要查找的内容, 然后在“替换为”框中输入要替换成的内容。如果不完全知道要查找的内容, 可以在“查找内容”框中使用通配符来指定要查找的内容。在“搜索”框中选择“向下”或“全部”作为方向; 再设置其他选项, 如区分大小写、全部字段匹配、只搜索当前字段。

如果要一次替换出现的全部内容, 则单击“全部替换”按钮; 如果要一次替换一个, 请单击“查找下一个”按钮; 如果要跳过下一个并继续查找出现的内容, 单击“查找下一个”按钮。

4.3.5 复制数据

Microsoft Access 也可以应用 Windows 的剪贴功能。

用户有时需要将一条或几条数据复制到其他表或窗体中。在表的“数据表”视图中, 选

中要复制的数据,选中方法见表 4.1、4.2。在“编辑”菜单中选中“复制”命令,或按 Ctrl + c 键,此时,这几条被选中的数据被复制到剪贴板上。

切换到要粘贴数据的对象中,选择粘贴的位置,单击“编辑”菜单中的“粘贴”命令,或按 Ctrl + v 键。在粘贴时,也许会有错误发生,一般有以下几个原因。

- 在没有选定目标的情况下,试图粘贴一个或多个字段。粘贴之前必须选择要粘贴到其上的字段。

- 粘贴的字段数多于要粘贴到的目标数据表的字段数,或者多于选中为目标的字段数。Microsoft Access 不会粘贴多出的字段。

- 粘贴的值与字段的数据类型不兼容。例如,将字母和数字的组合粘贴到 Number 数据类型的字段中。

- 粘贴的文本太长以至字段容纳不下。字段大小属性设置决定了字段的最大值。

- 试图将重复的值粘贴到表的主关键字或其索引属性设置为不重复的字段中。

- 试图将数据粘贴到隐藏字段中,这是不允许的。要显示隐藏的列,可单击“格式”菜单中的“撤消隐藏列”命令。

- 试图从中删除或粘贴数据到一个无效、锁定或自动编码字段。

- 用户不具有在数据库对象中添加或编辑数据的权限。

- 要粘贴的值在基于有效性规则、输入掩码、必填字段或允许空字符属性设置的字段中不允许使用。

- 在粘贴记录到数据表中时,Microsoft Access 按照数据表中的列的顺序放置它们,而不管字段的名称是什么,因此数据可能会粘贴到错误的字段上。

如果用户要移动一些数据,可利用剪切命令删除数据,再在要粘贴的位置重置数据。剪切数据会使数据从表格中消失,事实上与删除的功能相同,但效果略有不同。因为删除之后数据不见了,但剪切下来的数据仍存于剪贴板上,用户仍有反悔的机会,或转到另一表中。剪切操作是无法恢复的,因此 Microsoft Access 会出现消息框提醒用户。

4.3.6 删除记录

要删除记录,首先必须选择整条记录,用鼠标单击“行选定器”,整条记录将会呈现反白状态,表示用户已选择了该条记录。如果要选择多条记录,需按住鼠标左键,拖过要选的记录,再按 Del 键就可以了。这样删除后的数据就消失了,不能再复原了,因此,Microsoft Access 会显示一个消息框提醒用户,如图 4.18 所示。



图 4.18

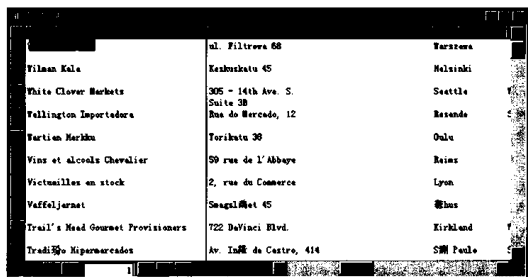
4.4 数据的排序与查寻

当打开表的“数据表”视图时，若用户已有主关键字，则 Microsoft Access 按输入的顺序排列。如果用户需要表以特定的顺序排列或查找某些数据，则只能利用 Microsoft Access 提供的排序与查找功能。

4.4.1 排 序

在前面为表建立索引中已介绍，建立索引就是为了进行排序。下面以“客户：表”为例，进行排序。

在“客户：表”的“数据表”视图中，在公司名称列的任何一个框中单击一下，然后按下工具栏的“升序”按钮，这时用户将会看到所有的记录都是按公司名称的字母顺序由小到大排列。当然，用户也可以按降序排列数据，只需单击工具栏中的降序按钮即可。图 4.19 所示即为按公司名称降序排列的表。



Vilaan Kala	ul. Filtreva 68	Varadero
White Clover Markets	Kenbuckatu 45	Helsinki
Wellington Importadora	305 - 14th Ave. S. Suite 3B	Seattle
Vertien Markku	Rua do Mercado, 12	Rosendo
Vins et alcools Chevalier	Torikatu 38	Oulu
Vietnamilles en stock	59 rue de l'Abbaye	Reims
Vaffeljernet	2, rue du Commerce	Lyon
Trail's Head Gourmet Provisioners	Sengulmet 45	Helsinki
Tradición Supermercados	722 DeVinci Blvd.	Kirkland
	Av. Inés de Castro, 414	São Paulo

图 4.19

对记录进行排序时，还有许多要注意的事项。

- 排序次序将与表、查询或窗体一起保存。如果某个新窗体或报表基于保存有排序次序的表或查询，则新窗体或报表将继承原有的排序次序。

- Microsoft Access 将在查询的结果或高级筛选中对多达 255 个字符的一个或多个字段进行排序。

- 排序次序取决于创建数据库时在“选项”对话框中的语言设置。如果要选中或更改此设置，请单击“工具”菜单上的“选项”，再单击“常规”标签来查看“新建数据库排序次序”下面的设置。

- 如果查询或筛选的设计网格包含了字段列表中的星号，则不能在设计网格中指定排序次序，除非在设计网格中也添加要排序的字段。

- 使用升序排序日期和时间，是指由较早的时间到较后的时间；使用降序排序时，则是

指由较后的时间到较早的时间。

- 在 Text 字段中保存的数字将作为字符串而不是数值来排序。因此，如果要以数值的顺序来排序，必须在较短的数字前面加上零，使得全部的文本字符串具有相同的长度。例如：要以升序来排序以下的文本字符串“1”、“2”、“11”和“22”，其结果将是“1”、“2”、“11”、“22”。对于不包含 Null 值的字段，另一个解决方案是使用 Val 函数来排序字符串的数值。例如：如果 Age 字段是包含数值的 Text 字段，在“字段”单元格指定 Val ([Age])，并且在“排序”单元格指定排序次序后，才会以正确的顺序来放置记录。如果只在 Text 字段之中保存数字或日期，可以考虑将表的数据类型更改为 Number、Currency 或 Date/Time。这样在对此字段排序时，数字或日期将会以数值或日期的顺序来排序，而不需要加入前面的零。

- 在以升序来排序字段时，任何含有空字段（包含 Null 值）的记录将列在列表中的第一条。如果字段中同时包含 Null 值和空字符串，包含 Null 值的字段将在第一条显示，紧接着是空字符串。

- 不能排序数据类型为 Memo、Hyperlink 或“OLE 对象”的字段。

4.4.2 查找数据

若用户想查找表中任何地方的数据，Microsoft Access 会提供有效的查寻和筛选性能。

为了在一个单一的字段上进行简单的查寻，先在表中选中那个字段，再按工具栏上的“查找”按钮，Microsoft Access 会弹出“在字段中查找”对话框，如图 4.20 所示。



图 4.20

在“查找内容”框中，用户需要键入要求 Microsoft Access 查找的数据。如果不完全知道要查找的内容，可以在“查找内容”框中使用通配符来指定要查找的内容。

在“搜索”框中选定搜索的方向：向上、向下、全部。在“匹配”框中选择匹配的条件，分为字段任何部分均符合、整个字段符合、字段开头符合三种。

还有其他诸如区分大小写、按格式搜索字段、只搜索当前字段各选择选项。根据需要设置这些选项。

如果要查找出现的第一个匹配内容，单击“查找第一个”按钮。如果要查找下一个和以后出现的内容，继续单击“查找下一个”按钮。

查找 Null 和空字符串与其他不同，下面详细介绍。

如果要查找未设置格式的空字段，请在“查找内容”框中键入 Null 或 IsNull，并确保“按格式搜索字段”复选框未被选中。如果空字段已设置了格式（如“Unknown”），请键入

设置了格式的字符串，并确保选中“按格式搜索字段”复选框。

如果要查找空字符串，请在“查找内容”框中键入不包含空格的双引号（""），并确保没有选中“按格式搜索字段”复选框。

在“搜索”框中，选择“向上”或“向下”。在“匹配”框中，确保选中“整个字段”，确保选中“只搜索当前字段”复选框。

如果要查找第一个空字段，请单击“查找第一个”按钮；如果要查找下一个或所有随后出现的空字段，请连续单击“查找下一个”按钮。

4.4.3 筛选数据

有时用户需要从原来的表记录集中筛选出一些符合要求的数据，组成一个新的集合，而不仅仅是查找。例如要在“客户：表”中找出国家是德国的公司名单，并进行打印，此时就需要将“客户：表”进行筛选。

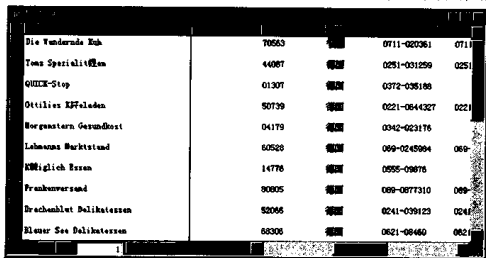
筛选方法有四种，分别为按窗体筛选、按选定内容筛选、输入筛选、高级筛选/排序。

按窗体筛选，是指使用字段值为空的当前窗体或数据表，在其中键入筛选的值，然后可以筛选出包含此值的记录。

按选定内容筛选，是指选定窗体或数据表中的值或部分值，据此找出包含此值的记录。内容排除筛选，则刚好相反，根据所给值，找出不包含该值的记录。

假设要筛选出国家为德国的记录，则只需在国家列中单击，然后选中国家为德国的记录，再在工具栏中单击“按选定内容筛选”按钮，屏幕上显示的是筛选出的满足条件的记录，如图 4.21 所示。图中国家列的值全是德国，还可以发现，工具栏中“应用筛选”按钮被按下，要回到筛选前的“客户：表”，只需单击工具栏中的“应用筛选”按钮即可。

输入筛选，是指使用“筛选目标”框中输入的值或表达式，以查找该指定值或表达式的所有记录。在表的“数据表”视图下，先要在要筛选的列中单击一下，选中该列，然后单击鼠



Die Vondernde Kuh	70563	德国	0711-020361	0711
Toms Spezialitäten	44087	德国	0251-031259	0251
QUICK-Stop	01307	德国	0372-036188	
Ottolien Pfladen	50739	德国	0221-064327	0221
Borgmestern Gesundheits	04179	德国	0342-023176	
Lehmans Marktstand	60520	德国	069-0245904	069
Königlich Essen	14776	德国	0555-09076	
Frankenversand	90805	德国	069-0877310	069
Drachenliet Delikatessen	52066	德国	0241-039123	0241
Kleiner See Delikatessen	68306	德国	0621-09460	0621

图 4.21

标右键，弹出快捷菜单，如图 4.22 所示。在图中可以看出，在快捷菜单上出现“筛选目标”框，在框中输入要筛选的文本，再按 Enter 键，屏幕上显示满足条件的记录。在“筛选目标”框中输入“德国”，就显示了图 4.21 的记录。

高级筛选/排序功能最为强大，可以应用于一个或多个字段进行排序和筛选。应用“高级筛选排序”进行筛选记录，步骤如下：

在“记录”菜单上单击“筛选”，然后单击“高级筛选/排序”，Microsoft Access 弹出如图 4.23 所示的“筛选”窗口。

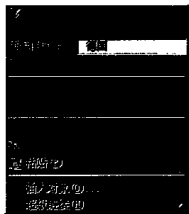


图 4.22

用户需要将用于筛选记录的值或准则的字段添加到设计网格中。在图中可以看出，Microsoft Access 已自动将所要筛选的表显示在窗体中，并且已列出其字段。在此窗体中，用户不能添加其他表或查询。用户只需从字段列表选定一个或多个字段，并将其拖曳到网格的列中，然后在字段的“排序”单元格中单击箭头，选择相应的排序次序。

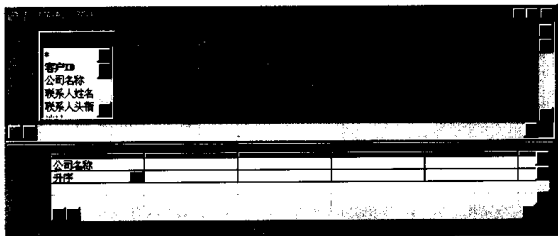


图 4.23

用户要是仅仅是排序，则不必在“准则”单元格中输入数据。若要筛选数据，用户需在“准则”单元中输入需要查找的值或表达式。在完成排序或筛选的规则后，单击工具栏上的“应用筛选”按钮以执行筛选。在执行时，Microsoft Access 会首先排序设计在网格最左边的字段，然后排序该字段右边的字段，以此类推。

图 4.24 给出了利用国家城市公司索引进行排序的“高级筛选/排序”窗体，图 4.25 给出了排序后的结果。

图 4.26 给出了利用国家城市公司索引进行筛选的“高级筛选/排序”窗体，图 4.27 给出了筛选的记录。

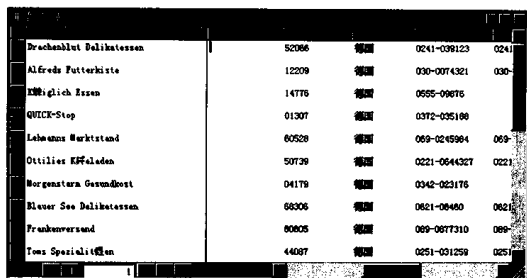
最后，表 4.3 给出了四种筛选方法的比较。

图 4.24

Bancho grande	1010	阿根廷	(1) 123-5555	(1)
Océano Atlántico Ltda.	1010	阿根廷	(1) 135-5333	(1)
Cactus Comides para llevar	1010	阿根廷	(1) 135-5555	(1)
Hungry Owl All-Night Grocers	Co. Cork	爱尔兰	2967 542	2967
Ernst Handel	8010	奥地利	7675-3425	7675
Piccolo und mehr	9020	奥地利	6582-9722	6582
Gourmet Lanchonetes	SP	巴西	(11) 555-0402	
Vellington Importadora	SP	巴西	(14) 555-6122	
Ricardo Adocicados	RJ	巴西	(21) 555-3412	
Que Pasa	RJ	巴西	(21) 555-4252	(21)

图 4.25

图 4.26



Drachenblut Delikatessen	52066	德国	0241-039123	0241
Alfreds Futterkiste	12209	德国	030-0074321	030
Königlich Essen	14776	德国	0555-09876	
QUICK-Stop	01307	德国	0372-035180	
Lehmanns Marktstand	60528	德国	069-0245984	069
Ottilies Käseladen	50739	德国	0221-064327	0221
Morgenstern Gesundkost	04179	德国	0342-023176	
Bleuer See Delikatessen	68306	德国	0621-06460	0621
Frankenversand	80805	德国	089-0677310	089
Toms Spezialitäten	44087	德国	0251-031259	0251

图 4.27

表 4.3 筛选方法比较

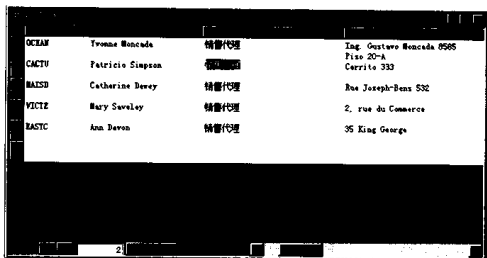
筛选目的	按选定内容筛选	按窗体筛选和 输入筛选目标	高级筛选/排序
搜索符合多个准则的记录	是（但是必须一次指定一个准则）	是（并且可以一次指定所有准则）	是（并且可以一次指定所有准则）
搜索符合一个准则或另一准则的记录	否	是	是
允许输入表达式作为准则	否	是	是
按升序或降序排序记录	否（但是，在应用筛选后，可以单击工具栏上的“升序”按钮或“降序”按钮来排序所筛选的记录）	否（但是，在应用筛选后，可以单击工具栏上的“升序”按钮或“降序”按钮来排序所筛选的记录）	是（并且可以对某些字段按升序排序，而对其他字段则按降序排序）

4.5 打印数据

用户可以利用“数据表”视图来打印表中的信息。由于有筛选与排序功能，用户可以自由地选取自己关心的数据，并且可以利用打印功能进行打印。

在打印之前最好去掉网格线，并且适当增大行高与列宽，以免在打印时显得过分拥挤。图 4.28 中显示了所要打印的经过筛选后的“客户”表。

打印可分几个步骤进行。



OCEAN	Tironne Boncada	销售代理	Ing. Gustavo Boncada 0505
CACTU	Patricio Simpson	销售代理	Piso 20-A Cerrito 333
BAISD	Catherine Dorey	销售代理	Rue Joseph-Benz 532
VICTZ	Mary Sevelay	销售代理	2. rue du Commerce
EASTC	Ann Devon	销售代理	35 King George

图 4.28

4.5.1 打印预览

在进行正式输入到打印机中进行打印之前，最好先利用打印预览功能观察打印出来后的样式，不满意可修改，避免不必要的损失。

要预览表，只需单击工具栏中的打印“预览”按钮，便可以看到图 4.29 的显示。此时鼠标指针变成一个放大镜状。用户可移动鼠标指针到报表的任何地方，并单击一下，便可以放大察看数据；再单击，则回到预览状态。

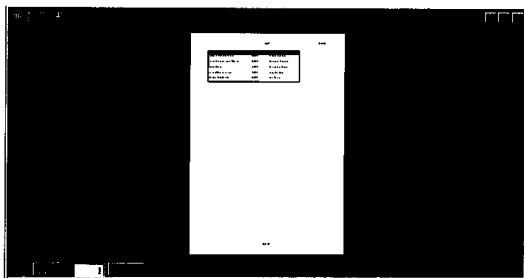


图 4.29

如果打印输出百多张纸，缩小画面要在页之间移动，用户可用 PgUp 与 PgDn 键。单击“关闭”按钮即可退出打印预览。单击“打印”按钮，把格式化数据打印输出。单击页面设置来指定页面及打印机的设置。用户也可以将此表输入到 Excel 和 Word 中进行发布。

4.5.2 打印设置

在“文件”菜单中单击“打印”命令，Microsoft Access 显示如图 4.30 的“打印”对话框。由图中可以看，在“打印”窗口中可以设置打印机、打印到文件或直接输出到打印机、打印范围、份数等，还可对页面进行设置。单击“设置”按钮，Microsoft Access 显示“页面设置”对话框，主要设置页边距等，如图 4.31 所示。

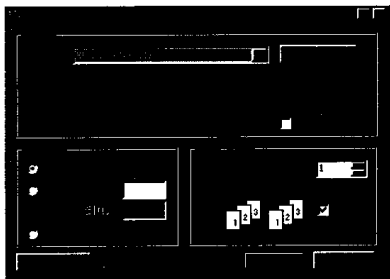


图 4.30

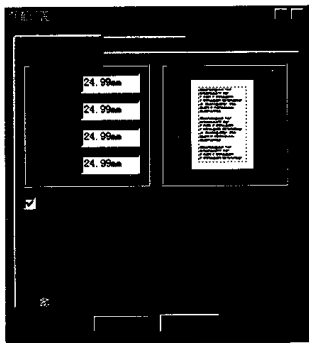


图 4.31

4.5.3 打 印

单击“确定”按钮，Microsoft Access 会显示如图 4.32 的 Microsoft Access 信息提示框，此时要是不打算打印，可立即按“取消”按钮。

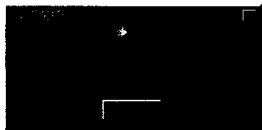


图 4.32

第五章 选择查询

本章，将介绍 Microsoft Access 的第二种对象——查询对象。虽然在前面几章中，已介绍了用表的“数据表”视图来操作数据，但是在大多数情况下，用户不需要同时看到表里所有字段的数据；而有时，用户却需要看到不同表内的某些字段的数据。在这种情况下，Microsoft Access 提供的查询对象就十分有用。

查询实际上就是从众多表中收集用户认为有用的字段的工具，收集到一起的数据组成一个“动态集合”。这个集合中的字段可能来自同一个表，也可能来自不同的多个表，但是，给使用者的感觉似乎它们本来就集中在一起，就像一个表一样。实际上，由查询产生的数据集合与表并无本质区别，用户同样可以查看、浏览、查询、打印，甚至也可以修改数据，并且这些修改会直接反映到表中，因为 Microsoft Access 会自动把任何修改反映到每个对应的表中。

与真正的表不同的是，由查询产生的动态集合并不真正存在于数据库内，只有运行这个查询时才会建立；当结束此查询时，Microsoft Access 会自动把任何值的修改反映到相应的表中，然后删除此动态集合。

查询可分为两种：一种是选择查询，是用来从不同的表内挑选字段，供用户查看、浏览、打印、查询等动作的；另一种叫操作查询，是专门用来修改数据用的，包括插入、删除、修改等动作。

5.1 查询的“设计”视图

查询有三种视图：“设计”视图、“数据表”视图与 SQL 视图。其中“设计”视图是在设计查询时用的，本节将介绍该视图的画面以及工具栏中的各按钮的作用。

5.1.1 “设计”视图

要打开查询的“设计”视图，只要在“数据库”窗口中按下“查询”按钮，如图 5.1 所示，在窗口的列表框中列出了数据库中所有的查询名。

用鼠标单击窗口内的查询对象，再单击“设计”按钮，Microsoft Access 就会在“设计”视图下打开“选择查询”，如图 5.2 所示。也许读者会发现，图 5.2 的窗体在以前出现过，事实上，此窗体与上一章中“高级筛选/排序”窗口相同，只是用途不同而已。

查询的“设计”视图分为上下两个部分。上半部分是用来显示表的字段列表框的，每个列表框均代表一个独立的表，里面所列的就是该表所有的字段名称。如果在此间显示的表之间有相互关系的话，Microsoft Access 会自动地将这些表联接起来，与关系窗口一样。

下半部分是查询设计网格，它十分类似于表。这部分的作用十分类似于设计表时的设计

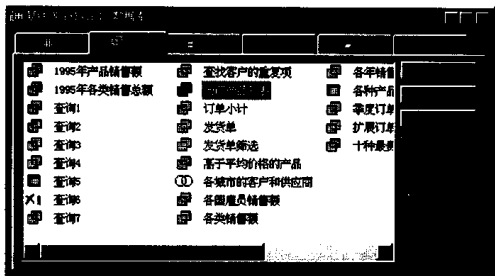


图 5.1

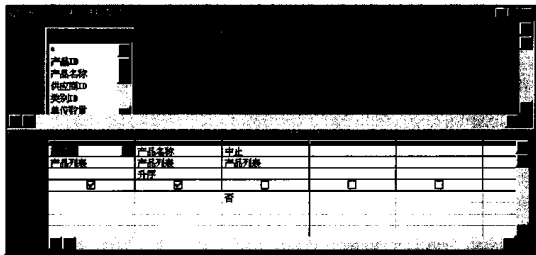


图 5.2

画面。查询设计网格的每一列均代表查询中的一个字段，查询里的域并不仅仅是在上半部分的字段列表中列出的字段，而且还可以是由多个表字段组合计算出的一个计算域，因为它并不实际存在于数据库中，因此又称为虚拟域。在每列的上方，都有一个灰色的小条，那是“列选定器”，通常在选中某一字段或改变设计视图中的列的宽度时，都需要用到它。

查询设计网格的第一行是用来设置字段的，在这里被设置的字段会加入到动态集合中。

网格的第二行是表列，在此行中显示了第一行中的字段的来源。由于在不同的表中可能含有相同名称的字段，为清晰起见，在第二行中显示表名。

第三行是用来设置这个字段需不需要经过排序，排序的方向有两种：升序排列与降序排列。如果用户不想对此字段进行排序，保存此行空白即可。

第四行是由选择框构成，选择是否要在“数据表”视图中显示相应的字段。Microsoft Access 会自动显示查询设计网格中的每一个字段，但有时用户却希望某些字段不在数据表中显示，只是由于它在设计查询时具有特殊的功能而引入的，比如某个客户的 ID 号，只是由

于在作比较时方便一些，而在数据表中没有作用，因此可将此字段的显示选择框清除，以通知 Microsoft Access 将该域从动态集合中删除。

网格的最后几行是准则，它是用来建立检查条件的，就像筛选数据时的输入值一样，只有符合准则的数据才能进入动态集中。准则可以不只一个。在准则下面有“或”行，在这行中可以输入第二个准则。当然下面还有许多“或”行，可以输入准则。

5.1.2 “设计”视图的工具栏

当进入“设计”视图时，屏幕上方的工具栏内会出现一些“设计”视图专用的工具，如图 5.3 所示。



图 5.3

- 视图按钮——由于当前视图是“设计”视图，单击此按钮，查询由“设计”视图转换成“数据表”视图。如果单击此按钮的箭头，则此按钮下拉，显示出查询的三种视图，选中其中一种视图，则查询就转换成那种视图。

- “数据表”视图按钮——单击此按钮，查询在“数据表”视图下显示。

- 保存按钮——按下此按钮，保存对查询所做的任何修改。

- 执行按钮——单击此按钮，Microsoft Access 执行当前查询，并以“数据表”视图显示动态集。

- 显示表按钮——此按钮是用来向查询窗口中加入表而设计的。单击此按钮，Microsoft Access 会弹出“显示表”对话框，用户在对话框中所选中所要加入的表，Microsoft Access 会在“设计”视图中显示此表的字段列表。

- 总计按钮——单击此按钮，可以打开查询设计网格中的总计行，用户可建立一个查询来计算所有组数据的总和。

- 上限值按钮——这是用来约定显示查询结果的。用户可在框内输入所要显示的范围，例如只显示产品名称的前 10 个符合上限值的记录。

- 属性按钮——单击此按钮，Microsoft Access 将根据光标位置的不同而显示不同对象的属性。光标焦点在查询设计窗体中，则显示查询属性；位于字段列表中，则显示字段列表属性；位于字段中，显示字段属性。

- 生成器按钮——单击此按钮，Microsoft Access 弹出表达式生成器，用于生成表达式。

- 数据库窗体按钮——单击此按钮，Microsoft Access 将焦点置于数据库窗体，并显示它。

5.1.3 查询属性设置

在进行设计查询之前，最好选设计查询属性。

单击工具栏中的属性按钮，此时应将焦点置于查询窗体中，或者当鼠标指针移至“设计”视图的标题栏时，单击鼠标右键，在快捷菜单中选择“属性”命令，如图 5.4 所示，然后 Microsoft Access 会弹出如图 5.5 的查询属性窗口。

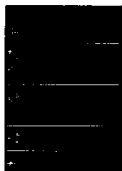


图 5.4

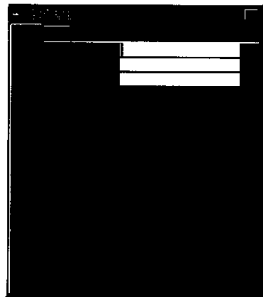


图 5.5

可以在属性窗口中设置以下属性。

- 说明——输入该查询的说明。
- 输出所有字段——选择“是”，则输出所有在网格中的字段；若选择“否”，则有的字段不显示。
- 上限值——与工具栏中的上限值按钮一样，它的功能将在查询设计中详细叙述。
- 唯一值——值是否只是唯一的。
- 唯一的记录——记录是否唯一，即是否每条记录均不相同。选择“是”，是重复记录不显示，若选择“否”，则显示所有的重复记录。
- 执行权限——指定该查询由谁来执行与使用，是所有者或用户。
- 源数据库——输入表和查询的源数据库名称，缺省值为当前数据库。
- 来源连接字符串——输入连接字符串的源数据库名称。
- 记录锁定——是否及如何锁定基础表或查询中的记录，选项有不锁定、锁定所有记录、锁定编辑记录。
- 记录集类型——决定哪些表中的数据可以编辑。只有选择动态集时，查询中的数据与表中的数据一起更新。快照型是不可更新表中的数据，动态集中的数据只是表中的一个像，不能更新。
- ODBC 超时——如果服务器没有响应，在多少时间内取消 ODBC 查询。空则为无时间限制。

- 筛选——和查询一起加载的筛选。
- 排序依据——和“查询”一起自动加载的排序次序。
- 最大记录数据——从 ODBC 数据源返回的最大记录数。

5.2 创建选择查询

创建新查询与创建其他 Microsoft Access 应用对象一样，也必须从“数据库”窗口开始。因此必须让“数据库”窗口为当前窗口的状态，只要单击工具栏上的“数据库”窗口按钮或按 F11 键即可。

单击“查询”按钮，在再单击“新建”按钮，则 Microsoft Access 显示如图 5.6 所示的“新建查询”窗口，在此窗口中提供了五种创建查询的方法：在“设计视图”下创建新查询，是不利用任何向导，自行创建查询；“简单查询向导”将利用用户挑选出的字段创建一个查询；“交叉表查询向导”创建的交叉表用压缩的、类似电子表格的格式显示数据；“查找重复项查询向导”是用来创建那种利用位于单个表或查询中的重复字段值查找记录的查询；“查找不匹配项查询向导”创建的查询，是在一个表与其他表没有关联记录的独立表中查找记录。

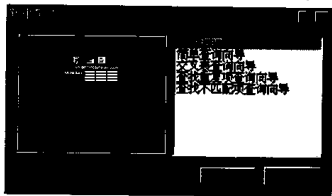


图 5.6

这里，先利用“设计视图”设计查询，选中“设计视图”，然后单击确定按钮，Microsoft Access 将显示查询的“设计视图”。由于是新建查询，因此视图是空的，并且 Microsoft Access 同时还弹出了“显示表”对话框，如图 5.7 所示。

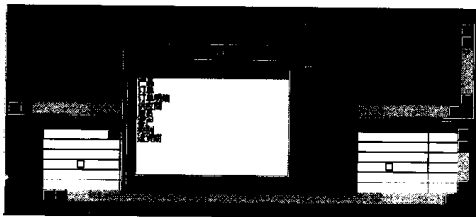


图 5.7

5.2.1 添加表/查询

在图 5.7 所示的“显示表”对话框中添加表或查询是十分容易的。

首先单击所要选择的对象种类名称。选择表，则按“表”标签；选择查询，则按“查询”标签；若按“表和查询”标签，则列表框中显示所有表与查询的名称。再双击所要选择的对象名称，则此表或查询就会在“设计视图”中显示它的字段列表。在选择完所需的表/查询之后，单击“关闭”按钮，关闭“显示表”对话框。如果在关闭了“显示表”对话框之后，还要添加表，用户只需单击工具栏中的“显示表”按钮，屏幕上会显示出“显示表”对话框。

5.2.2 更改联接

如果在查询中有多个表或查询，用户必须确定它们是如何相互联接的，这样 Microsoft Access 才能知道信息是如何相关的。如果它们没有相连，用户必须自行联接，方法可见第三章中创建关系一节。其视图如图 5.8 所示。

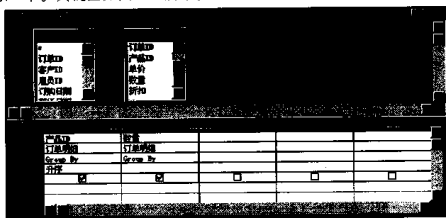


图 5.8

联接的类型对查询所选择的记录影响很大。可以根据用户的不同需要来更改联接类型。

双击视图中表或查询字段列表之间的联接线，Microsoft Access 会弹出“联接属性”对话框，如图 5.9 所示。一共有三种联接类型。

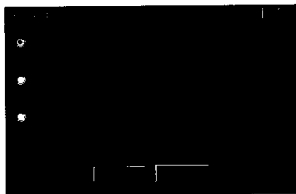


图 5.9

● 只包含来自两个表的联接字段相等处的行。此种联接类型下的查询只选择两个表的相连字段值相等的那些记录，这种类型被设为默认类型，而且一般也是要在这种类型下的查询。

- 包括所有“客户”的记录和那些联接字段相等的“订单”的记录。
- 包括所有“订单”的记录和那些联接字段相等的“客户”的记录。

在这两种类型下的查询，将从特定表中选择所有记录，而不管其他表是否具有相匹配的记录，这就势必导致在动态集的某些字段的值为空。

所有使用“联接属性”对话框创建的联接，只有在联接字段的值相等时，才会自动选取记录。要创建只有在在一个联接字段大于、小于、不等于、大于等于、小于等于另一个联接字段时，才选其中的记录，则放在 SQL 视图中创建一个 SQL 语句。SQL 语句在本书中不作介绍，若需要可参考有关书籍。

5.2.3 删除表

从“设计”视图中删除表十分简单：单击所要删除的表，再按 Del 键或单击鼠标右键，在显示的快捷菜单中选择“删除表”。

5.2.4 添加字段

建立查询的第二步是在字段列表选取需要的字段，并且填入查询设计网格中。

共有三种方法可以从字段列表向查询设计网格中加入字段，这样字段就被纳入查询的结果中。

● 单击字段列表中的字段名称，按住鼠标按钮，将字段名拖动至网格内的一个字段单元内。图 5.10 所示的是将产品表中的产品 ID 字段拖动到网格中，此时鼠标的指针会变成一个矩形。如果该单元内已有一个字段名，新的字段将替换现有的字段，而现有的字段将右移一个单元。

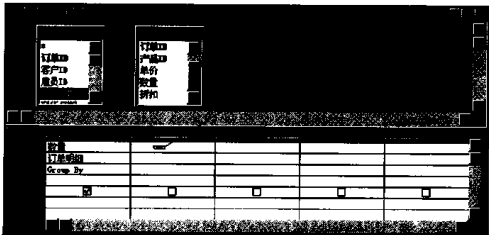


图 5.10

● 单击网格内的一个字段单元以显示下拉列表，它列出了查询窗口中的所有字段列表的字段，选择一个并放入该单元内。

● 单击网格内的一个字段单元，键入字段名。

一、增加多个字段

有时需要将所有的字段纳入到查询的结果中。比如，用户可能想用查询分离出特定的记录，但又想要所有的字段，基于这种情况 Microsoft Access 提供了向网格中加入所有字段的简易途径。

用户可双击字段列表的标题条，选中字段列表中的所有字段，然后从字段列表中单击并拖动任意字段放入字段单元内，这样所有的字段都将放入字段单元内。

用户也可以利用以上方法将任意多个字段放入网格中。可以有两种方法来选择多个字段：一种是按住 Ctrl 键，用户再单击各个想要选择的字段；另一种方法是单击字段，按住 Shift 键，并单击第二个字段，这样这两个字段间的所有字段都被选中。如果要选择的字段是相互连接在一起的，第二种方法最方便。

在字段列表中，有一行是 *（星号），它代表了此列表中的所有字段。如果将此星号拖到网格的某一字段单元中，网格中的那一单元显示后跟星号的表名，如图 5.11 所示。利用星号与前一种方法相比有所不同。有一种方法选取表中所有字段，不能随表结构的变化而变化；而使用星号来增加所有字段，即使表中增加了新的字段，新增字段还是包含在动态集中。

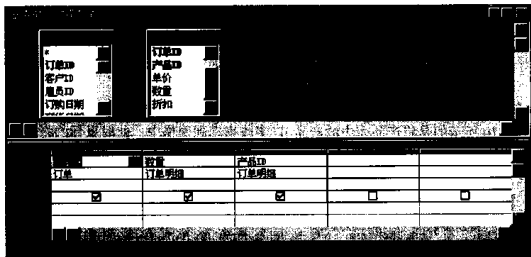


图 5.11

虽然星号对于选取所有字段的操作有方便之处，但对于排序、加判断条件等操作增加了麻烦。

二、插入、删除、移动字段

查询动态集中的字段排列顺序是按照它们在查询设计网格中的顺序来显示的。

要将一个新字段插入网格中的任意位置是十分容易的，只要将该字段从字段列表中拖到网格中的相应位置，在此位置的字段以及它右面的所有字段都将右移。

要移动网格中的字段，只要先用鼠标单击字段选择框选中该行，然后再单击字段选择框移到想要的位置。图 5.12 所示即为在查询设计网格中选中一个字段。

删除字段时，用户先单击字段选择框选中此行，然后再按 Del 键删除，此行右边的所有列相应左移。

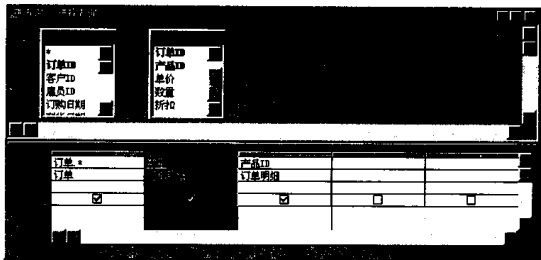


图 5.12

5.2.5 设置字段特性

一般情况下，查询字段继承了在表中定义的字段特性，但用户还可在查询中对字段属性作进一步的定义。

将焦点置于字段列中，单击工具栏中的属性按钮，Microsoft Access 将显示图 5.13 所示的“字段属性”对话框。在此对话框中，用户可为表设置字段说明、格式、输入掩码、标题等属性。设置这些属性并不影响表中字段的属性，而只是在动态集中显示数据有用。

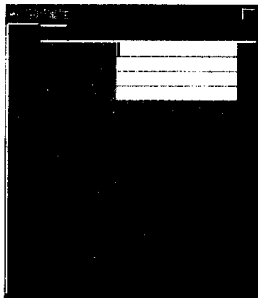


图 5.13

5.2.6 排序次序

在查询设计网格第三行是排序选项。要对查询中的某一字段进行排序，只需单击排序行中对应的该字段，在此网格中会显示出一个箭头按钮，单击它，下拉列表中有升序与降序两种，选择相应的项。

实际上，有三种选择。

- 升序——对字段中的值按从低到高的次序对记录进行排序。在文本字段中，中文按笔画进行排列，英文按字母顺序排列；在数字字段中，排序按数字从小到大次序进行；在日期字段中，排序则是从早到晚排列的。

- 降序——按值由高到低排序。

- 不排序——不对字段进行排序。这是 Microsoft Access 缺省的情况，选择不排序时，排序行空白。

用户有时需要建立多个字段的排序。Microsoft Access 在对多个字段进行排序时，使用最左边的字段作为排序基础；当最左边的字段相同时，则使用它右边的字段来作为决定排序位置的依据。因此在对多字段进行排序时，应注意字段的位置，要将最优先排序的字段放在左边。

5.2.7 显示字段

第四行是显示选择行，如果要使某一字段在动态集中不显示，则单击显示选择框，使之不再显示“×”。Microsoft Access 的默认值是显示网格中的每一个字段。这个选择对某些时候由于不想显示一些字段而这些字段又必须加入到查询中是十分有用的。

5.2.8 输入准则

向一个查询中输入准则与在设计表的设计时向字段中加入有效性规则相似。为了寻找单个值，只要简单地把它键入到约束的字段下面的准则行中就可以了。如果想检验的字段的数据类型是文本型，并且想要查寻的值在中间有空格，用户必须把每一个组放在引号中。如果用户不加引号，则 Microsoft Access 会自动在文本上加上引号。

如果想检验任何几个值的存在，只要在准则行中输入每一个值，它们会被“或”分隔开。用户可以在准则行中输入每个想要寻找的值，按 Enter 键。如果准则行中已有数据，它将自动下移至“或”行；也可以在第一个空行中输入想要寻找的值，如图 5.14 所示。

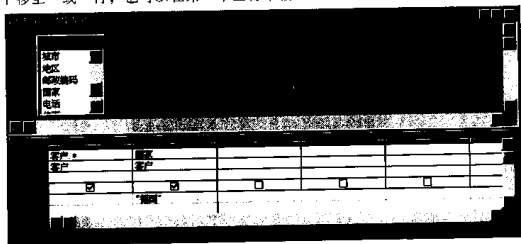


图 5.14

在查询的准则行中，也可以用比较操作符来寻找大于、小于、等于或不等于用户输入的值。

一、“与”和“或”

在设计查询时，用户必然会碰到使用多个条件进行查询的情况，此时就要利用到“与”和“或”逻辑操作。

1. 逻辑“与”

当用户需要查找多于一个字段中的值的记录时，只要简单地在准则行中对应的字段下面输入相应的条件即可。Microsoft Access 将这当作一个逻辑“与”，在查询动态集中得到的记录是满足所有这些准则的。例如要查找住在伦敦且又是销售代表的雇员时，应使用图 5.15 所示的查询，Microsoft Access 将只分离对两个条件均符合的记录。

姓氏	名字	头衔	城市	雇员
☒	☒	☒	销售代表	
			London	

图 5.15

2. “与”操作符

“与”操作符即为 and。如果用户要为单一字段输入两个或两个以上的条件，就必须使用 and 操作符。例如要查找雇用日期在 1993 年 4 月 1 日与 1994 年 4 月 1 日之间的所有雇员，应使用图 5.16 中的查询，此时准则应写为 >#93-4-1# and <=#94-4-1#。

头衔	职位名称	出生日期	雇用日期	地址
☒	☒	☒	>#93-4-1# and <=#94-4-1#	
			销售代表	London

图 5.16

3. “或”操作符

“或”操作符 or 可以实现在一个字段中查找多个值。如图 5.17 中的两种方式均可以实现“或”操作。只是使用 or 操作符比在每一行中输入所有的数据要简单。

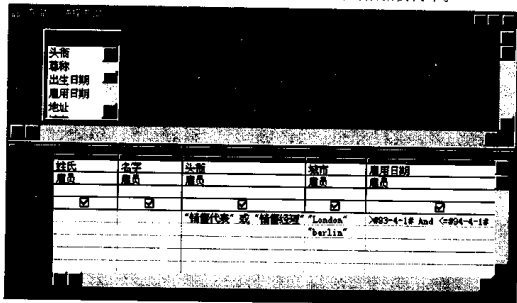


图 5.17

二、between、in 和 Like

除了大于、小于、等于、大于等于、小于等于、不等于等比较操作符以外，Microsoft Access 还提供了三个特殊的指示从句，这对于用户在记录集中指定所要的数据非常有用。

- between——用于指定一个值的范围。它必须与 and 一起作用，between...and 的功能与 > and < = 是一样的。例如，between 10 and 1000 等同于 >10 and < = 1000。

- in——用于指定某一值的列表，其中任一值都与查找的字段匹配。它与 or 操作符有相同的功能，但比 or 要简洁得多。例如 In (“美国”，“英国”，“德国”) 与 “美国” or “英国” or “德国” 是等效的。

- Like——用于查找文本字段中的模式。可以在 Like 比较字符串中包含某些字符和范围值来定义用户所需的字符模型。用 ? 表示该位置的任意字符；用 * 符表示该位置中多个字符；字符 # 指定了该位置的单个数字位；在括号中包括一个范围，来检验位置的字符范围；用 ! 表示例外。[0~9] 检验数字，[a~z] 检验字母，而 ![0~9] 表示检验该位置的字符是否是除 0~9 以外的任何字符。例如，式子 Like “?[a~k] d [0~9] *” 检验数据的第一个位置为任意一字符，第二位置是从 a 到 k 的字符，第三个位置字符 d，第四个字符是 0 到 9 的数字，其他位置为任何字符。

假设用户想在客户表中显示名字以 A 开头并且在 Berlin 或 London 的所有公司，图 5.18 中显示了“设计”视图。

三、操作标准中的日期和时间

Microsoft Access 把日期和时间存储成双精度浮点数。小数点左边的值代表天，小数部分存储一天的时间分数。用户不必关心日期与时间在内部的存储格式，因为 Microsoft Access 能自动将用户的输入转换成相应的数值。

图 5.18

在准则中输入日期时，用户必须在日期值两边加上 # 符号。日期有多种表示法，用户要选择最适合的日期表示法。例如，# May 20, 1997 #、# 5/20/97 # 和 # 20 - May - 1997 #。当用户在 Windows 的控制面板的国际部分中把国家设置为 U.S.，Microsoft Access 把前面的日期认为同一日期，当然 # 5: 30PM # 与 # 17: 30 # 两者均指下午 5: 30。

Microsoft Access 中拥有许多内部函数来帮助用户检验日期和时间值。

- Day (date) 返回 date 中的日期号，| - 31 |。例如，如果 date = # May 20, 1997 #，则 Day (date) = 20。想查询雇员表中雇佣日期是 10 日以后的雇员，只需在雇佣日期的准则中输入 Day ([雇佣日期]) > 10。

- Month (date) 返回 date 中的月号，| - 1 |。在上例中，Month (date) = 5。如果要查询雇佣日期是 4 月以后的雇员的名单，在雇佣日期的准则中输入 Month ([雇佣日期]) > 4。

- Year (date) 返回 date 中的年份。

- Weekday (date) 返回一个整数，代表某个日期是星期几。

- Hour (date) 此时 date 是一个时间，返回代表 date 是几点的整数，范围在 1 ~ 23 之间。

- Date 返回系统的当前时间。若查询 30 天前以后雇佣的雇员名单，用户需在雇佣日期字段的准则中输入 > Date () - 30。

- Now 返回当前系统的时间，包括日期信息。

5.2.9 计算值

用户可以使用表中任何字段的值作为计算的对象。由于用户不仅需要原来在表中存储的字段值，而且更需要存放在表中的隐含信息。例如，表中有产品的单价和产品的数量，用户可以由这两个字段的值导出该产品的总价值。如果用户需要在查询中显示这些导出信息，那么用户就必须在查询中建立一个新的字段。这个字段不同于其他一般的字段，它是由在表中的字段经过各种计算之后形成的。这些计算既包括简单的加、减、乘、除等算术运算，还包括由 Microsoft Access 提供的众多的内部函数，如 AND、OR、XOR、NOT 等逻辑运算。

在创建这种字段时，不再是简单地拖动字段列表中的字段，而是要在字段行中输入表达

式，在表达式中要体现所要进行的计算。与这些计算有关的操作符如下：

- + 两个数字表达式相加；
- - 两个数字表达式值相减；
- * 两个数字表达式相乘；
- / 第一个数字表达式被第二个数字表达式除；
- \ 把两个数字表达式四舍五入成整数并且第一个数被第二个数除，结果也四舍五入成整数；

- ^ 符号前的数为底，其后的数为幂值数；

- MOD 把两个数字表达式都四舍五入为整数，相除，然后返回余数；

● & 把第一个文本字符串连接到第二个文本字符串上来建立一个扩展文本字符串，如果两个表达式都是数字，在连接之前，Microsoft Access 把它们转换成一个文本字符串。

以雇员表为基础，创建查询，要将雇员的姓名连接在一起，就需要建立一个新字段。在字段行中的第一列中单击，在此输入表达式。由于要将姓氏字段与名字字段的文字相连，需要用到 & 连接符。表达式：[姓氏] & "，" [名字]，该实例的“设计”视图如图 5.19 所示，“数据表”视图的结果如图 5.20 所示。

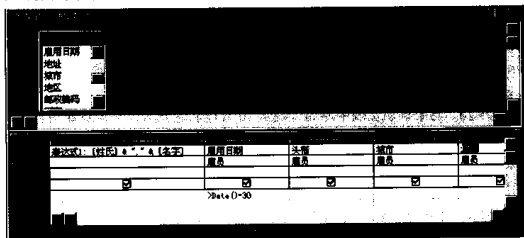


图 5.19

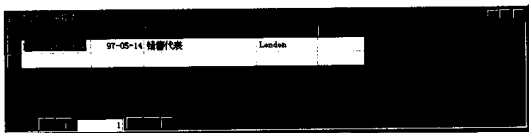


图 5.20

5.2.10 使用表达式生成器

对于更复杂的表达式，Microsoft Access 提供了表达式生成器。假设在订单明细表中建立查询，计算订单的总计值。打开一个新查询的“设计”视图，在显示表对话框中双击订单明细。

单击在查询设计网格上的任一空字段，然后单击工具栏上的“生成器”按钮，Microsoft

而规定的,在表名与字段名间用!号分隔开。

下一步应该来算订单中该产品的数量。先单击乘号(*)按钮,此时表达式框中的表达式变为[订单明细]![单价]*。在输入时,如果出错,则用户只需单击“撤消”按钮删除最后一次输入即可。然后再双击字段列表中的数量字段,此时,[订单明细]![数量]出现在表达式框中,并被设置在*号之后。

应将该再减去折扣部分。计算总价的表达式应为单价×数量×(1-折扣)。单击左括号按钮,输入1,再单击减按钮,双击字段列表中的折扣为字段,再单击右括号按钮,完成表达式。但由于此为货币值,最好使用货币型显示,因此需要使用函数对其值进行强制转换。单击“表达式生成器”左下方框中函数折叠卡,下拉显示两种函数类型,单击内部函数,在中间的框中显示了内部函数的种类,并在右框中显示其全部函数。缺省情况下是显示全部函数。在右框中选中 CCur 函数,双击之,则 CCur 将在表达式中显示。

最后完成的表达式如图 5.23 所示。

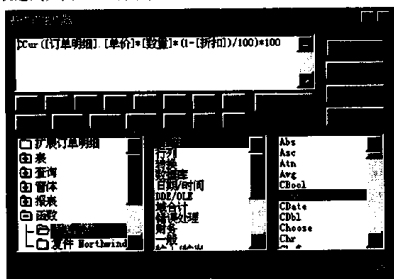


图 5.23

再单击“确定”按钮,将结果拼接到网格中。当用户以“数据表”视图打开该查询时,显示如图 5.24 所示的数据表。

	Singaporean Kikkian Fried Bee	\$9.80	10	0%	¥ 98.00
10248	Bocconcina di Giovanni	\$34.80	5	0%	¥ 174.00
10248	Alice Mutton	\$14.00	12	0%	¥ 168.00
10249	Tofu	\$18.00	9	0%	¥ 167.40
10249	Hanjiang Dried Apples	\$42.40	40	0%	¥ 1,696.00
10250	Louisiana Fiery Hot Pepper Sauce	\$16.80	15	15%	¥ 214.20
10250	Jack's New England Clam Chowder	\$7.70	10	0%	¥ 77.00
10250	Hanjiang Dried Apples	\$42.40	35	15%	¥ 1,281.40
10251	Louisiana Fiery Hot Pepper Sauce	\$16.80	20	0%	¥ 336.00
10251	Gustaf's Knäckebröd	\$16.80	6	5%	¥ 95.76
10251	Baroli Angelo	\$15.60	15	5%	¥ 222.30
10252	Sir Rodney's Marmalade	\$64.80	40	5%	¥ 2,482.40
10252	Guitoni	\$2.00	25	5%	¥ 47.50
10252	Camembert Pierrot	\$27.20	40	0%	¥ 1,088.00
10253	Bacalao	\$16.00	40	0%	¥ 640.00

图 5.24

5.2.11 指定字段名

在图 5.24 中,用户会发现,在数据表的最后一列,标题名是“表达式 1”,这显然不符合要求。用户当然希望能使数据表中看到的字段标题具有意义,Microsoft Access 提供了这种功能。

在设计查询时,在表达式前面,有“表达式 1:”。查询中每个字段都必须有字段名,字段的缺省值标题就是字段名。前式中的“表达式 1”即为此字段名,用户可以直接用其他文本代替它,而新输入的文本就是该字段的新名称。

对于其他字段,用户可在字段属性列表框中进行更改。要打开字段属性对话框,只需单击相应字段,再单击工具栏中的属性按钮。

5.2.12 合计查询

有时,用户并不是对表中的每一行都感兴趣,而只是关心一个总体概念。这时就需要用到总计。总计行在查询设计网格中,缺省情况下是不显示的;要显示该行,只需在工具栏中单击“总计”按钮。此时总计行中的每个字段对应值均为 Group By,它是总计项的缺省值。

假如,用户要看某张订单中各种产品的数量总和,此时就需要一个合计查询。

一、组中的合计

单击总计行中的任何一个字段列,在框中会显示一个向下的箭头,单击它,下拉显示了十二种合计功能。

- Group By 每一个字段中的记录将被分组而不合计,这是一种缺省情况。
- Sum 为每一组中该字段计算所有值的和,当然该字段类型只能是数值或货币型。
- Avg 为每一组中该字段计算所有值的算术平均值,同样它只能操作数字和货币型字段。
- Min 返回每组中一个字段的最低值。数字返回最小值,文本返回对照顺序的最低值,Microsoft Access 忽略 null 值。
- Max 返回每组中一个字段的最高值。数字返回最大值,文本返回对照顺序的最高值。
- Count 返回行的合计,要求这些行中指定的字段不能是 null 值。
- StDev 计算每一组中某字段所有值的统计标准差。如果该组只有一行,Microsoft Access 返回 null 值。
- First 返回该字段的第一个值。
- Last 返回该字段的最后一个值。

Expression 与 Where 将在下面作介绍。

现在来完成在订单中统计产品数的工作,先打开查询的“设计”视图,在字段列表中单击产品 ID 字段并拖到网格中的第一列,然后再单击数量并拖到网格的第二列中。

单击工具栏中的“总计”按钮,在网格中显示总计行,在数量列中将 Group By 改为 Sum,图 5.25 显示了此时的“设计”视图,图 5.26 显示了此查询的“数据表”视图。

当用户需要在字段中使用一个或多个合计功能时,可将总计行设为 Expression。例如,想设计一字段显示字段中值的范围:Max[单价]-Min[单价]。

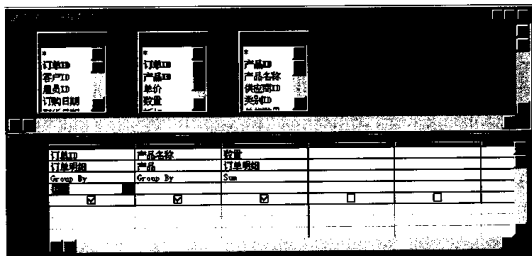


图 5.25

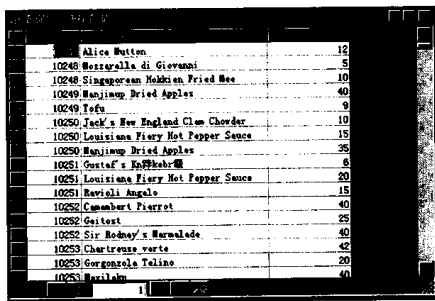


图 5.26

二、选择记录构成组

有时,用户在构成合计查询的组中不想包含一些记录,此时,用户可向查询中加入一个或多个过滤标准字段。

可在总计行中选择 Where 设置,并输入准则,说明将选择哪些数据,同时显示框自动消失,即不显示此字段。

要在订单明细表中查询某产品的平均单价、总数量,其“设计”视图如图 5.27 所示。图中产品 ID 字段的总计行设为 Where,显示选择框中空白,在准则行中输入 Ikura。

在运行查询时,只得到 Ikura 产品的合计,如图 5.28 所示。

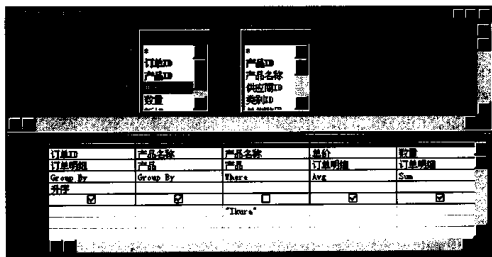


图 5.27

订单ID	产品名称	产品价格	订单明细	数量
10276	There's	¥ 24.00	15	24
10357	There's	¥ 24.00	30	15
10399	There's	¥ 24.00	16	30
10449	There's	¥ 24.00	14	16
10490	There's	¥ 24.00	20	14
10479	There's	¥ 24.00	20	20
10519	There's	¥ 31.00	18	20
10524	There's	¥ 31.00	2	18
10560	There's	¥ 31.00	5	2
10609	There's	¥ 31.00	10	5
10612	There's	¥ 31.00	70	10
10646	There's	¥ 31.00	18	70
10664	There's	¥ 31.00	24	18
10676	There's	¥ 31.00	2	24
10695	There's	¥ 31.00	20	2
10690	There's	¥ 31.00	15	20

图 5.28

5.2.13 参数查询

一般情况下,用户知道自己所要查询的值,那么在创建查询时就可以直接向准则行中输入值。而有时用户不确定要查找什么值,或者想形成一个通用的查询,那么用户可以使用参数查询。

参数查询是指查询中含有参数,当查询运行时,Microsoft Access 会提示用户输入值。

为了在查询中设置一个参数,用户只需在相应字段的准则行中输入一个带方括号的名字或短语,而不用向准则行中输入字段所要查找的范围值。

图 5.29 所示就是一个参数查询。

需要注意的是,参数名称可以包含字段名称,但不能与字段名称相同。对于显示日期的字段,可以显示类似于“请输入开始日期:”或“请输入结束日期:”这样的提示,以确定输入值的范围。

对查询中的参数,用户可以告诉 Microsoft Access 需要何种数据类型,当用户输入值时,Microsoft Access 会自动检测是否匹配。例如,当用户定义参数是数字时,参数就不接受字母。

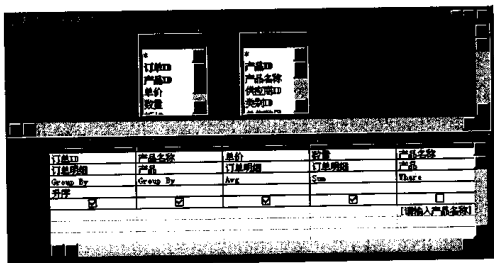


图 5.29

确定查询参数的类型是十分有用的。如果用户需要改变或确定查询参数的数据类型,从“查询”菜单中选择“参数”命令,Microsoft Access 会显示如图 5.30 的“查询参数”对话框。

在对话框的参数一列中,用户可以输入参数名,此时不用再加方括号。在数据类型列中,从下拉列表中选择合适的数据类型,完成定义后,单击“确定”按钮。

运行查询时,Microsoft Access 向用户提示给参数输入值,显示如图 5.31 所示的对话框。当输入“lkura”时,查询结果如图 5.32 所示。

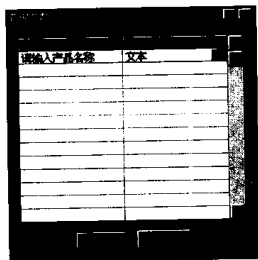


图 5.30



图 5.31

Chang	¥ 15.20	20
10258 Chang	¥ 15.20	50
10284 Chang	¥ 15.20	36
10298 Chang	¥ 15.20	40
10327 Chang	¥ 15.20	25
10335 Chang	¥ 15.20	7
10342 Chang	¥ 15.20	24
10393 Chang	¥ 15.20	25
10418 Chang	¥ 15.20	60
10435 Chang	¥ 15.20	10
10440 Chang	¥ 15.20	45
10469 Chang	¥ 15.20	40
10485 Chang	¥ 15.20	20
10504 Chang	¥ 19.00	12
10811 Chang	¥ 19.00	10
10822 Chang	¥ 19.00	20
10832 Chang	¥ 19.00	30

图 5.32

5.3 交叉查询

Microsoft Access 支持一种特殊类型的合计查询,就是交叉查询,它可以使数据按电子表格的格式进行显示。从图 5.33 可以看出“选择查询”产生的动态集与“交叉查询”产生的动态集之间的区别。“选择查询”仅在垂直方向按雇员和类别对总计值进行分组,这样每个雇员要显示多次,使不同雇员的总计值的比较较为困难。而“交叉查询”显示的是相同的信息,但在水平与垂直方向同时对数据进行分组,因而数据更为紧凑。

Callahan, Laura	海洋食品	¥ 147.00
Callahan, Laura	面食/谷类	¥ 608.00
Callahan, Laura	肉/家禽	¥ 351.00
Callahan, Laura	水	¥ 1,440.00
Callahan, Laura	糖果	¥ 1,578.00
Bevelio, Nancy	海洋食品	¥ 528.00

	¥ 147.00	¥ 608.00	¥ 351.00	¥ 1,440.00	¥ 1,578.00
Bevelio, Nancy	¥ 528.00	¥ 187.40	¥ 754.56	¥ 2,496.40	¥ 1,218.00
Faller, Andrew	¥ 182.40	¥ 100.80	¥ 873.60	¥ 524.00	¥ 945.00
Leverling, Janet	¥ 1,403.34			¥ 182.50	¥ 451.00
Peacock, Margie	¥ 2,200.20	¥ 2,578.00	¥ 614.40	¥ 4,368.00	¥ 1,410.00
Soyama, Michael		¥ 1,863.40	¥ 230.40	¥ 876.00	

图 5.33

要创建一个“交叉查询”,与创建“选择查询”并无多大区别,下面以一个例子说明创建“交叉查询”的步骤,并与“选择查询”作比较。假设要查询每个雇员销售产品的情况,条件如:货主国家为德国,订货日期在 1994 年 1 月 1 日与 1995 年 1 月 1 日。

首先用户在“数据库”窗口中单击“查询”按钮,再单击“新建”按钮,此时 Microsoft Access 弹出“新建查询”对话框,选中“设计”视图,再单击“选定”按钮。Microsoft Access 自动弹出显示表对话框,由于此“交叉查询”涉及到订单、订单明细、产品、类别、雇员等表,分别双击列表框中的表名,选中这些表,然后单击对话框中的“关闭”按钮。

先设计该查询为“选择查询”的情况。

选择类别表中的类别名称,由于对货主国家与定货日期有限制,也需将此两字段选入网格中。在显示雇员姓名时,格式为姓名,而在雇员表中只有姓氏字段与名字字段,因此需将两字段组合而成,在最左边字段中输入如下表达式:

姓名:[雇员]![姓氏]&","&[雇员]![名字]

用户想要看的是订单中的总价,因此在计算总价时输入表达式:

CCur([订单明细]![单价]*[订单明细]![数量]*
(1-[订单明细]![折扣]/100)*100)

并且由于要显示相同种类产品的总价,在合计行中选中 Sum。当然,在输入上面那些表达式时,可以利用“表达式生成器”。

接着再输入相应的限制项。在货主国家字段的准则中输入“德国”,在“订购日期”字段准则中输入

between #1-1-94# and #1-1-95#

并且将这两个字段的总计行均改为 Where。此时“设计”视图如图 5.34 所示。

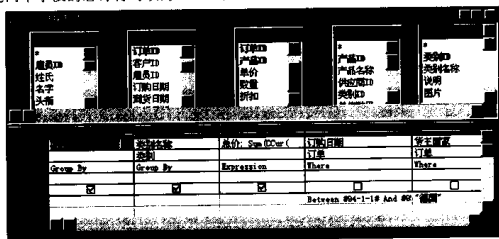


图 5.34

单击工具栏中的视图,运行此查询,结果如图 5.33 所示。

下面进行“交叉查询”的设计。

在设计完成上述步骤以后,由于查询类型须更改为“交叉表查询”,单击工具栏中的查询类型按钮,并且在下拉框中选中“交叉查询”。此时,用户会发现,在原来“选择查询”、“设计”视图中的显示行被交叉表行所替代。

单击交叉表行中的任一列,在框的右侧有一个箭头,单击它,会显示一下拉框,框中有四个选项。

- 行标题 输入“行标题”代表这个字段的名称将会被用作“行标题”,此字段形成一行。

● **列标题** 输入“列标题”代表这个字段的名字将会被用作“列标题”，此字段行成一列。

● **值** 输入“值”代表这个字段的内容将会被用作由行列形成的框中的值。在一个交叉查询中，只有一个字段的交叉表行可以被设为值。

● **不显示** 代表这个字段只是起一个辅助的作用，可能是为了作为条件使用的，在动态集中，它不会在电子表格中显示出来，这个域的总计行一般设为 Where。

在一个查询中，只能有一个字段作为列标题，一个字段作为值，但可以有多个字段的交叉表行为行标题与不显示。

在姓名列中，交叉行选为行标题，而类别名称字段是列标题，并令总价列的交叉表行为值。与“选择查询”不同的是，要将总价列中的总计行改为 Expression，并且将表达式改为：

Sum(CCur([订单明细]![单价]*[订单明细]![数量]*
(1-[订单明细]![折扣])/100)*100)

此时，就完成了交叉表的设计，视图如图 5.35 所示。

单击工具栏中的“视图”按钮，运行此“交叉查询”，结果如图 5.33 所示。

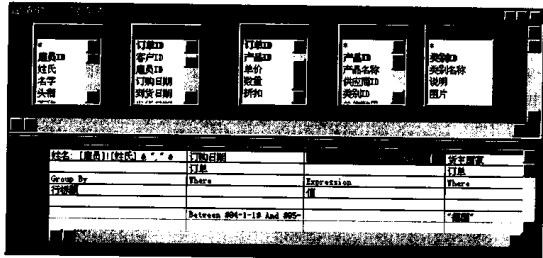


图 5.35

5.4 使用查询向导

用户会发现，自主创建查询是一件十分麻烦的事，在创建一些较常见的查询时，完全不必如此费事，Microsoft Access 提供了查询向导帮助创建查询。

用户在单击“新建”按钮后，Microsoft Access 会弹出“新建窗口”对话框，在框中有四种查询向导，下面分别介绍。

5.4.1 使用简单查询向导

选中“简单查询向导”，然后单击“确定”按钮，Microsoft Access 显示如图 5.36 所示的“简单查询向导”窗体。

通过“简单查询向导”创建查询，可以在一个或多个表或查询中指定的字段检索数据。如果需要，向导也可以对组成的全部记录进行总计、计数以及平均值的计算，并且可以计算字段中的最大值和最小值，但不能设置准则。

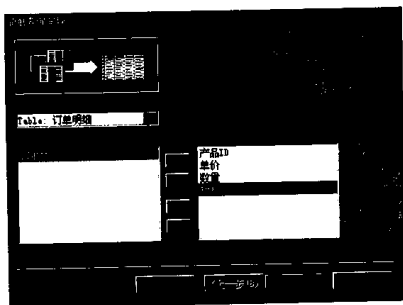


图 5.36

在“表/查询”列表框中选中查询所要基于的表或查询，在下面的可用字段框中列出了该表或查询中的字段，单击要选的字段，再单击“)”键，此时该字段会出现在右边的选定字段框中，按“<”键可将可用字段框中的所有字段均选中。重复以上操作，直至全部完成字段选择。

再单击“下一步”按钮，进入如图 5.37 所示的画面。在此步中，用户可为该查询命名，并且确定在完成创建后的工作。可以选择修改查询设计项，进入“设计”视图中更改设计。

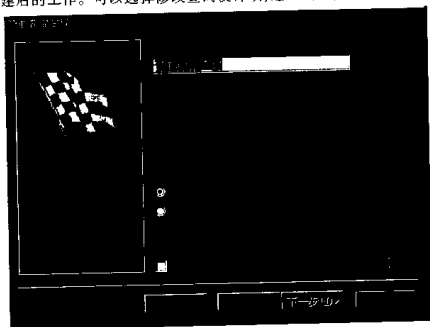


图 5.37

5.4.2 交叉表查询向导

在新建查询窗口中选中“交叉表查询向导”，Microsoft Access 显示如图 5.38 所示的画面。

在此窗口中,用户需要选中一个字段作为行标题,最多可以选三个。单击“下一步”,Microsoft Access 显示如图 5.39 所示的窗口。用户选中一字段作为列标题,再单击“下一步”,在此窗口中,用户确定对行与列的交叉点作的计算,最后完成查询的创建。

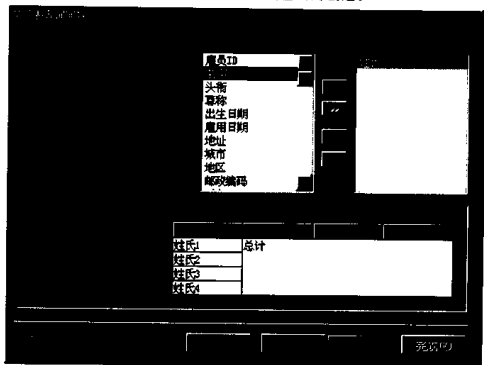


图 5.38

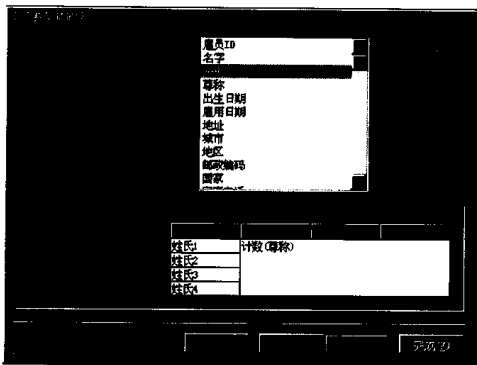


图 5.39

5.4.3 “查找重复项查询向导”与“查找不匹配项查询向导”

这两个查询向导特别有用,例如,当用户需要查看哪些客户没有订单,就可以使用“查找不匹配项查询向导”;而当用户需要查找有多个公司客户的城市时,就需要使用“查找重复项查询向导”。

向导的使用十分简单,用户只需按照向导提供的提示信息进行操作就可以了,这里不再重复叙述。

第六章 操作查询

在前一章中，已经叙述了“选择查询”。“选择查询”在使用时，只是简单的将各个表中的数据综合起来，形成一个动态集，并不对数据进行修改，它是查询中应用最为广泛的一种。还有另外一种查询——“操作查询”，它代表一种操作，如删除和输入。它是以选择查询为基础的，也需要挑选某些符合条件的数据，然后以整批的方式运行某些动作。操作查询分为删除查询、更新查询、追加查询和生成表查询。

下面就分别介绍这四种操作查询。

6.1 删除查询

利用删除查询，用户可从一个或多个表中删除一组记录。例如，可以用删除查询来删除没有定单的产品。使用删除查询，将删除整个记录，而不只是记录中所选择的字段。

如果允许连锁删除（例如，所有来自爱尔兰的顾客和他们的订单），可以使用单一删除查询来删除单个表、一对一关系或一对多关系中的多个表的记录。但是，如果需要用“一”表来包含“多”表，为了添加准则，必须执行两次查询，因为一个查询不能同时从主表和相关表中删除记录。

删除记录分为以下三种情况：

- 从单个表或一对一关系表中删除记录；
- 使用只包含一对多关系中“一”端的表的查询来删除记录；
- 使用包含一对多关系中两端的表的查询来删除记录。

首先要挑选删除的记录，就必须建立一个选择查询，实际上这也是建立查询的第一步。创建选择查询的步骤同上一章前几节叙述的一样。例如，要删除订单表中订单 ID 小于 10 250 的记录。

然后单击工具栏中的“查询类型”按钮，在下拉框中选中删除查询，用户会发现选择查询中的显示一行，换成了删除行。由于要删除订单表中的记录，因此将要删除的表——订单的字段列表中的星号拖到网格中，结束条件是当订单 ID < 10 250。因此需要标记准则，将设置准则的字段——订单 ID 拖到设计网格中。当光标从订单、* 列中移开时，删除列自动变为 From。在订单 ID 列中选择 Where，因为它是约束条件字段，并在准则行中输入 < 10 250，此时视图如图 6.1 所示。如果要预览即将删除的记录，单击工具栏中的“视图”按钮，则显示如图 6.2 的数据表。

如果要删除记录，单击工具栏上的“执行”按钮。

在利用删除查询进行删除时，需注意以下几点。

- 使用删除查询删除了记录之后，将不能撤消这个操作。因此，在执行删除查询之前，

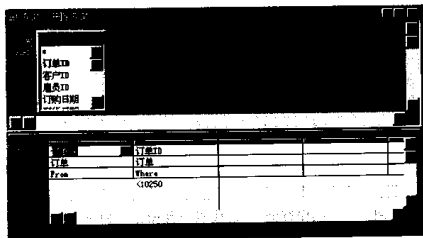


图 6.1

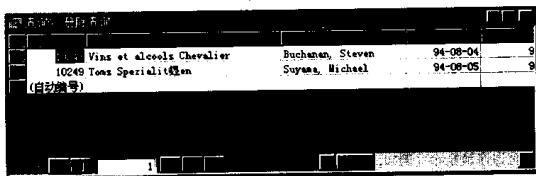


图 6.2

应该预览即将删除的数据，可以单击工具栏上的“视图”按钮来完成这个操作，可以在“数据表”视图中查看查询。

- 应该随时备份数据。如果不小心删除了数据，可以从备份的数据中取回它们。
- 在某些情况下，执行删除查询可能会同时删除相关表中的记录，即使它们并不包含在此查询中。当查询只包含一对多关系中的“一”端的表，并且允许对这些关系使用连锁删除时就可能发生这种情况。在“一”端的表中删除记录，同时也删除了“多”端的表中的记录。
- 如果在 Paradox、dBASE 或 FoxPro 表启动删除查询，而这些表已链接到数据库，则在开始运行后将不能中止。

6.2 更新查询

对一个或多个表中的一组记录作全局的更改，例如，可以将所有产品的价格提高 10 个百分点。使用更新查询，可以更改已存在表中的一批数据。

下面介绍利用更新查询更改记录的步骤。

首先要创建一个选择查询来选择正确的数据集。选择查询是操作查询的基础。在该选择查询中应包括所有的更新字段与准则字段。

然后，单击工具栏上的“查询类型”按钮旁边的箭头，然后单击“更新查询”按钮。用

户也许已经注意到,选择查询视图中的显示行变成“更改到”行。在准则字段中输入准则,然后在要更新的字段的“更新到”单元格中输入用来改变这个字段的表达式或数值。例如,要将订单明细表中的单价全部提升 10%,则表达式应写成 [单价] * 1.1,如图 6.3 所示。

如果要查看将要更新的记录列表,单击工具栏上的“视图”按钮即可。如果要进行更新,单击“执行”按钮。

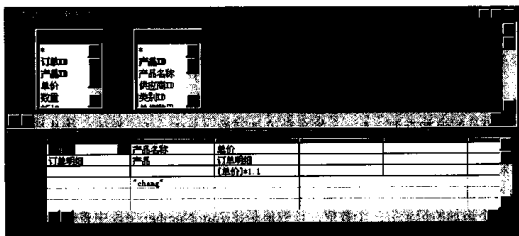


图 6.3

6.3 追加查询

从一个或多个表将一组记录追加到一个或多个表的尾部。例如,假设获得了某些新客户和包含这些客户信息表的数据库。为了避免键入所有这些内容,最好将它追加到“客户”表。在以下方面追加查询也十分有用。

- 基于准则追加字段。例如,可以仅追加大量订单客户的姓名和地址。
- 当表中的某些字段在其他表中不存在时才追加记录。例如,在“罗斯文”示例数据库中,“客户”表有 11 个字段。假设要从另一表来追加记录,这些记录必须匹配“客户”表中 11 个字段中的 9 个字段。追加查询将追加匹配字段中的数据并且忽略其他的数据。

下面介绍利用追加查询追加记录的步骤。

- (1) 新建一个选择查询,该查询应包含要追加记录到另一个表的表。
- (2) 在查询“设计”视图中,请单击工具栏上“查询类型”按钮旁边的箭头,然后再单击“追加查询”选项,将显示“追加”对话框,如图 6.4 所示。

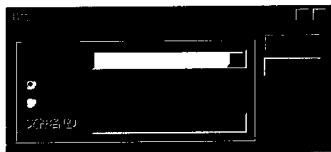


图 6.4

(3) 在“表名称”框中,输入要追加记录的表名称。

如果该表在当前打开的数据库中,请单击“当前数据库”,或单击“其他数据库”并键入存放这个表的数据库名,必要时键入路径。也可以输入到 FoxPro、Paradox 或 dBASE 数据库的路径,或输入到 SQL 数据库的连接字符串。

(4) 选定后,单击“确定”按钮。

从字段列表将要追加的字段、想要用来设置准则的字段拖动到查询设计网格中。如果已经在两个表中选择了相同名称的字段,Access 将自动在“追加到”行中填上相同的名称。如果在两个表中并没有相同名称的字段,在“追加到”行中将输入所要“追加到”表中字段的名称。

如果表中有“自动编号”的数据类型,则可以增加或不增加主关键字字段。

- 根据设计的追加查询,Access 将自动追加新的 AutoNumber 数值,或从原始表中获取数值。

- 如果要让 Access 自动增加 AutoNumber 数值,在创建此查询时,不要将 AutoNumber 字段拖动到查询设计网格中。使用这个方法,Access 将追加记录并且自动插入 AutoNumber 数值。第一个追加的记录有一个数值,该数值是比 AutoNumber 字段中曾输入过的最大数值还要大的值(即使记录包含了已经删除的最大 AutoNumber 数值)。

- 如果要保留来自原始表中的 AutoNumber 数值,在创建此查询时,将 AutoNumber 字段拖动到查询设计网格中。如果在要追加记录的表中的 AutoNumber 字段是某个主关键字,同时原始表和要追加记录的表包含重复的“自动编号”数值,就必须改用前一种方法。

- 如果两个表中所有的字段都具有相同的名称,可以只将星号拖动到查询设计网格中。但是,如果用户正在数据库副本上工作,则必须追加所有的字段。

- 如果需要指定准则,则在已经拖动到网格中的字段的“准则”单元格中,输入用于生成追加内容的准则。

- 假如要从客户备份表中向客户表中追加记录,追加准则是公司名称以 A 开头,则此时的追加查询如图 6.5 所示。

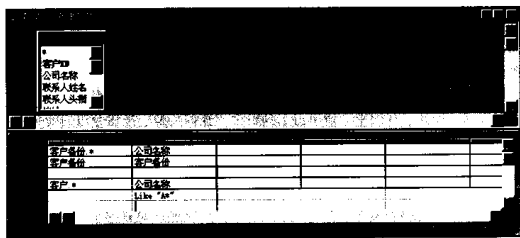


图 6.5

- 如果要预览即将添加的查询,请单击工具栏上的“视图”按钮。如果要返回查询“设计”视图,请再单击工具栏上的“视图”按钮。在“设计”视图中,可以进行所需的更

改。

- 如果要追加记录，请单击工具栏上的“执行”按钮。

6.4 生成表查询

生成表查询利用一个或多个表中的全部或部分数据来新建表。生成表查询可应用在以下方面。

- 创建用于导出到其他 Access 数据库的表。例如，可以创建包含“雇员”表的几个字段的表，然后将表导出到数据库以供有关人事单位使用。

- 创建从特定时间点显示数据的报表。例如，假设要在 1997 年 5 月 15 日打印显示以 1997 年 4 月 1 日早上 9 点的表中的数据为基础的的第一个季度销售总和的报表。以查询 SQL 语句为基础的报表，将从表中抽取最新数据（1997 年 5 月 15 日的数据），而不是特定日期和时间的记录。要保留 1997 年 4 月 1 日上午 9 点的完整数据，在该时间点创建生成表查询以检索所需要的记录，并且将它们保存在一个新表中，然后使用此表，而不是查询，作为报表的基础。

- 创建表的备份副本。

- 创建包含旧记录的历史表。例如，在从当前的“订单”表中删除记录之前，可以创建表来保存所有旧定单。

- 提高基于表查询或 SQL 语句的窗体和报表的性能。例如，假设要打印多个报表，且这多个报表基于五个包含总和的查询，可以通过下面的方法来加快速度。首先创建一个生成表查询，检索所需要的记录并且将它们保存在表中，然后可以将报表基于这个表或在 SQL 语句中指定的表。这样无须重新执行每一个报表的查询。但是，在执行生成表查询时，表中的数据处于冻结状态。

下面介绍利用追加查询建立新表的步骤。

(1) 新建一个选择查询，并选择包含要放到新表中的记录的表或查询。

(2) 在查询的“设计”视图中，单击工具栏上“查询类型”按钮旁边的箭头，然后单击“生成表查询”选项，显示“生成表查询”对话框，如图 6.6 所示。

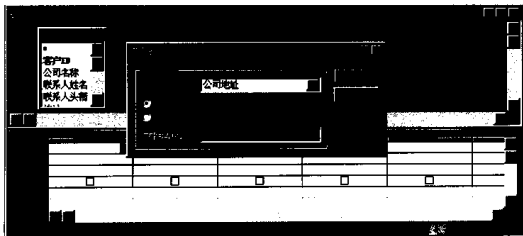


图 6.6

(3) 在“表名称”框中输入所要创建或替换的表名称。

(4) 单击“当前数据库”，将新表放入当前打开的数据库；或单击“另一数据库”并键入要放入新表的数据库名，必要时键入路径。

(5) 选定后，单击“确定”按钮。

用户从字段列表将包含在新表中的字段拖动到查询设计网格中。对于拖动到网格的字段，如果需要，在“准则”单元格里键入准则。假设用户需要创建客户表备份，则最后的“设计”视图如图 6.7 所示。

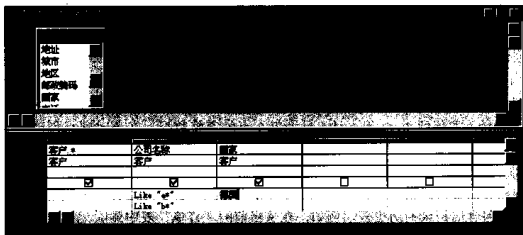


图 6.7

如果在新建表之前预览新表，请单击工具栏上的“视图”按钮。如果要回到查询“设计”视图并做一些更改或者执行查询，请单击工具栏上的“视图”按钮。如果要新建表，请单击工具栏上的“执行”按钮。

第七章 窗体基础

在数据库的使用中,窗体是用户与数据库之间的桥梁,它是 Microsoft Access 中最灵活的部分,它可以让用户的系统更丰富,更具有变化。

7.1 窗体的使用

窗体是用户和 Microsoft Access 应用软件之间的主要接口,它的应用可分为以下几种。

1. 显示与编辑数据

这是窗体最普遍的应用。窗体给用户提供一种方法来规定数据库中数据的显示。用户还可以利用窗体对数据库中数据进行修改、添加、删除。用户可以在窗体中设置选择,把部分或全部数据变为只读,并自动填充来自其他表的相关信息,计算要被显示的值,显示或隐藏建立在记录中其他窗体的值上的数据或被窗体用户选取的选择。

2. 控制应用程序流

窗体与 Visual Basic 中的窗体一样,它可以和函数、子程序等相结合,每个窗体中都可以用 Visual Basic for Application 编写代码,并且由代码完成相应的功能。例如,在窗体中用户可以放置按钮,单击这个按钮时,会触发一系列的操作,如弹出一个窗口显示提示信息等。

3. 接收数据输入

用户可以设计一个专用的窗体,作为数据库数据输入的通道。

4. 显示消息

在窗体中可以显示一些解释或警告信息。Microsoft Access 97 中的函数 MsgBox 可以用来显示信息。

5. 打印数据

在 Microsoft Access 中,除了报表对象之外,窗体也可以用来打印数据。

7.2 窗体视图

本节先介绍使用窗体时的视图——“窗体”视图。

在“数据库”窗口中,单击“窗体”按钮,在框中列出了 Northwind 数据库中所有的窗体,选中其中任一个,再单击“打开”按钮或双击之,则窗体就以“窗体”视图的形式显示。图 7.1 显示的是数据库中的“订单”窗体。由图中可以看出,窗体并没有像表的“数据表”视图那样的格式。窗体的画面是由用户设计的,而且它的操作也是由用户完成的。

在工具区内,像其他视图一样,都有专门适用于“窗体”视图的工具出现,如图 7.2 所示。

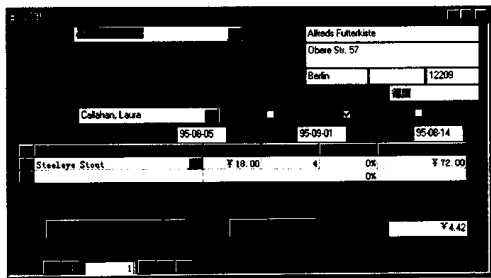


图 7.1



图 7.2

下面介绍这些工具条中的按钮。

● 视图按钮——当单击此按钮时，会使窗体切换到“设计”视图下。如果单击按钮中的箭头，会下拉显示三种视图——“设计”视图、“窗体”视图与“数据表”视图。也许用户会奇怪，为什么窗体还有“数据表”视图呢？事实上，窗体一般都是基于表或查询的，因此一般也有数据表。特别是对于那些用于输入或显示数据的窗体，它一般以“数据表”视图出现。

● 保存按钮——单击此按钮，保存用户对窗体所作的修改。

● 打印按钮——按下此按钮，用户可以将显示的窗体打印出来。

● 打印预览按钮——按下此按钮，屏幕上将出现当前窗体的“打印预览”画面，用户可以查看将要打印出来的样本，如果不满意，还可以作修改。

● 剪切按钮——按下此按钮，将当前选定的数据删除，并将此数据放置于剪贴板中。

● 复制按钮——按下此按钮，将当前所选取的数据制作一份拷贝于剪贴板上，而当前被复制的数据并不被删除。

● 粘贴按钮——按下此按钮，可以将剪贴板上的数据贴至当前选取的记录或元件上。

● 撤消按钮——要撤消用户最近的一次操作，则单击此按钮。

● 工具栏按钮——单击此按钮，在工具栏区内显示 Web 工具条。

- 升序按钮——先用鼠标在要排序的域中单击一下，然后单击此按钮，所有的记录将会根据这个域所对应的字段值按升序排列；用户不能为子表格中的数据进行排序。
- 降序按钮——先用鼠标在要排序的域中单击一下，然后按下此按钮，窗体中所有的记录会根据这个域所对应的字段值按降序排序；同样不能为子表中的数据排序。
- 按选定内容筛选按钮——直接在窗体内选定某个值的全部或部分，然后在工具栏上单击此按钮，以查找具有选定值的所有记录。
- 按窗体筛选按钮——在空白字段中指定要在窗体中搜索的值，可以键入该搜索值或从列表中选择。
- 应用筛选按钮——在完成筛选的建立后，单击此按钮，运行筛选，并在窗体中显示查寻得到的数据。
- 查找按钮——在所要查找的内容中单击一下，然后按此按钮，Microsoft Access 会显示查找对话框，然后 Microsoft Access 按输入的条件查寻相应的记录，并在窗体中显示记录。
- 新记录按钮——在窗体中单击一下，再单击此按钮，窗体中所有的域值均为空，待用户输入新的记录。
- 删除记录按钮——选中要删除的记录，然后单击此按钮，此记录将被删除，窗体将显示下一条记录。

7.3 窗体简介

下面介绍窗体的组成部分以及不同类型的窗体。

7.3.1 窗体组成

Microsoft Access 中的窗体可分为三个部分：页眉、主体、页脚。这三个部分都分别有自己的作用。

一般的窗体只有主体部分。在主体部分将放置控件，来完成窗体的功能。任何一个窗体均具有主体部分。一般将用户想要显示的基础表或查询中的信息放在窗体的主题部分。用户可以在窗体的顶部加一个标题，在窗体的底部加一个注脚，来显示那些在不同的记录中不需改变的信息。

页眉与页脚一般是用来放置一些辅助信息的。一般在页眉放置总体信息，如窗体的标题。如果窗体内是雇员的个人信息，则可以在页眉中显示雇员的姓名。页脚可放置选项信息。

在 Northwind 数据库的客户电话列表窗体中，既有页眉，也有页脚。在数据库窗体中双击“客户电话列表”窗体，则显示如图 7.3 所示。

在窗体顶部是窗体的页眉，在页眉中列出了公司名称、联系人、电话、传真等列标题。在窗体的最下部，灰色部分是页脚，它有显示字母的按钮。当用户在主体内移动光标时，窗体中的记录会上下滚动显示，而页眉与页脚却不会移动。

窗体的类别有很多，主要有以下几种：多页窗体、连续窗体、子窗体。



公司名称:	联系人:	电话:	传真:
Alfredo Fullerton	Marie Anders	030-0074321	030-0076545
Ana Trujillo Emparedados y helados	Ana Trujillo	(5) 555-4729	(5) 555-3745
Antonio Moreno Tacos	Antonio Moreno	(5) 555-3932	
Around the Horn	Thomas Hardy	(171) 555-7788	(171) 555-0750
Berglunds snabbköp	Christina Berglund	0921-12 34 65	0921-12 34 67
Blaug Seer Delikatessen	Hanna Moos	0621-08490	0621-08924
Blondel père et fils	Françoise Cléaux	88.60.15.31	88.60.15.32
Bonito Comidas preparadas	Marta Sommer	(91) 555 22 82	(91) 555 91 99
Bon app'	Laurence Labihien	91.24.45.40	91.24.45.41
Bottom-Dollar Markets	Elizabeth Lincoln	(804) 555-4729	(804) 555-3745

图 7.3

7.3.2 多页窗体

用户也许会由于需要显示许多信息,而且这些信息可分成几组,要是将所有这些信息均设计在一页中,那会使窗体变得十分复杂,而且查看也十分不方便。在这种情况下,用户可以考虑使用多页窗体。

在 Northwind 数据库中,雇员窗体是多页窗体。打开此窗体,用户将看到窗体的第一页,如图 7.4 所示。第一个显示的是个人信息,要是想查看每个记录的第一页,则可单击记录选择器中的箭头。为了了解雇员的第二页信息,即公司信息,可单击窗体中的滚动条,或按 PgDn 键。图 7.5 显示了雇员的第二页信息。



Northwind

15

Devolio

销售代表

Fulke, Andrew

92-05-01

5467

Photo of a woman with long dark hair.

图 7.4

用户还可以注意到,窗体中的顶部是用来显示该雇员的姓名的,在页面移动时,是不改

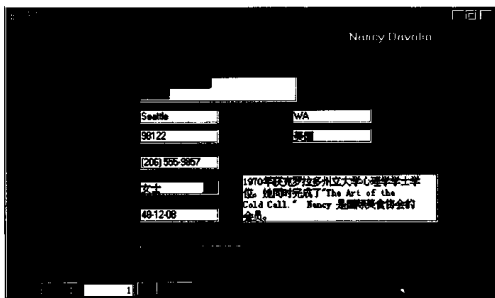


图 7.5

变的。它是该窗体的页眉。

7.3.3 连续窗体

有时，用户只需要在窗体中显示很少的几条记录，要是像普通窗体那样显示，会十分难看，此时可以使用连续窗体。它与普通窗体不同。普通窗体在一个窗体画面中只显示一条记录，而连续窗体在一个画面内显示已经格式化好的多条记录。

Northwind 数据库中的“产品列表”是连续窗体，如图 7.6 所示。用户可以用垂直滚动条来移动记录显示。

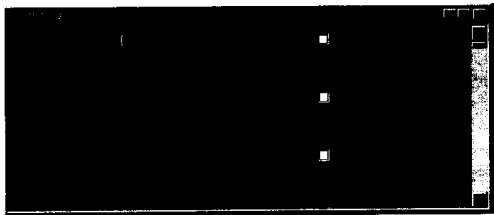


图 7.6

7.3.4 子窗体

在图 7.1 所示的“订单”窗体中，有一个类似于表的控件，用来显示数据，且里面的数据能随主窗体中数据的改变而改变，这种控件叫做子窗体，是一种比较特殊的窗体。

当然，既然有子窗体就有主窗体，“订单”窗体就是主窗体。事实上，两个窗体的数据

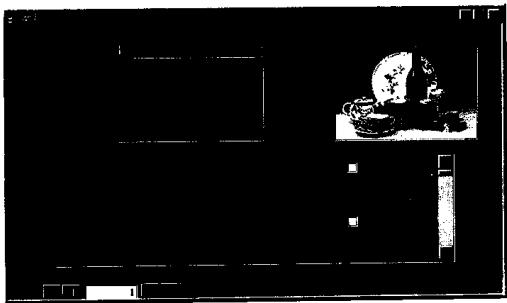


图 7.7

来源分别属于不同的查询对象，在定义好它们之后，只需要很简单地设置参数特性就行了。

7.4 在窗体中操作数据

7.4.1 观察数据

在窗体中进行数据操作与在“数据表”视图中，包括表、查询与窗体、报表，虽然看起来不大相同，但是实际上概念是相类似的。因为窗体也是基于表或查询而建立的。下面，以实例进行说明。

按照以前所述的方法打开数据库中的“类别”窗体，如图 7.7 所示。在这一窗体中，用户可以看到它有一个子窗体，实际上这个子窗体就是在上一节所说的连续窗体。用户可以用 Tab 键移动当前光标，当光标移至主窗体的最后一个域时，再按 Tab 键，就可以将光标移至子窗体内。实际上，用户要查看与当前主窗体中对应的子窗体的数据，只需利用鼠标在子窗体内单击一下，将焦点置于其上，然后按 PgDn 与 PgUp 键或用鼠标拖动右侧的滚动条就可浏览子窗体内的数据。由于子窗体内没有记录选定器，当用户需要查看这是第几个记录时，只需单击子窗体右侧滚动条中的按钮，并且按住不放，Microsoft Access 会自动显示提示框，说明这是总记录之第几条记录。

如果用户想查看主窗体内的下一条记录，只要用鼠标单击主窗体下部的记录选定器中的右箭头按钮，则窗体内就显示出下一条记录；单击左箭头按钮，则回到上一条记录。这与在表的“数据表”视图中的操作是一样的。像“订单”窗体中，子窗体就是以“数据表”视图出现的，它的操作与表操作完全一样。

前面已经提到过，除了“窗体”视图外，窗体还有一种“数据表”视图。“类别”窗体中的子窗体就是前面介绍的“产品列表”窗体。在缺省情况下，它是以“窗体”视图显示的，也可以用“数据表”视图显示它。

首先必须先关闭“类别”窗体。这是因为当“类别”窗体打开时，“产品列表”窗体是它的一个部分，也是打开的，如果此时再打开“产品列表”窗体，将发生冲突。然后双击“产品列表”窗体名，打开“产品列表”窗体，与图 7.6 相同。此时用户会发现，在工具栏内，工具条并未显示其他特殊的工具。虽然不能利用工具栏中的工具，但可以利用快捷菜单。将鼠标指针放置在窗体的标题栏中，单击右键，Microsoft Access 弹出快捷菜单。在快捷菜单中选择“数据表”视图，则窗体以该方式显示，如图 7.8 所示。

产品	数量	价格
Chang	10 boxes x 20 b	¥ 18.00
Chang	24 - 12 oz bott	¥ 19.00
Aniseed Syrup	12 - 550 ml bot	¥ 10.00
Chef Anton's Cajun S	48 - 6 oz jars	¥ 22.00
Chef Anton's Gumbo	36 boxes	¥ 21.35
Grandma's Boysenberry	12 - 8 oz jars	¥ 25.00
Uncle Bob's Organic	12 - 1 lb pkgs	¥ 30.00
Northwoods Cranberry	12 - 12 oz jars	¥ 40.00
Nishi Kobe Niku	18 - 500 g pkgs	¥ 97.00
Ikura	12 - 200 ml jar	¥ 31.00

图 7.8

7.4.2 增加新记录

在 Microsoft Access 中，增加新记录的步骤会因窗体的设计不同而有所不同。

对于已经设计好的、专门为输入数据的窗体来说，要增加一个新记录，就是直接调出专用窗体作为输入数据之用。用户需要输入新的产品，只要在打开“产品列表”以后，单击工具栏上的“新记录”按钮，则所有域中的值均为空白，只有已设置了默认值的域中有数值，而且在记录选位器中，表明记录是在记录的尾部。在域中输入相应的值，就完成了新记录的添加，如图 7.9 所示。

产品
产品ID: 1
产品名称: Escargot Nouveaux
供应商: Escargot Nouveaux
类别: 水果
单位数量: 1000
总价: ¥ 200.00
库存数量: 1000
订购数量: 100
再订购点: 100
中止

图 7.9

在有些窗体中，由于在设计时不允许增加新记录，也许它只是一个记录收集窗体。在这种窗体中，用户不能添加新的记录，“产品列表”窗体就是这种窗体。对于子窗体而言，能否添加新记录完全取决于它的来源。例如，在“订单”窗体中的子窗体，能在它上面添加新记录，只要在子窗体内单击一下，然后单击工具栏上的添加新记录按钮，则光标自动向窗体中的最后一行移动，并且等待输入新记录，如图 7.10 所示。

Item Name	Price	Quantity	Total
Alice Mutton	¥ 99.00	1	¥ 99.00
Aniseed Syrup	¥ 10.00	1	¥ 10.00

图 7.10

而在“类别”窗体中，它的子窗体就是“产品列表”窗体，因此它的子窗体是不能添加新记录的。虽然“类别”窗体允许增加新记录，但是由于子窗体不允许，因此在主窗体内添加记录之后，子窗体内的记录是一片空白，无法添加，只能从其他窗体中添加，如“产品”窗体。添加了新记录的“类别”窗体如图 7.11 所示。

图 7.11

7.4.3 修改与删除数据

在窗体中修改数据，会由于视图的不同而不同。

如果窗体是在“数据表”视图下，用户可以使用表中修改数据的方法修改数据。当然，有些字段是不能修改的，主要是一些生成字段，例如由计算而形成的总计字段。

如果窗体是在“窗体”视图下，而且图是单一表格，用户一次只能看到一个记录。若是连续表窗体，用户可以同时看到多个记录。在“窗体”视图，用户可用鼠标单击所要修改的域，然后键入自己所需的值，而且还可以复制与拼接数据，但一次只能完成一个记录的一个域的修改、复制等。

在修改时，用户会发现，有的域用户是不能选中的，这是由于在设计窗体时，已将这个域的属性设置为不能聚焦的，用户不能控制、选择。这样可以禁止用户修改一些不希望修改的字段。

在单一窗体与连续窗体中删除记录与在数据表上删除记录不同。首先，用户必须先选中一个记录，就像在做报表中选中一个记录一样。如果窗体被设计成含有记录选定器，只要单击选定器来选择记录就可以。如果不含记录选定器，则用户就必须从记录中找到该记录，选中记录以后，单击删除记录按钮，从窗体记录中删除它。

有时，窗体内的记录是与其他表或查询中的数据相关的，不能删除。例如，在“产品”窗体中删除记录时，将会显示提示框，说明由于与“订单明细”表中记录相关，不能删除，如图 7.12 所示。而且在单独打开子窗体时，用户也不能删除窗体中的记录。

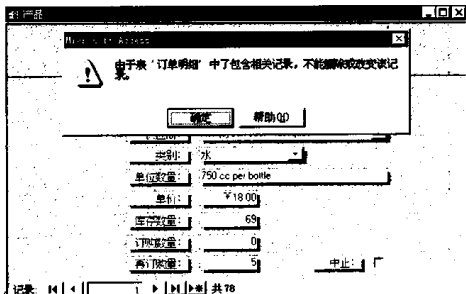


图 7.12

7.5 查找和排序数据

当用窗体来显示和编辑数据时，用户可以用在数据表中查找和排序数据大致一样的方法来查找数据，并以新的顺序排列。下面将介绍用查找命令以及更高级的方法来对窗体中的数

据进行查找和排序。

7.5.1 执行一个简单查找

如果用户需要在窗体内查找具有某一特征的数据,就必须利用 Microsoft Access 提供的查找功能。用户可先单击所要查找的相应的域,然后单击工具栏中的查找按钮,Microsoft Access 打开“在字段中查找”对话框,如图 7.13 所示。此对话框与表查找中的对话框一样。输入相应的值,然后按第三章中所叙述的内容进行操作,在窗体内,会显示查找得到的结果。在一个窗体上,可以进行任何域内的查找,包括计算值的查找。



图 7.13

7.5.2 排序

同数据表一样,在窗体内用户也可以按照某一域中的值进行排序。单击要排序的域,然后单击工具栏中相应的按钮,如果要按此域值的升序进行排列,则按“升序”按钮;如果按域值的降序排列记录,则按“降序”按钮。例如,在“产品”窗体中,要按产品名称升序排列记录,首先单击产品名称字段,焦点位于该字段内;然后单击“升序”按钮,则窗体内的记录就按产品名称字段升序排列。要注意的是,在表的“数据表”视图中,用户可以按任何字段的值进行排列,但在窗体中,用户只能按可以控制的字段进行排列,而不能利用像计算值等进行排列。在“产品”窗体中,用户不可能利用产品 ID 进行排序,因为这个字段不可控制。

7.5.3 筛选数据

在窗体中筛选数据,与在表中筛选数据十分相似。当用户从窗体记录中筛选出一些符合要求的记录时,需要使用 Microsoft Access 提供的筛选工具。同样,筛选方法有四种:按窗体筛选、按选定内容筛选、输入筛选与高级筛选。这四种方法各有用途,下面将分别叙述。

一、按窗体筛选

按窗体筛选是指在空白字段(取代数据)中键入或从列表中选择指定要在窗体或数据表中搜索的值,然后按此值进行筛选。

假设要在“定单”窗体记录中筛选出收单人为 Around the Horn,销售人为 Callahan Laura 或 Buchanan Steven 的所有定单。

首先在“数据表”视图中或在“窗体”视图中打开“定单”窗体,然后单击工具栏上的“按窗体筛选”按钮切换到“按窗体筛选”窗口,如图 7.14 所示。用户可以看到,窗口中所有的字段都是空的。单击要用于指定准则的字段,并要求记录筛选集合中的所有记录都必须满足该准则。如果在多个字段中指定筛选值,则筛选将仅返回那些同时满足所有这些值的

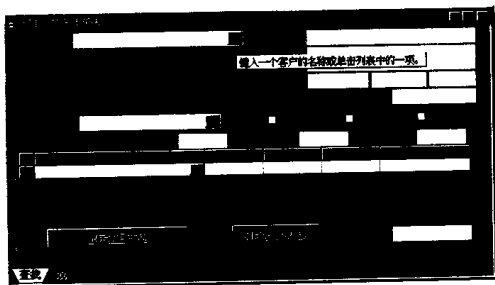


图 7.14

记录。

如果要建立准则，可以从字段列表中选择要搜索的字段值（如果列表中包含该值），或在字段中键入所需的值。建立准则时要注意如下几条。

- 要查找选中或没选中某个复选框、切换按钮或选项按钮的记录，可单击相应的复选框或按钮，直到它处于所需的状态。如果要将其回复中立的位置，使之不能作为准则来筛选记录，可继续单击复选框或按钮，直到它们变为灰色。
- 要查找某一特定字段为空或非空的记录，可在字段中输入 Is Null 或 Is Not Null。可以在带有 Memo、OLE 对象或 HyperLink 数据类型的字段，或在查询的计算字段中，从其列表上选择这些选项。
- 要使用准则表达式查找记录，可在适当的字段中键入表达式或使用表达式生成器来建立表达式。有关表达式的内容，参考前几章。
- 要指定筛选结果中的记录有选择性包含的值，可单击窗口底部的“或”标签，并输入相应的准则。

因此，在收单人字段中选中 Around the Horn，并且在销售人字段中选中 Buchanan Steven。然后按窗体下部的“或”标签，出现一个空白的“定单”窗体，再在收单人字段中选中 Around the Horn，并且在销售人字段中选中 Callahan Laura，至此完成了筛选的创建，如图 7.15 所示。

运行筛选时，单击工具栏上的“应用筛选”按钮，筛选将返回包含“查找”标签上所有指定值的记录，或第一个“或”标签上所有指定值的记录，或第二个“或”标签上所有指定值的记录，以此类推。图 7.16 显示的就是筛选后的结果。

二、按指定内容筛选

按指定内容筛选是指按选定窗体上的值或部分值，据此找出包含此值的记录。在 Microsoft Access 97 中，还有一种叫内容排除筛选，它与按指定内容筛选刚好相反，是指根据所给值，找出不包含此值的所有记录。

假设用户需要查询收货人为“Around the Horn”的所有订单，只需在“订单”窗体的收

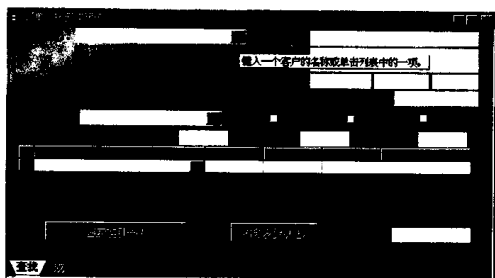


图 7.15

收单者(Q): Around the Horn
120 Hanover Sq
London WA1 1DP 英国

收货者(Q): Around the Horn
120 Hanover Sq
London WA1 1DP 英国

送货方式(Q): ☒ Speedy ☐ United ☐ Federal

销售人(Q): Callahan, Laura

订单ID: 11080 订购日期(Q): 97-05-29 到货日期(Q): 发货日期(Q):

产品	单价	数量	折扣	总价
Round-trip	¥ 10.00	1	0%	¥ 10.00
Alice Mutton	¥ 39.00	1	0%	¥ 39.00

显示全部产品 打印发货单(Q)

小计: ¥ 49.00
运费(Q): ¥ 0.00
合计: ¥ 49.00

记录: 14 共 2 (筛选)

图 7.16

货人字段中选中“Around the Horn”，然后单击工具栏中的“按指定内容筛选”按钮，运行筛选，就会显示出收货人字段符合准则的所有记录。

在保存表或窗体时，Microsoft Access 将同时保存筛选。在下次打开表或窗体时，如果需要，可以重新应用此筛选。

要进行内容排除筛选，筛选出不包含某一特定值的记录，只需在选择该值后单击鼠标右键，然后单击“排除选定内容筛选”即可。

在进行按指定内容筛选时，可以选择字段的全部或部分内容，选择的内容将决定筛选返回哪些记录，表 7.1 所列就是不同的筛选值对筛选结果的影响。

表 7.1 不同筛选值对筛选结果的影响

选 择	返回的记录	示 例
字段的整个内容(或将插入点放置在字段中,而不进行任何选择)	该字段的整个内容与所述值匹配	在城市字段中选择“柏林”,筛选将返回所有城市为“柏林”的记录
从字段的第一个字符开始的部分值	该字段值起始部分与所述值相同	在公司名字段值为“France restauration”的字段中选择“Fran”,筛选将返回所有公司名以“Fran”开始的记录,例如 Franchi S.p.A. 和 Frankenversand
字段第一个字符以后的部分值	整个字段或字段中的任何部分与所述值相同	在公司名字段为“Old World Delicatessen”的字段中选择“Del”。筛选将返回所有在公司名字段的任一处包含“del”的记录,例如 Ernst Handel, Gaier del gastrono 和 Que Delia

三、输入筛选

输入筛选是指使用“筛选目标”框中输入的值或表达式,以查找符合指定值或表达式的所有记录。

在“数据表”视图或“窗体”视图下,在要作为筛选准则的字段上单击鼠标右键,Microsoft Access 显示出快捷菜单;快捷菜单上出现“筛选目标”框,在框中输入要筛选的文本,再按 Enter 键,窗体中显示满足准则的记录。当用户在收货人中单击右键,在“筛选目标”框中填入 Around the Horn,如图 7.17 所示。运行窗体中显示所有收货人是 Around the Horn 的记录。

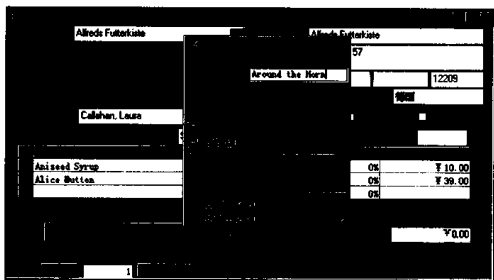


图 7.17

四、高级筛选

“按选定内容筛选”、“按窗体筛选”、“输入筛选”是筛选记录最容易的方法。如果可以在窗体、子窗体或数据表中找到并选择想要筛选记录包含的值的实例,可使用“按选定内容筛选”。如果要从列表中选择所需的值,而不想浏览数据表或窗体中的所有记录,或

若要一次指定多个准则，可使用“按窗体筛选”。如果焦点正位于窗体或数据表的字段中，而恰好需要在其中输入所搜索的值，或要将其结果作为准则的表达式，可使用“输入筛选”。而如果是更复杂的筛选，则可使用“高级筛选/排序”。

假设要在“定单”窗体记录中筛选出收单人为 Around the Horn，销售人为 Callahan Laura 或 Buchanan Steven 的所有定单。在“数据表”视图或在“窗体”视图中打开某个窗体，然后在“记录”菜单上单击“筛选”，然后单击“高级筛选/排序”，显示了筛选窗口。此窗口与查询窗口十分相似，在窗口上部框中已显示了与该窗体有关的表。将需要指定用于筛选记录的值或准则的字段添加到设计网格中。

如果要指定某个字段的排序次序，可单击该字段的“排序”单元格，然后单击旁边的箭头，选择相应的排序次序。Microsoft Access 会首先排序设计网格中最左边的字段，然后排序该字段右边的字段，以此类推。

在已经包含字段的“准则”单元格，可输入需要查找的值或表达式。此例中，用于筛选记录的值或准则的字段有定单查询中的公司名称和销售人字段，用鼠标单击相应字段并拖到设计网格中，并且在准则行中输入相应的准则。如图 7.18 所示。

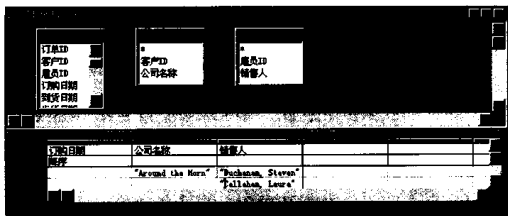


图 7.18

运行筛选时，单击工具栏上的“应用筛选”按钮以执行筛选。

在保存窗体时，Microsoft Access 将同时保存筛选，所以在下次打开表或窗体时，如果需要，可以重新应用此筛选。

五、用查询作为筛选

可以仅保存表、查询或窗体的某个筛选，也可以应用所创建的查询作为附加筛选。如果要使用查询作为筛选，必须符合某些特定要求。

如果要使用查询作为筛选，则必须符合某些特定的要求，即该查询要满足：

- (1) 必须与应用该查询作为筛选的数据表或窗体基于同一个基表或查询；
- (2) 不能包含其他表或查询；
- (3) 必须是选择查询（不能是交叉表查询或操作查询）；
- (4) 不能包含对字段中的值进行求和、计数或其他类型合计运算的字段。

要在窗体上应用现有的查询作为筛选。首先在工具栏中，单击“按窗体筛选”按钮，切换到“按窗体筛选”窗口下或在“记录”菜单上，指向“筛选”命令；然后单击“高级筛

选/排序”命令，切换到“高级筛选/排序”窗口下；然后再单击工具栏上的“装载查询”按钮，或单击鼠标右键，此时显示如图 7.19 所示的快捷菜单。在菜单中选择“从查询装载”，此时 Microsoft Access 就会显示“筛选”框，如图 7.20 所示。在“筛选”框中选择要作为筛选的那个查询，然后单击“确定”按钮，再单击工具栏上的“应用筛选”按钮执行筛选。

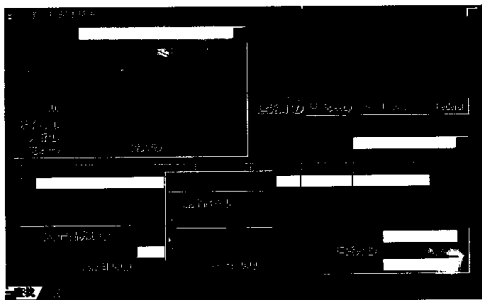


图 7.19

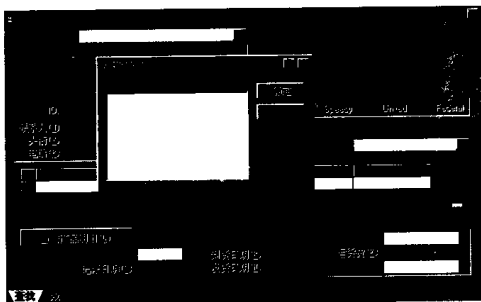


图 7.20

7.6 从窗体中打印数据

在窗体中的内容可以直接打印输出。当用户打开要打印的窗体后，单击工具栏中的“打印预览”按钮，先预览将要输出的窗体，要是不满意，用户还可以修改；满意了，再按打印按钮打印输出，要是需要设置页面或打印机，选中“文件”菜单中的“页面设置”命令或“打印”命令。图 7.21 所示为打印预览“订单”窗体时的画面。

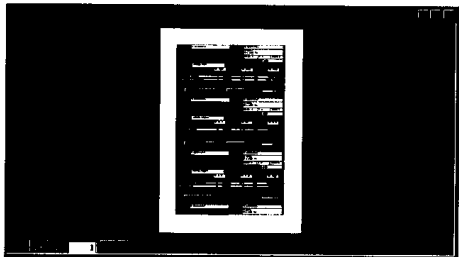


图 7.21

第八章 创建窗体

从日常应用的观点出发,窗体是用户在 Microsoft Access 中建立的对象中最重要的一个。窗体是用户运行应用程序时,时时能看到和操作的对象,本章将介绍如何建立窗体以及如何修饰窗体。

8.1 窗体的设计视图

在进行窗体的设计之前,必须先打开窗体的“设计”视图。

将焦点置于“数据库”窗口上,单击“窗体”按钮,然后选中“新建”按钮,Microsoft Access 会显示一个“新建窗体”对话框,如图 8.1 所示。在此对话框中,用户可以选择建立窗体的方法,共有七种方法。

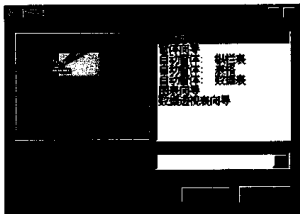


图 8.1

- 设计视图——是自主创建一个窗体。
- 窗体向导——按用户选定字段创建所需窗体。
- 自动窗体: 纵栏表——向导自动创建一个纵栏窗体。
- 自动窗体: 表格——向导自动创建一个表格窗体。
- 自动窗体: 数据表——向导自动创建一个数据表窗体。
- 图表向导——这个向导使用图表创建窗体。
- 数据透视图向导——向导使用 Excel 数据透视图创建窗体,在窗口下部需要用户选定该窗体的数据来源。

为了说明“设计”视图,先以“设计”视图方式创建窗体。选中“设计”视图,再单击“确定”按钮,则 Microsoft Access 显示如图 8.2 所示的“设计”视图。

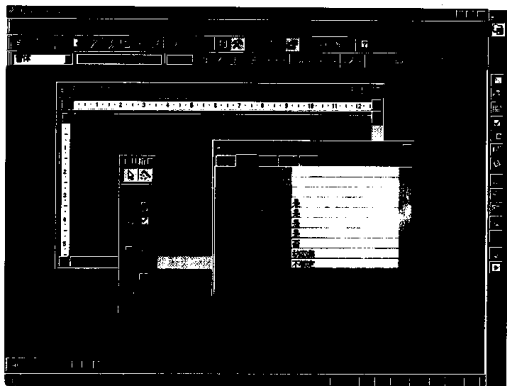


图 8.2

8.1.1 设计视图下的工具栏

“设计”视图下有其特有的一些工具，这些工具是 Microsoft Access 为了方便在“设计”视图下进行窗体的设计而显示的，如图 8.3 所示。

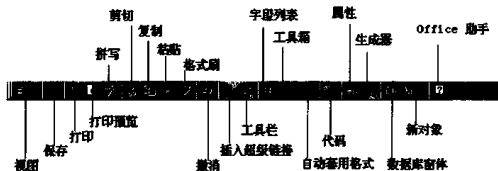


图 8.3

下面分别介绍这些工具。

● 视图按钮——单击此按钮，窗体以“窗体”视图显示。单击按钮右侧的箭头，下拉显示了窗体的三种视图，选中一种，Microsoft Access 以此种视图显示该窗体。

● 保存按钮——单击此按钮，保存用户对窗体所作的修改。

● 打印按钮——按下此按钮，用户可以将显示的窗体打印出来。

● 打印预览按钮——按下此按钮，屏幕上将出现当前窗体的“打印预览”画面，用户可以查看打印出来以后的样本，如果不满意，还可以作修改。

- 工具栏按钮——单击此按钮，在工具栏区内，显示 Web 工具条。
- 插入超级链接按钮——单击此按钮，Microsoft Access 将显示一个对话框，帮助用户创建一个超级链接。
- 字段列表按钮——当用户选择了某一表或查询作为窗体的数据来源，单击此按钮，屏幕上显示工具箱栏。
- 自动套用格式按钮——单击此按钮，Microsoft Access 会帮助用户调整窗体的画面。
- 代码按钮——单击此按钮，显示该窗体的代码。
- 属性按钮——单击此按钮，显示属性框。
- 生成器按钮——单击此按钮，Microsoft Access 弹出选择生成器窗口，选中相应的生成器，编辑用户所需的代码或表达式。

8.1.2 工作区

由图 8.2 中可以看到，Microsoft Access 只打开了一个空白的窗体主体区。前面介绍过，窗体中还有页眉与页脚。如果用户需要加上页眉或页脚，单击“视图”菜单中的“窗体页眉/页脚”命令，Microsoft Access 就会为工作区加上页眉与页脚。如果窗体不够大的话，用户可利用鼠标移动窗口与工作区。图 8.4 所示即为带的页眉与页脚的窗体工作区。

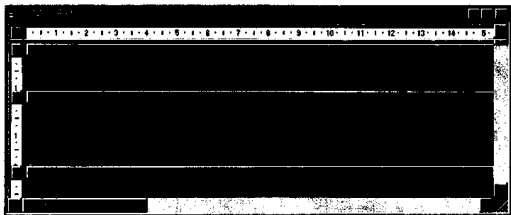


图 8.4

在图 8.4 中，用户还可以看到，在工作区中还有网格坐标和标尺，这是 Microsoft Access 为了方便用户在设计窗体时放置控件而提供的。如果要移走网格坐标和标尺，只要单击鼠标右键，在显示的快捷菜单中选择相应的命令即可。

8.1.3 工具箱

在窗体设计过程中，用户使用最频繁的是“工具箱”，它可以说是窗体设计的“命令中心”。在前面的“订单”窗体中可以看到，窗体内充满了各种大大小小的控件，将这些控件放置在窗体上必须通过工具箱来完成。单击工具栏中的“工具箱”按钮，工具箱就会出现，图 8.5 所示即为工具箱。

拖拉工具箱的标题条可以在屏幕上移动工具箱，也可以通过拖拉工具箱的边或中心来改变工具箱的形状。例如，要是用户喜欢将工具箱放在屏幕的上部，则工具箱移到屏幕上方后，Microsoft Access 就会将它作为工具条停



图 8.5

在那里。要关闭工具箱，只需再单击工具栏中的“工具箱”按钮即可。

工具箱中列出了窗体中常用的工具。

1. 指针工具

用来选定某一个文件，被选定的文件会变成当前的“工件”文件，以 Window 的术语来说，就是成为当前的“焦点”。以后所做的各种动作都将针对这个焦点文件来运行，例如用户可以改变所选择文件的大小、编辑内容或移动其位置。

2. 文本框

用来产生显示文本、数据、日期、时间或备忘录的数据框。用户可以将文本框内的数据内容与当前打开的数据表或数据查询的某一区域关联，利用此文本框来更新域内的数据内容。另外，文本框内的数据也可以是利用一表达式运算后的结果。

3. 选项组

利用此选项组可以包含一个或一个以上的切换按钮、选项按钮以及选项框。用户可以分别赋予每一个对象一个特定的数据值内容。在同一个组内的文件一次只能选择一个，而被选到的特定数值就成为这了选项组的代表数据值。同样，用户可以将选项组与当前打开的表格对象或查询对象的某一域相关联，利用此选项组来更新域内的数据内容。

4. 切换按钮

当表格内的参数数据有逻辑性的选项时，例如，On/off、True/False 或 Yes/No 选项，用户可以使用切换按钮来帮助数据的输入更直接、容易。当切换按钮被按下时，它的值为 1；若被按跳起来时，它的值为 0。另外，如果切换按钮被摆在选项组中时，它可以被指定一个特定的数值；当它被按下时，该数值就代表相应选项组的值。而且在同一个选项组中有多个一个以上的切换按钮，当用户按下其中一个切换按钮时，任何先前按下的切换按钮都会被设置为 Off 的状态。

5. 选项按钮

当表格内的参数数据有逻辑性的选项时，例如，On/Off、True/False 或 Yes/No 选项，用户可以使用选项按钮来使数据的输入更直接、容易。当选项按钮被按下时，它的值为 1；若被按跳起来时，它的值为 0。另外，如果选项按钮被摆在选项组中时，它可以被指定一个特定的数值；当它被按下时，该数值就代表相应选项组的值。而且当有多于一个选项按钮时，任何先前按下的选项按钮都会被设置为 Off 的状态。

6. 复选框

当表格内的参数数据有逻辑性的选项时，例如，On/Off、True/False 或 Yes/No 选项，可以使用复选框来使数据的输入更直接、容易。当复选框被设置时，它的值为 1；若被重设时，它的值为 0。另外，如果复选框被摆在选项组中时，它可以被指定一个特定的数值；当它被按下时，该数值就代表整个选项组的值。而且在同一个选项组中有多个一个以上的复选框时，当用户按下其中一个复选框时，任何先前按下的复选框都会被设置为 Off 的状态。

7. 组合框

组合框包含一个可以编辑的文本框和一个有可供选择的数据列表框。这个数据列表框的内容是来自组合框的行数据源属性特性参数，或是来自一个数据表或数据查询的某一域内容。组合框的数据列表，可按下文本框右端的向下箭头的按钮符号将它打开。如果这个组合框的内容关联到某一数据域时，用户可以利用组合框中的数据列表来选择输入该数据域的内

容。

8. 列表框

列表框的内容是来自该列表框的行数据源属性特性参数，或是来自一个数据表或数据查询的某一域内容。如果这个列表框的内容关联到某一数据域时，用户可以利用数据列表来选择输入该数据域的内容。

9. 子窗体/子报表

使用子窗体/子报表将另外一个数据表格放置在当前的数据表格上，用户可以将子窗体中显示的数据关联到主表格所显示的数据。

10. 非结合对象框

利用非结合对象框可以将具有“对象链接嵌入”功能的声音、图像或图形的数据对象放入当前的数据表格中，而此对象只是属于表格的一部分，并不会跟某一表格对象或查询对象的数据关联。

11. 结合对象框

利用此结合对象框可以将具有“对象链接嵌入”功能的声音、图像、或图形的数据对象放入当前的数据表格中，而此对象是属于表格中所显示数据的一部分，并跟某一表格对象或查询对象的数据有所关联。

12. 直线

利用直线可以在表格上绘制一些直线来使表格更美观。

13. 矩形

利用矩形可以在表格上绘制一些方框或是填满颜色的方块。

14. 分页符

使用分页符可以定义多页数据表格的分页位置。

15. 命令按钮

在数据表格上，用户可放置一些命令按钮，当某一命令按钮被按下时，其对应的 Microsoft Access 的宏对象或是 Visual Basic 的函数就会被打开。

16. 控件文件向导

按下这个切换按钮可以打开控制文件向导。当这个按钮是处于按下的状态时，每当用户建立一个新的选项组、组合框、列表框或者命令按钮时，这个向导将帮助用户输入控制文件特性参数。

17. 标签

可以使用标签控件来展示单个集合中的多页信息，这对于处理可分为两类或多类的标签是特别有用的。

18. 图像控件

可以使用此控件向窗体内加入图片。

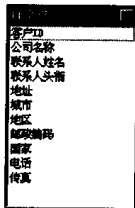
19. 其他控件

单击此按钮，Microsoft Access 会显示已加载的所有 Microsoft Access 控件。

8.1.4 字段列表

一般情况下，窗体都是基于某一表或查询的，因此，窗体内的一些控件要显示的也就是表或查询中的字段值。图 8.6 显示了窗体中的字段列表。单击工具栏中的“字段列表”按钮即可显示字段列表。

如果要在窗体内创建文本框来显示字段列表中的某一字段，用户只需用鼠标单击该字段，并拖到窗体内，窗体自动创建一文本框并与此关联。



8.6

8.1.5 属性窗口

窗体的每个部分或窗体中的每个控件，都有它们自己的特有属性，用户可在“属性”窗口中设置这些属性。例如，用户要显示窗体的属性，在窗体内单击，注意，应该单击窗体内不被主体、页眉、页脚覆盖的区域。然后，单击工具栏中的属性按钮，则显示如图 8.7 的属性窗口。

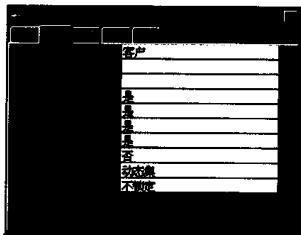


图 8.7

8.2 创建窗体

前面已经介绍了窗体的基本概念与操作，现在以一个例子来介绍如何创建窗体。

设计一个窗体，在窗体内显示客户的所有信息，如客户 ID、公司名称、联系人姓名、头衔、地址、城市、地区、邮政编码、国家、电话、传真。

先打开一个空白窗体，在“创建新窗体”对话框中输入的数据源是客户表，如图 8.1 所示。再单击确定按钮，则新窗体的画面如图 8.2 所示。

如果字段列表没有显示,则单击工具栏上的“字段列表”按钮,显示字段列表。如果没有“属性”窗口,则单击工具栏上的“属性”按钮,显示属性列表。如未显示“工具箱”窗口,则单击工具栏上的“工具箱”按钮。“工具箱”窗口可以以工具条方式存在,只要用鼠标单击“工具箱”窗口的标题栏,并移动到工具栏中。

下面介绍向窗体内加入控件。

8.2.1 向窗体内添加控件

向窗体内添加控件可分为两种情况。

一种是与客户表中的字段有关的控件，即该控件中显示字段中的值；另一种是与客户表无关的控件，例如只是显示标题的。

添加与字段相关的控件十分容易，这类控件主要是文本框。在字段列表中，选中需要显示的字段，用鼠标单击并拖动到空白的窗体主体区中。如果选择的字段很多，可以在用鼠标选择的同时一起按下鼠标右键与 Ctrl 键，这样就可以同时选中多个字段。如果要选择的字段是连在一起的，那么操作就方便了，只需先要在要选择的第一个字段上单击一下，然后到要选的最后一个字段，同时按下 Shift 键与鼠标左键，这样就会选中之间所有的字段，再拖动这些字段到窗体中。

每一个字段被拖到窗体内时，Microsoft Access 都自动将其设置为文本框，并且这些文本框的宽都是一样的，文本框与它的标题之间有大约 25.4 mm (1 in) 的间隔，而且文本框的标题就是字段的名称。

要添加与字段无关的控件，必须从工具箱中选择相应的控件。例如，需要添加一标题，选用标签控件。单击工具箱中的标签控件，然后将鼠标指针移到窗体内，移至放置该标签的区域，单击左键，拖动，在窗体内出现一矩形块，拉到大小合适时，松开按键。此时用户必须在矩形内输入文本，若不输入，创建标签无效。

此例中，需要将客户表中所有字段拖至窗体内，并且还要在页眉区创建一标签，标签名为“客户”，如图 8.8 所示。

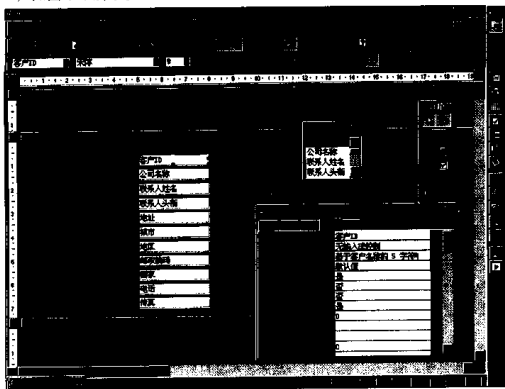


图 8.8

8.2.2 更改控件尺寸与移动控件

由于缺省时 Microsoft Access 建立文本框的大小为 25.4 mm (1 in)，大小并不一定对用户合适，特别是要使文本中的字变大时，文本框是不够大的，因此改变控件尺寸的操作十分常见。

更改控件尺寸和移动控件有两种方法，一种是直接在窗体内更改，另一种方法是利用“属性”窗口。

直接在窗体内更改，首先必须选中所要更改的控件。用鼠标单击控件，此时控件周围出现 8 个调整用的小方块，称为调整方块，如图 8.9 所示。不同位置的调整方块有不同的作用，在控件左上角较大的方块，是用来移动控件的，而其他方块是用来控制控件大小的。当鼠标指针指向移动位置用的调整方块时，指针会变成一只指着食指的小手。用户可以按下鼠标左键，移动控件到目标位置上，松开左键放置控件。此种方法可以将文本框的标签与文本分开移动。如果要移动整个文本框，移动鼠标指针到控件的边界，此时鼠标指针变成一只小手，按下鼠标左键，并拖动控件到目标位置上。

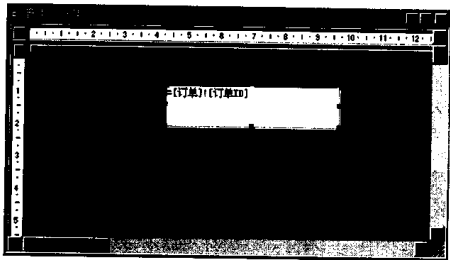


图 8.9

要改变大小时，移动鼠标指针，当指针变成多箭头时，按下左键，拖动箭头，当大小符合要求时，放开左键。在改变尺寸时，不同方块有不同作用，左右两侧的方块是用来改变长度的，而右下角的方块可以同时改变长与宽。

对于每一控件，在“属性”窗口中都有控制控件位置与尺寸的属性：左边距、上边距、高度、宽度。这些属性在“属性”窗口中的格式页中，如图 8.10 所示。更改这四个属性，也可以改变控件的尺寸与位置。

更改大小及位置后的“窗体”如图 8.11 所示。

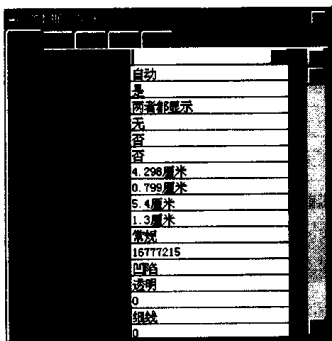


图 8.10

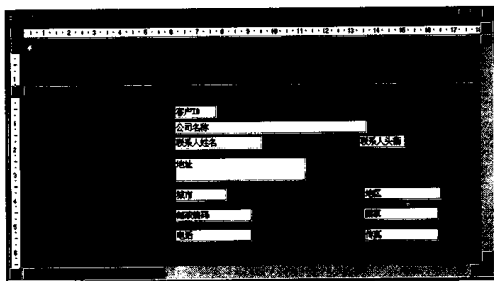


图 8.11

8.2.3 设置属性

在设置完控件的大小与位置后，还需设置一些其他的属性，这些属性在“属性”窗口中。

由于客户 ID 是主关键字，不希望用户更改，为了禁止选择客户 ID 控件，应把该控件的“是否锁定”属性设为“是”，如图 8.12 所示。

用户还可以利用属性窗口更改控件的颜色，文本的字体、大小、颜色等属性。

在本书的后面将会介绍，如果用户要编写宏或 Visual Basic 代码，就需要在相应窗体或控件的属性表中设置相应的属性。例如图 8.13 中就将属性设置为 [事件过程]。

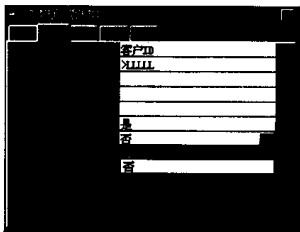


图 8.12

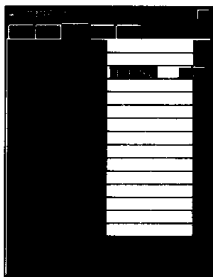


图 8.13

当更改完所有要更改的属性以后，用户可以单击“视图”按钮，在“窗体”视图下查看窗体的设计是否符合要求。图 8.14 显示了修改好的窗体。

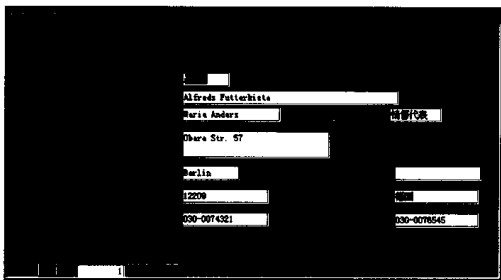


图 8.14

将窗体保存，只需单击工具栏中的“保存”按钮，然后在 Microsoft Access 弹出的如图 8.15 所示的“保存为”对话框中输入窗体的名字，按“确定”按钮，保存窗体。



图 8.15

8.3 使用窗体向导

由上一节中设计窗体的步骤可以看出,要自己设计一个窗体将是十分繁琐的。这一节将介绍利用 Microsoft Access 所提供的窗体向导,实现上一节中的例子。

首先在“创建新窗体”对话框中选中“窗体向导”,再在下面的数据源中输入“客户”,然后按“确定”按钮。

Microsoft Access 会显示如图 8.16 所示的对话框。在对话框中,用户可以选择所需的字段。当用户选定字段后,单击“>”按钮,可将该字段选到右边的列表框中;“>>”按钮可将左边可选字段全部选中,放到右边列表框中。此例中,由于需要显示所有的字段,因此单击“>>”按钮,然后单击“下一步”按钮,进入下一个画面。

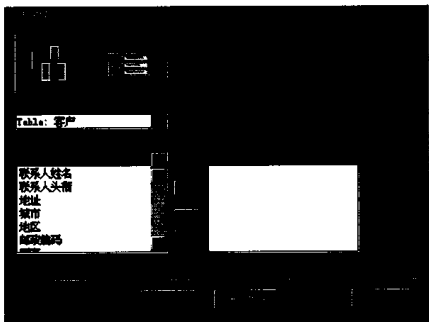


图 8.16

在图 8.17 所示的画面中,用户可以选择窗体采用的布局,共有四种:纵栏表、表格、数据表与调整表。我们选择纵栏表,再单击“下一步”按钮。

在图 8.17 中,用户可以选择窗体的样式,包括窗体的背景。用户可以在此选择自己喜欢的窗体命名,并可以选择创建完窗体之后是打开窗体还是修改窗体,如图 8.18 所示。然后,单击“完成”按钮,完成窗体的创建。

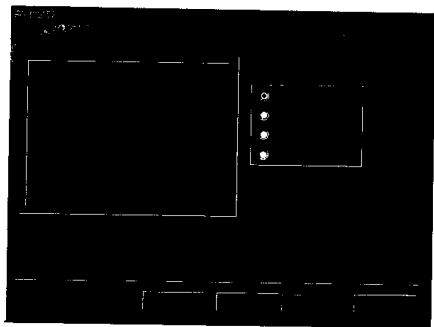


图 8.17

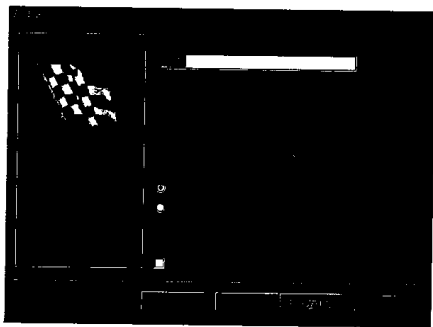


图 8.18

完成创建后的窗体如图 8.19 所示。

由进程可以看出，利用窗体向导创建窗体十分方便。用户应尽量利用它创建初步窗体，然后再进行修改。

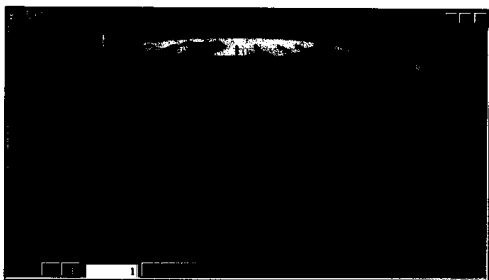


图 8.19

8.4 修饰窗体

如果窗体的功能只限于数据输入的话，那么在前几节中所提到的设计法则就已经足够使用了。在这一节中，将提出一些让用户在长期操作数据输入窗体时，能够更舒适、更不易输入错误数据的设计方式。另外，还介绍了在窗体中所用到的控制文件以及如何来设置它们的特性参数。

对于窗体而言，美观是最重要的，因此必须对窗体进行修饰，而在窗体中主要是控件。下面主要介绍对控件进行修饰。

8.4.1 控件位置

首先必须放置好控件的位置，调整单个控件的位置与大小在以前已经介绍过，这里要介绍的是整体上控件的位置。

调整整体上控件的位置有以下几种方式。

1. 使用网格对齐控件

在“设计”视图中打开相应的窗体，如果没有显示网格，请单击“视图”菜单上的“网格”命令，然后选择要调整的控件，并且在“格式”菜单上指向“对齐”命令，然后再单击“对齐网格”命令。

2. 控件互相对齐

在“设计”视图中打开相应的窗体，然后选择想要调整的控件，只能选择在同一行或同一列的控件。在“格式”菜单上，指向“对齐”命令，然后选择下列命令其中一项：

- 靠左，把控件的左缘与最左边的控件左缘对齐；
- 靠右，把控件的右缘与最右边的控件右缘对齐；
- 靠上，把控件的上缘与最上面的控件上缘对齐；
- 靠下，把控件的下缘与最下面的控件下缘对齐；

如果选定的控件在对齐之后可能重叠, Microsoft Access 并不会将它们重叠, 而会将这些控件的边相邻排列。

3. 平均间隔控件

在“设计”视图中打开相应的窗体, 在视图选择要调整的控件 (至少三个)。对于有附属标签的控件, 如文本框, 则只选择控件, 而不要选择其标签。然后在“格式”菜单上, 指向“水平间距”或“垂直间距”命令, 然后再单击“相同”命令。

Microsoft Access 将这些控件等间隔排列, 实际上只有位于中间的控件才会调整, 而顶层与底层的控件位置不变。

4. 增加或减少控件之间的间距

在“设计”视图中打开相应的窗体, 在视图选择要调整的控件, 然后在“格式”菜单上, 指向“水平间距”或“垂直间距”命令, 然后再选择“增加”或“减少”命令。在控件之间距增加或减少时, 最左侧 (水平间距) 及最顶端 (垂直间距) 的控件位置不变。

在图 8.14 所示的窗体中, 由于控件的位置未经过调整, 特别是一行中有两个文本框, 它们没有对齐。图 8.20 所示的窗体是经过调整以后的, 用户可能已经发现, 它比未经过调整的要美观多了。

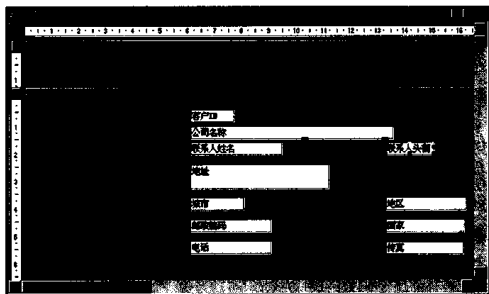


图 8.20

8.4.2 控件中的文本

控件中的文本是控件的脸面, 设置好看的文本会使窗体增色不少。调整文本可分为以下几个方面。

1. 更改控件中的文本字体与字号

单击含有将更改文本的控件, 然后在“格式”工具栏的“字体”框中, 单击要使用的字体。更改字体时, 所做的更改将应用于控件中的所有内容。在“格式”工具栏的“字号”框中, 输入要使用的字号。更改字号时, 所做的更改也将应用于控件的所有内容。在控件中, 不能更改个别字符的字号与字体。如果要调整控件的大小以适合新的字体, 可指向“格式”菜单中的“大小”命令, 然后单击“正好容纳”。

2. 更改控件中文本的对齐方式

单击含有要更改其文本的控件，然后在“格式”工具栏上，单击“居中”按钮使文本居中显示，“右对齐”按钮将文本右边对齐，或“左对齐”按钮将文本左边对齐。如果没有选择任何对齐按钮，则控件将使用常规对齐方式，即文本左对齐而数字右对齐。

3. 更改控件中的文本颜色

在窗体“设计”视图中打开窗体，单击含有要更改其文本的控件，然后在“格式”工具栏上单击“字体/前景色”按钮旁的箭头，Microsoft Access 将显示调色板，选择调色板中的颜色。

如果要对其他控件使用相同的颜色，可以只单击“字体/前景色”按钮（不需要单击按钮旁边的箭头以从调色板中选择颜色）。如果要同时为多个控件设置不同的颜色，可单击“字体/前景色”按钮旁的箭头，然后单击颜色调色板的标题栏并将它拖动到其他位置。这样就不必在应用颜色时，每次都单击“字体/前景色”按钮。

虽然图 8.20 所示的窗体中控件的位置已经十分满意了，但字的大小不太合适，需要更改。图 8.21 所示即为更改后的“窗体”视图。

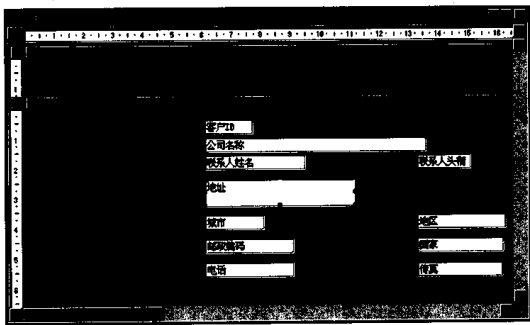


图 8.21

8.4.3 控件的特殊效果

1. 更改控件的背景色

单击要更改的控件，然后在“格式”工具栏上单击“填充/背景色”按钮旁的箭头，Microsoft Access 将显示调色板，在调色板中选择颜色。

如果要对其他控件或节使用相同的颜色，可以只单击“填充/背景色”按钮（不需要单击按钮旁边的箭头以从调色板中选择颜色）。如果要同时为多个控件设置不同的颜色，可单击“填充/背景色”按钮旁的箭头，然后选取颜色调色板的标题栏并将它拖动到其他位置。这样就不必在应用颜色时，每次都单击“填充/背景色”按钮。

如果调色板不包含所需的颜色，可按下述步骤创建自定义颜色。

打开控件的属性表,单击“前景颜色”(对控件中的文本)、“背景颜色”或“边框颜色”属性框。“属性”框旁边会出现“生成器”按钮,单击它,打开“颜色”的对话框。单击“规定自定义颜色”按钮以扩展对话框,如图 8.22 所示。

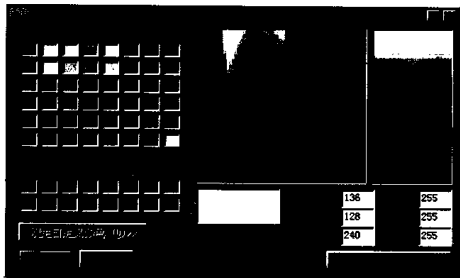


图 8.22

在对话框左边的“自定义颜色”框中选择一个颜色,然后单击颜色精选框中所需要的颜色。在颜色精选框中将显示全部可用的自定义颜色,用户可拖动亮度条旁边的箭头来定义颜色精选框中选颜色的亮度。Microsoft Access 在“颜色/实心”框中显示所创建的颜色。如果要查看与所创建的颜色最相近的实心颜色,可双击“颜色/实心”框的右边半框。最后单击“添加到自定义颜色”按钮,此时,在“自定义颜色”框中,Microsoft Access 将显示创建的自定义颜色。用户还可以在“红”、“绿”、“蓝”、“色调”、“饱和度”和“亮度”框中,键入数值来创建颜色。

2. 使控件变为透明

当控件为透明时,显示在控件中的颜色即为其控件、节或窗体的颜色。单击要更改的控件,然后在“格式”工具栏上,单击“填充/背景色”按钮旁的箭头,单击调色板顶部的“透明”按钮。需要说明的是,除列表框和切换按钮外的所有控件都可以更改为透明色,其中选项组的默认颜色即为透明。

3. 使控件具有凸起、凹陷、阴影、凿痕或蚀刻的效果

单击要更改的控件,选中此控件,然后在“格式”工具栏上单击“特殊效果”按钮旁的箭头,并且选取所需的效果。

如果要对其他控件使用相同的特殊效果,可以只单击“特殊效果”按钮(不需要再单击按钮旁边的箭头以选择其中的效果)。单击“格式”工具栏的“线条/边框宽度”按钮可以调整阴影的大小。

如果选择了“凸起”、“凹陷”、“蚀刻”或“凿痕”,Microsoft Access 将覆盖对“边框颜色”、“边框宽度”和“边框样式”属性的设置。如果选择了“蚀刻”,Microsoft Access 同时还会将文本框和组合框的“背景样式”属性设置为“透明”,并使列表框的“背景颜色”属

性设置与节的“背景颜色”属性值相同。

如果选择了凸起或凹陷的外观，但在窗体中却并未显示三维效果，可在“控制面板”中双击“显示器”选项，然后单击“外观”标签，并在“方案”框中单击“Windows 标准”。

4. 更改控件边框的宽度与颜色

要更改控件边框的宽度，只需单击要更改的控件，选中该控件，然后在“格式”工具栏上单击“线条/边框宽度”按钮旁的箭头，单击以选中所需的边框宽度。

更改控件边框的颜色，只需单击将更改的控件，然后在“格式”工具栏上单击“线条/边框颜色”按钮旁的箭头，Microsoft Access 将显示调色板，在调色板中选择用户所需的颜色。

如果要对其他控件使用相同的边框宽度，可以只单击“线条/边框宽度”按钮（不需要单击按钮旁边的箭头以从列表中选择宽度）。如果要对其他控件使用相同的颜色，可以只单击“线条/边框颜色”按钮（不需要单击按钮旁边的箭头以从调色板中选择颜色）。如果要同时为多个控件设置不同的颜色，可单击“线条/边框颜色”按钮旁的箭头，然后选取颜色调色板的标题栏并将它拖动到其他位置。这样就不必在应用颜色时，每次都单击“线条/边框颜色”按钮。

如果控件的“特殊效果”属性已设置为“凸起”、“凹陷”、“蚀刻”或“凿痕”，Microsoft Access 将在更改边框颜色与边框宽度的同时，将该设置更改为“平面”。

如果用户要隐藏窗体中的控件，只需在该控件的“可见性”属性框中单击“否”。

图 8.23 是向控件添加了特殊效果的客户窗体，它要比前一视图美观。

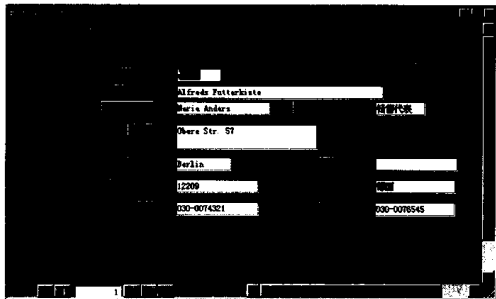


图 8.23

用户还可以向窗体中增加格式，单击工具栏中的“自动套用格式”按钮，屏幕上显示如图 8.24 的“自动套用格式”对话框。用户可在此对话框中选中喜欢的格式，在对话框中部的框中显示了该格式的示例。单击“确定”按钮向窗体中添加此格式。图 8.25 显示了套用格式后的窗体。

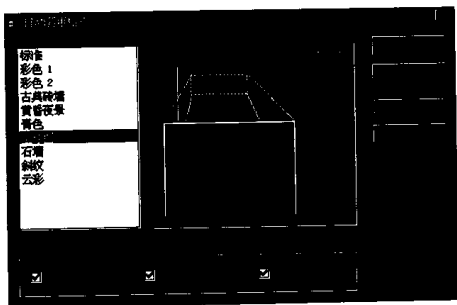


图 8.24

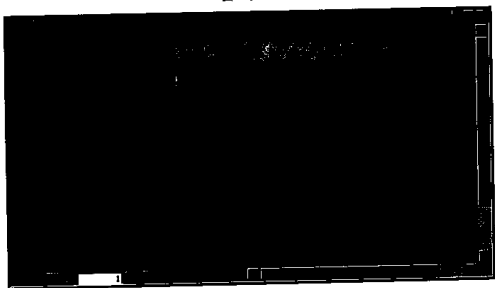


图 8.25

8.5 使用列表框与组合框

在许多情况下，从列表选择一个值要比记住一个值后键入它更快、更容易，选择列表框也可以帮助用户确保在字段之中输入的值是正确的。使用组合框，可以不需要太多的窗体空间，组合框就如同文本框和列表框的组合。

列表框中的列表是由数据行组成的，每行数据可以有一个或多个字段，每栏的字段标题可以有，也可以没有。如果一个多列的列表框是结合列表框，Microsoft Access 可以在其中的列保存数值。如果多列组合框是结合组合框，Microsoft Access 会将数值保存到其中一列。

可以使用非结合列表框或组合框来保存用于其他控件的值。例如，可以使用非结合列表框来限制另一列表框或自定义对话框中的值；也可以根据从列表框中选择的值，使用非结合

列表框查找一项记录。

如果在窗体中没有空间显示列表框，或者要在键入新值的同时，也能从列表中选择值，请使用组合框而不是列表框。

在组合框中输入文本或选择某个值时，如果该组合框是结合组合框，则输入或选择的值将插入到组合框所结合的字段内。

组合框有 LimitToList 属性，可以使用该属性控制列表中能输入数值或仅能在列表中输入符合某值的文本。

这两个控件各有优点。

- 列表框的优点——列表随时可见，并且控件的值限制在列表中可选的项目；如果要快速地移动到以特殊字母开头的第一个值，可以直接键入该字母；正在使用窗体输入或编辑数据时，不能添加列表中没有的值。

- 组合框的优点——由于列表在打开后才显示内容，所以在窗体中该控件使用较少的空间；可以键入组合框中值的开头几个字母，以便快速地查找到这个值，同时，也可以控制在列表中是否可以输入任何值或只可输入符合列表中的值。

因此，在窗体中用户可以用组合框或列表框替代一些文本框。例如，在前几节建立的窗体中，国家字段对应的文本框可用组合框替代。

在窗体的“设计”视图中，先删除国家字段对应的文本框，再按下工具箱中的组合框按钮，然后单击字段列表中的国家字段，按下国家域，拖到设计窗体的主表中原来国家框的位置，这时 Microsoft Access 在窗体中显示组合框向导对话框，如图 8.26 所示。



图 8.26

按向导提示的信息步骤完成组合框的创建。当用户单击确定按钮以后，窗体中出现一个关联到国家域的组合框。按上一节叙述的过程修改组合框控件。图 8.27 显示的是“窗体”视图下的画面。

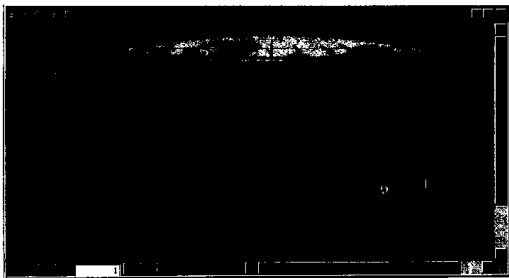


图 8.27

第九章 报表基础

前面学习了格式化和打印“数据表”视图中的表和查询，还可以利用这一方法产生一个信息简单列表的被打印的拷贝；利用窗体可以观察和修改数据，而且可以打印数据——包括来自几个表的数据。然而由于窗体的主要目的是允许用户观察显示在屏幕上的单一记录或相关记录的集合，并且它的外观很吸引人，因此，在数据库中打印和总结大量的数据集，窗体不是最好的方法。报表正是为了克服它们的缺点而形成的。

数据库有一个十分重要的作用就是可以综合一定范围内的数据，并打印或制作成报表。报表是经常使用的文件，特别是在企业中。因此在 Microsoft Access 中，报表是数据库对象中的六大对象之一。

9.1 报表的作用

报表是在数据库中建立数据抽取和计算的打印拷贝的最好方法，报表与其他打印数据方法相比，有以下两个主要优点：

- 报表能比较、总结和小计一套数据；
- 报表能建立用于产生清单、定单、邮递标志、产品介绍和其他输出。
报表是用来分组数据，独立地介绍每一分组和执行计算，具有以下功能：
- 可以为每一组定义分类表头与脚注；
- 可以执行复杂计算，不仅包括一组或一行，还包括交叉组；
- 除了页表表头和脚注以外，还可以定义一个完整的报表表头和脚注。

如同使用窗体一样，用户可以在报表的任何部分嵌入图形和图片，也可以在报表内嵌入子报表或子窗体。

9.2 浏览报表

下面，将利用 Northwind 数据库中的报表——“按季度汇总销售额”来浏览一下报表，对报表作一介绍。

9.2.1 工具条

首先要打开报表。单击“数据库”窗口中的报表页，选中该报表，然后用鼠标双击，打开报表。此时 Microsoft Access 显示如图 9.1 所示的“打印预览”视图。

“打印预览”下的工具条与其他视图下的工具条几乎完全不同，如图 9.2 所示。

- 视图按钮——单击此按钮，下拉显示“设计”视图与“打印预览”两项，再选择两

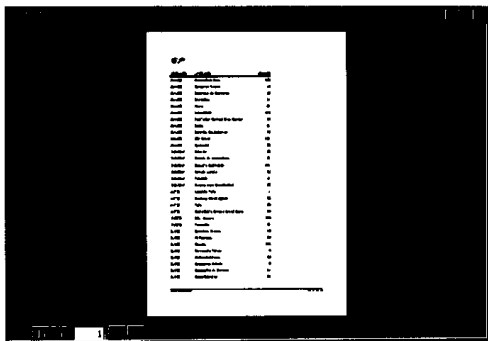


图 9.1

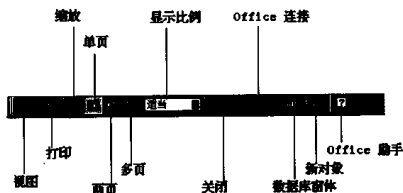


图 9.2

者之一，单击，可实现两种视图之间的转换。

- 打印按钮——单击该按钮，将报表送到打印机上。
- 缩放按钮——单击该按钮，能缩小以看到整个页的布局或放大看到细节。也可以把鼠标器指针放在报表的任何区域上，并单击它们以进行放大或缩小。
- 单页按钮——单击该按钮，在窗体内可以显示一页报表。
- 两页按钮——单击该按钮，在窗体内可以显示两页报表。
- 多页按钮——单击该按钮，在窗体内可以显示多页报表。
- 显示比例按钮——单击该按钮左侧的箭头，下拉显示列表框，其中列出了用户“打印预览”的比例。选中所需比例，Microsoft Access 将按此比例预览要打印的报表。
- Office 连接按钮——单击此按钮左侧的箭头，将显示用户可以用 Word 发布或用 Excel 分析此报表。单击其中的“用 Word 发布”按钮，可将报表输出作为一个 Word 文件存储，并可在 Word 97 中打开文件。单击“用 Excel 分析”按钮，用户可将报表输出到一个 XLS 格

式文件存储,并可在 Excel 97 中打开此文件。

- 关闭按钮——单击该按钮能关闭“打印预览”窗口,并返回以前的窗口。
- 数据库窗口按钮——单击此按钮,可回到数据库窗口。

9.2.2 报表的滚动

如果用户的窗口页面不大,那么只能看到部分的报表内容,所以用户需要用到窗口右边及下方的垂直及水平滚动条,来移动窗口内所显示的内容。

另外,在图 9.1 所示的报表窗口左下方有个“页数指示框”,如图 9.3 所示,其作用是翻阅报表的某一页,操作的方式与表格左下方的“记录指示框”很类似。为了依次向前移动一页,可以连续的单击箭头按钮一直到页码框的右边;也可以单击页码改变这个数字,并按 Enter 键跳到所要的页上。



图 9.3

9.2.3 打印预览页面

在图 9.1 所示的“打印预览”页面中,大致上可以划分为几个部分。

- 报表标题——这个报表标题是放在整份报表的最前面,可以利用报表标题来设计整个报表的封面。
- 报表注脚——报表注脚是放在整份报表的最后面,可以利用报表注脚来设计整个报表的封底。
- 页标题——通常也称为页首,放在一页报表的最上方。可以根据需要放置一些每一页固定要显示的数据,如:日期。
- 页注脚——通常也称为页尾,放在一页报表的最下方。可以根据需要放置一些每一页固定要显示的数据,如:页码。
- 分类标题——报表对象可以将数据分类后打印出来,在每个分类的类别内容开始打印前,可以加上一些简单的标题,当作类别的标题。如:类别的名称。
- 分类注脚——在每个分类的类别内容打印完毕后,可以加上一些简单的注脚,当作类别结束的说明。例如:类别内容数量的小计。
- 数据内容——根据所定义的内容顺序或类别范围,打印出数据内容。

9.2.4 子报表

正如可以在一个窗体中嵌入一个子窗体一样,用户也可以在一个报表中嵌入一个子报表。它对于计算一个报表中的总结信息以及在显示信息的报表内包括这个信息是很有用的。用户不需要两次定义这个总结报表,只需要把它作为一个子报表放在细节报表中,Microsoft Access 立即就把这个总结信息连接到相关细节中。

在 Northwind 数据库中,各年销售额报表就是一个会有子报表的报表。图 9.4 显示了“预览”视图下的销售额报表。图 9.5 显示了“预览”视图下的销售额子报表。从两图中可以看出,子报表是主报表中的一部分。

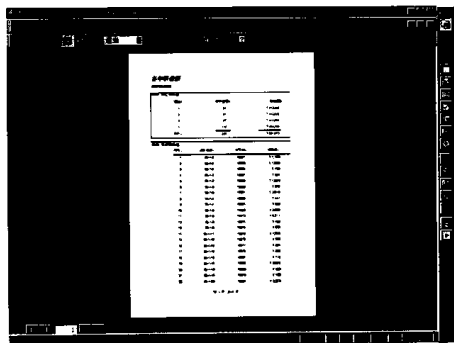


图 9.4

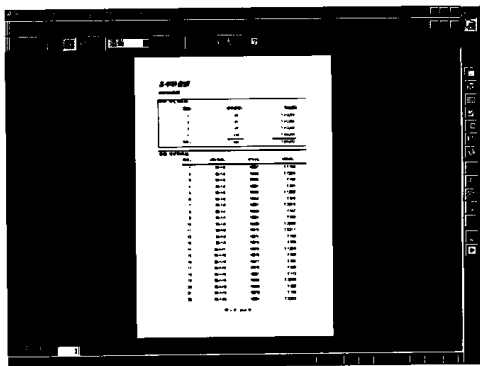


图 9.5

9.2.5 以不同的缩放比例预览报表

单击“打印预览”工具栏上“显示比例”框按钮旁的箭头，然后选择缩放比例或键入所需要的缩放比例；如果选中“适当”，Microsoft Access 将根据窗口大小来调整显示页的最佳的缩放比例。将鼠标指向表、窗体或报表，当鼠标变为放大镜时单击鼠标，可以在设置的缩放比例和“适当”选项之间切换。

9.2.6 报表中的对象

在报表中可以嵌入 OLE (Object Linking and Embedding)。嵌入和连接到报表中的对象一般是图形或图片。可以把一个图片或图形作为一个未连接对象嵌入到报表中，还可以将图片或图形作为一个来自数据库中的数据的未连接对象来连接它。

在定单数据库中的发货单报表中就有这样的图形，图 9.6 所示就是发货单报表。

发货单



Vinc et alcools Chevalier
59 rue de l'Abbaye
Reims 51100
法国

日期 1997年9月24日

发货行

Vinc et alcools Chevalier
59 rue de l'Abbaye
Reims 51100
法国

收货行

Vinc et alcools Chevalier
59 rue de l'Abbaye
Reims 51100
法国

图 9.6

9.3 打印报表

前面已经学习了观察打印预览中的报表基础，下面将介绍如何打印报表。

第一次打印报表以前，可能要检查页边距、页方向和其他页面设置的选项。

在“打印预览”或“设计”视图下打开报表（或从数据库窗口中打开），然后单击“文件”菜单中的“页面设置”命令，Microsoft Access 显示如图 9.7 的“页面设置”对话框。

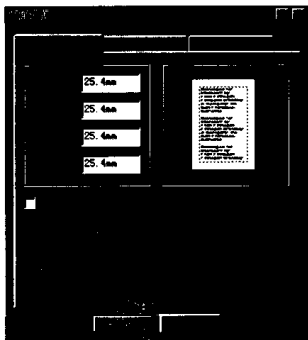


图 9.7

在对话框中单击下列标签，设置所需要的选项。

在“边距”标签，用户可以设置页边距并确认是否只打印数据。在“页”标签可以设置打印方向、页面大小和打印机型号。在“列”标签只设置窗体、报表和宏的列数、大小和列的布局。在选择完以后，单击“确认”按钮。

Microsoft Access 将保存窗体和报表页面设置选项的设置值，所以每个窗体或报表的页面设置选项只需设置一次。但是表、查询和模块每次打印时都要进行页面设置选项的设定。

在设置页边距、打印方向和其他的页面设置选项以后，用户就可以开始打印报表了。

在打印报表时，不要求报表是打开的，因此可以在数据库窗口中选定报表，当然也可以在“设计”视图、“打印预览”或“布局预览”中打开相应的报表，然后单击“文件”菜单中的“打印”命令，Microsoft Access 显示如图 9.8 所示的“打印”对话框。在“打印”对话框中进行以下设置。

- 在“打印机”下，指定打印机的型号。
 - 在“打印范围”下，指定打印所有页或者确定打印页的范围。
 - 在“份数”下，指定复制的份数和是否需要对其进行分页。
 - 如果用户只想打印到文件，则选中“打印到文件”复选框。
 - 单击“属性”按钮，Microsoft Access 会显示出打印机的“属性”对话框。在对话框中，用户可以选择纸张大小以及打印的效果。
 - 单击“设置”按钮，还可以进行页面设置。
- 在选择完以后，单击“确认”按钮开始打印。
- 如果要在不激活对话框的情况下打印报表，可直接单击工具栏上的“打印”按钮。



图 9.8

第十章 创建报表

创建一个报表很像建立一个窗体。在这章中，将使用操作窗体的许多技术，也将学习如何使用报表提供的特殊技术。

10.1 报表的设计视图

在设计报表之前，必须先打开报表的“设计”视图。

将焦点置于“数据库”窗口中，单击“报表”按钮，然后选中“新建”按钮，Microsoft Access 会显示一个“新建报表”的对话框，如图 10.1 所示。与“新建窗体”的对话框十分相似，在此对话框中，用户可以选择创建报表的方法。对话框中共有六种方式：

- 设计视图——不使用向导，用户自主在“设计”视图下创建报表；
- 报表向导——向导根据用户所选的字段来创建报表；
- 自动报表：纵栏表——向导自动创建一个纵栏报表；
- 自动报表：表格——向导自动创建表格报表；
- 图表向导——向导利用图表创建报表；
- 标签向导——向导创建的报表的格式是用于标签的。

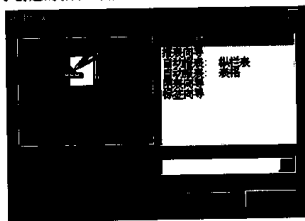


图 10.1

在对话框的下部，用户可以选定以上报表对象的数据来源，数据来源是数据库中的所有表和查询；用户也可以不选，那么就可以创建非结合型的报表。

为了能更加详细的说明如何建立报表，先在“设计”视图下创建报表。选中“设计视图”项，再单击“确定”按钮，则 Microsoft Access 显示了如图 10.2 所示的设计画面。

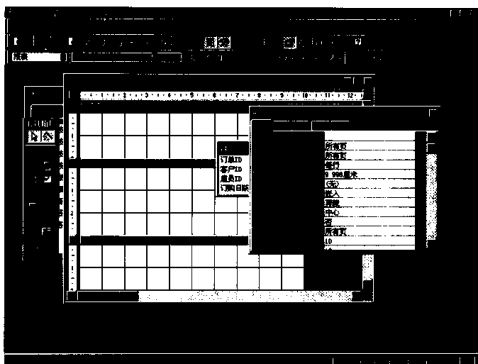


图 10.2

10.1.1 工具栏

同窗体的设计视图一样，也有其特有的一些工具按钮，如图 10.3 所示。



图 10.3

● 视图按钮——单击此按钮，报表以“打印预览”视图显示。单击按钮右侧的箭头，下拉显示了报表的三种视图：“设计”视图、“打印预览”与“版面预览”，选中其中一种，Microsoft Access 将以此种视图显示该报表。

● 保存按钮——单击此按钮，Microsoft Access 将保存用户对报表的修改。

● 打印按钮——单击此按钮，将此报表输出到打印机，但不显示“打印设置”对话框，而是以缺省设置打印输出。

● 排序与分组按钮——单击此按钮，Microsoft Access 将显示“排序与分组”对话框，在对话框内填入相应内容，进行数据的排序与分组。

“打印预览”按钮等其他工具与窗体设计视图中的按钮功能相同，不相同的按钮只有排序与分组按钮。

10.1.2 工具箱

在报表设计过程中，用户使用最频繁的是“工具箱”，它可以说是报表设计的“命令中心”。与窗体一样，报表也是由各种各样的控件组成的，将这些控件放置在报表上必须通过工具箱来完成。单击工具栏中的“工具箱”按钮，工具箱就会出现。图 10.4 所示即为工具箱，它与在窗体创建过程中出现的工具箱一样。

拖拉工具箱的标题条可以在屏幕上移动工具箱，也可以通过拖拉工具箱的边或中心来改变工具箱的形状，例如，要是用户喜欢将工具箱放在屏幕的上部，则工具箱移到屏幕上方后，Microsoft Access 就会将它作为工具条停在那里。要关闭工具箱，只需再单击工具栏中的“工具箱”按钮即可。

各种工具的使用方法，这里不再重复说明，与窗体创建一样。



图 10.4

10.1.3 工作区

Microsoft Access 打开报表的工作区时，在工作区中已有报表的主体与页面的页眉与页脚，除了页面的页眉与页脚以外还有报表的页眉与页脚。如果用户需要加上报表的页眉与页脚，只需单击“视图”菜单上的“报表页眉/页脚”命令，Microsoft Access 会自动在工作区上加上报表的页眉与页脚。如果报表的工作区不够大，用户可以利用鼠标移动报表窗口的边框和工作区的边界来改变工作区的大小。图 10.5 是带有报表和页面页眉与页脚的报表工作区。

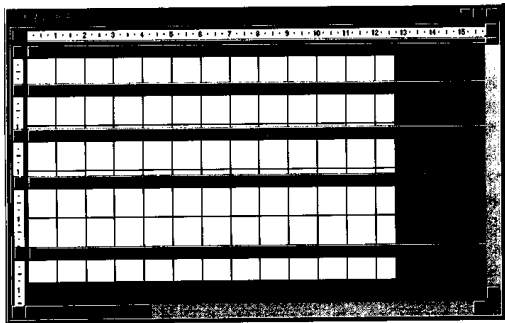


图 10.5

10.1.4 其他窗口

同窗体的“设计”视图一样，在报表的“设计”视图也有工具箱、字段列表与属性框。

一般情况下,报表都是以一定的表或查询作为它的数据基础的,因此都是与表相关联的,字段列表就是用来引出与此报表相关联的表或查询中的所有字段。要显示或隐藏字段列表,用户只需单击工具栏中的“字段列表”按钮。

如果要在报表内创建一文本框,来显示字段列表中的某一字段的数据,用户只需鼠标单击此字段,然后拖曳到报表中,报表就自动创建一个文本框,并且与此字段相结合。

同样,当用户需要更改或设置报表或报表中某个控件的属性时,就要用到报表的属性框。单击工具栏中的“属性”按钮,就可以控制属性框显示与否。

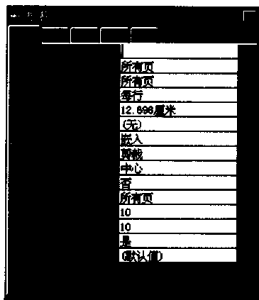


图 10.6 显示了报表的属性。

图 10.6

10.2 创建报表

这一节介绍在 Microsoft Access 中如何创建一个报表。

假设要创建一个报表,用来显示各类产品的种类名称。

首先打开一个空白报表。由于没有现成的表或查询作为数据源,在“新建报表”对话框中的数据源部分不输入任何内容,然后单击“确定”按钮,则 Microsoft Access 显示出空白报表设计画面。

由于没有确定结合数据源,因此必须先给报表创建数据源。用户可以用两种方法来创建数据源。

一种是回到“数据库窗口”内,只要单击 F11 键或单击工具栏上的“数据库窗口”按钮,即可。然后单击“查询”按钮,再单击“新建”按钮,准备创建一个查询,将此查询作为报表的数据源。

另一种是在报表设计画面中,如果已经有属性框,则单击报表,使属性框中的内容是报表属性。可以将鼠标指针移至报表的标题栏中,然后单击鼠标右键,选中属性命令。在属性框中选中“数据”标签,然后在数据源属性上单击一下,此时在此属性文本框右侧出现生成器按钮,而且还会在框内出现下拉按钮。单击此下拉按钮,可以从表与查询列表中选一个作为数据源。由于此例中没有现成的表或查询作为数据源,因此单击“生成器”按钮,则 Microsoft Access 显示了查询生成器,与建立新查询时一样,Microsoft Access 自动显示了“显示表”对话框。

下面可以按照建立新查询的步骤,选取字段列表中的字段,可以进行排序,设置准则,最后单击工具栏中的保存按钮,Microsoft Access 会弹出“另存为”对话框,用户可以为该查询输入名称,再单击确定按钮,完成新查询的生成。或者用户也可直接关闭“查询生成器”,此时 Microsoft Access 显示提示框询问用户是否保存此查询的 SQL 语句,选择是,则在属性框中的数据源属性变为一条 SQL 语句。若是前者,则是该查询的名称。

图 10.7 显示的是查询的设计画面，而图 10.8 显示了设置数据源属性后的属性框。此时，工具栏中的“字段列表”按钮已经可用，单击此按钮，字段列表框显示在屏幕上。下面开始报表的设计。

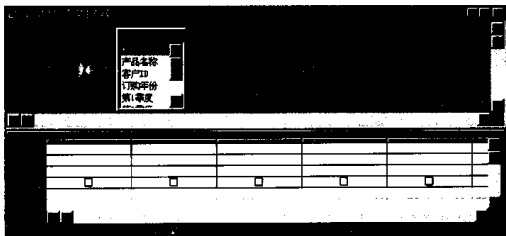


图 10.7

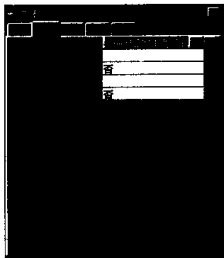


图 10.8

10.2.1 向报表工作区内添加控件

同窗体中一样，向报表中添加控件也可分成两种情况：一种是与字段列表中的字段相结合的控件；另一种是与字段不相结合的非结合控件，例如，用来显示标题的标签控件。

添加与字段相关的控件十分容易，这类控件主要是文本框。在字段列表中，选中需要显示的字段，用鼠标单击并拖动到空白的报表工作区。如果选择的字段很多，可以在用鼠标选择的同时一起按下鼠标左键与 Ctrl 键，这样就可以同时选中多个字段。如果要选择的字段是连在一起的，那么，移到要选的最后—个字段，同时按下 Shift 键与鼠标左键，这样就会选中两者之间所有的字段，再拖动这些字段到报表中。

每一个字段被拖到报表内时，Microsoft Access 都自动将其设置为文本框，并且这些文本框的宽度都是一样的。在文本框与它的标题之间有大约 25.4 mm (1 in) 的间隔，而且文本框的标题就是字段的名称。

要添加非结合的控件，必须从工具箱中选择相应的控件。例如，需要添加一标题，选用标签控件。选中工具箱中的标签控件，单击之，然后将鼠标指针移到报表内，至放置该标签的区域，单击左键，拖动，在报表内出现一矩形块，拉到大小合适时，松开按键。此时用户必须在矩形内输入文本，否则创建标签无效。

如果要使用向导来帮助创建控件，请确保按下了工具箱中的“控件向导”工具。可以使用向导来创建命令按钮、列表框、组合框、子窗体及选项组，还可以创建图表或数据透视表的控件。

有时用户需要创建计算控件。下面介绍创建计算控件的方法：

单击工具箱中要作为计算控件的控件类型工具。一般情况下使用文本框，它是经常用来显示计算数值的控件类型，但是也可以使用任何有“控件来源”属性的控件。在报表上，单击要放置控件的地点。然后执行下列操作之一。

- 如果控件是文本框，则可以直接在控件中键入表达式。
- 如果控件不是文本框，或即使控件是文本框，但是使用表达式生成器来创建表达式。

在选定了控件以后，若未打开属性表，则单击工具栏上的“属性”按钮来打开控件的属性表，接着在“控件来源”属性框中键入表达式，或单击“生成器”按钮来打开“表达式生成器”，按照以前叙述的方法来创建表达式。在创建表达式时要注意以下几点：在计算控件中，每个表达式前都要加上等号(=)运算符；如果在“控件来源”属性框中需要更多空间来键入表达式，请按 Shift + F2 键来打开“显示比例”框；如果是基于查询的窗体，可能要将表达式放入查询，而不是计算控件中。

在此例中，需要为列表中的类别名称、产品名称与库存量字段创建结合控件，再为产品数目与日期创建计算字段，还需要在报表页眉中创建一标签控件，如图 10.9 所示。

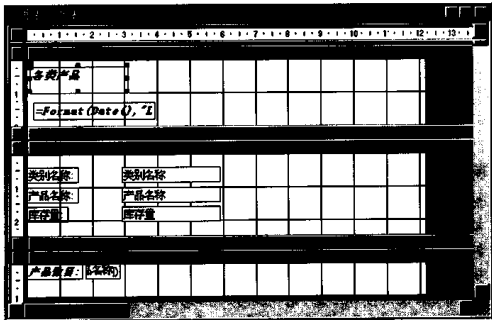


图 10.9

10.2.2 更改控件尺寸及位置

在用户最初创建报表时，对控件的位置及尺寸设置并不一定是十分精确的，因此必须对控件的尺寸和位置进行调整。

控件的尺寸调整在窗体设计中已经详细叙述过了，这里不再重复。

更改控件的整体位置时需要同时选中多个控件，下面是选择方法。

- 选择相邻的控件——在报表的“设计”视图中，可以指定所拖动包含控件的矩形是只需要接触到控件，或必须完全包围控件才能加以选择。请在“工具”菜单中单击“选项”命令，再单击“报表”标签，然后选择“选定行内”选项组中的所需选项。如果要从选择组中删除一个或多个控件，请按下 Shift 键，然后单击要删除的控件。

- 选择不相邻或重叠的控制——在报表的“设计”视图中，按下 Shift 键，然后单击每一个要选择的控件。

- 选中同一列网格中的所有控件——在竖标尺中单击标尺中相应的位置即可。

10.2.3 设置属性

在报表中，需要对其中的控件设置属性，例如，显示的字体、大小以及文本的位置、显示格式，用以修饰控件。

10.2.4 保存

单击工具栏中的“保存”按钮，Microsoft Access 弹出“另存为”对话框，在其中的文本框中输入相应的报表名称，再按“确定”按钮，保存窗体。图 10.10 显示了“另存为”对话框。



图 10.10

10.3 使用报表向导

上一节中设计的报表是十分简单的，但在设计时却十分烦琐。实际上，在 Microsoft Access 中设计报表还有一种更简捷的方法，就是利用 Microsoft Access 提供的报表向导。

在“新建报表”对话框中选中报表向导，并且在下面的数据源中选中该报表是基于表或查询；如果没有现成的表或查询则退出，首先创建查询。再单击“确定”按钮，Microsoft Access 显示如图 10.11 所示的对话框。在此对话框中，用户可以选择数据源。当数据源选定后，在下面的可选字段列表中显示所有字段，右边框为用户选中的字段列表。用户按“>”键则选中左边字段列表中的当前字段，按“>>”选中全部字段。

当选择好字段以后，单击“下一步”按钮，进入下一个画面，如图 10.12 所示。在该画面中，用户需要确定分组，分组的概念将在下一节中介绍。

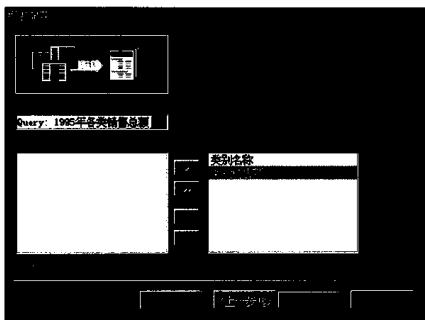


图 10.11

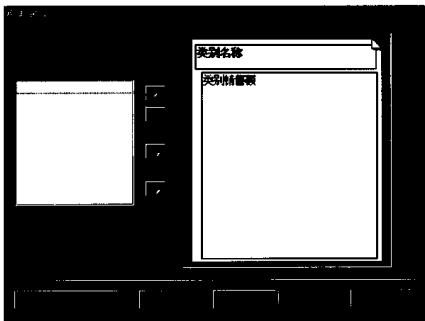


图 10.12

我们选择按类别名称进行分组。在完成分组选择以后，再单击“下一步”按钮，向导进入“排序”画面。用户可在此画面中确定在每个分组中按什么字段进行排序，最多可选10个字段，最上面的优先级越高，如图10.13所示。

再单击“下一步”按钮，用户可在此画面中挑选以何种方式布局报表，共有多种形式：纵栏表、表格、对齐，并且可以选择方向：纵向或横向。再单击“下一步”按钮，用户可以在此画面中选取自己喜欢的样式，同窗体中一样。再单击，进入最后一个画面，用户在此画面中需要输入报表名，并且可以选择在报表创建完以后，以何种方式继续操作，如图10.14

所示。

在图 10.15 中显示了创建完以后的报表“预览”视图。在创建此报表过程中，用户应可以感觉到，利用向导创建报表是十分方便的。

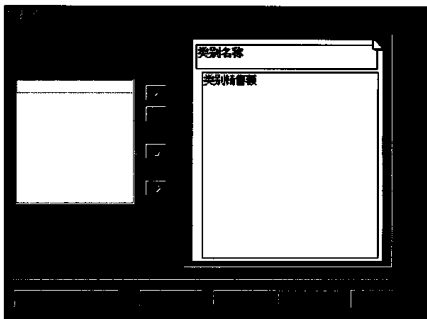


图 10.13

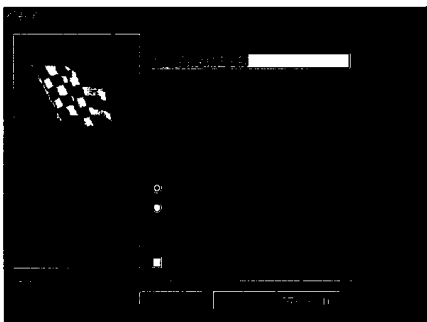
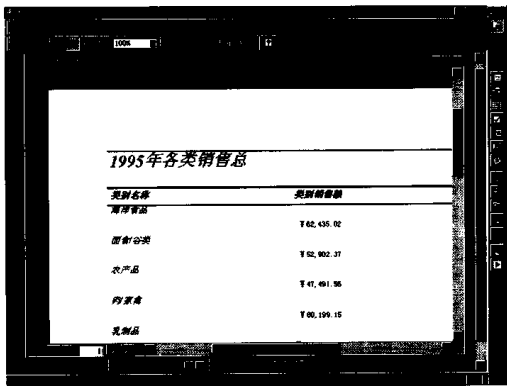


图 10.14



类别名称	类别销售数
海洋食品	¥ 62,435.02
面食谷类	¥ 52,902.37
农产品	¥ 47,491.95
肉蛋禽	¥ 90,199.15
乳制品	

图 10.15

10.4 报表类型

由“新建报表”对话框中，用户可以看出，报表有几种不同的类型。

最简单的报表是自动报表，一个自动报表只是简单地 will 字段名称和字段内容列表。自动报表也有两种形式，一种是纵栏表形式，另一种是表格形式。

纵栏式报表是指一个接一个地显示字段名和它们的内容。要创建纵栏式的报表，可在报表向导的布局画面中选择纵栏表，或在新建报表时选择“自动报表：纵栏表”，再选中相应的数据源，此时，Microsoft Access 自动创建纵栏表。图 10.16 显示的是通过“自动报表：纵栏表”向导创建的报表。这种布局不易阅读，对于较复杂的表或查询，利用这种形式是比较难看的。不过，假如要创建只包括想显示的几个字段的查询，并且将使用自动报表的数据源，自动报表就变得十分有用。

表格形式报表与表格相同，是一个接一个地显示每个字段的记录，这普遍应用于打印输出一个表中数据的报表。同样，可以在报表向导的布局画面中选中表格，使报表以此形式显示。当然，也可以选择“自动报表：表格”，并选中相应表或查询，来创建基于此表或查询的报表。图 10.17 显示的是表格形式的报表，显然它的形式非常简单，与表的“数据表”视图毫无差别。

邮件标签是另一种类型的报表。创建标签报表可以使用 Microsoft Access 提供的标签向导。在“新建窗体”对话框中选中标签向导，在下面数据源框中选中客户表，如图 10.18 所示，然后单击“确定”按钮，进入标签向导。

Microsoft Access 显示如图 10.19 的对话框，在对话框中用户可以选择标签。单击对话

类别名称	类别销售额
海洋食品	¥ 82,435.02
粮食谷类	¥ 52,902.37
农产品	¥ 47,491.56
肉禽蛋	¥ 80,199.15
乳制品	

图 10.16

类别名称	类别销售额
海洋食品	¥ 82,435.02
粮食谷类	¥ 52,902.37
农产品	¥ 47,491.56
肉禽蛋	¥ 80,199.15
乳制品	

图 10.17

框中的“自定义”按钮，显示“新标签大小”对话框，再单击“新建”按钮，Microsoft

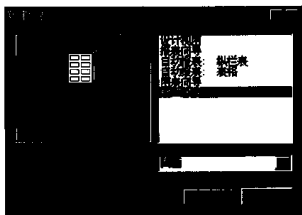


图 10.18

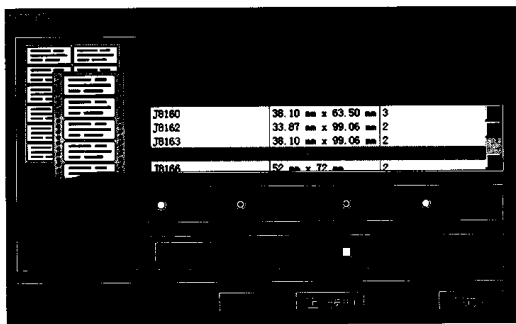


图 10.19

Access 显示如图 10.20 所示的“新建标签”窗口。用户可以在对话框中输入标签名，还可以选择度量单位、标签类型、方向参数，并且可在窗体下部的对话框中详细定义标签的尺寸。定义完后单击“确定”按钮，完成新标签的创建。

在选择完标签大小以后，单击“下一步”按钮，在此窗体中，用户可以定义文本外观。然后再单击“下一步”按钮，在如图 10.21 所示的窗体中，用户可以选择字段作为标签中的选项，并且需要按 Enter 键来定义标签格式。

再单击“下一步”按钮，在此窗口中，用户可以用可用字段中的一个或几个字段对标签进行排序。再单击“下一步”按钮，到向导的最后一个窗口上，在此窗口中定义标签报表的名字。单击“完成”按钮，显示标签报表。以何种视图显示由用户在最后一个窗口中决定。图 10.22 显示了利用标签向导创建的标签报表。

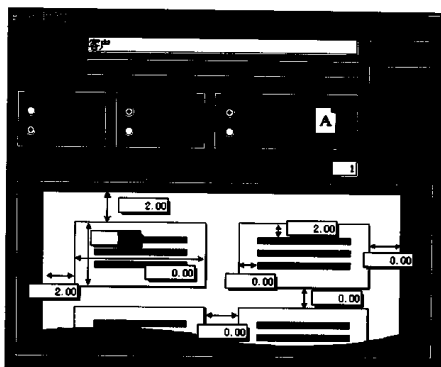


图 10.20

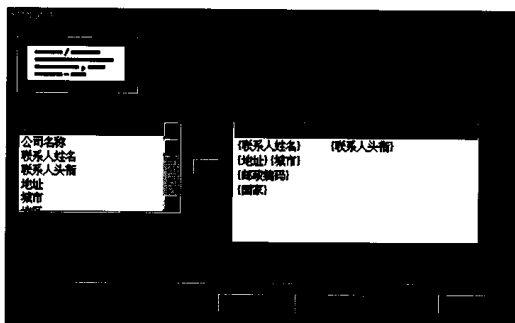


图 10.21

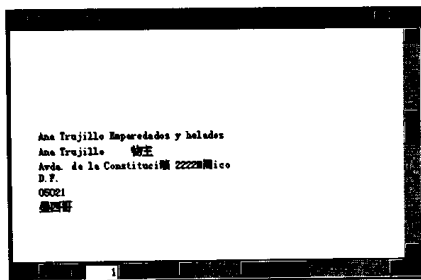


图 10.22

10.5 排序与分组

下面介绍报表的排序与分组。在报表中最多可为 10 个字段或表达式进行分组。

首先在“设计”视图中打开相应的报表，单击工具栏上的“排序与分组”按钮，显示“排序与分组”框，如图 10.23 所示。在对话框中设置字段，为报表中的数据设置排序顺序。

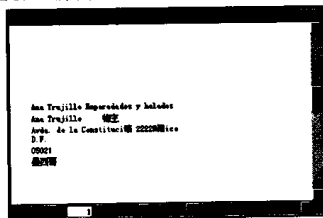


图 10.23

在“字段/表达式”列的第一行选择一个字段名称，或键入一个表达式。第一行的字段或表达式具有最高的排序优先级（最大的设置），第二行则具有次高的排序优先级，依此类推。完成“字段/表达式”列的填充以后，Microsoft Access 将把“排序顺序”设置为“升序”，即从 A 到 Z 或从 0 到 9 排序。如果要改变排序顺序，可以在“排序顺序”列表中选择“降序”，即从 Z 到 A 或从 9 到 0 排序。

在框中单击要设置分组属性的字段或表达式，在下面的主属性框中设置分组属性。如果要创建一个组级别并设置其他分组属性，必须将“组标头”或“组注脚”设置为“是”。组标头是用来在记录组的开头放置信息，例如组名称或组总计数；组注脚是用来在记录组的结

尾放置信息，例如组名称或组总计数。

下面介绍分组属性。

- 组标头——为字段或表达式添加或删除组标头。
- 组注脚——为字段或表达式添加或删除组注脚。
- 分组形式——指定对值的分组方式。可用的选项取决于分组字段的数据类型。如果是

按表达式分组，将显示所有数据类型的所有选项。

- 组间距——为分组字段或表达式的值指定有效的组间距。
- 保持同页——指定 Microsoft Access 是否在一页中打印组的所有内容。

下面将按字段数据类型对分组时的不同情况进行介绍。

1. 按 Text 字段分组记录

在按 Text 字段分组记录时，可以将“分组形式”属性设置为表 10.1 所列的范围。如果“分组形式”属性设置为“前缀字符”，则“组间距”属性可以设置为对分组字段值有效的任何数字。如果“分组形式”属性设置为“每一个值”，则“组间距”属性值为 1，如表 10.1 所示。

表 10.1 按 Text 字段分组记录

将“分组形式”属性设置为	分组记录
每一个值	包含与字段或表达式中相同值的记录
前缀字符	包含与字段或表达式中前 n 个相同字符的记录

在 Northwind 数据库中，“按字母顺序的产品列表”报表中的记录是按产品名称中的第一个字母的顺序排列，图 10.24 显示了该报表的“打印预览”视图。表 10.2 列出了该报表

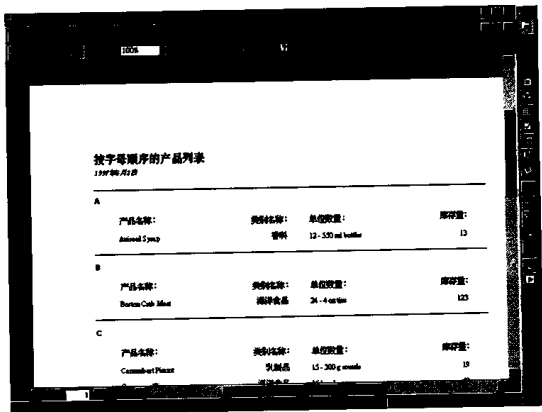


图 10.24

的排序与分组设置。“排序与分组”框的“字段/表达式”列第一行中的“产品名称”，将产品按产品名称的第一个字母进行分组。第二行中的“产品名称”则在每个字符下按产品名称的字符顺序对记录进行排序。

表 10.2 报表的排序与分组设置

字段/表达式	排序顺序	组标头	组注脚	分组形式	组间距	保持同页
产品名称	升序	是	是	前缀字符	1	整个组
产品名称	升序	否	否	每一个值	1	否

2. 按日期/时间字段分组记录

当按日期/时间字段分组记录时，可将“分组形式”属性设置为表 10.3 所列的范围。对除“每一个值”以外的所有选项，可以将“组间距”属性设置为对分组字段或表达式值有效的任何数字。如果“分组形式”属性设置为“每一个值”，则“组间距”的属性值为 1，如表 10.3 所示。

表 10.3 按日期/时间字段分组记录

将“分组形式”属性设置为	如下分组记录
每一个值	字段或表达式中的相同值
年	同一历法年中的日期
季	同一历法季度中的日期
月	同一月份中的日期
周	同一周中的日期
日	同一天的日期
时	同一小时中的时间
分	同一分钟中的时间

在 Northwind 数据库中，“按年度汇总销售额”报表显示各年每季度的销售情况，报表的组标头显示了年份，组注脚显示该年度总计值，如图 10.25 所示。表 10.4 列出了该报表主要的排序与分组设置。“排序与分组”框的“字段/表达式”列第一行中的“发货日期”按年份对记录进行分组。第二行中的“发货日期”则将每年的记录按季度进行分组。

表 10.4 “按年度汇总销售额”报表排序与分组设置

字段/表达式	排序顺序	组标头	组注脚	分组形式	组间距	保持同页
发货日期	升序	是	是	年	1	所有的组
发货日期	升序	否	否	季	1	不

3. 按 AutoNumber 字段、Currency 字段或 Number 字段分组记录

当按 AutoNumber 字段、Currency 字段或 Number 字段分组记录时，可以将“分组形式”

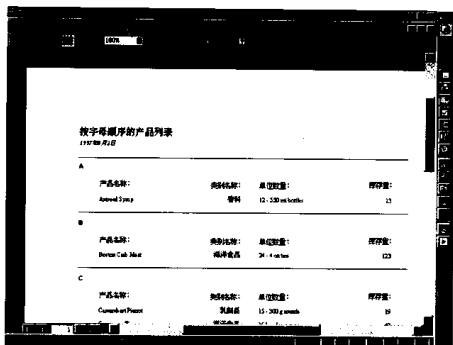


图 10.25

属性设置为表 10.5 所列的范围。如果“分组形式”属性设置为“间隔”，则“组间距”属性可以设置为对分组字段或表达式值有效的任何数值。如果“分组形式”属性设置为“每一个值”，则“组间距”的属性值为 1，如表 10.5 所示。

表 10.5 按三字段分组记录

将“分组形式”属性设置为	如下分组记录
每一个值	字段或表达式中的相同值
间隔	位于指定间隔中的值

Microsoft Access 从 0 开始对自动编号、货币和数字字段进行分组。例如，当“分组形式”属性设置为“间隔”、“组间距”属性设置为 5 时，Microsoft Access 将按如下方式分组记录：0 到 4，5 到 9，10 到 14 等等。

在 Northwind 数据库中，“按金额汇总销售额”报表中以 1 000 美元为间隔对记录进行分组，图 10.26 显示了该报表的“打印预览”视图。表 10.6 列出了该报表的排序与分组设置。“排序与分组”框的“字段/表达式”列第一行中的“销售额”将记录以间隔 100 进行分组。第二行中的“销售额”则以每 100 美元为间隔对记录按升序进行排序。

表 10.6 “按金额汇总销售额”报表排序与分组设置

字段/表达式	排序顺序	组标头	组注脚	分组形式	组间距	保持同页
销售额	升序	否	是	间隔	100	所有的组
销售额	升序	否	是	每一个值	1	不

按字母顺序的产品列表
1998年1月1日

A			
产品名称:	类别名称:	单位数量:	库存量:
Asinod Syrup	香料	12 - 530 ml bottles	12
B			
产品名称:	类别名称:	单位数量:	库存量:
Boston Crab Meat	海鲜食品	26 - 4 oz tins	123
C			
产品名称:	类别名称:	单位数量:	库存量:
Creamed art Potatoes	乳制品	15 - 300 g cans	19

图 10.26

第十一章 宏

当使用 Microsoft Access 作一些实际工作的时候, 用户可能会发现自己在翻来复去重复某一项工作。可以通过建立宏来执行这些重复的任务。宏可以节省用户的时间, 特别是当宏里面包含着很长一系列的工作时, 意义更为重大。

宏是用来自动完成特定任务的操作集。它可以是一个或多个操作的集合, 其中每个操作实现特定的功能, 例如打开某个窗体或打印某个报表。宏可以使某些普通的任务自动完成, 例如, 可设置某个宏, 在用户单击某个命令按钮时运行该宏, 以打印某个报表。

宏可以是包含操作序列的一个宏, 也可以是某个宏组, 使用条件表达式可以决定在某些情况下运行宏时, 某个操作是否进行。

11.1 宏的作用

Microsoft Access 提供了许多不同类型的宏操作, 可以用来自动执行应用程序。

- 可以用宏打开或关闭任何可用视图中的任何表、查询、窗体或报表。
- 可以用宏来打开一个“打印预览”中的报表, 或把一个报表直接送到打印机上; 也可以把一个报表中的输出数据送到一个 RTF 文件或一个 Microsoft Excel 文件中, 然后打开 Microsoft Word 或 Microsoft Excel 中的文件; 也可以把报表中的输出数据直接作为一个电子邮件发出。
- 可以使用宏来执行一个选择查询, 或一个操作查询。
- 可以使用一个宏, 根据任何检验数据库中、窗体中、报表中的值的条件, 来建立一个操作的宏, 可以用宏来执行其他宏或 Microsoft Access 中的模块。
- 可以用宏来设置任何窗体或报表控件的值。
- 可以用宏实现对数据的过滤, 进入报表, 查找窗体或查询中的数据。
- 可以用宏来打开或关闭任何标准的 Microsoft Access 工具条或自己定制的工具条。
- 可以用宏执行任何 Microsoft Access 菜单上的命令。
- 可以用宏来移动、缩小、放大或存储 Microsoft Access 工作区中的任何窗口, 并规定它们的大小。
- 可以用宏显示一信息性的消息, 使计算机发出警告信号; 也可以在执行操作查询时, 禁止某些警告信息。
- 可以用宏重新命名数据库中的任何对象。

用户在后面将会发现, 宏能做几乎任何工作。但是在 Microsoft Access 97 中, 由于采用了 Microsoft Visual Basic for Application 作为它的编程语言, 使宏的地位有所下降, 但是它仍起着十分重要的作用。

11.2 宏中的工具条

Microsoft Access 为操作宏提供了定制的工具条。在“数据库窗口”中单击“宏”按钮，并单击“新建”按钮，用户就可以打开一个新的宏窗口，窗口的工具条如图 11.1 所示。

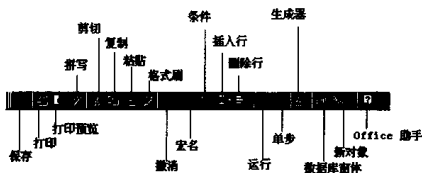


图 11.1

- 保存按钮——单击此按钮，保存用户对宏所作的修改。
- 打印预览按钮——单击此按钮，Microsoft Access 打印预览宏定义。
- 宏名按钮——单击此按钮，在宏的定义窗口内显示宏列。
- 条件按钮——单击此按钮，在宏的定义窗口中显示条件列。
- 插入行按钮——在宏定义表中，单击其中要插入的那一行，再单击此按钮，将在此行的上一行中插入一空行。
- 删除行按钮——单击宏定义表中要删除的那一行，再单击此按钮，Microsoft Access 将删除此行。
- 运行按钮——单击此按钮，运行宏。
- 单步按钮——单击此按钮，使一次有一个动作通过宏，这个选择对于调度复杂的宏是十分有用的。在用户设置了这个选择以后，它一直保持，直到关闭它。
- 生成器按钮——Microsoft Access 提供了一个高级的表达式生成器对话框，帮助用户定义宏动作参数的输入。
- 数据库窗口按钮——单击此按钮，使数据库窗口成为当前窗口。

11.3 建立宏

在数据库窗体中单击“宏”按钮，在窗口内显示了数据库中所有宏的列表，再单击“新建”按钮，Microsoft Access 进入“宏”定义窗口，如图 11.2 所示。在窗口顶部，有两列：操作列与备注列。

- 操作：列出想用这个宏来执行的所有动作，宏按照列表中所表明的顺序来执行这些操作。
- 备注：增加操作的描述，让用户更容易明白这个宏的功能；备注是可选的，但如果忽略它们，理解与维护较早的宏定义将变得困难。
- 操作参数：显示于窗口的左下角的操作参数面板中；大部分动作都需要参数，它们是

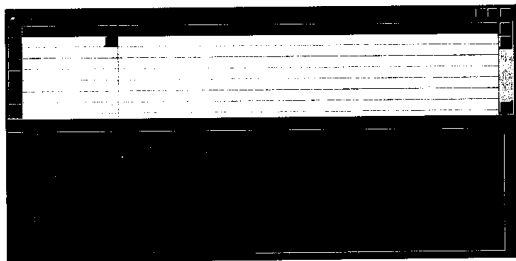


图 11.2

控制操作执行方式的选择。

● 帮助信息：在“宏”窗口的右下方区域显示了一个短的帮助信息；这个信息是随当前光标所在的“宏”窗口中的不同位置而变化的。

其实还有两列，“宏名”与“条件”，它们在缺省情况下是不显示的。单击工具条中的“宏名”按钮，在窗口中显示“宏名”；单击工具条中的“条件”按钮，在窗体中显示“条件”列。

- 宏名：可以输入宏的名称，这在一个宏组中是必须的。
- 条件：在此列中输入条件表达式，可以用来决定在什么情况下执行宏。

11.3.1 操作

Microsoft Access 提供了许多不同类型的操作供用户使用，可分为如下几个方面。

● AddMenu 为一个窗体或报表将菜单添加到自定义菜单栏中。在菜单栏中的每一个菜单都需要一个独立的 AddMenu 操作。同样，为窗体、窗体控件或报表添加自定义快捷菜单，也需要一个独立的 AddMenu 操作。

● Apply Filter 将一个过滤器或查询放到一个表格中，用它来限制和决定表格所基于的记录的类型序号。

● Beep 使得计算机发出“嘟嘟”的响声向用户报警。

● Close 关闭一个对象。

● DoMenuItem 可执行任何 Microsoft Access 命令。所指定执行的指令必须与动作执行所在的环境条件相适合。

● FindRecord 可以查找到符合所指定的条件的一个记录。

● Maximize 使得当前窗口扩到最大。

● OpenForm、OpenQuery、OpenReport 和 OpenTable 打开指定的对象。

● Print 打印当前自动选择的对象。

● Quit 退出 Microsoft Access。

● RunMacro 运行在当前的宏中指定的宏。

- **Select Object** 选择一个用户指定的对象。然后就可以执行应用于当前选择对象的操作，诸如打印那个对象等。

- **Show All Record** 移掉当前选择的表格中的所有限制条件，因此在一个表或查询中它所基于的所有记录都将被显示出来。

这只是一个宏所能执行的操作的示例列表，只包含了一些常用的操作。

11.3.2 操作参数

一项动作的参数指定该项动作运用的对象以及控制动作执行方式的选择。

在宏中的大多数动作都需要参数。举例来讲，假如使用的是“打开表”动作，就必须包含一个指定应该打开哪个表的参数。

当在动作列表中高亮度显示一个动作时，它的参数将出现在屏幕下半部分的操作参数区域中，就如在设计一个表的时候每一个字段的特性将显示在屏幕的下半部分中一样。

图 11.3 显示的是“打开表”（OpenTable）动作的参数。显然必须指定要打开的表名字，也可以指定打开表所在的视图（“数据表”视图、“设计”视图或“打印预览”），并且指定增添（允许增加新的记录）、编辑（允许编辑存在的记录或增加新的记录）或只读（仅允许查看记录）等数据模式。

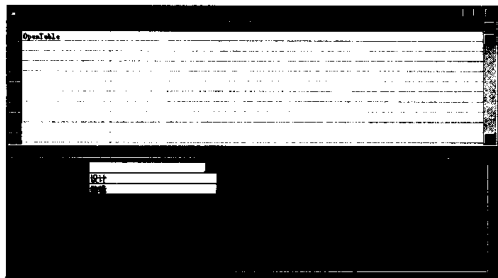


图 11.3

11.3.3 向宏中添加操作

在“宏”窗口的“操作”列单击第一个空白行。如果要在两个操作行之间插入一个操作，请单击插入行下面的操作行的行选定器，然后在工具栏上单击“插入行”按钮。用户既可以直接在“操作”列输入操作名，也可在“操作”列单击箭头显示下拉列表，从列表中单击要使用的操作，然后为操作键入相应的说明。它是可选的。图 11.4 所示即为利用下拉列表来选取操作。

对有些操作，用户还需要在窗口的下部操作参数区域中指定参数。

● 从“编辑”菜单里选择“复制”、“剪切”以复制或剪切此行，然后从“编辑”菜单里选择“粘贴”，将此行放到别的位置。

11.4 运行宏

在创建完宏之后，用户就可以运行这些已创建好的宏了。宏的运行可分为多种不同的情况，下面分别介绍宏的运行情况。

在执行宏时，Microsoft Access 将从宏的起始点启动，并执行宏中所有操作直到到达另一个宏（如果宏是在宏组中）或者到达宏的结束点。

可以从其他宏或事件过程中直接执行宏，或者将执行宏作为对窗体、报表、控件中发生的事件做出的响应。例如，可以将某个宏附加到窗体中的命令按钮上，这样在用户单击按钮时就会执行相应的宏。也可以创建执行宏的自定义菜单命令或工具栏按钮，将某个宏指定至组合键中，或者在打开数据库时自动执行宏。

宏的执行可分为如下几种情况。

11.4.1 直接执行宏

用户可在多种情况直接执行宏。

- 如果要从“宏”窗口中执行宏，只需单击工具栏上的“执行”按钮。
- 如果要从“数据库”窗口中执行宏，单击“宏”标签，然后双击相应的宏名。
- 如果要在 Microsoft Access 的其他地方执行宏，则在“工具”菜单上单击“执行宏”，然后单击“宏名”框中相应的宏，Microsoft Access 显示了如图 11.5 所示的“执行宏”对话框，用户单击下拉按钮，从下拉列表中选择宏，并单击“确定”按钮以运行宏。
- 如果要从窗体“设计”视图或报表“设计”视图中执行宏，请将鼠标指向“工具”菜单中的“宏”，再单击“执行宏”，Microsoft Access 显示了“执行宏”对话框，用户单击下拉按钮，从下拉列表中选择宏，并单击“确定”按钮以运行宏。



图 11.5

在通常情况下直接执行宏只是进行测试。可以在确保宏的设计无误之后，将宏附加到窗体、报表或控件中，以对事件做出响应，也可以创建一个执行宏的自定义菜单命令。

11.4.2 从窗体或报表中运行

在报表与窗体中，用户可以将宏作为某个对象的事件来运行。要从窗体或报表中运行宏，只需单击在“设计”视图中的相应控件或窗口属性表中的事件标签。在相应的事件属性上单击，从下拉列表中选择相应的宏，当此事件发生时，Microsoft Access 就自动运行该宏。

例如，用户要求打开某个窗体的显示信息“你现在打开了窗体”，就可以先创建一个宏，

如图 11.6 所示。该宏中运用了 MsgBox 操作。自己创建一个简单窗体，其中可以只设一个命令按钮。在它的“设计”视图下，单击此命令按钮，在它的属性框中单击事件标签，找到单击事件，单击属性框，在下拉列表中选中的宏，如图 11.7 所示。

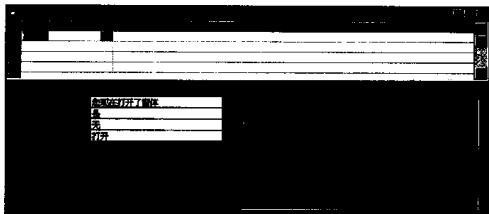


图 11.6

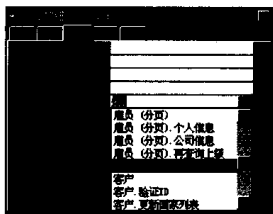


图 11.7

完成窗体的定义后单击“视图”按钮，将窗体在“窗体”视图下显示，单击“按钮”，则 Microsoft Access 会显示信息框，如图 11.8 所示。由于在宏的属性中，设置了鸣叫，因此



图 11.8

单击时，还伴随蜂鸣声。

当然，用户也可在窗体或报表的事件属性中单击“生成器”按钮，在 Microsoft Access 自动显示的“选择生成器”对话框中选中“宏生成器”，并单击“确定”按钮，如图 11.9 所示，则 Microsoft Access 显示宏的编辑环境，并且要求用户输入宏的名称，如图 11.10 所示。在创建完宏以后，在事件上显示宏名称。如图 11.11 所示。

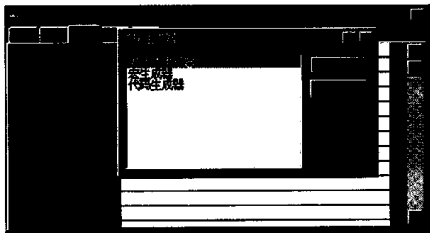


图 11.9

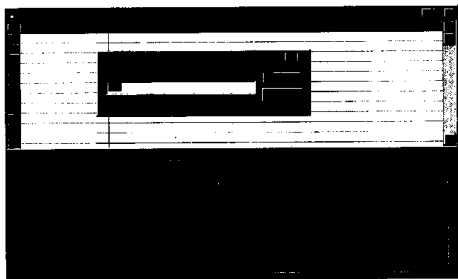


图 11.10

11.4.3 执行在另一个宏或 Visual Basic 程序中的宏

如果要在其他的宏或 Visual Basic 程序中执行宏，必须将 RunMacro 操作添加到相应的宏或程序中。

如果要将在 RunMacro 添加操作到宏中，请单击空白操作行的操作列表中的“Run - Macro”，并且将 Macro Name 参数设置为要执行的宏名。

如果要将在 RunMacro 操作添加到 Visual Basic 程序中，请在程序中添加 DoCmd 对象的 RunMacro 方法，然后指定要执行的宏名。例如，下列 RunMacro 方法将执行宏 My Macro: DoCmd.RunMacro "My Macro"。

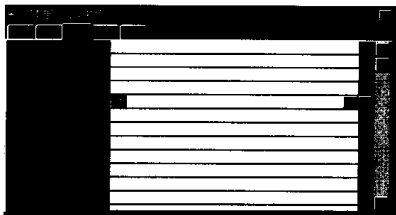


图 11.11

11.4.4 存储宏

在宏进行运行之前,必须存储它。单击工具栏中的“保存”按钮,Microsoft Access 会显示“另存为”对话框,与图 11.10 中相同,输入相应的名字,并单击“确定”按钮,保存宏。

11.4.5 检验宏

前面提到过,检验宏一般是直接运行。如果宏与窗体或报表有关,则必须在窗体与报表中进行检验。下面以直接运行方式进行测试。

Microsoft Access 为用户提供了检验宏的好办法。它是通过允许用户单步运行宏来实现的。

为了激活单步运行,首先在“设计”视图下打开宏,单击工具栏中的“单步运行”按钮,当按钮凹下时,说明单步执行已发生作用。

在单步执行宏时,每单击一次工具栏中的“执行”按钮,Microsoft Access 就会弹出一个“单步执行宏”对话框,在这个对话框中,显示了宏的名称、操作名称、操作参数、条件信息。

以刚才建立的宏为例,进行刚才建立的宏名字的测试。打开测试宏的窗体,单击工具栏中的“单步”按钮,然后再单击“执行”按钮,Microsoft Access 显示了如图 11.12 的“单步执行宏”对话框。单击对话框中的“单步执行”按钮,对话框中的操作将继续进行,并且用户将看到运行的结果,如图 11.13 所示。单击“确定”,跳过它。如果运行的宏中有一个以上的宏,Microsoft Access 将继续显示如图 11.12 所示的对话框,此时对话框中显示下一个操作。此例中由于只有一个操作,因此回到宏的“设计”视图下。

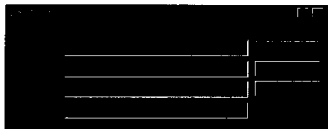


图 11.12



图 11.13

当用户在正常运行时,遇到宏的错误,Microsoft Access 将提示,显示一个对话框,来阐明错误,如图 11.14 所示。然后,用户将看到“操作失败”对话框,与“单步执行宏”对话框十分相似,如图 11.15 所示。它含有引起问题运作的相关信息,这时只要按“暂停”按钮,然后回到设计状态编辑宏,解决问题。



图 11.14



图 11.15

需要注意的是,用户单步执行完宏后,应再单击一次工具栏中“单步”按钮,取消“单步”执行宏;否则,在以后运行所存宏时,均按单步执行。

11.5 宏的高级技术

前面叙述宏的工具条时,曾经讲过工具条中的“宏名”按钮与“条件”按钮,在前几节中都未曾使用过,在本节中将介绍与这两个按钮有关的内容。

11.5.1 建立多操作宏

在 Microsoft Access 中,一个宏中可以定义一个以上的操作,并且可以定义操作执行的顺序。

图 11.16 即为定义了多操作的宏,在该宏中定义了三个操作:

- Echo 操作:当宏执行时冻结屏幕;
- Close 操作:关闭“产品列表”窗体;
- Open form 操作:打开“产品”窗体。

在多操作宏中,宏的运行次序是由上到下。因此首先运行 Echo 操作,由于属性中设置“响应打开”为 No,因此,在宏运行时就看不到无关动作在屏幕上闪动。Close 操作关闭了

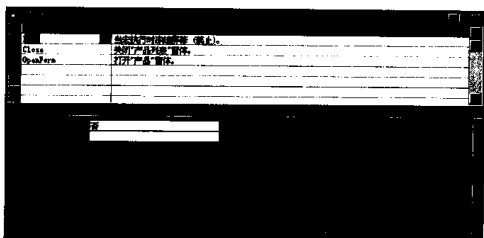


图 11.16

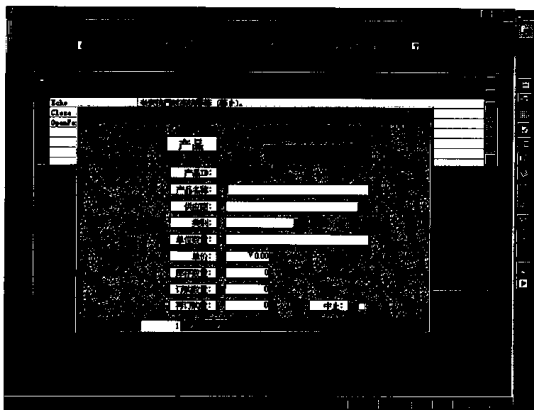


图 11.17

“产品列表”窗体。OpenForm 操作打开了“产品”窗体，运行此宏结果如图 11.17 所示，打开了产品窗体。

11.5.2 创建宏组

单击工具栏中的“宏名”按钮即可在“设计”视图的窗体中增加一列宏名列，再单击一次，又可以隐藏此列。

只有在建立宏组时，才要用到宏的名字。有时，用户需要创建许多宏，而这些宏一般情

况下是不从“数据库”窗口中运行的，此时，用户若还是将每个宏都命名保存到“数据库窗体”中，那么数据库窗口将变得杂乱无章，以至于在寻找宏时会变得很不方便。为了避免这种杂乱无章的情况，Microsoft Access 使用了宏组，即一个宏组是一组宏的集合，只有宏组的名称才加到数据库窗口中，而不是将每一个宏名称都加入窗口中。

为建立宏组，只需将每个宏的名字加入到它的第一项操作的左边的宏名列中，每一个宏名称代表一个宏。由于一个宏可能有多个操作，因此，在宏组的窗口内，同一宏的所有操作的宏名列中，只能在第一项操作的左边宏名列中可以有宏名，其他均应空白。图 11.18 显示了 Northwind 数据库中的一个宏组。

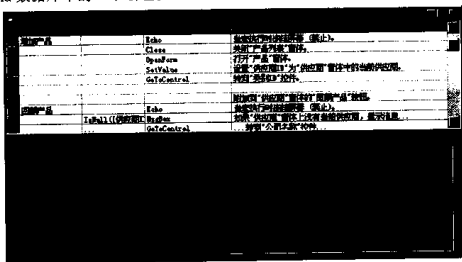


图 11.18

在保存一个宏组时，只要单击“保存”按钮，在显示的“另存为”对话框中输入用户定义的宏组名，再单击“确定”按钮，就保存了该宏组。在“数据库”窗口内，只显示宏组名。

假如要在“数据库”窗口中运行宏组，则只有宏组中的第一个宏可以运行，而其他宏不能直接运行，但可以用其他办法运行。要引用宏组中的宏，可用下面的语法：宏组名.宏名。例如“产品”宏组中有“汽车”宏，则在 RunMacro 操作中，填写属性时，应填写“产品.汽车”。

11.5.3 使用宏中的条件操作

单击宏“设计”视图工具栏上的“条件”按钮，会在宏设计窗体内增加一条件列，再单击一次取消条件列。

1. 在宏中使用条件的原因

在某些情况下，可能希望仅当特定条件为真时才在宏中执行一个或多个操作。例如，如果在某个窗体中使用宏来校验数据，可能要显示相应的信息来响应记录的某些输入值，另一信息来响应另一些不同的值。在这种情况下，可以使用条件来控制宏的流程。

如图 11.19 所示的宏，当“国家”字段没有值（即这个值是 Null），则下面的宏将下达 StopMacro 的操作；如果窗体上“邮政编码”字段数值的长度或格式与“国家”字段中相应

国家不符时，将执行几对 MsgBox 和 CancelEvent 操作之一。

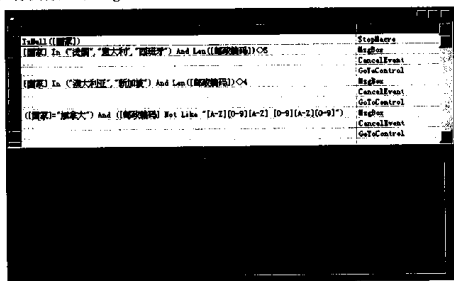


图 11.19

2. 在宏中使用条件

在“条件”列的要设置条件的行中键入相应的条件表达式，在“操作”列中，输入设置条件为真时要 Microsoft Access 执行的操作。如果这个条件结果为真，则 Microsoft Access 将执行此行中的操作。在紧跟此操作的“条件”栏内键入省略号 (...)，就可以使 Microsoft Access 在条件为真时可以执行这些操作。

用户可以用“表达式生成器”来创建表达式，用鼠标右键单击“条件”单元格，然后单击“生成器”按钮。在宏“条件”列不能使用 SQL 表达式。

执行该宏时，Microsoft Access 将求出第一个条件表达式的结果。如果这个条件的结果为真，Microsoft Access 将执行此行所设置的操作以及紧接着此操作且在“条件”栏内有省略号 (...) 的所有操作。例如，图 11.20 所示的宏，只有当“供应商 ID”字段没有数值（即这个值是 Null）时才会执行 MsgBox 和 StopMacro 操作。

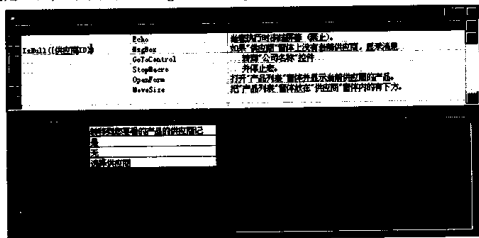


图 11.20

在执行完条件操作以后, Microsoft Access 将执行宏中任何空“条件”字段的附加操作, 直到到达另一个表达式、宏名或退出宏。

如果这个条件式的结果为假, Microsoft Access 则会忽略这个操作以及紧接着此操作且在“条件”字段内有省略号 (…) 的操作, 并且移到下一个包含其他条件或空“条件”字段的的操作。

3. 宏条件示例

表 11.1 列举了一些条件表达式, 帮助用户熟悉条件表达式的建立。

如果要 Microsoft Access 暂时忽略某个操作, 请输入 False 为条件。暂时忽略操作有助于找出宏中的问题。

表 11.1 条件表达式

使用下列表达式	执行此操作的条件
[城市] = "巴黎"	"巴黎"是执行宏的窗体中"城市"字段的值
Dcount (" [订单号码]", " 订单") > 35	"订单"表的"订单号码"字段的数目超过 35 项
Dcount (" * ", " 订单明细", " [订单 ID]" = Forms! [订单]! [订单号码]) > 3	"订单明细"表的"订单 ID"字段与"订单"窗体的"订单 ID"字段相等, 并且其数目超过 3 项
[发货日期] Between # 2 - Feb - 1995 # And # 2 - Mar - 1995 #	执行此宏的窗体上的"发货日期"字段值在 1995 年 2 月 2 日和 1995 年 5 月 2 日之间
Forms! [产品]! [库存数量] < 5	"产品"窗体的"库存数量"字段的值小于 5
IsNull ([姓名])	执行此宏的窗体上的"姓名"字段值是 Null (没有值)。这个表达式等价于 [姓名] Is Null
[国家] = " UK" And Forms! [销售总数]! [订货总数] > 100	执行此宏的窗体上的"国家"字段值是 UK, 且在"销售总数"窗体内的"订货总数"字段值大于 100
[国家] In (" 法国", " 意大利", " 西班牙") And Len ([邮政编码]) < > 5	执行此宏的窗体上"国家"字段值是法国、意大利或西班牙, 且"邮政编码"的字符长度不等于 5
MsgBox (" Confirm changes?", 1) = 1	在 MsgBox 函数显示的对话框中单击"确定"按钮。如果在对话框中单击了"取消"按钮, Microsoft Access 将忽略这个操作

第十二章

Visual Basic 基础

前面曾经介绍过, Microsoft Access 97 乃至整个 Microsoft Office 97, 都是以 Visual Basic 作为编程语言的, 当用户需要使用 Microsoft Access 97 超出界面以外的功能时, 就必须使用 Visual Basic。本章将介绍用户在使用 Visual Basic 时会遇到的术语, 并且还将介绍 Microsoft Access 的组成部分——模块的一些特点, 特别是在 Microsoft Access 97 中的新特点。

12.1 用 Visual Basic 代替宏

在 Microsoft Access 中, 通过宏或者用户界面可以完成许多任务; 而在其他许多数据库程序中, 要完成相同的任务就必须通过编程。使用宏还是用 Visual Basic 来创建应用程序, 取决于需要完成的任务。

12.1.1 什么时候使用宏

对于简单的细节工作, 譬如打开和关闭窗体、显示和隐藏工具栏或运行报表等, 使用宏是一种很方便的方法。它可以简便迅速地将已经创建的数据库对象联系在一起, 因为不需要记住各种语法, 并且每个操作参数都显示在“宏”窗口的下半部分。

除了利用宏的简易性以外, 还可以使用宏来创建全局赋值键, 在首次打开数据库时执行一个或一系列操作; 通过“启动”对话框也可以在打开数据库时指定特定的操作, 例如打开窗体。

12.1.2 什么时候使用 Visual Basic

对于以下的情况, 应该使用 Visual Basic 而不要使用宏。

- 使数据库易于维护。因为宏是独立于使用它的窗体和报表的对象, 所以一个包含用于响应窗体和报表上的事件的事件的宏的数据库将变得难以维护; 相反, Visual Basic 事件过程创建在窗体或报表的定义中, 如果把窗体或报表从一个数据库移动到另一个数据库, 则窗体或报表所带的事件过程也会同时移动。

- 创建自己的函数。Microsoft Access 包含许多内置的函数, 例如用于计算利息费用的 IPmt 函数。在计算时使用这些函数可以避免创建复杂的表达式。使用 Visual Basic 可以创建自己的函数, 通过这些函数可以执行表达式难以胜任的复杂计算, 或者用来代替复杂的表达式。此外, 也可在表达式中使用自己创建的函数对多个对象应用操作。

- 显示错误信息。当用户在使用数据库遇到预料之外的事情时, Microsoft Access 将显示一则错误信息, 但该信息对于用户而言可能是莫名其妙的, 特别是当用户不熟悉 Microsoft Access 时; 而使用 Visual Basic 则可以在出现错误时检测错误, 并显示指定的信息或执行某些操作。

● 创建或操作对象。在大多数情况下，在对象的“设计”视图中创建和更改对象是最简单的方法；但在某些情况下，可能需要在代码中对对象进行定义。使用 Visual Basic 可以操作数据库中的所有对象，包括数据库本身。

● 执行系统级别的操作。虽然在宏中执行 RunApp 操作可以从一个应用程序运行另一个基于 Windows 或 MS-DOS 的应用程序，但是在 Microsoft Access 以外使用宏具有很大的局限性；而使用 Visual Basic 则可以查看系统中是否存在每个文件，或者通过自动化或动态数据交换 (DDE) 与另外一个基于 Windows 的应用程序（如 Excel）进行通讯，还可以调用 Windows 动态链接库 (DLLs) 中的函数。

● 一次操作多个记录。使用 Visual Basic 可一次浏览一个记录集或是单个记录，并对每个记录执行一个操作；而宏只能对整个记录集进行操作。

● 将参数传送给 Visual Basic 过程。在创建宏时可以在“宏”窗口的下半部分设置宏操作的参数值，但在运行宏时不能改变它们；而使用 Visual Basic 则可在程序运行期间将参数传递给代码，或者使用变量参数，这在宏中是难以做到的，因而使得运行 Visual Basic 过程时具有更大的灵活性。

12.2 基础知识

在介绍 Visual Basic 之前，首先介绍在 Microsoft Access 中用 Visual Basic 编程时经常要遇到的术语。

12.2.1 事件

事件是由对象识别并可以定义如何响应的操作，它是一种特定的操作，在某个对象上发生或对某个对象发生。Microsoft Access 可以响应多种类型的事件：鼠标单击、键盘上的键按下、数据更改、窗体打开或关闭及许多其他类型的事件。事件的发生通常是用户操作的结果，但也可以由 Visual Basic 语句引起，还可以由系统触发。使用与事件相关联的属性时，可告知 Microsoft Access 执行宏，调用 Visual Basic 函数或者运行事件过程来响应事件。

12.2.2 事件驱动的应用

在 Microsoft Access 中任何一类对象都自动拥有一组预先定义好的事件，当用户需要窗体、报表或控件对其中的事件作出特定的响应时，只需要在此事件中用 Visual Basic 编写事件过程。

以下就是一个典型的事件驱动的应用中的响应过程。

(1) 当用户启动此应用时，Microsoft Access 会按照启动对话框中的设定值显示启动窗体。

(2) 在打开启动窗体后，启动窗体以及窗体内的控件接收事件，这些事件可以由用户和窗体中的代码触发。

(3) 对这些事件按照事件过程中的代码作出响应。

(4) 应用等待下一个事件。

12.2.3 模 块

在 Microsoft Access 数据库中, 用户编写的代码保存在模块中。模块是将 Visual Basic for Applications 声明和过程作为一个单元进行保存的集合。

模块有两个基本类型: 类模块和标准模块。模块中的每一个过程都可以是一个函数过程或一个子程序。

一、类模块

窗体和报表模块都是类模块, 而且它们各自与某一窗体或报表相关联。窗体和报表模块通常都含有事件过程, 该过程用于响应窗体或报表中的事件。可以使用事件过程来控制窗体或报表的行为以及它们对用户操作的响应, 例如, 用鼠标单击某个命令按钮。

为窗体或报表创建第一个事件过程时, Microsoft Access 将自动创建与之关联的窗体或报表模块。如果要查看窗体或报表的模块, 请单击窗体或报表“设计”视图中工具栏上的“代码”命令。

窗体或报表模块中的过程可以调用已经添加到标准模块中的过程。

在 Microsoft Access 95 中, 类模块仅可以在与窗体或报表相关联时出现。在 Microsoft Access 97 中, 类模块不仅可以脱离窗体或报表而独立存在, 并且这种类型的模块可以在“数据库”窗口的“模块”选项卡中显示。使用“模块”选项卡中的类模块可以创建自定义对象的定义。

二、标准模块

标准模块包含的是通用过程和常用过程, 这些通用过程不与任何对象相关联, 常用过程可以在数据库中的任何位置执行。在标准模块中, 用户可以调用表达式、宏、事件过程或其他标准模块中的过程。

单击“数据库”窗口中的“模块”选项卡, 可以查看数据库中标准模块的列表。图 12.1 所示就是订单数据库中的标准模块。窗体、报表和标准模块也都在“对象浏览器”中显示出来, 图 12.2 就是在“对象浏览器”中显示出来的订单数据库中的模块。

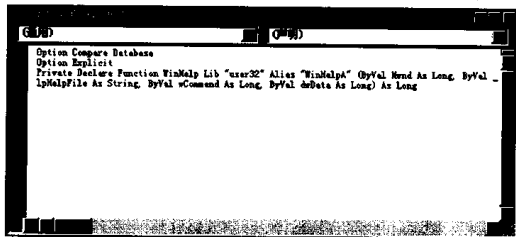
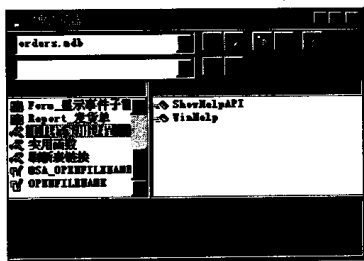


图 12.1



12.2

三、模块的组成

模块无论是类模块还是标准模块，均是由以下几部分组成的。

1. 声明部分

在这个部分中，用户可以定义变量、常量、用户自定义类型和外部过程。声明部分与过程部分是相互隔离的，而且声明部分中的内容可以被模块中的所有过程所调用。在此部分中定义的变量和常量等都是全局的。如果用户只需要定义在当前过程中使用的变量或常量，可在当前过程中定义。

2. 事件过程

事件过程是一种自动执行的过程，用来对用户或程序代码启动的事件或系统触发的事件作出响应。在这个部分中，主要包括应用于某个特定对象的子过程。这些过程是对用户或系统事件作出的响应，例如单击鼠标事件，事件过程一般包含于窗体或报表中的窗体模块或报表模块中。

3. 通用过程

通用过程是不与特定的对象或事件相联系的，它不仅出现在标准模块中，还可以出现在类模块中。通用过程可以是子过程也可以是函数。

四、过程

过程是包含 Visual Basic for Applications 代码的单位。它包含一系列的语句和方法，以执行操作或计算数值。

语句是一个完整的单位，用于表达特定的运算、声明或定义。通常在过程或声明节中，语句是单独成行的；不过在有些情况下，通过使用冒号（:）分隔语句，可将多个语句放在同一行中。

例如, Dim filename as String 即为语句, 它的作用是将 filename 这个变量声明为字符串。Visual Basic 编写的程序是由众多的语句组成的。语句是程序的基本单位。

方法是在特定对象上执行的一种过程，类似于语句或函数。例如，对于 DoCmd 对象它有多方法，如在下例中用到的 OpenForm 方法，是用来打开某个窗体的。

下列事件过程使用 OpenForm 方法打开“订单”窗体：

```
Private Sub OpenOrders_Click ()  
DoCmd.OpenForm "Orders"  
End Sub
```

在 Microsoft Access 中有两种过程：Sub 过程与函数过程。

Sub 过程执行一个操作或一系列的运算，但是不返回值。它可以自己创建子程序或使用 Microsoft Access 所创建的事件过程模板。

数据库的每一个窗体和报表，都有内置的窗体模块或报表模块，这些模块包含事件过程模板。可以向其中添加程序代码，使得当窗体、报表或其上的控件中发生相应的事件时，运行这些程序代码。当 Microsoft Access 识别到事件在窗体、报表或控件中已经发生时，将自动地执行为对象和事件命名的事件过程。例如，可以编写一个事件过程，在用户单击“OK”按钮时，系统弹出消息框，并在消息框中显示“Hello, 欢迎您使用 Microsoft Access!”上例事件过程就是在窗体或报表中的“OpenOrders”按钮被单击时，系统需要执行的动作，即打开“Orders”窗体。

许多向导（例如，命令按钮向导）在创建对象的同时也创建对象的事件过程。用户可以查看这些事件过程以及这些过程如何工作的示例。

函数过程（通常简单地称为函数）将返回一个值，例如计算结果。Visual Basic 包含许多内置函数，用户使用这些函数时，不需要定义，可以直接使用，例如，Now 函数可返回当前的日期与时间。除了这些内置函数外，也可以创建自己的自定义函数。为了使程序易读，用户应该将一些功能比较完整的部分创建为自定义函数，特别是对于那些经常使用的功能块。

因为函数有返回值，所以可以在表达式中使用。可以在 Microsoft Access 很多地方的表达式中使用函数，例如在 Visual Basic 语句或方法中，在一些属性设置中，或在窗体或报表的筛选或查询的准则表达式中。

以下是函数过程的一个示例：

```
Function FirstOfNextMonth ()  
FirstOfNextMonth = DateSerial (Year (Now), Month (Now) + 1, 1)  
End Function
```

FirstOfNextMonth 函数返回下个月第一天的日期。

这个自定义函数包含一个赋值语句，将表达式的结果（位于等号（=）的右边）指定给函数名 FirstOfNextMonth（位于等号（=）的左边）。该函数使用了内置的 DateSerial、Year、Now 和 Month 函数来计算结果。在自定义函数中，如果用户需要返回值，只需要将要返回的值用赋值语句赋给自定义函数名。

创建函数以后，就可以在 Microsoft Access 任何地方的表达式中使用该函数。例如，在窗体中，某一文本框是用来显示日期的，并且在缺省情况下，它的显示值是下一个月的第一天。用户只要通过在属性表中将文本框控件的 DefaultValue 属性设置为以下表达式：

```
= FirstOfNextMonth ()
```

就指定文本框将显示下个月的第一天作为它的默认值。

图 12.3 所示即为将此文本框的 DefaultValue 属性设置为该表达式。

需要注意的是,要将函数作为属性设置,函数必须位于窗体或报表的模块中,或位于标准模块中,在与窗体或报表无关的类模块中不能使用函数。图 12.4 中显示的就是定义在该窗体中的 FirstOfNextMonth 函数。

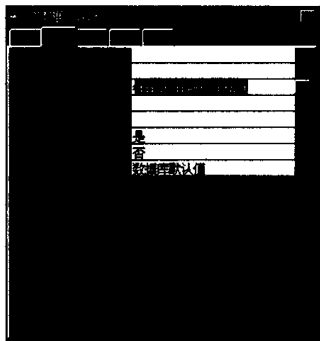


图 12.3

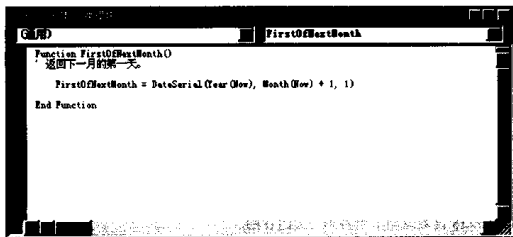


图 12.4

12.3 模块窗口新特点

用户要打开窗体或报表的模块窗口,只要在数据库窗口下选中要打开的窗体,然后在工具条中单击代码按钮,Microsoft Access 会以“设计”视图打开这个窗体,并且显示该窗体的代码。代码所在的窗口就是模块窗口,如图 12.5 所示就是打开订单数据库中的订单窗体

的模块窗口。此时, Microsoft Access 会显示适合于模块窗口使用的工具条和菜单。这些工具条和菜单的使用将在本书的后面再做叙述。

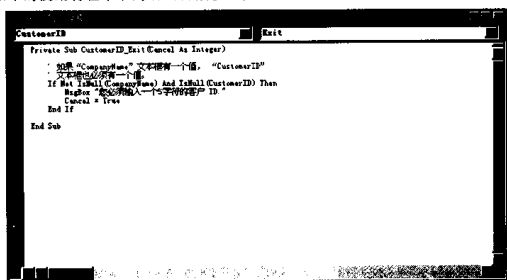


图 12.5

Microsoft Access 97 中的模块窗口比以前更易使用。由于使用了彩色代码标记, 使代码更易阅读, 而且自动创建清单功能和数据提示功能使输入代码更便捷、准确。

单击“工具”菜单上的“选项”命令, 即可打开“选项”对话框, 如图 12.6 所示。对

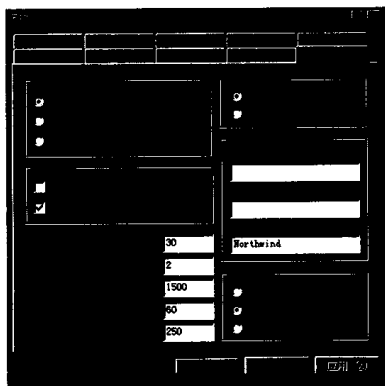


图 12.6

话框中的“高级”标签包含一个新的选项组：“捕获错误”。该选项组具有两个新的选项：“在类模块内中断”和“无法处理错误时中断”。Microsoft Access 95 的“选项”对话框“Modules”标签上的 Break On All Errors 选项，现在与上述两个新选项一起包含在“捕获错误”选项组中。

这些选项决定了 Visual Basic 在代码发生错误时如何进入中断方式。它们按下列方式影响 Visual Basic 的行为。

● “出现任何错误即中断”选项——Visual Basic 在标准模块和类模块中所有处理和未处理的错误处中断。当中断时使产生错误的过程所在的模块成为活动模块，而错误所在的代码行呈高亮度显示，这时可以按 Alt + F8 或 Alt + F5 来运行或单步执行错误代码。

● “在类模块内中断”选项——Visual Basic 在类模块和标准模块中的所有未处理错误处中断。当模块中的某一过程产生错误时，该模块将成为活动模块，而错误所在的代码行呈高亮度显示，这时可以按 Alt + F8 或 Alt + F5 来运行或单步执行错误代码。使用该选项可以调试类模块中的代码。

● “无法处理错误时中断”选项——Visual Basic 在标准模块中的所有未处理错误处中断。如果类模块中的某一过程产生错误，Visual Basic 将在模块中调用该过程的代码行处中断。使用该选项不能在类模块内部进入中断模式。

使用 SetOption 方法可以在代码中设置这些选项。使用 GetOption 方法可以返回这些选项的值。

新的“调试”菜单，如图 12.7 所示，包含两个新命令和一些 Access 95 的“运行”菜单或“工具”菜单上已有的命令。“调试”菜单只有在模块“设计”视图中才能使用。

● “编译并保存所有模块”命令可编译项目中的所有模块，并在编译阶段保存它们。

● “移出”命令从调用命令的行开始执行所有嵌套过程中的全部程序代码，然后返回到调用该命令的过程的上级过程。例如，假设有由 A 到 E 的过程嵌套，A 是调用树中的第一个过程。从 B 使用“移出”命令，而 B 调用过程 C，C 调用过程 D，D 调用过程 E，B 完成执行后返回到 A。对于已经进入过程中，但不希望继续单步执行的情况，该命令非常有用。

Access 97 还包含了在“模块”窗口中写入程序代码的许多新选项，其中一些在下面将有所说明。如果要设置这些选项，可单击“工具”菜单上的“选项”，在图 12.6 中单击“模块”标签，如图 12.8 所示。

● 编写过程时，“自动列表成员名单”选项将提供相关对象、属性及方法的列表。当键入对象名称时，Access 会显示可能跟随在该对象名称之后的对象、属性或方法的列表，自动帮助完成语句。要完成正在键入的语句，可以单击列表中的项目或继续键入程序代码。如果继续键入程序代码，列表将显示与键入的代码最符合的项目。如果要输入从列表中选择的项目，可按 Ctrl + Enter 或 Tab 键。如果要输入选定的项目，并将光标移动到下一行，可按 Enter 键。如果要关闭列表，请按 Esc 键。

● 在“模块”窗口中键入数据时，“自动显示快速信息”选项将提供语法信息。当键入



图 12.7

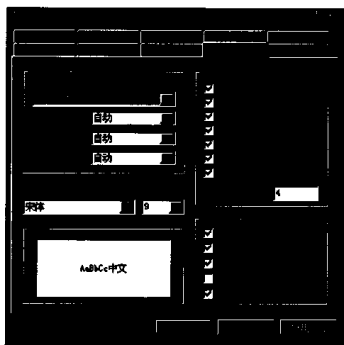


图 12.8

跟随着空格或开放括号的过程或方法名称时，在正在书写的程序代码下面会自动显示关于过程语法信息的提示，包括需要使用的参数。

● 当程序代码处于中断方式时，“自动数据提示”选项将提供变量或表达式的值。选择需要检查的变量或表达式，将鼠标移向它，可以显示变量或表达式当前的值。

“模块”窗口支持可用于完全自定义彩色码的语法，以及可用于设置“选项”对话框（单击“工具”菜单上的“选项”可打开该对话框）“模块”标签的文本格式选项。它同时也支持连接线字符，使得可以跨行书写同一条程序代码。连接线字符由一个空格和下划线（_）组成，如下例所示：

```
MsgBox " You are about to delete 10 records." _  
& " Do you want to continue?"
```

在“选项”对话框的“模块”标签上，可以设置其他各种不同的环境选项。例如，选中“整个模块窗口”复选框可以查看整个模块。如果清除此框，一次只查看一个过程（单击“模块”窗口左下角的按钮可以切换该选项）。也可以设置其他程序代码选项，例如“请求时编译”和“要求变量”声明。在“选项”对话框的“高级”标签上，可以设置命令行参数和条件编译参数。

“模块”窗口中的许多功能，可用来帮助调试程序代码。“快速监视”对话框能够立即提供所选变量或表达式的值。只需选择变量或表达式，然后按 Shift + F9，或单击“调试”菜单上的“快速监视”即可，如图 12.9 所示。

当程序代码处于中止状态时，可以使用“调试”菜单上的“设置下一句”和“显示下一句”命令。“设置下一句”命令能够跳过部分程序代码，将执行点移到另一个语句，或回头重新执行某一特定语句。“显示下一句”命令则显示当前正在执行的程序代码中的下一条执行语句。



图 12.9

开发环境提供“End”、“Reset”和“Go/Continue”按钮以及相应的“Run”菜单命令。当因为程序运行错误而单击工具栏上的“End”按钮，或对话框中的“End”按钮来退出程序代码的执行时，模块级变量将保留它们的值。要重置模块级变量，可单击工具栏上的“Reset”按钮。

这些调试工具在本书后续部分有详细的介绍。

第十三章

Visual Basic 编程

本章将介绍 Visual Basic 中的数据类型、变量、常量和数组，这些是进行 Visual Basic 编程时必不可少的内容。

13.1 Visual Basic 中的数据类型

首先对 Visual Basic 所支持的数据类型有一个总体上的了解。在表 13.1 中列出了 Visual Basic 所支持的数据类型以及存储空间大小与范围。

表 13.1 数据类型概述

数据类型	存储空间	范 围
Byte	1 个字节	0~255
Boolean	2 个字节	True 或 False
Integer	2 个字节	-32 768~32 767
Long (长整型)	4 个字节	-2 147 483 648~2 147 483 647
Single (单精度浮点型)	4 个字节	负数时从 -3.402 823E38~-1.401 298E-45; 正数时从 1.401 298E-45~3.402 823E38
Double (双精度浮点型)	8 个字节	负数时从 -1.797 693 134 862 32E308~-4.940 656 458 412 47E-324; 正数时从 4.940 656 458 412 47E-324~1.797 693 134 862 32E308
Currency (变比整型)	8 个字节	从 -922 337 203 685 477.580 8~-922 337 203 685 477.580 7
Decimal	14 个字节	没有小数点时为 +/-79 228 162 514 264 337 593 543 950 335, 而小数点右边有 28 位数时为 +/-7.922 816 251 426 433 759 354 395 033 5; 最小的非零值为 +/-0.000 000 000 000 000 000 000 000 000 1
Date	8 个字节	100 年 1 月 1 日~9 999 年 12 月 31 日
Object	4 个字节	任何 Object 引用
String (变长)	10 字节加字符串长度	0~大约 20 亿
String (定长)	字符串长度	1 到大约 65 400
Variant (数字)	16 个字节	任何数字值, 最大可达 Double 的范围
Variant (字符)	22 个字节加字符串长度	与变长 String 有相同的范围
用户自定义 (利用 Type)	所有元素所需数目	每个元素的范围与它本身的数据类型的范围相同。

13.1.1 使用数据类型

在用 Visual Basic 编程时, 除非有其他的指定, 否则未声明变量会被指定成 Variant 数据类型。这个数据类型可使编写程序变得较容易, 但它并不总是使用中最有效率的数据类型。

有关变量声明的内容将在本章稍后叙述。

一、Variant 数据类型

Variant 数据类型是所有没被显式声明（如未用 Dim、Private、Public 或 Static 等语句）为其他类型变量的数据类型。Variant 数据类型并没有类型声明字符。

Variant 是一种特殊的数据类型，除了定长 String 数据及用户定义类型外，可以包含任何种类的数据。Variant 数据类型的变量包含有字符串、日期、时间、Boolean 或数值，并且它们自动包含的值会转换。Variant 也可以包含 Empty、Error、Nothing 及 Null 特殊值。可以用 VarType 函数或 TypeName 函数来决定如何处理 Variant 中的数据。

数值型 Variant 值需要 16 字节的内存（此点只有在大型的过程或复杂的模块中才具有意义），并且在访问时会比任何显式地定义的数据类型来得慢。显式地定义的数据类型是指在上面介绍的除 Variant 数据类型以外的其他数据类型，如字符串、日期、时间、Boolean 或数值型。

下面的语句创建一个 Variant 变量：

```
Dim myVar
Dim yourVar As Variant
theVar = " This is some text."
```

在例子中，第一句语句 Dim myVar 的作用与第二句语句 Dim yourVar As Variant 的作用是相同的，最后的语句中并没有显式地声明变量 theVar，但在 Visual Basic 中允许使用在程序中未定义的变量，这是由于 Visual Basic 会自动声明隐含性的或自动性的变量。所谓隐含性的声明变量就是指定成 Variant 数据类型。有关自动声明变量的内容将在稍后再叙。

Variant 数据类型的数据值可以是任何整型或实型数，负数时，范围从

$-1.797\ 693\ 134\ 862\ 315\text{E}308 \sim -4.940\ 66\text{E}-324$

正数时，则从

$4.940\ 66\text{E}-324 \sim 1.797\ 693\ 134\ 862\ 315\text{E}308$

通常，数值 Variant 数据保持为其 Variant 中原来的数据类型。例如，如果把一个 Integer 赋值给 Variant，则接下来的运算会把此 Variant 当成 Integer 来处理。然而，如果算术运算针对含 Byte、Integer、Long 或 Single 之一的 Variant 执行，并当结果超过原来数据类型的正常范围时，则在 Variant 中的结果会提升到较大的数据类型。如 Byte 则提升到 Integer，Integer 则提升到 Long，而 Long 和 Single 则提升为 Double。当 Variant 变量中有 Currency、Decimal 及 Double 值超过它们各自的范围时，会发生错误。

用户可以用 Variant 数据类型来替换任何数据类型，这样会更有适应性。如果 Variant 变量的内容是数字，它可以用字符串来表示数字或是用它实际的值来表示，这将由上下文来决定，例如：

```
Dim myVar As Variant
myVar = 98 052
```

在前面的例子中，myVar 内有一实际值为 98 052 的数值。像期望的那样，算术运算符可以对 Variant 变量运算，其中包含数值或能被解释为数值的字符串数据。如果用 + 运算符来将 myVar 与其他的 Variant 或数值类型的数字相加，结果便是一算术和。对于 Variant

数据而言,最为复杂的是两个 Variant 数据之间的运算,有关这一方面的内容用户可以查看有关 Visual Basic 的书,在这里不作详细介绍。

尚未初始化(给定初始值)的 Variant 的值为变量 Empty 值。Empty 的值是随着数据类型不同而不同的,例如内含 Empty 的 Variant 在数值的上下文中表示 0,如果是在字符串的上下文中则表示零长度的字符串(“”)。需要注意的是,用户不应将 Empty 与 Null 弄混,Null 是表示 Variant 变量确实含有一个无效数据,而 Empty 是 Variant 数据类型中的初始设定值,它是有效数据。

二、其他数据类型

虽然 Variant 数据类型可以包含所有的其他数据类型,并且可以实现自动的转换,但是由于 Variant 数据类型在访问时会比任何显式地定义的数据类型来得慢,而且占用的内存要远远超出显式地定义的数据类型,若有下列情形时必须考虑使用其他的数据类型。

- 程序非常大并且使用很多变量。
- 程序的执行速度必须尽可能的快。
- 将数据直接写到随机存储文件中。

除了 Variant 之外,支持的数据类型包括 Byte、Boolean、Integer、Long、Single、Double、Currency、Decimal、Date、Object 以及 String。当一个变量要保存的值是一个单一的类型时,用户可以用比 Variant 更确定的类型,因为这样可以提高程序的运行速度,并且可以节省大量的内存空间。例如,当一个变量所要保存的是一个整数,此时用户就可以将此变量声明为 Integer 类型,而没有必要声明为 Variant 类型。

用户可使用 Dim、Public、Private、Static 语句去声明一个指定类型的变量,例如:

```
Dim X As Integer
```

上述语句声明变量 X 是一个整型,根据表 1 中的数据,它的数值范围介于 -32 768 ~ 32 767 之间。如果试着去设置超出此范围的数值给 X,则会有错误发生。如果试着去指定一个分数给 X,则此分数会自动四舍五入。例如:

```
X = 32 768 ' 产生错误
```

```
X = 5.9 ' 设置 X = 6
```

下面将分别介绍各种数据类型。

1. 数值数据类型

Visual Basic 的数值数据类型包括 Byte、Integer、Long、Single、Double、Currency 等类型。如果变量的值为整数时,用户可以将此变量声明为 Integer 或 Long 类型的变量,因为它们比 Variant、Double、Currency 等类型运行更快,更节省内存空间。整型变量在 For...Next 循环中极有用处。

如果变量的值为小数时,用户可以将此变量声明为 Single、Double 或 Currency 类型的变量。

Currency 数据类型的变量存储为 64 位(8 个字节)整型的数值形式,然后除以 10 000 给出一个定点数,其小数点左边有 15 位数字,右边有 4 位数字。这种表示法的范围可以从 -922 337 203 685 477.580 8 ~ 922 337 203 685 477.580 7,可以看出, Currency 变量值的小数点右边有四位。Currency 数据类型在货币计算与定点计算中很有用,在这种场合精度特别重要。Single 和 Double 类型的浮点数的范围要比 Currency 类型大的多。

数值数据类型还有一种是 Byte 类型, Byte 变量存储为单精度型、无符号整型、8 位 (1 个字节) 的数值形式, 范围在 0~255 之间。Byte 数据类型在存储二进制数据时很有用。

2. 字符串数据类型

如果变量的值始终为字符串, 而不是数值时, 用户可以将此变量声明为 String 类型的变量。同数值类型变量的声明一样, 用户在声明 String 类型变量时, 也用 Dim 语句。

String 之字符码的范围是 0~255。字符集的前 128 个字符 (0~127) 对应于标准的 U.S. 键盘上的字符与符号。这前 128 个字符与 ASCII 字符集中所定义的相同; 后 128 个字符 (128~255) 则代表特殊字符, 例如国际字符、重音符号、货币符号及分数。

字符串有两种: 变长与定长的字符串。变长字符串最多可包含大约 20 亿 (2^{31}) 个字符。定长字符串可包含 1 到大约 64KB (2^{16}) 个字符。

在缺省情况下, 声明后的字符串变量是变长字符串, 即此变量的值的长度可以变化。当然, 用户也可以将字符串变量声明为定长字符串, 只是在声明变量时需要指明此字符串的长度, 如下例就声明了长度为 50 的字符串变量:

```
Dim strAny As String * 50
```

此时, 如果用户赋给 strAny 的字符串中的字符个数少于 50, 则系统将剩下的空位用空格补足; 如果赋给 strAny 的字符串的长度大于 50, 则超出的部分将被撇弃, 例如:

```
Dim strJust4 As String * 4Dim strAny As String
```

```
strAny = " Database" Debug.Print strAny
```

```
strJust4 = strAnyDebug.Print strJust4
```

则显示在调试窗口中的值如下:

```
Database
```

```
Data
```

此例如图 13.1 所示。

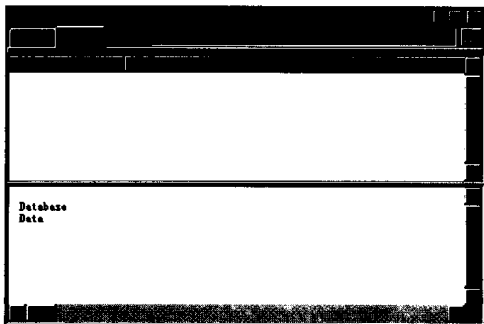


图 13.1

3. 布尔 (Boolean) 数据类型

如果一个变量的值只包括 yes/no 或 on/off 信息, 则用户可将此变量声明为 Boolean 类型。

Boolean 变量存储为 16 位 (2 个字节) 的数值形式, 但只能是 True 或是 False。它的缺省值是 False, 使用关键字 True 与 False 可将 Boolean 变量赋值为这两个状态中的一个。

当转换其他的数值类型为 Boolean 值时, 0 会转成 False, 而其他的值则变成 True。当转换 Boolean 值为其他的数据类型时, False 成为 0, 而 True 成为 -1。

4. 日期数据类型

用户需要使用日期或时间时, 可以声明一个 Date 变量, 用它来存储日期。Date 变量存储为 IEEE 64 位 (8 个字节) 浮点数值形式, 其可以表示的日期范围从 100 年 1 月 1 日到 9999 年 12 月 31 日, 而时间可以从 0:00:00~23:59:59。任何可辨认的文本日期都可以赋值给 Date 变量。日期文字须以数字符号 (#) 括起来, 例如, #January 1, 1993# 或 #1 Jan 93#。

Date 变量会根据计算机中的短日期格式来显示。时间则根据计算机的时间格式 (12 或 24 小时制) 来显示。

当其他的数值类型要转换为 Date 型时, 小数点左边的值表示日期信息, 而小数点右边的值则表示时间。午夜为 0, 而中午为 0.5。负整数表示 1899 年 12 月 30 日之前的日期。

5. 对象数据类型

Object 变量存储为 32 位 (4 个字节) 的地址形式, 其为对象的引用。利用 Set 语句, 声明为 Object 的变量可以赋值为任何对象的引用。下例中就是将 objDb 声明为 Object 变量, 并且将它设为 Northwind 数据库。

```
Const conFilePath As String = _  
    "c: \ Program Files \ Microsoft office \ office \ Sample \ "  
Dim objDb As Object  
Set objDb = OpenDatabase (conFilePath & "Northwind.mdb")
```

13.1.2 数据类型转换

有时用户需要将某种类型的数据转换成另一种类型的数据, 每个函数都可以强制将一个表达式转换成某种特定数据类型。例如, 当用户需要在文本框中显示某个变量的值, 而这个变量是数值类型时, 就必须将这个变量的值转换为字符串类型。下例使用 CStr 函数将一数值转换为 String, 程序如下:

```
Dim MyDouble As Double  
Dim MyString As String  
MyDouble = 437.324 ' MyDouble 为 Double 类型  
MyString = CStr (MyDouble) ' MyString 的内容为 "437.324"
```

在“调试窗口”中用户可看到这两个变量的值, 如图 13.2 所示。

Visual Basic 提供了许多数据类型转换函数, 这些函数如表 13.2 所示, 它的形式以 CStr 函数:

CStr (expression)

为例。函数名称决定返回类型。

表 13.2 Visual Basic 所提供的数据类型转换函数

函数	返回类型	expression 参 数 范 围
CBool	Boolean	任何有效的字符串或数值表达式
CByte	Byte	0~255
CCur	Currency	-922 337 203 685 477.580 8~922 337 203 685 477.580 7
CDate	Date	任何有效的日期表达式
CDbl	Double	负数从: -1.797 693 134 862 32E308~-4.940 656 458 412 47E-324; 正数从: 4.940 656 458 412 47E-324~1.797 693 134 862 32E308
CDec	Decimal	零变比数值, 即无小数位数值, 范围为+/-79 228 162 514 264 337 593 543 950 335。对于 28 位小数的数值, 范围则为+/-7.922 816 251 426 433 759 354 395 033 5; 最小的可能非零值为 0.000 000 000 000 000 000 000 000 000 1
CInt	Integer	-32 768~32 767, 小数部分四舍五入
CLng	Long	-2 147 483 648~2 147 483 647, 小数部分四舍五入
CSng	Single	负数为-3.402 823E38~-1.401 298E-45; 正数为 1.401 298E-45~3.402 823E38
CVar	Variant	若为数值, 则范围与 Double 相同; 若不为数值, 则范围与 String 相同
CStr	String	CStr 返回值依据 expression 参数而定

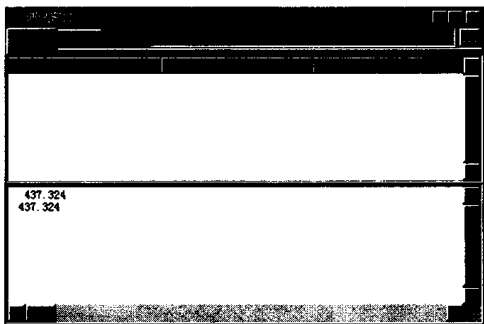


图 13.2

如果传递给函数的 expression 超过转换目标数据类型的范围, 将发生错误。通常, 在编码时可以使用数据类型转换函数来体现某些操作的结果应该表示为特定的数据类型, 而不是

缺省的数据类型。例如，在单精度、双精度或整数运算发生的情况下，使用 CCur 来强制执行货币运算。

应该使用数据类型转换函数来代替 Val，以使国际版的数据转换可以从一种数据类型转换为另一种。例如，当使用 CCur 时，不同的小数点分隔符、千分位分隔符和各种货币选项，依据系统的区域设置都会被妥善识别。

当小数部分恰好为 0.5 时，CInt 和 CLng 函数会将它转换为最接近的偶数值。例如，0.5 转换为 0、1.5 转换为 2。CInt 和 CLng 函数不同于 Fix 和 Int 函数。Fix 和 Int 函数会将小数部分截断而不是四舍五入，并且 Fix 和 Int 函数总是返回与传入的数据类型相同的值。

使用 IsDate 函数，可判断 Date 是否可以被转换为日期或时间。CDate 可用来识别日期文字和时间文字以及落入可接受的日期范围内的数值。当转换一个数字成为日期时，是将整数部分转换为日期，小数部分转换为从午夜起算的时间。CDate 依据系统上的区域设置来决定日期的格式。如果提供的格式为不可识别的日期设置，则不能正确判断年、月、日的顺序。另外，长日期格式，若包含有星期的字符串，也不能被识别。

CVDate 函数也提供对早期 Visual Basic 版本的兼容性。CVDate 函数的语法与 CDate 函数是完全相同的，不过，CVDate 是返回一个 Variant，它的子类型是 Date，而不是实际的 Date 类型。因为现在已有真正的 Date 类型，所以 CVDate 也不再需要了。转换一个表达式成为 Date，再赋值给一个 Variant，也可以达到同样的效果。也可以使用这种技巧将其他真正的数据类型转换为对等的 Variant 子类型。

CDec 函数不能返回独立的数据类型，而总是返回一个 Variant，它的值已经被转换为 Decimal 子类型。

13.1.3 数据类型对比

在 Microsoft Access 中，有四种不同的情况：表“设计”视图、“查询参数”对话框、Visual Basic 以及查询的 SQL 视图可能需要指定数据类型。在 Microsoft Access 中的数据类型与 Visual Basic 的数据类型的对应匹配关系见表 13.3。由表中可以看出，只有少数 Microsoft Access 的数据类型在 Visual Basic 中没有直接匹配的类型，但是它能用其他的 Visual Basic 数据类型来代替它。

Microsoft Access 在数据查询时，还会用到一种数据类型——BINARY 数据类型，在 Visual Basic 中也没有匹配的数据类型，但实际上 Microsoft Access 本身不使用 BINARY 数据类型，它仅用于其他支持 BINARY 数据类型的数据库产品的查询或链接表中。

表 13.3 字段数据类型与 Visual Basic 数据类型的对应匹配关系

字段数据类型	Visual Basic 数据类型
Autonumber (Long Integer)	Long
Autonumber (Replication ID)	没有匹配的 Visual Basic 数据类型
Currency	Currency
Date/Time	Date
Hyperlink	没有匹配的 Visual Basic 数据类型
Memo	String
Number (Byte)	Byte
Number (Integer)	Integer
Number (Long Integer)	Long
Number (Single)	Single
Number (Double)	Double
Number (Replication ID)	没有匹配的 Visual Basic 数据类型
OLE Object	Byte 数据类型数组
Text	String
Yes/No	Boolean

13.2 变 量

变量是用来存放 Visual Basic 程序运行时的值，就一般情况而言，在变量能使用之前，要先声明变量，当然，前面已经提到过，在 Visual Basic 中，可以不必声明变量，下面将做详细介绍。

13.2.1 变量的声明

在编写 Microsoft Access 的过程或模块时，用户一般使用 Dim 语句来声明变量。Dim 语句的语法为

```
Dim variablename [As type]
```

在 Dim 语句中，用户必须给变量起一个变量名称，在给变量起名时需要注意一些事项，在以后的篇章中有详细叙述。

As type 子句是用来定义这个变量的数据类型的，这些数据类型就是上面介绍的，如果用户省略了这个子句，Visual Basic 就缺省的认为这个变量是 Variant 类型。

在 Microsoft Access 中声明变量时，Access 会自动显示数据类型列表，用户可以从列表中选取数据类型，当然也可以键入数据类型，图 13.3 即为在定义变量时 Access 显示的数据类型列表。

一个声明语句可以放到过程中以创建属于过程的级别的变量，或在声明部分可将它放到模块顶部，以创建属于模块级别的变量。

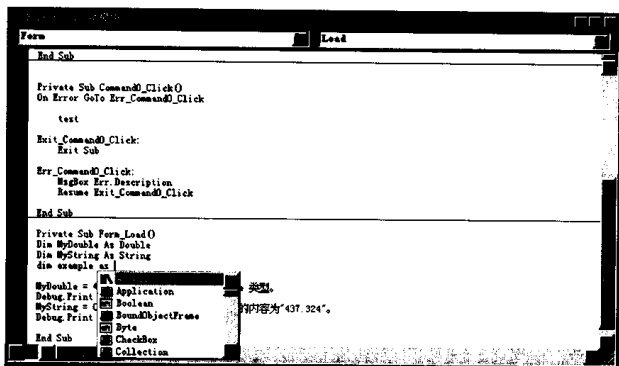


图 13.3

下面的示例：

```
Dim strName As String
```

创建了变量 strName 并且指定为 String 数据类型。如果该语句出现在过程中，则变量 strName 只可以在此过程中被使用。如果该语句出现在模块的声明部分，则变量 strName 可以被此模块中所有的过程所使用，但是不能被同一工程中不同的模块所含过程使用。为了使变量可被工程中所有的过程使用，则在变量前加上 Public 语句，如以下的示例：

```
Public strName As String
```

为了便于编写程序，用户可以在一个语句中声明几个变量；而为了指定数据类型，必须将每一个变量的数据类型包含进来。在下面的语句

```
Dim intX As Integer, intY As Integer, intZ As Integer
```

中，变量 intX、intY 与 intZ 被声明为 Integer 类型。而在下面的语句

```
Dim intX, intY, intZ As Integer
```

中，变量 intX 与 intY 被声明为 Variant 类型；只有 intZ 被声明为 Integer 类型。

在 Visual Basic 中可以简单地通过一个赋值语句来隐含声明变量。例如用户可以用如下语句

```
Function SafeSqr (ByVal dblNum As Double) As Double
    DblTemp = Abs (dblNum)
    SafeSqr = Sqr (dblTemp)
End Function
```

定义一个函数。在定义此函数时，并没有定义 DblTemp 这个变量。由于 Visual Basic 具有隐含声明的功能，用户可以利用变量名来创建一个变量，而没有必要一定要在使用变量前，利用 Dim 语句来声明它。

虽然这样隐含声明十分方便，但是它也会导致一些难以发现的错误，主要是由于用户在输入代码时，很容易将变量名输错。如假设用户编写了如上的一段程序，但在输入时变成了这样：

```
Function Safesqr (ByVal dblNum As Double) As Double
    DblTmp = Abs (dblNum)
    SafeSqr = Sqr (dblTmp)
End Function
```

这两段程序似乎是一样的，但在输入 DblTmp 时错输成 DbtTmp，由于 Visual Basic 具有隐含定义变量的功能，系统就分不清 DblTmp 是一个新的变量，还是输入时发生错误，于是 Visual Basic 就将 DbtTmp 看作是一个新的变量，并且在函数中没有对它进行赋值，它的值就还是 Variant 类型的缺省值 0，因此这个函数的返回值始终为 0。

由于变量名的拼写错误是十分常见的，并且如果不给予提醒是十分难以发现的，解决这一问题的最简单的方法就是对所有的变量作声明。这样如果程序中有变量名发生了拼写错误，用户将会得到通知，Visual Basic 中提供了这一选项，但并不强迫用户一定要使用它。使用显示声明还有一个好处就是充分利用内存资源。

如果用户选择了隐含声明，那么所有的变量都隐含声明为 Variant 类型。Variant 类型变量比大多数其他类型的变量需要更多的内存资源。显示声明变量为指定的数据类型，则应用程序将更有效，而且显示声明所有变量减少了命名冲突以及拼写错误的发生率。使用显示声明可避免错误输入已有的变量名，也可在变量的范围不清晰时避免代码混乱。

如果不想使 Visual Basic 生成隐含声明，可以将 Option Explicit 语句放置于模块中所有的过程之前。这是一个不出现在过程中的语句，它要求对模块中所有的变量做显示声明。如果模块包含 Option Explicit 语句，则当 Visual Basic 遇到一个先前未定界的变量或拼写错误，会发生编译错误。

用户希望与模块相连的所有事件过程和函数都能使用这一限制，应该将 Option Explicit 语句放在模块代码窗口中的通用部分。

在 Microsoft Access 中，可以确保所有的新模块均能自动将 Option Explicit 语句包含到模块中。请在“工具”菜单上，单击“选项”，然后单击“模块”选项卡并选中“要求声明变量”选项，如图 13.4 所示。

一旦选中了该选项，所有其他由 Microsoft Access 创建或打开的数据库，都将自动设置该选项。

但是用户在选择该选项后，只有新模块才具有 Option Explicit 设置。所以必须将 Option Explicit 语句输入到任何已存在模块的声明节中；或者，当模块打开时，在“工具”菜单上，单击“选项”，然后单击“模块”选项卡，并且选中“要求声明变量”选项。要将 Option Explicit 放置在通用部分中，用户可以遵循下列步骤。

首先对于某一个模块，单击工具条中的代码按钮，打开代码窗口，然后在对象列表框的对象列表中选择通用对象，在过程列表中选择声明，在代码窗体中键入 Option Explicit。

在用户设置了 Option Explicit 以后，Visual Basic 将不再允许用户使用那些未经声明的变量，此时，如果用户要使用一个未经定义的变量时，将会出现一条错误信息，仍以 Safesqr 为例，此时将出现如图 13.5 所示的情况。



图 13.4

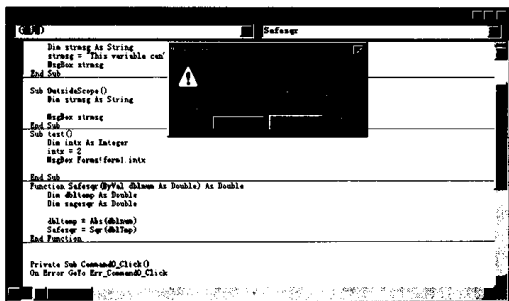


图 13.5

在一般情况下，用户要尽量使用 Option Explicit 语句，这样可以避免许多麻烦。

13.2.2 变量的范围与可见性

范围指的是变量、常数或过程被其他过程使用的可用性。当声明一个变量时已决定了它的范围。例如，用户在过程中定义了一个变量，那么，只有在这个过程中的变量才能使用此

变量的值，它的范围级别是过程级别。但是有时候，用户希望变量的值能被模块中的所有过程调用，甚至被所有的模块所调用，使用范围不同的变量必须用不同的方法进行定义。

Microsoft Access 的范围和可见性有三个范围级别：过程级别、私有模块级别和公共模块级别。过程级范围应用于过程中声明的变量和常量。公用或私有模块级范围则应用于包含在模块上部的变量、常量、用户自定义类型以及 Declare 语句，或者应用于模块中定义的程序、函数过程等。表 13.4 显示了不同的元素在不同范围级别中的区分。

表 13.4 不同元素在不同范围级别中的区分

范围级别	元素类型	范 围	注 释
过程级	变量	私有	
	常量	私有	
模块级	变量	除非声明为 Public，否则为私有	
	常量	除非前面有 Public 关键字，否则为私有	在类模块中不能声明公用常量
	用户自定义类型	除非前面有 Private 关键字，否则为公用	类模块中声明的用户自定义类型前面必须具有 Private 关键字
	过程	除非前面有 Private 关键字，否则为公用	窗体和报表模块中的事件过程默认为私有过程

一、定义过程的级别范围

在程序内定义的变量或常数是不能在过程外看到的，只有声明此变量的过程可以使用它们，这样的变量可以称之为局部变量。用户可以用 Dim 或 Static 关键字来定义这些局部变量，如

```
Dim intTemp As Integer
```

```
Static intPermanent As Integer
```

用 Dim 定义的局部变量只有在它所在的过程运行时才有值，而由 Static 定义的局部变量在整个程序运行期间均有值。由于由 Dim 定义的局部变量只有在运行时才有值，因此它可以作为中间变量用于中间计算；并且由于在相应的过程外是看不到这个变量的，因此用户可以在不同的过程定义同一个名称的变量。例如，用户可以分别在三个过程中定义 intTemp 变量，不同过程中的 intTemp 不相互影响。

在下面的示例中，第一个过程显示一个包含字符串的信息框；而第二个过程则显示一个空白的信息框，因为变量 strMsg 对于第一个程序是本地的。

```
Sub LocalVariable ()
```

```
    Dim strMsg As String
```

```
    strMsg = " This variable can't be used outside this procedure."
```

```
    MsgBox strMsg
```

```
End Sub
```

```
Sub OutsideScope ()
```

```
    MsgBox strMsg
```

```
End Sub
```

上例的结果如图 13.6 和 13.7 所示, 其中图 13.6 是在 Command1_Click 事件中运行 LocalVariable 子过程所得的结果; 而图 13.7 是在 Command1_Click 事件中运行 OutsideScope 子过程所得的结果。

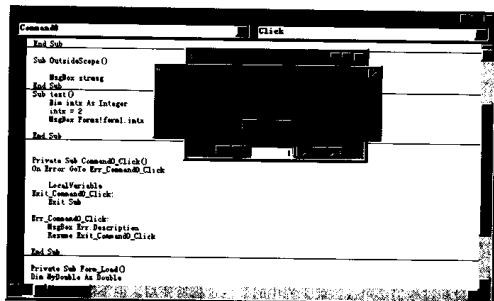


图 13.6

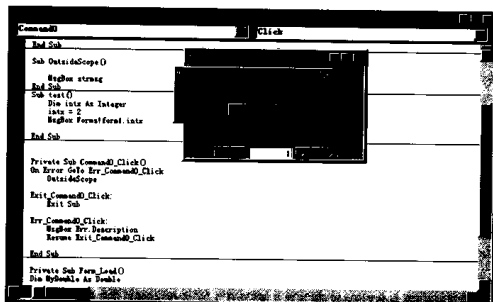


图 13.7

二、定义私有模块级别的范围

在模块的声明部分中, 用户可以定义模块级别变量和常数。模块级别变量有两种: 私有或公有的。在工程中, 公有变量对于所有模块中的所有过程是可用的; 而私有变量只对于所属模块中的过程是可用的。用户可以在窗体、报表和标准模块顶部的声明部分中, 用 Dim 或 Private 定义模块级别的变量。由 Dim 语句在声明部分中所声明变量, 其范围的缺省值为私有。然而, 若在变量的前面加上 Private 关键字, 则其在代码中的范围将更为明显, 但实

际上,与用 Dim 定义的变量是没有区别的。下面是定义变量的示范语句:

```
Dim intTemp As Integer
```

```
Private intTemp As Integer
```

在下面的示例中,字符串变量 strMsg 可被定义在此模块中的任何过程使用。当第二个过程被调用时,它会在对话框中显示出字符串变量 strMsg 的内容。

```
' 添加下面的语句到模块的声明部分
```

```
Private strMsg As String
```

```
Sub InitializePrivateVariable ()
```

```
    strMsg = " This variable can't be used outside this module."
```

```
End Sub
```

```
Sub UsePrivateVariable ()
```

```
    MsgBox strMsg
```

```
End Sub
```

公共过程若是在标准模块或对象类模块中的话,则任何引用工程都可以使用它。为了限制当前工程中模块内的所有过程,可以在模块的声明部分加上 Option Private Module 语句。如此则公有变量和过程仍然可被当前工程中的其他过程所使用,但不能被引用工程所使用。

三、定义公共模块级别的范围

如果声明一个模块级别变量为公用,它将可被工程中的所有过程使用。声明公共模块级别的变量需要用 Public 关键字。同私有的模块级别变量一样,声明公共变量也要在模块顶部的声明部分,在过程中是不能用 Public 来声明公共变量的。定义公共模块变量肯定要在两个以上的模块中使用,它可以在任何一个模块中定义,但最好将所有的公共变量集中在一个模块中定义,这样便于代码的阅读以及变量的查找。

在下面的示例中,字符串变量 strMsg 可被工程中模块的所有过程使用。

```
' 包括在模块的声明部分中
```

```
Public strMsg As String
```

除了事件过程之外,所有过程的缺省值都是公共的。当 Visual Basic 创建一个事件过程时,在过程的声明前面会自动的加上 Private 关键字。对于其他的过程来说,若不想使它为公共的,必须利用 Private 关键字来显示地声明过程。

可以从引用的工程中,去使用它定义在标准模块或类模块中的公共过程、变量以及常数;然而,必须先设置一个对它们所定义工程的引用。公共过程、变量以及常数若不是定义在标准模块或类模块,例如窗体模块或报表模块中,都不可以被引用工程所使用,因为这些模块对于所驻留的工程而言是私有的。

对于定义的变量,还需要注意以下几点。

- 用户在标准模块中定义的公用元素,可用于当前数据库及引用数据库中的所有过程,除非该模块包含 Option Private Module 语句。如果包含该语句,则公用元素只能用于当前数据库中的过程。

- 类模块,如数据库中的窗体、报表等默认为私有元素。用户在类模块中定义的公用元素只能用于当前数据库,不能在引用数据库中使用。

- Microsoft Access 数据库中定义的变量、常量、过程、用户自定义类型以及 Declare 语

句不能在 Microsoft Access 以外的应用程序的项目中使用。

- 要引用类模块中的模块级变量或过程，必须使用类模块的名称。例如，假设在名为 Class1 的类模块中声明了一个名为 varOrders 的模块级变量，则在标准模块或“调试”窗口中使用该变量时，必须以 Class1.varOrders 的方式引用它。当窗体名称为两个或多个字时，应在窗体的类名称上加上括号。

- 类模块中定义的模块级变量只在类实例存在期间可用，当类实例被取消后，变量也就不再可用。

- 在标准模块中初始化一个模块级变量后，不管代码是否运行，该变量将保持其值，除非变量被指定为其他值、代码重置或关闭数据库。

- 如果类模块中创建的过程与标准模块中某一过程同名，则通常调用标准模块中的过程，除非使用了类模块的类名来完整引用该过程。

13.3 范围和变量的名称

13.3.1 Visual Basic 的命名规则

当在 Visual Basic 的模块中为过程、常数、变量以及参数命名时，用户需要参考使用下列的规则。

- 第一个字符必须使用英文字母。

- 不能在名称中使用空格、句点 (.)、惊叹号 (!) 或 @、&、\$、# 等字符。

- 名称的长度不可以超过 255 个字符。

- 通常，使用的名称不能与 Visual Basic 本身的 Function 过程、语句以及方法的名称相同，必须退出使用与程序语言的关键字相同的名称。若所使用的内在语言函数、语句或方法与所指定的名称相冲突，则必须显示地识别它。常规会在内建函数、语句或方法的名称之前加上关联的类型库的名称。例如，如果有一个名为 Left 的变量，则只能用 VBA.Left 来调用 Left 函数。

- 不能在范围的相同层次中使用重复的名称。例如，不能在同一过程中声明两个命名为 intTemp 的变量。然而，可以在同一模块中声明一个私有的命名为 intTemp 的变量和过程级别的命名为 intTemp 的变量。

同时需要注意的是，Visual Basic 不区分大小写，但它会在名称被声明的语句处保留大写，因此 Inttemp 与 intTemp 是同一个变量。

13.3.2 命名冲突

用户在 Microsoft Access 中编写 Visual Basic 程序，由于种种原因，用户也许会试着创建或使用先前已定义的标识符，这时就会发生命名冲突。在某些情况下，命名冲突会生成像“发现二义性的无效名称”或“在当前范围内的声明重复”等错误。命名冲突如果没有被检测到，则会导致代码错误，如此将会生成错误的结果，特别是在第一次使用之前没有明确声明所有的变量。

可以由了解数据、对象和过程中标识符的范围特性，来避免大部分的命名冲突。前面已经介绍过 Visual Basic 拥有三个范围级别：过程级别、私有模块级别和公共模块级别。

当一个标识符有下列情形时可能发生命名冲突。

- 在一个以上范围级别中为可见的。
- 在同一范围级别中具有两种不同的意义。

举例来说，在分离的模块中可能有相同的名称，因此，可以在模块名称为 Mod1 和 Mod2 中定义过程名称为 MySub。如果每一个过程都被同属于一个模块的程序所调用，则不会有命名冲突发生。然而，如果 MySub 是从第三个模块中用如下语句

MySub

调用的，则发生错误，因为没有任何的条件来区别这两个 MySub 过程。此时用户可以在每一个标识符前加上一个限定符，其可由模块名称或是工程名称来组成，如此可解决大部分的命名冲突。例如用户要在 Mod2 的 MySub 中调用 Mod1 中的 MySub 作为参数，则可用如下代码

YourProject.Mod2.MySub YourProject.Mod1.MySub

来完成。由此看出，用户可以使用任何限制符组合去区分相同的标识符。Visual Basic 将每一个对标识符的引用与一个符合标识符的声明做对比。例如，如果 MyID 在工程的两个模块 (Mod1 与 Mod2) 中被声明为 Public，则在 Mod2 中指定声明的 MyID，并不需要在 Mod2 里加上限制符；但若在 Mod1 中指定上述的 MyID，则必须以 Mod2.MyID 来定位它。当 Mod2 在不同但直接引用的工程中，则上述方式仍然可行。然而，如果 Mod2 是在一个间接引用的工程中，也就是一个被直接引用工程所引用的工程，则引用 Mod2 中的变量名称 MyID 时需要用工程名来加以限制。如果是从第三个直接引用的模块来引用 MyID，则在匹配时会以第一个找到符合下面规则的声明为主。

直接引用的工程，以它们出现在“工具”菜单中的“引用”对话框的顺序为准。

由命名冲突所引起的典型错误包括有仿真两可的名称、重复声明、未声明标识符以及找不到过程。在模块开头加上 Option Explicit 语句，可以强迫在变量被使用前对变量做明确的声明，如此可以避免一些可能的命名冲突以及和标识符有关的错误。

13.3.3 命名冲突时的范围

以下所列是在 Microsoft Access 中可能遇到的一些命名冲突。

1. 标准模块和类模块中的过程及变量

在 Microsoft Access 中，如果标准模块和类模块中包含名称相同的公用过程，在不特别指明引用的情况下，将调用标准模块中的过程。如果要调用类模块中的过程，必须使用类名进行完整引用，对标准模块和类模块中的同名变量也一样。

例如，在订单数据库中，在标准模块中有一名为 CountOrders 的函数，用户在其他模块中要调用这个函数时，可用如下代码：

CountOrders

在名为“订单”的窗体模块中也有一个名为 CountOrders 的函数，用户在其他模块中可以使用代码

Form.Orders.CountOrders

调用该函数。

2. 不同级别的变量具有相同的名称

如果一个模块级的变量与该模块某一过程中的过程级变量名称相同，则必须使用模块名称来完整引用模块级变量，该规则对标准模块、窗体模块以及报表模块同样适用。例如，在名为 Important Functions 的标准模块中同时声明了名为 varScope 的模块级和过程级变量，则可以使用 [Important Functions].varScope 来引用模块级变量。需要说明的是在同一模块中最好避免使用同名变量。

下面这段程序的结果如图 13.8 所示。

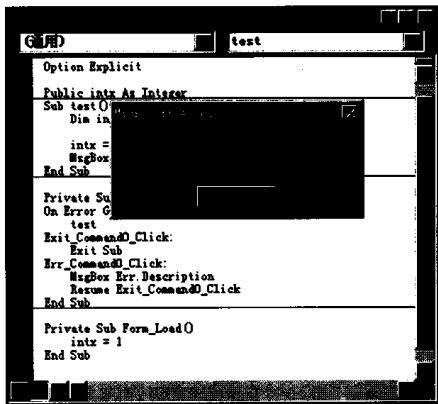


图 13.8

- 这是 Form1 的窗体模块中的代码，用户可以看到在 test 子过程中与窗体
- 模块中均定义了 inx 变量，但是在 test 中的 inx 永远是指局部变量，而不
- 是模块级的全局变量

```
Option Explicit
```

```
Public inx As Integer
```

```
Sub test ()
```

```
    Dim inx As Integer
```

```
    inx = 2
```

```
    MsgBox Forms! form1.inx
```

```
End Sub
```

```
Private Sub Command0_Click ()
```

```
On Error GoTo Err_Command0_Click
```

```
test
```

```
Exit_Command0_Click;
Exit Sub
Err_Command0_Click:
MsgBox Err.Description
Resume Exit_Command0_Click
End Sub

Private Sub Form_Load ()
    intx = 1
End Sub
```

3. 类模块中的常量、变量以及过程、属性名称

窗体或报表模块中的常量、变量或过程名称不能与窗体或报表的属性或方法名称相同。例如，要创建一个名为 Name 的 Sub 过程，将造成 Microsoft Access 的编译错误，因为该窗体具有一个名为 Name 的属性。

有时用户命名变量时刚好变量名与模块中的过程、属性的名称相同，此时就会发生冲突。如果在窗体或报表模块中，局部变量的名称与模块中的控件名称相同，在过程中引用控件时必须使用 Me 关键字，从下例：

```
Private Sub Form_Click ()
    Dim Text1, Caption
    ' 假设在此窗体中也有一个控件名为 Text1, Caption 是控件的属性
    Text1 = " Variable"           ' Text1 是局部变量
    Me.Text1 = " Control"        ' 要对控件进行操作必须使用 Me 关键字
    Text1.Top = 0                 ' Visual Basic 将提示这句语句错误
    Me.Text1.Top = 0              ' 要对控件进行操作必须使用 Me 关键字
    Caption = " Orders"          ' Caption 是局部变量
    Me.Caption = " Orders"       ' 要对属性进行操作必须使用 Me 关键字
End Sub
```

就可以看到这一点。

在模块中定义的变量是不允许与该模块中的过程名发生冲突的。

由于变量发生冲突时，它的规则十分复杂，在原则上，用户在定义变量时应该使用不同的变量名称，增加程序的可读性，并减少许多麻烦。

13.3.4 变量的生存周期

变量保留其值的这段时间，称为生存周期。变量的值可能在整个生存周期内都在改变，但它仍然保留着一些值。当变量失去了范围之后，它也就不再保存着任何一个值。

当过程开始运行时，所有的变量都会被初始化。一个数值变量会初始化成 0，变长字符串被初始化成零长度的字符串（""），定长字符串会被填满 ASCII 字符码 0 所表示的字符或是 Chr (0)，Variant 变量会被初始化成 Empty，用户自定义类型中每一个元素变量会被当成个别变量来做初始化。

当声明一个对象变量时，内存中虽有保留空间，但它的值会被设置成 Nothing，直到利用 Set 语句对它指定一个对象引用。

如果在代码的运行期间, 变量的值一直没有改变, 则它会继续保有它的初始值直到它丢失范围为止。

Dim 语句声明过程的级别变量将保留一个值, 直到此过程退出为止。如果该过程调用其他的过程, 则在这些过程正在运行的同时, 属于调用者过程的变量也保留它的值。

有时候, 用户希望保存某个过程局部变量的值, 那么用 Static 关键字来声明就可以达到这一点。如果过程的级别变量是用 Static 关键字来声明的, 则只要代码正在任何模块中运行, 此变量仍会保留它的值; 而当所有的代码都完成运行后, 变量会失去它的范围和它的值, 所以它的存活期和模块级别的变量是一样的。在

```
Function RunningTotal (ByVal dblNum As Double) As Double
    Static dblAccumulate As Double

    dblAccumulate = dblAccumulate + dblNum
    RunningTotal = dblAccumulate
End Function
```

中, 变量 dblAccumulate 被定义为 Static, 只要程序在此模块中运行, 它就保留有值。

模块级别的变量与静态变量是不同的, 在标准模块或类模块中变量会保留它的值, 直到停止运行代码。在对象类模块中, 只要仍有一个属于此对象类的实例存在, 则变量会一直保留它的值。模块级别的变量会一直占用内存资源, 直到重新设置它们的值, 所以只有在必要时才使用它们。

如果在 Sub 或 Function 语句前加上 Static 关键字, 则在此过程中所有过程级别的变量的值被保留在调用期间, 无论它们是用 Dim 定义的还是用 Static 定义的。

13.4 对象变量

在 Microsoft Access 中, 可以将变量定义为特定的对象类型, 例如 Object 或 Variant, 这就是对象变量。特定的对象类型包括 Microsoft Access 对象、数据访问对象 (DAO)、Visual Basic Collection 对象、类模块实例和自动化对象。以下的示例:

```
Dim frmOrderForm As Form      ' 声明为窗体对象
Dim tdfOrders As TableDef      ' 声明为表对象
Dim colThings As New Collection ' 声明为新的 Collection 对象
Dim obj As New Class1          ' 声明为一个新的类模块实例
```

显示各种可用对象。

在定义完对象变量以后, 内存中虽有保留空间, 但它的值会被设置成 Nothing, 直到利用 Set 语句对它指定一个对象引用。如果在对象变量声明中使用 New 关键字, 将新建一个对象, 并将变量自动指向该对象, 而不需要使用 Set 语句。

在窗体模块或报表模块中使用 Me 关键字, 可以引用与该模块相关联的窗体或报表。关键字 Me 可以在引用当前窗体时不用提供完整的名称, 因此用户不需要知道窗体的名称即可引用窗体。

例如, 在标准模块中创建如下过程:

```
Sub ChangeFormColor (frmCurrent As Form)
```

```
frmCurrent.Section (acDetail) .BackColor = RGB (Rnd * 256, _  
    Rnd * 256, Rnd * 256)
```

```
End Sub
```

可用来改变窗体“主体”节的颜色。在窗体模块中可以调用该过程，并将 Me 关键字作为参数，传递给过程。Me 关键字代表调用了该过程并在其中运行此过程的窗体。

例如，可在窗体节的 Click 事件中调用该过程。将“主体”节的 OnClick 属性设置为 [事件过程]，然后在窗体模块中创建如下的 Sub 过程：

```
Private Sub Detail_Click ()  
    ChangeFormColor Me  
End Sub
```

当然，用户也可以使用窗体的名称来应用窗体。例如，如果窗体的名称为“订单”，那么也可以使用下面的语句

```
ChangeFormColor Forms! Orders
```

调用该过程。

前面已经介绍过，创建对象变量需要完成两件工作，首先必须声明对象变量，在声明对象变量后还需要指定对象变量到某一对象。

13.4.1 声明一个对象变量

使用 Dim 语句或其他的声明语句 (Public、Private 或 Static) 去声明对象变量。一引用对象的变量必须是 Variant、Object，或是一个对象的指定类型，如窗体 (Form)、报表 (Report)、表 (TableDef) 等。例如，下列声明：

```
' 声明 MyObject 为 Variant 数据类型  
Dim MyObject  
' 声明 MyObject 为 Object 数据类型  
Dim MyObject As Object  
' 声明 MyObject 为 Font 类型  
Dim MyObject As Font
```

是有效的。如果用户定义的对象变量要等到过程运行时才知道指定的对象类型，可以利用 Object 数据类型来声明。用户可以使用 Object 数据类型来创建对任何对象的一般的引用。

如果知道指定的对象类型，必须将对象变量声明成所知道的对象类型。例如，如果应用程序包含 Sample 对象类型，则可以用下列的语句：

```
Dim MyObject As Object      ' 声明为一般的对象  
Dim MyObject As Sample     ' 只声明为 Sample 对象
```

来声明一个对象变量为 Sample 对象类型。

声明指定的对象类型系统会提供自动的类型检查，并且在输入对象类型时会自动显示对象类型，输入代码更快，并增加可读性。

13.4.2 赋值对象变量给对象

在定义完对象变量后，用户必须使用 Set 语句赋一个对象给对象变量，此时该对象变量才有意义。可以用对象表达式或是 Nothing 给对象赋值。例如，下列的对象变量赋值：

```
Set MyObject = YourObject      ' 赋值对象引用
Set MyObject = Nothing         ' 中断关联
```

是有效的。

在 Microsoft Access 中, 还可以同时使用 New 关键字和 Set 语句来声明对象变量, 并赋值一对象给它。例如:

```
Set MyObject = New Object      ' 创建和赋值
```

设置一个对象变量等于 Nothing, 会中断此对象变量与任何特定对象的关联, 如此可预防因意外的改变变量而更改对象。在关闭关联对象后, 对象变量总是设置为 Nothing, 所以可以检测对象变量是否指到有效的对象。例如:

```
If Not MyObject Is Nothing Then
' 变量引用有效的对象
...
End If
```

当然, 该检测不能绝对的决定用户是否已关闭包含对象变量所引用对象的应用程序。

13.5 创建自定义数据类型

如果学过 C 语言的读者肯定对 C 语言中的结构印象很深, 在 Visual Basic 中的自定义数据类型的功能就与 C 语言中的结构相类似, 当用户想只使用单一的变量来表示许多相关的信息时, 自定义数据类型是十分有用的。

13.5.1 声明自定义数据类型

使用 Type 语句可以创建用户自定义数据类型。用户只能在标准模块的声明部分使用 Type 语句。如果要在类模块中声明用户定义类型, 则必须在 Type 语句之前加上 Private 关键字; 如果没有加上 Private 关键字, Microsoft Access 将产生编译错误。

用户自定义数据类型同其他数据类型一样, 也可以用相应的关键字来将它定义为 Private 或 Public, 例如:

```
Private Type YourType
Public Type YourType
```

当用户需要使用同一对象的相关信息时, 用户可以定义用户自定义类型。它可包含所有的相关信息, 例如用 MyType 这一数据类型来定义用户信息, 可包含姓名、生日、性别等。也就是说, 用户在定义时类型可包含一个或多个某种数据类型的数据元素、数组或一个先前定义的用户自定义类型。MyType 定义如下所示:

```
Type MyType
    MyName As String      ' 定义字符串变量存储一个名字
    MyBirthDate As Date   ' 定义日期变量存储一个生日
    MySex As Integer      ' 定义整型变量存储性别 (0 为女, 1 为男)
End Type
```

用户只能在模块级别中使用 Type 语句。一旦已使用 Type 语句声明用户定义的类型, 则可以在声明范围的任意地方用 Dim、Private、Public、ReDim 或 Static 来声明该类型的变

量。不同的声明方法将会得到不同使用范围——局部变量、私有模块级变量、公有模块级变量的该类型的变量，例如：

```
Dim infoMy As MyType
```

表 13.5 说明了声明用户自定义数据类型的位置与该类型变量的范围的关系。

表 13.5 自定义数据类型的位置与该类型变量的范围的关系

过程/模块	可将自定义数据类型声明为	该类型变量可为
过程	不支持	不支持
标准模块	Private 或 Public	Private 或 Public
类模块	Private	Private

在标准模块中，用户定义的类型被默认为公用的，如果用户想将此类型定义为私有，则可用 Private 关键字改变这种可见性；但在类模块中，用户定义的类型只能是私有的，而且不能用 Public 关键字改变可见性。

13.5.2 变量赋值

给声明完的用户自定义数据类型的变量赋值与对象的属性设置一样，如

```
infoMy.MyName = "Smith"
```

```
infoMy.MyBirthDate = #9/9/74#
```

对该变量值的引用也和对象属性的引用一样，如

```
If infoMy.MyBirthDate > #1/1/70# Then
```

```
...
```

```
End If
```

用户还可以将统一类型的变量相互赋值，例如：

```
infoMy = info Your
```

假设 info Your 也是 MyType 类型，在赋值完以后，变量 infoMy 的组成元素的值与 infoYour 的组成元素的值完全相同。

13.5.3 用户自定义类型中含有数组

在用户定义的类型中可以含有固定大小的数组，例如：

```
Type SystemInfo
```

```
varCPU As Variant
```

```
lngMemoru As Long
```

```
strDiskDrives (1 To 100) As String ' 固定大小的数组
```

```
intVideoColors As Integer
```

```
curCost As Currency
```

```
dtePurchase As Variant
```

```
End Type
```

需要注意的是，在声明用户定义类型中的固定大小的数组时，必须用数字或常数而不是变量来声明数组的维数，如下代码：

```
Type SystemInfo
```

```
varCPU As Variant
```

```
lngMemoru As Long
```

```
strDiskDrives (numDisk) As String ' numDisk 是程序中的一个变量
```

```
intVideoColors As Integer
```

```
curCost As Currency
```

```
dtePurchase As Variant
```

```
End Type
```

将导致 Microsoft Access 编译错误, 如图 13.9 所示。

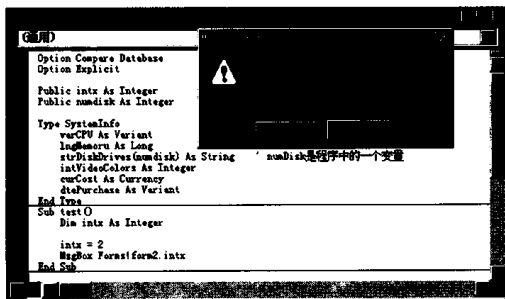


图 13.9

在用户定义的类型中可以含有固定大小的数组, 例如:

```
Type SystemInfo
```

```
varCPU As Variant
```

```
lngMemoru As Long
```

```
strDiskDrives () As String ' 动态数组
```

```
intVideoColors As Integer
```

```
curCost As Currency
```

```
dtePurchase As Variant
```

```
End Type
```

与前面讲述的变量的引用、赋值一样, 含有数组的自定义类型的变量的引用、赋值与对象属性的引用、赋值相同, 例如:

```
Dim sysMine As SystemInfo
```

```
sysMine.strDiskDrives (1) = "1.44 MB"
```

用户还可以定义自定义类型数组:

```
Dim sysAll (1 To 100) As SystemInfo
```

按照同样的方法, 可以对该数组中的元素进行赋值与引用, 例如:

```
sysAll (5) .varCPU = "386SX"  
sysAll (intX) .strDiskDrives (2) = "100M SCSI"
```

13.5.4 自定义数据类型嵌套

有时由于对象比较复杂,在定义数据类型时,需要用到数据类型嵌套。数据类型嵌套是指在定义一个数据类型时,它的元素中包含了其他的自定义数据类型,这样可以将任何复杂的对象用自定义数据类型来表述,例如:

```
Type DriveInfo  
    DriveType As String  
    Size As Long  
End Type  
Type SystemInfo  
    varCPU As Variant  
    lngMemoru As Long  
    strDiskDrives (16) As DriveInfo  
    curCost As Currency  
    dtePurchase As Variant  
End Type
```

下面的代码:

```
Dim sysAll (100) As SystemInfo  
sysAll (1) .strDiskDrives (1) .DriveType = "Floppy"
```

介绍了如何使用这样的嵌套。

13.6 使用常数

用户在编写 Visual Basic 程序时,代码中可能包含经常出现的常数值,如数据存储的起始时间,或某些数值非常难以记忆并且无明确意义,如颜色,在这种情况下,将这些数值定义为常数,可使代码更容易读取与维护。常数是一个有意义的名称,它可以替换一个号码或字符串而且值不会改变。常数一经定义,用户就不能在代码中修改它,或如同操作变量般的赋一个新值给常数。

Microsoft Access 支持三种类型的常量:

- 符号常量——可用 Const 语句创建,并且可在模块中使用;
- 固有常量——这是 Microsoft Access 或引用库的一部分;
- 系统定义常量——True、False 和 Null。

13.6.1 符号常量

用户可能经常要在代码中反复使用相同的值,或者发现代码中经常有一些没有明显意义的数字。在这种情况下,就可以在出现数字或字符串的地方使用具有明显含义的符号常量或用户定义的常量来增加程序代码的可读性与可维护性。例如以下代码:

```
Calculate.Visible = True
```

将会比

```
Calculate.Visible = -1
```

可读性强得多，尽管两者等效。

需要声明的常数都是符号常量。通过声明一个常数，可以将有意义的名称赋予一个值，因此在编写代码时，常常需要使用常数，除了使用已经可以在对象浏览器中可以查到的常数以外，其他常数必须在模块的声明部分进行声明。在 Microsoft Access 中，用户必须使用 Const 语句去声明一个常数，并且设置它的值；而在常数声明完之后，则不能加以更改或赋予新值。

在声明部分，可以在过程中或者在模块顶部声明常数。在上一节中，已经介绍过，常数与变量一样，也有三个范围级别：过程级别、私有模块级别和公共模块级别。同变量一样，当声明一个常数时已决定了它的范围。用户定义常量可以使用 Const 语句，它的语法结构如下：

```
[Public | Private] Const constname [As type] = expression
```

Const 语句的语法在表 13.6 中说明。

表 13.6 Const 语句语法说明

部 分	描 述
Public	可选的参数，模块级别所使用的关键字，用这些关键字声明所有模块中所有过程中的可用常数。在过程中不允许
Private	可选的参数，模块级别的关键字，用来声明只在已声明的模块中才可以使用的常数。在过程中不允许
Constname	必需的参数，它是常数的名称，需要遵循 Visual Basic 变量命名约定，命名约定详见变量部分
Type	可选的参数，常数的数据类型，可以是 Byte、Boolean、Integer、Long、Currency、Single、Double、Decimal（当前不支持）、Date、String 或 Variant。对每个已声明的常数使用 As type 子句
Expression	必需的参数，它可以是文字、其他常数或除 Is 之外的任何算术及逻辑运算符的组合

如果用户在定义常数时，不加 Public 或 Private，则根据默认，常数为私有的。过程中的常数都是私有的，一经定义，常数的可见性不可改变。模块级别中的常数缺省为私有的。用户若要声明一个公共模块级别常数，则可以在 Const 语句前加上 Public 这个关键字；也可以利用在 Const 语句前加上 Private 关键字来明确的声明一个私有的常数，使用户可更容易的去读取并且解释代码。然而，在类模块中，常数只能为私有的，且不能用 Public 关键字来改变其可见性。

下面的示例：

```
Public Const conAge As Integer = 34
```

声明了一个 Public 常数 conAge 为 Integer 类型，并且指定它的值为 34。

常数可以声明成下列数据类型：Boolean、Byte、Integer、Long、Currency、Single、Double、Date、String 或 Variant 中的一种。因为已经知道常数的值，所以可以指定 Const 语句中的数据类型。

有时用户想在同一行中声明多个常数，则只需逗号来分隔每个常数赋值，同时为了指

定数据类型，必须将每一个常数的数据类型包含进来。当用这种方式将多个常数声明组合起来时，如果使用 Public 或 Private 关键字，则应将其用到全部常数上。

在下面的语句

```
Public Const conAge As Integer = 34, conWage As Currency = 35 000
```

中，常数 conAge 和 conWage 被声明为 Integer 类型并且同为 Public 常数。

如果在定义时，没有明确使用 As type 来声明常数的类型，则常数会选用最适合 Expression 的类型。用户定义常数时的表达式可以是数值，如

```
Const conPI = 3.141 592 65
```

此时虽然没有定义 conPI 的数据类型，但是 Microsoft Visual Basic 会自动将它定义为数值类型 (Single)。

用户还可以按照上例的方法，将常数定义为 String 类型，如

```
Const conVersion = "Version 7.0"
```

用户还可以对常数使用数值表达式，或者根据前面定义的常量来定义新的常量，如

```
Const conPI2 = conPI * 2
```

在指定给常数的表达式 (expression) 中不得使用变量、用户自定义函数或 Visual Basic 固有的函数 (比如 Sqr)。比如用户需要定义一个常数是 10 的平方根，输入如下代码：

```
Const SquareRootOfTen = Sqr (10)
```

将产生错误。而必须先计算，然后再编写：

```
Const SquareRootOfTen = 3.162 277 66
```

还需要说明的是，Sub、Function 或 Property 过程中所声明的常数，对该过程来说是局部的；而在过程外，模块中所声明的常数在整个这个模块中都被定义了。在任何可使用表达式的地方都可使用常数。

使用常数可使程序形成文档，易于修改，与变量不同的是，在程序运行时，常数可以避免意外的改动。

13.6.2 固有常量

除了用 Const 语句定义常量之外，Microsoft Access 还自动定义了许多固有常量，并且提供了对 Visual Basic for Applications 常量和 Data Access Objects (DAO) 常量的访问。用户还可以在其他的引用对象库中使用常量。固有的常数或系统定义的常数是由应用程序与控件提供的。提供对象库的其他的应用程序如 Microsoft Access、Microsoft Excel 以及 Microsoft Project，也提供常数列表。这些常数可与它们所属的对象、方法以及属性等一起使用。Visual Basic 的常数都列在 Visual Basic (VB) 库、Visual Basic for Applications (VBA) 库以及 Data Access Object (DAO) 程序库之中。需要注意的是 Visual Basic 仍会对之前的 Visual Basic 或 Visual Basic for Applications 所定义的常数做识别。用户可以提升常数到 Object Browser 之中，已经在 Object Browser 中所列的常数，不用在的应用程序中声明。

一、固有常量的名称

在早期版本的 Visual Basic 中，常数的名称都是用大写字母串加上一横下划线而成，例如：

TILE_HORIZONTAL

由于对象库的增多,固有常数越来越容易混淆,即具有相同名称的常数存在于多个对象库中,而这些常数可能具有不同的值。为了避免这一麻烦,一般采用两种方法来限定常数名称,即

- 通过前缀
- 通过库引用

1. 用前缀限定常数

在系统中,所有对象提供的固有的常数都以混合大小写格式显示,即以两个字符为前缀来指出定义这些常数的对象库。例如,由 Visual Basic 以及 Visual Basic for Applications 对象库而来的常数都会以“vb”为开端;由 Data Access 对象库而来的常数都会以“db”为开端;而由 Microsoft Excel 对象库而来的常数都会以“xl”为开端。下面的示例:

```
vbTileHorizontal
dbAppendOnly
xlDialogBorder
grdAlignCenter
```

解释了如何应用前缀字符于自定义控件,其通常是与类型库相关的。

2. 用库引用限定常数

用户也可利用下列的语法:

```
[libname.] [module name.] constname
```

来限定对常数的引用。

限定常数的语法见表 13.7。

表 13.7 限定常数的语法

单 元	描 述
Libname	可选的。它是指定义此常数的类型库。对于多数的自定义控件来说,它也是控件的类名。如果忘掉了控件的类名,可将鼠标的光标移动到工具箱中此控件的位置。该类名会显示在“工具提示”中
Module name	可选的。定义此常数的类型库所包含的模块的名称。确定它,用户可以利用“对象浏览器”来找到模块的名称
Constname	在类型库中定义的常数名称

例如:

```
Thread.LeftJustify
```

二、查看已有的固定常量

前面已经提到过,用户可以用对象浏览器来查看所有可用对象库中的固有常量列表。要查看 Microsoft Access 中可利用的常量(有好几百个这种常量),可以按以下步骤进行。

- (1) 打开数据库中的模块,单击工具条中的代码按钮,进入模块的代码窗口。
- (2) 此时在工具条中有对象浏览器按钮,单击此按钮;或者按键 F2;还可以从工具菜单中选择对象浏览器选项。此时,系统将弹出“对象浏览器”窗口,如图 13.10 所示。
- (3) 要查看所有可以使用的常量,则在对象浏览器窗口中的引用框中选择所有库,此

(5) 在搜索结果列表框中选择其中的一个类，窗口中的右下角会显示这个类中的所有成员，图 13.11 所示即为 Access 库中的 Constants 类中的所有成员。

(6) 选中其中的一个成员，在窗口的最下部会显示该成员的值以及它的来源，图 13.11 中选中的是 acApplyFilter 常量，它的值是 1，位置在 Access.Constants 类中。

在此用户可以看出，固有常量利用前缀来区别不同库中的常量，常量的两个字母前缀指明了定义该常量的对象库。来自 Microsoft Access 库的常量以“ac”打头；来自 DAO 库的常量以“db”打头；而来自 VBA 库的常量则以“vb”打头，例如：

acApplyFilter 是来自 Microsoft Access 中的常量；dbAppendOnly 是来自 DAO 中的常量；而 vbCurrency 则是来自 VBA 中的常量。

因为固有常量所代表的值在 Microsoft Access 的今后版本中可能改变，所以应该使用常量而不是常量的实际值。前面已经看到，用户可以通过在“对象浏览器”中选择常量来显示常量的值。

在“对象浏览器”中还可以查看已打开的数据库中的常量，如图 13.12 所示即为 db2 数据库中的 Form1 中定义的常量 a，它的值为 2。

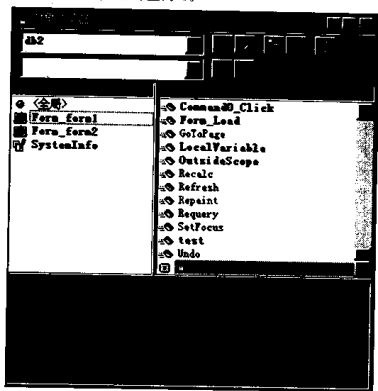


图 13.12

三、使用范围

所有的固有常量都可在宏或 Visual Basic 中使用，并且任何时候这些常量都是可用的。用户可以在任何允许使用符号常量或用户定义常量的地方，包括表达式中，使用固有常量。下面的例子：

```
Dim varNum As Variant
If VarType (varNum) = vbCurrency Then
```

```
Debug.Print " varNum contains Currency data."
```

```
Else
```

```
Debug.Print " varNum doesn't contain Currency data."
```

```
End If
```

说明了如何使用固有常量 vbCurrency 来确定 varNum 变量是否为 Variant 类型（对于该类型 VarType 函数将返回 6（当前值））。

在 Microsoft Access 中主要有以下几种类型的固有常量、操作常量、DAO 常量、事件过程常量、关键字常量、RunCommand 方法常量（有一个操作常量的子集）、安全常量、Visual Basic 常量和 VarType 函数常量。

13.6.3 操作常量

在 Visual Basic 运行 DoCmd 对象的方法时可使用这些操作常量（DoCmd 对象在 Microsoft Access 是十分有用的，在后面将有介绍，这里先介绍它的常量）。在 DoCmd 对象中使用这些常量就跟在宏中运行相应的操作一样。表 13.8 列举了其中的一部分常量。

表 13.8 部分 DoCmd 对象常量说明

DoCmd 对象常量	用 于
acForm	Close、CopyObject、DeleteObject、Rename、RepaintObject、Save、SelectObject、TransferDatabase
acLink	TransferDatabase、TransferSpreadsheet
acMacro	Close、CopyObject、DeleteObject、Rename、RepaintObject、Save、SelectObject、TransferDatabase
acNewRec	GoToRecord
acNext	GoToRecord
acNormal	OpenForm
acPreview	OpenForm
acPrevious	GoToRecord
acQuery	Close、CopyObject、DeleteObject、Rename、RepaintObject、Save、SelectObject、TransferDatabase
acReadOnly	OpenQuery、OpenTable
acReport	Close、CopyObject、DeleteObject、Rename、RepaintObject、Save、SelectObject、TransferDatabase
acSaveNo	Close
acSaveYes	Close
acTable	Close、CopyObject、DeleteObject、Rename、RepaintObject、Save、SelectObject、TransferDatabase
acToolbarNo	ShowToolbar
acToolbarYes	ShowToolbar
acViewDesign	OpenQuery、OpenReport、OpenTable
acViewNormal	OpenQuery、OpenReport、OpenTable
acViewPreview	OpenQuery、OpenReport、OpenTable

1. 事件过程常量

许多事件均有自己的常量，这是为了在用户编程时能十分方便的将这些常量作为事件过程中的参数使用，十分直观。例如在窗体的 AfterDelConfirm 事件过程中可以使用以下常量：

```
acDeleteCancel
acDeleteOK
acDeleteUserCancel
```

在窗体的 ApplyFilter 事件过程中可以使用以下常量：

```
acApplyFilter
acCloseFilterWindow
acShowAllRecords
```

2. 关键字常量

关键字常量用来代替关键字，这些常量可用于 Microsoft Access 的 KeyDown 和 KeyUp 事件过程中。例如，常量 vbKeyF1 代表 F1 键，常量 vbKey1 代表数字 1 键。

3. RunCommand 方法常量

用户在 Visual Basic 中使用 RunCommand 方法时可以用这些常量。在代码窗口中，用户需要用 RunCommand 方法常量来代替在宏中给 RunCommand 操作的提供值。例如，用户在 Visual Basic 中要做保存操作时，就需要用到 RunCommand 方法常量 acCmdSave，同样的操作在宏中就可使用 Save 命令。

4. 安全常量

这些常量是用来确定用户访问数据库的权限的。

5. Visual Basic 常量

它是 Visual Basic for Applications 为了使程序设计变得更为简单而定义的一些常数。这些常数也有很多种，这里就不再叙述，有需要的读者可查看 Microsoft Access 的帮助文件。

6. VarType 常量

VarType 是 Visual Basic 中的一种，它是 Visual Basic 为了便于程序的编写而定义的，用户可在代码中的任何地方用下列常数代替实际值。这些常量在 VarType 函数中特别有用，函数的返回值就需要用这些常量来判断才能得出变量的类型。常量如表 13.9 所示。

表 13.9 VarType 函数中常用的常量

常 数	值	描 述
vbEmpty	0	Empty (未初始化)
vbNull	1	Null (无有效数据)
vbInteger	2	整数
vbLong	3	长整数
vbSingle	4	单精度浮点数
vbDouble	5	双精度浮点数
vbCurrency	6	货币值
vbDate	7	日期
vbString	8	字符串

续表 13.9

常 数	值	描 述
vbObject	9	对象
vbError	10	错误值
vbBoolean	11	布尔值
vbVariant	12	Variant (只与变体中的数组一起使用)
vbDataObject	13	数据访问对象
vbDecimal	14	十进制值
vbByte	17	位值
vbArray	8192	数组

13.6.4 系统定义常量

用户在 Microsoft Access 的任何地方都可以使用系统定义的常量: True、False 和 Null。例如, 可在以下的宏条件表达式中使用 True; 将窗体的 Visible 属性设置为 True, 如:

```
Forms! Employees.Visible = True
```

在 Microsoft Access 的任何地方都可以使用 Null 常量。例如, 可通过下面的表达式:
= Null

用 Null 常量来设置窗体控件的 DefaultValue 属性。

总之, 用户最好用常量代表那些不变的 Number 数据类型或字符串值, 这样可以增加 Visual Basic 代码的可读性, 并且使代码更加容易维护。此外, 使用固有常量还可以保证: 即使常量所代表的值在日后的 Microsoft Access 版本中被改变了也能使代码正常运行。

13.7 数 组

在用户编程时, 也许要用到一组具有相同数据类型的值的变量, 例如十个用户的代号, 此时可以声明一个数组来代表这一组变量。在存储时, 数组可以认为是单一的变量, 但是它存储很多值, 并且可以在使用时将这些值利用唯一的索引号分开; 而常规的变量只能存储一个值。数组是连续可索引的具有相同数据类型的元素所组成的集合, 数组中的每一个元素具有唯一的索引号, 更改其中的一个元素并不会影响其他元素。

数组中的所有元素均具有相同的数据类型。当然, 如果数据类型是 Variant 时, 数组元素的数据类型可以是各种类型, 例如 String、Number、Date/Time 或 Object。用户可以声明各种类型的数组。

13.7.1 固定大小数组的声明

数组的声明方式和其他的变量是一样的, 它可以使用 Dim、Static、Private 或 Public 语句来声明, 这完全取决于该数组的应用范围。

- 要创建 Public 数组, 则在模块的声明部分用 Public 语句来声明。
- 要创建模块级的数组, 则在模块的声明部分用 Dim 或 Private 语句来声明。

- 要创建局部数组，则在模块的过程中用 Dim 或 Static 语句来声明。

一、确定上下界

用户在声明固定大小数组时就已经确定了数组的上界。数组的上界就是声明语句中在数组名后的括号内的数。例如，以下语句是在模块的声明部分中的数组声明语句：

```
Dim intCount (14) As Integer
Dim dblSum (21) As Double
```

这两句语句均声明了一个模块级的数组，前一句声明了有 15 个元素的整数数组，索引号范围为 0 到 14，上界为 14；后一句声明了有 21 个元素的双精度数组，索引号范围为 0 到 20，上界为 20。

如果要创建公共数组，则需要用 Public 代替 Dim，当然该语句仍需要写在模块的声明部分，例如：

```
Public intCount (14) As Integer
Public dblSum (21) As Double
```

如果要创建局部数组，则需要用 Dim 语句或 Static 语句，当然该语句不能写在模块的声明部分，而应该在过程中，例如：

```
Dim intCount (14) As Integer
Dim dblSum (21) As Double
```

这样声明后的数组的元素个数都是上界加 1，也就是说，数组的下界都是 0。一般习惯上用户比较喜欢下界为 1，这样比较形象，为了达到这一点，可以使用 Option Base 语句。

Option Base 语句是在模块级别中用来声明数组下标的默认下界。它的语法如下：

```
Option Base {0 | 1}
```

因为缺省的基为 0，所以 Option Base 语句决不是必需的。如果使用，则在任何过程之前，此语句一定出现在模块中。Option Base 仅能在模块中出现一次，而且一定在包含维数的数组声明之前。

Option Base 语句只影响它所在的模块中的数组的下界。

如果要使数组的下界为 1，则在该模块的声明部分中加入如下语句：

```
Option Base 1
```

如图 13.13 所示即为在模块中加上该语句后的代码窗口。

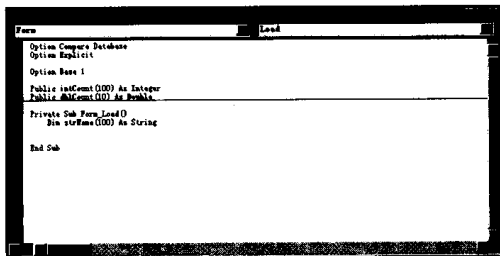


图 13.13

在加入此语句后，前面例子中的元素个数分别为 14（从 1 到 14）和 20（从 1 到 20）。

在 Visual Basic 中，还有另外一种方法来确定数组的上界和下界，这就是使用 To 关键字。例如：

```
Dim intCount (1 To 15) As Integer
Dim dblSum (100 To 120) As Double
```

在这两句声明语句中，数组 intCount 的索引范围为 1 到 15，共 15 个元素；数组 dblSum 的索引范围为 100 到 120，共 21 个元素。

二、多维数组

数组可以是多维的，通常这叫做多维数组，在 Visual Basic 中最多可以声明变量到 60 个维数。例如，下列的语句：

```
Dim sngMulti (9, 9) As Single
```

声明一个 10 乘 10 的两维数组。

如果将数组想成矩阵，则第一个参数代表行，而第二个参数代表列，这个数组的结构就像一个 10 乘 10 的矩阵。

在声明多维数组时，行列均可以使用 To 关键字。例如上例中，将下界改为 1 而不使用 Option Base 语句：

```
Dim sngMulti (1 To 10, 1 To 10) As Single
当然这可以扩展到任意维，例如：
Dim sngMulti (3, 1 To 10, 11 To 20) As Single
```

三、初始化数组

数组在定义完之后，它的值还只是数组的数据类型的缺省值，只有经过初始化之后才有意义。对一维数组的初始化一般使用 For...Next 循环来进行，例如下例中就是将数组的值初始化为 5：

```
Static intCount (1 To 15) As Integer
Dim intX As Integer
For intX = 1 To 15
    intCount (intX) = 5
Next intX
```

用户可以使用嵌套的 For...Next 语句去处理多重维数组。下列过程：

```
Sub FillArrayMulti ()
    Dim intI As Integer, intJ As Integer
    Dim sngMulti (1 To 5, 1 To 10) As Single

    ' 用值填入数组。
    For intI = 1 To 5
        For intJ = 1 To 10
            sngMulti (intI, intJ) = intI * intJ
        Next intJ
    Next intI
End Sub
```

将一个两维数组的所有元素都填入 Single 值。

13.7.2 动态数组

有时, 用户无法事先知道数组的大小, 此时就需要用到动态数组。动态数组, 是在执行代码时去改变数组大小。使用动态数组是 Visual Basic 中较为复杂的一部分, 它可以帮助用户更有效的管理内存。例如, 用户可以申请一个动态数组, 在使用完以后就释放这些内存, 这样可以使有限的内存发挥最大的作用。在无法预知数组大小的情况下, 另一种方法是申请一容量足够大的静态数组, 然后只使用其中的一部分, 这样就浪费了许多宝贵的内存资源, 甚至有时会导致 Microsoft Access 由于内存太低而无法运行。使用动态数组是十分有必要的。

要申请动态数组, 用户可以利用 Static、Dim、Private 或 Public 语句来声明, 方法与固定大小数组的声明一样, 只是在声明语句中数组名后的括号内为空, 如下所示:

```
Dim sngArray () As Single
```

用户在声明完动态数组以后, 还不能使用, 因为这时候系统并未给该数组分配内存, 只有在使用 ReDim 语句确定数组的维数与大小以后, 才能对数组进行初始化。例如:

```
ReDim sngArray (10)
```

假设此时 Option Base 为 1, 那么此时 sngArray 数组具有 10 个元素。

ReDim 语句只能在过程级别中使用, 与 Dim 和 Static 语句不同, ReDim 语句在运行时产生动作。

用户可以重复地使用 ReDim 语句来改变数组中的元素数目和维数。但是, 不能声明一种数据类型的数组, 稍后又用 ReDim 来将数组变成另一种数据类型, 除非数组是 Variant 的数组。如果数组是 Variant 的数组, 则可用 As type 子句改变元素的类型对于过程中的数组范围。每当需要时, 用户可以使用 ReDim 语句去更改动态数组, 然而当做这个动作时, 数组中存在的值会丢失。若要保存数组中原来的值, 则可以使用 ReDim Preserve 语句来扩充数组。例如, 下列的语句:

```
ReDim Preserve varArray (UBound (varArray) + 10)
```

将 varArray 数组扩充了 10 个元素, 而原本数组中的当前值并没有消失掉。

需要注意的是, 在使用了 Preserve 关键字的情况下是不允许改变数据类型的, 即使是 Variant 类型。

如果使用了 Preserve 关键字, 则只能调整最后数组维数的大小, 而不能改变维数的数目。例如, 如果数组只有一维, 则可以调整维数的大小, 因为它是最后且唯一的维数。但是, 如果数组的维数在二维以上, 则只能调整最后维数的大小, 并一直保留数组的内容。下例:

```
ReDim X (10, 10, 10)
```

```
...
```

```
ReDim Preserve X (10, 10, 15)
```

指出可以增加动态数组最后维数的大小, 且不必删除数组中的任何现有数据。

同样, 当已使用了 Preserve, 则只能通过改变上界来改变数组的大小; 如果改变下界, 则会造成错误。

如果使用 ReDim Preserve 重新定义数组大小时, 使数组变得比本来更小, 则被删除的元

素中的数据会丢失。

下例使用 ReDim 语句来配置或重配置动态数组变量的存储空间。示例中假设 Option Base 为 1。

```
Dim MyArray () As Integer      ' 声明动态数组
ReDim MyArray (5)              ' 配置 5 个元素类型为 Variant
For I = 1 To 5
    MyArray (I) = I            ' 设置数组初值。
Next I
```

下列语句将重新调整数组的大小，且将所有元素内容清除。

```
Redim MyArray (10)             ' 将数组重新调整为 10 个元素。
For I = 1 To 10
    MyArray (I) = 1            ' 设置数组初值。
Next I
```

下列语句也重新调整数组的大小，但不清除元素内容。

```
Redim Preserve MyArray (15)    ' 将数组重新调整为 15 个元素
```

第十四章 创建 Visual Basic 程序

前一章详细介绍了 Visual Basic 的语法知识，熟悉了 Visual Basic 中的数据类型，如何定义变量和常量以及数组，本章将介绍如何使用 Visual Basic 进行编程创建过程，进而组成数据库。

14.1 Visual Basic 编程环境

要对窗体或报表进行 Visual Basic 编程，可以在窗体或报表的设计视图下，单击工具条中的代码按钮，Microsoft Access 就会显示出如图 14.1 所示的窗体模块窗口；如果要编写标准模块代码，则在数据库窗口中单击模块页，屏幕上会显示出该数据库中的所有标准模块，如图 14.2 所示。如果要修改某一个模块，则选中该模块，然后单击“打开”按钮，Microsoft Access 就会进入模块窗口；如果要新建一个模块，则单击“新建”按钮，Microsoft Access 就会显示一个新的模块窗口。

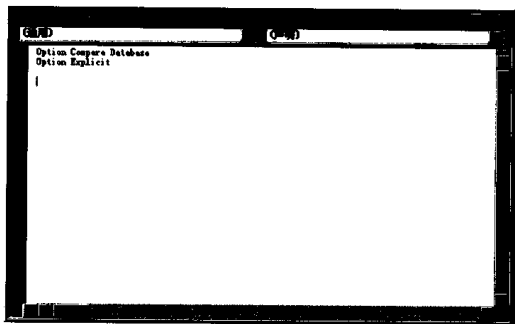


图 14.1

同表、窗体、查询和报表的“设计”视图一样，在模块的设计状态下，Microsoft Access 也会显示适合于编写模块的工具，此时工具条如图 14.3 所示。

● 插入模块按钮——单击该按钮，会显示下拉的菜单，有三个选项：模块，类模块，标准模块，选中其中之一，则 Microsoft Access 会插入一个新的模块。

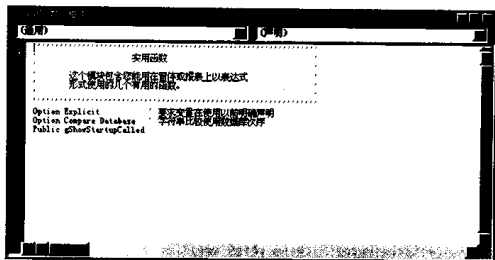


图 14.2

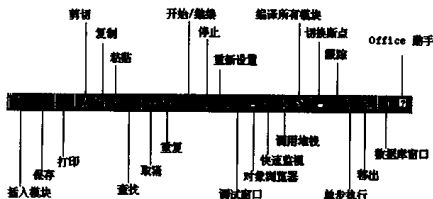


图 14.3

- 保存按钮——单击此按钮，Microsoft Access 会保存当前模块中的代码。
- 打印按钮——单击此按钮，打印当前模块中的代码。
- 剪切、复制、粘贴按钮——功能与其他工具条中的功能一样。
- 查找按钮——单击此按钮，Microsoft Access 会显示如图 14.4 所示的查找对话框，用户可以向对话框中写入要查找的对象和范围，来查找这些对象。
- 取消按钮——单击此按钮，能消除用户对代码所做的修改。
- 重复按钮——单击此按钮，能重复键入对代码所做的修改。
- 开始/继续按钮——单击此按钮，可以开始执行代码，或在遇到一个断点以后，可以继续执行代码直到下一个断点。
- 停止按钮——单击此按钮，可以停止代码的执行。
- 重新设置按钮——在发生错误后，需要重新开始执行代码，则单击此按钮即可。
- 调试窗口按钮——单击此按钮，Microsoft Access 会显示调试窗口。
- 对象浏览器按钮——单击此按钮，Microsoft Access 会显示对象浏览器。
- 快速监视按钮——选中要监视的表达式，再单击该按钮，Microsoft Access 会显示该表达式的快速监视窗口。

- 编译所有模块按钮——单击该按钮，编译当前数据库中所有模块中的所有代码。如果存储了编译的修改程序，Microsoft Access 将在第一次运行这个程序时，编译模块。

- 切换断点按钮——在要设为断点的代码行中单击此按钮，就可以将此行代码设置为断点；如果要取消这个断点，只需重复操作。

- 跟踪按钮——在一个断点上停留后，单击该按钮，则执行一行代码；如果遇到调用其他过程的语句，可以进入被调用过程的第一行代码，以允许用户单步执行被调用过程的每一行代码。

- 单步执行按钮——在一个断点上停留后，单击该按钮，则执行一行代码；如果遇到调用其他过程的语句，则把这句语句作为一个整体执行，而进入被调用的过程中。

- 移出按钮——单击该按钮，Microsoft Access 就运行到当前焦点所在过程的结尾。

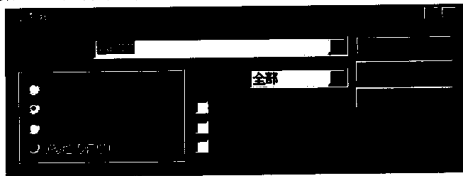


图 14.4

14.2 编写 Visual Basic 语句

Visual Basic 中的语句是一个完整的命令。它可以包含关键字、运算符、变量、常数以及表达式。Visual Basic 语句可分为三类：

- 声明语句
- 赋值语句
- 可执行语句

下面作一下简要介绍。

14.2.1 声明语句

在声明语句中，用户可以给变量、常数或程序取名称，并且指定一个数据类型。前面介绍的变量常数的声明都是在声明语句中实现的。

14.2.2 赋值语句

赋值语句指定一个值或表达式给变量或常数。赋值语句通常会包含一个等号 (=)。最简单的赋值语句就是给变量赋值，例如：

```
Dim varX As Integer  
varX = 5
```

还有如下示例：

```
Sub Question ()
```

```
Dim yourName As String
yourName = InputBox (" What is your name?")
MsgBox " Your name is " & yourName
End Sub
```

将 InputBox 函数的返回值给变量 yourName。

赋值时还可以使用 Let 语句，是可选的，通常为省略。例如，上述的赋值语句可以写成：

```
Let yourName = InputBox (" What is your name?")
```

Set 语句可被用来指定一个对象给已声明成对象的变量，而 Set 关键字是必备的。在下面示例：

```
Sub ApplyFormat ()
Dim myCell As Range
Set myCell = Worksheets (" Sheet1").Range (" A1")
With myCell.Font
    .Bold = True
    .Italic = True
End With
End Sub
```

中，Set 语句指定 Sheet1 上的一个范围给对象变量 myCell。

设置属性值的语句也是一个赋值语句。下面的示例：

```
ActiveCell.Font.Bold = True
```

将设置对于活动单元格 Font 对象的 Bold 属性。

14.2.3 可执行语句

在 Visual Basic 程序中，仅有声明语句和赋值语句是不能完成任何任务的，执行语句是 Visual Basic 程序中的关键部分。一个可执行的语句可以执行初始化动作，还可以执行一个方法或者函数，并且可以循环或从代码块中分支执行。可执行的语句通常包含数学或条件运算符。

下面的示例：

```
Sub ApplyFormat ()
Const limit As Integer = 33
For Each c In Worksheets (" Sheet1").Range (" MyRange").Cells
    If c.Value > limit Then
        With c.Font
            .Bold = True
            .Italic = True
        End With
    End If
Next c
MsgBox " All done!"
End Sub
```

使用 For Each...Next 语句来重复名称为 MyRange 范围内的每个单元格，而此范围在活动的 Microsoft Excel 工作簿的 Sheet1 中。

示例中的 If...Then 语句检查单元格的值。如果它的值超过 33，则 With 语句设置单元格中 Font 对象的 Bold 和 Italic 属性，而 If...Then 语句会以 End If 结束。

With 语句可以节省代码键入次数，因为所包含的语句会自动执行紧接在 With 关键字后面的对象。

Next 语句会去调用单元格集合中的下一个单元格，而此单元格集合是包含于 MyRange 中。

MsgBox 函数（它显示一个固有的 Visual Basic 对话框）显示一个信息用来指出 Sub 过程已经完成执行。

在编写代码时，通常情况下是将一个语句写在同一行中；但有时由于代码太长，一行难以写下，此时可以利用一个续行符“-”将语句连续到下一行中。下面的示例中，可执行语句 MsgBox 被接续的写在三行中：

```
Sub DemoBox ()
    ' 该过程声明一个字符串变量，指定它的值为 Claudia，然后显示一个连接的消息
    Dim myVar As String
    myVar = " John"
    MsgBox Prompt: = " Hello " & myVar, -
        Title: = " Greeting Box", -
        Buttons: = vbExclamation
End Sub
```

为了增强程序的可读性，在程序中加入注释是最好的方法。注释可以为读代码的人解释过程或是特别的命令。Visual Basic 在运行过程时，会忽略掉注释。

有两种方法来说明一个注释语句。现在最常用的方法是使用一个单引号（'），它不是在波浪符号（~）下的撇号，而是在双引号（"）下的单引号。下面是一个例子：

```
Private Sub Command1_Click ()
    ' 用户单击按钮时将发生如下动作
    '
End Sub

' 用户还可以使用老式的 Rem 关键字：
Private Sub Command1_Click ()
    Rem 用户单击按钮时将发生如下动作
    Rem
End Sub
```

通常，程序以行缩排来提高可读性。注释可以加在过程的任何地方。如果用户想在程序的行尾加上注释，使用单引号是比较简便的；而 Rem 形式需要在前面加上一个冒号，如：

```
If intX = 5 Then
    MsgBox " 变量值为 5"      ' 如果变量值为 5，则显示信息并结束
    Exit
Else
    intX = 2 * intY - 1      : Rem 如果变量值不是 5，继续
```

End If

按照缺省规定，注释会以绿色文本显示，如图 14.5 所示，图中注明了各种语句及其他部分。

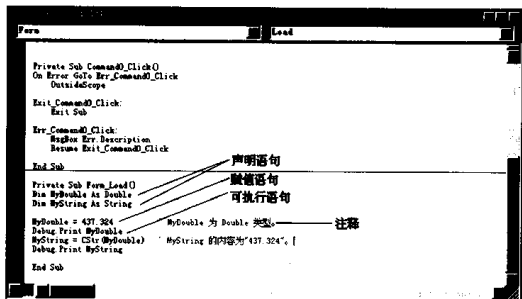


图 14.5

14.3 创建事件过程

前面已经介绍过，Visual Basic 的过程是由一系列语句组成的具有一定功能的，可以执行一个操作或者计算的单元。事件过程就是能对事件作出反应的过程，下面就以创建一个命令按钮的 Click 事件为例，说明如何创建事件过程。

图 14.6 显示的是创建以后的窗体，它由一个命令按钮和一个文本框构成，当用户单击命令按钮时，在文本框中就会显示出“Hello, World!”，如图 14.7 所示。

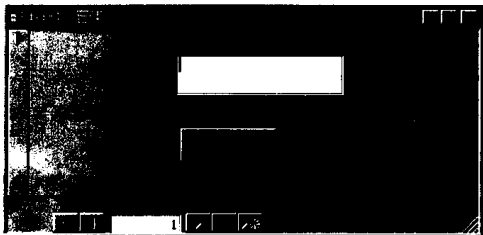


图 14.6

由于事件过程是窗体或报表中的一部分，因此首先必须创建好窗体中的控件，在这个例

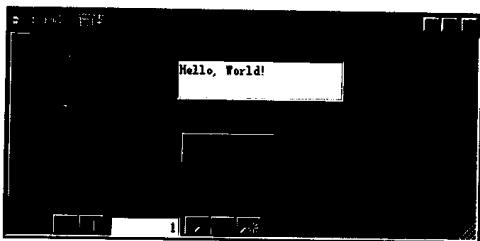


图 14.7

子当中，需要向窗体中加入命令按钮和文本框。在窗体的代码中，引用这些控件对象是通过对象名称来实现的。虽然在将控件拖入窗体后，Visual Basic 会缺省的给控件设置一个名称，在这个例子的窗体中，文本框的缺省名是 Text0，命令按钮的缺省名是 Command1，但这样的名称对于一个好的程序员而言，肯定是不可取的。为了使控件名更具有代表性，用户在对控件进行命名时，控件名称最好要具有一定的意义。在本例中，将文本框命名为 Message，将命令按钮命名为 OK。要对控件进行命名，只要在该控件的属性窗口中设置它的属性就可以了。图 14.8 所示就是将文本框的名称属性设置为 Message。

在创建好事件过程所在的窗体和控件，设置好控件和窗体的属性以后，用户就可以在模块窗口开始编写第一个事件过程。

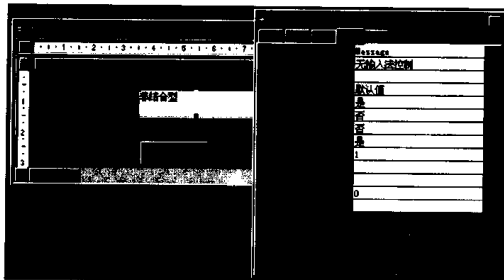


图 14.8

以编写 OK 命令按钮的事件过程为例，详细说明编写的过程。

首先在窗体的“设计”视图中选中所要编写代码的对象，例如窗体、报表、控件或其他部分，在此例中就是 OK 命令按钮，然后单击鼠标右键，Microsoft Access 就会显示如图

14.9 所示的快捷菜单。在此菜单中选中“处理事件”命令，此时 Microsoft Access 会显示如图 14.10 所示的“选择生成器”对话框。在所提供的选项中选择代码生成器，然后单击“确定”按钮，Microsoft Access 会打开模块窗口，并且为用户所选的对象的缺省事件过程创建样板。对于此例中的命令按钮而言就是 Click 事件，图 14.11 所示就是此时的模块窗口。

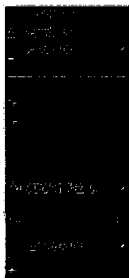


图 14.9

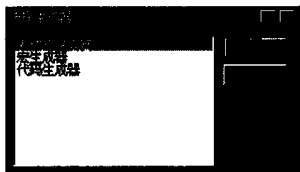


图 14.10

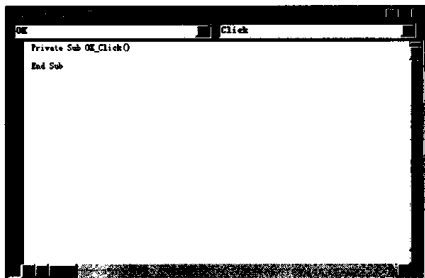


图 14.11

由图 14.11 中可以看到，Microsoft Access 已经在模块窗口中自动输入了事件过程的框架，即 Private Sub OK_Click [] 和 End Sub，用户可以在这两个语句之间输入所需要的语句，来完成预先设计好的动作。在此例中，为了能在单击 OK 命令按钮以后，在 Message 文本框中显示“Hello, World!”，就需要在 Private Sub 和 Sub 之间输入如下语句：

Message = " Hello, World!"

最后要保存刚才的模块。当用户保存模块时, Microsoft Access 会自动将命令按钮的单击事件属性设置为 [事件过程], 如图 14.12 所示。到此, 事件过程就编写完成了, 你就可以运行它了。

在工具条中选中“窗体”视图按钮, 将窗体切换到“窗体”视图状态, 然后单击 OK 按钮, 此时 Microsoft Access 就运行刚才创建的事件过程, 在文本框中显示“Hello, World!”。

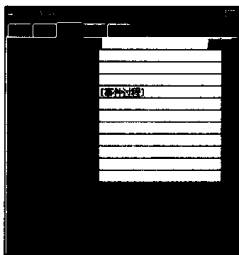


图 14.12

14.4 窗体模块和报表模块

在“Hello, World!”的例子中, 是在对象快捷菜单中使用“处理事件”命令来创建一个新的事件过程。在 Microsoft Access 中, 除了这种方法以外, 还有其他的方法来打开窗体模块和报表模块, 创建或修改其中的事件过程。

如果用户想在打开窗体模块和报表模块的同时打开它的模块窗口, 那么用户可以在数据库库窗口中选中相应的窗体和模块, 然后单击工具条中的代码按钮。当然, 在窗体和报表处于“设计”视图下时, 用户也可以使用代码按钮来打开它们的模块窗口。

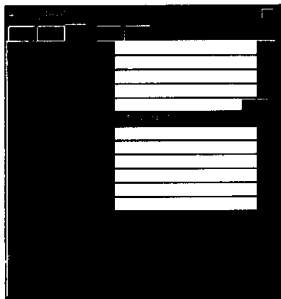


图 14.13

在上例中, 所要创建的事件过程就是该控件的缺省事件, 因此在打开模块窗口时, 显示的就是这个事件。实际上, 用户可以从属性表中直接创建事件过程。

首先以“设计”视图打开窗体或报表, 然后在相应的对象上单击鼠标右键, 此时 Microsoft Access 会显示前面图 14.9 所示的快捷菜单; 在菜单中选中属性, 属性窗口就会显示在屏幕上; 在属性窗口中单击事件按钮, 此时属性表中显示的是该对象的所有事件; 在该属性表中选中所要创建的事件, 单击下拉箭头, 选择 [事件过程], 如图 14.13 所示。这样该事件的属性就被设置为事件过程, 最后单击右端的创建按钮。

如果这个事件属性已经有事件过程与之相联系, Microsoft Access 就会打开模块窗口并显示这个事件过程; 如果这个事件属性原来是空的, Microsoft Access 会显示创建生成器对话框, 在所提供的选项中选择代码生成器, 然后单击确定按钮。Microsoft Access 会打开模块窗口, 并且为用户所选的过程创建样板,

与“Hello, World!”例子中一样。

14.4.1 编写和修改代码

在模块窗口中编写和修改代码就像在文本编辑器中编辑文本一样，同样可以进行删除、插入、复制和粘贴等操作。

在 Microsoft Access 97 中，模块具有很多新的特征，其中有一些特征在编写代码时十分有用，能提高效率。

(1) 在 Microsoft Access 97 的模块窗口中编写 Visual Basic 代码时，当输完某些对象或变量，Microsoft Access 会显示一系列下拉选项——这些选项都是这个变量或者对象可以使用的方法或属性，用户就可以直接在这些选项中选择相应的项，这样就可以使用户在编写代码时变得十分方便。例如，用户在模块窗口的某一过程中输入了一个对象变量，在句号的后面想输入该对象变量的方法或属性，Microsoft Access 自动显示它的方法和属性列表。图 14.14 所示，就是在输入一对象变量以后的方法和属性列表。

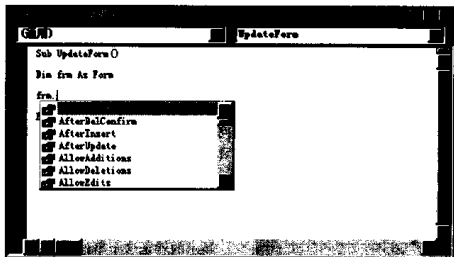


图 14.14

当然用户也可以自己输入代码，并且在输入代码时，列表中的选项也自动跟踪，显示最近已输入代码的那一项，按 Tab 键就选中这一项；用户也可以按 Enter 键来选中这项，但模块中的代码也换行，如果这一行代码未结束，就还需要返回上一行。如果要隐藏列表，则按 Esc 键。

要使自动创建语句发生作用，用户可以在工具菜单中的选项中单击模块按钮，选中“自动列表成员名单”检查框，如图 14.15 所示。“自动列表成员名单”的功能并不只是在编写代码时有用，有时用户也许还需要显式地使用几个功能，特别是在编辑修改以前写的代码时。要显示已经在模块中存在的对象的“属性/方法列表”，用户可以用鼠标右键单击这个对象的方法或属性，Microsoft Access 就会显示如图 14.16 所示的快捷菜单。在菜单中选中自动列表成员名单，在该对象的方法或对象的位置就会显示出“属性/方法列表”。

(2) 快速信息是 Microsoft Access 97 提供的又一个有用的工具。当用户在输入过程或方法名称，并在名称后跟了空格或括号时，Microsoft Access 会在该行的下面自动显示一个提

示信息，给出了这个过程或方法的语法信息，例如需要输入的参数等。

要使快速信息发生作用，用户可以在工具菜单中的选项中单击模块按钮，选中自动显示快速信息检查框，如图 14.15 所示。对于那些已经存在的变量、常量和过程，也可以通过用鼠标右键单击它们，并用显示的快捷菜单中的快速信息或参数信息来显示这些提示信息。如图 14.17 所示就是显示了快速信息的模块窗口。

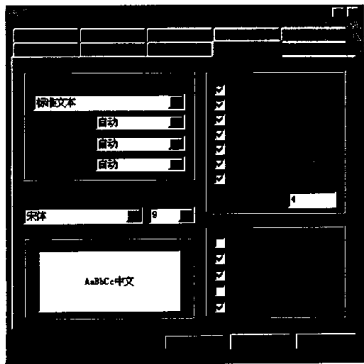


图 14.15



图 14.16

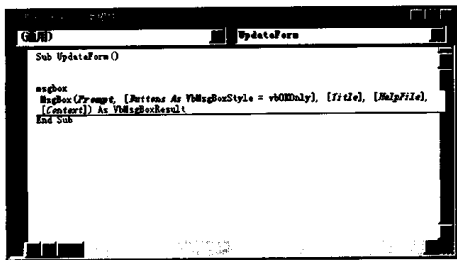


图 14.17

(3) 自动语法检查能在光标离开某行代码时，对该行中的代码进行语法检查，如果发现错误，Microsoft Access 会显示错误信息。要使自动语法检查发生作用，用户可以在工具菜单中的选项中单击“模块”按钮，选中“自动语法检查”检查框，如图 14.15 所示。

(4) 在模块窗口中, 如果要某处的代码移到另一个地方, 可以使用 Microsoft Access 97 提供的拖放正在编辑的文本功能, 而不必使用复制和粘贴命令。有了这个功能, 用户只要在模块窗口中选中所要移动的代码, 然后拖到新的地方就可以了。要使它发生作用, 用户可以在工具菜单中的“选项”中单击“模块”按钮, 选中拖放正在编辑的文本检查框, 如图 14.15 所示。

14.4.2 在过程之间切换

在修改模块中的代码时, 用户可以通过 Ctrl + Page Down 和 Ctrl + Page Up 键在过程之间切换; 除了这种方法以外, 还可以通过在对象框和过程框中选择相应的对象和过程来进行切换, 而且还可以直接创建几个新的过程。

一、对象框

对象框中显示的是所有可以选取的对象, 如窗体、窗体中的控件以及通用过程等。单击窗体右侧的箭头, Microsoft Access 会下拉显示所有与该窗体或报表相连的对象。在下拉的对象列表中选择对应的对象, 此时, 在过程框中就会自动显示该对象的所有过程, 直接显示的就是该对象的缺省过程。

二、过程框

如果用户在对象框中选的是通用, 当用户还没有输入自定义的 Sub 或 Function 过程时, 那么在过程框中显示的是声明, 此时用户可在代码窗口中进行模块级的公共变量和常量的定义, 以及用户自定义数据类型的定义。如果用户还有自定义的过程, 那么在过程框中会显示这些过程。要查看这些自定义过程, 可以单击过程框右侧的下拉箭头, 在显示的下拉过程列表中选择相应的过程。单击, Microsoft Access 会在模块窗口中显示该过程的代码。如果用户在对象框中选择的是其他的对象, 那么在过程框中显示的是当前过程名, 或是该对象的缺省事件。单击过程框右侧的下拉箭头, 会显示这个对象的所有事件。那些已经在模块窗口中存在的事件, 在下拉事件列表中是以黑体显示的。图 14.18 所示就是 OK 命令按钮的所有事件, 其中 Click 是以黑体显示的。

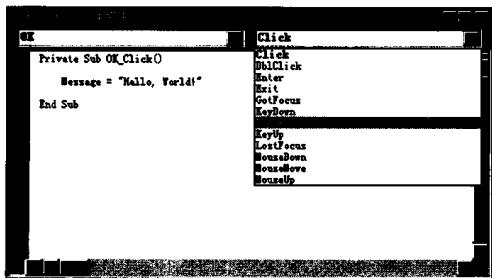
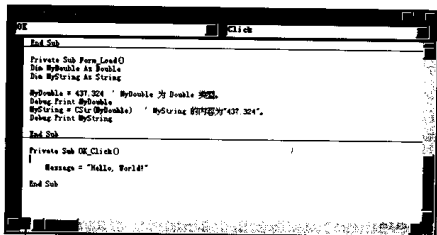


图 14.18

在模块窗口中可以在同一屏中显示所有的过程，也就是说，过程的代码是连续显示的；也可以以分割开的方式显示，即在模块窗口中一次只能显示一个事件过程。要在这两种方式之间进行切换，用户可以在工具菜单中的选项中单击“模块”按钮，从选中或消除整个模块窗口检查框来实现，最简单的方法是单击在模块窗口的左下角中过程视图和整个模块视图按钮来实现，如图 14.19 所示。



单击此按钮，可以将过程连续显示在模块窗口中

图 14.19

三、书签

如果用户要编写一个大型的应用，在调试程序或编写程序时，就需要在不同的模块和过程之间进行切换。由于过程太多，用户很容易忘记原来的位置，为了解决这个问题，Microsoft Access 97 提供了书签这一工具。用户可以在模块中的某个位置设置书签，Microsoft Access 会找寻书签来帮助用户返回原来的位置。

在模块窗口中设置书签十分简单。如果用户要在当前行设置书签，只要在编辑菜单中选择书签命令，然后单击于菜单中的书签命令，此时 Microsoft Access 就会在当前行的左侧边缘中显示一个兰色的方块来指示在该行中设置了书签。图 14.20 所示就是设置了书签后的模

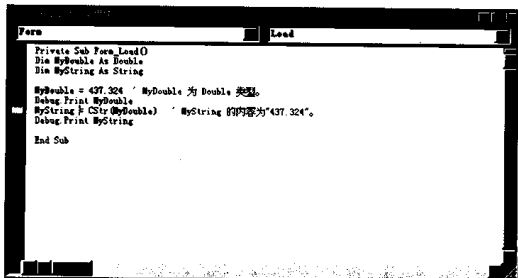


图 14.20

块窗口。

如果用户要回到模块中的书签位置，可以在编辑菜单中选择书签命令，然后单击子菜单中的下一个书签或前一个书签命令。如果要清除所有的书签，则单击子菜单中的清除所有书签命令。

14.5 创建过程

前面已经介绍了如何编写 Microsoft Access 中的事件过程。事件过程是 Visual Basic 编程的核心，但是它们不能过于复杂。如果一个事件过程的代码长于一页，或者甚至长于一屏，它可能就太长了，难以调试，此时应当考虑在一个或多个 Visual Basic 的通过程中作一些工作。前面介绍过，在 Visual Basic 中有两类通用过程，第一类是 Function 过程，用来生成新的函数，以扩展 Visual Basic 中的内置函数；另一类是 Sub 过程，是用来帮助将大任务分成小任务，或者使重复的操作自动进行。

14.5.1 Function 过程

如果用户在窗体、报表或查询中，需要重复使用某一个表达式，那么就可以使用一个函数过程来代替这个表达式。例如，假设经常要计算下一个月的第一天的时间，这经常使用在记账的时间上，就可以使用如下表达式：

```
= DateSerial (Year (Now), Month (Now) + 1, 1)
```

来进行计算。但是，经常输入这个表达式是一件十分麻烦的事，而且十分容易输错，为了解决这个问题，可以使用函数过程来执行这个计算。使用函数过程代替表达式来执行计算有以下几个优点。

- 可以保证计算过程每次都一样执行，而不用担心会在什么地方输错。
- 如果用户需要修改计算的过程，那么只要在函数过程中修改一次就可以了，而不用在任何用到该计算的地方均修改。
- 在函数过程中，能执行复杂的操作，例如 If...Then 语句，而这些在单一的表达式中是不可能完成的。
- 在调试时十分方便。
- 在函数过程中可以用注释语句来解释函数的功能。

下面将以 FirstOfNextMonth 函数为例，介绍创建函数的详细过程。这个函数的作用是计算下一个月的第一天的日期，在订单数据库设定 BillingDate 文本框时有用。

如果用户要在其他窗体或报表模块中使用这个函数，最好将此函数放在标准模块中。建立新的标准模块的方法与建立新的窗体或报表的方法完全一样。在数据库的数据库窗口中单击“模块”按钮，此时在窗口中显示数据库中所有的标准模块；要建立新的标准模块，则单击窗口中的“新建”按钮，此时 Microsoft Access 会显示一个新的模块窗口。由于在工具菜单中的选项中，设置了选项，在新的模块窗口中 Microsoft Access 会自动显示两个 Option 语句，如图 14.21 所示。这是建立函数过程的第一步。

然后用户就该定义函数过程了。定义函数过程需要用到 Function 语句，下面先介绍 Function 语句。

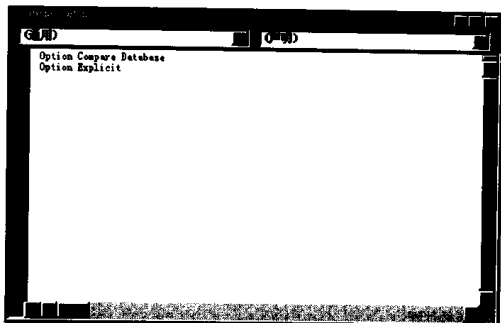


图 14.21

一、函数过程的语法

Function 语句的语法如下所示：

```
[Public | Private] [Static] Function name [ (arglist) ] [As type]
    [statements]
```

End Function

Function 语句的语法见表 14.1。

表 14.1 Function 语句的语法

部 分	描 述
Public	可选的参数。表示任何模块中的所有其它过程都可访问 Function 过程。如果在包含有 Option Private 的模块中使用这个参数，则不能在工程之外使用此过程
Private	可选的参数。表示 Function 过程只能在声明它的模块中的其他过程中使用
Static	可选的参数。表示在两次调用之间将保留 Function 过程的局部变量。Static 不会影响 Function 之外声明的变量，即使在过程中用到它们
name	必需的参数。Function 的名称，遵循标准变量命名约定
arglist	可选的参数。变量列表，这些变量表示在调用用逗号隔开的多个变量时传递到 Function 过程的参数
Type	可选的参数。Function 过程返回值的数据类型，可以是 Byte、Boolean、Integer、Long、Currency、Single、Double、Date、String（除了固定长度），或 Object、Variant 或任何用户定义类型。不能返回任何类型的数组，但可以返回包含数组的 Variant
Statements	可选的参数。在 Function 过程执行的任何一组语句
Expression	可选的参数。Function 的返回值

由上表可知，函数过程是可以有返回值的，并且可以带参数，而且参数序列 arglist 中的每个参数都要遵循如下的语法：

[Optional] [ByVal | ByRef] [ParamArray] varname [()] [As type] [= defaultvalue]

ByVal 表示按值的方式传递参数；而 ByRef 表示按地址的方式传递参数，在 Visual Basic 中默认为 ByRef。

如果未显式地用 Public、Private 或过程来指定，则按照默认，Function 是公用的。

如果没有使用 Static，则局部变量的值在每两次调用之间不会保留下来。

在 Microsoft Access 中，Public Function 过程可用于当前数据库以及引用 Microsoft Access 数据库的其他所有过程中，但不能用于其他任何应用程序。

如果在任何模块中，通过在其前面加上 Private 关键字，而将 Function 过程声明为私有的，那么该程序只能用于同一模块的其他过程中。

如果 Function 过程在私有模块中被声明为公用过程，例如类模块中，那么该过程可用于该数据库内的其他所有过程；但不能用于其他 Microsoft Access 数据库。

Exit Function 语句会使 Function 过程立即停止，程序会从调用 Function 过程的语句的下一行语句处继续执行。在 Function 过程的任何位置可出现任意多个 Exit Function 语句。

二、创建函数过程

由于在标准模块中没有像在窗体模块中编写事件过程那样，Microsoft Access 已经准备好了过程的框架，在编写函数过程时要自己创建框架。在 Option Explicit 语句的下一行，或者标准模块中的任意一个空行中输入 Function，然后空格，再输入函数过程的名称，在本例中就是 FirstOfNextMonth，再按 Enter 键。此时，Microsoft Access 会自动检查输入的代码，如果没有错误，Microsoft Access 就会再显示一遍，并且加入一个空行和 End Function 语句，在过程框中显示该函数过程的名称 FirstOfNextMonth。如果模块窗口被定义为过程视图，那么该过程的框架就如图 14.22 所示。

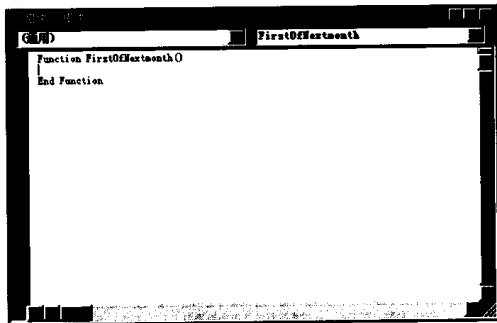


图 14.22

也许读者已经注意到了，在函数名的后面 Microsoft Access 又自动添加了一对圆括号。在圆括号内用户可以根据需要增加函数的参数，由于例子中的函数没有参数，因此在圆括号

内不用输入任何代码。

下面就可以在函数的框架内输入代码了。在函数中执行计算与在 Microsoft Access 的其他地方执行计算没有什么区别，也是利用表达式，所不同的只是在函数过程中为了能将结果返回，必须将该表达式的值赋值给函数名称。这种赋值可出现在过程中的任何位置，且不限次数。如果没有赋值给函数名称，则过程会返回默认值：数值函数返回 0，字符串函数返回零长度字符串（""），Variant 函数返回 Empty；如果没有在 Function 中（使用 Set）把对象引用赋值给函数名称，则返回对象引用的函数将返回 Nothing。在本例中，只要将如下代码

```
FirstOfNextMonth = DateSerial (Year (Now), Month (Now) + 1, 1)
```

输入 Function 和 End Function 之间。此时模块窗口如图 14.23 所示。

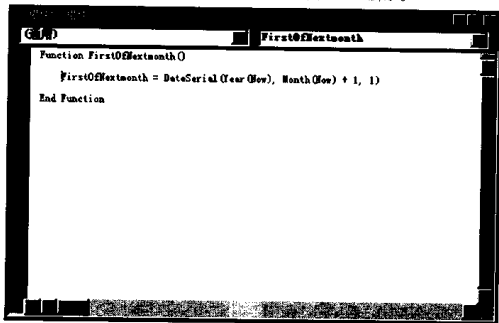


图 14.23

在编写完代码以后，用户就该进行对代码的编译和调试工作。有关编译和调试的内容在下一章中再介绍。

三、使用函数过程

在调试完代码后，函数就可以运行和使用了。用以代替表达式的函数几乎可以在 Microsoft Access 中的任何地方使用，大体上可以分为：

- 在其他 Visual Basic 过程中使用；
- 在窗体、报表和查询中的某些字段中替代表达式进行计算；
- 在查询或宏中作为条件使用。

在本例中，函数过程是用作记账日期文本框显示数据的来源，只要将该文本框的控件来源属性设置为该函数名即可。

首先在“设计”视图下打开订单窗体，选中记账日期文本框，在它的属性列表中将控件来源属性设置为

```
= FirstOfNextMonth ()
```

如图 14.24 所示。此时，在订单窗体中输入一个新的订单时，记账日期文本框会自动显示下

一个月的第一天的日期。

四、带有参数的函数过程

前面已经讲到，函数过程可以带有一个或多个参数。这些参数是在函数过程中作为变量来使用的，实际上大部分函数都带有参数，本例中的函数也可以变成带有参数。

例如，前面编写的 FirstOfNextMonth 函数返回的只是以计算机的系统时间为参照的下个月的第一天的日期。当然这对于新的订单是没有问题的，而对于那些上个月的订单，在记账日期文本框中显示的应该是这个月第一天的日期，而现在显示的仍是下个月的第一天的日期，因此必须修改这个函数过程。要使这个函数返回的是以订购日期为基准的下个月的第一天的日期，就需要在函数中加入参数。

此时，FirstOfNextMonth 函数的代码如下：

```
Function FirstOfNextMonth ( dtmAny As Date) As Date
    FirstOfNextMonth = DateSerial (Year (dtmAny), Month (dtmAny) + 1, 1)
End Function
```

如图 14.25 所示，在订单窗体中记账日期文本框的控件来源属性设置为
= FirstOfNextMonth ([Order Date])

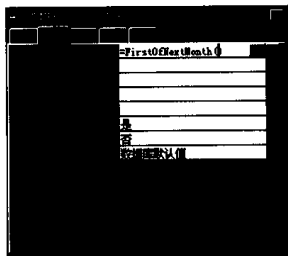


图 14.24

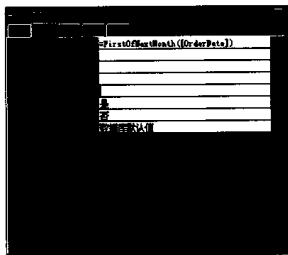


图 14.25

14.5.2 Sub 过程

虽然函数能够完成任何任务，但是由于函数在任何情况下，都是提取数据进行处理，然后返回一个值；但有时并不一定要返回一个值，而有时又要求返回两个甚至更多的值，此时就需要用到 Sub 过程。

要建立一个新的 Sub 过程，与函数过程一样，如果要在标准模块中建立 Sub 过程，则要先打开标准模块窗口。在数据库的数据库窗口中单击“模块”按钮，此时在窗口中显示数据库中所有的标准模块。要建立新的标准模块，则单击窗口中的“新建”按钮，此时 Microsoft Access 会显示一个新的模块窗口。由于在工具菜单的选项中，设置了选项，在新的模块窗口中 Microsoft Access 会自动显示两个 Option 语句。

创建 Sub 过程，需要使用 Sub 语句，下面先介绍 Sub 语句的语法。

一、Sub 语句的语法

Sub 语句的语法如下所示：

```
[Private | Public] [Static] [ByVal] Sub name [ (arglist)] [As type]  
    [statements]
```

End Sub

可以将过程类型定义为 Byte、Integer、Long、Single、Double、Currency、String、Boolean、Date、Object 或 Variant，如果用户在定义时没有定义类型，Microsoft Access 就缺省定义为 Variant。

Sub 语句的语法中的详细内容与 Function 语句的基本相同，这里不再重复。

如果没有使用 Public、Private 显式指定，则默认 Sub 过程是公用的。如果没有使用 Static，则在调用之间不保留局部变量的值。

在 Microsoft Access 中，标准模块中的公用 Sub 程序可用于当前数据库及引用 Microsoft Access 数据库中的所有其他过程；然而，不能用于其他的应用程序。

如果在任何模块内声明 Sub 过程为私有的，那么该过程只能用于同一模块中的其他过程。

如果 Sub 过程在私有模块中声明为 Public，例如类模块中，那么该程序可用于当前数据库内的其他所有过程；但并不能用于其他的 Microsoft Access 数据库。

Exit Sub 语句会使 Sub 过程立即停止。程序会从调用该 Sub 过程的语句的下一行语句处继续运行。在 Sub 过程的任何位置可以出现任意多个 Exit Sub 语句。

二、创建 Sub 过程

同 Function 过程的创建过程一样，在用户输入 Sub、空格和 Sub 过程名以后，Microsoft Access 会自动为 Sub 过程建立框架。例如，要创建一个名为 MultiBeep 的 Sub 过程，则在输入 Sub MultiBeep 后，Microsoft Access 会显示与创建函数时差不多的模块窗口，在过程框中显示的是该过程的名称 MultiBeep，并且在框架中加入了圆括号。

然后，用户就可以开始输入代码了。为了能够改变蜂鸣的次数，就需要在过程中增加参数，下面就是该过程的代码：

```
Sub MultiBeep (intBeeps As Integer)  
    Dim intX As Integer, lngY As Long  
    For intX = 1 To intBeeps  
        Beep  
        For lngY = 1 To 10000  
            Next lngY  
        Next intX  
    End Sub
```

这样就完成了 Sub 过程的创建。

三、调用 Sub 以及 Function 过程

用户可以使用 Call 关键字来调用一个过程，它的语法结构如下：

```
[Call] name [argumentlist]
```

但是，调用一个过程时，并不一定要使用 Call 关键字。如果使用 Call 关键字来调用一个需要参数的过程，argumentlist 就必须加上括号；如果省略了 Call 关键字，那么也必须省略 argumentlist 外面的括号。如果使用 Call 语法来调用内建函数或用户定义函数，则函

数的返回值将被丢弃。

可以使用 Sub 过程去组织其他的过程，因此可以较容易了解并调试它们。在下面的示例

```
Sub Main ()
    MultiBeep 56
    Message
End Sub

Sub MultiBeep (numbeeps As Integer)
    For counter = 1 To numbeeps
        Beep
    Next counter
End Sub

Sub Message ()
    MsgBox " Time to take a break!"
End Sub
```

中，Sub 过程 Main 传递参数值 56 去调用 Sub 过程 MultiBeep。运行 MultiBeep 后，控件返回 Main，然后 Main 调用 Sub 过程 Message。Message 显示一个信息盒。当按“确定”键时，控件会返回 Main，接着 Main 退出执行。

下面的示例：

```
Sub Main ()
    HouseCalc 99800, 43100
    Call HouseCalc (380950, 49500)
End Sub

Sub HouseCalc (price As Single, wage As Single)
    If 2.5 * wage <= 0.8 * price Then
        MsgBox " You cannot afford this house."
    Else
        MsgBox " This house is affordable."
    End If
End Sub
```

展示了调用具有多个参数的 Sub 过程的两种不同方法。当第二次调用 HouseCalc 时，因为使用 Call 语句所以需要利用括号将参数括起来。

在调用 Function 过程时，为了使用函数的返回值，必须指定函数给变量，并且用括号将参数封闭起来，如下例所示：

```
Answer3 = MsgBox (" Are you happy with your salary?", 4, " Question 3")
```

如果不在意函数的返回值，可以用调用 Sub 过程的方式来调用函数。如下面示例

```
MsgBox " Task Completed!", 0, " Task Box"
```

所示，可以省略括号，列出参数并且不要将函数指定给变量。

如果在上述例子中若包含括号，则语句会导致一个语法错误。

Sub 或 Function 过程中的语句可以利用命名参数来传递值给被调用的过程，可以将命名参数以任何顺序输出。命名参数的组成是由参数名称紧接着冒号以及等号 (: =)，然后指定一个值给参数。

下面的示例

```
MsgBox Title: = " Task Box", Prompt: = " Task Completed!"
```

使用命名参数来调用不具返回值的 MsgBox 函数。

下面的示例

```
answer3 = MsgBox (Title: = " Question 3", -
```

```
Prompt: = " Are you happy with your salary?", Buttons: = 4)
```

使用命名参数调用 MsgBox 函数，将返回值指定给变量 answer3。

第十五章

调试 Visual Basic 程序代码

一旦程序有了一定的复杂程度，不管如何仔细的编排程序，也不管如何小心的计划，它都可能不按程序员的意愿工作；因此程序员不得不对程序不厌其烦的进行修改。在修改程序之前，必须找到程序中出错的代码。对程序进行修改的过程就是调试代码，Visual Basic 提供了专门的工具进行调试。

15.1 避免错误

在 Visual Basic 中，程序中的错误会在下列三种情况下表现出来。

1. 编译错误

这种错误是由于在程序中使用语句错误引起的。例如，用户在编写代码时将关键字输错了，或遗漏了必须的标点；还有就是本来应该成对的语句，如 If 和 End If、For 和 Next 而没有成对匹配地编入。Visual Basic 会在编译时检查出这类错误。

2. 运行错误

这类错误的发生并不是由于语法上的错误引起的，而是在程序的运行过程中发生的，主要是由于进行了一些非法的操作。最为明显的例子就是在除法中分母为零。例如：

`Speed = Miles / Hours`

如果变量 Hours 为零，则这个操作就是非法的，即使它在语法上并没有错误。这种错误只有在运行时才可能发现，而且并不是所有的运行错误都是那么容易发现的。

3. 程序逻辑错误

逻辑错误是最难以处理的一种错误。它是由于在代码中有逻辑上的错误，但程序在运行时并没有非法的操作，只是它的运行结果是错误的。Visual Basic 不能发现这种错误，只有用户对这个错误的结果进行分析以后，才能发现。

要想成功的作好调试工作，用户就必须知道程序的下一步是要作什么工作，这样就必须有良好的编程风格，一般需要遵循以下几条。

(1) 对于那些具有独立作用的代码，要放在 Sub 或 Function 过程中，这样可以使程序简洁，而且功能明确。

(2) 在编写代码时要加注释。加了注释的代码，在时间长了以后，代码的功能才能一目了然。

(3) 在每个模块中都加入 Option Explicit 语句。加入该语句后，当用户在使用未定义变量时，Visual Basic 就会产生编译错误。事实上，编译错误中很大一部分是由于错写了变量名称引起的。

(4) 在给变量取名称时，应该使用统一的命名制度。如果变量名称都是有意义的，能使

用户在读代码时一目了然。例如，用户可以在变量名前加上 `pv` 来表示公共变量，加上 `str` 前缀表示该变量是 `String` 型的。

(5) 在声明对象变量或其他变量时，应该尽量使用确定的对象类型或数据类型，而尽量不要使用 `Object` 和 `Variant`。这样既可以加快代码的运行速度，而且当用户错误的将其他类型的变量赋给它时，`Visual Basic` 又会自动提示错误。

15.2 调试工具及其功能

用户对程序进行调试的主要任务是从程序中找到错误。也许用户在编写代码时使用了错误的操作符号，或者是使用函数的方法不对，有时还会忘记初始化变量。这些错误并没有明显的标志。`Microsoft Access` 提供了专门的调试工具对这些错误进行调试，这些工具对调试程序中的逻辑错误尤其有用。

当系统的焦点在代码窗口时，`Microsoft Access` 将会显示最适合于调试的菜单和工具条。在此时，`Microsoft Access` 的菜单中会有调试菜单，单击此菜单，用户就可以看到调试所需的工具，如图 15.1 所示；当然这些测试和调试所需的大部分工具也可以在工具条上找到，如图 15.2 所示。调试工具通常在程序临时挂起时使用。表 15.1 列出了一些可以使用的调试工具。

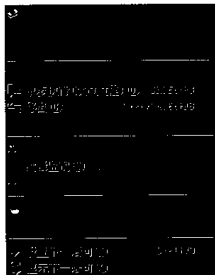


图 15.1

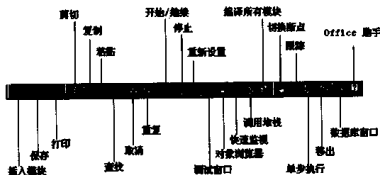


图 15.2

表 15.1 调试工具

工 具	键盘等效	功 能
前往/继续	Alt + C (F5)	启动新程序或继续运行
结尾	Alt + E	强制结束程序运行
重新设置	Alt + R	重新运行程序
跟踪	F8	单步执行程序 (一次一条语句)

续表 15.1

工 具	键盘等效	功 能
单步执行	Shift + F8	类似单步执行工具, 但将过程和函数调用作为一步处理
运行到光标	Ctrl + F8	运行程序到目前光标位置
移出	Ctrl + Shift + F8	运行完当前过程
添加监视	Alt + A	向调试窗口中增加监视的表达式
编辑监视	Alt + E	编辑调试窗口中的表达式
快速监视	Shift + F9	直接在窗口中查看表达式的值
切换断点	Alt + T (F9)	增加/取消断点
清除所有断点	Ctrl + Shift + F9	将程序中的所有断点清除

15.3 启动 Visual Basic 程序代码

在执行 Visual Basic 程序代码时, Microsoft Access 会自动的将程序中的所有模块进行编译, 如果有任何编译错误, 系统就会显示提示信息。用户也可以主动对程序进行编译。在调试菜单中选中“编译所有模块”或“编译所装模块”以及“编译并保存所有模块”命令, 系统就会对程序中对应的模块进行编译。有关这部分内容可见第二章。

在通过编译以后, 用户就可以调试程序中的运行错误和逻辑错误了。这两种错误必须在程序运行过程中才能发现, 因此必须首先启动 Visual Basic 程序代码。启动 Visual Basic 代码, 只要单击工具条中的前往/继续按钮即可。

在调试过程中有些运行错误是十分明显的, 用户十分容易发现, 而且修改简单。例如, 用户本来想在打开窗体时, 使订单数据库中订单窗体中的 Details 命令按钮失效, 这只需将控件的 Enabled 属性值设为 False 即可; 但用户在编写代码时却将控件没有的属性 Locked 属性设置为 True, 代码如下:

```
Forms! 订单! Details.Locked = True
```

在程序运行到这句语句时, 就会产生一个运行错误, 如图 15.3 所示。

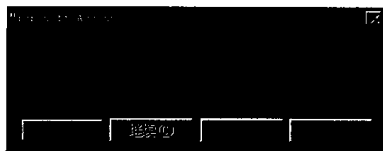


图 15.3

修改这个错误十分简单, 只要将代码改为

```
Forms! 订单! Details.Enabled = False
```

程序又可以正常运行了。此时要继续执行代码, 单击工具条中的前往/继续按钮即可。

15.4 重新设置代码

最为普遍的修改是对变量声明的修改，需要重新设置代码。此时用户想要继续执行代码，Microsoft Access 会提示信息，进行选择：或者是接受修改并重新设置代码，或者是继续执行代码但刚才的修改无效。

要重新设置代码，用户可单击工具条中的“重新设置”按钮。当用户重新设置代码后，所有变量均丢失了刚才的值；Variant 类型变量的值被置为 Empty，数值类型变量的值为 0，对象变量被设置为 Nothing。

15.5 暂停 Visual Basic 程序代码的执行

要使用大部分 Microsoft Access 提供的调试工具，用户必须使程序处于挂起状态，也就是暂停 Visual Basic for Applications 程序代码，此时，程序代码仍然在执行，只是在运行语句之间暂停，变量和对象的属性保持它们的值，在模块窗口中显示当前运行的代码。图 15.3 所示即为程序处于挂起状态。

在用户对程序进行调试时，用户希望当运行到被怀疑有错的代码行时，将程序挂起，以便对程序的运行情况进行检查，如检查当前的变量值；还可以单步执行程序代码，观察代码的运行。要将程序挂起，用户可以用以下几种方法。

当 Visual Basic 在运行时遇到断点，系统就会在运行到该断点处将程序挂起。用户可以在任何可执行语句和赋值语句处设置断点；但不能在声明语句和注释行处设置断点，否则 Microsoft Access 就会显示错误信息说明在该行处不能设置断点，如图 15.4 所示。

如果需要监视一段代码的执行情况，用户可以在这段代码前面设置断点。例如，代码中有一段代码是用来设置变量值的，用户想查看这段代码执行时各变量的值的变化，只需在这段代码的第一行处设置断点。



图 15.4

用户不能在程序运行时设置断点，只有在编写

程序代码或程序处于挂起状态时才可以设置断点。要设置断点，用户可以采用以下两种方法：一种是在模块窗口中，将光标移到要设置断点的行，然后按 F9 键，或单击工具栏上的“切换断点”按钮，也可以在调试菜单中选中切换断点；另一种是在模块窗口中，用鼠标单击要设置为断点的代码行的左侧边缘。在设置完断点以后，Microsoft Access 会在代码行的左侧边缘显示断点指示——红色圆圈，并且该行的代码是以红色为背景色的白色文本显示，如图 15.5 所示。

如果要清除断点，可以将插入点移到设置断点的程序代码行，然后在工具栏上单击“切换断点”，或在断点代码行的左侧边缘单击，此时已经清除了该断点，代码恢复原来的颜色。如果要继续执行程序代码，可以单击“执行”菜单上的“前往/继续”命令。

Visual Basic 会在当前挂起的语句的左侧边缘显示一个黄色箭头，并且该语句位于背景色为黄色的矩形中，如图 15.6 所示。黄色箭头与黄色矩形指明该语句是当前语句，或者说

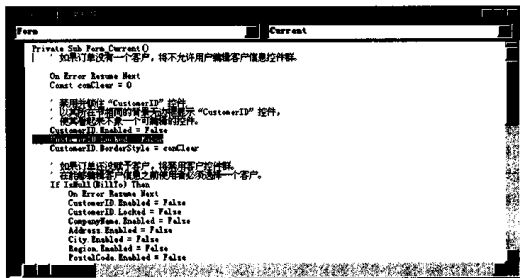


图 15.5

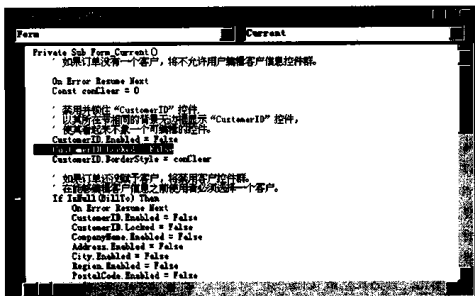


图 15.6

是下一句要执行的语句。如果当前语句也是一个断点, 则断点与当前语句的指示符号均会在左侧边缘显示。图 15.6 中的第九行就是当前语句。

在 Microsoft Access 97 中, 用户要查看程序中变量、对象属性的值, 最简单的方法就是使用调试窗口。当用户的焦点在模块窗口中时, 在视图菜单中选中调试窗口, 或在工具条中单击“调试窗口”按钮, 此时 Microsoft Access 就会显示调试窗口。关于调试窗口的使用将在稍后再做详细介绍。

向过程中添加 Stop 语句, 或在程序代码执行时按 Ctrl + Break 键, 可以挂起程序代码。

Stop 语句是加在程序中的, 当程序执行到该语句时, 程序就被挂起。实际上它的作用与断点的作用一样, 只是它更明显, 而且当用户将数据库关闭后, 所有的断点都自动消除了, 但 Stop 语句却还在代码中。当用户不再需要断点时, 可以使用调试菜单中的“清除所

有断点”命令将所有的断点清除，但 Stop 语句却必须逐句删除，比较麻烦。

15.6 执行 Visual Basic 程序代码

如果用户准确的知道哪句语句产生了运行错误，那么使用单一的断点就足够了。但是，一般情况下，用户只可能知道在哪一部分可能会产生错误，因此只能用单一断点到达该区域，然后用单步执行来检查代码的运行是否符合要求。

逐步执行 Visual Basic for Applications 程序代码，可以帮助识别发生错误的位置，并且可以查看是否每一行程序代码都产生了预期的结果。

在前面介绍的调试工具中，读者已经看到 Microsoft Access 一共提供了四种运行的调试工具，下面分别介绍。

15.6.1 跟踪

如果要单步执行每一行程序代码，包括被调用的过程中的程序代码，可单击工具栏上的“跟踪”按钮或按 F8 键。在执行了跟踪命令后，Visual Basic 运行当前语句，并且自动移到下一个语句，同时将程序挂起。有时在一行程序中写了多个语句，它们用冒号隔开，在用跟踪命令进行运行时，Visual Basic 将逐句执行这一行中的每一个语句，而断点只是应用于这一行中的第一个语句。

如果用户不再需要跟踪执行程序了，可以按 F5 键来执行余下的程序，直到下一个断点。

15.6.2 单步执行（跳过程程）

如果要单步执行每一行程序代码，并且将任何被调用的过程作为一个单位执行，可单击工具栏上的“单步执行（跳过程程）”按钮或按 Shift + F8。它与跟踪单步执行的区别就在于当执行代码到调用其他过程时，跟踪是将当前行转移到该过程中，在此过程中一行一行的执行；而单步执行则是将调用其他过程的语句当作统一的一句，将该过程执行完毕，然后进入下一语句。例如，在程序中有一句：

```
GetQueryCriteria strCriteria
```

如果在调试时，用户使用的是跟踪命令，则当系统执行到该语句时，模块窗口就会显示 GetQueryCriteria 过程，并且该过程的第一句语句就是当前语句。如果用户需要检查被调用过程，则在调试时使用跟踪命令是最合适的。

如果在调试时，用户使用的是单步执行命令，则当系统执行到该语句时，模块窗口还是显示当前模块，而 Visual Basic 在执行 GetQueryCriteria 过程。在执行完该过程后，调用过程的最后一句就是当前语句。如果用户不需要调试被调用过程的代码，即在调试时，可以直接跳过被调用的过程，则使用单步执行（跳过程程）命令是最合适的。

15.6.3 移出

如果要执行当前过程中的剩余代码，然后返回调用树中前一个过程的下一行程序代码，可单击工具栏上的“移出”按钮。当移出命令执行时，Visual Basic 就会将该过程中未执行的语句全部执行完，包括在过程中调用的其他过程，并且也是一步完成的。当执行完这个过

程，程序返回到调用该过程的过程后，移出命令执行完毕。如果这个过程没有被其他过程调用，那么移出命令的作用和继续命令的作用是相同的，都是将所有的代码运行完。

15.6.4 运行到光标



图 15.7

如果要执行当前行之前的程序代码，然后中断，以便单步执行以后的每一行程序代码，可单击“调试”菜单上的“运行到光标”。这个命令当用户可以确定在某一范围内的语句是绝对正确的，而对后面的语句的正确性难以保证时，十分有用。例如，用户对程序中的一个大循环有怀疑，此时就可以在这个大循环语句之前，将光标置于该语句中，然后在“调试”菜单中选中“运行到光标”命令，或单击鼠标右键，此时 Visual Basic 会显示如图 15.7 所示的快捷菜单，在菜单中选中“运行到光标”命令。

需要注意的是，该命令只有当程序处于挂起状态时，才有用。

15.6.5 设置下一语句

有时用户知道程序中在某一部分中有错误，在运行时想跳过这部分程序继续执行，当然这不解决问题，到最后你还得来调试这部分代码，但是，跳过这部分代码，用户就可以继续调试下面的代码。或者有时用户想改变前面已经定义的变量值，或更改前面的代码，这就需要重新执行前面的代码。在 Visual Basic 中具有这种功能，用户可以自由设置下一句要执行的语句。具体方法是：在程序中选定要执行的下一语句，单击鼠标右键，此时 Visual Basic 会显示如图 15.7 所示的快捷菜单，在菜单中选中“设置下一语句”命令。同“运行到光标”命令一样，“设置下一语句”命令也必须在程序处于挂起时才能使用。

15.7 查看变量值

在 Visual Basic 中，用户查看变量值的方法有很多种，这也是 Microsoft Access 97 中新增加的功能。

15.7.1 在模块窗口中查看数据

当程序处于挂起状态，用户在模块窗口中查看代码时，经常想同时查看程序中变量和常量的值，在 Microsoft Access 97 中提供了这样的功能。用户只要将鼠标指针放在用户所想查看的变量或常量的上方，过一会儿，Microsoft Access 就会在该变量或常量名称的下方直接显示它的值，如图 15.8 所示。

在模块窗口中查看数据值，是最简单的方法。但是，有时用户需要查看几个变量或一个表达式的值，或者需要查看对象以及对象的属性，这时就不能在模块窗口中查看了。此时用户可以使用“调试窗口”来查看数据值。

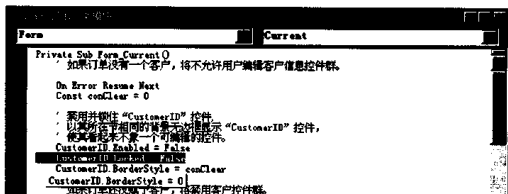


图 15.8

15.7.2 “调试窗口”的新功能

Microsoft Access 97 的“调试窗口”由三个窗格组成, 分别是: 功能很像 Access 旧版本中的 Immediate 窗口的“即时调试”(“立即窗口”窗格、显示当前堆栈框架中全部变量和它们的值的“局部”窗格和“监视”窗格。单击“调试”窗口中的“局部”标签或“监视”标签, 用户可以在“局部”窗格和“监视”窗格之间进行切换。而无论单击哪一个标签, “即时调试”窗格都保持可见, 如图 15.9 所示。

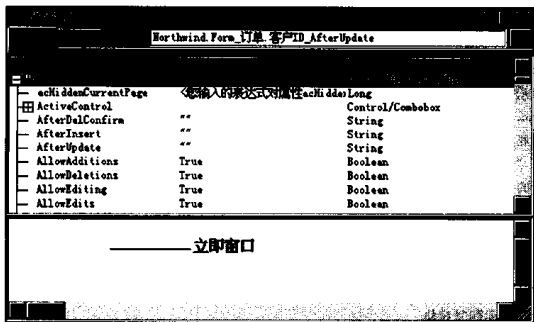


图 15.9

要查看“调试窗口”, 可以打开一个模块, 然后单击工具栏上的“调试窗口”按钮; 也可以通过单击“视图”菜单上的“调试窗口”命令或在任何时候按 Ctrl + G, 来打开“调试窗口”。

1. “局部”窗格

“局部”窗格有三列显示变量的列表: 表达式、值和类型。有些变量, 如用户自定义类型、数组和对象等, 可以包含级别信息。对于这些变量在其名称左边有“展开/折叠”按钮,

该按钮可以控制级别信息的显示。当“局部”窗格可见时，它会在执行方式更改为中断方式（比如执行到断点或单步执行程序）时，自动更新其中的信息。

列表中的第一个变量是一个特殊的模块变量。对于类模块，它是系统定义变量 Me。Me 是对当前模块定义的当前类的实例的引用。因为它是对象引用，所以能展开以显示当前类实例的全部属性和数据成员。对于标准模块，它是当前模块的名称，并且也能展开以显示当前模块中所有的模块级变量。在“局部”窗格中可以更改变量的值，但是不能更改其表达式或类型设置。

2. “监视”窗格

“监视”窗格可以在程序代码执行时，查看表达式或变量的值。要设置监视表达式，可以单击“调试”菜单中的“添加监视”命令。增强的“监视”窗格提供下列新功能：

- 展开/折叠级别信息；
- 调整列标题大小；
- 就地编辑值。

3. “即时调试”窗格

在“即时调试”窗格中可以执行任何 Sub 或 Function 过程，包括事件过程。要从“即时调试”窗格执行定义在某个类模块中的过程，除非当前正处于该模块的中断方式（这时由于该模块属于当前范围，因此不必加模块名称），否则必须在过程名前加上该类模块的名称。

在“调试窗口”的“即时调试”窗格中键入下列语句：

```
Form_Employees.ListNames
```

将执行“雇员”窗体的类模块中定义的 ListNames 过程。

而示例：

```
Form_Employees.PersonalInfo_Click
```

执行“雇员”窗体中的 PersonalInfo 命令按钮的 Click 事件过程。

4. “调试窗口”的其他功能

新的“调试窗口”自动提供关于程序代码的状态信息。当没有程序代码执行时，“调试窗口”上方的状态栏显示“<准备>”，一旦程序代码开始执行，状态栏将显示当前数据库的名称以及正在执行的过程及其所属模块的名称。

用户可以通过单击“生成器”按钮，可以从“调试窗口”中查看“调用堆栈”窗口。下面分别介绍“调试窗口”中的各个部分在调试程序时的作用。

15.7.3 使用即时窗格

当用户在调试程序时，有时需要检查 Function 过程的运行结果是否正确，或者检查对象的属性设置、字段的值和控件设置是否有误。这些工作可以在“调试窗口”中使用“即时调试”窗格来完成。使用它，可以检查 Visual Basic for Applications 程序代码行的执行结果。“即时调试”窗格是一种剪贴板窗口，在其中可以立即计算语句、方法和子程序的值。“即时调试”窗格主要用于以下几种情况：

- 对 Function 或 Sub 过程进行测试；
- 查看控件和属性值；
- 计算并显示表达式的值；

● 显示代码中的调试信息。

一、测试过程

在“调试窗口”的“即时调试”窗格中，用户可以执行 Visual Basic 表达式或语句，例如调用 Sub 或 Function 过程。要对过程进行测试，有两种方法，一种是在“即时调试”窗格中显示从代码中返回的值；另一种是在“即时调试”窗格中直接运行过程，包括对过程的不同参数的运行，这与在模块窗口中运行过程没有两样。

如果用户想在“即时调试”窗格中显示代码中的运行结果，可以使用 Debug 对象的 Print 方法。Debug 对象是“即时调试”窗格的缺省对象，要在 Visual Basic 中使用“即时调试”窗格，就只能使用 Debug 对象。

例如，假设用户编写了一个名为 DueDate 的函数：

```
Public Function DueDate (ByVal AnyDate As Date) As Variant
```

```
    DueDate = DateSerial (Year (AnyDate), Month (AnyDate) + 1, 1)
```

```
End Function
```

它是用来返回传递给它的参数时间的下一个月的第一天。

如果用户要检测这个函数是否能正确运行，就可以在“即时调试”窗格中输入以下代码：

```
Debug.Print DueDate (#4-12-96#)
```

Visual Basic 就会运行这个函数，并显示所得结果：5/1/96。

由于 Debug 对象是“即时调试”窗格的缺省对象，因此在“即时调试”窗格中书写代码时，可以省略 Debug，即上面的代码与如下代码：

```
Print DueDate (#4-12-96#)
```

等效。为了方便起见，在“即时调试”窗格中，用户可以用“?”来代替 Print 方法。例如，用户要确定由 DueDate 返回的日期是星期几，就可以在“即时调试”窗格中输入以下代码：

```
? Weekday (DueDate (#4-12-96#))
```

Visual Basic 就会运行并显示结果：4，这就表示 1996 年 5 月 1 日是星期三。

同样在“即时调试”窗格中也可以运行 Sub 过程。在运行时，需要输入 Sub 过程的参数。例如，用户要测试一个名为 SizeIt 的过程，可以输入如下代码：

```
SizeIt 5000, 3000
```

读者也许已经发现，在代码中没有 Print 方法，这是因为 Sub 过程不返回值，因此在“即时调试”窗格中也就没有值可以显示。

在“即时调试”窗格中，用户可以运行所有 Microsoft Visual Basic 中的所有内置函数，但是控制流程语句必须写在同一行中。在下例

```
For int1 = 1 To 20 : Print 2 * int1 : Next int1
```

中由于 For 循环只有一行，因此它可以在“即时调试”窗格中运行。

需要注意的是，在“即时调试”窗格中输入代码时，如果所要测试的过程必须在当前过程中可见，也就是说，在“即时调试”窗格中，用户可以直接查看的值包括本过程中的变量、常量、Function 过程、Sub 过程以及其他过程模块中的公共过程和公共变量；如果要运行其他模块的私有过程，必须引用全名。

二、显示控制和属性

在“即时调试”窗格中，用户可以计算任何合法的表达式，包括表达式中带有 Microsoft Access 对象或 Data Access Objects (DAO) 的值和属性。例如：

? Forms! SalesReps.RecordSource

Employees

在上例中，第一行中是要计算的表达式，第二行是显示在“即时调试”窗格中的计算结果——数据库中 SalesReps 窗体的 RecordsSource 属性值。

三、用代码在“即时调试”窗格中显示值

如果用户在测试代码时，想在程序运行过程中显示代码中的表达式的值，可以使用前面介绍的 Debug 对象的 Print 方法在“即时调试”窗格中显示结果，如前面的 DueDate 函数；如果要显示函数中当前时间的年和月份，则可以输入如下代码：

```
Public Function DueDate (ByVal AnyDate As Date) As Variant
    Debug.Print "Year"; Year (AnyDate); " Month"; Month (AnyDate)
    DueDate = DateSerial (Year (AnyDate), Month (AnyDate) + 1, 1)
End Function
```

这样，无论在何时调用该函数，均会在“即时调试”窗格中显示当前年和月。例如，用户在“即时调试”窗格中调用该函数时，参数 AnyDate 为 4-12-96，在单步运行到 Debug.Print 行时，就会在“即时调试”窗格中显示年(1996)，月(4)，如图 15.10 所示。

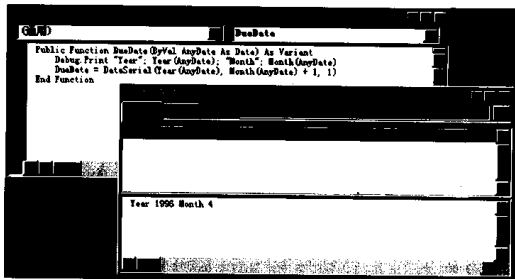


图 15.10

15.7.4 在“调试窗口”中监视数据

用户在调试程序时，有一些错误不是简单的由一个错误引起的，而且有一些错误是在程序中变量值为某一值时才会发生，在这种情况下，用户就需要在程序运行时监视程序中的这些变量。

当程序被挂起时，用户就可以使用“调试窗口”中的上半部分——无论是“局部”窗格还是“监视”窗格。“局部”窗格会自动显示当前过程中的所有变量和对象，“监视”窗格则显示用户指定程序中的表达式。

假设在 DueDate 函数中设置了一个断点，在调用这个程序运行到断点时，程序被挂起，此时用户就可以在“局部”窗格中检查函数的运行情况。图 15.11 所示就是检查函数时的情况。

在图中可以看到，调试窗口的上部有一标题栏指示了当前的模块和当前的过程。通过查

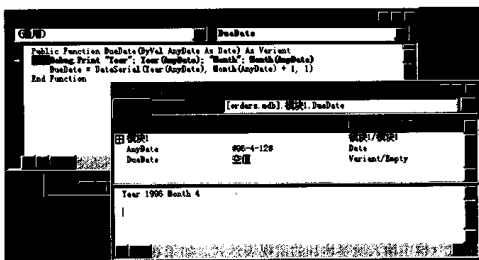


图 15.11

看此标题栏，用户可以清晰的看到当前的运行模块和过程，即使是在不同的模块之间切换。

15.7.5 在“调试窗口”中查看对象

在“调试窗口”中可以显示对象，例如，窗体和控件。当过程运行时，该过程所属的窗体或报表以及它所包含的对象变量就可以在“局部”窗格中显示出来。在“调试窗口”中显示对象，如果该对象还包含子对象，则在该对象名的左侧会有一个加号；如果要查看它的子对象，只要单击该加号即可。例如，当前运行的是窗体中的某个命令按钮的 Click 事件，此时的“调试窗口”如图 15.12 所示，要查看窗体的详细对象资料，单击其中的加号，此时会在“局部”窗格中显示窗体中的所有控件和属性，并且加号会变成减号；如果要还原原来的状态，只要单击减号即可。

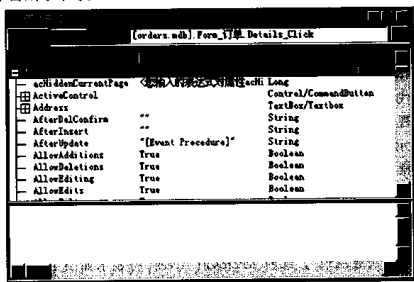


图 15.12

15.7.6 使用监视表达式监视数据

对于监视变量和对象而言,使用“局部”窗格是最佳的选择;如果用户还需要监视某一特定表达式的值,则可以使用“监视”窗格。在“监视”窗格中显示所有用户为监视那些特定的表达式而建立的监视表达式,当“调试窗口”是活动的时候,如果监视表达式有意义时,“监视”窗格中会显示该表达式的值。

当程序开始执行之前或者当程序处于挂起状态时,用户可以向“监视”窗格中增加监视表达式。用户只需单击“调试”菜单上的“添加监视”命令,Visual Basic 会显示如图 15.13 所示的“添加监视”对话框。

由图可以看出,该对话框由三部分组成:“表达式”、“上下文”和“监视类型”。

在“表达式”框中输入所要增加的监视表达式。如果在模块窗口中已经选择了表达式,它将自动地显示在对话框中。如果没有表达式显示,那么可键入要计算的表达式。表达式可以是变量、属性、函数调用或任何其他有效的表达式。除了键入表达式以外,也可以在模块窗口中选择表达式,并将其拖动到“监视”窗格中。

在“上下文”框中用户可以确定该表达式的范围,主要是两部分,即模块和过程。通过确定这两部分,就可以决定表达式的范围。需要注意的是尽量选择适合需要的最小范围,因为选择全部的程序或模块将减慢程序代码的执行速度。

在对话框中,用户还可以定义系统如何反应监视表达式,可在“监视类型”中选择下列选项:

- 如果要显示监视表达式的值,单击“监视表达式”;
- 如果要在表达式计算为 True 时挂起,单击“值为真时中断”;
- 如果要在更改表达式的值时挂起,单击“值更改时中断”。

由此可知 Visual Basic 提供了程序执行时在某种情况下程序被挂起的功能。它提供了当表达式计算为 True 时挂起程序和表达式的值更改时挂起程序。这样用户在调试程序时,不仅仅可以使用跟踪命令,还可以使用 Visual Basic 提供的这一功能。例如,在循环中,希望程序在循环计数器为某一值时将程序挂起,就可以在“监视”窗格中增加一表达式,然后在“添加监视”对话框中选择“值为真时中断”。

当执行程序代码时,在“监视”窗格将显示所设置的表达式的值。如果要修改监视表达式,可以在“监视”窗格中选择要修改的表达式,然后单击“调试”菜单上的“编辑监视”,Visual Basic 会显示如图 15.14 所示的对话框。该对话框与“添加监视”对话框几乎一样,只是在“编辑监视”对话框中增加了“删除”按钮,如果用户要删除监视表达式,可在“编辑监视”对话框中选择要删除的表达式,然后单击“删除”。

在调试 Visual Basic 程序代码时,用户可以使用“快速监视”来查看那些未被定义为监视表达式的表达式。要使用“快速监视”,必须先在模块窗口中选择要查看值的表达式,然后在工具栏上单击“快速监视”,此时 Microsoft Access 将显示如图 15.15 所示的“快速监视”对话框,在其中可以查看表达式和表达式的当前值。在对话框中单击“添加”,即可向“调试”窗口“监视”窗格的监视表达式列表中添加该表达式。如果 Visual Basic 不能计算所选择的表达式,则“添加”按钮失效。

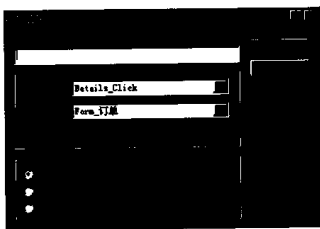


图 15.13

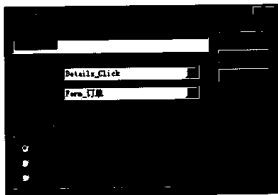


图 15.14



图 15.15

15.7.7 设置控件、属性、变量

“调试窗口”中的“局部”窗格和“监视”窗格,不仅可以用来检测变量和表达式的值,而且还可以用来给控件、属性、变量设置新值。这对于调试那些仅仅是变量在某一值下会发生的错误,特别有用。用户可以在程序处于挂起时改变变量的值,然后继续运行程序来检验程序。

要更改变量或属性值,用户可以在“局部”窗格或“监视”窗格中单击所要更改的变量的值,然后键入新的值,按 Enter 键。图 15.16 所示即为在“局部”窗格中更改变量值的情况。

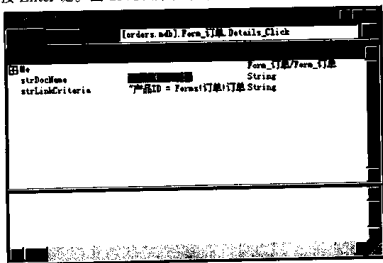


图 15.16

用户还可以在即时窗格中利用代码来更改值,例如,用户可以在即时窗格中键入如下代码:

```
Forms! SalesReps! Title = "Sales Executive"  
Forms! SalesReps! Title.Visible = False  
intRows = 50
```

来更改设置。在更改完以后,用户可以继续运行程序来观看更改所引起的结果。

15.7.8 跟踪 Visual Basic 的过程调用

在调试程序代码期间,当挂起 Visual Basic for Applications 程序代码时,可以使用“调用过程”对话框来查看那些已经开始执行但还没完成的过程列表。用户持续在工具栏上单击“调用堆栈”,Microsoft Access 将在列表的最上方显示最近被调用的过程,接着是次最近调用的过程,以此类推,如图 15.17 所示。如果要显示调用列表中下一个过程的语句,可以单击“显示”。

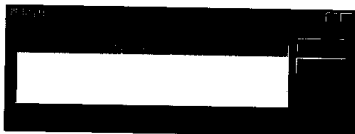


图 15.17

15.8 优化 Visual Basic 程序代码

通常,提高 Visual Basic for Applications 程序代码运行速度的最佳途径是选择一种更有效的方法,但有些技巧也可以帮助提高代码的运行速度。

下面将介绍在 Microsoft Access 97 中提高代码的运行速度的一些技巧。

- 在运行前,最好确保数据库中的代码已经编译。用户只需在模块窗口中单击“调试”菜单中的“编译并保存所有模块”命令即可。

- 使用显式变量声明。通过在“选项”对话框中的“模块”选项卡上设置“需要声明变量”选项,可以在要求过程中使用变量之前显式地声明变量。

- 在声明变量时尽可能使用确定的变量类型。例如,要定义一个代表窗体的变量,应将其定义成 Form 类型而不是 Object 或 Variant 类型。

- 通过变量来引用属性、控件和数据访问对象。如果要多次引用窗体上的属性或控件值,或者引用数据访问对象或其属性值,最好通过创建和引用对象变量的方法而不要使用完整的标识符。

- 在事件过程中用 Me 关键字引用窗体。

- 在适当的时候使用字符串函数。在处理字符串时使用 \$ 函数(例如 Str\$ 函数),而在处理 Variant 或其他数据类型时使用非 \$ 函数。

- 计算时尽可能使用 Integer 或 Long 数据类型。

- 使用动态数组和 Erase 或 ReDim 语句申请内存。
- 尽可能使用常量。
- 在数据库中, 使用书签进行快速定位。在返回某个特定记录时使用书签来代替 FindNext 或其他方法。
- 对索引字段使用 FindRecord 和 FindNext 方法。这些方法在用于索引字段时最为有效。
- 组织好模块。Visual Basic 是在需要的时候加载模块, 也就是说要调用某个模块中的过程时才将该模块加载到内存中。将相关的过程放在同一个模块中可以使 Visual Basic 只在必要的时候加载额外的模块。
- 删除无用代码和变量。在开发和修改应用程序时, 可能会遗留一些不再被调用的过程或不再使用的变量。检查您的代码, 找到并删除这样的过程和变量。

15.9 错误和错误处理

在设计应用程序时, 必须考虑出现错误时的情况。有两个原因会导致应用程序出错。首先, 在运行时某些条件可能会使原本正确的代码产生错误。例如, 如果代码尝试打开一个已被删除的表, 就会出错。其次, 代码可能在逻辑上有错, 以致不能执行产生所需结果。例如, 如果在代码中数值被 0 除, 就会出现错误。如果没有作任何错误处理, 在代码出错时 Visual Basic 将停止代码的运行并显示一个错误信息, 而用户会对此感到迷惑和沮丧。通过把错误处理例程包含在代码中来处理任何可能产生的错误, 将可以防止这类问题的发生。

当添加错误处理代码到过程中时, 应当考虑在出现错误时过程会如何选择执行路径。选择执行错误处理程序路径的第一步是允许错误处理程序将某些形式的 On Error 语句包含在过程中。On Error 将在发生错误事件时指引程序的执行。如果没有 On Error 语句, 在出现错误时 Visual Basic 只是简单地中止程序执行并显示一则错误信息。

如果具有错误处理的过程产生了错误, Visual Basic 不会显示一般的错误信息, 而是选择执行路径到错误处理程序。当执行到一个错误处理程序时, 该程序就被激活。在活动的错误处理程序中, 可以确定出现的错误类型, 并以所选的方式处理它。

Microsoft Access 提供了两类对象: Err 对象和 DAO Error 对象, 它们包含发生的错误信息。

15.9.1 出现错误时选择执行路径

错误处理程序指定发生错误时过程如何响应。例如, 在出现特定的错误时您可能需要中止程序的运行, 或者需要改正导致错误的条件并恢复程序执行。On Error 和 Resume 语句决定了出现错误事件时如何继续执行。

On Error 语句可允许或不允许执行有关的错误处理例程。如果打开了错误处理例程, 当出现错误时将会执行错误处理例程。

On Error 语句有三种形式: On Error GoTo label、On Error GoTo 0 和 On Error Resume Next。On Error GoTo label 语句允许执行有关的错误处理例程, 而该例程就是从上面这个 label 语句出现的地方开始的。您应当在可能出现错误的第一行代码前先打开错误处理例程。

在错误处理程序被激活并有错误出现时，执行就会跳转到由 label 参数指定的代码行上。

由 label 参数指定的代码行是错误处理例程的起始行。例如，以下过程：

```
Function MayCauseAnError ()
    On Error GoTo Error_MayCauseAnError
    ' Include code here that may generate error.
    .
    .
    .
Error MayCauseAnError:
    ' Include code here to handle error.
    .
End Function
```

指定出现错误时，执行跳转到标号为 Error_MayCauseAnError 的代码行上。

On Error GoTo 0 语句使过程中的错误处理无效。该语句并不是把 0 行指定为错误处理代码的起始，即使过程中包含有标号为 0 的代码行。如果代码中包含 On Error GoTo 0 语句，则在过程完全运行时错误处理程序将自动地变得无效。On Error GoTo 0 语句会重置 Err 对象的属性，这与使用 Err 对象的 Clear 方法一样。

On Error Resume Next 语句会忽略导致错误的代码行并跳转执行到错误代码行的下一行。此时程序执行并没有中止。如果要在执行可能导致错误的代码行之后立即检查 Err 对象的属性，并且在过程本身而不是在错误处理程序中处理错误的话，即可使用 On Error Resume Next 语句。

Resume 语句使程序执行从错误处理例程跳转回过程的主体。在错误处理例程中包含 Resume 语句可以从过程某一定点上继续执行程序。然而 Resume 语句并不是必需的；您可以在运行完错误处理例程之后就结束过程。

Resume 语句有三种形式。其中 Resume 或 Resume 0 语句将返回发生错误的代码行。而 Resume Next 语句将返回至错误代码行的下一行。Resume label 语句则会返回至由 label 参数指定的代码行。label 参数必须指定一个行标志或一个行号。

特别是当用户要作出修改时可使用 Resume 或 Resume 0 语句。例如，如果应用程序提示用户选择要打开的表的名称，而用户输入了一个并不存在的表名，则应用程序可再次提示用户并在导致错误的语句上继续程序的执行。

使用 Resume Next 语句可以在错误处理程序中改正错误并且跳过错误代码行重新执行程序。使用 Resume label 语句可在过程中由 label 参数指定的代码行上继续执行程序。例如，您可能要在“退出”例程上恢复程序的运行，如下所述。

15.9.2 退出过程

在过程中包含一个处理错误例程的同时，还应该包括“退出”例程，以便只是在发生错误时才运行错误处理例程。您可以像指定错误处理例程一样用行标识指定“退出”例程。

例如，可以将“退出”例程添加到前面章节的例程中。如果没有出现错误，“退出”例程在程序主体之后执行；而出现错误时，在执行完错误处理例程之后将跳转至“退出”例程。“退出”例程包含了一个 Exit 语句。例如：

```
Function MayCauseAnError ()
```

```
On Error GoTo Error_MayCauseAnError
:
' Include code here that may generate error.

Exit_MayCauseAnError:
Exit Function

Error_MayCauseAnError:
.
' Include code to handle error.

' Resume execution with exit routine to exit function.
Resume Exit_MayCauseAnError
End Function
```

15.9.3 处理复杂过程中的错误

当一个复杂过程出现错误，而该过程没有错误处理程序时，Visual Basic 将根据调用关系向后搜索其他过程中的错误处理程序，而不是简单地中止执行程序，这就使得程序代码有机会在被调用的过程中纠正错误。例如，假设过程 A 调用过程 B，过程 B 调用过程 C。如果过程 C 出现错误而其中没有错误处理程序，Visual Basic 就会先检查 B 再检查过程 A 以找到错误处理程序。如果找到了，程序将跳转到其中执行；否则中止程序执行并显示错误信息。

当在活动的错误处理程序中出现错误时，Visual Basic 也会根据调用关系向后搜索错误处理程序。使用 Err 对象的 Raise 方法可用提升活动的错误处理程序中的错误来强制 Visual Basic 根据调用关系向后搜索。这在处理错误程序中出现没有预料到的错误时是很有用的。如果出现预料之外的错误，而在错误处理程序中又再次产生了该错误，Visual Basic 将根据调用关系跳转执行到另外一个可能处理该错误的程序。

例如，假设有一个过程 C，该过程打开了错误处理程序，但并未纠正出现的错误。当错误处理程序检查了所有预料之中的错误之后，它可能再次产生已有的错误。Visual Basic 此时将根据调用关系跳转执行至过程 B 的错误处理程序中（如果有的话）来纠正错误。如果过程 B 没有错误处理程序，或不能纠正错误并且又再次生成了错误，那么执行将跳转回至过程 A 的错误处理程序（假设该程序存在）。

要换一个角度来阐明这个概念，可以假设有一个复杂的过程，其中包含了一个错误处理程序，用来处理预料中的类型不匹配错误。而在过程 C 的某个代码行上出现了一个预料之外的错误：值被 0 除，并且在过程中包含了一个再次产生原有错误的语句，则执行将根据调用关系跳转到另一个允许的错误处理程序（如果存在该程序的话）。如果已在调用关系中的另一个过程中纠正了“值被 0 除”的错误，那么上述错误就可被纠正。但如果代码不再次产生错误，则会继续执行程序而不纠正“值被 0 除”的错误。这反过来有可能会在复杂程序集中产生其他的错误。

总之，在以下情况下 Visual Basic 会根据调用关系向后搜索可行的错误处理程序。

- 过程中出现错误，而该过程又没有包含打开的错误处理程序。
- 在活动的错误处理程序中出现错误。如果使用 Err 对象的 Raise 方法来提升一个错误，

就可强制 Visual Basic 根据调用关系向后搜索可行的错误处理程序。

15.9.4 获得有关错误的信息

在执行跳转至错误处理例程之后，您的代码必须确定发生了何种错误并处理它。Visual Basic 和 Microsoft Access 提供了几种可用来获取特定错误信息的语言元素。每种语言元素都适用于不同类型的错误。因为错误将发生在应用程序的不同部分，所有必须根据所期望的错误在代码中选择使用不同的语言元素。

可用于错误处理的语言元素有：

- Err 对象；
- Error 对象和 Errors 集合；
- AccessError 方法；
- Error 事件。

15.9.5 Err 对象

Err 对象是由 Visual Basic 提供的。当出现 Visual Basic 错误时，有关错误的信息存储在 Err 对象中。Err 对象每次只维护一个错误的信息。当出现新的错误时，Err 对象将更新为新错误的信息。

要获取某个特定错误的信息，可以使用 Err 对象的属性和方法。Number 属性是 Err 对象的默认属性，它将返回出现错误的标志号。Err 对象的 Description 属性则返回与 Visual Basic 错误有关的描述字符串。Clear 将清除 Err 对象的当前错误信息。Raise 方法可产生一个特定的错误并使 Err 对象的属性包含关于该错误的信息。

下面的例子说明了如何在可能导致类型不匹配错误的过程中使用 Err 对象：

```
Function MayCauseAnError ()  
    ' Declare constant to represent likely error.  
    Const conTypeMismatch As Integer = 13  
    On Error GoTo Error_MayCauseAnError  
        .  
        ' Include code here that may generate error.  
        .  
        .  
Exit_MayCauseAnError:  
    Exit Function  
Error_MayCauseAnError:  
    ' Check Err object properties.  
    If Err = conTypeMismatch Then  
        .  
        ' Include code to handle error.  
        .  
    Else  
        ' Regenerate original error.  
        Dim intErrNum As Integer  
        intErrNum = Err
```

```
Err.Clear
Err.Raise intErrNum
End If
' Resume execution with exit routine to exit function.
Resume Exit_MayCauseAnError
```

End Function

在上面的例子中，Raise 方法用来产生原有的错误。如果出现除了类型不匹配以外的错误，程序执行将根据调用关系跳转回至另外一个错误处理程序（如果存在该程序的话）。

Err 对象提供了所有有关 Visual Basic 错误的所需信息；但是，它并没有给出 Microsoft Access 错误或 Microsoft Jet 数据库引擎错误的完整信息。Microsoft Access 和数据访问对象 (DAO) 提供了附加的语言元素来处理这些错误。

15.9.6 Error 对象和 Errors 集合

Error 对象和 Errors 集合是由 DAO 提供的。Error 对象代表了一个 DAO 错误。单独的 DAO 的存在可能导致多个错误，特别是执行 ODBC 操作时。发生在数据存取操作过程中的每个错误都有一个相关联的 Error 对象。与 DAO 对象有关的所有 Error 对象都存储在 Errors 集合中，其中最低层次的错误为集合中的第一个对象，最高层次的错误为集合中的最后一个对象。

出现 DAO 错误时，Visual Basic 的 Err 对象将包含 Errors 集合中第一个对象的错误号。要确定是否出现了其他的 DAO 错误，可检查 Errors 集合。Errors 集合中的 Error 对象的 Number 和 Description 属性应该和 Visual Basic 中的 Err 对象的 Number 和 Description 属性相匹配。

一、AccessError 方法

使用 Err 对象的 Raise 方法可产生一个没有出现过的 Visual Basic 错误并确定与该错误有关的描述字符串。然而，不能使用 Raise 方法产生 Microsoft Access 错误或 DAO 错误。要确定与实际上没有发生的 Microsoft Access 错误或 DAO 错误有关的描述字符串，可使用 AccessError 方法。

二、Error 方法

使用 Error 事件可跟踪发生在 Microsoft Access 窗体或报表中的错误。例如，用户在数据类型为 Date/Time 的字段中输入文本就产生了 Error 事件。如果您把 Error 事件过程添加到“雇员”窗体上，然后在“雇佣日期”字段中输入一个文本值，就会运行 Error 事件过程。

Error 事件过程有一个整型参数 DataErr。在运行 Error 事件过程时，DataErr 参数将包含出现的 Microsoft Access 错误的编号。确定错误编号的唯一方法就是在事件过程中检测 DataErr 参数。在产生 Error 事件之后 Err 对象并没有包含错误信息。可在 AccessError 方法中使用 DataErr 参数的值来确定错误编号和描述字符串。

在 Microsoft Access 97 中提供 Error 语句和 Error 函数的目的只是为了向后兼容。在编写新代码时，最好使用 Err 和 Error 对象、AccessError 函数和 Error 事件来获取错误信息。

第十六章

形成整体数据库

前面介绍了数据库中的五种最主要的对象：表、查询、窗体、报表和宏以及 Visual Basic 语言。现在简要介绍一下如何将数据库完善，并且加入一定的安全机制，还需要创建用户自己的菜单、快捷菜单、工具条等附属对象。

16.1 使用外部数据

在使用数据库时，由于数据量特别大，而且是公用的，因此不可能只使用自己的数据，不可避免的，要使用外部数据。外部数据多种多样，可以是另外一个数据库中的数据、一个文本文件、一个 HTML 文件或一个电子表格。Access 可以提供以下的外部数据：

- Microsoft Access 数据库以及其他利用 Microsoft Jet 数据库工程的数据库，例如使用 Visual Basic、Visual C++ 或 Microsoft Excel 创建的数据库；

- Microsoft FoxPro 2.0、2.5、2.6、3.0 版，但是只读的；
- dBASE III、dBASE IV 和 dBase V；
- Microsoft Excel 3.0、4.0、5.0、7.0、8.0 版工作表；
- Lotus 1-2-3 WK1、WK3、WKS 的扩展表；
- HTML 文件中的管状数据；
- ODBC 数据库，例如 Microsoft SQL Server 4.2 以上的数据库。

在使用外部数据时，它必须允许许多用户读/写，这样用户就可以在连接外部数据时，直接在自己的应用程序中更改它。一般而言，外部数据在使用时以表的形式出现。

16.1.1 链接外部数据

链接外部数据是指用户与外部数据库建立关系，而用户在使用链接表时，就和使用本地表没有任何差别。用户可以利用链接表创建窗体，报表只是在执行的效率上有所不同，因为在使用链接表数据时，Access 需要从源数据库中调入数据，这需要时间。

Access 在“数据库窗口”中使用不同的图标来标识不同数据库下链接进来的表，图 16.1 所示即为一个链接表。用户可以发现在表前面有一个箭头，这表明这个表是此数据库与外部数据库相链接而得到的。当然，图标的另外部分是用来指明源数据库的类型，或表的类型。

要链接外部表，用户只要单击“文件”菜单，找到“获取外部数据”命令，将显示下拉列表，选中“链接表”，单击，如图 16.2 所示。Access 将显示“链接”对话框，如图 16.3 所示。此对话框与打开数据库的对话框十分相似。

用户首先选择要链接的对象所在的数据库的类型，在“文件类型”框中，单击下拉箭

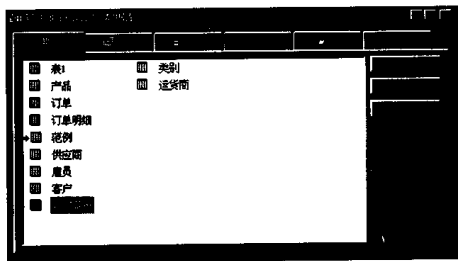


图 16.1

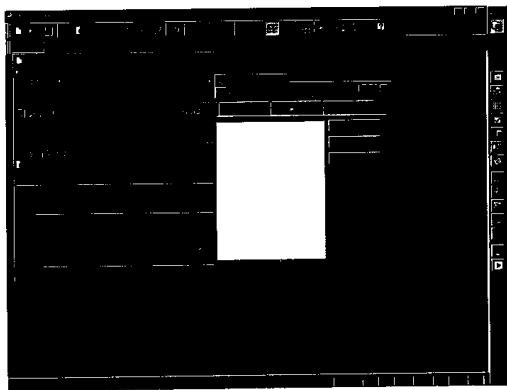


图 16.2

头，从下拉列表中选中用户所需的文件类型。这里列出了在 Access 中能使用的所有外部数据类型。然后在查找范围中，选中相应的源文件所在的目录。如果需要从 Web 上查找，则用户可以单击查找范围中的下拉列表中的 Internet 位置，Access 将弹出如图 16.4 所示的“添加/更改 FTP 位置”对话框。用户如果要连到一个新 FTP 服务器上，则需要填入相应的 FTP 地址及其他设置，最后单击“确定”按钮，添加至图 16.3 的列表中。用户只需选中相应的 FTP 地址，Access 自动连接到相应的 FTP 服务器中。

在 FTP 或 HTTP Internet 服务器上导入或链接（只读）数据，实际上，可以通过

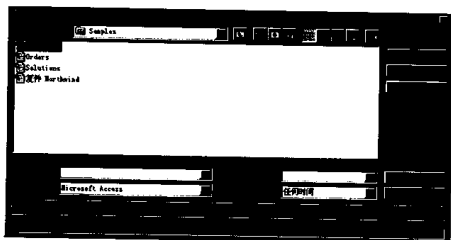


图 16.3

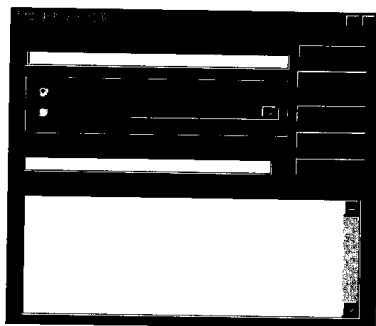


图 16.4

Internet 来复制数据而无需网络服务器连接。Access 将整个文件复制到本地计算机的 Microsoft Internet Explorer 高速缓冲文件夹中，此过程可能会花费一些时间，因为可能有大量数据要经过 Internet 来传送。

在链接时，链接表是一个原始的远程数据源的数据快照。在表处于打开状态时，将看不见远程位置的数据更新；但是在下一次打开表时，Access 将更新高速缓冲文件夹中的本地副本。

如果要链接 HTML 表和列表，则在“链接”对话框里的“文件类型”框中，请单击“HTML Document (*.HTML, *.htm)”选项；单击“查找范围”框右边的箭头，选择要导入或链接的 HTML 文件所在的驱动器和文件夹；然后双击相应的文件名。按照“链接 HTML 向导”中的提示进行操作。如果要编辑导入/导出的规格或指定不同的文件和字段格式，请单击“高级”按钮。如果 HTML 文件包含多个表或列表，对于每一个要导入或链接

的表或列表，则重复进行。

链接的 HTML 表或列表是只读的。在进行链接时，嵌入在 HTML 文件中的表单元格中的表将作为单独的表处理，而嵌入在表单元格中的列表将作为单元格的内容处理，列表上的每个项目将以回车换行符来分隔。

Access 将根据默认值把 HTML 链接转换为 HyperLink 数据类型字段，但只有在表字段或列表中的所有值都包含了由 <AHREF> 标记所定义的超级链接地址时，才会进行转换。在使用“导入 HTML 向导”或“链接 HTML 向导”时，可以更改数据类型。Access 将忽略嵌入在 HTML 表或列表中的 GIF 和 JPEG 图像。

对于超出数据行或列的数据，Access 97 将复制每个单元格的数据；而 Excel 97 则保存第一个或左上方的单元格中的数据，然后让其他单元格保持空白。

16.1.2 导入数据

如果用户需要导入数据，则单击“文件”菜单上的“获取外部数据”项，然后选中“导入表”命令，则 Access 显示“导入”对话框。该对话框的操作与“链接”对话框相同。

导入数据是将外部数据直接存入本数据库中，用户可以直接使用这些数据，而且与本地数据库一样，不存在效率的问题。

在选择完数据库或源文件以后，单击“确定”按钮，Access 会显示如图 16.5 所示的“导入对象”对话框，与图 16.6 所示的“链接表”对话框稍有不同的是，在“导入对象”对

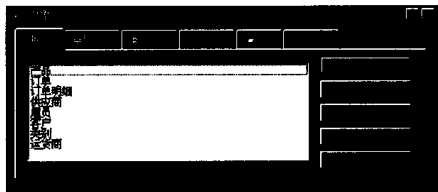


图 16.5

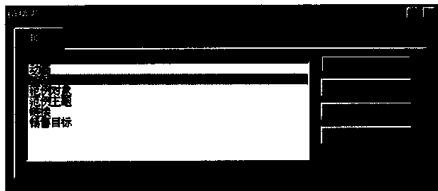


图 16.6

话框中用户可以选择数据库中的所有对象，而在“链接表”中导入的只能是表。选中相应的对象名，然后单击“确定”按钮，Access 将导入或链接该对象。

导入与链接后表的图标相同。

16.1.3 导入与链接

在 Access 数据库中，导入的数据将用新表创建导入数据的一个副本，在这一过程中源表或源文件不会改变。链接数据使用户能够读取外部数据源中的数据，并在大多数情况下更新这些数据，但不导入，外部数据源的格式并不改变，因此可以继续用创建文件的程序来使用它；但也可以用 Access 来添加、删除或编辑它的数据。

如果已确定数据将只在 Access 中使用，则应该使用导入方式。Access 对其自身的表的工作速度较快，而且如果需要，可以像在 Access 中创建的其他表一样，修改导入的表以满足需要。导入数据时，不能往已有的表上追加数据（除非是导入电子数据表或文本文件）。但是，已经导入了某个表之后，就可以执行追加查询，以便将其数据添加到另一个表中。

如果要使用的数据也将由 Access 之外的程序更新，则应该使用链接方式。使用这种方式，既可以保持当前更新、管理和共享数据的方法，而又可以使用 Access 来处理数据。例如，可以用外部数据创建查询、窗体和报表，将外部数据和 Access 表中的数据联合使用，甚至在其他正在以原始程序使用外部数据时还可以进行查看和编辑。

Access 可以链接其他 Access 数据库中的表，例如，可能需要使用网络上共享的另一个 Access 数据库中的表。如果要将某个数据库中的全部表都储存在网络服务器上，而将窗体、报表和其他对象保留在共享数据库的用户之间复制的另一个数据库中，则这种方法特别有用。使用“工具”菜单上“加载项”命令的“数据库拆分器”子命令，就可以很容易地将现有数据库拆分为两个数据库。

除表以外，也可以从另一个 Access 数据库中导入窗体或报表等数据库对象。

Access 使用不同的图标来代表链接表和存储在当前数据库中的表。如果删除链接表的图标，将会删除对表的链接，但并不删除外部表本身。

16.1.4 导入与链接另一个 Access 数据库

一、导入或链接表

在如图 16.3 所示的“链接”（或“导入”）对话框中的“文件类型”框内，确保已选定了 Access（.mdb），然后单击“查找范围”框右边的箭头，选定要导入或链接的 Access 数据库（.mdb）所在的驱动器和文件夹，然后双击数据库图标，Access 将显示如图 16.5 所示的“导入对象”对话框，单击要导入或链接的每个表。如果要导入的只是所选定的表的定义（而不是其中所包含的数据），则单击“选项”按钮，然后在“导入表”下单击“只导入定义”。如果要同时导入关系、自定义菜单和工具栏或者导入/导出规范，请单击“选项”按钮，然后在“导入”下选定要导入的项，如图 16.7 所示。

如果导入已经链接的表，Access 将不会导入数据，相反，它将表链接到它的数据源（实际上是复制链接信息）。如果要导入或链接的数据库有数据库密码，则在继续下去之前必须提供密码。链接这样的数据库表可能会产生意外的后果。如果要链接受到密码保护的 Access 数据库中的表，必须输入正确的密码。如果输入的密码正确，Access 将把数据库的密码与定

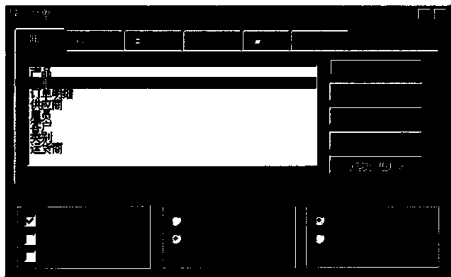


图 16.7

义链接表的信息一起保存。定义链接之后，任何能够打开表链接的数据库的用户，都能打开链接的表。当用户打开链接表时，Access 将采用保存的密码打开表所在的数据库。如果表所在的数据表的密码已经更改，下次在 Access 打开链接表之前将要求输入新的密码。

如果在相同的 Access 数据库中链接两个表，则两个表在其他数据库中已建立的任何关系都将会保留下来。如果导入包含 Lookup 字段的表，则应该也导入 Lookup 字段所引用的表或查询。如果不这样做，以“数据表”视图打开导入表时，Access 将会对每一个找不到的表或查询显示一个错误信息。如果要纠正这个问题，请导入丢失的表或查询。如果不能或不要导入丢失的表或查询，请在“设计”视图中打开导入的表，单击 Lookup 字段所引用的丢失的表或查询，单击“查找”标签，然后对“文本框”设置其“显示控件”属性。请对每个引用丢失的表或查询的 Lookup 字段重复这一过程。

二、导入其他对象

在如图 16.3 所示的“链接”（或“导入”）对话框中的“文件类型”框内，确保已选定了 Access（*.mdb），然后单击“查找范围”框右边的箭头，选定要导入或链接的 Access 数据库（*.mdb）所在的驱动器和文件夹，然后双击数据库图标。

在“导入对象”对话框中，单击要导入的每类对象的标签，然后单击要导入的每个对象。如果导入选择查询，而且要作为表来导入，则请单击“选项”按钮，然后在“导入查询”下单击“作为表”。操作查询将以查询导入，而不管此选项如何设置。同样，如果导入时要包括关系、自定义菜单和工具栏或者导入/导出规范，请单击“选项”按钮，然后在“导入”下选定要包括的项，如图 16.7 所示。

三、导入自定义的工具栏和菜单

用户可以从数据库导入所有自定义的菜单和工具栏。但是，如果所导入的菜单和工具栏与 Access 的菜单和工具栏的名称相同时，将不会导入这些工具栏、菜单或快捷菜单。首先在“文件类型”框中选择了 Access（*.mdb）；然后在“查找范围”框中，选择需要导入的 Access 数据库驱动器和文件夹（*.mdb）；然后双击数据库。

在“导入对象”对话框中，单击“选项”按钮。在“导入”标题下，选中“菜单及工具栏”复选框，如图 16.7 所示。

四、导入所有的数据库对象

如果需要，用户可以一次将一个数据库的所有对象导入到另一个数据库中。当创建了一个安全的 Access 数据库并要改变数据库的拥有者时，这种作法特别有用。

首先在“文件类型”框中，确保已选定了“Access (.mdb)”，然后单击“查找范围”框右侧的箭头，选定要从其中导入的 Access 数据库所有的驱动器 and 文件夹，双击数据库图标。如果要导入所有的表，请单击“全选”按钮。单击“查询”标签以显示可能的查询，然后单击“全选”按钮来导入全部查询。对于窗体、报表、宏和模块，请单击相应的数据库对象标签，然后单击“全选”按钮。如果有其他特殊的要求，单击“选项”按钮，选中相应的选项。图 16.8 所示即为选中了所有的窗体。

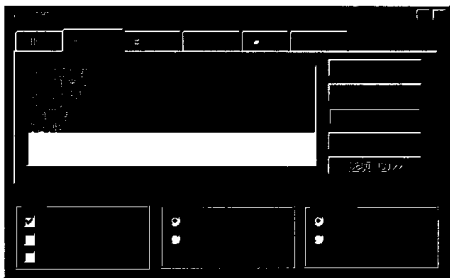


图 16.8

16.2 创建一个启动窗体

作为一个应用的数据库，有许多窗体，而且它们之间是有联系的，但是，又不能直接进行切换，因此，必须具有一个应用的控制中心，这就是启动窗体。一般在程序中，启动窗体是作为控制中心来使用的，当然也有的程序将启动窗体作为介绍或提示窗体使用。

在启动窗体中，用户可以启动所有的工作，两项任务之间的切换也需要由启动窗体来完成。先将一个窗体关闭，回到启动窗体状态下，然后再进行下一项任务。启动窗体是用户在运行应用程序时首先看到的窗体，因此必须美观。而且为了使用方便，在设计启动窗体时，应该将应用程序中主要任务集中在一个窗体中。

在设计启动窗体时，如果用户将来要执行的工作比较集中，例如订单，一般而言，用户只要每天输入订单的详细内容，即用户工作主要集中在一个窗体中，则应把这个窗体作为启动窗体。图 16.9 所示为 Orders 数据库的启动窗体，在这个窗体中，用户可以完成订单的制定，并且浏览相关内容，例如产品的详细资料等。

与此相对应的，如果用户的任务十分复杂，在数据库中有大量的窗体与报表，不能肯定

哪一个主要的，则可以将启动窗体设计成一个任务切换面板，将用户的主要任务用几个命令按钮来激活，打开用户所需的窗体与报表，这样可以使应用界面十分整洁而且简单。Northwind 数据库中的启动窗体就是这种类型的，如图 16.10 所示。

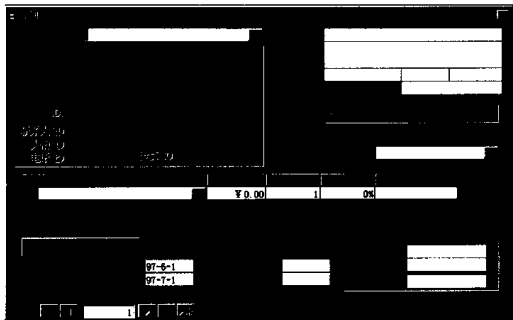


图 16.9

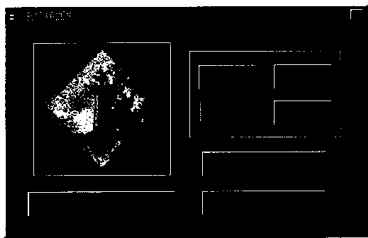


图 16.10

还有另一种类型的启动窗体，如图 16.11 所示的 Solutios 数据库中的启动窗体。它与前两种不同。这个启动窗体是作为一个浏览工具使用的，用户可以直接在窗体上查找自己所需的内容。

在设计完启动窗体后，用户必须在启动选项中设置参数。

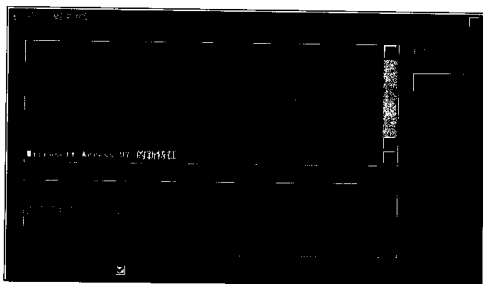


图 16.11

16.3 工具栏、菜单栏及快捷菜单

工具栏、菜单栏及快捷菜单在应用程序中是必不可少的，它允许在 Access 中按自己的方式组织命令，这样就可以快速地查找并且使用它们。在 Access 的旧版本中，工具栏仅包含按钮。由于在此版本中使用了新的技术，现在的工具栏可以只包含按钮或只包含菜单，或者包含两者的组合。这表明菜单栏和快捷菜单只是目前工具栏上的不同类型，因此可以用相同的方法自定义此三种类型。例如，虽然内置菜单栏按默认值显示在屏幕的上方，并且包含标准的菜单，如“文件”、“编辑”和“视图”；但是也可以通过增加或删除按钮和菜单，或将其移动到不同的位置来进行自定义。除自定义的内置菜单栏、快捷菜单和工具栏之外，也可以创建自己的自定义工具栏、菜单栏及快捷菜单。

16.3.1 利用宏创建工具栏、菜单栏及快捷菜单

新版本不再需要使用宏来创建菜单栏和快捷菜单，但是仍然支持宏，所以可以使用基于宏的已有应用程序，用户还可以从宏中创建 Access 97 样式的菜单栏或快捷菜单，并且可以从 Access 旧版本所创建的宏中创建新样式的菜单栏或快捷菜单。

首先在“数据库”窗口中，单击“宏”标签，然后单击所要创建菜单栏或快捷菜单的宏名。如果要从以前所用的自定义菜单宏中创建新样式的菜单栏或快捷菜单，仅选择顶层菜单宏，而不须选择宏组，然后指向“工具”菜单中的“宏”，单击“用宏创建菜单”或“用宏创建快捷菜单”命令。图 16.12 所示的即为利用客户宏来创建菜单，此时会在工具栏区域显示客户菜单，如图 16.13 所示。

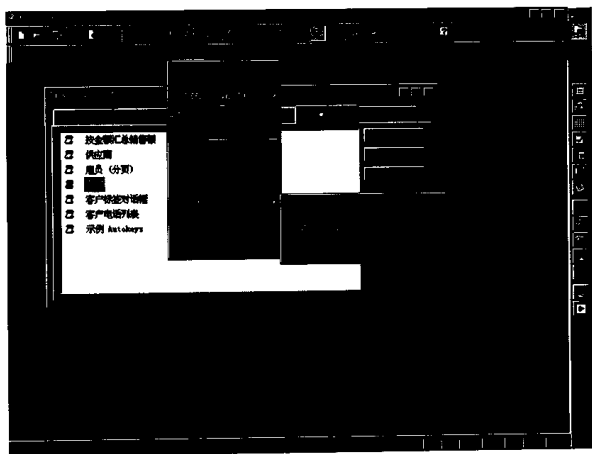


图 16.12



图 16.13

在 Access 中新建了菜单栏或快捷菜单之后，不会删除原始宏（或正在使用菜单栏宏和伴随菜单宏的组进行操作的宏）。除了只包含 AddMenu 或 RunCommand 操作（以前的 DoMenuItem）的宏之外，新的菜单栏仍然取决于原始宏。

如果要创建和自定义工具栏、菜单栏及快捷菜单，并且设置外观和行为属性，可以将指针指向“视图”菜单中的“工具栏”，然后单击“自定义”命令，Access 会显示如图 16.14 所示的“自定义”对话框来自定义工具栏、菜单栏或快捷菜单。

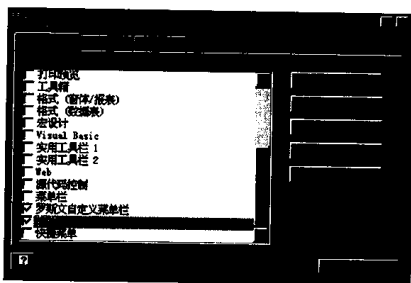


图 16.14

16.3.2 为当前数据库创建自定义工具栏

在“工具栏”标签单击“新建”按钮，Access 会显示如图 16.15 的“新建工具栏”。用户在“工具栏名称”框中键入所需要的名称，然后单击“确定”按钮，此时在屏幕上显示一个空白的工具框。在“工具栏”标签上单击“属性”按钮，Access 会显示如图 16.16 所示的“工具栏属性”对话框，在对话框中设置其他所需的属性。



图 16.15

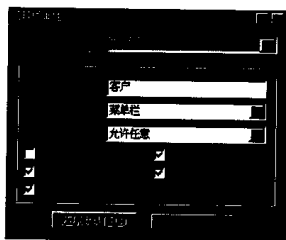


图 16.16

用户可以改工具栏名称。在“类型”框中，可以选择用户想要的类型，共有三种：工具栏、菜单栏和弹出式。此时选择工具栏，在“停靠方式”中可以限制此工具栏的位置。单击“关闭”按钮回到“自定义”对话框。现在新工具栏位于“自定义”对话框后面。如果要参阅新工具栏，将“自定义”对话框移动到旁边。

此时只是完成了工具栏的定义，还要从“自定义”对话框中添加按钮或从另一个工具栏中移动或复制按钮。用户也可以将菜单添加到工具栏，方法见“为当前的数据库创建自定义菜单栏”。

Access 能自动创建包含按钮的工具栏, 该按钮可以执行已有的宏。只需在“数据库”窗口中单击“宏”标签, 然后单击所要创建的工具栏的宏名, 再指向“工具”菜单中的“宏”命令, 然后单击“用宏创建工具栏”命令。

16.3.3 为当前的数据库创建自定义菜单栏

在“工具栏”标签单击“新建”按钮, Access 会显示如图 16.15 的“新建工具栏”。用户在“工具栏名称”框中键入所需要的名称, 然后单击“确定”按钮, 此时在屏幕上显示一个空白的工具框。在“工具栏”标签上单击“属性”按钮, Access 会显示如图 16.16 所示的“工具栏属性”对话框。在“类型”列表上单击“菜单栏”, 再设置其他所需的属性, 然后单击“关闭”按钮。此时新菜单栏现在位于“自定义”对话框的后面, 如果要查看新建的菜单栏, 将“自定义”对话框移动到旁边。

此时只是完成了菜单栏的定义, 如果要添加自定义菜单, 在“自定义”对话框中单击“命令”标签, Access 显示如图 16.17 所示的画面。在“类别”框中, 请单击“新菜单”, 然后从“命令”框中, 将“新菜单”拖动到已经显示的菜单栏上。在菜单栏上, 在添加的菜单上单击鼠标右键, 然后在如图 16.18 所示的快捷菜单上的“命名”框键入名称, 按 Enter 键。将菜单添加到工具栏中的方法与此类似。

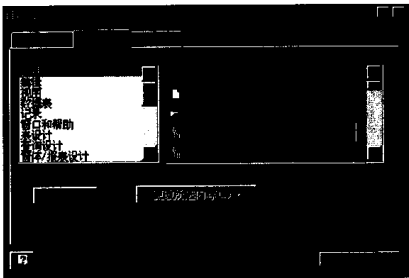


图 16.17

此外还要从“自定义”对话框中添加命令, 或从其他菜单中移动或复制命令, 以及添加内置的菜单。

对于 Access 旧版本创建的应用程序, Access 仍然支持菜单栏宏。

16.3.4 为当前数据库创建自定义快捷菜单

在“工具栏”标签上单击“新建”按钮, Access 会显示如图 16.15 的“新建工具栏”。用户在“工具栏名称”框中键入所需要的名称, 然后单击“确定”按钮, 此时在屏幕上显示一个空白的工具框。在“工具栏”标签上单击“属性”按钮, Access 会显示如图 16.16 所示

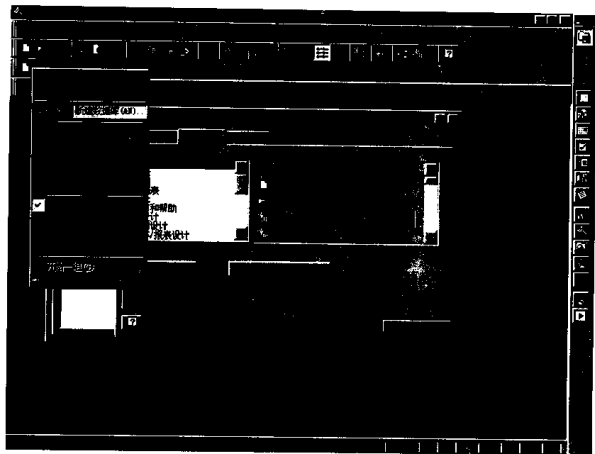


图 16.18

的“工具栏属性”对话框。在“类型”列表中，单击“弹出式”，设置所需的“允许自定义”属性，然后单击“关闭”按钮，回到“自定义”对话框。

Access 将该快捷菜单添加到“快捷菜单”工具栏，其中包含所有的自定义快捷菜单。在“工具栏”标签的“工具栏”框中单击“快捷菜单”，Access 会显示如图 16.19 所示的“快捷菜单”工具栏。在“快捷菜单”工具栏上，单击“自定义”类别，用户可从“自定义”对话框中添加命令，或从其他菜单中移动或复制命令。

16.3.5 将按钮添加到“菜单栏”、“快捷菜单”工具栏或其他工具栏上

将鼠标指针指向“视图”菜单上的“工具栏”，然后单击“自定义”命令，在显示的“自定义”对话框中选中要添加按钮的“菜单栏”、“快捷菜单”工具栏或其他工具栏。然后单击“命令”标签，在“类别”框上单击要按钮执行命令的类别。表 16.1 列出了各种情况。

表 16.1 添加按钮情况

如果要添加	请单击
内置命令	相应的菜单或查看类别
浏览 HTML 页的按钮	Web
操作 SourceSafe 的按钮	程序代码控件

续表 16.1

如果要添加	请单击
创建自定义控件的命令	ActiveX 控件
在默认视图中, 显示窗体、报表或其他数据库对象的命令	下列之一: 所有表、所有查询、所有窗体、所有报表
运行宏的命令	所有宏

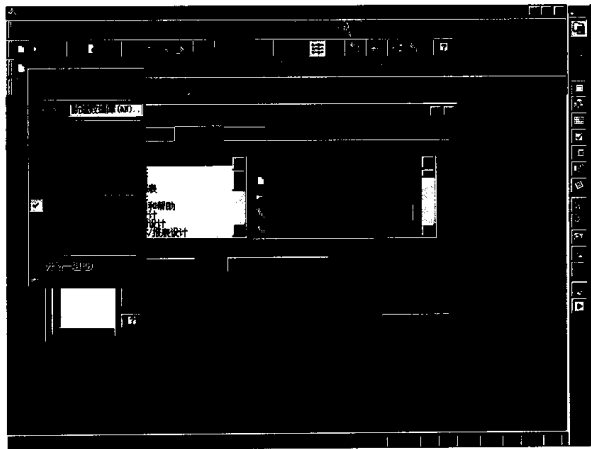


图 16.19

然后, 从“命令”框中将所需要的按钮、对象或 ActiveX 控件拖动到已经显示的“菜单栏”、“快捷菜单”工具栏或其他工具栏上。

如果要快速地添加打开数据库对象的按钮, 可以从“数据库”窗口中将对象拖动到工具栏上。Access 可以自动创建包含执行已有宏的按钮的工具栏。

16.3.6 移动或复制工具栏按钮

首先在屏幕上显示要移动或复制按钮的工具栏和目的工具栏。

当“自定义”对话框打开时, 如果要移动工具栏按钮, 请在同一个工具栏或另一个工具栏上将按钮拖动到新的位置。如果对话框是关闭的, 按下 Alt 键, 并且将按钮拖动到新的位置。

当“自定义”对话框打开时, 如果要复制工具栏按钮, 则按下 Ctrl 键, 并且将按钮拖

动到新的位置。如果对话框是关闭的，按下 Ctrl + Alt 键，并且将按钮拖动到新的位置。

16.3.7 将自定义工具栏、菜单栏、快捷菜单附加于窗体或报表

首先在“设计”视图中打开相应的窗体或报表，然后单击窗体选择器或报表选择器，再单击工具栏上的“属性”按钮，在“菜单栏”属性框中指定所需的工具栏名称、菜单栏名称或快捷菜单名称。

16.3.8 显示或隐藏工具栏、菜单栏或快捷菜单

将鼠标指针指向“视图”菜单上的“工具栏”，然后单击“自定义”命令。在“工具栏”标签上选择或清除要显示或隐藏项目的复选框。如果要显示特别的快捷菜单，选择“快捷菜单”，单击要显示的快捷菜单类别，然后单击快捷菜单的名称。如果要显示自定义的快捷菜单，则单击“自定义”类别。

如果要快速地隐藏浮动工具栏，请单击“关闭”按钮。如果要快速地显示或隐藏自定义的工具栏或菜单栏，在任何工具栏或菜单栏上单击鼠标右键，然后在快捷菜单上单击工具栏或菜单栏的名称，如图 16.20 所示。



图 16.20

16.4 设置数据库安全

Access 提供了设置数据库安全的两种传统方法：为打开的数据库设置密码；或设置用户级安全，以限制用户访问或更改数据库的哪一部分。除这些方法之外，还可将数据库保存为 MDE 文件以删除数据库中可编辑的 Visual Basic 代码，以此来防止对窗体、报表和模块的设计进行修改。

16.4.1 设置密码

最简单的方法是为打开的数据库设置密码。设置密码后，打开数据库时将显示要求输入密码的对话框。只有键入正确密码的用户才可以打开数据库。这个方法是安全的（Access 对密码进行加密，因此直接查看数据库文件是无法得到密码的），但只应用于打开数据库。在数据库打开之后，数据库中的所有对象对用户都将是可用的（除非定义用户级安全）。对于在某个用户组中共享的数据库或是单机上的数据库，设置密码通常就足够了。

下面介绍添加数据库密码的步骤。

(1) 关闭数据库，如果数据库在网络上共享，应使所有其他用户关闭该数据库。然后为数据库复制一个备份并将其存储在安全的地方。

(2) 单击“文件”菜单中的“打开数据库”命令，选中“独占”复选框，然后打开数据库。

(3) 单击“工具”菜单“安全”子菜单上的“设置数据库密码”命令。此时 Access 显示“设置数据库密码”对话框。在对话框的“密码”框中，键入自己的密码。密码是区分大小写的。然后在“验证”框中再次键入密码以进行确认，最后单击“确定”按钮。如果两次输入不相同，Access 会出现提示信息提示用户在“验证”框中重新输入密码进行确认。

这样密码即设置完成。下一次打开数据库时，将显示要求输入密码的对话框。如果要复制数据库，则不要使用数据库密码；如果设置了密码，复制的数据库将不能同步。

16.4.2 用户级安全

设置数据库安全最灵活和最广泛的方法是设置用户级安全。这种安全类似于很多网络中使用的方法，它要求用户在启动 Access 时确认自己的身份并键入密码。在工作组信息文件中，用户标识为组的成员。Access 提供两种默认的组：管理员（称为管理员组）和用户（称为用户组），但也可定义其他的组。

可以为组 and 用户授予权限，规定他们如何使用数据库中的对象。例如，用户组的成员可以查看、输入或修改“顾客”表中的数据，但不能更改表的设计；或者只允许查看包含订单数据的表，而不能访问“工资”表。管理员组的成员则对数据库中的所有对象都具有完全的权限。如果要设置更细致的控制，可以创建自己的组账号，为其指定适当的权限，然后将用户添加到组中。

使用用户级安全的三个主要原因是：

- 保护代码的知识产权；
- 防止用户无意地更改应用程序所依赖的代码或对象而破坏应用程序；
- 保护数据库中的敏感数据。

16.4.3 MDE 文件

如果数据库包含 Visual Basic 程序代码，将数据库保存为 MDE 文件将编译所有模块、删除所有可编辑的源代码并且压缩目标数据库。Visual Basic 程序代码将继续执行，但是不能进行查看或编辑，并且数据库的大小将因程序代码的删除而减少。除此之外，内存将优化使用，这将有助于提高性能。

将数据库保存为 MDE 文件可防止下列的操作：

- 在“设计”视图中查看、修改或创建窗体、报表或模块；
- 增加、删除或更改对对象库或数据库的引用；
- 使用 Access 或 VBA 对象模式的属性或方法更改程序代码，因为 MDE 文件不包含源代码。使用“选项”对话框，更改数据库的 VBA 项目名称；
- 导入或导出窗体、报表或模块。但是，表、查询及宏可以由导入或导出到非 MDE 数据库。MDE 数据库中的任何表、查询或宏可以导入到另一数据库，但是窗体、报表或模块不能导入到另一数据库。

在将数据库保存为 MDE 文件之前，用户必须确保保存了原始数据库的副本。如果需要修改保存为 MDE 文件的数据库中的窗体、报表或模块，必须打开原始数据库加以修改，然后再次保存为 MDE 文件。如果以后还要对窗体、报表或模块的设计进行修改，将包含表的数据库保存为 MDE 文件会生成复杂且不一致的不同版本的数据。由于这个原因，将数据库保存为 MDE 文件对一个前端/后端应用程序的前端数据库是非常合适的。

同样，在 Access 的将来版本中，不能打开、转换或运行 Access 97 的 MDE 文件中的程序代码。将 Access 97 的 MDE 文件转换为将来版本的唯一方法是打开创建 MDE 文件的原始数据库并进行转换，然后将转换的数据库以 MDE 文件保存。

要保存为 MDE 文件的数据库，如果数据库采用用户级安全，必须符合特定准则；如果数据库是同步复制的，必须先删除同步复制系统表和属性；如果数据库引用另一个数据库或加载项，必须将这个引用链中所有的数据库或加载项保存为 MDE 文件。

生成 MDE 文件的方法如下：

- (1) 关闭数据库，如果正在多用户环境中工作，请确保所有其他用户已经关闭数据库；
- (2) 在“工具”菜单上，指向“数据库实用工具”，然后单击“生成 MDE 文件”命令；
- (3) Access 将显示如图 16.21 所示的“保存数据库为 MDE”对话框，在对话框中指定要另存为 MDE 文件的数据库，并且单击“生成 MDE”按钮；

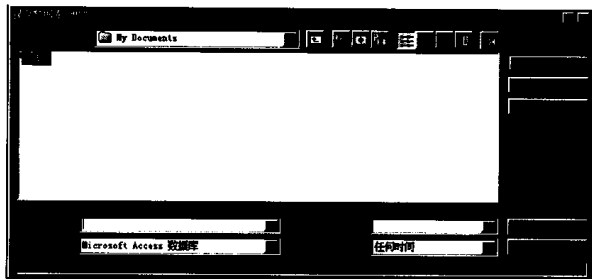


图 16.21

(4) Access 显示如图 16.22 所示的“保存 MDE 为”对话框，在对话框中指定数据库的名称、驱动器以及文件夹；

- (5) 单击“确认”按钮完成 MDE 数据库的生成。

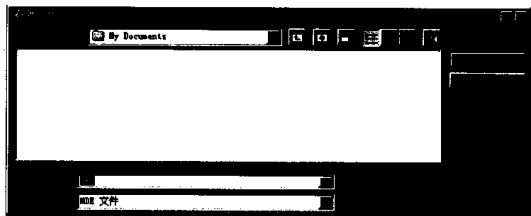


图 16.22

16.4.4 设置启动选项

要设置启动选项，单击“工具”菜单上的“启动”命令，Access 显示如图 16.23 所示的“启动”对话框。

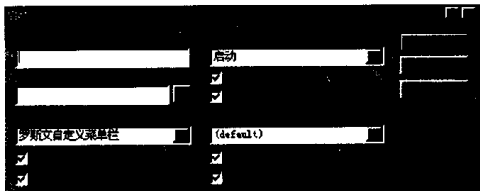


图 16.23

在“应用程序的标题”框中填入用户给应用程序起的标题。如果不填，在运行时只显示 Access 作为标题。在“应用程序图标”框中，可单击旁边的“生成器”按钮选择图标。在“菜单栏”框中，添入从下拉列表中选出的菜单栏。在“快捷菜单栏”框中，添入从下拉列表中选出的快捷菜单栏，缺省时为 (default)。在“显示窗体”框中，单击当前数据库中的要设为启动窗体的窗体名。如果不希望用户查看或使用跟随该窗体显示的数据库窗口，可清除“显示数据库窗口”复选框。但是，即使清除了“显示数据库窗口”复选框，数据库窗口仍然是可以访问的。这是由于 Access 中有一些特殊键还有效，只要按 F11 键，“数据库窗体”即可显示。

如果在数据库窗口、“调试窗口”、菜单栏或模块中的特殊键有效，用户就可以通过这些特殊键对数据库进行修改，因此为了数据库的安全性，必须使在数据库窗口、“调试窗口”、菜单栏或模块中的特殊键失效。方法如下：单击对话框中的“高级”按钮，Access 显示更多的复选框，清除或选中“使用 Access 特殊键”复选框，以使表 16.2 所示的按键失效。

表 16.2 特殊键功能

按 键	作 用
F11 或 Alt + F1	将数据库窗口置于顶层
Ctrl + G	将“调试窗口”置于顶层
Ctrl + F11	在自定义菜单栏与内置菜单栏之间切换
Ctrl + BREAK	停止正在运行的程序代码，并在“模块”窗口中显示当前模块

无论是否选中“使用 Access 特殊键”复选框，只要“允许在发生错误后查看代码”复选框处于选中状态，就可以在发生“执行期间错误”后显示“模块”窗口来调试程序代码。如果清除了“使用 Access 特殊键”复选框，并在“菜单栏”框中指定自定义菜单栏，则不能访问内置菜单栏。即使同时清除了“使用 Access 特殊键”和“显示数据库窗口”复选框，

用户仍然可以访问数据库窗口。只要用户多次试图打开数据库列表中的同一个数据库，该数据库自动显示在“文件”菜单上，就可以访问数据库窗口。如果要防止用户访问该列表，可以用自定义菜单替换“文件”菜单。

由于以上原因，为了数据库的安全，应该将所有复选框清除，除非用户另有要求，如图 16.24 所示。

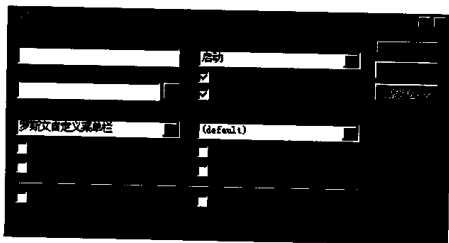


图 16.24

还要说明的是：“启动”选项只适用于当前的数据库或应用程序；“启动”属性不会覆盖特定的工具栏、菜单栏、快捷菜单、窗体或报表的属性设置。例如，“启动”对话框中的“菜单栏”与“快捷菜单栏”选项不会覆盖特定窗体或报表的自定义菜单栏或快捷菜单的属性设置。因此，在打开窗体或报表时，Access 会显示附加于该窗体或报表的自定义菜单栏，而不是“启动”对话框中所指定的全局菜单栏。

用户可以使用“启动”对话框来代替 AutoExec 宏。AutoExec 宏在“启动”选项生效之后运行，因此，应避免在 AutoExec 宏中执行任何会更改启动选项设置效果的操作。例如，如果在“启动”对话框的“显示窗体”框中指定了某个窗体，同时又在 AutoExec 宏中使用了 OpenForm 操作，则 Access 将首先显示“启动”对话框中指定的窗体，然后立即显示 OpenForm 操作中指定的窗体。

如果数据库中已定义了用户级安全，则通过检查用户是否具有数据库管理员权限，就可以阻止用户更改“启动”对话框中的设置。

如果要忽略“启动”选项设置，可在打开数据库时，按下“忽略”(Shift)键。除非再次打开该数据库或应用程序，否则“启动”对话框中的这些设置不会生效。



[General Information]

书名=Access 97中文版使用指南

作者=王劲松

页数=308

SS号=10011960

出版日期=1998年4月第1版