Access 数据库应用实例完全解析

启明工作室　编著

人民邮电出版社

需要全本请访问 www.ertongbook.com

图书在版编目（**CIP**）数据

Access 数据库应用实例完全解析 / 启明工作室编著．—北京：人民邮电出版社，2006.4

ISBN 7-115-14666-7

Ⅰ．A… Ⅱ．启… Ⅲ．关系数据库－数据库管理系统，Access Ⅳ．TP311.138

中国版本图书馆 CIP 数据核字（2006）第 027795 号

内 容 提 要

本书深入浅出、循序渐进地通过 7 个应用系统的开发制作过程，为读者详细介绍如何应用 Access 2003 进行系统开发。本书附赠光盘中还为每个实例制作了 PPT 文件，详细介绍和演示实例开发所需的技术和过程，方便读者学习和理解。

本书案例贴近实际，内容精练，适合大中专院校的学生和软件开发人员参考使用。

混合编程之案例详解

Access 数据库应用实例完全解析

◆ 编　　著　启明工作室

责任编辑　刘　浩

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：21

字数：510 千字　　2006 年 4 月第 1 版

印数：1 – 6 000 册　　2006 年 4 月北京第 1 次印刷

ISBN 7-115-14666-7/TP・5343

定价：36.00 元（附光盘）

读者服务热线：（010）67132692　印装质量热线：（010）67129223

前　言

数据库是信息化过程中不可缺少的一环，不管是进销存管理、教务管理、财务管理或是各种各样的资料都需要整理和保存，Access 数据库就是处理这些事务的最佳助手。经过多年的发展，Access 已经成为桌面数据库领域开发与应用的标准，广泛应用于公司办公自动化、信息管理系统、中小型数据库开发等方面。

和其他单纯介绍 Access 2003 基本知识的图书不同，本书通过 7 个应用系统的开发制作过程，为读者提供了极有针对性的应用实例，所选用的实例都是现实中非常需要的，每一个案例独立成为一个完整的单元。通过对实例的讲解，透彻地解析了 Access 2003 的各种功能。这些案例分别为：教务管理系统、超市进销存管理系统、学校工会管理系统、项目管理系统、财务管理系统、资产管理系统、考勤管理系统。

对于以上的 7 个案例，我们从系统的需求分析开始，建立系统的信息流程图，为系统设计数据库和数据表，并在此基础上建立和完善数据库系统：建立各种各样的查询，构造灵活多样的窗体，设计形式丰富的报表，以及搭建功能强大的宏和模块。每一个实例都完整解析出来，使读者不仅了解如何构建管理信息系统，而且对 Access 2003 的强大功能又有一个重新的认识。

本书最大的特点是以应用开发为中心，尤其是以公司的信息管理为切入点，囊括了各个领域的公司信息管理，对于急于实现公司办公自动化的办公室管理人员来说，本书为他们节约了时间，达到事半功倍的效果。

本书由陈文执笔编写，张文松、陈杰、李海燕、赵威等同志在整理材料方面做了很多工作，再次深表谢意。

由于作者水平有限，书中难免有不足和疏忽之处，恳请广大专家读者批评指正（电子函件：book_better@sina.com）。

编者

2006 年 4 月

目　　录

第 1 章　Access 数据库系统设计实用技术 ····· 1

1.1　数据库设计技术 ····· 1
1.2　Access VBA 编程技术 ····· 4
1.3　数据库优化技术 ····· 31

第 2 章　教务管理系统 ····· 35

2.1　概述 ····· 35
2.2　系统分析与总体设计 ····· 35
2.3　登录窗体和系统主界面设计 ····· 41
2.4　学生信息管理模块设计 ····· 47
2.5　教师信息管理模块设计 ····· 64
2.6　选课信息管理模块设计 ····· 75

第 3 章　学校工会管理系统 ····· 81

3.1　概述 ····· 81
3.2　系统分析与总体设计 ····· 81
3.3　登录窗体和系统主界面设计 ····· 92
3.4　查询任务模块设计 ····· 99
3.5　档案编辑模块设计 ····· 113
3.6　政治面貌统计模块设计 ····· 117
3.7　会员花名册设计 ····· 121
3.8　用户管理模块设计 ····· 123
3.9　菜单设计 ····· 130
3.10　系统启动设置 ····· 132

第 4 章　超市进销存管理系统 ····· 133

4.1　概述 ····· 133
4.2　系统分析与总体设计 ····· 133
4.3　登录窗体和系统主界面设计 ····· 142
4.4　商品进货数据录入模块设计 ····· 149
4.5　商品库存数据查询模块设计 ····· 151
4.6　商品上柜数据录入模块设计 ····· 154
4.7　销售业绩查询汇总模块设计 ····· 157

4.8 交接班结转清模块设计……161
4.9 销售数据录入模块设计……162

第 5 章 项目管理系统 ……165

5.1 概述 ……165
5.2 系统分析与总体设计……165
5.3 登录窗体和系统主界面设计……174
5.4 客户管理模块设计……180
5.5 计时卡管理模块设计……195
5.6 帐单查询管理模块设计……200

第 6 章 财务管理系统 ……205

6.1 概述 ……205
6.2 系统分析与总体设计……205
6.3 登录窗体和系统主界面设计……213
6.4 日记账和账户管理模块的设计……220
6.5 账务查询模块的设计……223
6.6 账务结算模块的设计……235
6.7 账务分析模块的设计……236

第 7 章 资产管理系统 ……240

7.1 概述 ……240
7.2 系统分析与总体设计……240
7.3 登录窗体和系统主界面设计……251
7.4 基本操作模块设计……268
7.5 查询任务模块设计……270
7.6 操作任务模块设计……274
7.7 分析任务模块设计……282

第 8 章 考勤管理系统 ……292

8.1 概述 ……292
8.2 系统分析与总体设计……292
8.3 登录窗体和系统主界面设计……300
8.4 添加记录模块设计……304
8.5 记录检索模块设计……311
8.6 考勤统计模块设计……320
8.7 上下班时间模块设计……324

需要全本请访问 www.ertongbook.com

第 1 章　Access 数据库系统设计实用技术

设计 Access 数据库系统时需要掌握的基础与技巧包括：

- Access 数据库基本对象的设计，比如表、查询、窗体和报表等的设计。
- 数据库设计方法，例如需求分析、概念结构设计、逻辑设计、物理设计和数据库实施。
- Access VBA 语言基础及编程方法。
- Access 数据库的分析优化。

1.1　数据库设计技术

数据库系统的一个核心问题就是如何设计能满足用户要求、性能良好的数据库，这是数据库设计（Database design）的基本要求。本节讨论数据库设计的方法，重点介绍数据库的需求分析、概念设计及逻辑设计 3 个方面。

数据库设计是将现实世界中的信息根据数据库的组织结构在计算机中表现。数据库的设计可分为两大部分：

- 数据库的逻辑设计，即数据库管理系统要处理的数据库全局逻辑结构；
- 数据库的物理设计，它是在逻辑结构已确定了的前提下设计数据库的存储结构。

为完成这两大部分的设计工作，整个设计过程可分为 6 个阶段，如图 1.1 所示。

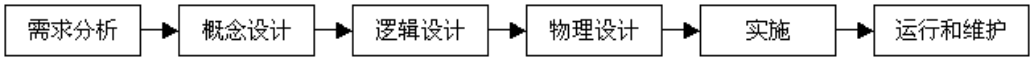


图 1.1　数据库设计步骤

（1）需求分析阶段。进行数据库设计首先必须准确地了解分析用户需求（包括数据和处理）。需求分析是整个设计过程的基础，是最困难、最耗时间的一步。需求分析做得不好，甚至会导致整个数据库设计返工重做。

（2）概念结构设计阶段。概念结构设计是整个数据库设计的关键，它通过对用户需求进行综合、归纳与抽象，形成一个独立于具体数据库管理系统的概念模型（实体模型）。

（3）逻辑结构设计阶段。逻辑结构设计是将概念结构转换为某个数据库管理系统所支持的数据模型（关系模型），并对其进行优化。

（4）数据库物理设计阶段。数据库物理设计是为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存储方法）。

（5）数据库实施阶段。在数据库实施阶段，设计运用数据库管理系统提供的数据语言及其宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并试运行。

（6）数据库运行和维护阶段。数据库经过试运行之后，即可投入正式运行。在数据库系

统运行过程中必须不断地对其进行评价、调整和修改。

设计一个完善的数据库是不可能一蹴而就的，它往往是上述 6 个阶段的不断反复的过程。

1.1.1 需求分析

要设计一个有效的数据库，必须用系统工程的观点来考虑问题。在系统分析阶段，设计者和用户双方要密切合作，共同收集和分析数据管理中信息的内容和用户的要求。在调研中，首先要了解设计中的数据库所管理的数据将覆盖哪些工作部门，每个部门的数据来自何处，它们依照什么样的原则处理加工这些数据，在处理完毕后输出哪些信息到其他部门。

其次要确定系统的边界，在与用户充分讨论的基础上，确定计算机数据处理范围，确定哪些工作要由人工来完成，确定人机界面。最后得到业务信息流程图。信息流程图中的每个子系统都可抽象为如图 1.2 所示的框图。

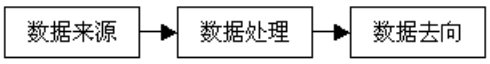


图 1.2　信息流程图

在系统分析过程中，要确定数据管理中的信息内容和处理要求。信息内容是指数据库系统所要用到的数据的名称、类型、值域的范围，数据与数据之间的联系的类型、方式（是一对一、一对多还是多对多）以及数据库中需要存储哪些数据。

处理要求是指用户需要数据库系统完成哪些处理功能，响应时间要求如何，是否采用批处理方式，各种处理有无优先级别。

该阶段工作是否能准确地反映实际系统的信息流程情况和用户对数据库系统的要求，直接影响到以后的各阶段的工作，并影响到数据库系统将来运行的效率，因为分析阶段的工作是整个数据库设计的基础。

分析阶段工作的成果是：

（1）需求说明书。在调查用户需求的基础上，提供一份说明书，一般由分层的数据流程图和数据词典两大部分组成。

（2）初始数据库规模文档。这是在需求说明书的基础上总结出来的文档，主要是解答数据库规模、数据来源等一系列问题。

（3）硬件建议文档。在确定初始数据库文档之后，应提出一份硬件需求建议文档，主要指出磁盘大小、数目，CPU 的指标，内容容量等。这份硬件建议文档可以帮助用户配置硬件系统。

1.1.2 概念结构设计

概念结构设计所涉及到的数据是独立于硬件和软件系统的，它的目标是以用户可以理解的形式来表达信息的流程，从而便于与不熟悉计算机的用户交换意见。概念结构设计阶段所得到的概念结构应能充分反映现实世界中实体和实体之间的联系，应是一个现实世界的真实模型。这个模型是各种数据模型的共同基础，它易于向关系、层次和网状模型转换。

在概念结构设计阶段，面对错综复杂的现实世界，要想直接得到全局视图是较困难的。一般需要两个阶段。第一阶段是根据分析阶段所收集到的用户对数据和处理的需求，为产生全局

视图，得到每个用户各自的局部视图，对每个用户的局部数据结构进行描述。在第二阶段是在定义了各用户的局部视图的基础上，利用一定的工具分析各个局部视图，并把它们合并成一个统一的全局数据结构，即全局视图。

全局视图被称为数据库概念模型，它应满足以下要求：

- 尽可能地消除冗余的数据和冗余的数据联系，保持最小的冗余度；
- 能准确地反映每个局部视图的要求；
- 不影响对数据处理的要求，在合并前能处理的，在合并后同样也能处理；
- 内部不允许发生矛盾。

1.1.3　逻辑设计

逻辑结构设计的任务是把概念结构转换为数据库管理系统能处理的数据模型，即网状、层次或关系模型。

把概念结构模型转换成关系数据模型的过程，是把实体和实体之间的关系转换成一个个二维关系表，中间存在着多种可能的组合，必须从中选取一个性能好的关系模式集合作为关系数据库的模式。在设计关系模式时要强调“独立联系、独立表达”这条原则。

到逻辑设计结束，应得到用户系统的关系模型和各个关系的模式结构，并能说明其中的一行约束和要求等的具体内容。

1.1.4　物理设计

对于给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程称为物理结构设计。数据库的物理结构主要指数据库的存储记录格式、存储记录安排和存储方法，这些都依赖于所使用的系统。在网状模型和层次模型系统中，这一部分内容较复杂，因为它们用指针表示记录的联系。关系模型系统比较简单一些，仅包含索引机制、空间大小、块的大小等内容。

在物理结构设计中，应先确定数据库的物理结构，然后对物理结构进行评价，评价的重点是时间和空间的效率。数据的存储决定了数据库占用多少空间，数据的处理决定了操作的效率。物理结构设计应尽量减少占用的存储空间，也应尽量减少操作次数，使响应时间越快越好。如果评价结果满足原设计的要求，则转向物理实施。否则，就重新修改或重新设计物理结构，有时甚至要回到逻辑设计阶段修改数据模型。

物理设计完成之后，就应该得到详细的磁盘分配方案、存储方案、各种基表的详细信息等，根据这些信息就可以上机建立数据库。

1.1.5　数据库实施

对数据库的物理设计初步评价完成之后，就可以开始建立数据库了。数据库实施主要包括：用数据定义语言（DDL）定义数据库结构，组织数据入库，编制与调试应用程序，数据库试运行。

所谓使用 DDL 定义数据库结构就是使用数据库管理系统的建库命令，建立相应的用户数据库结构。组织数据入库就是将装载在其他介质上的数据输入到数据库中去。为了完成相应的操作和检索，需要编制很多程序，从而形成一个程序系统来使用该数据库，这部分是程序设计的任务。一切就绪之后，就可以试运行数据库了。这样的一个初始系统就可以运行了。

1.1.6 系统管理和维护

数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。数据库投入运行标志着开发任务的基本完成和维护工作的开始，但并不意味着设计过程的终结。由于应用环境在不断地变化，数据库在运行的过程中物理存储也会不断变化，对数据库的设计进行评价、调整、修改等维护工作是一项长期的任务，也是设计工作的继续和提高。

在数据库运行阶段，对数据库经常性的维护工作主要是由数据库管理员完成的，它包括以下 4 方面内容。

（1）数据库的备份和恢复。数据库的备份和恢复是系统正式运行之后最重要的维护工作之一。数据库管理员要针对不同的应用要求指定不同的备份计划，定期对数据库和日志文件进行备份，以保证一旦发生故障，能利用数据库备份和日志文件备份，尽快地将数据库恢复到某种一致性状态，并尽可能减少对数据库的破坏。

（2）数据库的安全性、完整性控制。数据库管理员必须对数据库安全性和完整性控制负责任。根据用户的实际需要授予不同的操作权限。此外，在数据库运行过程中，由于应用环境的变化，对安全性的要求也会发生变化，比如有些数据原来是机密，现在可以公开查询了，而新加入的数据又可能是机密的。这样系统中用户的级别也会相应变化。这些都需要数据库管理员根据实际情况修改原有的安全性控制。同样，由于应用环境的变化，数据库的完整性约束条件也会变化，也需要数据库管理员不断修正，以满足用户要求。

（3）数据库的性能监督、分析和改进。在数据库运行过程中，监督系统运行，对监测数据进行分析，改进系统性能是数据库管理员的又一项重要任务。目前许多数据库管理系统产品都提供了监测系统性能参数的工具，数据库管理员可利用这些工具方便地得到系统运行过程中一系列性能参数的值。数据库管理员应该仔细分析这些数据，判断当前系统是否处于最佳状态。如果不是，则需要通过调整某些参数来进一步改进数据库性能。

（4）数据库的重组织和重构造。数据库运行一段时间后，由于记录的不断增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存储效率，使数据库性能下降，这时数据库管理员就要对数据库进行重组织，或部分重组织、数据库的重组织不会改变原设计的数据逻辑结构和物理结构，只是按原设计要求重新安排存储位置，回收垃圾，减少指针链，提高系统性能。数据库管理系统一般都提供了供重组织数据库使用的实用程序，帮助数据库管理员重新组织数据库。

1.2 Access VBA 编程技术

1.2.1 VBA 概述

VBA 是 Microsoft Office 系列内置的编程语言，功能强大，而且是面向对象（OOP）的。如果用户具有 Visual Basic 基础，那么下面的学习将更加轻松。

同其他任何面向对象的编程语言一样，VBA 里也有对象、属性、方法和事件。对象就是代码和数据的组合，可将它看作单元，例如，表、窗体或文本框等都是对象。每个对象由类来定义。属性是对象特性，如大小、颜色和状态等。方法是指对象能执行的动作，如刷新等。事件是指对象可以辨认的动作，如单击鼠标或按下某键等，可以编写代码针对这些事件来做响应。

下面编写一个简单的 VBA 实例来实现文本显示和清除功能，设计步骤如下。

（1）新建一个窗体，放置两个按钮控件和一个文本框控件，按钮的名称分别定义为“com 显示”和“com 清除”，按钮的标题分别定义为“显示”和“清除”，文本框的名称定义为“txt 你好”，如图 1.3 所示。

（2）将光标放到窗体设计视图上，单击鼠标右键，在弹出的快捷菜单中选择“事件生成器”选项，弹出如图 1.4 所示的窗口。

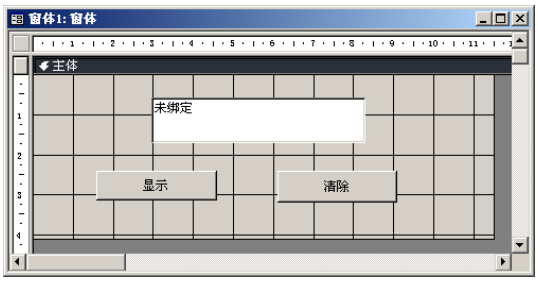


图 1.3　添加控件

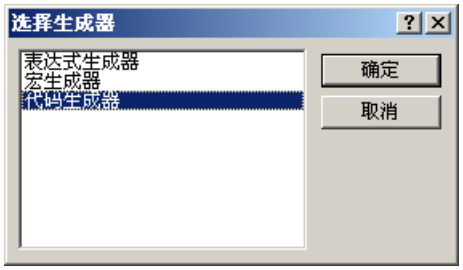


图 1.4　“选择生成器”窗口

（3）选择“代码生成器”，单击“确定”按钮弹出如图 1.5 所示的代码编辑窗口。

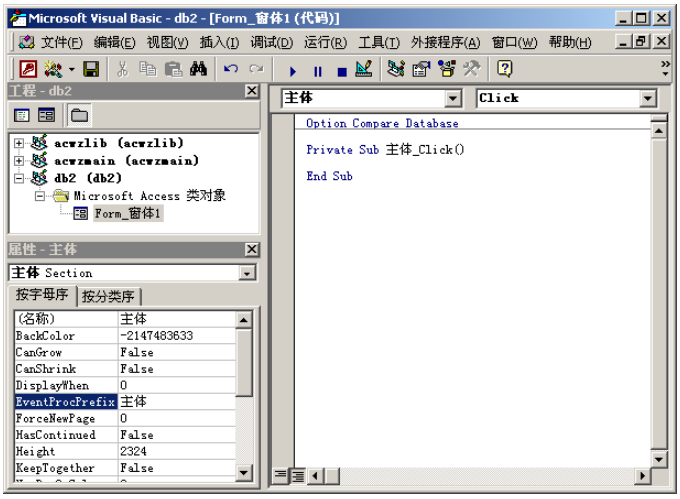


图 1.5　代码编辑窗口

（4）删除图 1.5 所示右侧窗口自动生成的代码，然后在其中写入以下代码：('后的文字为代码注释）

```
Option Compare Database
Private Sub com 清除_Click()          '单击清除按钮事件
    Me.txt 你好.SetFocus              'SetFous 方法，使文本框具有焦点
    Me.txt 你好.Text =""              '使文本框的 Text 属性为“”(空)
End Sub                               '结束过程
Private Sub com 显示_Click()          '单击显示按钮事件
    Me.txt 你好.SetFocus               'SetFous 方法，使文本框具有焦点
```

```
    Me.txt 你好.Text = "你好！欢迎学习 VBA"          '使文本框的 Text 属性为"你好…"
End Sub              '结束过程
Private Sub Form_Load()          '窗体加载事件
    Me.Caption ="欢迎！"          '设使窗体的 Caption(标题)属性为"欢迎"
End Sub
```

提示：程序中的“Me”表示该程序所属的对象，比如这里的“Me”就表示窗体对象。VBA 通常使用“Me”来设置对象中控件的属性或方法，设置语法如下：

```
Me.控件名.属性名（或方法名）
```

VBA 对象包括“com 显示”、“com 清除”按钮和“txt 你好”文本框，这 3 个 VBA 对象分别具有它们各自的属性、方法和事件，比如“com 显示”按钮的标题属性为“显示”，并且它有 Click 事件，从代码注释中可以看出，该按钮的 Click 事件是当单击该按钮时在“txt 你好”文本框中显示“你好！欢迎学习 VBA”。

（5）输入代码后，关闭代码编辑窗口，回到窗体设计视图，单击工具栏上的保存按钮，在弹出的“另存为”窗口中输入“欢迎!”作为窗体名称，如图 1.6 所示。

（6）单击“确定”按钮即保存窗体，然后单击工具栏上的窗体视图按钮运行窗体，单击“显示”按钮，结果如图 1.7 所示，单击“清除”按钮，则文本框中的文字消失。

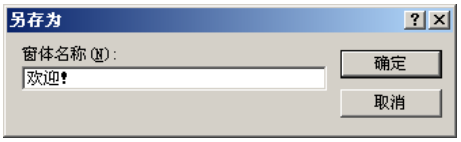


图 1.6　“另存为”窗口

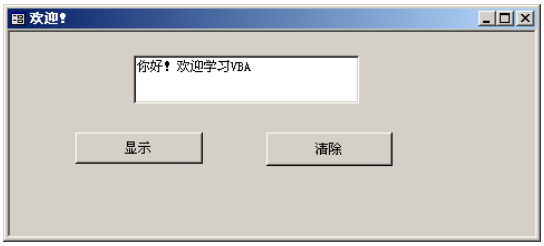


图 1.7　简单示例

1.2.2　VBA 语法基础

1．程序的书写格式

（1）注释语句

好的程序一般都有注释语句，可以方便程序的维护工作。即使是程序员自己，如果程序中没有注释，在一段时间以后，要读懂自己的程序，也并非易事。

在 VBA 程序中，注释可以通过以下两种方式实现：

①使用 Rem 语句，格式如下：

```
Rem 注释语句
```

②用“'”号，格式如下：

```
'注释语句
```

提示：Rem 注释在语句之后时要用冒号隔开。'注释无需使用冒号。

下面举例说明注释的用法，代码如下：

```
Rem 声明两个变量
Dim MyStr1,MyStr2
```


需要全本请访问 www.ertongbook.com

```
MyStr1 = "Hello"    :Rem 注释在语句之后要用冒号隔开。
MyStr2 = "Goodbye"  '这也是一条注释；无需使用冒号。
```

（2）连写和换行

程序语句一般一句一行，但如果需要在一行中写多句代码时，可以使用“：”来分隔不同的语句，例如：

```
Dim MyName As String
MyName="比尔盖茨"
```

可以写成下面一行：

```
Dim MyName As String ：MyName="比尔盖茨"
```

有时一句代码太长，书写不方便，可以将一句代码分开写成几行。此时要在每行代码末尾加空格和下划线“_”，例如：

```
Dim MyStr1,MyStr2,MyStr3,MyStr4,MyStr5,MyStr6,MyStr7,MyStr8,MyStr 9,MyStr10
```

可以分成两行：

```
Dim MyStr1,MyStr2,MyStr3,MyStr4,MyStr5, _
    MyStr6,MyStr7,MyStr8,MyStr 9,MyStr10
```

2．变量和常量

（1）声明常量和变量

变量用于临时存储数值、计算结果或数据库中的任意对象，其声明方法如下：

```
Dim 变量名 As 数据类型
```

变量的数据类型如表 1.1 所示。

表 1.1　　变量的数据类型

数据类型	存储空间大小	范围
Byte	1 字节	0～255
Boolean	2 字节	True 或 False
Integer	2 字节	-32 768 ～ 32 767
Long(长整型)	4 字节	-2 147 483 648 ～ 2 147 483 647
Single(单精度浮点型)	4 字节	负数时从 -3.402823E38 ～ -1.401298E-45；正数时从 1.401298E-45 ～ 3.402823E38
Double (双精度浮点型)	8 字节	负数时从-1.79769313486231E308～ -4.94065645841247E-324；正数时从 4.94065645841247E-324 ～ 1.79769313486232E308
Currency(变比整型)	8 字节	-9 223 372 036 854 775 808～9 223 372 036 854 775 807
Decimal	14 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 +/-7.9228162514264337593543950335；最小的非零值为 +/-0.0000000000000000000000000001
Date	8 字节	100 年 1 月 1 日到 9999 年 12 月 31 日
String(变长)	10 字节	0 到 20 亿
String(定长)	字符串	长度 1～65 400
Variant(数字)	16 字节	任何数字值，最大可达 Double 的范围

续表

数据类型	存储空间大小	范围
Variant(字符)	22 字节	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

举例如下：

```
Dim MyName As String          '声明一个名为 MyName 的字符串变量
MyName="盖茨"                 '给变量赋值
MyName="乔布斯"               '改变量的值
```

常量就是在应用程序的运行中值不能改变的对象，如 Null、True 和 False 等。

声明常量需要使用 Const 语句：

```
Public Const 表达式
```

例如，声明一个在所有模块中都使用的常量 PI，代码如下：

```
Public Const PI=3.1415926        '这个值不能像上面例子中的变量那样改变或赋予新值
```

（2）变量和常量的作用域

作用域决定了变量或常量起作用的范围，分为过程内、单个模块内以及所有模块内 3 种。

通常用 Dim 声明作用域在过程内或单个模块内的变量，用 Const 声明作用域在过程内或单个模块内的常量。例如：

```
Dim MyName As String   '该变量只能被变量所在的过程或模块使用
Const PI=3.1415926       '常量 PI 只能被常量所在的过程或模块使用
```

通常用 Public Dim 声明作用域为所有过程或模块的变量，用 Public Const 声明作用域为所有过程或模块的常量。例如：

```
Public Dim MyName As String   '该变量可以在所有过程或模块中被引用
Public Const PI=3.1415926       '常量 PI 可以在所有过程模块中使用
```

（3）静态变量和非静态变量

使用 Dim 语句声明的变量，在过程结束之前，系统一直保存着它的值，但如果在过程之间调用时就会丢失数据，这种变量称为非静态变量。与之对应的是静态变量，可以使用 Static 语句声明静态变量，使用 Static 声明的变量在模块内一直保留其值，直到模块被复位或重新启动。即便是在非静态过程中，用 Static 语句来显式声明只在过程中可见的变量，其存活期也与定义了该过程的模块的存活期一样长。

Static 语句的语法与 Dim 相同，只是将 Dim 关键字换为 Static 而已。下面的语句声明了一个静态变量 MySex。

```
Static MySex As Boolean
```

清除过程中的静态变量的方法是：选择“运行”菜单下的“重新设置”命令。也可以用 Static 关键字声明函数和子程序，以便在模块的生存期内保留函数和子程序内的所有局部变量。

（4）数组

数组是一系列数据类型相同的元素的有序集合。数组可以存储很多值，而变量只能存储一个值。数组的声明如下：

```
Dim 数组名([n],[1 To m,1 To k]) As 数据类型
```

其中 n 表示数组的大小，而 1 To m,1 To k 表示声明的是一个 m×k 的二维数组。

数组的声明方式和变量是一样的，它可以使用 Dim、Static、Private 或 Public 语句来声明。

下面的语句声明了 3 个数组，其中 Array1 是大小为 10 的数组，Array2 是一个 10×20 的二维数组，而 Array3 则是动态数组，即数据大小不定。

```
Dim Array1(10) As String
Dim Array2(1 To 10, 1 To 20) As String
Dim Array3() As String
```

在使用数组变量的某个值时，只需引用该数组名并在其后的括号中赋以相应的索引即可，如 Array1(5)。

（5）表达式

表达式用来求取一定运算的结果，由变量、常量、函数、运算符和括号构成。VBA 中包含丰富的运算符，有算术运算符、关系运算符、逻辑运算符和连接运算符。各种运算符及其描述如表 1.2～8-5 所示。

表 1.2　　**算术运算符**

符号	描述	符号	描述
^	求幂	\	整除
-	负号	Mod	求余
*	乘	+	加
/	除	-	减

表 1.3　　**比较运算符**

符号	描述	符号	描述
=	等于	<=	小于等于
<>	不等于	>=	大于等于
<	小于	Is	对象引用比较
>	大于		

表 1.4　　**逻辑运算符**

符号	描述	符号	描述
Not	逻辑非	Xor	逻辑异或
And	逻辑与	Eqv	逻辑等价
Or	逻辑或	Imp	逻辑隐含

表 1.5　　**连接运算符**

符号	描述
＋或&	字符串连接

表达式是由各种运算符将变量、常量和函数连接起来构成的，但在表达式的书写过程中要注意运算符不能相邻，乘号不能省略，括号必须成对出现。包含多种运算符的表达式将按运算符优先级进行计算。

各种运算符的优先级顺序为从函数运算符、算术运输符、连接运算符、关系运算符到逻辑运算符逐渐降低。如果在运算表达式中出现了括号，则先执行括号内的运算，在括号内部，仍按运算符的优先顺序计算。

3．程序结构

（1）条件判断结构

VBA 条件判断结构共有两类：If 语句和 Select Case 语句。If 语句先计算逻辑表达式，然后根据表达式的值是 True 还是 False 来决定下一步执行的语句块；Select Case 语句可同时处理多种情况。

①If 语句。If 语句根据测试条件的结果作出不同反应。If 条件语句有 3 种语句形式。

a. If…Then

If…Then 语句又有两种形式，分别为单行形式和多行形式。

单行形式的语法为：

```
If 条件 Then 语句
```

其中，“条件”是数值或字符串表达式。若“条件”为 True，则执行 Then 后面的语句。“语句”可以是多个语句，但多个语句要写在一行。例如：

```
If I> 10 Then A = A + I   I=0   B=2*A
```

多行形式的语法为：

```
If 条件 Then
   语句
End If
```

多行形式使用 End If 作为结束标志，对于执行多条不方便写在同一行的语句时，使用这种形式会使代码整齐美观。例如上面的条件句可以写成：

```
If I> 10 Then
   A = A + I
   I=0
   B=2*A
End If
```

b. If...Then...Else

在两种条件中选择一种来执行，可使用 If...Then...Else。语法格式如下：

```
If 条件 Then
   语句
Else
   语句
End If
```

若“条件”为 True，则执行 Then 后面的语句；否则，执行 Else 后面的语句。例如，如果 UpdateFlag 的值为 True 则显示消息为“Update Successfully”，否则显示信息为“Failed!”，代码如下：

```
If UpdateFlag Then
   MsgBox "Update Successfully"
Else
   MsgBox "Failed"
End If
```

c. If…Then…ElseIf…Else

从 3 种或 3 种以上的条件中选择一种，可使用 If…Then…ElseIf…Else。语法格式为：

```
IF  条件 1 Then
   语句 1
ElseIf  条件 2 Then
   语句 2
…
[ElseIf  条件 N Then
   语句 N]
Else
   语句 N+1
End If
```

若“条件 1”为 True，则执行 Then 后的语句 1；否则，判断“条件 2”，若为 True 时，执行语句 2，依次类推，当所有的条件都不满足时，执行 Else 块的语句 N+1。

例如，下面的代码通过对销售额进行判断，给出雇员的评价和佣金。

```
If Sales>15000 Then                          '若销售额大于 15000
   Commission=Sales*0.08
   Rating="Excellent"
ElseIf Salses>12000 And Salses<=15000 Then       '若销售额介于 15000 和 12000 之间
   Commission=Sales*0.06
   Rating="Good"
ElseIf Salses>8000 And Salses<=12000 Then        '若销售额介于 8000 和 12000 之间
   Commission=Sales*0.05
   Rating="Adequate"
Else                                             '若销售额小于 8000
   Commission=Sales*0.04
   Rating="Need Improvement"
End If
```

②Select Case 语句。从上面的例子可以看出，如果条件复杂，分支太多，使用 If 语句就会显得很烦杂，而且程序不易阅读。使用 Select Case 语句可写出结构清晰的程序。

Select Case 语句可根据表达式的求值结果，选择其中的一个分支语句执行，其语法如下：

```
Select Case  表达式
Case  表达式 1
   语句
Case  表达式 2
   语句
…
Case  表达式 3
   语句
```

```
Case Else
   语句
End Select
```

Select Case 后的表达式是必要参数，可以为任何数值表达式或字符串表达式；每个 Case 后的表达式，是多个“比较元素”的列表，其中可包含“表达式”、“表达式 To 表达式”或“Is<比较操作符>表达式”几种形式。每个 Case 后的语句都可包含一条或多条语句。

在程序执行时，如果有一个以上的 Case 子句与“检验表达式”匹配，VBA 也只执行第一个匹配的 Case 后面的“语句”。如果前面的 Case 子句与检验表达式都不匹配，则执行 Case Else 子句中的“语句”。

使用 Select Case 改写上例，代码如下：

```
Select Case Sales
Case Is>15000
   Commission=Sales*0.08
   Rating="Excellent"
Case 12000 To 15000
   Commission=Sales*0.06
   Rating="Good"
Case 8000 To 12000
   Commission=Sales*0.05
   Rating="Adequate"
Case Else
   Commission=Sales*0.04
   Rating="Need Improvement"
End Select
```

（2）循环结构

VBA 提供了循环语句来处理重复的操作，循环语句有 5 种形式：Do 循环、While 循环、For…Next 循环、With 循环和 GoTo。下面将分别介绍这 5 种循环语句，同时也介绍与循环有关的语句 Exit。

①Do 循环。Do 循环可用在重复某些操作直到满足给定条件才终止操作的情况。Do 循环的语法如下：

```
Do[{While | Until} 条件]
   [语句]
[Exit Do]
   [语句]
Loop
```

或

```
Do
   [语句]
[Exit Do]
```


需要全本请访问 www.ertongbook.com