

Access 2003



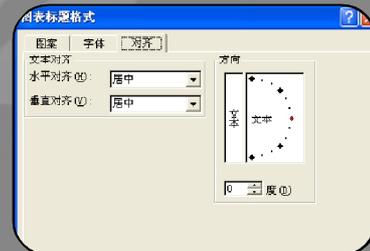
# part

# 14

## Access报表高级应用

- ◆ 报表设计视图组成
- ◆ 修改图表
- ◆ 建立折叠栏报表
- ◆ 建立邮件归并报表
- ◆ 应用计算和汇总

用户在真正使用Access时，会发觉对所创建报表常常需要进行一定调整，用于弥补向导功能的不足或添加某些新功能，这些将是本章重点介绍的内容。



本章将讨论如何在“设计”视图中使用报表；如何在“设计”视图对报表的图表进行必要的调整；在报表中如何利用窗体完成一定的操作；学习邮件归并报表的使用和与 Word 进行合并的操作；最后，将对报表中的字段设计计算和汇总格式。

## 1 报表设计视图组成

在开始报表的设计之前，必须了解报表的几个基本组成部分。报表就是利用其不同的组成部分，将从数据库数据中所提取的信息，有机地展现在用户的面前。对报表所进行的设计也就是围绕着这些组成部分中的控件进行调整。

### 1.1 报表的节

可以在报表的“设计”视图中清楚地看到报表的每一组成部分——节，报表中的内容是以节作为单位划分的。所有的节在“设计”视图中水平方向无限伸展。报表由五个节组成，分别是：报表页眉、报表页脚、页面页眉、页面页脚和报表主体，其中报表主体还包括标头内容和注脚。上述每一个节都有其特定的目的，而且按照一定的顺序打印在页面及报表上，节的具体位置和作用如表14-1所示。



表 14-1 报表中节的具体位置和其作用

报表中位置		作用
报表页眉	只出现在报表开始的位置	在报表中显示标题、徽标、图片以及其他报表的标识物
页面页眉	位于每页的最上部	显示字段标题、页号、日期和时间
报 表 主 体	组标头	位于每个字段组的开始处，需对字段排序或分组的地方 显示标题和总结性的文字
	主体内容	每个有下划线的记录都有相对应的详细内容 显示记录的详细内容
	组注脚	位于每个字段的末端，报表主体之后和页脚之前 显示计算和汇总信息
页面页脚	位于每页的最下部	可以是日期、页号、报表标题或其他信息
报表页脚	只出现在报表结束位置	总结性文字



在“设计”视图中，节代表各个不同的带区，而且报表所包含的每1节一般只能被指定一次。在打印报表中，某些节可以指定很多次。可以通过放置控件来确定在每1节中显示内容的位置，例如标签和文本框。通过对使用相同数值的记录进行分组，可以进行一些计算或简化报表使其易于阅读。在此报表中，相同年份的销售情况将分在同一个组中。报表的最后的输出的结果如右图所示。

学号	综合成绩	姓名	性别	课程学	成绩
20022001					
平均分 = 20022001 (新增记录)					
平均分					
20022002					
平均分					

在默认情况下，通过“新建报表”对话框中“设计视图”选项新建一个报表，系统会在新报表中自动包含有页眉、页脚和主体这3个节。下面将对报表中各个节的详细内容加以介绍。

## 报表的页眉

在报表页眉中的任何内容只能在报表的开端处出现一次，报表页眉节上面为报表页眉栏，下面为页眉，它出现在报表窗口的开端。报表页眉内存放的数据出现在报表的开始部分，即第一页的页眉之前。一个典型的报表页眉包括报表标题，还可以添加日期和公司标识等控件。

由于报表页眉只能在报表中出现一次，因此，利用分页控件的功能可以将报表中的报表页眉单独作成是一个报表的开始或封面，把报表页脚作成是一个总结或结束页。

报表页眉的主要作用是便于用户搞清楚所浏览数据信息的主体，因此，需要利用某些特殊功能，例如在报表页眉中添加一些线条、方框、阴影和颜色等，对报表页眉进行装饰，以突出报表的主题。

如果要在设计视图中显示或隐藏报表页眉，可以在“视图”菜单中选择或取消选择“报表页眉/页脚”命令即可。

## 页面页眉

页面页眉中的文本或字段控件等内容通常出现在报表的每页顶端，位于页眉栏和报表主体栏中间。一个典型的页眉包括页数、报表标题或字段标签等。例如，大家比较熟悉的汇总报表中的起栏目标题头。同样，可以利用某些特殊效果对页眉进行处理，突出其包含的内容。

如果要在设计视图中显示或隐藏页面页眉，可以在“视图”菜单中选择或取消选择“页面页眉/页脚”命令即可。

## 报表的组标头

组标头通常用一个特殊的标签来标识，表明报表主体中显示出来的字段内容。

当报表主体使用Access报表的排序和分组特性的时候，Access就会为报表中每个字段添加一个页眉字段标签，按照这些字段对记录进行分组。在页面页眉和页面页脚之间是报表主体，即报表的详细信息段。主体中显示每条记录。

组标头可以有多个，利用多级可以对显示的主要信息内容进行细分。可以利用组标头属性表对组标头的显示进行设定。



## 报表的主体

报表主体中显示在当前表或查询中的每条记录的详细信息。可以利用计算字段对每行数据进行某种计算，例如，利用产品数量乘以单价算出总计金额。

通过报表设计窗口的“可见性”属性来设置是否需要显示报表中的某个字段。通过关闭主体的显示，就能够显示一个不带主体或只显示某些组的总计报表。

## 报表的组注脚

使用组注脚可以对报表主体中分组之后的所有记录进行统计。每次分组改变时，这种类型的字段会自动复位为0。

对于每个组注脚中文本框的属性对话框中，都有一个“运行总和”属性，可以利用该属性改变计算总计的方法。

## 页面页脚

页面页脚通常包含有页码或控件总计。在一个很大的报表中，可能报表主体中的记录很多，使得在每页中没有一个统计数字，这就需要在报表设计中既包含页码总数又有报表主体中被分组的记录总数。

一般来说，在显示页面页脚时，都是采用将文本控件Page与表达式Page结合在一起打印出页码，页码文本框中的内容一般为：“第 '&[Page]&' 页”。

## 报表的页脚

报表页脚位于一个报表设计视图中的最底部，是在所有记录数据和报表主体都输出之后打印在报表的结束处。在输出时和报表页眉一样，只出现一次。

典型的报表页脚显示所有记录的汇总结果，例如记录的总计数、平均值和百分率等。

当在报表页脚中添加了输出的内容，最后一页中的页面页脚会在报表页脚的输出内容之后输出。

# 2

## 报表和图表

Access

在报表中添加和编辑控件的方法和在窗体中的操作完全一样。在本小节中将介绍有关在报表中添加或编辑控件的操作，以及如何修改报表中的图表。如果用户在使用窗体时也添加了图表，对它的编辑可以根据本节所介绍的内容进行操作。

其实，如果在 Access 内部建立了一个图表，则需要通过使用图表应用程序中的工具进行编辑，修改某些特性以增强图表的显示效果。下面将通过结合图表程序介绍修改图表的过程。

对图表的修改操作主要包括以下几个方面的内容。

- 修改图表类型以便以不同的图表类型显示数据。
- 操作三维视图以便使显示更为精确。



- 加入网格线以便更好地反映数据显示。
- 在图表中添加或修改数据。
- 加入位图到图表中，使其显示更为专业化。
- 给图表输入自由格式文本，并使用特殊效果进行强调。
- 移动并修改插图说明。
- 用线条和箭头注释图表。
- 用颜色和图案修改某些图形对象。

在介绍图表的编辑操作之前，读者需要先熟悉图表窗口。

## 熟悉图表窗口

当编辑报表中的图表时，首先需要双击鼠标激活图表应用程序。在图表应用程序中实际上还包括两部分，它们分别是：数据表对话框，与图表中显示内容有关的数据工作表；图表界面，显示与数据工作表相对应的图表。在数据表对话框中可以加入、修改或删除数据。在数据表对话框中所进行的任何修改，都会反映在图表界面中，这就使得用户可以根据实际中的需要调整数据表中数值大小。通过缩放图表界面的大小，可以修改图表的视图，便于根据需要更好地观察窗口中的内容。



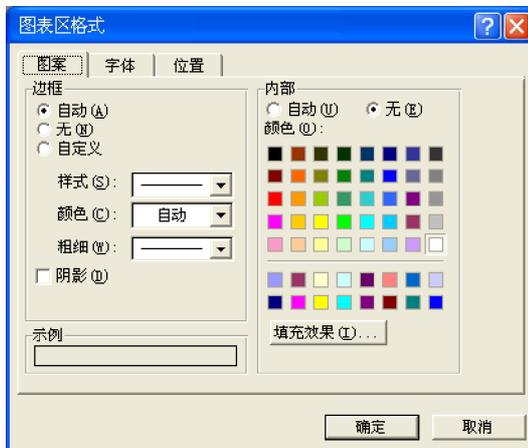
### 提示

在图表应用程序中，以及Access报表的“设计”视图中所显示的图表都是系统默认的一个图表，它的特点是图形的大小、是否包含插图说明等图表属性与实际输出的图表相同，但是图表中的数据，即坐标轴所代表的的数据，是与最后输出不一样的。因而可以将它视为一个图表的图例。

## 图表组件

在图表界面中的图表区显示的是由一些组件有机地结合在一起之后构成的一个图表。在图表区中包括如下基本组件。图表标题；背景墙；坐标轴；数据系列；网格线；插图说明——图例。

对于每个图表中的组件都有一个格式属性设置对话框，可以利用设置组件的属性来改变一个图表的输出显示效果。但是图表区作为一个整体，系统同样为它提供了一个格式设置的对话框，在该对话框中包括3个选项卡：图案、字体和位置。在“图案”选项卡中，可以对图表



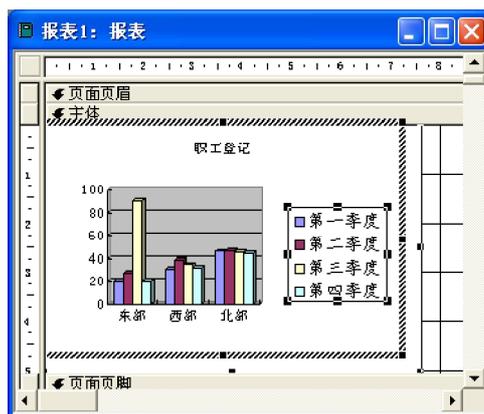
中坐标绘制区以外的空间进行颜色的填充, 以及设置图表的边框。选择颜色填充选项中的一种颜色, 单击“确定”按钮之后就可以看到在坐标绘制区以外的图表空间填充了选取的颜色。

### 提示

在此处对字体的设置是对于图表中所有文本都有效的, 在后面会发现在图表其他组件的格式设置对话框中也会包含有对字体设置的选项卡, 但是那些设置只是针对于图表某一组件中的文本。

单击“图案”选项卡中的“填充效果”按钮, 还可以对所进行的填充增加一些特殊的效果。单击“填充效果”按钮之后, 会出现“填充效果”对话框, 在该对话框中包括: 过渡、纹理、图案和图片 4 个选项卡。

如果用户想知道在编辑图表时 Access 报表“设计”视图界面, 就需要在窗口任务栏中选取 Access 应用程序, 当打开 Access 窗口时, 可以看到此时报表中的图表上已被一组斜线所覆盖, 上面对图表中非坐标区所进行的颜色填充的编辑操作结果, 在原图表中也同样发生了变化。因此, 可以看到在图表程序中, 对图表所做的编辑操作会马上在原图表中反映出来。



下面就对图表中的几个组件分别加以介绍。

### 1. 图表标题

在一个报表的图表中往往包括一个图表标题, 主要对图表的内容加以说明。它通常位于图表的正中间的顶部, 由一组说明性文本组成。通过用鼠标拖移, 可以将它放置在图表中的任意地方。也可以利用鼠标选中图表标题并对它的内容进行修改。

双击图表标题之后, 可以打开“图表标题格式”对话框。在该对话框中包括 3 个选项卡: 图案、字体和对齐。“图案”和“字体”选项卡中的内容与在“图表区格式”对话框中的“图案”和“字体”选项卡的内容一





样，只是它们所设置的范围有所不同。“对齐”选项卡中包括对文本对齐方式的设置。在其中的“文本对齐”选项区中，可以根据需要设置文本水平和垂直对齐的方式，每种对齐方式都可以分为5种。在“方向”选项区中，可以任意调整文本在图表中的方向，可以通过其中类似于钟表的列表框直接设置文本的方向，也可以在下面的文本框中输入一个准确的角度值。

## 2. 背景墙

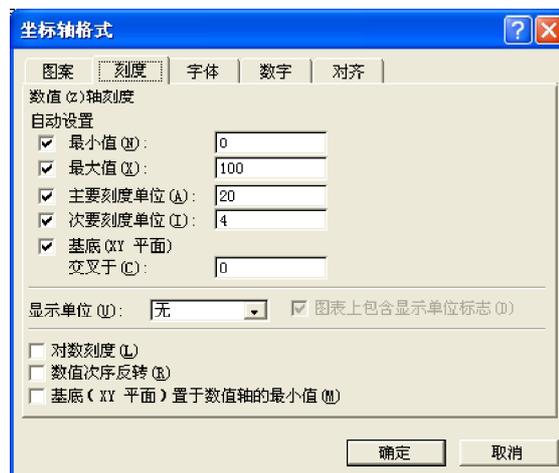
背景墙是图表的主要区域，图表的坐标轴均包括在其中。在背景墙中可以有网格线和表示数据的图形。双击坐标轴包括区的空白处，则会弹出“背景墙格式”对话框，在该对话框中只包含“图案”选项卡。



## 3. 坐标轴

坐标轴的X轴和Y轴分别代表一组对数据的分类类别和数据数值的大小，具体X轴和Y轴代表什么要看用户的设置。当坐标轴代表分类类别时，则坐标轴又称为分类轴；当坐标轴代表数据数值的大小时，则坐标轴又称为数据轴。

当用户双击坐标轴时，会出现“坐标轴格式”对话框，在该对话框中包括5个选项卡：图案、刻度、字体、数字和对齐。在“图案”选项卡中，可以看到其中没有了颜色调色板，取而代之的是对坐标轴的刻度线的设置。设置选项包括主、次刻度线类型和刻度线标签，可以根据需要选取刻度线样式。在“刻度”选项卡中，可以看到对坐标轴代表的数值大小、刻度线单位以及坐标轴的位置等进行设置。设置坐标轴，可以起到美化图表的作用。在“数字”选项卡中，可以对坐标轴上的数字进行设置。其中包括11种数字类型选项，还有一个对数字类型进行自定义的选项。



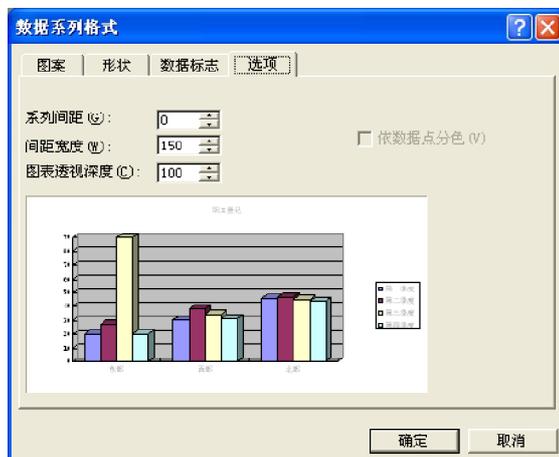
### 提示

“坐标轴格式”对话框中的选项与用户选取的X轴或Y轴有关。用户可以分别选取坐标轴来查看它们的区别，主要区别在“刻度”和“对齐”选项卡中。

#### 4. 数据系列

数据系列在图表中以一定的图形代表数据表中的数字，位于坐标系的空间内。数据系列的图形样式与图表类型有关。

直接双击图表中任一数据系列图形之后，可以打开“数据系列格式”对话框，该对话框中包括5个选项卡：图案、坐标轴、误差线Y、数据标签和选项。除了“图案”选项卡之外，其余都是前面格式对话框中所没有的。“选项”选项卡中的“重叠比例”选项是比较有用的，当用户对图表中数据的分类比较多时，并且如果数据系列图采用条形时，图表看起来就会比较拥挤。这时可以通过增加“重叠比例”文本框中的数值，使得数据系列图形之间存在一个叠加，节省出坐标系的一定空间；或者改变“分类间距”文本框中的数值，也可以改变图表中数据系列图形的分布。



#### 提示

利用该对话框的“图案”中的选项对数据系列图形的颜色进行设置时，是对某一个分类系列的颜色进行设置。例如，如果双击设置图表中的第一个条形图形时，就会改变图表中每组按照分类轴的第一个图形的颜色，即所有表示第一季度数据的图形颜色都会改变。同时，原图表中相应位置的图形也同样会发生改变。

另外，用户连续两次单击鼠标左键之后，就可以选取某个确定的数据系列图形。这时如果再双击鼠标左键之后就会出现一个“数据点格式”对话框，在对话框中所进行的设置都是关于这个数据系列图形的。

#### 5. 网格线

网格线在图表中的作用主要是便于用户观看数据信息，利用与坐标轴平行的直线，可以方便地了解数据系列图形所代表的数据的数值大小。一般利用向导创建的图表中，系统默认在图表中显示与分类轴相平行的直线，用户可以在“网格线格式”对话框中对其进行设置。在图表中网格线的位置处双击鼠标左键，会打开“网格线格式”对话框。在该对话框中，有两个选项卡：图案和刻度。在“图案”选项卡中只可对线条进行设置，而不能对颜色进行设置；在“刻度”选项卡中，包括有与“坐标轴格式”对话框的“刻度”选项卡中相同的选项。

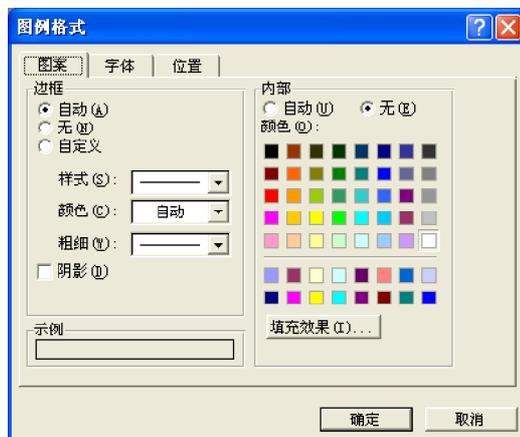




## 6. 图例

图例在图表中所起的作用就是对图表中的数据系列图形进行说明，因为用户在绘制图表时，往往是采用一个或两个字段作为数据分类的标准，这就需要在图表中利用图例加以说明，说明图表中各种图形各代表什么意义。

在图表中直接双击图例组件，会打开“图例格式”对话框，在该对话框中包括有3个选项卡：图案、字体和位置，其中“图案”和“字体”选项卡的作用在前面已经介绍了。在“位置”选项卡中，可以对图例所放置的位置进行设置，共有靠下、右上角、靠上、靠右和靠左5个位置供用户选择。



### 提示

在“图例格式”对话框中对图例在图表中所处位置进行得设置不够直观，用户可以利用在后面介绍的“图表选项”对话框的“图例”选项卡对图例进行设置，因为在该对话框中有一个演示窗口。不过对于熟悉图表布局的用户来说，还是可以直接利用“图例格式”对话框进行设置。



## 修改图表类型

实际上，数据系列的图形样式与图表类型有着极为密切的关系，数据系列的图形样式在很大程度上影响着—个图表的类型。

对图表类型的选取对于新建的图表是必须的，往往在创建一个图表之前，不会对一组数据在绘制成图形表之后的显示加以预测。因此在创建完一个图表后，仍需对图表类型进行修改。

图表程序为用户提供了多种图形，包括二维和三维图表。具体的图形有：柱形图、条形图、折线图、饼图、XY散点图、雷达图、气泡图、面积图和圆环图等。

通过在“图表”菜单中选择“图表类型”命令，打开“图表类型”对话框。在该对话框中包括两个选项卡：标准类型和自定义类型。在每个选项卡之中，用户都可以查看改变图表类型之后的效果图。在“标准类型”选项卡中有两个区域：图表类型和子图表类型。在“图表类型”列表框中列出了一些基本的图表类型，当选取一种基本类型之后，在“子图表类型”选项区中会有几种与之相应的子类型。单击“按下不放可查看示例”按钮，就可以显示图表的效果图。



## 提示

在“自定义类型”选项卡中所包含的图表类型只是相对于以往标准类型所说的，它们都是系统为用户提供的图表类型。

## 3

## 折叠栏报表

Access

Access 为用户提供了一种特殊的报表——折叠栏报表，它允许报表在输出时，像邮件标签输出的结果一样，不是在单列上输出，而是根据输出段落的宽窄，将报表以多栏的形式打印输出。这种方法可以为用户节省纸张。

该报表可用于电话簿、报纸或其他期刊的设计。例如，可以事先将所有的公司职员联系方法在计算机中按照一定顺序建立折叠栏报表，然后通过打印机将它打印输出，这样当需要与某个职员进行联系时，就可以方便快速地在报表中找到。

折叠栏报表与邮件标签的区别关键在使用的目的不同，折叠栏报表的使用目的是用来查阅大量相同类型数据中的某个数据；而邮件标签的目的是将数据分开输出，以便分别使用。折叠栏报表中通常带有分组的段、页面标题和注脚等；而在邮件标签中一般只有报表主体这一节。

建立折叠栏报表的一般过程如下。

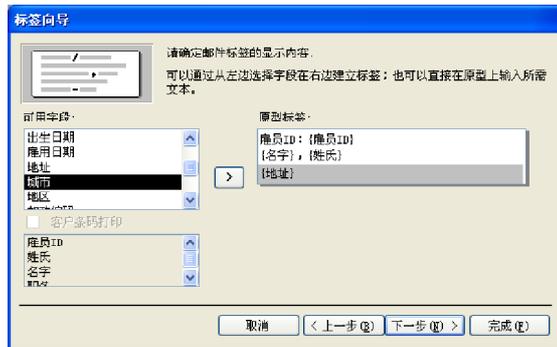
- (1) 决定要将数据以多少栏显示。
- (2) 建立一个报表，以及它的报表主体和相关信息输出。
- (3) 在页面设置对话框中设置合适的选项。
- (4) 打印预览报表的输出。

### 建立折叠栏报表

建立折叠栏报表所使用的方法与建立其他类型报表的方法相同。用户可以采用两种方法来建立一个折叠栏报表：一是利用报表的设计视图进行创建，然后在页面设置中对打印输出方式进行设置，将报表的列数设置为所需的栏数；二是在Access的标签向导基础之上完成折叠栏报表的主体创建，然后在报表中添加相应的报表标题和其他控件。

两种建立折叠栏报表的方法所建立的结果基本相同，在这里就以利用标签向导来建立折叠栏报表为例来进行说明，折叠栏报表的显示内容为“罗斯文商贸”公司中职员基本情况信息，具体操作步骤如下。

- 1 利用Access中的标签向导建立一个大概的报表主体。在向导中设置邮件标签显示内容时，利用向导所提供的功能输入一个基本框架，并输入一定的文本格式和相应的字段。





- 2 在报表的“设计”视图中，可在生成的标签中添加报表的页面页眉和页脚组件。在页面页眉中可以输入报表的标题和其他说明文本，以及功能控件，例如，日期时间的显示等，在页面页脚中添加页码和统计计算等。本示例中将在步骤1创建的标签中的报表的页面页眉中添加一个文本框，在文本框中输入“罗斯文商贸公司职员基本情况报表”，另外添加一个显示目前日期的控件，设定日期的显示格式。在报表的页面页脚中添加一个显示页码的控件。



## 打印折叠栏报表

在打印折叠栏报表之前，必须利用“页面设置”对话框中的各选项对报表的页面进行设置。在该对话框中可以设置打印的列数、行数以及具体每段输出内容的尺寸大小等属性。如果不对报表在打印之前进行页面设置，可能会造成错误的输出结果。

与在报表“设计”视图中直接建立折叠栏报表相比，按照上面所介绍的操作创建折叠栏报表可以省去对报表页面尺寸的设置，在标签向导中系统会根据对标签纸的选择自动设置页面尺寸。

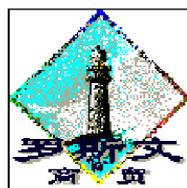
另外，需要提醒用户的是，在页面设置中有一个对报表打印方向的设置选项，打印方向分为先行后列和先列后行这两种，可以根据查阅数据的习惯加以选择。

## 4

## 邮件归并报表

Access

在 Access 中还有一类报表是邮件归并报表，该报表的特点是将某些数据库中的字段嵌入到大量文本之中。例如，一封致全体职员的信件，在该信件中如职员姓名、地址等内容都是存放在数据库同样的字段中。本节将介绍如何在 Access 中完成一个邮件归并报表的创建。一般在创建邮件归并报表时，需要明确报表中所有将要包含的数据字段，根据要求建立相应的数据表，即完成数据的汇总工作。之后，便可以进行一个邮件归并报表的创建了。下面介绍两种创建方法。



Nancy, Davolio (女士)  
507 - 20th Ave. E. Apt. 2A  
Seattle WA 98122

亲爱的 Nancy (女士)：

您在公司工作了不到五年的时间，我代表公司感谢您所完成的工作，希望今后能够再接再厉，发挥作用。

作为公司的一名员工，您的工作态度和所完成工作，将直接影响到公司今后的发展。为了公司的明天，为了我们将来生活的更加美好，让我们携起手来，与公司一起步入2000年。

## 标签向导

在建立邮件归并报表时，用户可以直接在报表“设计”视图中创建一个归并报表，但是这种创建方法比较烦琐。利用Access 的标签向导可以简化建立过程。

首先，在数据库窗口的“报表”界面中单击“新建”按钮，在弹出的“新建报表”对话框中选择数据源，并选择“标签向导”选项，单击“确定”按钮即可启动标签向导。在选择标签纸时，这里可能需要自己自定义一种规格的纸张页面，使得自定义的页面与信件的页面相同。然后，在标签布局设计中，输入信件中的文本。

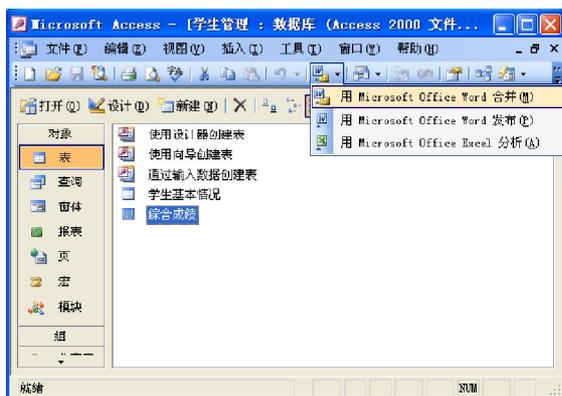
接着，在报表“设计”视图对报表进行编辑，添加公司的标志或其他说明文本，以及一些对信件有用的控件，例如信件打印日期、时间和页码等。

利用该方法创建邮件归并报表时，可以在报表中添加表达式。表达式可以用来显示一个文本，也可以定义某种字段的输出，还可以对某些数据进行计算输出。例如，在本示例中，对于信件中开始的称呼就可以建立一个表达式，在其中输入“= "亲爱的"& [名字] & " (" & [尊称] & ") : ”。

## Word 合并

Access具有“用Microsoft Word合并”的功能。利用它，可以建立一个邮件归并报表，使得输出的格式标准并且美观，真正符合字处理软件的要求，具体操作步骤如下。

- 1 在“数据库”窗口中选取报表内容中需要数据的来源——表或查询，单击工具栏上的“Office 链接”按钮旁的下拉箭头。
- 2 在弹出的下拉菜单中选择“用Microsoft Word合并”命令，将会弹出“Microsoft Word 邮件合并向导”对话框。





- 3 在“Microsoft Word 邮件合并向导”对话框中选择将要进行的操作，可以在步骤1中选取的表或查询中的数据与一个已有文档进行链接，或者是创建一个新的文档，然后链接数据到文档。在本示例中，选中“创建新文档并将数据与其链接”单选按钮，创建一个新文档。选择完毕之后，单击“确定”按钮即可打开Word程序。



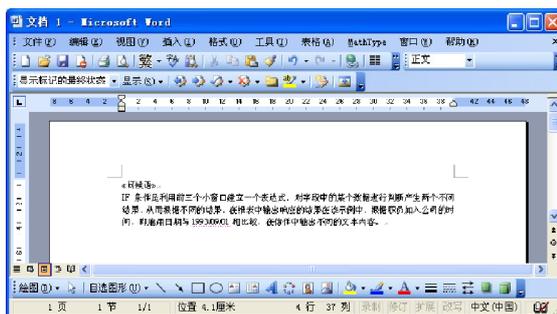
- 4 在Word界面中的“插入Word域”下拉菜单中可以选择添加所需的字段，即职员姓名、家庭住址和邮政编码等字段。在添加字段的时候，可以在所添加的字段之间输入某些连接符号。



- 5 在信件的正文中添加文本时，可以利用“插入Word域”下拉菜单中的命令在生成报表时对输出的文本进行控制。例如，在本设计示例中，带有阴影显示的文本就是利用“If...Then...Else”域对输出的文本进行控制。



- 6 按照上述过程创建了一个邮件归并报表之后，可以通过单击“合并到新文档”按钮来显示一份报表。利用“邮件合并”工具栏上的记录定位器可以选择输出相应的记录结果。下图所示的是一份邮件归并报表的输出结果。



### 提示

如果用户是利用打开邮件归并报表的主文档来启动Access程序，尽管用户可以在Access的数据库中进行其他操作，但是，当用户关闭主文档时，相应的Access程序也将被关闭。



## 5

## 应用计算和汇总

Access

在本章的前面几节中，对报表中的图表、折叠栏报表和邮件归并报表的建立和编辑做了介绍，其实在报表中常常还需要对数据进行汇总或其他计算。一般来说，用户在建立一个数据库时，通常只是将一些基本的，或者说是原始的数据添加到数据库的字段中，通过在报表中建立计算和汇总，可以对数据库中数据进行总体信息的输出。



## 建立计算字段

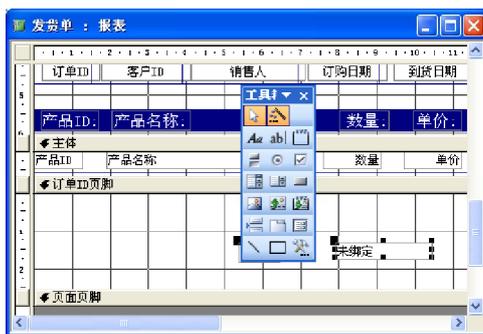
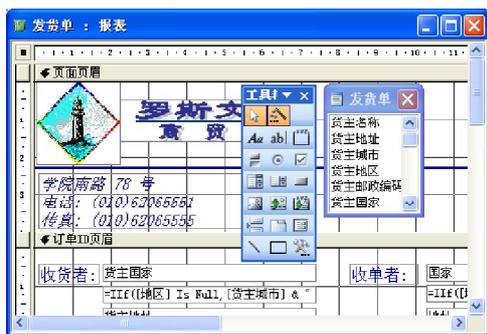
要在报表中显示汇总数据，就必须建立计算字段，利用计算字段计算所需的数据，并把它在计算控件中显示出来。

计算控件就是以表达式作为数据来源的控件，表达式可以使用表或查询中的字段数据。一般来说，具有“数据来源”属性的控件都可以作为一个计算控件，但是，通常都用非结合文本框作为计算控件。

建立计算控件的方法与在窗体中一样，现在就以建立发货单中的“合计”计算控件的设计为例，来说明计算控件的建立，具体操作步骤如下。

原始文件	Sample\Part14\罗斯文商贸.mdb
学习要点	在报表中建立计算字段

- 1 打开本书配套光盘中的Sample\Part14\罗斯文商贸.mdb文件，在“设计”视图中打开“发货单”报表。
- 2 在报表的“设计”视图中，利用工具箱中的“文本框”按钮在需要输出计算结果的设计页面中添加一个文本框。



- 3 在文本框处于选中的状态时，单击工具栏上的“属性”按钮，打开该控件的属性对话框。

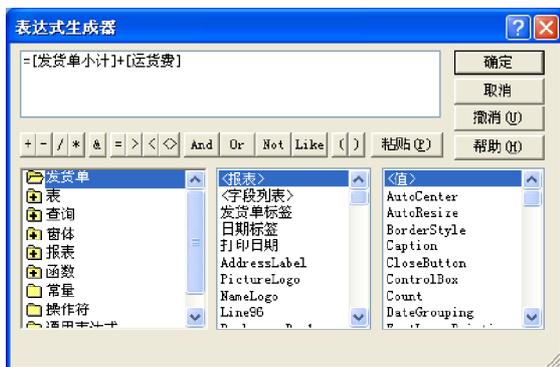




- 4 在属性对话框的“数据”选项卡中单击“控件来源”属性框右边的“生成器”按钮, 打开“表达式生成器”对话框。



- 5 在“表达式生成器”对话框中输入计算表达式, 例如, 计算客户订单中所有产品所花费的金额, 输入下列表达式: = [发货单小计]+[运货费]。



- 6 保存后运行报表, 结果如下图所示。



### 提示

如果计算控件是文本框, 则可以在“设计”视图中直接键入表达式, 但是每个表达式之前都要加上等号(=)运算符。在“表达式生成器”对话框中输入表达式时, 可以省略等号的输入。

## 对记录分组

在报表中对一组数据计算其汇总信息之前, 必须在数据库中存在一个记录的分组, 若没有分组将无法完成汇总计算。

组就是指由具有某种相同信息的记录组成的集合。将报表分组之后, 不仅同一类型的记录显示在一起, 而且还可以为每个组显示概要和汇总信息, 因而可以提高报表的可读性和易懂性。

在建立报表时, 可以利用数据库中不同类型的字段对记录进行分组。例如, 可以按照“日期/时间”字段进行分组, 也可以按“文本”、“数字”和“货币”字段分组, 但不能按“OLE对象”和“超级链接”字段分组。

当按不同字段分组时, 除了可以利用整个字段本身作为分组原则, 还可以指定分组字段的细节内容作为分组依据。例如, 在利用“日期/时间”字段进行分组时, 可以指定按照年、月、日等对记录进行分组, 将属于相同年份、月份和日期的记录归到同一组中。在利用“文本”字段

进行分组时，可以只取字段的前几个字符作为分组依据。

如果要在报表中对记录进行分组，首先要利用报表“设计”视图中的“排序与分组”命令，建立分组所依据的字段或表达式，并通过设置分组的组属性来实现报表的汇总功能。为了说明如何在报表中分组记录，下面以“发货单”报表为例来进行说明，具体操作步骤如下。

原始文件	Sample\Chapter14\罗斯文商贸.mdb
学习要点	在报表中对记录进行分组

- 1 打开本书配套光盘中的Sample\Chapter14\罗斯文商贸.mdb文件，在“设计”视图中打开“发货单”报表。



- 2 单击工具栏中的“排序与分组”按钮, 将会弹出“排序与分组”对话框。在本示例中将在“字段/表达式”选项中选择“订单ID”和“产品ID”字段，将“订单ID”字段设为分组依据，将“产品ID”字段设为排序字段。



## 汇总数据

创建报表的一个主要目的就是要根据数据库中的大量数据，提出一个汇总的信息。因此，在报表中常常在给出细节信息时，还要给出每组的汇总数据和整个报表的汇总数据。

具体地说，在报表中，计算汇总信息可以分为两类：一类是对每组进行汇总，另一类是对整个报表进行汇总。如果要计算每组的汇总信息，则在组标头中放置分组字段的同时，还要在组注脚中添加计算控件，根据每组中的细节信息计算相应的数据；如果要计算整个报表的汇总信息，则可以在报表页眉或报表页脚中建立计算控件，输出显示所需要的信息。下面就以计算组总计说明如何在报表中汇总数据。

计算组总计实际上就是计算同一个组中某一类型数据的总和。在报表中，利用组总计可以直接查看各组的数据，并对各组数据进行比较，这项功能对于用户分析数据来说是非常有用的。在“发货单”报表中，就是利用组总计对每份订单进行了一个统计，得到每份订单的产品价格之和与订单总价。设计组总计的过程如下。

- (1) 在报表“设计”视图中打开要计算组总计的报表。
- (2) 对报表进行分组，利用“排序与分组”对话框设定报表的分组字段，必须至少在设定时将一个分组字段或表达式组属性中的“组页脚”属性设为“是”。
- (3) 在报表中的组注脚中，根据需要添加相应的计算控件。例如，在“发货单”报表中的



“小计”计算字段的表达式就是：`=Sum ([总价])`。利用Sum函数，对订单中所有产品的价格进行了求和。

完成上述步骤之后，可以在“打印预览”视图中查看所建立的组总计计算控件的输出结果。这里需要提醒用户，除了可以使用Sum函数进行计算汇总信息，还可以使用其他系统提供的汇总函数，如Min函数（求最小值）、Max函数（求最大值）、Avg函数（求平均值）和Count函数（计算个数）等。在表达式中正确运用这些函数，可以增加报表的汇总功能。



Access 2003



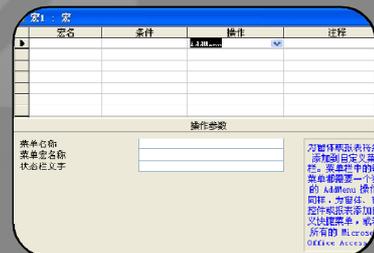
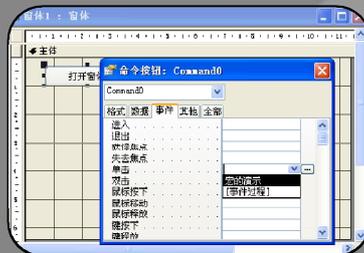
# part

# 15

## 宏设计

- ◆ 宏的创建和设计
- ◆ 宏的执行和调试
- ◆ 常用的宏

在Access 中，除了表、查询、窗体和报表之外，还有两个比较重要的对象——宏和模块。在本章将对详细介绍宏的相关使用方法，希望通过本章的学习，读者对Access 中的宏不再陌生，并能够熟练地利用宏的功能对数据库进行管理。



## 1 宏的概述

宏属于 Access 中的高级应用，很多初学者在学习到宏的时候往往选择将其放弃，原因是宏开始涉及到命令。如果读者要真正使用 Access 设计出功能丰富的系统，宏和下一章将要介绍的模块是必须掌握的。只有将两者结合应用才能不受 Access 中的常用向导及视图功能的限制。

宏是 Access 中执行特定任务的操作或操作集合，其中的每个操作均实现特定的功能，是由 Access 本身提供的。宏可以是包含操作序列的一个宏，也可以是由多个宏组成的宏组。使用条件表达式可以决定在某些条件下运行宏时，某个操作是否执行。

利用宏可以打开或关闭窗体和报表，显示或隐藏工具栏，检索更新特定记录等。

以下对宏的几个基本概念给读者进行介绍。

### 操作

操作是由 Access 本身提供的，可以把它看成是完成一定功能的程序模块，当然这种比喻并不十分恰当，因为用户不能去修改这些所谓的程序模块。Access 提供了完成许多功能的操作，根据用途可将它们分成几类：窗体和报表中的数据处理；执行特定任务；导入、导出；对象处理；其他。

右图所示的宏包含 3 个操作—— OpenForm、Maximize 和 Close，分别实现打开一个窗体，使该窗体最大化，以及关闭该窗体的功能效果，这 3 个操作就构成了一个宏。每次运行该宏时，Access 都将执行这些操作。



### 宏和宏组

宏由一个或多个操作组成，而将多个宏组织起来就得到了宏组，这可以有助于用户方便地对数据库进行管理。右图所示就是一个宏组，它由两个宏组成，名字分别是“宏1”和“宏2”，其中“宏2”中的操作 Beep 是让计算机发出一种警告声音。



## 提示

运行宏时将顺序执行其中的每一个操作，但运行宏组时并不是顺序执行其中的每个宏。实际上，从根本上来说，宏组只是对宏的一种组织方式，宏组并不可执行，可执行的只是组中的各个宏。

## 宏的执行条件

在前面已经介绍了宏中的操作是顺序执行的，但在使用中常常会遇到分支情况或判断是否继续执行的情况，因此Access提供了操作是否执行的条件判断，只有该操作符合一定条件时才执行它。

对右图所示的宏组，要执行操作Set Value，则需要条件列“计数器=10”表达式为真。至于该表达式是什么意思，稍后将对表达式作比较详细的解释。



## 2

## 宏的创建和设计

Access

在 Access 中创建宏是一件非常轻松的事情，通过使用 Access 本身自带丰富的宏功能，读者将会发现，如果只是做一个小型的数据库，程序的流程用宏就可以完全实现，而无须使用更为复杂的 VBA。

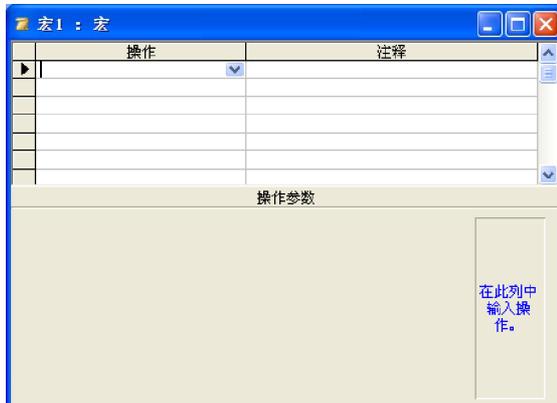
## 利用设计视图创建宏

宏设计视图用于宏的创建和设计，类似于窗体的设计视图。进入宏设计视图的具体操作步骤如下。

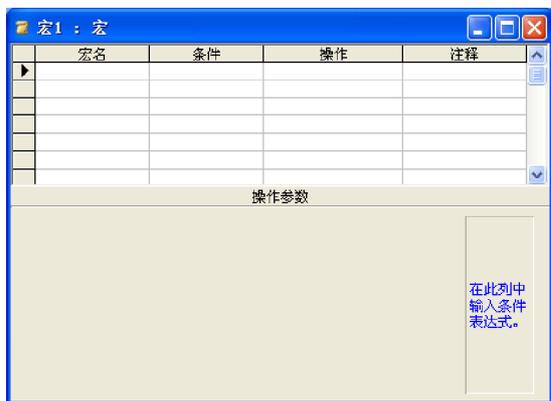
- 1 打开要创建的“数据库”窗口，在“对象”列表框中选择“宏”选项。



- 2 单击“数据库”窗口工具栏中的“新建”按钮即可进入宏设计视图。



- 单击工具栏的“宏名”和“条件”按钮，即可显示这两列。宏窗口下半部分是操作参数区，其左边是具体的参数及其设置，右边是帮助说明区。



- 在窗口上半部分的操作列中任选一个操作，其参数和说明便会显示在宏窗口的下半部分。



## 创建和设计宏

在宏窗口的“操作”列中可以直接输入操作名，也可以从其下拉列表中选择一个操作。当添加一个操作后，应当在“注释”列中加入说明性的文字，以便于将来使用时更易于理解。大多数操作需要指定参数，可以参照说明在下拉列表中选择。

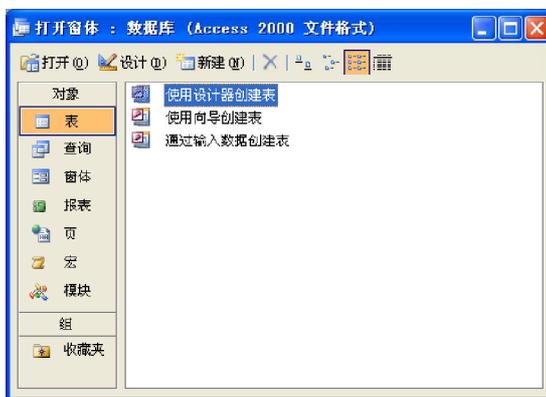
在定义了一个有多个操作的宏后，可能需要对其中的某些操作顺序进行改变。单击操作所在行的左端，该行将呈反色显示，此时可将它拖动到想要改变的位置。也可以删除该行，只需按Del键或者单击工具栏中的“删除行”按钮即可。要插入某个操作时，先用鼠标左键单击要插入的位置，再单击工具栏“插入行”按钮即可。

宏会按照由上至下的顺序执行直至结束。如果从初始的宏中调用了另外一个宏，那么，被调用的宏会启动，执行完毕后再把控制权返回给初始的宏。

因为宏大多是由控件事件触发来执行的，所以在此结合窗体控件来展示宏的创建和执行。

原始文件	Sample\Part15\打开窗体.mdb
学习要点	创建一个宏完成“打开窗体”操作

- 打开光盘中的Sample\Part15\打开窗体.mdb文件，并将它复制到本地硬盘。双击打开该数据库。

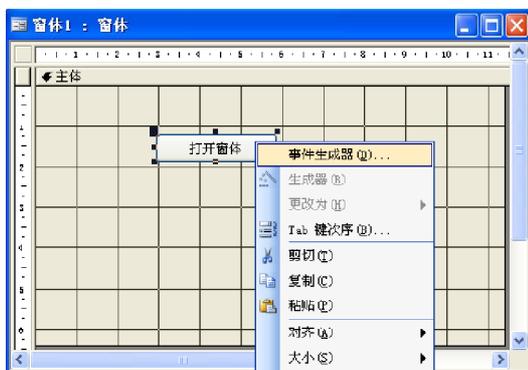




- 2 建立一个窗体，取名为宏的演示，在上面添加一个按钮，如果出现按钮向导的对话框，单击“取消”按钮。将按钮标题命名为“打开窗体”。



- 3 选中按钮，单击鼠标右键，在弹出的快捷菜单中选择“事件生成器”命令。此时会弹出“选择生成器”对话框。



- 4 从中选择“宏生成器”选项，单击“确定”按钮后在弹出的“另存为”对话框中输入宏名称。之后单击“确定”按钮进入宏设计视图。



- 5 在宏窗口的“操作”列中选择 OpenForm 选项，在下面参数的“视图”中选择“窗体”选项。关闭宏的设计视图并保存宏。



- 6 在设计窗体中打开“宏的演示”，查看按钮的属性对话框，将在“事件”选项卡中将“单击”设为“宏的演示”，表示单击此按钮时触发该宏的执行。



- 7 运行窗体“宏的演示”。





- 8 单击“打开窗体”按钮，将触发“宏的演示”宏的执行，该宏将打开“产品”窗体。

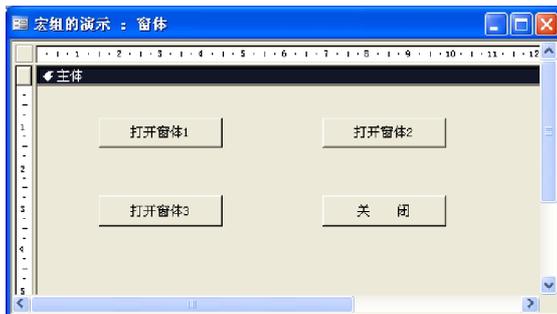
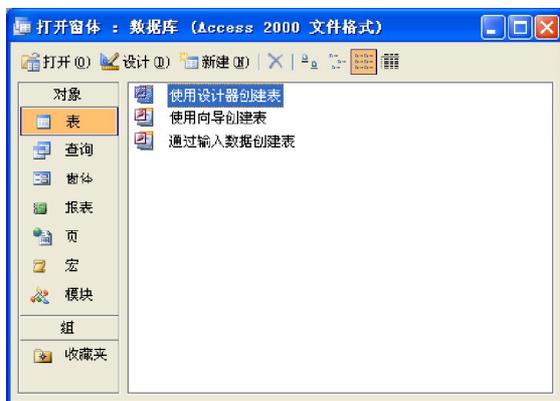


## 创建和设计宏组

把多个宏放在一起将组成一个宏组，它的创建和设计完全类似于宏。下面创建一个宏组，命名为“宏组的演示”。它由4个宏组成，前3个分别用于打开3个窗体，最后1个用于关闭窗体。在宏组中执行宏时，Access 将执行操作列中的操作和操作列中当“宏名”列为空时立即跟随的操作。

原始文件	Sample\Part15\打开窗体.mdb
学习要点	创建一个宏组执行打开窗体操作

- 1 打开光盘中的Sample\Part15\打开窗体.mdb文件，并将它复制到本地硬盘。双击打开该数据库。
- 2 建立一个窗体，将其命名为“宏组的演示”，并添加如下图所示的4个按钮。

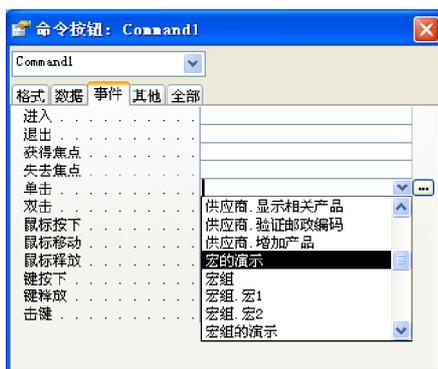


- 3 按照前面介绍的方法设计宏组，如右图所示。





- 4 为了在宏组中执行宏，使用这样的格式调用宏：宏组名 + 句点 + 宏名。在按钮属性对话框的“事件”选项卡的“单击”下拉列表中选择合适的宏。

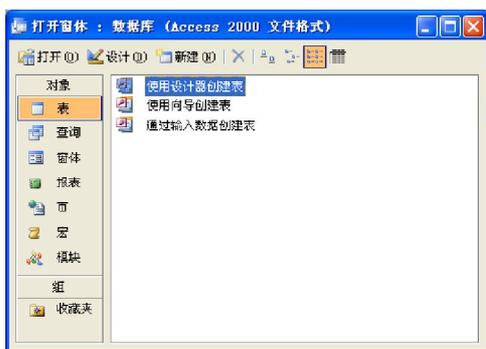


## 在宏中使用条件

条件即为逻辑表达式。宏将根据条件结果的真或假而沿着不同的路径执行。如果这个条件结果为真，则 Access 将执行此行中的操作。在紧跟此操作的“条件”列中键入省略号，就可以使 Access 在条件为真时执行这些操作；如果这个条件式的结果为假，Access 则会忽略这个操作以及紧接着此操作且在“条件”列中有省略号的操作，并且移到下一个包含其他条件字段的操作。如果“条件”列为空，Access 会执行该操作并计算下一个表达式。下面仍然结合实例来进行说明。

原始文件	Sample\Part15\打开窗体.mdb
学习要点	创建一个条件宏完成对数值大小的判断

- 1 打开本书配套光盘中的 Sample\Part15\打开窗体.mdb 文件，并将它复制到本地硬盘。双击打开该数据库。



- 2 建立一个窗体，并将其命名“条件宏”。之后添加一个文本框，并将其命名为“输入一个整数”，再添加一个“确定”按钮。



- 3 选中按钮，单击鼠标右键，在弹出快捷菜单中选择“事件生成器”命令，从中选择“宏生成器”选项，然后给宏取名为“条件宏”，进入宏设计视图。它将实现根据用户在文本框中输入的不同数字显示不同的消息。



- 4 右图所示为运行效果，当输入了一个数字 101，将弹出一个对话框，其中显示“您输入了一个大于 100 的整数”。



### 3 宏的执行和调试

在执行宏时，Access 将从宏的起点启动，并执行宏中符合条件的所有操作，直至宏组中所有需要执行的宏执行完毕为止。可以从其他宏或事件过程中直接执行宏，也可将执行的宏作为对窗体、报表和控件中发生的事件作出的响应。如前所述，可以将某个宏附加到窗体的命令按钮上，这样在用户单击按钮时就会执行相应的宏，也可创建执行宏的自定义菜单命令或工具栏按钮而将某个宏指定到组合键中，或者在打开数据库时自动执行宏。

当创建一个宏后可能会需要对宏所实现的功能进行检查，Access 提供了两个工具，以帮助用户解决在使用宏时遇到的问题。本节主要就宏的执行和调试进行说明。

#### 宏的执行

用户可以直接执行创建好的宏，通常有如下所示的几种执行方法。

- 在“数据库”窗口的“宏”选项界面中双击相应的宏名执行该宏。
- 在宏“设计”视图窗口单击工具栏的“执行”按钮执行正在设计的宏。
- 利用“工具”菜单执行宏。选择“工具>宏>执行宏”命令，然后在弹出的“执行宏”对话框的“宏名”下拉列表中选择需要执行的宏。
- 在窗体、控件、报表和菜单中调用执行宏，如前所述。
- 自动执行宏。将宏的名字设为 AutoExec，则在每次启动该数据库时，将自动执行该宏，这为数据库初始化方面提供了很大方便。

另外，宏还可嵌套执行，也就是说在一个宏中可以调用执行另一个宏。具体方法是：这里假设在宏中加入操作 RunMacro，并将操作 RunMacro 的参数“宏名”设为想要执行的宏。这样在执行新创建的宏时，就可以执行已有宏中的操作，而不必再在新建的宏中逐一添加所需的操作。





## 宏的调试

在宏的执行得到异常的结果时，就要使用宏的调试工具来单步执行。单击工具栏中的“单步”按钮或者选择“运行>单步”命令，向通常那样执行宏，就进入了宏的单步执行模式。在单步执行模式下的“单步执行宏”对话框中会有下列3个选项。

- 单步执行：执行在对话框列出的操作，如果没有错误，下一个操作会出现在对话框中。
- 停止：停止该宏的执行并关闭该对话框。
- 继续：关闭单步模式并继续执行该宏的后继部分。

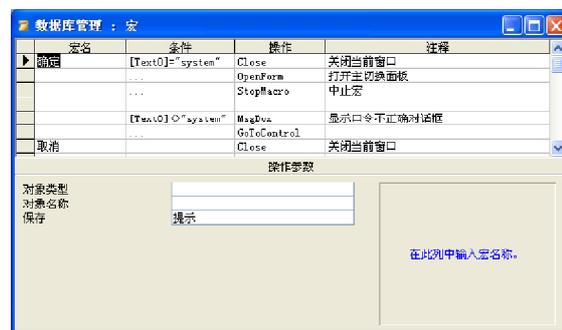
如果在用户创建的宏中存在错误，在单步执行宏时将会在窗口中显示“操作失败”对话框。Access将显示出错操作的名称、参数以及相应的条件，利用该信息可以了解宏出错的原因，然后就可进入宏“设计”视图进行修改。



## 设置管理员口令宏组

通常在管理信息系统中，为了保护数据库不随便让人浏览或修改以确保数据库的安全（后面有专门章节讲述数据库安全的相关知识），在进入管理系统前通常要用户输入用户名和密码，这个实例就将创建一个将包含条件的宏组，用来实现对口令的检验，如果口令正确，就先关闭这个身份核对窗口，再打开另一个窗体；如果口令不正确，将出现信息框要求重新输入口令。具体的设计步骤如下。

- 1 首先需要建立一个窗体，将其命名为“管理员身份核实对话框”。之后在窗体中添加文本框，将其命名为“请输入管理员口令”，用来接收用户输入的口令。再建立一个“确定”按钮和一个“取消”按钮。
- 2 下面开始设计宏组。建立一个宏组，“条件”列中的省略号 (...) 表示条件为真时可以连续执行这些操作。



- 3 为窗体上的“确定”按钮添加宏“数据库管理. 确定”，为“取消”按钮添加宏“数据库管理. 取消”。



- 4 运行窗体，输入口令system就可进入数据库管理窗口。若输入的口令不正确或者没有输入口令，就会显示口令错误对话框，提示用户在重新输入口令。



## 4 常用的宏

宏的操作功能是非常丰富的，如果只是做一个小型的数据库，程序的流程用宏就可以完全实现，而无须使用VBA。

有的宏操作是没有参数的（如Beep），而有的操作必须指定参数才行（如OpenForm）。通常，按排列顺序来设置操作的参数是很好的方法，因为选择某一参数将决定该参数后面参数的选择。表 15-1 所示为 Access 常用宏的简单介绍。

表 15-1 常用宏操作

操作	说明
Beep	通过计算机的扬声器发出嘟嘟声
Close	关闭指定的Access 窗口。如果没有指定窗口，则关闭活动窗口
GoToControl	把焦点移到打开的窗体、数据表中当前记录的特定字段或控件上
Maximize	放大活动窗口，使其充满Access 窗口
Minimize	将活动窗口缩小为 Access 窗口底部的小标题栏
MsgBox	显示包含警告信息或其他信息的信息对话框
OpenForm	打开窗体，并通过选择窗体的数据输入与窗口方式，来限制窗体所显示的记录
OpenReport	在“设计”视图或打印预览中打开报表或立即打印报表
PrintOut	打印打开数据库中的活动对象，也可以打印数据表、报表、窗体和模块
Quit	退出Access程序
RepaintObject	完成指定数据库对象的屏幕更新。更新包括对象的所有控件的所有重新计算
Restore	将处于最大化或最小化的窗口恢复为原来的大小
RunMacro	运行宏。该宏可以在宏组中
SetValue	对窗体、窗体数据表或报表上的字段、控件或属性的值进行设置
StopMacro	停止当前正在运行的宏

Access 2003



# part

# 16

## 模块设计

- ◆ VBA 编程基础
- ◆ VBA 的基本对象
- ◆ 基本数据库操作
- ◆ VBA 程序调试

VBA是Microsoft Office系列软件的内置编程语言，该语言的语法与独立运行的Visual Basic编程语言互相兼容。它使得在Microsoft Office系列应用程序中快速开发应用程序变得更加容易，并且可以完成特殊和复杂的操作。



## 1

## VBA 概述

Access

相信大多数的读者对于 VB 都比较熟悉，至少应该是听说过。VB 就是微软公司推出的可视化 BASIC 语言，用它来编程非常简单。因为它简单，而且功能强大，所以微软公司将它的一部分代码结合到了 Office 中，形成现在所说的 VBA。它的很多语法都继承自 VB，因此可以像编写 VB 语言那样来编写 VBA 程序，以实现某个功能。当这段程序通过编译以后，Office 将这段程序保存在 Access 中的一个模块里，并通过类似在窗体中激发宏的操作那样来启动这个“模块”，从而实现相应的功能。

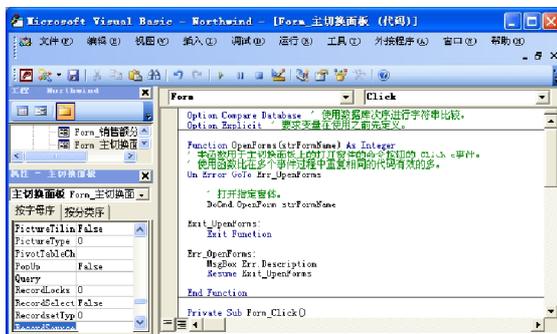
VBA 提供了一个编程环境和相应的一门语言，应用它可以自行定义应用程序以扩展 Office 的性能，将 Office 与其他软件相集成，并使 Office 一体化为一系列商业处理中的一环。通过使用 VBA 构建定制程序会充分利用 Office 提供的功能和服务。

在 Office 中提供的 VBA 开发界面称为 VBE (Visual Basic Editor)，在 VBE 中可编写 VBA 函数和过程。Access 的 VBE 界面与 Word, Excel 和 PowerPoint 的 VBA 开发界面基本一致。

## VBE 界面

在 Access 中有多种方式打开 VBE 窗口：可以先单击数据库窗口的“对象”列表框中的“模块”选项，然后双击所要显示的模块名称，就会打开 VBE 窗口并显示该模块的内容；也可以单击数据库窗口工具栏上的“新建”按钮，就会在 VBE 中创建一个空白模块；还可以通过在“数据库”窗口中，选择“工具>宏>Visual Basic 编辑器”命令打开 VBE 窗口。

VBE 界面由主窗口、工程资源管理器窗口、属性窗口和代码窗口组成。通过主窗口的“视图”菜单可以显示其他的窗口，这些窗口包括：对象窗口、对象浏览器窗口、立即窗口、本地窗口和监视窗口等，通过这些窗口可以方便用户开发 VBA 应用程序。



### (1) 菜单

在 VBE 中共有文件、编辑、视图、插入、调试、运行、工具、外接程序、窗口和帮助 10 个菜单。各个菜单的说明如表 16-1 所示。

表 16-1 VBE 中的菜单及其说明

菜单	说明
文件	文件的保存、导入和导出等基本操作
编辑	基本的编辑命令
视图	控制 VBE 的视图
插入	插入过程、模块、类或文件



(续表)

菜单	说明
调试	调试程序的基本命令, 包括监视、设置断点等
运行	运行程序的基本命令, 包括运行、中断等命令
工具	用来管理VB的类库等的引用、宏以及VBE编辑器的选项
外接程序	管理外接程序
窗口	设置各个窗口的显示方式
帮助	用来获得Microsoft Visual Basic的链接帮助以及网络帮助资源

## (2) 工具栏

在VBE窗口中默认情况下显示的是“标准”工具栏, 用户可以通过“视图”菜单的“工具栏”子菜单来显示“编辑”、“调试”和“用户窗体”工具栏, 甚至还可以自行定义工具栏。“标准”工具栏上包括了创建模块时常用的命令按钮, 关于这些命令按钮及其功能的介绍如表16-2所示。

表 16-2 “标准”工具栏常用按钮功能

按钮	按钮名称	功能
	视图Microsoft Office Access	切换到Access 窗口
	插入	单击该按钮右侧箭头, 在弹出的下拉菜单中包含“模块”、“类模块”和“过程”3个选项, 选一项即可插入新模块
	运行子过程 / 用户窗体	运行模块中的程序
	中断	中断正在运行的程序
	重新设置	结束正在运行的程序
	设置模式	在设计模式和非设计模式之间切换
	工程资源管理器	用于打开工程资源管理器
	属性窗口	用于打开属性窗口
	对象浏览器	用于打开对象浏览器

## (3) 窗口

在VBE窗口中, 提供了工程资源管理器窗口、属性窗口、代码窗口、对象窗口、对象浏览器窗口、立即窗口、本地窗口和监视窗口等多个窗口, 可以通过“视图”菜单控制这些窗口的显示。下面对常用的工程资源管理器窗口、属性窗口和代码窗口做简单介绍。

### ① 工程资源管理器窗口

“工程资源管理器”窗口中列出了在应用程序中用到的模块文件。在窗口上部可单击“查看代码”按钮显示相应的代码窗口; 或者单击“查看对象”按钮显示相应的对象窗口; 也可单击“切换文件夹”按钮隐藏或显示对象文件夹。

### ② 属性窗口

“属性窗口”列出了所选对象的各种属性, 可“按字母序”和“按分类序”查看属性。可以编辑这些对象的属性, 这通常比在“设计窗口”中编辑对象的属性要更加方便和灵活。

为了在“属性窗口”显示Access类对象, 应先在“设计”视图中打开对象。双击“工程窗口”上的一个模块或类, 相应的“代码窗口”就会显示相应的指令和声明, 但只有类对象在“设计”视图中也打开时, 对象才在“属性窗口”中被显示出来。

### ③ 代码窗口

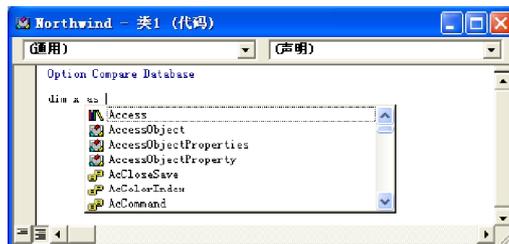
在“代码窗口”中可以输入和编辑VBA代码, 也可以打开多个“代码窗口”来查看各个模块的代

码，而且还可以方便地在“代码窗口”之间进行复制和粘贴操作。“代码窗口”对于代码中的关键字以及普通代码通过不同颜色加以区分，使之一目了然。

## 在“代码窗口”中编程

VBE 的“代码窗口”中包含了一套成熟的开发和调试系统。在“代码窗口”的顶部是两个下拉列表，左边是对象下拉列表，右边是过程下拉列表。对象下拉列表中列出的是所有可用的对象名称，选择某一对象后，在过程下拉列表中将列出该对象所有的事件过程。在“工程资源管理器”窗口中双击任何 Access 类或模块对象都可以在“代码窗口”中显示相应的代码，然后就可以对它进行检查、编辑和复制。

进行过 VB 编程的用户，一定对于 VB 方便友好的编程界面十分熟悉。VBE 继承了 VB 编辑器的众多功能，例如自动显示快速信息、快捷的上下文关联帮助以及快速访问子过程等功能。在代码窗口中输入命令时，VBA 编辑器会自动显示关键字列表供用户参考和选择。



利用“代码窗口”的相关功能，用户可以轻松地进行 VBA 应用程序的代码编写。正确地编写 VBA 应用程序的代码，首先要注意的就是程序的书写格式，下面做简要的介绍。

### (1) 注释语句

通常，一个好的程序一般都有注释语句。这对程序的维护以及代码的共享都有重要的意义。在 VBA 程序中，注释可以通过使用 Rem 语句或用“'”号实现。例如下面的代码中分别使用了这两种方式进行注释。

```
Rem 声明两个变量
Dim MyStr1, MyStr2 As String
MyStr1 = "Hello":      Rem MyStr1 赋值为 "Hello"
MyStr2 = "World"      ' MyStr2MyStr1 赋值为 "World"
```

其中 Rem 注释在语句之后要用冒号隔开，因为注释在代码窗口中通常以绿色显示，因此可以避免写错。

### (2) 连写和换行

在通常情况下，程序语句为一句一行，但有时候对于十分短小的语句，可能需要在—行中写几句代码，这时需要用到冒号(:)来分开几个语句。对于太长的代码可以用下划线(\_)将其截断为多行。

## 2

## VBA 基础知识

Access

## 数据类型

VBA 也同其他的编程语言一样，都要对数据进行操作。因此，VBA 支持多种数据类型，为用户编程提供了方便。表 16-3 列出了 VBA 程序中主要的数据类型，以及它们的存储要求和取值范围。



表 16-3 VBA 支持的数据类型

数据类型	类型名称	存储空间	取值范围
Byte	字节型	1 字节	0~255
Boolean	布尔型	2 字节	True或False
Integer	整型	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: -3.402823E38 ~-1.401298E-45; 正数: 1.401298E-45~3.402823E38
Double	双精度浮点型	8 字节	负数: -1.79769313486232E308~-4.94065645841247E-324; 正数: 4.94065645841247E-324 ~ 1.7976931486232E308
Currency	货币型	8 字节	-922 337 203 685 477.5808~922 337 203 685 477.5807
Decimal	十进制小数型	14 字节	无小数点时: +/-79 228 162 514 264 337 593 543 950 335 有小数点时又有28位数时: +/-7.9228162514264337593543950335; 最小的非零值: +/-0.0000000000000000000000 Decimal 数据类型只能在 Variant 中使用
Date	日期型	8 字节	100 年 1 月 1 日 到 9999 年 12 月 31 日
Object	对象	4 字节	任何对象引用
String(fixed)	定长字符串	10 字节+字符串长	0~大约20亿
String(variable)	变长字符串	字符串长	1~大约65 400
Variant(数字)	变体数字型	16 字节	任何数字值, 最大可达Double的范围
Variant(字符)	变体字符型	22 字节+字符串长	与变长String有相同的范围
Type	自定义类型	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

和其他的语言类似, VBA 可以自定义数据类型, 使用 Type 语句就可以实现这个功能。用户自定义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。Type 语句的语法如下:

```
Type TypeName
    定义语句
End Type
```

例如下面的 Type 语句, 定义了 MyType 数据类型, 它由 MyFirstName, MyLastName, MyBirthDate 和 MySex 组成。

```
Type MyType
MyFirstName As String ' 定义字符串变量存储一个名字
MyLastName As String ' 定义字符串变量存储姓
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
End Type
```



在VBA代码中可以通过声明和使用指定的常量或变量来临时存储数值、计算结果或操作数据库中的任意对象。

### 1. 变量的声明

声明变量有两个作用，一是指定变量的数据类型，二是指定变量的适用范围。VBA应用程序并不要求在过程中使用变量以前明确地进行声明。如果使用一个没有明确声明的变量，VBA会默认地将它声明为Variant数据类型。

虽然默认的声明很方便，但可能会在程序代码中导致严重的错误。因此使用前声明变量是一个很好的编程习惯。在VBA中可以强制要求在过程中使用变量前必须进行声明，方法是在模块通用节中包含一个Option Explicit语句。它要求在模块级别中强制对模块中的所有变量进行显式声明。

使用Dim语句来声明变量，该语句的功能是：声明变量并为其分配存储空间。Dim语句的语法如下：

```
Dim Variable_Name As DataType
```

其中变量名称可以像命名字段名一样，但是变量名不能包括空格键或其他字符（除了下划线）。另外，变量不能使用VBA关键字作为名字，因为关键字也称为保留字。例如，下面声明了字符串变量MyName。

```
Dim MyName As String
```

声明之后，就可以通过表达式给它赋值。也可以在同一行内声明多个变量，例如：

```
Dim Var1,Var2 As Integer, Var3 As String
```

其中Var1的类型为Variant，因为声明时没有指定它的类型。在变量声明时，对于用户自定义的数据类型与常规的数据类型没有区别，只要在使用之前定义了该数据类型即可。

### 2. 常量的声明

常量可以看作是一种特殊的变量，它的值经设置后就不能够更改或赋予新值。对于程序中经常出现的常数值，以及难以记忆且无明确意义的数值。使用声明常量可使代码更容易读取与维护。使用Const语句来声明常数并设置其值，Const语句的语法如下：

```
Const Const_Name = expression
```

声明一个常数PI：

```
Const PI=3.1415926
```

同样也可以在同一行里声明多个常量。

### 3. 变量和常量的作用域

变量和常量的作用域决定它们在VBA代码中的作用范围。变量在第一次声明时开始有效，用户可以指定变量在其有效范围内反复出现。一个变量有效，被称为是可见的，意味着可为其赋



值，并可在表达式中使用它，否则变量是不可见的。当变量不可见时若使用这个变量，则实际是创建了一个同名的新变量。对于常量也是一样。

可以在声明变量和常量的时候，对它的作用域也作出相应的声明。如果希望一个变量能被数据库中所有过程所访问，需要在声明时加上 **Public** 关键字。用 **Private** 语句可以显式地将一个变量的适用范围声明为在模块内，但这不是必须的，因为 **Dim** 和 **Static** 所声明的变量默认为在模块内私有。

#### 4. 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，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。使用 **Static** 语句可以声明静态变量，这样一来，该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。

清除过程中的静态变量的方法是：选择“运行>重新设置”命令。也可以用 **Static** 关键字来声明函数和子程序，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。

#### 5. 数组

可以用一个数组来表示一组具有相同数据类型的值。数组是单一类型的变量，可以存储很多值，而常规的变量只能存储一个值。定义了数组之后，可以引用整个数组，也可以只引用数组的个别元素。

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。标量变量（非数组）与数组变量的不同在于通常必须指定数组的大小。若数组的大小被指定的话，则它是个固定大小数组。若程序运行时数组的大小可以被改变，则它是个动态数组。数组是否从 0 或 1 开始索引，可以根据 **Option Base** 语句的设置，如果 **Option Base** 没有指定为 1，则数组索引从 0 开始。也可以使用 **To** 子句进行设定。若声明为动态数组，则可以在执行代码时去改变数组大小。可以利用 **Static**、**Dim**、**Private** 或 **Public** 语句来声明数组，并使括号内为空。

下面的语句声明了 3 个数组，其中 **Array1** 是大小为 11 的数组，**Array2** 是一个  $10 \times 20$  的二维数组，而 **Array3** 则是动态数组。

```
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。

#### 6. 表达式

表达式用来求取一定运算的结果，由变量、常量、函数、运算符和圆括号构成。在 VBA 中包含丰富的运算符，通过算术运算符、关系运算符、逻辑运算符和连接运算符，可以完成各种运算。各种运算符及其描述依次如表 16-4~表 16-7 所示，具体的每个运算符的功能和用法这里就不作介绍了。



表 16-4 算术运算符

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

表 16-5 比较运算符

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

表 16-6 逻辑运算符

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

表 16-7 连接运算符

符号	描述
+ 或&	字符串连接

表达式就是由各种运算符将变量、常量和函数连接起来构成的。但是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。对于包含多种运算符的表达式在计算时，将按预定顺序计算每一部分，这个顺序被称为运算符优先级。各种运算符的优先级顺序为从函数运算符、算术运算符、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。



## 程序控制语句

在 VBA 中的程序，按其语句代码执行的先后顺序，可以分为顺序程序结构、条件判断结构和循环程序结构。对于不同的程序结构要采用不同的控制语句才能达到预定的效果。下面介绍 VBA 中的这些控制语句。



## 1. If条件句

If 语句利用应用程序根据测试条件的结果对不同的情况作出反应。If 条件语句有如下 3 种形式。

### (1) If...Then

在程序需要作出“或者”的选择时，应该使用该语句。该语句又有两种形式，分别为单行形式和多行形式。

单行形式的语法如下：

```
If 条件 Then 语句
```

其中，“条件”是一个数值或一个字符串表达式。若“条件”为 True（真），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。

多行形式的语法如下：

```
If 条件 Then  
    语句  
End If
```

可以看出，与单行形式相比，只是要执行的语句通过 End If 来标志结束，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。

### (2) If...Then...Else

如果程序必须在两种条件中选择一种，则使用 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。

下面的代码判断如果 UpdateFlag 的值为 True 则显示一条消息 Update Successfully，否则显示一条信息 Failed!。

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

### (3) If...Then...Elseif...Else

如果要从 3 种或 3 种以上的条件中选择一种，则使用 If...Then...Elseif...Else。语法格式如下：

```
If 条件 1 Then  
    语句  
Elseif 条件 2 Then
```



```
    语句  
[Elseif 条件2 Then  
    语句]...  
Else  
    语句  
End If
```

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

下面的语句利用IF...Then通过对销售额进行判断，给出雇员的评价和佣金。

```
If Sales>15000 Then  
    Commission=Sales*0.08  
    Rating="Excellent"  
Elseif Sales>12000 And Sales<=15000 Then  
    Commission=Sales*0.06  
    Rating="Good"  
Elseif Sales>8000 And Sales<=12000 Then  
    Commission=Sales*0.05  
    Rating="Adequate"  
Else  
    Commission=Sales*0.04  
    Rating="Need Improvement"  
End If
```

## 2. Select 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语句就会显得很凌乱，而且程序变得不易阅读。这时可使用Select Case语句来写出结构清晰的程序。

Select Case语句可根据表达式的求值结果，选择执行几个分支中的一个，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语句语法中的Select Case后的表达式是必要参数，可为任何数值表达式或字符串表达式；在每个Case后出现表达式，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”以及“Is<比较操作符>表达式”等几种形式。每个Case后的语句都可



包含一条或多条语句。

续表

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，则VBA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”。

下面的语句利用Select...Case通过对销售额进行判断，给出雇员的评价和佣金。

```
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
```

### 3. Do...Loop语句

在许多实例中，用户需要重复一个操作直到满足给定条件才终止程序。例如，用户希望检查单词、句子或文档中的每一个字符，或者对有许多元素的数组赋值。循环就可用于这种情况下，一种较为通用的循环结构的形式是Do...Loop语句，它的语法如下：

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

其中，“条件”为可选参数，是数值表达式或字符串表达式，其值为True或False。如果条件为Null（无条件），则被当作False。While子句和Until子句的作用正好相反，如果指定了前者，则当<条件>是真时继续执行；如果指定了后者，则当<条件>为真时循环结束。如果把While或Until子句放在Do子句中，则必须满足条件才执行循环中的语句；如果把While或Until子句放在Loop子句中，则在检测条件前先执行循环中的语句。

在Do...Loop语句中的任何位置上均可放置任意个数的Exit Do语句，以便随时跳出Do...Loop循



环。Exit Do 语句通常用于条件判断之后，例如 If...Then，在这种情况下，Exit Do 语句将控制权转移到紧接在 Loop 命令之后的语句。如果Exit Do 使用在嵌套的Do...Loop语句中，则Exit Do 会将控制权转移到Exit Do 所在位置的外层循环。

下面的语句通过Do...Loop 循环为一个数组赋值。

```
Dim MyArray(10) As Integer
Dim i As Integer
i = 0
Do While i <= 10
    MyArray(i) = i
    i = i + 1
Loop
```

#### 4. While...Wend语句

在VBA中支持While...Wend 循环，它与Do While...Loop 结构相似，但不能在循环的中途退出。它的语法如下：

```
While 条件
语句
Wend
```

如果条件为True，则所有的语句都会执行，一直执行到Wend 语句，然后再回到While 语句，并再一次检查条件，如果条件还是为True，则重复执行，如果不为True，则程序会从Wend 语句之后的语句继续执行。While...Wend 循环也可以是多层的嵌套结构。每个Wend 匹配最近的While 语句。

下面的语句通过While...Wend 循环为一个数组赋值。

```
Dim MyArray(10) As Integer
Dim i As Integer
i = 0
While i <= 10
    MyArray(i) = i
    i = i + 1
Wend
```

在VBA中提供While...Wend 结构是为了与Basic的早期版本兼容，用户应该逐渐抛弃这种使用法，而使用Do...Loop 语句这种结构化与适应性更强的方法来执行循环。

#### 5. For...Next语句

For 循环可以将一段程序重复执行指定的次数，循环中使用一个计数变量，每执行一次循环，其值都会增加（或减少）。语法格式如下：

```
For 计数器 = 初值 To 末值 [步长]
语句
```



```
[Exit For]
语句
Next [计数器]
```

其中，“计数器”是一个数值变量。若未指定“步长”，则默认为1。如果“步长”是正数或0，则“初值”应大于等于“末值”；否则，“初值”应小于等于“末值”。VBA在开始时，将“计数器”的值设为“初值”。在执行到相应的Next语句时，就把步长加（减）到计数器上。

在循环中可以在任何位置放置任意个Exit For语句，随时退出循环。Exit For经常在条件判断之后使用（例如If...Then），并将控制权转移到紧接在Next之后的语句。可以将一个For...Next循环放置在另一个For...Next循环中，组成嵌套循环。不过在每个循环中的计数器要使用不同的变量名。

下面的语句通过For...Next循环为一个数组赋值。

```
Dim MyArray(10) As Integer
Dim i As Integer
For i=0 To 10
MyArray(i)=i
Next i
```

## 6. For Each...Next语句

For Each...Next语句针对一个数组或集合中的每个元素，重复执行一组语句。语法格式如下：

```
For Each 元素 In 组或集合
语句
[Exit For]
语句
Next 元素
```

For...Each...Next语句中的元素用来遍历集合或数组中所有元素的变量。对于集合来说，这个元素可能是一个Variant变量、一个通用对象变量或任何特殊对象变量。对于数组而言，这个元素只能是一个Variant变量。组或集合是数组或对象集合的名称。

如果集合中至少有一个元素，就会进入For...Each块执行。一旦进入循环，便先针对组或集合中第一个元素执行循环中的所有语句。如果在该组中还有其他的元素，则会针对它们执行循环中的语句，当组中的所有元素都执行完了，便会退出循环，然后从Next语句之后的语句继续执行。

可以在循环中任何位置放置任意个Exit For语句，以便随时退出循环。Exit For经常在条件判断之后使用（例如If...Then），并将控制权转移到紧接在Next之后的语句。可以将一个ForEach...Next循环放在另一个之中来组成嵌套式ForEach...Next循环，但是每个循环的元素必须是惟一的。

下面的语句通过For Each...Next循环为一个数组赋值并打印这些值。



```
Sub demo()  
Dim MyArray(10) As Integer  
Dim i As Integer  
Dim x As Variant  
  
For i = 0 To 10      ' 数组赋值  
    MyArray(i) = i  
Next i  
  
For Each x In MyArray 'For...Each...Next 循环  
    Debug.Print x    ' 打印数组中的每一项  
Next x  
End Sub
```

### 7. With...End With语句

With 语句可以对某个对象执行一系列的操作，而不用重复指出对象的名称。例如，要改变一个对象的多个属性，可以在 With 控制结构中加上属性的赋值语句，这时候只是引用对象一次而不是在每个属性赋值时都要引用它。它的语法格式如下：

```
With 对象  
    语句  
End With
```

下面的语句使用 With 语句来给同一个对象的几个属性赋值。

```
With MyLabel  
    .Height = 2000  
    .Width = 2000  
    .Caption = "This is MyLabel"  
End With
```

当程序一旦进入 With 块，对象就不能改变。因此不能用一个 With 语句来设置多个不同的对象。可以将一个 With 块放在另一个之中，而产生嵌套的 With 语句。但是，由于外层 With 块成员会在内层的 With 块中被屏蔽住，因此必须在内层的 With 块中，使用完整的对象引用来指出在外层的 With 块中的对象成员。

### 8. Exit语句

Exit 语句用于退出 Do...Loop, For...Next, Function, Sub 或 Property 代码块。它包含 Exit Do, Exit For, Exit Function, Exit Property 和 Exit Sub 几个语句。

下面的例子使用 Exit 语句退出 For...Next 循环、Do...Loop 循环及子过程。

```
Sub ExitStatementDemo()  
Dim I, MyNum  
Do                                ' 建立无穷循环。  
    For I = 1 To 1000            ' 循环 1000 次。  
    
```



```

        MyNum = Int(Rnd * 1000) ' 生成一随机数。
        Select Case MyNum      ' 检查随机数
            Case 7: Exit For    ' 如果是 7, 退出 For...Next 循环
            Case 29: Exit Do     ' 如果是 29, 退出 Do...Loop 循环
            Case 54: Exit Sub    ' 如果是 54, 退出子过程
        End Select
    Next I
Loop
End Sub

```

## 9. GoTo 语句

无条件地转移到过程中指定的行，它的语法如下：

### GoTo 行标签

其中行标签用来指示一行代码。行标签可以是任何字符的组合，以字母开头，以冒号(:)结尾。行标签与大小写无关，必须从第一列开始对其进行标注。GoTo 语句将用户代码转移到行标签的位置，并从该点继续执行。

下面使用 GoTo 语句在一个过程内的不同程序段间作流程控制。

```

Sub GotoStatementDemo()
Dim Number, MyString
    Number = 1          ' 设置变量初始值
                        ' 判断 Number 的值以决定要完成那一个程序区段（以“行标签”来
                        表式）
    If Number = 1 Then GoTo Line1 Else GoTo Line2
Line1:                 ' 行标签
    MyString = "Number equals 1"
    GoTo LastLine      ' 完成最后一行
Line2:                 ' 行标签
                        ' 下列的语句根本不会被完成。
    MyString = "Number equals 2"
LastLine:              ' 行标签
    Debug.Print MyString ' 将""Number equals 1"" 显示在“立即”窗口
End Sub

```

太多的 GoTo 语句，会使程序代码不容易阅读及调试。在 VBA 中使用 GoTo 语句只有一个目的，就是用 On Error GoTo Label 语句处理错误。



## 过程和模块

过程是 VBA 代码的容器。在 VBA 中有 3 种过程，分别是子过程、函数过程和属性过程。虽然在某些情况下，3 者的功能会有所重合，但是每种过程都有其独特和唯一的用途。而模块则是过程的容器。模块有两种基本类型：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。



### 1. 子过程

子过程是一系列由 Sub 和 End Sub 语句所包含起来的 VBA 语句，它们会执行动作却不能返回一个值。使用子过程可以执行动作、计算数值以及更新并修改内置的属性设置。

Sub 语句必须声明一个子过程名。对于事件除了过程有非常程式化的名称，过程命名通常要遵循标准的变量命名约定，必须以字母开头，长度不能超过 255 个字符，不能包含空格和标点，不能是 VBA 的关键字、函数和操作符名称。子过程可有参数，例如常数、变量或表达式等。如果一个子过程没有参数，则它的子语句必须包含一个空的圆括号。

下面是在 VBE 的代码窗口中直接输入代码编写子过程实现加法的例子。

```
Sub Demo()  
    Call Int_Add(1, 2) ' 调用 Int_Add 子过程求 1 和 2 的和  
End Sub  
  
Sub Int_Add(a As Integer, b As Integer)  
    Dim Result As Integer  
    Result = a + b  
    Debug.Print Result  
End Sub
```

将这段代码粘贴到一个新建的模块中，单击工具栏上的“运行”命令按钮，在弹出的“宏”对话框中选择运行 Demo 子过程。

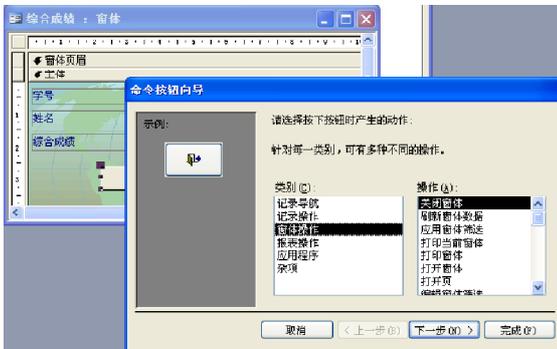


运行结果如右图所示。



在 Access 的窗口设计视图中，使用工具栏的“控件向导”命令，在往窗体中添加控件的同时，会有向导提示自动生成 30 多个功能的 VBA 代码。

可以在窗体的“设计”视图中，将工具栏上的“控件向导”命令按钮激活。例如，这时候选择按钮控件在窗体上绘制，然后出现“命令按钮向导”对话框，可以在其中选择不同的操作，则会依据所选的操作生成相应的子过程。





这里选择了关闭窗口操作，则生成了如下的关闭窗口的子过程代码。

```
Private Sub Close_Cmd_Click()
On Error GoTo Err_Close_Cmd_Click
    DoCmd.Close

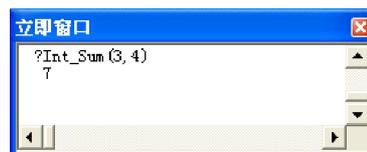
Exit_Close_Cmd_Click:
    Exit Sub

Err_Close_Cmd_Click:
    MsgBox Err.Description
    Resume Exit_Close_Cmd_Click
End Sub
```

## 2. 函数过程

函数过程是一系列由 Function 和 End Function 语句所包含起来的 VBA 语句。函数过程和子过程有两点不同。首先，函数可以返回一个值，因此在表达式中可以将其当作变量一样使用；其次，函数不能作为事件处理过程。其他方面二者很类似，例如，函数的传递参数可以是常数、变量或表达式。函数要在过程的一个或多个语句中指定一个值给函数名称来返回值。

下面这个函数过程求两个整数的和，将这段代码粘贴在“代码窗口”对话框中，可以通过在“立即窗口”对话框中输入一个问号后接函数名来激活函数，右图所示为激活这个函数求 3 与 4 的和等于 7。



```
Function Int_Sum(a As Integer, b As Integer)
    Dim Result As Integer
    Int_Sum = a + b
End Function
```

## 3. 属性过程

Property（属性）过程是一系列的 Visual Basic 语句，它允许程序员去创建并操作自定义的属性。Property 过程可以用来为窗体、标准模块以及类模块创建只读属性。声明 Property 过程的语法如下：

```
[Public | Private] [Static] Property {Get | Let | Set} 属性名 [(参数)] [As 类型]
语句
End Property
```

Property 过程通常是成对使用的：Property Let 与 Property Get 一组，而 Property Set 与 Property Get 一组，这样声明的属性既可读又可写。单独使用一个 Property Get 过程声明一个属性，那么这个属性是只读的。Property Set 与 Property Let 功能类似，都可以设置属性。不同的是 Property Let 将属性设置为等于一个数据类型，而 Property Set 则将属性设置等于一个对象的引用。



下面的语句使用Property Let语句定义给属性赋值的过程,使用Property Get语句定义获取属性值的Property过程。

```
Dim CurrentColor As Integer
Const BLACK = 0, RED = 1, GREEN = 2, BLUE = 3

' 设置绘图盒的画笔颜色属性
' 模块级变量CurrentColor 设为用于绘图的颜色值
Property Let PenColor(ColorName As String)
    Select Case ColorName          ' 检查颜色名称字符串
        Case "Red"
            CurrentColor = RED      ' 设为 Red
        Case "Green"
            CurrentColor = GREEN    ' 设为 Green
        Case "Blue"
            CurrentColor = BLUE     ' 设为 Blue
        Case Else
            CurrentColor = BLACK    ' 设为缺省值
    End Select
End Property
' 用一个字符串返回画笔的当前颜色
Property Get PenColor() As String
    Select Case CurrentColor
        Case RED
            PenColor = "Red"
        Case GREEN
            PenColor = "Green"
        Case BLUE
            PenColor = "Blue"
    End Select
End Property
' 下面的代码通过调用Property Let 过程来设置绘图盒的 PenColor 属性
PenColor = "Red"

' 下面的代码通过调用Property Get 过程来获取画笔的颜色。
ColorName = PenColor
```

#### 4. 类模块

类模块包含新对象定义的模块。当创建类的新实例时,即创建新对象。在模块中定义的过程成为该对象的属性和方法。类模块分为3种基本变形:窗体类模块、报表类模块和自定义类模块。窗体模块中包含了在指定的窗体或其控件的事件所触发的所有事件过程的代码。这些过程用于响应窗体中的事件。可以使用事件过程来控制窗体的行为,以及它们对用户操作的响应。报表模块与窗体模块类似,不同之处是工程响应和控制的是报表的行为。



数据库的每一个窗体/报表都有内置的窗体/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。自定义类模块不与窗体和报表相关联，允许用户定义自己的对象、属性和方法。

### 5. 标准模块

在标准模块中可以放置希望供整个数据库的其他过程使用的过程，这些过程不与任何对象相关联。单击“数据库”窗口的“对象”列表框中选择“模块”选项，可以查看数据库中标准模块的列表。

类模块和标准模块的不同点在于存储数据方法的不同。标准模块的数据只有一个备份。这意味着标准模块中一个公共变量的值改变以后，在后面的程序中再读取该变量时，它将得到改变后的值。而类模块的数据，是相对于类实例（也就是由类创建的每一对象）而独立存在的。同样的，标准模块中的数据在程序作用域内存在，也就是说，它存在于程序的存活期中；而类模块实例中的数据只存在于对象的存活期，它随对象的创建而创建，随对象的撤消而消失。最后，当变量在标准模块中声明为 Public 时，则它在工程中任何地方都是可见的；而类模块中的 Public 变量，只有当对象变量含有对某一类模块实例的引用时才能访问。

## 3

## VBA 对象

Access



### 理解对象、属性、方法和事件

VBA 是一种面向对象的语言，因此进行 VBA 的开发，必须理解对象、属性、方法和事件这几个概念。

学好 VBA 的诀窍之一就是要以“对象”的眼光去看待整个程序设计，而这个诀窍对于 VBA 的学习来说也同样适用。“对象”是面向对象程序设计的核心，明确这个概念对理解面向对象程序设计来说至关重要。对象的概念来源自在生活之中。对象就是一个事物，它可以是任何东西，如一座房子、一张桌子、一部电脑和一次旅行等。因此，在现实生活中，人们随时随地都在和对象打交道。

如果把问题抽象一下，会发现这些现实生活中的对象有两个共同的特点：第一，它们都有自己的状态，例如一个球有自己的质地、颜色和大小等属性；第二，它们都具有自己的行为，比如一个球可以滚动、停止或旋转。在面向对象的程序设计中，对象的概念就是对现实世界中对象的模型化，它是代码和数据的组合，同样具有自己的状态和行为。只不过对象的状态在这里用数据来表示，称为对象的属性；而对象的行为用对象中的代码来实现，称为对象的方法。不同的对象有不同的方法，当然也不排除有部分重叠。

对于 VBA 来说，最重要 VBA 应用程序对象就是用户所创建的窗体中出现的控件，所有的窗体、控件和报表等都是对象。而窗体的大小、控件的位置等都是对象的属性。这些对象可以执行的内置操作就是该对象的方法，通过使用这些方法可以控制对象的行为。

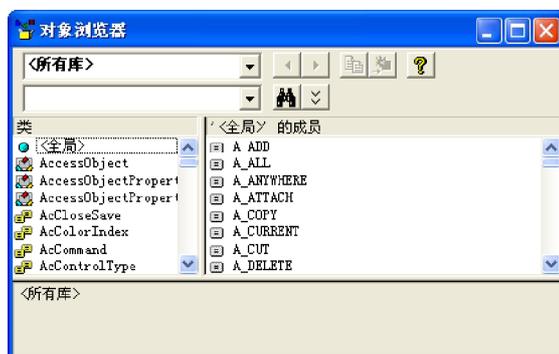
在 Access 的窗体设计视图中，可以通过“属性”对话框查看和设置对象的各个属性，通过上面的下拉列表框可以选择不同的对象进行查看。



对象的事件是一种特定的操作，它在某个对象上发生。例如，发生在窗体上的鼠标单击、数据更改、窗体打开或关闭及许多其他类型的操作都是事件。通常情况下，事件的发生是用户操作的结果。VBA 是基于事件驱动编程模型的，事件驱动编程是应用程序中的对象响应用户操作。在事件驱动的应用程序中，并不是按照预订的顺序执行，而是通过响应各种事件来运行不同的代码过程。这些事件可以由用户操作引起的，也可是来自系统、其他应用程序和应用程序的内部消息触发。通过使用事件过程，可以为在窗体、报表或控件上发生的事件添加自定义的事件响应。

用户不必关心所使用的对象需要响应的事件类型，因为在 Access 中的每一个窗体和控件都有一个预定义的事件集，它们能够自动识别属于事件集中的事件。对象所识别的事件类型多种多样，但多数类型为大多数控件所共有。例如，一个命令按钮和窗体都有可以对 Click 或 DblClick 这样的事件做出响应。某些事件只可能发生在某些对象上。相同事件发生在不同对象上所得到的反应是不一样的，造成这种差异是因为这些事件的事件过程不同。

类包含新对象的定义，通过创建类的新实例，可以创建新对象，而类中定义的过程就成为该对象的属性和方法。可以通过“对象浏览器”窗口查看各个库中的类，从而可以了解使用这些类创建的对象属性、方法和事件。在窗口中可以查看各个库中的类的列表，在列表右侧的窗格中显示在类中定义的对象属性、方法和事件。其中以  标志的是属性， 标志的是方法，而  标志的则是事件。对于选中的属性、方法和事件，在窗口的最下方会有简单的说明。



## VBA 的对象句法

前面介绍了对象和对象的属性、方法等概念，在编程的过程中需要引用对象、属性和方法。属性和方法不能够单独使用，它们必须和对应的对象一起使用。用于分隔对象和属性以及方法的操作符是“.”，称作点操作符。

引用属性的语法如下：



### 对象. 属性名

一般的属性都是可读写的，这样就可以通过上面的语法读取或是为属性赋值。例如，下面的代码用于读取 MyForm 对象的 Width 属性，并设置 MyForm 对象的 Caption 属性。

```
Width=MyForm.Width
MyForm.Caption="My Form"
```

引用方法的语法如下：

对象. 方法名 (参数1, 参数2...)

如果引用的方法没有参数，则可以省略括号。例如下面的代码引用 MyForm 对象的 Refresh 方法。

```
Form.Refresh
```

在 Access 中，要确定一个对象可能需要通过多重对象来实现。例如，要确定在 MyForm 窗体对象上的一个命令按钮控件 Cmd\_Button1，需要使用加重运算符“!”来逐级确定对象。

```
MyForm!Cmd_Button1
```



## 创建对象和模块

### 1. 创建对象

创建对象最简单的方法是在窗体的“设计”视图中通过工具栏创建各种控件对象。不过，用这种方法可以创建的对象十分少，因此对于大多数对象需要通过用对象变量创建对象引用。

除了存储值以外，变量可引用对象。如同给变量赋值一样，可把对象赋给变量，且引用包含对象的变量比反复引用对象本身有更高的效率。

用对象变量创建对象引用，首先要声明变量。声明对象变量的方法和声明其他变量一样，要用 Dim, ReDim, Static, Private 和 Public，语法如下：

```
Dim | ReDim | Static | Private | Public 对象变量 As [New]类
```

其中可选的 New 关键字可隐式地创建对象。如果使用 New 来声明对象变量，则在第一次引用该变量时将新建该对象的实例，因此不必使用 Set 语句来给该对象引用赋值。

如果在对象变量的声明时，没有使用 New 关键字，则要使用 Set 语句将对象赋予变量。

```
Set 对象变量 = [New] 对象表达式
```

通常使用 Set 将一个对象引用赋给变量时，并不是为该变量创建该对象的一份副本，而是创建该对象的一个引用。可以用多个对象变量引用同一个对象。由于这些变量只是该对象的引用，而不是对象的副本，因此对该对象的任何改动都会反应到所有引用该对象的变量。不过，如果在 Set 语句中使用 New 关键字，那么实际上就会新建一个该对象的实例。

例如，下面的代码声明了一个对象变量 anyForm，并使用 New 关键字隐式地创建对象，它可以



引用应用程序中的任何窗体，接着声明了一个能够引用应用程序中的任何文本框的 anyText 对象变量，然后用 Set 语句为 anyText 赋值。

```
Dim anyForm As New Form
Dim anyText As TextBox
Set anyText=New TextBox
```

还可以利用 CreateObject 或 GetObject 函数来初始化对象变量。CreateObject 创建一个 ActiveX 对象并返回该对象，这样就可以将 CreateObject 返回的对象赋给一个对象变量实现变量的初始化。如果要使用当前实例，或者要启动该应用程序并加载一个文件，可以使用 GetObject 函数。

CreateObject 函数的语法如下：

```
CreateObject (类名称)
```

例如，下面的代码创建一个到 Word 的引用。

```
Dim WordApp As Object ' 定义存放引用对象的变量
Set WordApp = CreateObject("word.application")
```

GetObject 返回文件中的 ActiveX 对象的引用，函数的语法如下：

```
GetObject (路径, 类名称)
```

其中“路径”用来指定包含对象的文件的路径和文件名。例如，下面的代码使用 GetObject 函数可以访问 C:\CAD\SCHEMA.CAD 文件中的 FIGMENT.DRAWING 对象，并将该对象赋给对象变量。

```
Dim CADObject As Object
Set CADObject = GetObject("C:\CAD\SCHEMA.CAD", "FIGMENT.DRAWING")
```

虽然过程结束时，过程中声明的变量被取消了，但下面的做法仍不失为一个良好的编程习惯：使用对象变量显式地退出已经自动化的应用程序，并通过将对象变量设置为 Nothing 关键字来取消这个对象变量。

下面的语句声明对象变量并赋值使用，在过程结束后取消对象变量。

```
Sub SendDataToWord()
    ' 定义对象变量
    Dim wdApp As Word.Application
    Dim wdDoc As Word.Document

    ' 实例化对象变量
    Set wdApp = New Word.Application
    Set wdDoc = wdApp.Documents.Add

    ' 在新文档中添加文本然后保存
    With wdDoc
```



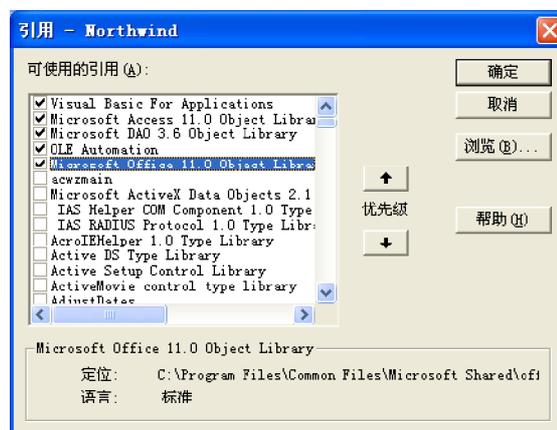
```

        .Range.Text = "Automation is cool!"
        .SaveAs "C:\AutomateWord.doc"
        .Close
    End With

    ' 取消对象变量
    Set wdDoc = Nothing
    wdApp.Quit
    Set wdApp = Nothing
End Sub

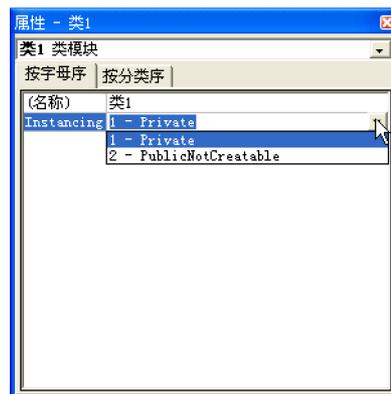
```

要运行该过程，可以将以上代码复制到任意 Access 的代码模块中，然后在菜单中选择“工具>引用”命令，在弹出的“引用”对话框的“可使用的引用”列表框中设置对 Word 对象库的引用，即可顺利运行该段代码。

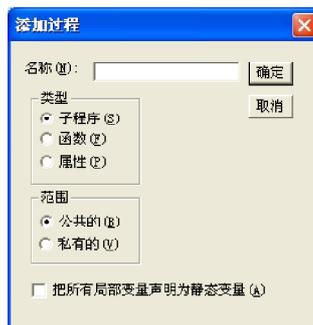


## 2. 创建类模块

用户可以自己编写类模块，以创建自定义的对象、属性和方法。选择“插入>类模块”命令，可以新建一个类模块，这时可以在“属性窗口”中为类设置名称和 Instancing 属性。类模块的“属性窗口”如右图所示。其中“名称”属性用来指定类的名称，因为类模块是对象的构架，因此在为类命名时，最好用一个能够表达该类的功能的名称。Instancing 属性用来设置当用户设置了一个到该类的引用时，这个类在其他工程中是否可见。这个属性有两个值 Private 和 PublicNotCreatable。如果 Instancing 属性的值设置为 Private，引用用户工程的工程在对象浏览器中不能够看到这个类模块，它也不能够使用这个类的实例进行工作。如果设置为 PublicNotCreatable，则引用工程可以在对象浏览器中看到这个类模块。引用工程可以使用类模块的一个实例进行工作，但是被引用的用户工程要先创建这个实例。引用工程本身不能够真正创建实例。



可以通过选择“插入>过程”命令，在弹出的“添加过程”对话框中向类模块中插入子过程、函数或属性。“添加过程”对话框如右图所示，用户可以在通过该对话框选择添加的过程的种类以及范围。



由于一个类模块代表了一个在运行时可按需要创建的对象，因此非常希望客户程序在开始使用对象或是退出它之后能够完成某些处理。这就可以由Class\_Initialize和Class\_Terminate子过程来完成。一旦对象被调入内存，在此对象的引用还没有返回到创建此对象的客户程序之前，它的Class\_Initialize子过程就被调用。当对象引用被设置为Nothing时，Class\_Terminate子过程将被调用。

下面是一个简单的类模块的例子，在其中创建了一个名为MyClass的类。在类模块中声明了一个模块级变量MyName，在对象初始化时，将MyName的值赋为MyClass，用户可以通过Name属性来读取或是设置MyName变量的值。在对象退出时将MyName变量赋为空字符串，代码如下。

```
Option Compare Database
Option Explicit
Dim MyName As String      ' 声明模块级变量用来存储Name属性的值

' 返回属性值
Public Property Get Name() As Variant
    Name = MyName
End Property

' 设置属性值
Public Property Let Name(ByVal vNewValue As Variant)
    MyName = vNewValue
End Property

' 对象初始化处理
Private Sub Class_Initialize()
    MyName = "MyClass"
End Sub

' 对象退出时处理
Private Sub Class_Terminate()
    MyName = ""
End Sub
```

在下面的代码中，创建MyClass类的实例MyFirstClass对象，读取和设置该对象的Name属性，



并通过立即窗口查看，代码如下。

```
Public Sub MyClass_Test()
Dim MyFirstClass As MyClass      ' 声明 MyFirstClass 对象

    Set MyFirstClass = New MyClass
    ' 使用 New 关键字为 MyFirstClass 对象赋值
    Debug.Print MyFirstClass.Name
    ' 在立即窗口中输出 MyFirstClass 对象 Name 属性值
    MyFirstClass.Name = "MyClass New Name"
    ' 为 MyFirstClass 对象 Name 属性赋新值
    Debug.Print MyFirstClass.Name
    ' 在立即窗口中输出 MyFirstClass 对象 Name 属性值
    Set MyFirstClass = Nothing    ' 退出 MyFirstClass 对象
End Sub
```

## 使用 Access 的对象模型

Access 对象是由 Access 定义的一种对象，它与 Access 界面或应用程序的窗体、报表和数据访问页相关，而且，可以用来对输入和显示数据所采用的界面的元素进行编程。VBA 通过使用 Access 的集合和对象可以操纵 Access 中的窗体、报表、页以及它们所包含的控件。可以利用这些对象来格式化和显示数据，并使得用户向数据库中添加数据成为可能。另外，Access 还提供了许多可以用来与 Access 应用程序一起工作的其他对象，例如 CurrentProject, CurrentData, CodeProject, CodeData, Screen 和 DoCmd 对象等。可以利用“对象浏览器”和 VBA 的帮助来获得个别对象、属性、过程和事件的信息。下面介绍一些常用的 Access 对象的使用。

### 1. Application 对象

Application 对象引用活动的 Access 应用程序。使用 Application 对象，可以将方法或属性设置应用于整个 Access 应用程序。在 VBA 中使用 Application 对象时，首先确认 VBA 对 Access 对象库的引用，然后创建 Application 类的新实例并为其指定一个对象变量，如以下示例所示：

```
Dim appAccess As New Access.Application
```

也可以通过用 CreateObject 函数来创建 Application 类的新实例，如下：

```
Dim appAccess As Object
Set appAccess = CreateObject("Access.Application")
```

创建 Application 类的新实例之后，可以使用 Application 对象提供的属性和功能创建和使用其他 Access 对象。例如，可以使用 OpenCurrentDatabase 或 NewCurrentDatabase 方法打开或新建数据库；可以通过用 Application 对象的 CommandBars 属性返回对 CommandBars 对象的引用；可以使用该引用来访问所有的 Microsoft Office 2003 命令栏对象和集合；还可以通过 Application 对象处理其他 Access 对象。

下面的这段代码创建了一个 Application 对象，使用 Application 对象的 OpenCurrentDatabase 方法



打开 NorthWind 数据库，再使用它的子对象 DoCmd 的 OpenForm 方法打开“订单”窗体。其中 Application 对象的 OpenCurrentDatabase 方法可以打开一个已有 Access 数据库 (.mdb) 作为当前的数据库。

```
Option Compare Database

' 声明 Application 对象
Dim appAccess As Access.Application

Sub DisplayForm()
    ' 将字符串初始化为数据库路径
    Const strConPathToSamples = "C:\Program Files\Microsoft Office\Office10\Samples\"

    strDB = strConPathToSamples & "Northwind.mdb"
    ' 新建 Microsoft Access 实例
    Set appAccess = CreateObject("Access.Application")
    ' 使用 OpenCurrentDatabase 方法在 Microsoft Access 窗口中打开数据库
    appAccess.OpenCurrentDatabase strDB
    ' 打开“订单”窗体
    appAccess.DoCmd.OpenForm "订单"
End Sub
```

## 2. Form 对象、Forms 集合和 Controls 对象、Control 集合

Form 对象引用一个特定的 Access 窗体。Form 对象是 Forms 集合的成员，该集合是所有当前打开窗体的集合。在 Forms 集合中，每个窗体都从 0 开始编排索引。通过按名称或按其在集合中的索引引用窗体，可以引用 Forms 集合中的单个 Form 对象。如果要引用 Forms 集合中指定的窗体，最好是按名称引用窗体，因为窗体的集合索引可能会变动。如果窗体名称包含空格，那么名称必须用方括号 ([ ]) 括起来。引用 Forms 集合中的单个 Form 对象的语法如表 16-8 所示。

表 16-8 引用 Forms 中 Form 的语法

语法	示例
Forms!formname	Forms!OrderForm
Forms![form name]	Forms![Order Form]
Forms("formname")	Forms("OrderForm")
Forms(index)	Forms(0)

每个 Form 对象都有一个 Controls 集合，其中包含该窗体上的所有控件。要引用窗体上的控件，可以显式或隐式地引用 Controls 集合。例如，下面的代码引用 OrderForm 窗体上名为 NewData 的控件。

```
' 隐式引用
Forms!OrderForm!NewData
' 显式引用
Forms!OrderForm.Controls!NewData
```



### 3. Modules 对象和 Module 集合

Modules 集合包含 Access 数据库中所有打开的标准模块和类模块，不论模块是未经编译的、已经编译的、处于中断模式还是包含正在运行的代码。Module 对象用于引用标准模块或类模块，可以返回对 Modules 集合中特定的标准或类 Module 对象的引用。

下面的代码返回一个对 Modules 集合中指定窗体 Module 对象的引用并将其赋予一个模块对象变量。

```
Dim MyModule As Module  
Set MyModule = Modules!Form_Employees
```

### 4. DoCmd 对象

使用 DoCmd 对象的方法，可以从 Visual Basic 运行 Access 操作。这些操作可以包括执行如关闭窗口、打开窗体和设置控件值等任务。例如，可以使用 DoCmd 对象的 OpenForm 方法来打开一个窗体，或者使用 Hourglass 方法将鼠标指针改为沙漏图标。

DoCmd 对象的大多数方法都有参数，其中的某些参数是必需的，而其他一些是可选的。如果省略可选参数，那么这些参数将被假定为特定方法的默认值。

下面的代码用于在“窗体”视图中打开一个窗体并移到一条新记录。

```
Sub ShowNewRecord()  
    DoCmd.OpenForm "Employees", acNormal  
    DoCmd.GoToRecord , , acNewRec  
End Sub
```

## 4

## VBA 程序调试

Access

调试是查找和解决 VBA 程序代码错误的过程。当程序代码执行时，会产生以下两种类型的错误。

### (1) 开发错误

开发错误包括语法错误和逻辑错误。语法错误可能由输入错误、标点丢失或是不适当的使用某些关键字等产生的。例如，遗漏了配对的语句（例如，If 和 End，以及 If 或 For 和 Next）、程序设计上违反了 VBA 的规则（例如，拼写错误、少一个分隔点或类型不匹配等）。逻辑错误是指应用程序未按设计执行，或生成了无效的结果。这种错误是由于程序代码中不恰当的逻辑设计而引起的。这种程序在运行时并未不属于非法操作，只是运行结果不符合要求。

### (2) 运行时错误

运行时错误是在程序运行的过程中发生的。有运行时错误的代码在一般情况下运行正常，但是遇到非法数据或是系统条件禁止代码运行时（例如，磁盘空间不足等）就会发生错误。

编写容易理解、可维护的代码和使用有效的调试工具可以减少和排除上述的错误。



### 良好的编程风格

为了避免不必要的错误，应该保持良好的编程风格。通常应遵循以下几条原则。

- ① 模块化。除了一些定义全局变量的语句以及其他的说明性语句之外，具有独立作用的非说明性语句和其他代码，都要尽量地放在 Sub 过程或 Function 过程中，以保持程序的简洁性，并清晰明了地按功能来划分模块。
- ② 多注释。编写代码时要加上必要的注释，以便以后或其他用户能够清楚了解程序的功能。
- ③ 变量显式声明。在每个模块中加入 Option Explicit 语句，强制对模块中的所有变量进行显式声明。
- ④ 良好的命名格式。为了方便地使用变量，变量的命名应采用统一的格式，尽量做到能够“顾名思义”。
- ⑤ 少用变体类型。在声明对象变量或其他变量时，应尽量使用确定的对象类型或数据类型。少用 Object 和 Variant。这样可加快代码的运行，且可避免出现错误。

## “调试”工具栏及功能

VBE 提供了“调试”菜单和“调试”工具栏（其中的命令和按钮的功能均一一对应），选择“视图>工具栏>调试”命令，即可弹出“调试”工具栏。

“调试”工具栏上各个按钮的功能说明如表 16-9 所示。

表 16-9 “调试”工具栏命令按钮说明

命令按钮	按钮名称	功能说明
	设计模式	打开或关闭设计模式
	运行子过程/用户窗体	如果光标在过程中则运行当前过程。如果用户窗体处于激活状态，则运行用户窗体；否则将运行宏
	中断	终止程序的执行，并切换到中断模式
	重新设置	清除执行堆栈和模块级变量并重新设置工程
	切换断点	在当前行设置或清除断点
	逐语句	一次执行一句代码
	逐过程	在代码窗口中一次执行一个过程或一句代码
	跳出	执行当前执行点处过程的其余行
	本地窗口	显示“本地窗口”对话框
	立即窗口	显示“立即窗口”对话框
	监视窗口	显示“监视窗口”对话框
	快速监视	显示所选表达式的当前值的“快速监视”对话框
	调用堆栈	显示“调用堆栈”对话框，列出当前活动过程调用

## 调试方法及技巧

### 1. 执行代码

VBE 提供了多种程序运行方式，通过不同的运行方式运行程序，可以对代码进行各种调试工作。

#### (1) 逐语句执行代码

逐语句执行是调试程序时十分有效的工具。通过单步执行每一行程序代码（包括被调用过程中的程序代码）可以及时、准确地跟踪变量的值，从而发现错误。如果要逐语句执行代码，可单击“调试”工具栏上的“逐语句”按钮，之后，VBE 运行当前语句，并自动转到下一条



语句，同时将程序挂起。

对于在一行中有多条语句用冒号隔开的情况，在使用“逐语句”工具时，将逐个执行该行中的每条语句。

### (2) 逐过程执行代码

如果希望执行每一行程序代码，而不关心在代码中调用的子过程的运行，并将其作为一个单位执行，可单击“调试”工具栏上的“逐过程”按钮。逐过程执行与逐语句执行的不同之处在于执行代码调用其他过程时，逐语句是从当前行转移到该过程中，在此过程中一行一行地执行，而逐过程执行则将调用其他过程的语句当作一个语句，将该过程执行完毕，然后进入下一语句。

### (3) 跳出执行代码

如果希望执行当前过程中的剩余代码，可单击“调试”工具栏上的“跳出”按钮。在单击“跳出”按钮后，VBE 会将该过程未执行的语句全部执行完，包括在过程中调用的其他过程。执行完过程后，程序返回到调用该过程的过程，“跳出”命令执行完毕。

### (4) 运行到光标处

选择“调用>运行到光标处”命令，VBE 就会将程序运行到当前光标处。当用户可确定某一范围的语句正确，而对后面语句的正确性不能保证时，可用该命令运行程序到某条语句，再在该语句后逐步调试。这种调试方式通过光标来确定程序运行的位置，操作十分简便。

### (5) 设置下一语句

在 VBE 中，用户可自由设置下一步要执行的语句。当程序已经挂起时，可在程序中选择要执行的下一条语句，单击鼠标右键，并在弹出的快捷菜单中选择“设置下一条语句”命令。

## 2. 暂停代码运行

VBE 提供的大部分调试工具，都要在程序处于挂起状态才能有效，这时就需要暂停 VBA 程序的运行。在这种情况下，程序仍处于执行状态，只是暂停于正在执行的语句之间，变量和对象的属性仍然保持，当前运行的代码在模块窗口中被显示出来。

如果要将语句设为挂起状态，可采用以下几种方法。

### (1) 断点挂起

如果 VBA 程序在运行时遇到了断点，系统就会在运行到该断点处时将程序挂起。可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。在程序运行时不能设置断点，只有在编写程序代码或程序处于挂起状态才可设置断点。

可以在模块窗口中，将光标移到要设置断点的行，按 F9 键，或单击“调试”工具栏上的“切换断点”按钮来设置断点，也可以在模块窗口中，单击要设置断点行的左侧边缘部分，即可设置断点。

如果要消除断点，可将插入点移到设置了断点的程序代码行，然后单击“调试”工具栏上的“切换断点”按钮，或者在断点代码行的左侧边缘单击。

### (2) Stop 语句挂起

给过程中添加 Stop 语句，或者在程序执行时按 Ctrl+Break 键，也可将程序挂起。Stop 语句是添加在程序中的，当程序执行到该语句时将被挂起。它的作用与断点类似，但当用户关闭数据库后，所有断点都会自动消失，而 Stop 语句却还在代码中。如果不再需要断点，则可选择“调试>清除所有断点”命令将所有断点清除，但 Stop 语句须逐行清除，比较麻烦。

### 3. 查看变量值

VBE 提供了多种查看变量值的方法，下面简单介绍各种查看变量值的方式。

#### (1) 在“代码窗口”中查看数据

在调试程序时，希望随时查看程序中的变量和常量的值，这时候只要将光标指向要查看的变量和常量，就会直接在屏幕上显示当前值。这种方式最简单，但是只能查看一个变量或常量。如果要查看几个变量或一个表达式的值，或者需要查看对象及其属性，就不能直接通过光标指向该对象或表达式在“代码窗口”中来查看了。

#### (2) 在“本地窗口”对话框中查看数据

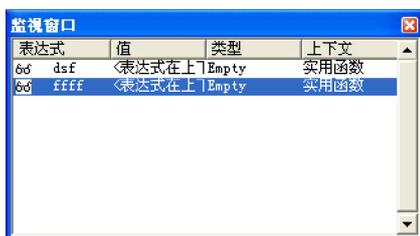
可单击“调试”工具栏上的“本地窗口”按钮打开“本地窗口”对话框。在该对话框中有 3 个列表，分别显示“表达式”、表达式的“值”和表达式的“类型”。有些变量，如用户自定义类型、数组和对象等，可包含级别信息。这些变量的名称左边有一个加号按钮，可通过它控制级别信息的显示。

列表中的第一个变量是一个特殊的模块变量。对于类模块，它的系统定义变量为 Me。Me 是对当前模块定义的当前类实例的引用。因为它是对象引用，所以能够展开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，并且也能展开显示当前模块中所有模块级变量。在“本地窗口”对话框中，可通过选择现存值，并输入新值来更改变量的值。



(3) 在“监视窗口”对话框中查看变量和表达式。在程序执行过程中，可利用“监视窗口”查看表达式或变量的值。可选择“调试>添加监视”命令来设置监视表达式。通过“监视窗口”可展开或折叠级别信息、调整列标题大小以及就地编辑值等。



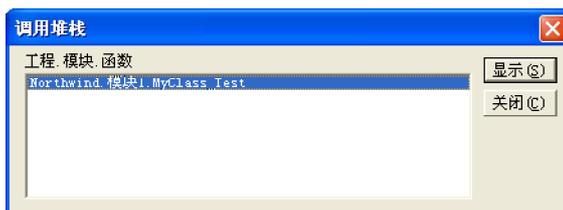
#### (4) 在“立即窗口”对话框中查看结果

使用“立即窗口”可检查一行 VBA 代码的结果。可以键入或粘贴一行代码，然后按下 Enter 键来执行该代码。可使用“立即窗口”检查控件、字段或属性的值，显示表达式的值，或者为变量、字段或属性赋予一个新值。“立即窗口”是一种中间结果暂存器窗口，在这里可以立即求出语句、方法和 Sub 过程的结果。

可以将 Debug 对象的 Print 方法加到 VBA 代码中，以便在运行代码过程中，在“立即窗口”对话框中显示表达式的值或结果，这在前面的众多的示例中都有应用。

#### (5) 跟踪 VBA 代码的调用

在调试代码过程中，当暂停 VBA 代码执行时，可在“调用堆栈”对话框中查看那些已经开始执行但还未完成的过程列表。如果持续在“调试”工具栏上单击“调用堆栈”按钮，Access 会在列表的最上方显示最近被调用的过程，接着是早些被调用的过程，依次类推。



Access 2003



# part

# 17

## 定制与管理Access

- ◆ 分析优化数据库
- ◆ 安全管理
- ◆ 保护数据库

在 Access 中提供了很多工具来构造、管理和优化数据库。由于数据库内保存大量数据，因此需要考虑数据库的安全性问题。本章主要内容包括数据表分析、数据表优化和安全管理。

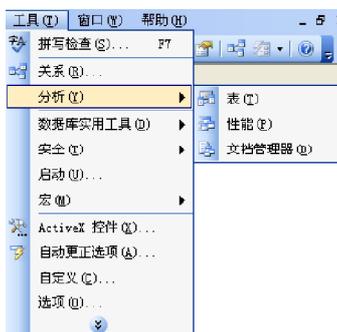


为了便于用户更安全地使用数据库中的数据，本章将针对数据表的分析、优化和数据库安全管理来加以介绍，使用户了解如何在 Access 数据库和数据表中保护和优化数据。

## 1 分析优化数据库

通过前面章节的介绍，读者可能已经掌握了数据库的基本操作，但有时建立的数据库使用起来却发现很慢，那是因为数据库在建立的时候，没有对它进行过优化分析。现在就讲讲数据库的优化分析。

先打开一个要进行分析的数据库，然后选择“工具>分析”命令，在其子菜单上有“表”、“性能”和“文档管理器”3个命令。这3个命令可以对相应的内容进行优化。



### 优化表

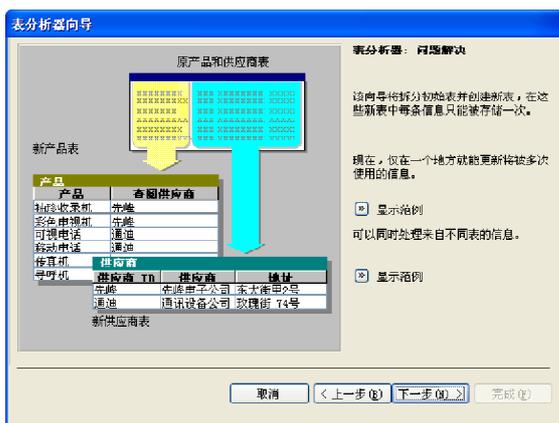
原始文件	Sample\Part17\产品管理.mdb
学习要点	优化产品管理数据库中的表

- 1 打开本书配套光盘中的Sample\Part 17\产品管理.mdb文件，并将它复制到本地硬盘。双击打开该数据库。对表进行优化需要选择“工具>分析>表”命令，将会弹出“表分析器向导”对话框。
- 2 在这个向导的初始界面中提供了建立表时常见的一些问题。这就是表或查询中多次存储了相同的信息，而且重复的信息将会给带来很多问题。看完了这些，就可以单击“下一步”按钮。





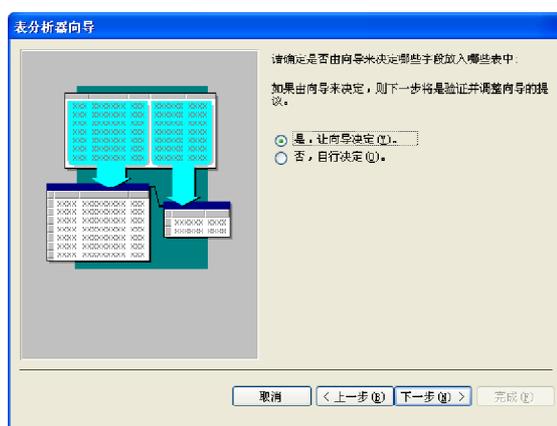
- 3 第二步的界面中显示这个分析器是怎样解决第一步中遇到的问题。解决的办法是将原来的表拆分成几个新的表，使新表中的数据只被存储一遍。



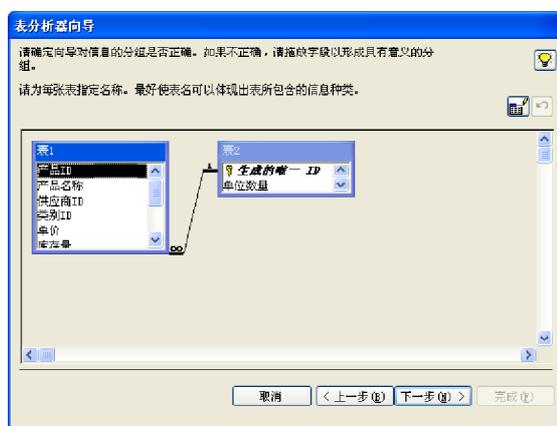
- 4 单击“下一步”按钮，在这一步界面的“表”列表框中选择需要做分析的表。虽然Access提示只选择有重复信息的表，但最好对所有的表都做一个分析，这样并花不了很多时间，反而能使效果更加明显。



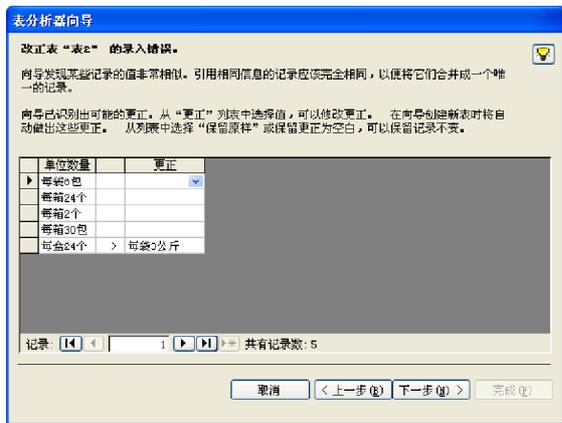
- 5 选择好要分析的表以后，单击“下一步”按钮，在这一步的界面中选择“是”，让向导决定选项，这样就可以让Access自动完成对这个表的分析工作。



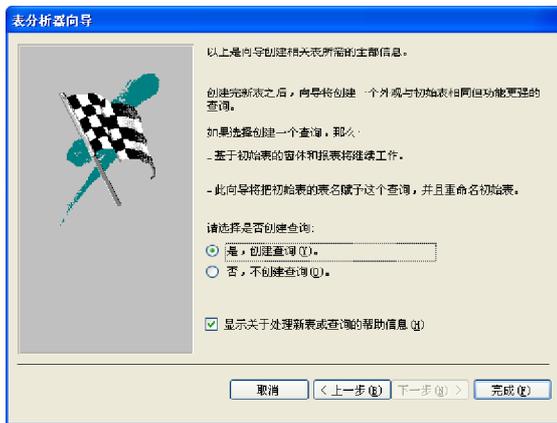
- 6 单击“下一步”按钮，通过分析就会在屏幕上弹出一个对话框，在这个对话框中将为每张表指定名称。现在Access的分析向导已经将的表拆分成了几个表，并且在各个表之间建立起了相应的关系。



- 7 单击“下一步”按钮后，弹出如下图所示的界面，在其中将对数据表中的数据错误进行改正。



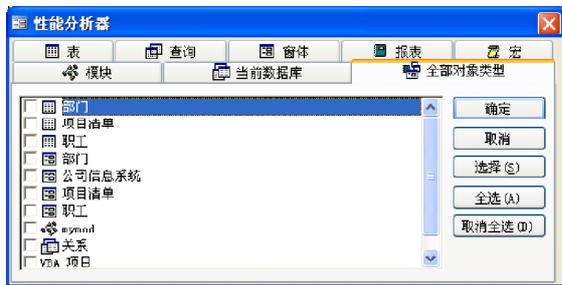
- 8 单击“下一步”按钮后，弹出如下图所示的界面，在其中提示是否创建查询来保存分析结果，最后单击“完成”按钮即可。



## 性能分析

刚才对表进行了分析，下面对其进行性能分析，具体操作步骤如下。

- 1 选择“工具>分析>性能”命令，将会弹出“性能分析器”对话框。现在就可以对整个数据库进行性能分析了。为了使用的方便，常常切换至“全部对象类型”选项卡。
- 2 单击“全部对象类型”选项卡上的“全选”按钮，这样虽然会多花一些时间进行性能分析，但却更加有效。完成这些后，单击选项卡上的“确定”按钮，现在Access就开始为数据库进行优化分析了，并将结果显示在如下图所示的界面中。



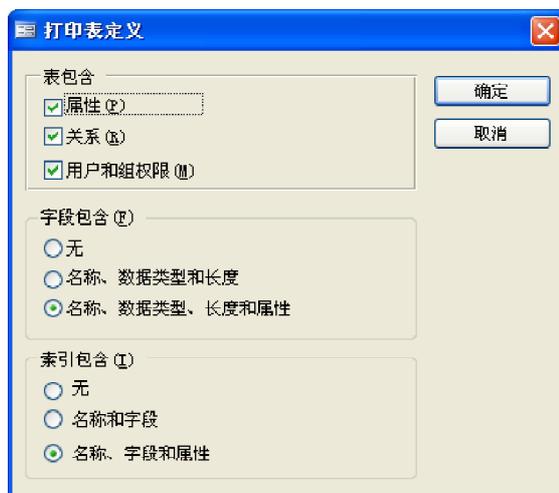
在列表框中每一项前面都有一个符号，每个符号都代表一个意思，在其中都有介绍。现在如果在列表框中有“推荐”和“建议”，就单击“全选”按钮，这时在列表框中的每个选项都被选中了。

## 使用文档管理器

选择“分析>文档管理器”命令，将会弹出“文档管理器”对话框。在这个对话框中也有选项卡，切换至“全部对象类型”选项卡，然后在相应的列表框中选择需要的对象名，选好以后，单击“确定”按钮就可以将这些选项的各种内容显示出来，如果需要可以将这些内容打印出来。



在该对话框上有一个“选项”按钮，这个按钮是用来确定打印表的含义。单击这个按钮后，会弹出“打印表”对话框。在这个对话框中包含“表包含”、“字段包含”、“索引包含”3个选项区，选择其中不同的选项，会改变打印表中将要显示的信息的内容。当完成这些工作，单击“确定”按钮即可。有了这些信息，有经验的Access使用者就可以从打印出的信息资料上分析出所建立的数据库有哪些问题了。



## 2

## 安全管理

用户在使用Access数据库的过程中，还会积累很多的经验，这些经验有利于用户更加合理的使用数据库。现在就来介绍Access数据库的安全管理。

通常用户建立的数据库并不希望所有的人都能使用，并修改数据库中的内容。这就要求数据库实行更加安全的管理，即限制一些人的访问及修改数据库中的内容。这样一来，访问者必须输入相应的密码才能对数据库进行操作，而且输入不同密码的人所能进行的操作也是有限制的。除了这些，数据库的安全还包括对数据库中的数据进行编码和解码工作。这样建立需要保密的数据库就不能被别人轻易攻破，相应起到了安全保密的作用。

要进行安全管理就需要实现刚才所说的那几个目标，在Access中提供了相应的几个命令来实现这些目标。

现在就来为数据库“成绩查询”添加安全管理。要对那个数据库进行安全管理，首先需要将这个数据库打开，然后选择“工具>安全”命令，这时在“安全”命令的子菜单中显示5个命令，其中每个命令都能执行一定的功能。



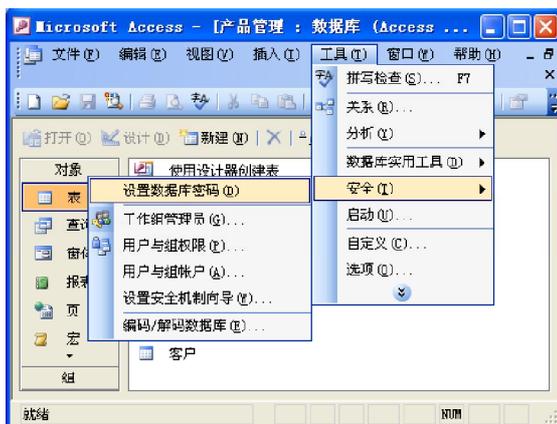
## 设置和取消数据库密码

原始文件	Sample\Part17\产品管理.mdb
学习要点	设置和取消数据库密码

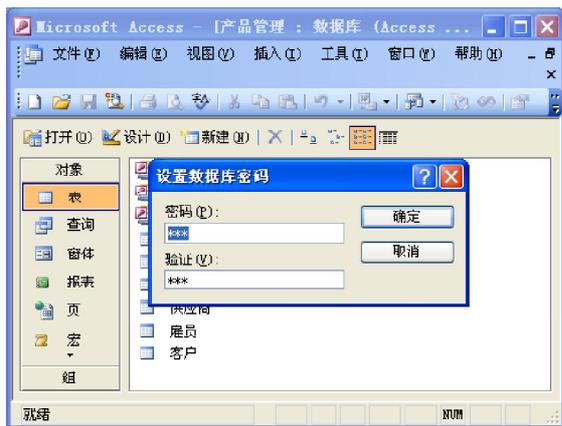
1 打开本书配套光盘中的Sample\Part 17\产品管理.mdb文件，并将它复制到本地硬盘。双击打开该数据库。



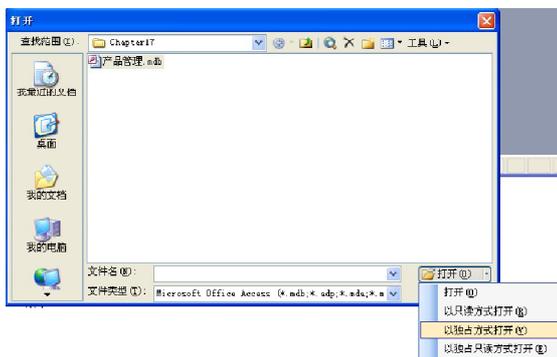
2 选择“工具>安全>设置数据库密码”命令将会弹出“设置数据库密码”对话框。



3 在该对话框的“密码”文本框中输入数据库密码，并在“验证”文本框中再输入一遍刚才输入的密码，以保证输入的密码不会因为误输入而造成以后无法打开自己的数据库。将这些设置完成以后，单击“确定”按钮即可。



4 记住要设置数据库的密码时，必须要保证这个数据库是独占打开方式，方法如下。单击工具栏上的“打开”按钮，这时会弹出“打开”对话框。单击其中的“打开”按钮旁的下拉箭头，在弹出的下拉菜单中选择“以独占方式打开”命令即可。





- 5 按照刚才输入密码的过程再来一遍，这样就可以给这个数据库设置密码了。当下次打开这个数据库的时候，就会发现在打开数据库之前就会在屏幕上出现一个对话框，要求输入这个数据库的密码。只有输入正确的密码才能打开这个数据库，否则就不能打开这个数据库。



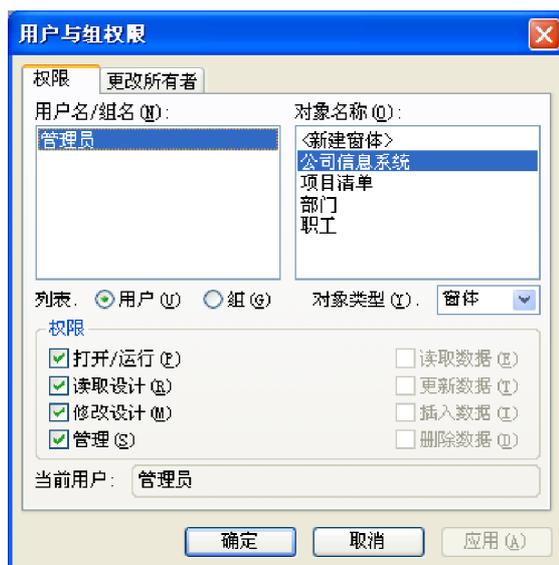
- 6 撤消密码也很简单，当给一个数据库已经设置了一个密码后，要想撤消这个密码，就再用独占方式打开这个数据库，然后选择“工具>安全>撤消数据库密码”命令，这时会在屏幕上弹出“撤消数据库密码”对话框，在“密码”文本框中输入正确密码后单击“确定”按钮即可。



## 设置用户与组的权限和账号

单纯的密码只能起到不能打开这个数据库的作用，要使数据库的使用者拥有不同的权限，即有的人可以修改数据库的内容，而有的人只能查看数据库的内容而不能修改。这就需要为不同的用户或某群用户组设置权限了。

选择“工具>安全>用户与组权限”命令，将会弹出“用户与组权限”对话框。在这个对话框中，可以更改不同用户对数据库或其中的某个对象的访问权限。假如想使某用户只能打开运行数据库的窗体，而不想要他能打开其他的表或查询，只需要先在该对话框的“权限”选项卡的“用户名组名”列表框中选择该用户，然后在“对象类型”下拉列表中选择“窗体”选项，再勾选“权限”选项区中的“打开/运行”复选框，单击“确定”按钮就可以使该用户在使用中只能看到窗体，而不能修改其他的数据内容了，也不能看到表或查询了。





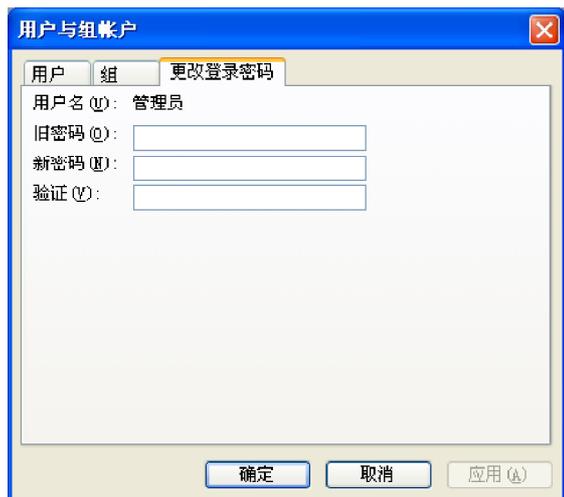
上面只讲了怎样设置用户或组的权限，但怎样才能将所需要的人都设有一定的权限，当然还必须给每个用户或组一个账号，这样才能进行管理，以便分配权限。要实现这个功能，则需选择“工具>安全>用户与组账户”命令，将会弹出“用户与组账户”对话框。



在这个对话框中有3个选项卡，“用户”用来在现有的组中管理用户的名称。在这个选项卡中单击“新建”按钮，在弹出的“新建用户组”对话框中添加新的用户。也可以单击“删除”按钮将组中的这个用户删去。单击“清除密码”按钮即可取消原来这个用户的密码。在“组成员关系”选项区中，通过中间的选择按钮就可以控制用户所在的组了。不同的组可以有不同的权限，经过这样设置以后，处于同一组中的用户就有了同样的权限了。



在“组”选项卡中可以单击“新建”按钮添加组，也可以单击“删除”按钮删除一个组。当一个用户用原来的密码登录到Access中以后，为了安全原因，可以在“更改登录密码”选项卡中修改自己的访问密码。当设置好这些以后，单击“确定”按钮就可以了。





## 编码/解码数据库

对于一个普通的Access数据库文件来说，由于可以使用一些工具避开输入它的密码，而直接读取里面的数据表，因此必须有一种方法将这种数据库文件进行编码，以防止非法访问情况，这样这个数据库才能算是安全的。

如果要对某个数据库文件进行编码，那么选择“工具>安全>编码/解码数据库”命令，就会弹出“数据库编码后另存为”对话框。在这个对话框输入编码以后保存的数据库文件名。完成后单击“保存”按钮就可以将这个数据库编码了。如果要解码这个数据库，只要按照该步骤再作一次，只是选取的是编码的数据库文件，新生成的是解码以后的文件。



## 3

## 保护数据库

Access

Access在单个文件中存放与一个数据库有关的所有内容的方法既有优点也有缺点。优点是一个系统只有一个文件，在硬盘上没有大量的文件，无需记住哪个程序与哪个数据库有关；缺点是若该文件损坏，则失去数据库的可能性就很大，原因是与一个数据库关联的所有内容存放在单个文件中，如果该文件因为某些原因而损坏，则数据库就不能正常运行，因此放置丢失数据的最保险的方法是进行定期的备份。

## 备份数据库

确保数据保存可靠性的一种传统、有效的方法是对数据库进行备份。备份的方法与在计算机中复制其他文件没有区别，具体操作步骤如下。

- (1) 确认已经关闭要进行备份的数据库。
- (2) 打开资源管理器。
- (3) 进入包含要备份数据库文件的子目录。
- (4) 在要备份的数据库文件上单击鼠标右键，在弹出的快捷菜单中选择“发送”命令。
- (5) 在“发送”级联菜单中选择“3.5英寸软盘”命令。操作系统将自动复制数据库文件到备份软盘上，其产生的备份文件与原始文件所占存储空间相同。
- (6) 用同样的方法，将SYSTEM.MDW文件进行备份。该文件保存着在Access中登记的用户信息，如果备份的是编码的数据库系统，丢失该文件将无法再打开该系统。

## 修复数据库

当数据库文件发生问题时，就需要对被破坏的数据进行修复，上节讲到对数据库进行备份的目的就是为了在出现问题后能够采取补救措施。使用备份文件进行修复虽然不能恢复备份后的工



作，但是已经最大限度地减小了损失，只需重做上次备份后的工作即可。

另外，在 Access 中提供了一种修复数据库的工具。当打开数据库时，如果该数据库出现问题，就会弹出提示对话框，可以单击“是”按钮来进行修复。如果修复成功将弹出提示修复成功的对话框。

在使用数据库时如果出现问题，可以选择“工具>数据库实用工具>压缩和修复数据库”命令来修复数据库。



### 压缩数据库

在对数据库中的一个或多个表删除或修改记录时，数据库文件可能会分成很多碎片。使数据库在硬盘上占据比其所需空间更大的磁盘空间，并且由于碎片位于数据库文件存放在硬盘上许多分散的簇中，使响应时间也变长了。虽然有些软件可以用来清除磁盘碎片，但是如果所需的只是对 Access 的一个数据库去碎片，就没有必要借助于软件。Access 系统本身提供了使用菜单命令压缩数据库的功能，可以实现数据库文件的高效存储。

Access 2003



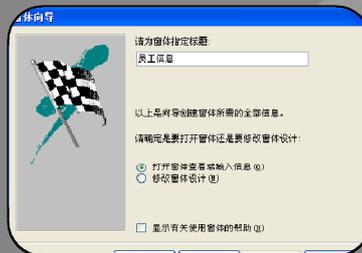
# part

# 18

## 进销存管理系统

- ◆ 系统总体设计
- ◆ 数据库设计
- ◆ 系统模块设计

应用进销存系统，企业能够把生产、销售和库存有机地结合起来，及时有效地把销售信息反馈到生产系统，使得企业做到合理生产、及时销售、减少库存量以及减少积压，相应会提高运作效率。



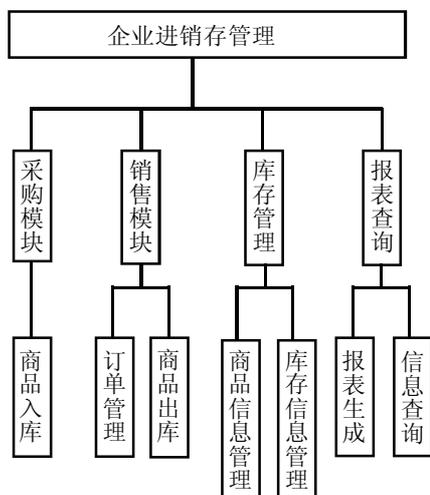
本章只介绍了本系统中包含数据表的逻辑设计，并没有详细介绍其设计过程，读者可根据前面介绍的方法进行操作。

## 1 系统总体设计

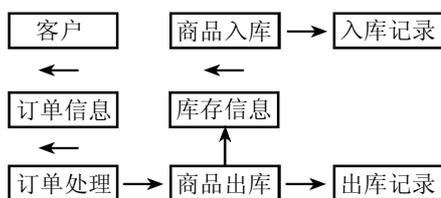
首先，进销存管理系统必须对企业当前库存信息进行记录，包括商品数量、商品类型、所属仓库和该仓库负责人等；商品出入库时，系统自动进行记录，同时修改相应的库存数值，系统还要提供一系列的报表，为企业管理者提供参考；最后，系统还应允许管理者对商品信息和各个仓库信息进行添加和修改，以适应不断变化的市场。

企业进销存管理系统功能模块可以分为4部分，每个部分根据实际应用又可以再细分成二级功能模块。

- 采购模块对产品的入库进行管理。
- 销售模块对订单的处理和产品的出库、发货进行管理。
- 库存管理模块对该企业的库存商品进行管理和配置，对商品的价格、数量等信息进行配置，对库存地点信息进行管理。
- 报表查询模块对进销存系统进行统计分析，并生成相应的报表。



整个系统的数据流程如右图所示。从图中可以看出，整个处理流程由客户的订单发起，企业进销存管理系统对用户的订单进行处理，确定商品出库的数量和地点，并且修改当前的库存信息，如果某个商品的库存量低于一定的数量，将会启动商品入库流程，并相应修改库存信息。



## 2 数据库设计

数据库的设计分为两个部分，首先对企业进销存管理系统的设计进行需求分析，确定在数据库中需要保存的信息，然后确定每个表的字段属性和主键等数据库信息。

### 数据库需求分析

根据系统设计和数据流程图可知，数据库中需要存放以下信息。



- 系统主切换面板和报表切换面板中的显示信息：显示名称、对应的处理方式、处理参数和关联显示的窗口或者面板。
  - 商品的基本信息：商品编号、商品名称、商品说明、单价和供应商编号。
  - 仓库信息：仓库编号、商品编号和库存数量。
  - 客户信息：客户编号、客户名称，送货地址和联系电话等。
  - 订单信息：订单编号、客户编号、商品编号和商品数量等。
  - 出库记录：记录编号、订单编号、出库日期、经手人员工编号、商品编号和出库数量等。
  - 入库记录：记录编号、入库日期、经手人员工编号、商品编号和仓库编号等。
  - 送货方式：送货方式编号和送货方式等。
  - 仓库编号：编号、地址和说明。
  - 员工信息：员工编号、所处仓库编号、联系电话和所属部门经理编号等。
- 供应商信息、编号、供应商名称和地址等。

## 数据表逻辑结构设计

(1) 商品表，该表存放商品的基本信息，结构如表 18-1 所示。

表 18-1 商品表

列名	数据类型	主键	必填字段
商品编号	数字	是	是
商品名称	文本	否	否
商品说明	文本	否	否
单价	货币	否	否
供应商编号	数字	否	否

(2) 仓库信息表，该表存放商品的库存信息，结构如表 18-2 所示。

表 18-2 库存信息表

列名	数据类型	主键	必填字段
仓库编号	数字	是	是
商品编号	数字	否	是
当前库存数量	数字	否	否

(3) 客户信息表，该表存放客户对应的联系信息，结构如表 18-3 所示。

表 18-3 客户信息表

列名	数据类型	主键	必填字段
客户编号	数字	是	否
客户名称	文本	否	否
送货地址	文本	否	否
联系人姓名	文本	否	否
邮政编码	文本	否	否
联系电话	文本	否	否
传真号码	文本	否	否



(4) 订单信息表, 该表存放订单的基本信息, 结构如表 18-4 所示。

表 18-4 订单信息表

列名	数据类型	主键	必填字段
订单编号	自动编号	是	是
客户编号	数字	否	否
商品编号	数字	否	否
数量	数字	否	否
订购日期	日期/时间	否	否
送货日期	日期/时间	否	否
运费	数字	否	否
当前状态	文本	否	否

(5) 出库记录表, 该表存放出库商品的记录, 包括商品本身的基本信息和对应的订单编号以及仓库编号, 结构如表 18-5 所示。

表 18-5 出库记录表

列名	数据类型	主键	必填字段
编号	自动编号	是	是
订单编号	数字	否	否
仓库编号	数字	否	是
出库日期	日期/时间	否	否
经手人员工编号	数字	否	否
商品编号	数字	否	否
数量	数字	否	否
送货方式	数字	否	否

(6) 入库记录表, 该表存放入库的记录, 结构如表 18-6 所示。

表 18-6 入库记录表

列名	数据类型	主键	必填字段
编号	自动编号	是	是
仓库编号	数字	否	是
入库日期	日期/时间	否	否
经手人员工编号	数字	否	否
商品编号	数字	否	否
数量	数字	否	否

(7) 送货方式表, 该表存放送货方式的信息, 结构如表 18-7 所示。

表 18-7 送货方式表

列名	数据类型	主键	必填字段
送货方式编号	自动编号	是	是
送货方式	文本	否	否



(8) 仓库编号表, 该表存放仓库及存放地点的信息, 结构如表 18-8 所示。

表 18-8 仓库信息表

列名	数据类型	主键	必填字段
编号	自动编号	是	是
地址	文本	否	否
说明	文本	否	否
负责人编号	数字	否	否

(9) 供应商表, 该表存放供应商的基本信息, 结构如表 18-9 所示。

表 18-9 供应商表

列名	数据类型	主键	必填字段
编号	自动编号	是	否
供应商名称	文本	否	否
地址	文本	否	否
联系人姓名	文本	否	否
邮政编码	文本	否	否
电话号码	文本	否	否
传真号码	文本	否	否

(10) 员工信息表, 该表存放员工的基本信息, 结构如表 18-10 所示。

表 18-10 供应商表

列名	数据类型	主键	必填字段
员工编号	数字	是	是
仓库编号	数字	否	否
姓名	文本	否	是
性别	文本	否	否
所属经理编号	数字	否	否
地址	文本	否	否
电子邮件	文本	否	否
电话	文本	否	否
附注	备注	否	否



## 数据库关系设计

教务管理系统数据库中的表之间是存在一定关系的。需要根据数据表的逻辑设计, 相应来建立数据库表之间的关系, 具体操作步骤如下。

1 在数据库窗口中，单击工具栏上的“关系”按钮，弹出关系窗口。如果数据库没有定义任何关系，将会显示一个空白的“关系”窗口以及“显示表”对话框。



2 在“显示表”对话框的“表”选项卡中双击选取要作为定义关系的表的名称，然后关闭“显示表”对话框，在“关系”窗口中会添加选中的表。这里将数据库中所有的表都添加到关系窗口中。



**提示**

如果在添加的表之间存在主键字段的关系，则系统会自动地在两个表间建立联接线。

3 定义其他表之间的关系。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，例如，将仓库信息表中的“仓库编号”字段拖到仓库编号表中的“编号”字段，将会弹出“编辑关系”对话框。



4 单击“创建”按钮即可关闭“编辑关系”对话框，并创建了这两个信息表之间的关系。其他表之间的关系均可按以上方法创建，最终创建的关系如下图所示。



## 3

## 系统模块设计

Access

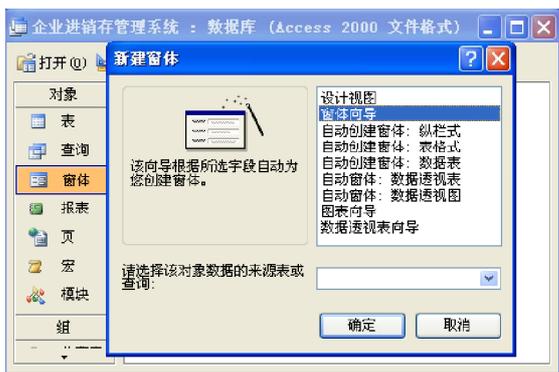
在进行系统模块设计时，需要先设计各功能窗体模块，最后再设计主切换面板，因为使用“数据库实用工具”设计主切换面板时需要涉及这些窗体模块。

## 库存管理模块设计

### 1. 仓库信息窗体的设计

利用仓库信息窗体可以查看和修改企业仓库信息，具体操作步骤如下。

- 1 在 Access 中新建一个窗体。在“新建窗体”对话框中选择“窗体向导”选项，单击“确定”按钮。
- 2 在弹出的“窗体向导”对话框的“表/查询”下拉列表中选择“仓库信息”表，单击按钮 >> 选择全部字段。



- 3 单击“下一步”按钮，系统弹出如下图所示的界面，在其中选择“纵栏表”布局。
- 4 单击“下一步”按钮，系统弹出如下图所示的界面，在其中选择“标准”样式。



- 单击“下一步”按钮，系统弹出如下图所示界面，在其中的文本框中，输入“仓库信息”，选中“打开窗体查看或输入信息”单选按钮。



- 单击“完成”按钮，系统弹出如下图所示的对话框，可以向其中添加仓库信息。



## 2. 商品信息窗体的设计

利用商品信息窗体可以对商品信息进行设置，它的建立过程和“仓库信息”窗体相似，读者按着前面的方法操作即可，这里不再过多介绍。

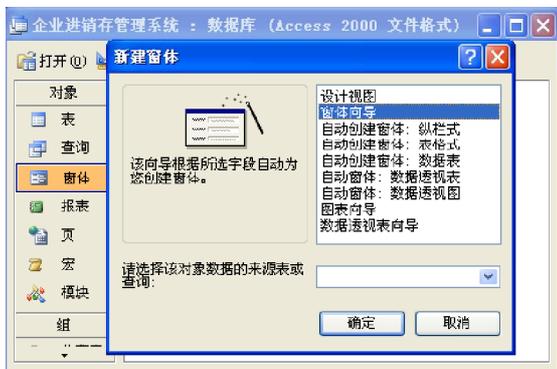


# 采购管理模块设计

## 1. 员工信息窗体的设计

利用员工信息窗体可以对企业员工信息进行查看和修改，具体操作步骤如下。

- 在 Access 中新建一个窗体。在“新建窗体”对话框中选择“窗体向导”选项，单击“确定”按钮。





- 在弹出的“窗体向导”对话框的“表/查询”下拉列表中选择“员工信息”表，单击 >>> 按钮选择全部字段。



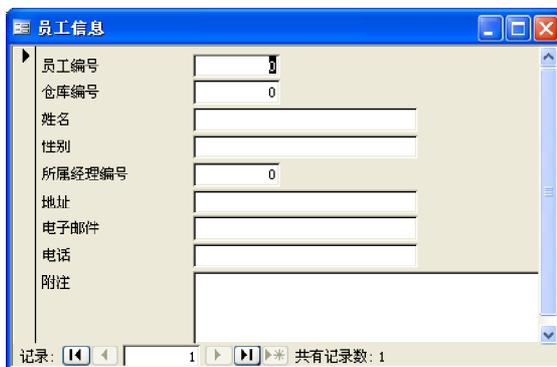
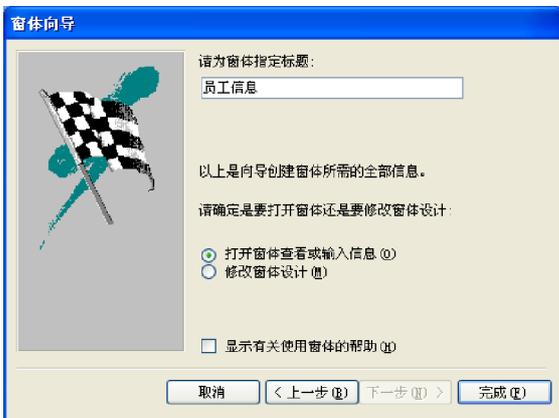
- 单击“下一步”按钮，系统弹出如下图所示的界面，在其中选择“纵栏表”布局。

- 单击“下一步”按钮，系统弹出如下图所示的界面，在其中选择“标准”样式。



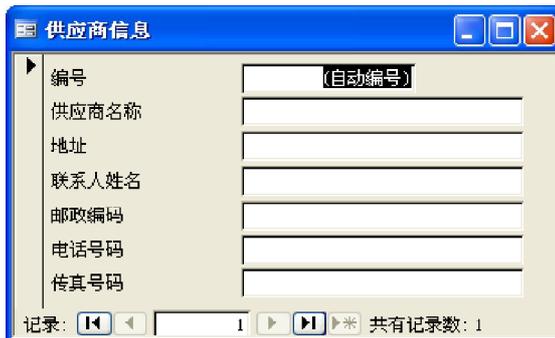
- 单击“下一步”按钮，系统弹出如下图所示的界面，在其中的文本框中输入“员工信息”，选中“打开窗体查看或输入信息”单选按钮。

- 单击“完成”按钮，系统弹出如下图所示的对话框，可以向其中添加员工信息。



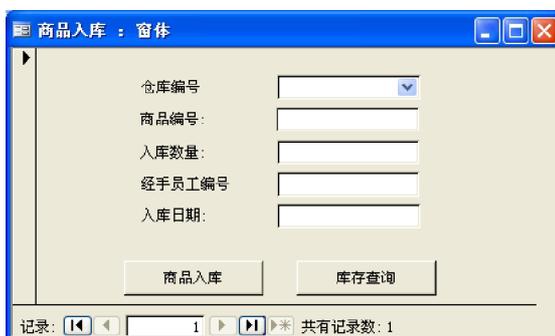
## 2. 供应商信息窗体的设计

利用供应商窗体可以添加或修改供应商信息。它的建立过程和“员工信息”窗体类似。

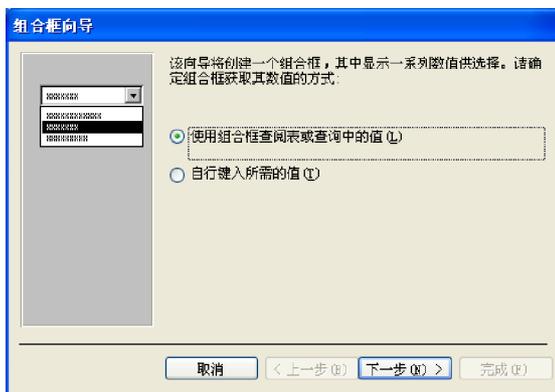


## 3. 商品入库窗体的设计

利用商品入库窗体可以添加商品入库记录或修改当前商品库存信息。如果该商品是第一次入库，则在库存信息中添加相应记录；如果已经存在，则修改相应数量。具体操作步骤如下。



- 1 在 Access 中新建一个空白窗体。在“新建窗体”对话框中选择“设计视图”选项，单击“确定”按钮。单击工具箱上面的“组合框”按钮，在窗体上添加一个组合框，系统弹出“组合框向导”对话框。



- 2 保持默认设置单击“下一步”按钮，弹出选择表或查询界面，在表或查询列表框中选择“表：仓库信息”选项。





- 3 单击“下一步”按钮，弹出选择字段界面，从“可用字段”列表框中选择“仓库编号”字段，将其添至“选定字段”列表框中，作为选定字段。



- 4 单击“下一步”按钮，在弹出的界面中用户可以指定排序的次序，这里用户自行进行设置即可。单击“下一步”按钮，弹出指定列宽度界面。



- 5 单击“下一步”按钮，弹出为组合框指定标签界面。



- 6 输入组合框的标签为“仓库编号”，单击“完成”按钮，则窗体“设计”视图中添加了组合框控件。



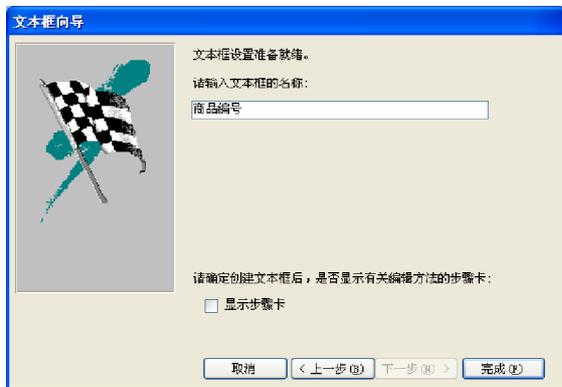
- 7 单击工具箱上的“文本框”按钮，系统弹出“文本框向导”对话框。



- 8 单击“下一步”按钮，在界面的“输入法模式”下拉列表中选择“输入法关闭”选项。



- 9 单击“下一步”按钮，在当前界面的文本框中输入“商品编号”，单击“完成”按钮。



- 10 重复步骤7~步骤9的操作步骤，再添加一个文本框，设置标签和文本框名称为“入库数量”。



- 11 继续添加一个文本框，设置标签名为“经手员工编号”，设置文本框名称为“员工编号”。



- 12 继续添加一个文本框，设置标签和文本框名称为“入库日期”。



- 13 单击工具箱中的“命令按钮”按钮，单击窗体空白处，将会弹出“命令按钮向导”对话框。单击“取消”按钮，在属性对话框中将按钮的“名称”和“标题”属性都改为“商品入库”。



- 14 用同样的方法再添加一个命令按钮，在属性对话框中将按钮的“名称”和“标题”属性都改为“库存查询”。





下面为两个按钮添加事件代码。

(1) “商品入库”按钮的功能是：当用户设置相应的仓库编号，输入商品编号、入库数目、经手人员工编号和入库日期后，单击该按钮时，如果当前系统中存在该商品的信息，则添加入库记录并修改相应库存信息，否则提示用户先到商品信息窗体中添加该商品信息。事件代码如下。

```
Private Sub 商品入库_Click()
    Dim sql As String
    Dim rst As ADODB.Recordset
    Dim number As Integer
    ' 以下程序中的Me表示"商品入库"窗体
    If IsNull(Me![仓库编号]) Then ' 如果没有选择仓库编号
        MsgBox "请选择仓库"
        DoCmd.GoToControl "仓库编号"
    ElseIf IsNull(Me![商品编号]) Then ' 如果没有选择商品编号
        MsgBox "请输入商品编号"
        DoCmd.GoToControl "商品编号"
    ElseIf IsNull(Me![入库数量]) Then
        MsgBox "请输入数量"
        DoCmd.GoToControl "入库数量"
    ElseIf IsNull(Me![员工编号]) Then
        MsgBox "请输入员工编号"
        DoCmd.GoToControl "员工编号"
    ElseIf IsNull(Me![入库日期]) Then
        MsgBox "请输入日期"
        DoCmd.GoToControl "入库日期"
    Else
        sql = "select * from 商品 where"
        sql = sql & " [商品编号]=" & Me![商品编号]
        Set rst = New ADODB.Recordset
        rst.ActiveConnection = CurrentProject.Connection
        rst.CursorType = adOpenDynamic
        rst.LockType = adLockOptimistic
        rst.Open sql
        If Not (rst.EOF) Then
            ' 添加入库记录
            sql = "select * from 入库记录"
            rst.Close
            Set rst = Nothing
            Set rst = New ADODB.Recordset
            rst.ActiveConnection = CurrentProject.Connection
            rst.CursorType = adOpenKeyset
            rst.LockType = adLockOptimistic
            rst.Open sql
            With rst
```



```
        .AddNew
        ! 仓库编号 = Me![仓库编号]
        ! 商品编号 = Me![商品编号]
        ! 数量 = Me![入库数量]
        ! 经办员工编号 = Me![员工编号]
        ! 入库日期 = Me![入库日期]
    End With
rst.Close
Set rst = Nothing
' 添加或者修改库存信息
sql = "select * from 库存信息 where [仓库编号]=" & Me![仓库编号]
sql = sql & " and [商品编号]=" & Me![商品编号]
Set rst = New ADODB.Recordset
rst.ActiveConnection = CurrentProject.Connection
rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.Open sql
If Not (rst.EOF) Then
    number = rst!当前库存数量
    number = number + Me![入库数量]
    rst!当前库存数量 = number
    rst.Close
    Set rst = Nothing
    sql = "当前库存数量为：" & number
    MsgBox sql
    Exit Sub
Else
    With rst
        .AddNew
        ! 仓库编号 = Me![仓库编号]
        ! 商品编号 = Me![商品编号]
        ! 数量 = Me![入库数量]
    End With
    Exit Sub
    Me.Visible = False
End If
Else
    rst.Close
    Set rst = Nothing
    MsgBox "系统中没有该商品的信息，请先添加商品详细信息"
    Exit Sub
    Me.Visible = False
End If
End If
End Sub
```



(2) “库存查询”按钮的功能是：用户选择仓库编号，单击该按钮，如果当前仓库中存在该商品的信息，显示库存信息，否则提示用户库存为空。事件代码如下。

```
Private Sub 库存查询_Click()
    Dim sql As String
    Dim rst As ADODB.Recordset
    Dim number As Integer
    If IsNull(Me![仓库编号]) Then ' 如果没有选择仓库编号
        MsgBox "请选择仓库"
        DoCmd.GoToControl "仓库编号"
    ElseIf IsNull(Me![商品编号]) Then ' 如果没有选择商品编号
        MsgBox "请输入商品编号"
        DoCmd.GoToControl "商品编号"
    Else
        sql = "select * from 库存信息 where [仓库编号]=" & Me![仓库编号]
        sql = sql & " and [商品编号]=" & Me![商品编号]
        Set rst = New ADODB.Recordset
        rst.ActiveConnection = CurrentProject.Connection
        rst.CursorType = adOpenDynamic
        rst.LockType = adLockOptimistic
        rst.Open sql
        If Not (rst.EOF) Then
            number = rst!当前库存数量
            rst.Close
            Set rst = Nothing
            sql = "当前库存数量为：" & number
            MsgBox sql
            Exit Sub
        Else
            rst.Close
            Set rst = Nothing
            MsgBox "当前仓库中没有该商品库存"
            Exit Sub
        End If
    End Sub
End Sub
```



## 销售管理模块设计

销售管理模块（即商品出库）可以添加商品出库记录或修改当前商品库存信息，如果当前库存量大于出库数量，则修改相应的数量；否则提示出错，提醒用户库存不足，以及当前可以出库的数量，用户如果将出库的数量修改为和当前库存一致，则删除相应的库存信息。



表18-11所示为商品出库窗体的属性值，根据这个属性表来设计窗体。

表 18-11 “商品出库”窗体属性值

对象名称	属性名称	属性值
Command1	标题	商品出库
	单击	=HandleButtonClick(1)
Command2	标题	库存查询
	单击	=HandleButtonClick(2)
Command3	标题	订单查询
	单击	=HandleButtonClick(3)

下面为窗体中的3个按钮添加代码。

(1) “库存查询”按钮的代码和“商品入库”中的代码相同，详见源文件。

(2) 单击“订单查询”按钮，系统返回该订单的基本信息，例如是否已完成、订单数量等。具体代码如下。

```
Private Sub 订单查询_Click()  
    Dim sql As String  
    Dim rst As ADODB.Recordset  
    Dim number As Integer  
    If IsNull(Me![订单编号]) Then  
        MsgBox "请输入订单编号"  
        DoCmd.GoToControl "订单编号"  
    Else  
        sql = "select * from 订单记录 where [订单编号]=" & Me![订单编号]  
        Set rst = New ADODB.Recordset  
        rst.ActiveConnection = CurrentProject.Connection  
        rst.CursorType = adOpenDynamic  
        rst.LockType = adLockOptimistic  
        rst.Open sql  
        If Not (rst.EOF) Then  
            number = rst!数量  
            rst.Close  
            Set rst = Nothing  
            sql = "订购数量为：" & number  
            MsgBox sql  
            Exit Sub  
        Else  
            rst.Close  
            Set rst = Nothing  
            MsgBox "没有该编号的订单"  
            Exit Sub  
        End If  
    End If  
End Sub
```



(3) 单击“商品出库”按钮，系统添加相应的出库记录并修改相应的库存信息，如果当前系统中没有此商品的库存信息，则不添加出库记录，并提示用户。具体代码如下。

```
Private Sub 商品出库_Click()
    Dim sql As String
    Dim rst As ADODB.Recordset
    Dim number As Integer

    If IsNull(Me![仓库编号]) Then
        MsgBox "请选择仓库"
        DoCmd.GoToControl "仓库编号"
    ElseIf IsNull(Me![商品编号]) Then
        MsgBox "请输入商品编号"
        DoCmd.GoToControl "商品编号"
    ElseIf IsNull(Me![出库数量]) Then
        MsgBox "请输入数量"
        DoCmd.GoToControl "出库数量"
    ElseIf IsNull(Me![员工编号]) Then
        MsgBox "请输入员工编号"
        DoCmd.GoToControl "员工编号"
    ElseIf IsNull(Me![出库日期]) Then
        MsgBox "请输入日期"
        DoCmd.GoToControl "出库日期"
    ElseIf IsNull(Me![订单编号]) Then
        MsgBox "请输入订单编号"
        DoCmd.GoToControl "订单编号"
    ElseIf IsNull(Me![送货方式]) Then
        MsgBox "请输入送货方式"
        DoCmd.GoToControl "送货方式"
    Else
        sql = "select * from 商品 where"
        sql = sql & " [商品编号]=" & Me![商品编号]
        Set rst = New ADODB.Recordset
        rst.ActiveConnection = CurrentProject.Connection
        rst.CursorType = adOpenDynamic
        rst.LockType = adLockOptimistic
        rst.Open sql
        If Not (rst.EOF) Then
            ' 修改库存信息
            sql = "select * from 库存信息 where [仓库编号]=" & Me![仓库编号]
            sql = sql & " and [商品编号]=" & Me![商品编号]
            Set rst = New ADODB.Recordset
            rst.ActiveConnection = CurrentProject.Connection
            rst.CursorType = adOpenDynamic
            rst.LockType = adLockOptimistic
        End If
    End If
End Sub
```



```
rst.Open sql
If Not (rst.EOF) Then
    number = rst!当前库存数量
    number = number - Me![出库数量]
    rst!当前库存数量 = number
    rst.Close
    Set rst = Nothing
    sql = "当前库存数量为：" & number
    MsgBox sql
    Exit Sub
Else
    rst.Close
    Set rst = Nothing
    MsgBox "没有该商品的库存信息,不能出库"
    Exit Sub
End If
' 添加出库记录
sql = "select * from 出库记录"
rst.Close
Set rst = Nothing
Set rst = New ADODB.Recordset
rst.ActiveConnection = CurrentProject.Connection
rst.CursorType = adOpenKeyset
rst.LockType = adLockOptimistic
rst.Open sql
With rst
    .AddNew
    !订单编号 = Me![订单编号]
    !仓库编号 = Me![仓库编号]
    !出库日期 = Me![出库日期]
    !商品编号 = Me![商品编号]
    !经办员工编号 = Me![员工编号]
    !数量 = Me![出库数量]
    !送货方式 = Me![送货方式]
End With
rst.Close
Set rst = Nothing
Else
    rst.Close
    Set rst = Nothing
    MsgBox "系统中没有该商品的信息,请先添加商品详细信息"
    Exit Sub
    Me.Visible = False
End If
```



```
End If
End Sub
```

## 🔑 订单处理模块设计

利用订单处理窗体可以添加订单信息，如右图所示。



表 18-12 为订单处理窗体主要的属性值，根据这个属性表来设计窗体。

表 18-12 “订单处理”窗体属性值

对象名称	属性名称	属性值
Command1	标题	接受订单
	单击	=HandleButtonClick(1)

下面为窗体中的“接受订单”按钮添加如下代码。

```
Private Sub 接受订单_Click()
    Dim sql As String
    Dim rst As ADODB.Recordset
    Dim number As Integer
    If IsNull(Me![客户名称]) Then
        MsgBox "请选择客户名称"
        DoCmd.GoToControl "仓库编号"
    ElseIf IsNull(Me![商品编号]) Then
        MsgBox "请输入商品编号"
        DoCmd.GoToControl "商品编号"
    ElseIf IsNull(Me![数量]) Then
        MsgBox "请输入数量"
        DoCmd.GoToControl "入库数量"
    ElseIf IsNull(Me![员工编号]) Then
        MsgBox "请输入员工编号"
        DoCmd.GoToControl "员工编号"
    ElseIf IsNull(Me![送货日期]) Then
        MsgBox "请输入送货日期"
        DoCmd.GoToControl "送货日期"
    ElseIf IsNull(Me![送货方式]) Then
        MsgBox "请输入送货方式"
        DoCmd.GoToControl "送货方式"
    Else

```



```
sql = "select * from 商品 where"  
sql = sql & " [商品编号]=" & Me![商品编号]  
Set rst = New ADODB.Recordset  
rst.ActiveConnection = CurrentProject.Connection  
rst.CursorType = adOpenDynamic  
rst.LockType = adLockOptimistic  
rst.Open sql  
If Not (rst.EOF) Then  
    ' 添加入库记录  
    sql = "select * from 入库记录"  
    rst.Close  
    Set rst = Nothing  
    Set rst = New ADODB.Recordset  
    rst.ActiveConnection = CurrentProject.Connection  
    rst.CursorType = adOpenKeyset  
    rst.LockType = adLockOptimistic  
    rst.Open sql  
    With rst  
        .AddNew  
        ! 仓库编号 = Me![仓库编号]  
        ! 商品编号 = Me![商品编号]  
        ! 数量 = Me![入库数量]  
        ! 经办员工编号 = Me![员工编号]  
        ! 入库日期 = Me![入库日期]  
    End With  
    rst.Close  
    Set rst = Nothing  
Else  
    rst.Close  
    Set rst = Nothing  
    MsgBox " 系统中没有该商品的信息，请先添加商品详细信息 "  
    Exit Sub  
    Me.Visible = False  
End If  
End If  
End Sub
```



## 报表查询窗体模块设计

利用报表查询窗体可以根据设定的查询条件对订单、出入库情况进行查询，最后生成相应报表。





在设计报表查询窗体之前，需要先设计该模块中的 3 个查询和 3 个报表。

### 1. “出库”查询

该查询返回指定时间段内的商品出库情况，具体设计步骤如下。

- 在数据库窗口的“查询”界面中，双击“在设计视图中创建查询”选项，在“显示表”对话框的“表”选项卡的列表框中选择“出库记录”、“客户信息”、“订单信息”和“商品”表。



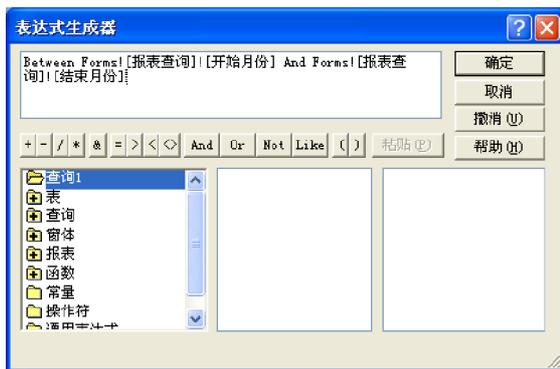
- 单击“添加”按钮，将选择的表添加到查询设计窗口中，然后单击“关闭”按钮。



- 向查询设计窗口中添加以下字段：“出库记录”表中的编号、商品编号、出库日期、仓库编号和订单编号字段，“商品”表中的商品名称字段，“客户信息”表中的客户编号和客户名称字段。



- 将光标定位在查询设计窗口的“出库日期”文本框中，单击鼠标右键，在弹出的快捷菜单中选择“生成器”命令。在系统弹出的“表达式生成器”对话框中设置以下条件：Between Forms! [报表查询]! [开始月份] And Forms! [报表查询]! [结束月份]。



- 5 单击“确定”按钮，完成查询的设计，单击工具栏上的“保存”按钮，指定名称为“出库”后保存。



- 6 按照相同的方法设计入库查询，最终的查询设计窗口如下图所示。



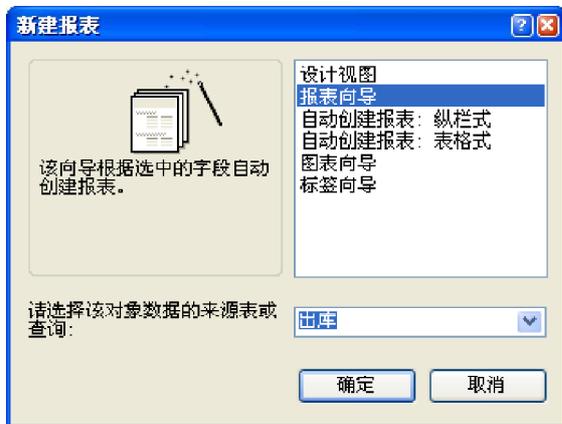
- 7 按照相同的方法设计订单查询，最终的查询设计界面如右图所示。



## 2. “出库查询”报表

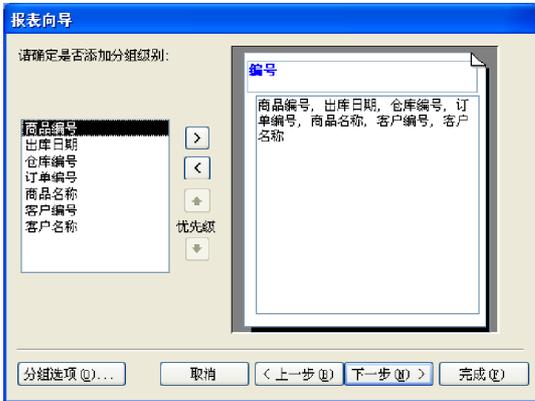
该报表可以对指定时间段内的商品出库情况进行汇总，具体设计步骤如下。

- 1 在数据库窗口下的“报表”界面中，单击“新建”按钮，弹出“新建报表”对话框，在其中选择“报表向导”选项。
- 2 在弹出的“报表向导”对话框的“表/查询”下拉列表中选择“出库”选项。单击“下一步”按钮，弹出表和字段选择界面。

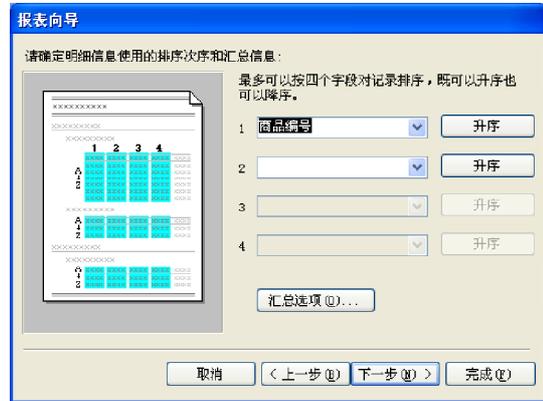




- 3 在“可用字段”列表框中选择所有的字段。单击“下一步”按钮，弹出如下图所示的界面。



- 4 选择“编号”作为分组级别。单击“下一步”按钮，弹出如下图所示的界面。



- 5 选择按照“商品编号”排序。单击“下一步”按钮，弹出如下图所示的界面。



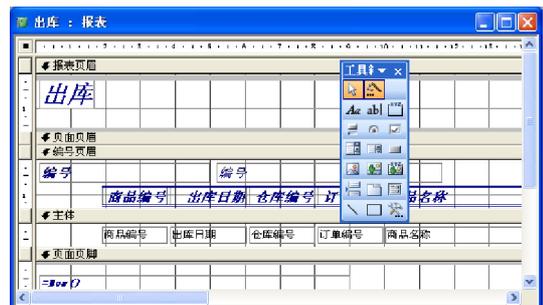
- 6 设置“布局”为“分级显示1”形式，“方向”为“横向”，勾选“调整字段宽度”复选框。单击“下一步”按钮，弹出如下图所示的界面。



- 7 设置样式为“组织”形式。单击“下一步”按钮，弹出如下图所示的界面。



- 8 指定标题为“出库”，选中“修改报表设计”单选按钮，单击“完成”按钮，在设计视图中调整报表各个显示框的大小的位置，最终效果如下图所示。



按照和以上相同的方法分别设计“入库查询”和“订单查询”报表。

### 3. “报表查询”窗体的设计

表 18-13 所示为商品出库窗体的属性值。

表 18-13 “报表查询”窗体属性值

对象名称	属性名称	属性值
Command1	标题	订单查询
	单击	=HandleButtonClick(1)
Command2	标题	出库查询
	单击	=HandleButtonClick(2)
Command3	标题	入库查询
	单击	=HandleButtonClick(3)

根据属性表设计窗体，添加控件后的窗体设计视图如右图所示。



下面为窗体中的 3 个按钮添加代码。

#### (1) 出库查询

“出库查询”按钮的功能是根据设定的开始和结束日期给出这段时期内的“出库查询”报表。



#### (2) 入库查询

“入库查询”按钮的功能是根据设定的开始和结束日期给出这段时期内的“入库查询”报表。





### (3) 订单查询

“订单查询”按钮的功能是根据设定的开始和结束日期给出这段时期内的“订单查询”报表。



### 主切换面板窗体模块设计

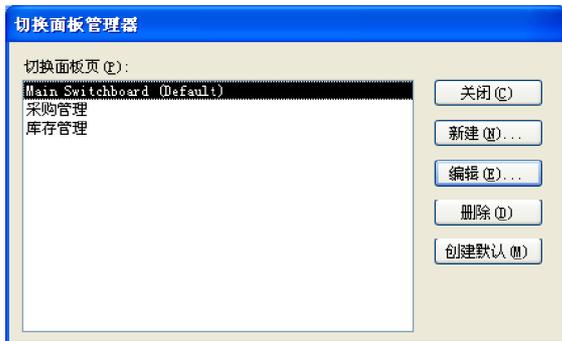
在设计完所有的功能窗体后，最后需要一个主窗体来对这些窗体进行管理，Access 提供的数据库实用工具——切换面板管理器能够完成这项功能，它能够将数据库中的窗体或报表根据功能模块划分进行分层管理。

右图所示为系统的主切换面板。

实现的效果为单击“库存管理”和“采购管理”按钮可以相应打开“库存管理”和“采购管理”子切换面板。单击其他功能按钮可以打开前面设计的窗体。下面来设计切换面板。



- 1 选择“工具>数据库实用工具>切换面板管理器”命令，系统弹出“切换面板管理器”对话框。



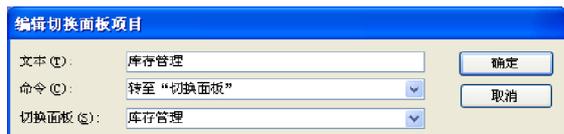
- 2 在该对话框中单击“编辑”按钮，弹出“编辑切换面板页”对话框。



- 3 在该对话框中单击“新建”按钮，弹出“编辑切换面板项目”对话框。



- 4 在该对话框的“文本”文本框中输入“库存管理”，因为“库存管理”模块下包含仓库信息和商品信息管理，所以这里在“命令”下拉列表中选择“转至‘切换面板’”选项，然后在“切换面板”下拉列表中选择“库存管理”子切换面板。



- 5 单击“确定”按钮，同样的方法可以在主切换面板中设置“采购管理”、“销售管理”、“订单处理”和“报表查询”项目。



- 6 在主切换面板中还可以添加一个退出系统的按钮。



#### 提示

在本系统的“库存管理”和“采购管理”子切换面板中需要分别添加一个返回主切换面板的按钮，这样便于用户进行操作。

Access 2003



# part

# 19

## 教务管理系统

- ◆ 系统总体设计
- ◆ 数据库设计
- ◆ 系统模块设计

随着当前经济的发展，社会的信息化程度越来越高，学校在信息化的过程中担任了很重要的角色，其信息化速度发展也比较快。本章开发的系统用来管理学校教务的各种信息数据。





在本章中只介绍了本系统中包含数据表的逻辑设计，并没有详细介绍其设计过程，读者可根据前面介绍的方法进行操作。

## 1

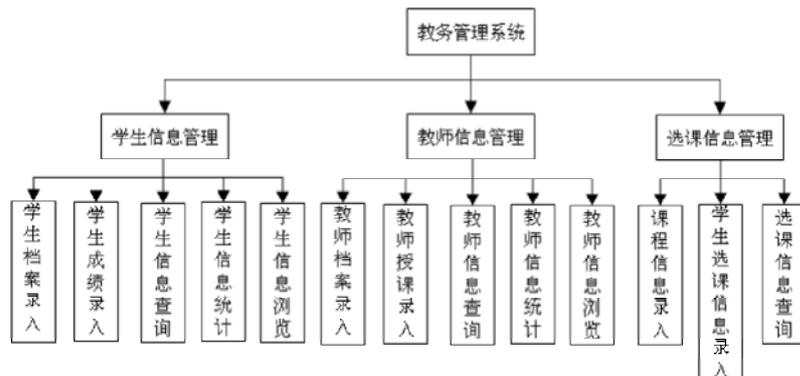
## 系统总体设计

Access

本系统实现与学生、教师、课程及成绩等相关的各种信息的系统化、规范化和自动化效果。本实例中的教务管理系统主要完成的功能如下。

- 学生信息管理，管理学生的基本档案和成绩，可以浏览、增加、修改和删除学生档案信息及成绩信息。
- 教师信息管理，管理教师的基本信息以及教师的授课信息，可以浏览、增加、修改和删除教师信息及其相应信息。
- 选课信息管理，管理课程信息录入、学生选课信息录入以及学生选课信息查询。

根据需求分析，建立如下图所示的教务管理系统的逻辑模型。



## 2

## 数据库设计

Access



## 数据表逻辑结构设计

## 1. 学生档案表的设计

学生档案表用于记录学生的基本信息，包括学号、姓名、性别及出生日期等，其逻辑结构如表 19-1 所示。

表 19-1 学生档案表

字段名	字段类型	格式	索引	说明
学号	文本	标准	有	学生的编号
姓名	文本	标准	无	学生姓名
性别	文本	标准	无	学生性别
出生日期	日期/时间	标准	无	学生的出生日期
政治面貌	文本	标准	无	学生的政治面貌



(续表)

字段名	字段类型	格式	索引	说明
班级编号	文本	标准	无	学生所属班级的编号
毕业学校	文本	标准	无	学生毕业的学校

## 2. 课程名表的设计

课程名表用于学校所开设的课程信息，包括课程名称及相应的学分等，其逻辑结构如表 19-2 所示。

表 19-2 课程名表

字段名	字段类型	格式	索引	说明
课程编号	文本	标准	有	课程的编号
课程名	文本	标准	无	课程的名称
课程类别	文本	标准	无	课程所属类别
学分	数字	标准	无	课程对应学分

## 3. 学生选课信息表的设计

学生选课信息表用于记录每个学生所选课程的详细信息，其逻辑结构如表 19-3 所示。

表 19-3 学生选课信息表

字段名	字段类型	格式	索引	说明
选课ID	文本	标准	有	在学生所选课程中的编号
课程编号	文本	标准	无	课程的编号
学号	文本	标准	有	学生的编号

## 4. 教师档案表的设计

教师档案表用于记录教师的基本信息，包括学历、职称以及所在院系等，其逻辑结构如表 19-4 所示。

表 19-4 教师档案表

字段名	字段类型	格式	索引	说明
教师编号	文本	标准	有	教师的编号
姓名	文本	标准	无	教师的姓名
性别	文本	标准	无	教师的性别
工作时间	日期/时间	标准	无	教师工作的时间
政治面貌	文本	标准	无	教师的政治面貌
学历	文本	标准	无	教师的最高学历
职称	文本	标准	无	教师的职称
系别	文本	标准	无	教师所在的院系
联系电话	数字	标准	无	教师的联系电话

## 5. 教师授课信息表的设计

教师授课信息表用于记录每个教师所授课程的信息，包括课程在哪一学年、哪一学期讲授，以及具体的授课地点和授课时间等，其逻辑结构如表 19-5 所示。



表 19-5 教师授课信息表

字段名	字段类型	格式	索引	说明
授课ID	文本	标准	有	在教师所授课程中的编号
课程编号	文本	标准	无	课程的编号
教师编号	文本	标准	无	教师的编号
班级编号	日期/时间	标准	无	班级的编号
学年	文本	标准	无	课程开设的学年
学期	数字	标准	无	课程开设的学期
学时	数字	标准	无	课程的学时
授课地点	文本	标准	无	课程的上课地点
授课时间	日期/时间	标准	无	课程的上课时间

## 6. 学生成绩表的设计

学生成绩表用于记录学生所选课程的成绩信息，其逻辑结构如表 19-6 所示。

表 19-6 教师授课信息表

字段名	字段类型	格式	索引	说明
成绩ID	文本	标准	有	学生所选的课程成绩的编号
学号	文本	标准	无	学生的编号
学年	文本	标准	无	课程开设的学年
学期	数字	标准	无	课程开设的学期
课程编号	文本	标准	无	课程的编号
成绩	文本	标准	无	某门课程的成绩



## 数据库关系设计

教务管理系统数据库中的表之间是存在一定关系的。根据数据表的逻辑设计，建立数据库表之间的关系，具体操作步骤如下。

- 1 在数据库窗口中，单击工具栏上的“关系”按钮，弹出“关系”窗口。如果数据库没有定义任何关系，将会显示一个空白的“关系”窗口，并且弹出“显示表”对话框。
- 2 在“显示表”对话框的“表”选项卡中双击选取要作为定义关系的表的名称，然后关闭“显示表”对话框，在“关系”窗口中会添加选中的表。这里将数据库中所有的表都添加到“关系”窗口中。



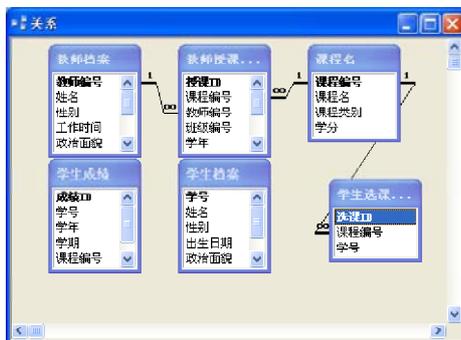


**提示**

如果添加的表之间存在主键字段的关系，则系统会自动地在两个表间建立联接线。这里的课程名表和学生成绩表中的“课程编号”字段之间存在主键字段的关系。

3 定义其他表之间的关系。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，例如，将教师档案表中的“教师编号”字段拖到教师授课信息表中的“教师编号”字段，会出现“编辑关系”对话框。

4 单击“创建”按钮关闭“编辑关系”对话框，就创建了教师档案表和教师授课信息表之间的关系。教师档案表和教师授课表、教师授课表和课程名表之间的关系可按以上方法创建，最终创建的关系如下图所示。



## 3 系统模块设计

教务管理系统含有 3 个功能模块：学生信息管理模块、教师信息管理模块及选课信息管理模块。首先设计这 3 个功能模块，然后再设计一个切换面板来管理它们。下面首先介绍这些功能模块的设计过程。

### 学生信息管理模块的设计

根据系统功能模块图可知，学生信息管理模块包含学生档案录入、学生成绩录入、学生相关信息查询、学生相关信息统计以及学生相关信息浏览。下面分别介绍这些子功能窗体的设计过程。

#### 1. 学生档案录入

学生档案录入窗体主要用于添加新的学生信息，在该窗体的设计视图中包含 8 个标签控件、5 个文本框控件、2 个矩形控件、2 个组合框控件和 5 个按钮控件，其中的主要控件属性如表 19-7 所示。



表 19-7 “学生档案录入”窗体属性值

对象名称	属性名称	属性值
标签1 (标签控件)	标题	录入学生档案信息
命令19 (按钮控件)	标题 单击	上一记录 [事件过程]
命令13 (按钮控件)	标题 单击	下一记录 [事件过程]
命令14 (按钮控件)	标题 单击	添加记录 [事件过程]
命令15 (按钮控件)	标题 单击	保存记录 [事件过程]
命令16 (按钮控件)	标题 单击	退出 [事件过程]

下面定义窗体中各个功能按钮的事件过程。

(1) “上一记录”按钮(命令19)事件代码。“命令19”按钮的功能是,单击“上一记录”按钮时,“学生档案录入”窗体中显示上一条学生记录。

```
Private Sub 命令19_Click()
    On Error GoTo Err_命令19_Click

    DoCmd.GoToRecord , , acPrevious

    Exit_命令19_Click:
    Exit Sub

    Err_命令19_Click:
    MsgBox Err.Description
    Resume Exit_命令19_Click

End Sub
```

(2) “下一记录”按钮(命令13)事件代码。“命令13”按钮的功能是,单击“下一记录”按钮时,“学生档案录入”窗体中显示下一条学生记录。

```
Private Sub 命令13_Click()
    On Error GoTo Err_命令13_Click

    DoCmd.GoToRecord , , acNext

    Exit_命令13_Click:
    Exit Sub

    Err_命令13_Click:
    MsgBox Err.Description
    Resume Exit_命令13_Click

End Sub
```

(3) “添加记录”按钮(命令14)事件代码。“命令14”按钮的功能是,单击“添加记录”按钮时,“学生档案录入”窗体中各组合框和文本框显示为空,这时可以输入新的学生信息。

```
Private Sub 命令14_Click()
    On Error GoTo Err_命令14_Click

    DoCmd.GoToRecord , , acNewRec

    Exit_命令14_Click:
    Exit Sub

    Err_命令14_Click:
    MsgBox Err.Description
    Resume Exit_命令14_Click

End Sub
```

(4) “保存记录”按钮(命令15)事件代码。“命令15”按钮的功能是,单击“保存记录”按钮时,可以将添加到“学生档案录入”窗体中的学生信息保存到学生档案表中。

```
Private Sub 命令15_Click()
    On Error GoTo Err_命令15_Click

    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSav

    Exit_命令15_Click:
    Exit Sub

    Err_命令15_Click:
    MsgBox Err.Description
    Resume Exit_命令15_Click

End Sub
```



(5) “退出”按钮（命令16）事件代码。  
“命令16”按钮的功能是，单击“退出”按钮时，可以退出“学生档案录入”窗体。

```

命令16 Click
Private Sub 命令16_Click()
On Error GoTo Err_命令16_Click

    DoCmd.Close

Exit_命令16_Click:
Exit Sub

Err_命令16_Click:
MsgBox Err.Description
Resume Exit_命令16_Click

End Sub
    
```

## 2. 学生成绩录入

学生成绩录入窗体主要用于添加新的学生成绩信息，在该窗体的设计视图中包含标签、文本框、矩形、子窗体/子报表和按钮等控件，其中的主要控件属性如表19-8所示。



表 19-8 “录入学生选课成绩”窗体属性值

对象名称	属性名称	属性值
标签7（标签控件）	标题	录入学生选课成绩
学生成绩表 子窗体 (子窗体/子报表控件)	源对象	学生成绩表 子窗体
命令14（按钮控件）	标题 单击	添加记录 [事件过程]
命令15（按钮控件）	标题 单击	保存记录 [事件过程]
命令16（按钮控件）	标题 单击	退出 [事件过程]

下面来定义窗体中各个功能按钮的事件过程。

(1) “添加记录”按钮（命令14）事件代码。“命令14”按钮的功能是，单击“添加记录”按钮时，录入学生选课成绩窗体中各组合框和文本框显示为空，这时可以输入新的学生选课成绩信息。

```

命令14 Click
Private Sub 命令14_Click()
On Error GoTo Err_命令14_Click

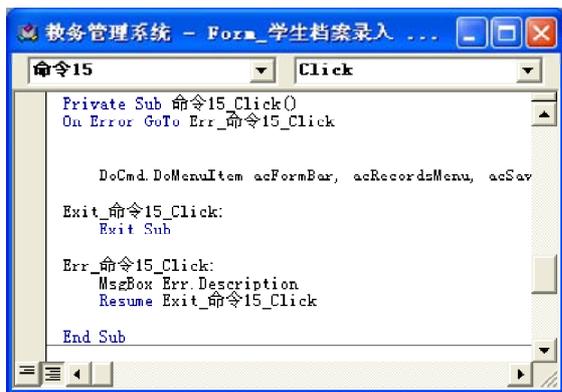
    DoCmd.GoToRecord , , acNewRec

Exit_命令14_Click:
Exit Sub

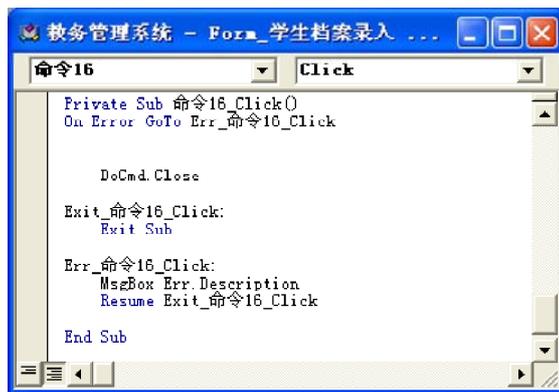
Err_命令14_Click:
MsgBox Err.Description
Resume Exit_命令14_Click

End Sub
    
```

(2) “保存记录”按钮(命令15)事件代码。“命令15”按钮的功能是,单击“保存记录”按钮时,添加到“录入学生选课成绩”窗体中的学生选课成绩信息保存到学生成绩表中。



(3) “退出”按钮(命令16)事件代码。“命令16”按钮的功能是,单击“退出”按钮时,可以退出“录入学生档案”窗体。



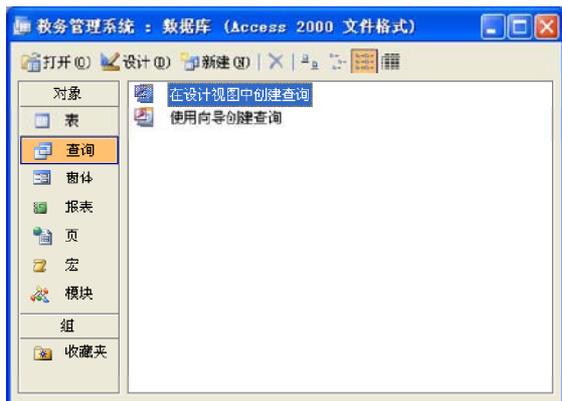
### 3. 学生信息查询

学生信息查询包括两种查询:学生基本信息查询和学生成绩查询。学生基本信息查询中包括按姓名查询、按学号查询和按班级查询;学生成绩查询包括查询不及格学生、按班查询不及格学生、按班查询90分以上学生、按学年学期查询不及格学生以及查询平均分低于班平均分学生。

#### (1) 查询设计

在设计学生信息查询窗体之间,需要首先设计上面提到的这些查询,下面首先设计“按姓名查”查询。

1 在数据库窗口中选择“查询”选项,双击“在设计视图中创建查询”选项。

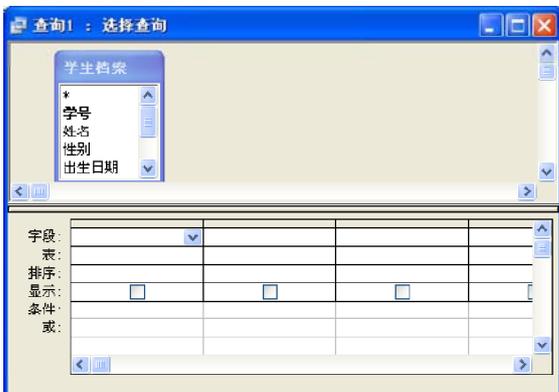


2 在弹出的“显示表”对话框中列出了可供查询设计使用的表或查询。





- 3 切换至“表”选项卡，选中“学生档案”表，单击“添加”按钮并单击“关闭”按钮。



- 4 在字段列表框中双击 \* 号，将表中所有字段添加到查询设计窗口中，结果如下图所示。



- 5 在查询设计视图中再添加“姓名”字段，但是“显示”栏中设置为不显示（即取消勾选），然后在对应“条件”栏中输入“请输入学生姓名”。



- 6 单击工具栏中的“保存”按钮，在弹出的“另存为”对话框中将该查询命名为“按姓名查”。



- 7 根据相同的方法设计“按学号查”查询。



- 8 根据相同的方法设计“按班级查”查询。



9 根据相同的方法设计“查不及格学生”查询。



10 根据相同的方法设计“按班查不及格学生”查询。



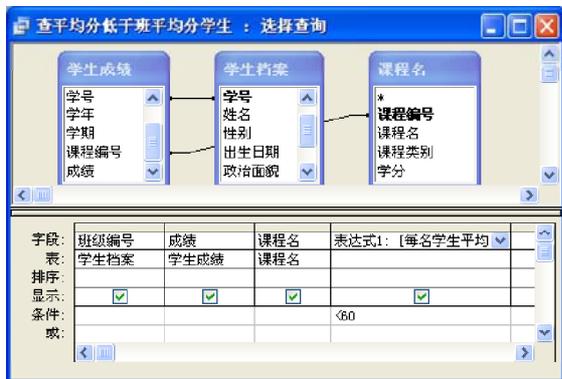
11 根据相同的方法设计“按班查90分以上学生”查询。



12 根据相同的方法设计“按学年学期查不及格学生”查询。



13 根据相同的方法设计“查平均分低于班平均分学生”查询。



(2) 窗体设计

在“学生信息查询”窗体的设计视图中包含标签、选项卡、列表框和按钮等控件，其中的主要控件属性如表19-9所示。





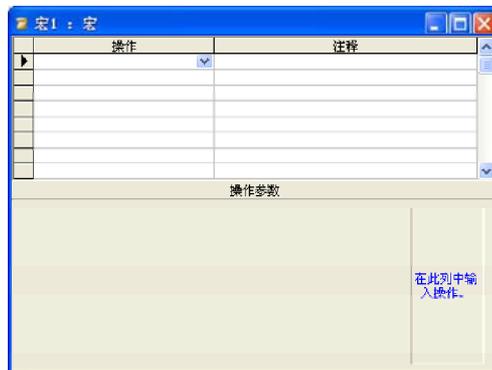
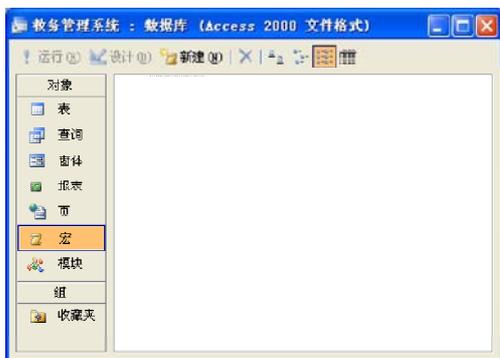
表 19-9 学生信息查询窗体属性值

对象名称	属性名称	属性值
标签4 (标签控件)	标题	学生信息查询
列表2 (列表框控件)	行来源	SELECT 学生档案表.学号, 学生档案表.姓名, 学生档案表.性别, 学生档案表.出生日期, 学生档案表.政治面貌, 学生档案表.班级编号, 学生档案表.毕业学校 FROM 学生档案表
列表8 (列表框控件)	行来源	SELECT 学生成绩表.成绩ID, 学生成绩表.学号, 学生成绩表.学年, 学生成绩表.学期, 学生成绩表.课程编号, 学生成绩表.成绩 FROM 学生成绩表
命令4 (按钮控件)	标题 单击	按姓名查 查询学生.按姓名查
命令5 (按钮控件)	标题 单击	按学号查 查询学生.按学号查
命令6 (按钮控件)	标题 单击	按班级查 查询学生.按班级查
命令10 (按钮控件)	标题 单击	查不及格学生 查询学生.不及格学生信息
命令16 (按钮控件)	标题 单击	按班查不及格学生 查询学生.按班查不及格学生
命令11 (按钮控件)	标题 单击	按班查90分以上学生 查询学生.90分以上学生信息
命令12 (按钮控件)	标题 单击	按学年学期查不及格学生 查询学生.某学期某课不及格信息
命令14 (按钮控件)	标题 单击	查平均分低于班平均分学生 查询学生.查平均分低于班平均分学生

(3) 宏设计

当单击“按姓名查”按钮时，需要给出查询的结果，这里使用宏来完成这一操作。下面是具体的设计步骤。

- 1 在打开报表数据库窗口的“对象”列表框中单击“宏”选项。单击数据库窗口中的“新建”按钮，弹出宏设计视图。
- 2 在设计视图上方表格中含有“操作”和“注释”两项，在下方可以设置操作的参数，当没设定操作时不会显示操作参数设置区域。

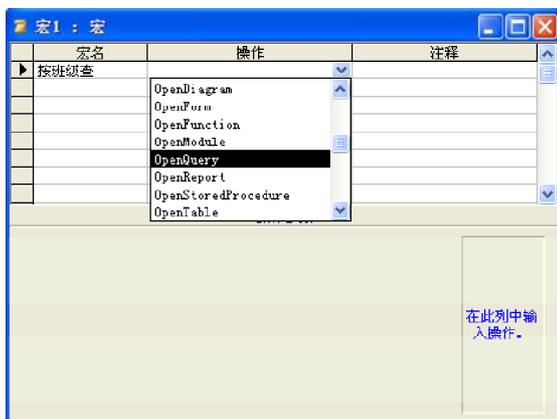




3 选择“视图>宏名”命令，结果如右图所示。这里设计的宏组包含以下8个宏：按姓名查、按学号查、按班级查、查不及格学生、按班查不及格学生、按班查90分以上学生、按学年学期查不及格学生以及查平均分低于班平均分学生。



4 设计“按班级查”宏，其余的宏可以依此方法设计。在“宏名”栏中输入“按班级查”，然后单击“操作”栏下的下拉列表框按钮，从弹出的下拉列表中选择 OpenQuery 选项。



5 按照同样的方法设计其他宏，然后单击工具栏上的“保存”按钮，在弹出的“另存为”对话框的“宏名称”文本框中将该宏组命名为“查询学生”。



#### 4. 学生信息统计

学生信息统计包括两种统计：档案信息统计和学生成绩统计。其中档案信息统计中包括每班学生人数和每班男女人数，学生成绩统计包括每班不及格人数、每班每门课平均分、每班平均成绩以及每名学生平均成绩。

##### (1) 查询设计

在设计学生信息统计窗体之前，需要首先设计上面提到的这些统计查询，设计的方法和上节中设计“按姓名查”的方法一样。下面分别给出这些查询的最终设计视图。



1 “每班学生人数”查询。“每班学生人数”查询统计每班学生的总人数，设计视图如下图所示。



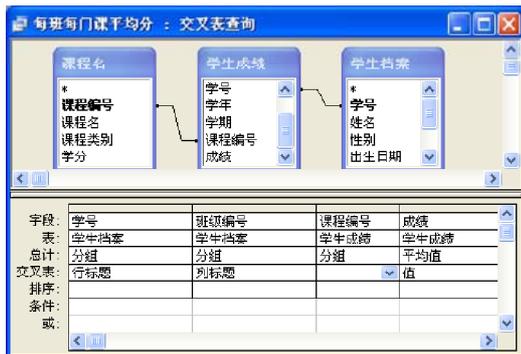
2 “每班男女人数”查询。“每班男女人数”查询统计每班学生的男女人数，设计视图如下图所示。



3 “每班不及格人次”查询。“每班不及格人次”查询统计每班不及格人次，设计视图如下图所示。



4 “每班每门课平均分”查询。“每班每门课平均分”查询统计每班每门课的平均分，设计视图如下图所示。



5 “每班平均成绩”查询。“每班平均成绩”查询计算每班的平均成绩，设计视图如下图所示。



6 “每名学生平均成绩”查询。“每名学生平均成绩”查询计算每班的平均成绩，设计视图如下图所示。



## (2) 窗体设计

在学生信息统计窗体的设计视图中包含标签、选项卡、列表框和按钮等控件，其中的主要控件属性如表 19-10 所示。



表 19-10 学生信息统计窗体属性值

对象名称	属性名称	属性值
标签4 (标签控件)	标题	学生信息统计
列表2 (列表框控件)	行来源	SELECT 学生档案表.学号, 学生档案表.姓名, 学生档案表.性别, 学生档案表.出生日期, 学生档案表.政治面貌, 学生档案表.班级编号, 学生档案表.毕业学校 FROM 学生档案表
列表8 (列表框控件)	行来源	SELECT 学生成绩表.成绩ID, 学生成绩表.学号, 学生成绩表.学年, 学生成绩表.学期, 学生成绩表.课程编号, 学生成绩表.成绩 FROM 学生成绩表
命令4 (按钮控件)	标题 单击	每班学生人数 学生统计.每班人数
命令5 (按钮控件)	标题 单击	每班男女人数 学生统计.男女生人数
命令10 (按钮控件)	标题 单击	每班不及格人次 学生统计.每班不及格人次
命令11 (按钮控件)	标题 单击	每班每门课平均分 学生统计.每班每门课平均分
命令12 (按钮控件)	标题 单击	每班平均成绩 学生统计.每班平均成绩
命令13 (按钮控件)	标题 单击	每班学生平均成绩 学生统计.每班学生平均成绩

## 5. 学生信息显示

在设计学生信息显示窗体前，需要事先设计有关学生信息的两个报表：学生档案报表和学生成绩统计报表。



(1) 学生档案报表

学生档案报表主要用于显示学生的基本信息报表，在该报表的设计视图包含标签和文本框等控件，其中的主要控件属性如表 19-11 所示。



表 19-11 学生档案报表属性值

对象名称	属性名称	属性值
标签33 (标签控件)	标题	学生档案表
文本26 (文本框控件)	控件来源	= "汇总" & "" 学生档案表_学号' = " & "" & [学生档案表_学号] & "" (" & Count(*) & "" & IIf(Count(*)=1, "明细记录", "项明细记录") & "")=HandleButtonClick(2)
文本34 (文本框控件)	控件来源	=Now()

(2) 学生成绩统计报表

学生成绩统计报表主要用于显示学生的成绩报表，在该报表的设计视图包含 16 个标签控件和 14 个文本框控件，其中的主要控件属性如表 19-12 所示。



表 19-12 学生成绩统计报表属性值

对象名称	属性名称	属性值
标签21 (标签控件)	标题	学生成绩统计表
成绩之 Sum (标签控件)	控件来源	=Sum([成绩])
成绩总计Sum (标签控件)	控件来源	=Sum([成绩])

在学生信息显示窗体的设计视图包含标签、文本框、子窗体 / 子报表和按钮等控件，其中的主要控件属性如表 19-13 所示。



表 19-13 学生信息显示窗体属性值

对象名称	属性名称	属性值
标签9 (标签控件)	标题	学生信息显示
学生成绩表 子窗体1 (子窗体/子报表控件)	源对象	学生成绩表 子窗体1
命令10 (按钮控件)	标题 单击	显示学生基本信息报表 [事件过程]
命令11 (按钮控件)	标题 单击	显示学生成绩统计报表 [事件过程]
命令12 (按钮控件)	标题 单击	退出 [事件过程]

下面来定义窗体中各个功能按钮的事件过程。

(1) “显示学生基本信息报表”按钮(命令10)事件代码。“命令10”按钮的功能是,单击“显示学生基本信息报表”按钮时,打开学生基本信息报表。



```

Private Sub 命令10_Click()
On Error GoTo Err_命令10_Click
Dim stDocName As String
stDocName = "学生基本信息表"
DoCmd.OpenReport stDocName, acPreview '打开报表
Exit_命令10_Click:
Exit Sub
Err_命令10_Click:
MsgBox Err.Description
Resume Exit_命令10_Click
End Sub

```

(2) “显示学生成绩统计报表”按钮(命令11)事件代码。“命令11”按钮的功能是,单击“显示学生成绩统计报表”按钮时,打开学生成绩统计报表。



```

End Sub
Private Sub 命令11_Click()
On Error GoTo Err_命令11_Click
Dim stDocName As String
stDocName = "学生成绩统计报表"
DoCmd.OpenReport stDocName, acPreview
Exit_命令11_Click:
Exit Sub
Err_命令11_Click:
MsgBox Err.Description
Resume Exit_命令11_Click
End Sub

```

(3) “退出”按钮(命令16)事件代码。“命令16”按钮的功能是,单击“退出”按钮时,可以退出学生信息显示窗体。



```

Private Sub 命令16_Click()
On Error GoTo Err_命令16_Click

DoCmd.Close

Exit_命令16_Click:
Exit Sub

Err_命令16_Click:
MsgBox Err.Description
Resume Exit_命令16_Click

End Sub

```

## 教师信息管理模块设计

根据系统信息流程图,教师信息管理模块包含教师档案录入、教师授课信息录入、教师相关信息查询、教师授课信息统计以及教师授课信息浏览。下面分别介绍这些子功能窗体的设计过程。



### 1. 登录教师档案信息

教师档案录入窗体主要用于添加新的教师信息，在该窗体的设计视图中包含标签、文本框、矩形和按钮等控件，其中的主要控件属性如表19-14所示。



表 19-14 “录入教师档案”窗体属性值

对象名称	属性名称	属性值
标签1 (标签控件)	标题	录入教师档案信息
命令56 (按钮控件)	标题	上一记录
	单击	[事件过程]
命令55 (按钮控件)	标题	下一记录
	单击	[事件过程]
命令54 (按钮控件)	标题	添加记录
	单击	[事件过程]
命令25 (按钮控件)	标题	保存记录
	单击	[事件过程]
命令24 (按钮控件)	标题	退出
	单击	[事件过程]

下面来定义窗体中各个功能按钮的事件过程。

(1) “上一记录”按钮（命令56）事件代码。“命令56”按钮的功能是，单击“上一记录”按钮时，录入教师档案窗体中显示上一条教师记录。

(2) “下一记录”按钮（命令55）事件代码。“命令55”按钮的功能是，单击“下一记录”按钮时，录入教师档案窗体中显示下一条教师记录。



(3) “添加记录”按钮(命令54)事件代码。“命令54”按钮的功能是,单击“添加记录”按钮时,录入教师档案窗体中各组合框和文本框显示为空,这时可以输入新的教师信息。

```

Private Sub 命令54_Click()
    On Error GoTo Err_命令54_Click

    DoCmd.GoToRecord , , acNewRec

    Exit_命令54_Click:
    Exit Sub

    Err_命令54_Click:
    MsgBox Err.Description
    Resume Exit_命令54_Click

End Sub
    
```

(4) “保存记录”按钮(命令25)事件代码。“命令25”按钮的功能是,单击“保存记录”按钮时,将添加到录入教师档案窗体中的教师信息保存到教师档案表中。

```

Private Sub 命令25_Click()
    On Error GoTo Err_命令25_Click

    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSave

    Exit_命令25_Click:
    Exit Sub

    Err_命令25_Click:
    MsgBox Err.Description
    Resume Exit_命令25_Click

End Sub
    
```

(5) “退出”按钮(命令24)事件代码。“命令24”按钮的功能是,单击“退出”按钮时,退出录入教师档案窗体。

```

Private Sub 命令24_Click()
    On Error GoTo Err_命令24_Click

    DoCmd.Close

    Exit_命令24_Click:
    Exit Sub

    Err_命令24_Click:
    MsgBox Err.Description
    Resume Exit_命令24_Click

End Sub
    
```

## 2. 录入教师授课信息

录入教师授课信息窗体主要用于添加新的教师授课信息,在该窗体的设计视图中包含标签、文本框、矩形、组合框和按钮等控件,其中的主要控件属性如表 19-15 所示。

表 19-15 录入教师档案窗体属性值

对象名称	属性名称	属性值
标签1 (标签控件)	标题	录入教师授课信息
命令9 (按钮控件)	标题	上一记录
	单击	[事件过程]
命令8 (按钮控件)	标题	下一记录
	单击	[事件过程]
命令10 (按钮控件)	标题	添加记录
	单击	[事件过程]
命令11 (按钮控件)	标题	保存记录
	单击	[事件过程]
命令12 (按钮控件)	标题	退出
	单击	[事件过程]



下面来定义窗体中各个功能按钮的事件过程。

(1) “上一记录”按钮(命令56)事件代码。“命令56”按钮的功能是,单击“上一记录”按钮时,“录入教师档案”窗体中显示上一条教师记录。

```

命令56 Click
End Sub
Private Sub 命令56_Click()
On Error GoTo Err_命令56_Click

DoCmd.GoToRecord , , acPrevious

Exit_命令56_Click:
Exit Sub

Err_命令56_Click:
MsgBox Err.Description
Resume Exit_命令56_Click

End Sub

```

(2) “下一记录”按钮(命令55)按钮事件代码。“命令55”按钮的功能是,单击“下一记录”按钮时,“录入教师档案”窗体中显示下一条教师记录。

```

命令55 Click
Private Sub 命令55_Click()
On Error GoTo Err_命令55_Click

DoCmd.GoToRecord , , acNext

Exit_命令55_Click:
Exit Sub

Err_命令55_Click:
MsgBox Err.Description
Resume Exit_命令55_Click

End Sub

```

(3) “添加记录”按钮(命令54)事件代码。“命令54”按钮的功能是,单击“添加记录”按钮时,“录入教师档案”窗体中各组合框和文本框显示为空,这时可以输入新的教师信息。

```

命令54 Click
Private Sub 命令54_Click()
On Error GoTo Err_命令54_Click

DoCmd.GoToRecord , , acNewRec

Exit_命令54_Click:
Exit Sub

Err_命令54_Click:
MsgBox Err.Description
Resume Exit_命令54_Click

End Sub

```

(4) “保存记录”按钮(命令25)事件代码。“命令25”按钮的功能是,单击“保存记录”按钮时,将添加到“录入教师档案”窗体中的教师信息保存到教师档案表中。

```

命令25 Click
Private Sub 命令25_Click()
On Error GoTo Err_命令25_Click

DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSave

Exit_命令25_Click:
Exit Sub

Err_命令25_Click:
MsgBox Err.Description
Resume Exit_命令25_Click

End Sub

```

(5) “退出”按钮(命令24)事件代码。“命令24”按钮的功能是,单击“退出”按钮时,退出“录入教师档案”窗体。

```

命令24 Click
Private Sub 命令24_Click()
On Error GoTo Err_命令24_Click

DoCmd.Close

Exit_命令24_Click:
Exit Sub

Err_命令24_Click:
MsgBox Err.Description
Resume Exit_命令24_Click

End Sub

```



### 3. 教师相关信息查询

教师相关信息查询包括：按姓名查、按编号查、按系名查、按职称查、按班级查和按地点查。在设计“教师相关信息查询”窗体之间，需要首先设计上面提到的这些查询，下面分别给出这些查询的最终设计视图。

#### (1) “按姓名查”查询

“按姓名查”查询根据教师的姓名来进行查询，设计视图如下图所示。



#### (2) “按编号查”查询

“按编号查”查询根据教师的编号来进行查询，设计视图如下图所示。



#### (3) “按系名查”查询

“按系名查”查询根据教师所在的系名来进行查询，设计视图如下图所示。



#### (4) “按职称查”查询

“按职称查”查询根据教师的职称来进行查询，设计视图如下图所示。



#### (5) “按班级查”查询

“按班级查”查询根据教师所在的班级来进行查询，设计视图如右图所示。



## (6) “按地点查” 查询

“按地点查” 查询根据教师的授课地点进行查询，设计视图如下图所示。

在课程及选课信息查询窗体的设计视图中包含标签、文本框、子窗体 / 子报表和按钮等控件，其中的主要控件属性如表 19-16 所示。



表 19-16 课程及选课信息查询窗体属性值

对象名称	属性名称	属性值
标签16 (标签控件)	标题	课程及选课信息查询
命令21 (按钮控件)	标题	按姓名查
命令21 (按钮控件)	单击	查询教师. 按姓名查
命令22 (按钮控件)	标题	按编号查
命令22 (按钮控件)	单击	查询教师. 按编号查
命令24 (按钮控件)	标题	按系名查
命令24 (按钮控件)	单击	查询教师. 按系名查
命令25 (按钮控件)	标题	按职称查
命令25 (按钮控件)	单击	查询教师. 按职称查
命令26 (按钮控件)	标题	按班级查
命令26 (按钮控件)	单击	查询教师. 按班级查
命令27 (按钮控件)	标题	按地点查
命令27 (按钮控件)	单击	查询教师. 按地点查
命令20 (按钮控件)	标题	退出
命令20 (按钮控件)	单击	[事件过程]

## 4. 教师基本信息及授课信息统计

教师基本信息及授课信息统计包括：统计有课教师人次、统计各系教师人数、统计各职称教师人数和教师每周学时数。

在设计“教师基本信息及授课信息统计”窗体之间，需要首先设计上面提到的这些统计查询，下面分别给出这些查询的最终设计视图。



(1) “统计有课教师人次” 查询  
“统计有课教师人次” 查询统计有课教师的总人数，设计视图如下图所示。



(2) “统计各系教师人数” 查询  
“统计各系教师人数” 查询统计各系教师的总人数，设计视图如下图所示。



(3) “统计各职称教师人数” 查询  
“统计各职称教师人数” 查询统计各职称教师人数，设计视图如下图所示。



(4) “教师每周学时数” 查询  
“教师每周学时数” 查询统计每个教师每周的学时数，设计视图如下图所示。



在教师基本信息及授课信息统计窗体的设计视图中包含标签、文本框、子窗体/子报表和按钮等控件，其中的主要控件属性如表 19-17 所示。

表 19-17 教师基本信息及授课信息统计窗体属性值

对象名称	属性名称	属性值
标签16 (标签控件)	标题	教师基本信息及授课信息统计
命令21 (按钮控件)	标题	统计有课教师人次
	单击	教师统计. 统计有课教师人次
命令25 (按钮控件)	标题	统计各系教师人数
	单击	教师统计. 统计各系教师人数
命令26 (按钮控件)	标题	统计各职称教师人数
	单击	教师统计. 统计各职称教师人数
命令28 (按钮控件)	标题	教师每周学时数
	单击	教师统计. 教师每周学时数

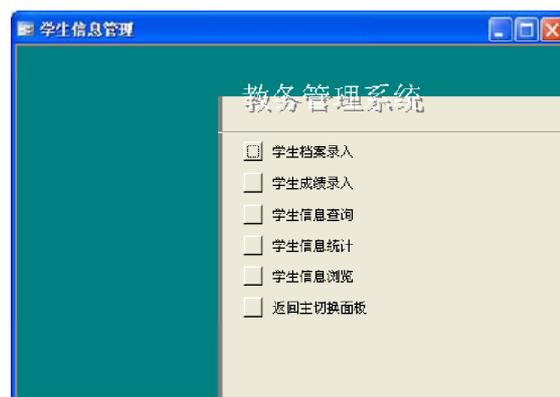
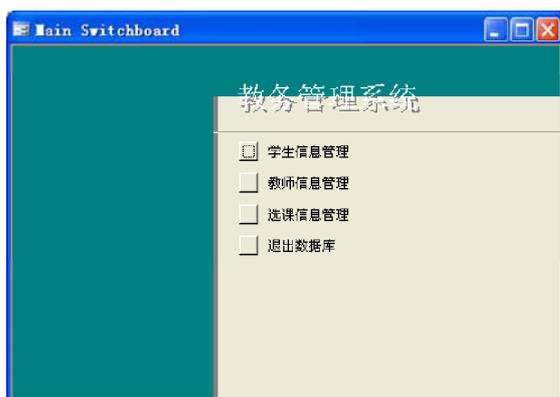


## 主切换面板窗体模块的设计

在设计完所有的功能窗体后，最后设计系统的切换面板，具体的设计过程参见第 18 章的最后一节。

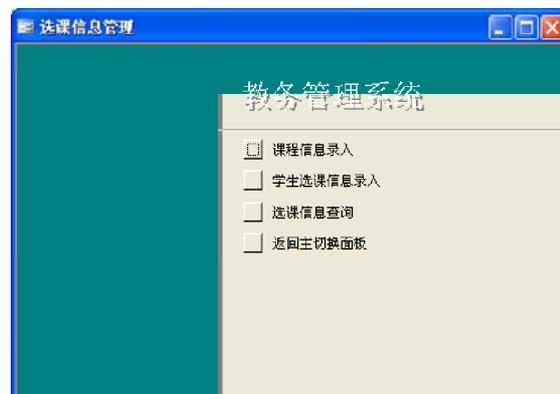
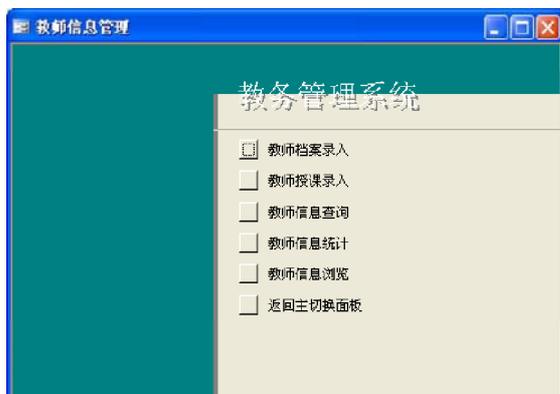
下图所示为系统的主切换面板。

单击“学生信息管理”前的按钮可以打开“学生信息管理”子切换面板。



单击“教师信息管理”前的按钮可以打开“教师信息管理”子切换面板。

单击“选课信息管理”前的按钮可以打开“选课信息管理”子切换面板。





Access 2003



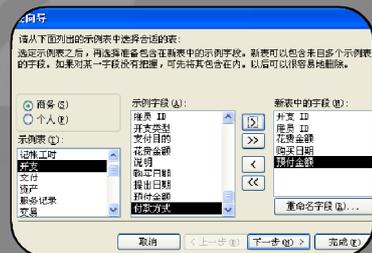
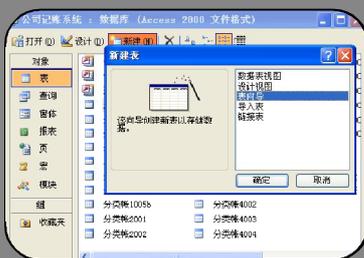
# part

# 20

## 财务管理系统

- ◆ 系统总体设计
- ◆ 数据库设计
- ◆ 系统模块设计

财务管理是公司日常经营中一项很重要的业务，公司需要利用财务系统来管理各种财务信息。本章展示了一个具备财务管理常用功能系统的开发过程。



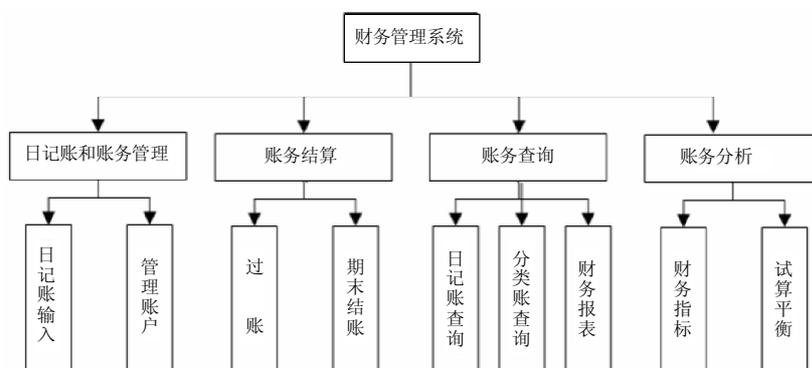
在本章中没有介绍数据库的具体设计过程，读者可根据前面介绍的方法进行设计。

## 1 系统总体设计

财务管理系统主要完成以下功能。

- 财务信息输入，管理公司的日记账和分类账，可以浏览、增加、修改和删除日记账和分类账。
- 财务信息查询，用于对公司的日记账和分类账进行查询。
- 财务信息分析，用于分析公司的财务状况。

根据需求分析，建立财务管理系统的逻辑模型，如下图所示。



## 2 数据库设计

### 数据表逻辑结构设计

#### 1. 分类账表的设计

分类账表用于记录一段时间的分类账信息，其逻辑结构如表20-1所示。

表 20-1 分类账表

字段名	数据类型	字段宽度	格式	索引	说明
账户编号	文本	50	标准	无	账号编号
日期	日期/时间		标准	无	开户日期
日记账编号	文本	50	标准	无	对应的日记账编号
借	货币		标准	无	借贷金额
贷	货币		标准	无	贷款金额
余额	货币		标准	无	分类账余额
摘要	备注		标准	无	摘要信息

#### 2. 日记账表的设计

日记账表用于记录每天的借贷信息，其逻辑结构如表20-2所示。



表 20-2 记账表

字段名	数据类型	字段宽度	格式	索引	说明
序号	自动编号	长整型	标准	有(无重复)	该条目的记账编号
记账编号	文本	50	标准	无	企业自定义格式的记账编号
账户编号	文本	50	标准	无	企业自定义的分类账编号
日期	日期/时间		短日期	无	业务发生日期
借	货币		货币	无	借方
贷	货币		货币	无	贷方
摘要	备注		标准	无	业务备注
已过账	是/否		标准	无	是否已经转录至分类账

### 3. 系统常量表的设计

系统常量表用于记录系统的常量参数信息，其逻辑结构如表20-3所示。

表 20-3 系统常量表

字段名	数据类型	字段宽度	格式	索引	说明
名称	文本	50	标准	无	系统关键字
数值	数字	长整型	标准	无	常量数值
备注	备注		标准	无	对该常量的说明

### 4. 账户列表的设计

账户列表记录公司中每个账户的基本信息，其逻辑结构如表20-4所示。

表 20-4 账户列表

字段名	数据类型	字段宽度	格式	索引	说明
序号	自动编号	长整型	标准	无	账户自动编号
账户编号	文本	50	标准	有(无重复)	企业自定义格式的账户编号
会计科目	文本	50	标准	无	账户名称
明细账	文本	50	标准	无	明细账目名称
账户一级类型	文本	50	标准	无	账户粗略分类,如“资产”和“负债”
账户二级类型	文本	50	标准	无	账户详细分类,如“流动资产”和“长期负债”
抵减账户	是/否		是/否	无	是否属于抵减账户
借方余额	货币		标准	无	账户建立时借方余额
贷方余额	货币		标准	无	账户建立时贷方余额
建立日期	日期/时间		标准	无	账户建立时间

## 数据表设计

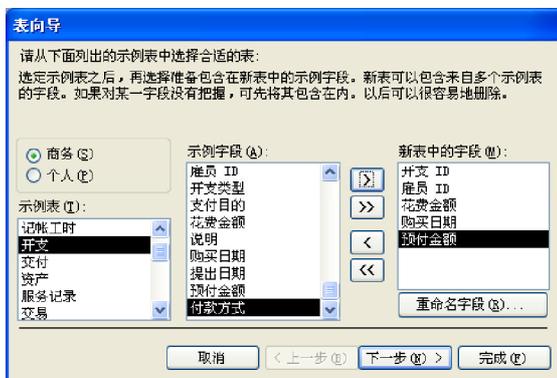
创建或打开“系统”数据库以后，就可以创建和设计相应的数据表。

1. 使用向导创建分类账表

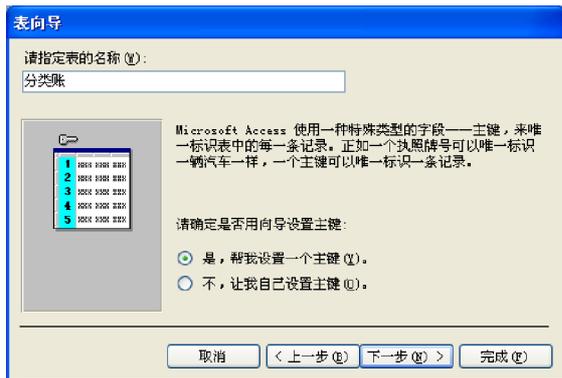
1 打开数据库窗口。在“对象”列表框中选择“表”选项，弹出创建表选择界面。单击“新建”按钮，将会弹出“新建表”对话框。在其中双击“表向导”选项，将会弹出“表向导”对话框。在其中选中“商务”单选按钮，在“示例表”列表框中选择“开支”选项。



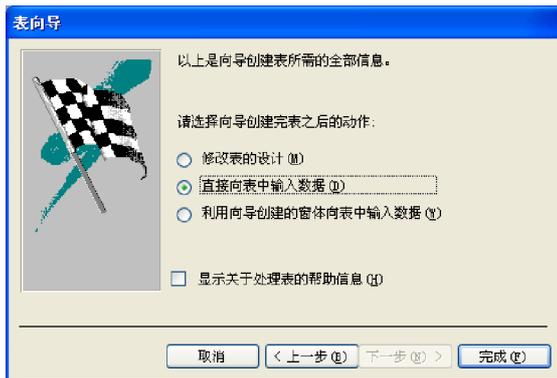
2 在“示例字段”列表框中选择所需的字段名，添加以下字段：“开支ID”、“雇员ID”、“说明”、“预付金额”、“预付金额”、“购买日期”、“花费金额”。单击“重命名字段”按钮，把“开支ID”改为“账户编号”，把“花费金额”改为“余额”，把“说明”改为“摘要”，把“预付金额”改为“贷”，把“购买日期”改为“日期”。



3 单击“下一步”按钮，在弹出的界面中将表命名为“分类账”，并选中“是，帮我设置一个主键”单选按钮。



4 单击“下一步”按钮，在下一界面中设置与其代表的关系。之后，单击“下一步”按钮，弹出如下图所示的界面。



5 选中“直接向表中输入数据”单选按钮，单击“完成”按钮，结果如右图所示。

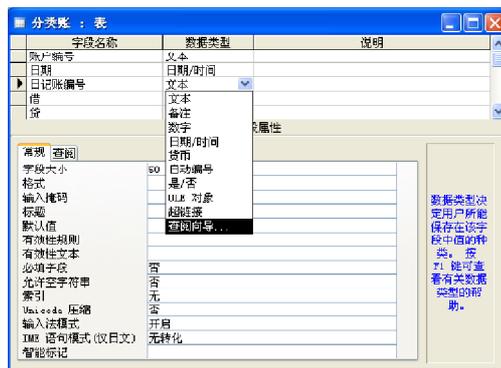
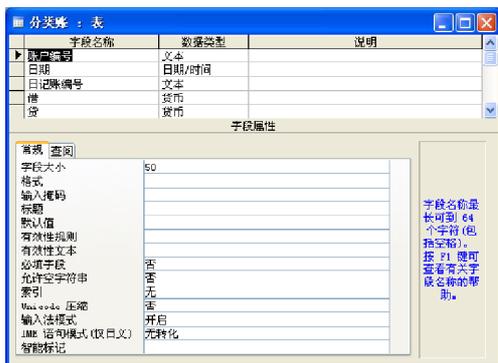




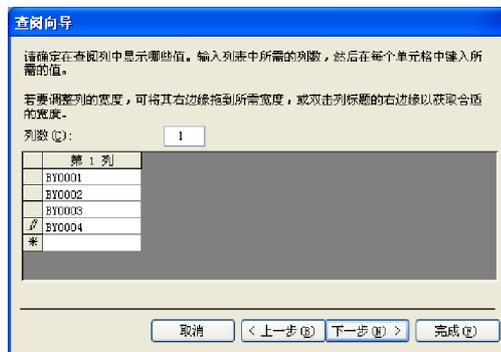
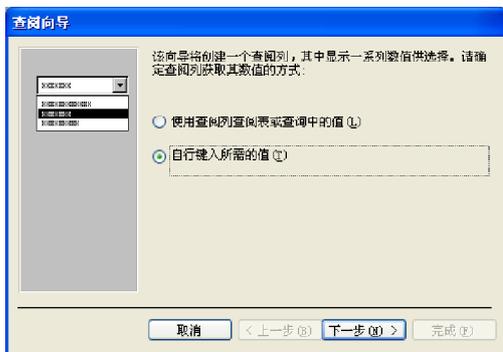
## 2. 使用设计视图创建分类账表

使用向导设计的分类账表中有些字段的数据类型并不符合要求，例如日记账编号在实际应用中可能已经指定了一些固定的编号，使用时只需要从下拉列表框中进行选择，而不需要输入。下面在设计视图中利用查阅向导来进一步设计。

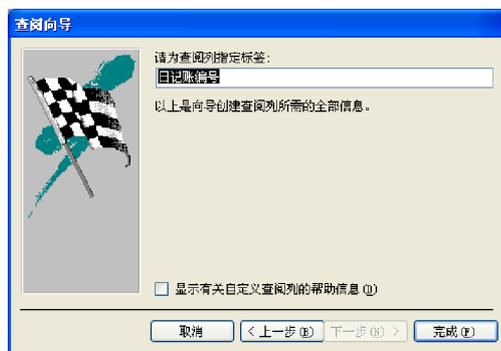
- 1 打开数据库窗口，并在“对象”列表框中选择“表”选项。选中“分类账”表，然后单击“设计”按钮，进入“设计”视图。
- 2 在弹出的窗口的“日记账编号”的“数据类型”下拉列表中选择“查阅向导”选项。



- 3 在弹出的“查阅向导”窗口中选中“自行键入所需的值”单选按钮，之后单击“下一步”按钮。
- 4 在当前界面中将列数设定为1，在文本框中为“日记账编号”输入值，如 BY0001, BY0002, BY0003, BY0004。

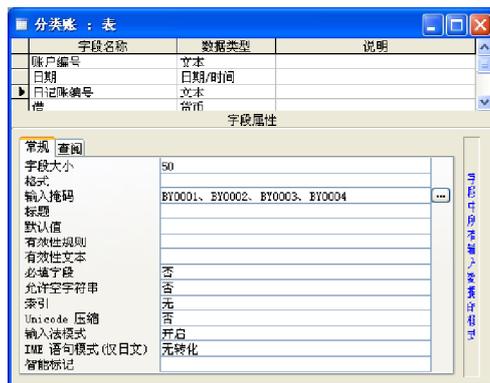


- 5 单击“下一步”按钮，在当前界面中查阅列指定标签，这里保持系统的默认值“日记账编号”。





- 6 单击“完成”按钮，最终效果如右图所示。



## 3

## 系统模块设计

Access

财务管理系统的功能窗体包括：主界面、日记账和账户管理、账务查询、账务结算以及账务分析。下面介绍这些窗体的设计过程。

### 主界面窗体模块的设计

和前两章的切换面板不同，本章建立一个窗体来管理系统的功能窗体，其功能和切换面板的功能相似。

主窗体中有标签、框架和按钮等对象，表 20-5 所示为主窗体的控件属性设置。这里只是列出了一些重要的属性设置，没有列出的都采用默认设置。



表 20-5 主界面窗体属性值

对象名称	属性名称	属性值
标签0 (标签控件)	标题	系统
cmdInput (按钮控件)	标题 单击	日记账和账户管理 [事件过程]
cmdQuery (按钮控件)	标题 单击	账务查询 [事件过程]
cmdOperate (按钮控件)	标题 单击	账务结算 [事件过程]
cmdAnalysis (按钮控件)	标题 单击	账务分析 [事件过程]



(续表)

对象名称	属性名称	属性值
cmdExit (按钮控件)	标题	退出系统
	单击	[事件过程]
cmdHide (按钮控件)	标题	关闭窗口
	单击	[事件过程]

要实现窗体中按钮的各项功能，需要对按钮添加事件过程，这些按钮都是用来打开相应的窗体的。下面为每个按钮编写代码。

(1) “日记账和账户管理”按钮 (cmdInput) 代码设计。cmdInput 按钮的功能是，单击“日记账和账户管理”按钮时，打开日记账和账户管理窗体。

```

Private Sub cmdInput_Click()
    DoCmd.OpenForm "日记账和账户管理"
End Sub
    
```

(2) “账务查询”按钮 (cmdQuery) 代码设计。cmdQuery 按钮的功能是，单击“账务查询”按钮时，打开账务查询窗体。

```

Private Sub cmdQuery_Click()
    DoCmd.OpenForm "账务查询"
End Sub
    
```

(3) “账务结算”按钮 (cmdOperate) 按钮代码设计。cmdOperate 按钮的功能是，单击“账务结算”按钮，打开账务结算窗体。

```

Private Sub cmdOperate_Click()
    DoCmd.OpenForm "账务结算"
End Sub
    
```

(4) “账务分析”按钮 (cmdAnalysis) 代码设计。cmdAnalysis 按钮的功能是，单击“账务分析”按钮，打开账务分析窗体。

```

Private Sub cmdAnalysis_Click()
    DoCmd.OpenForm "账务分析"
End Sub
    
```

(5) “关闭窗口”按钮 (cmdHide) 代码设计。cmdHide 按钮的功能是，单击“关闭窗口”按钮，关闭该主界面窗体。

```

Private Sub cmdHide_Click()
    On Error GoTo Err_cmdHide_Click
    DoCmd.Close
Exit_cmdHide_Click:
    Exit Sub
Err_cmdHide_Click:
    MsgBox Err.Description
    Resume Exit_cmdHide_Click
End Sub
    
```

(6) “退出系统” (cmdExit) 代码设计。cmdExit 按钮的功能是，单击“退出系统”按钮时，关闭该主界面窗体，同时退出 Access。

```

Private Sub cmdExit_Click()
    On Error GoTo Err_cmdExit_Click
    DoCmd.Quit
Exit_cmdExit_Click:
    Exit Sub
Err_cmdExit_Click:
    MsgBox Err.Description
    Resume Exit_cmdExit_Click
End Sub
    
```

## 日记账和账户管理窗体模块的设计

在日记账和账户管理窗体中有文本框、框架和按钮等对象，表20-6所示为系统的日记账和账户管理窗体的控件属性设置。

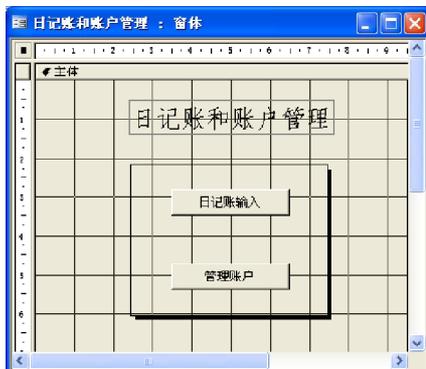


表 20-6 日记账和账务管理窗体属性值

对象名称	属性名称	属性值
cmdDaybookInput (按钮控件)	标题	日记账输入
	单击	[事件过程]
cmdEditAccounts (按钮控件)	标题	管理账户
	单击	[事件过程]

要实现窗体中按钮的各项功能，需要对按钮添加事件过程，这些按钮分别用来打开相应的窗体，下面为每个按钮编写代码。

(1) “日记账输入”按钮 (cmdDaybookInput) 代码设计。cmdDaybookInput 按钮的功能是，单击“日记账输入”按钮，打开输入日记账窗体。



(2) “管理账户”按钮 (cmdEditAccounts) 代码设计。cmdEditAccounts 按钮的功能是，单击“管理账户”按钮，打开管理账户窗体。



从上可知系统的输入任务包括：日记账输入和管理账户，下面介绍这两个窗体的设计过程。

### 1. 日记账输入窗体的设计

在输入日记账窗体中有文本框、标签、组合框和按钮等对象，表20-7所示为系统的“日记账输入”窗体的控件属性设置。

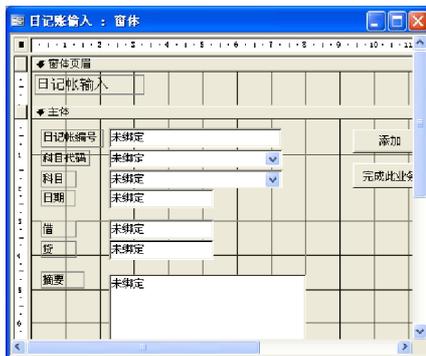




表 20-7 “输入日记账”窗体属性值

对象名称	属性名称	属性值
Label37 (标签控件)	标题	日记账输入
cboAccount (组合框控件)	行来源类型	值列表
	行来源	1001; 1002; 1003001
	更新后	[事件过程]
cmdAdd (按钮控件)	标题	添加
	单击	[事件过程]
cmdClose (按钮控件)	标题	关闭
	单击	[事件过程]
cmdNext (按钮控件)	标题	完成此业务
	单击	[事件过程]

要实现窗体中按钮的各项功能，需要对按钮添加事件过程，下面为每个按钮编写代码。

(1) 窗体初始化代码设计。窗体初始化用来为窗体中科目代码和科目后的列表框进行初始化，初始化的值分别来自账户列表的“账户编号”和“会计科目”。

```

Private Sub Form_Load()
    '窗体加载时的预处理
    为cboAccount和cboAccountNo添加内容
    OpenRS "SELECT * FROM 帐户列表", rs
    Do Until rs.EOF
        cboAccount.AddItem rs("帐户编号")
        cboAccount.AddItem rs("会计科目") & " " & rs("明细帐")
        rs.MoveNext
    Loop
    设置txtDate的内容为当前日期
    txtDate = Date
End Sub
    
```

(2) SaveJournal 过程代码设计。SaveJournal 过程用于保存窗体内的日记账数据，在窗体的“添加”和“完成此业务”按钮事件代码中都调用了此过程。

```

Private Sub SaveJournal()
    '使用窗体内数据保存日记账
    OpenRS "SELECT * FROM 日记账", rs
    rs.AddNew
    rs("日记账编号") = txtDaybookNo
    rs("帐户编号") = cboAccountNo
    rs("日期") = txtDate
    If IsNumeric(txtCredit) Then rs("借") = CCur(txtCredit)
    If IsNumeric(txtDebit) Then rs("贷") = CCur(txtDebit)
    'CCur:将实数转换为Currency类型
    rs("摘要") = txtMemo
    rs.Update
End Sub
    
```

(3) “科目”组合框 (cboAccount) 的AfterUpdate 过程代码设计。cboAccount组合框的AfterUpdate 过程用于对选择的科目进行自动更新。

```

Private Sub cboAccount_AfterUpdate()
    '使cboAccount变动时cboAccountNo也同时变动，调用ListIndex属性值
    cboAccountNo = cboAccountNo.ItemData(cboAccount.ListIndex)
    txtDaybookNo = cboAccountNo + "-" + GetNextNo(cboAccountNo)
    '将光标移动到txtCredit控件上方便输入。
    txtCredit.SetFocus
End Sub
    
```

(4) “科目代码”组合框 (cboAccountNo) 的AfterUpdate 过程代码设计。cboAccountNo 组合框的AfterUpdate过程用于对选择的科目代码进行自动更新。

```

Private Sub cboAccountNo_AfterUpdate()
    '使cboAccountNo变动时cboAccount也同时变动
    cboAccount = cboAccount.ItemData(cboAccountNo.ListIndex)
    txtDaybookNo = cboAccountNo + "-" + GetNextNo(cboAccountNo)
    txtCredit.SetFocus
End Sub
    
```

(5) “添加”按钮 (cmdAdd) 代码设计。“cmdAdd”按钮的功能是，单击“添加”按钮，将输入的新日记账记录保存在日记账表中。

```

Private Sub cmdAdd_Click()
    '一项分录已经输入完成，而该业务的还需要做其他分录
    '将当前窗体的内容保存到数据库
    SaveJournal
    同一业务的txtDate、txtMemo应相同，设置成不能更改
    txtDate.Enabled = False
    txtMemo.Enabled = False
    '借贷相抵，在很多情况下可以节省输入劳动。
    iTemp = txtCredit
    txtCredit = txtDebit
    txtDebit = iTemp
End Sub
    
```

(6) “完成此业务”按钮 (cmdNext) 代码设计。cmdNext 按钮的功能是，单击“完成此业务”按钮，清除窗体中的记录信息，以便进行下一条记录的添加操作。

```

窗体管理系统 - Form_日记帐输入 (代码)
cmdNext Click
Private Sub cmdNext_Click()
    '---将业务的所有分录式已完成
    '---将当前窗体的内容保存至数据库
    SaveJournal
    '清除窗体内容
    txtDate.Enabled = True
    txtMemo.Enabled = True
    txtDaybookNo = ""
    txtDate = Date
    txtMemo = ""
    txtCredit = ""
    txtDebit = ""
End Sub
    
```

## 2. 管理账户窗体的设计

在“管理账户”窗体中有文本框、标签、组合框、复选框和按钮等对象，表 20-8 所示为系统的“管理账户”窗体的控件属性设置。



表 20-8 管理账户窗体属性值

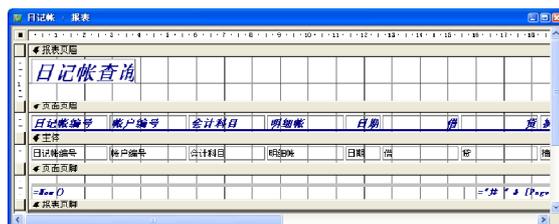
对象名称	属性名称	属性值
Label 21 (标签控件)	标题	管理账户
序号_Label (标签控件)	标题	序号
帐户编号_Label (标签控件)	标题	帐户编号
借方余额_Label (标签控件)	标题	借方余额
建立日期_Label (标签控件)	标题	建立日期
会计科目_Label (标签控件)	标题	会计科目
序号_Label (标签控件)	标题	Present Time:
Text2 (文本框控件)	控件来源	=Format(Date(),"长日期") & " " & Format(Time(),"长时间")
CmdExit (按钮控件)	标题	关闭
	单击	[事件过程]

## 🔑 账务查询窗体模块的设计

下面先设计该模块中的查询和财务报表。

### 1. 日记帐查询报表的设计

日记帐查询用于生成给定时间的日记帐信息，在窗口中输入要查询的日期，单击“确定”按钮即可生成该日期的日记帐信息。





## 2. 财务指标报表的设计

本节创建公司财务指标报表，公司的财务指标包括有流动比率、酸性测试比率、年存货周转率以及资产负债率。通过在报表中设计 4 个文本框来完成这四个财务指标的计算，并将结果显示在报表中。

在报表中有标签和按钮等对象，表20-9所示是公司财务指标报表的控件属性设置。



表 20-9 财务指标报表属性值

对象名称	属性名称	属性值	含义
Label0 (标签控件)	标题	公司财务指标	
Text3 (文本框控件)	控件来源	=GetSum("","","流动资产")/GetSum("","","流动负债")	流动比率
Text5 (文本框控件)	控件来源	=(GetSum("现金")+GetSum("银行存款")+GetSum("短期投资")+GetSum("应收账款"))/GetSum("","","流动负债")	酸性测试比率
Text7 (文本框控件)	控件来源	=2*GetSum("销售成本", "", "", True)/(GetSum("应收账款", "", "", True, Date()+GetSum("应收账款", "", "", True, Date()-365))	年存货周转率
Text9 (文本框控件)	控件来源	=GetSum("","负债")/GetSum("","资产")	资产负债率

## 3. 记账报表的设计

本节创建记账报表，用来显示记账信息，具体操作步骤如下。

- 1 进入数据库窗口中的“报表”界面，单击“新建”按钮，弹出“新建报表”对话框。
- 2 在其中选择“报表向导”选项，单击“确定”按钮，弹出如下图所示的“报表向导”对话框。





3 在其中选择“日记账”为源表，并选择其中的所有字段，单击“下一步”按钮，弹出如下图所示的界面。



4 在其中选择“序号”为分组级别，单击“下一步”按钮，弹出如下图所示的界面。



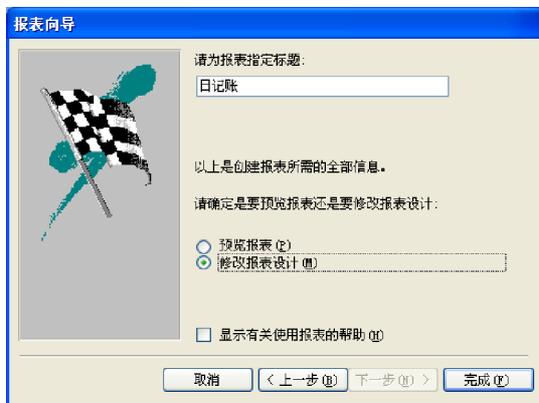
5 在其中设置按“日记账编号”升序排序，单击“下一步”按钮，弹出如下图所示的界面。



6 在其中选择“递阶”布局和“纵向”选项，单击“下一步”按钮，弹出如下图所示的界面。



7 在其中选择“组织”样式，单击“下一步”按钮，弹出如下图所示的界面。



8 在其中为报表命名为“日记账”，选择“修改报表设计”选项，最后单击“确定”按钮，最终效果如下图所示。





#### 4. 损益表报表的设计

本节创建的损益表报表用于显示和打印相关的损益信息，报表的控件设计很简单，主要通过报表类模块的设计完成这些功能。设置完的损益表报表如右图所示。



在代码编辑窗口，需要输入如下代码。

```
Option Compare Database

Private Sub Report_Activate()
' 显示报表的日期
    lblDate.Caption = dDateStart & "-" & dDateEnd
End Sub

Private Sub 主体_Print(Cancel As Integer, PrintCount As Integer)
' 输出报表内容
    Dim crcProfit As Currency
    'crcProfit 存放利润值
    crcProfit = 0
    Me.Print " 营业收入 "; Tab(60); GetIncome(" 营业收入 ", crcProfit) & vbCrLf;
    Tab(10)          ' Tab()用于设定在哪一列打印。
    Me.Print " 减: 产品销售成本 "; Tab(60); GetIncome(" 销售成本 ", crcProfit)
    & vbCrLf; Tab(1)
    Me.Print " 产品销售费用 "; Tab(60); GetIncome(" 销售费用 ", crcProfit) &
    vbCrLf; Tab(1)
    Me.Print " 产品销售税金及附加 "; Tab(60); GetIncome(" 产品销售税金及附加 ",
    crcProfit) & vbCrLf; Tab(1)
    Me.Print " 产品销售利润 "; Tab(60); crcProfit & vbCrLf; Tab(10)
    Me.Print " 加: 其他业务利润 "; Tab(60); GetIncome(" 其他业务利润 ", crcProfit)
    & vbCrLf; Tab(10)
    Me.Print " 减: 管理费用 "; Tab(60); GetIncome(" 管理费用 ", crcProfit) &
    vbCrLf; Tab(1)
    Me.Print " 财务费用 "; Tab(60); GetIncome(" 财务费用 ", crcProfit) & vbCrLf;
    Tab(1)
    Me.Print " 营业利润 "; Tab(60); crcProfit & vbCrLf; Tab(1)
    Me.Print " 投资收益 "; Tab(60); GetIncome(" 投资收益 ", crcProfit) & vbCrLf;
    Tab(1)
    Me.Print " 营业外收入 "; Tab(60); GetIncome(" 营业外收入 ", crcProfit) &
    vbCrLf; Tab(10)
    Me.Print " 减: 营业外支出 "; Tab(60); GetIncome(" 营业外支出 ", crcProfit)
```



```
& vbCrLf; Tab(1)
    Me.Print " 利润总额 "; Tab(60); crcProfit & vbCrLf; Tab(10)
End Sub
```

### 5. 资产负债表报表的设计

资产负债表报表用来统计两类信息：资产信息和负债及所有者权益。其中资产信息包括：现金和现金等价物、应收款项、坏账准备、应收账款净值、存货、流动资产总计、固有资产原值、折旧以及固定资产净值等。负债及所有者权益包括：应付账款、应付股利、短期借款、即将到期的长期借款、长期借款、负债总计、实收资本、资本公积、盈余公积、未分配利润以及所有者权益总计等。

在报表中有标签及文本框等对象，表20-10所示为公司资产负债表报表的控件属性设置。



表 20-10 资产负债表报表属性值

对象名称	属性名称	属性值	含义
Text69 (文本框控件)	控件来源	=Date()	日期
Text14 (文本框控件)	控件来源	=GetSum("现金")+GetSum("银行存款")	应收账款
Text16 (文本框控件)	控件来源	=GetSum("坏账准备")	减：坏账准备
Text22 (文本框控件)	控件来源	=GetSum("存货")	存货
Text24 (文本框控件)	控件来源	=GetSum("","","流动资产")	流动资产总计
Text26 (文本框控件)	控件来源	=GetSum("固定资产")	固定资产原值
Text30 (文本框控件)	控件来源	=GetSum("固定资产折旧")	减：折旧
Text32 (文本框控件)	控件来源	=GetSum("固定资产")-GetSum("固定资产折旧")	固定资产净值
Text37 (文本框控件)	控件来源	=GetSum("应付股利")	应付股利
Text39 (文本框控件)	控件来源	=GetSum("短期借款")	短期借款
Text41 (文本框控件)	控件来源	=GetSum("即将到期的长期借款")	即将到期的长期借款
Text43 (文本框控件)	控件来源	=GetSum("长期借款")	长期借款
Text45 (文本框控件)	控件来源	=GetSum("","","负债")	负债
Text47 (文本框控件)	控件来源	=GetSum("实收资本")	实收资本
Text49 (文本框控件)	控件来源	=GetSum("资本公积")	资本公积
Text51 (文本框控件)	控件来源	=GetSum("盈余公积")	盈余公积
Text53 (文本框控件)	控件来源	=GetSum("未分配利润")	未分配利润
Text55 (文本框控件)	控件来源	=GetSum("","","所有者权益")	所有者权益
Text59 (文本框控件)	控件来源	=GetSum("应收账款")	应收账款



### 6. 设计账务查询窗体

“账务查询”窗体中有文本、框架对象和按钮等对象，表 20-11 所示为系统的“账务查询”窗体的控件属性设置。



表 20-11 账务查询窗体属性值

对象名称	属性名称	属性值
主体	特殊效果	平面
cmdDaybookQuery (按钮控件)	标题	日记账查询
	单击	[事件过程]
cmdClassifyQuery (按钮控件)	标题	分类账查询
	单击	[事件过程]
cmdCreateReports (按钮控件)	标题	财务报表
	单击	[事件过程]

要实现窗体中按钮的各项功能，需要对按钮添加事件过程，下面来为每个按钮添加事件代码。

(1) “分类账查询”按钮 (cmdClassifyQuery) 代码设计。cmdClassifyQuery 按钮的功能是，单击“分类账查询”按钮时，打开分类账查询窗体。

(2) “财务报表”按钮 (cmdCreateReports) 代码设计。cmdCreateReports 按钮的功能是，单击“财务报表”按钮时，打开财务报表窗体。



(3) “日记账查询”按钮 (cmdDaybookQuery) 代码设计。cmdDaybookQuery 按钮的功能是，单击“日记账查询”按钮时，打开日记账查询窗体。



### 7. 日记账查询窗体的设计

日记账查询窗体中有文本框、标签、复选框和按钮等对象，表 20-12 所示为系统的日记账查询窗体的控件属性设置。

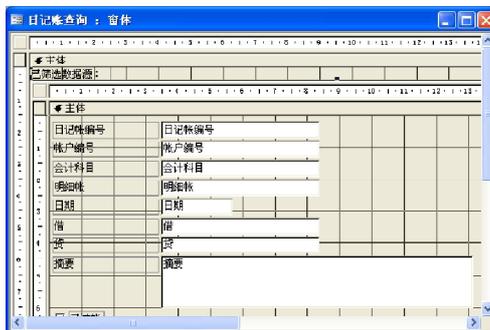


表 20-12 日记账查询窗体属性值

对象名称	属性名称	属性值
主体	特殊效果	平面
cmdRefresh (按钮控件)	标题	刷新
	单击	[事件过程]
cmdOpenReport (按钮控件)	标题	打开报表
	单击	[事件过程]
cmdExit (按钮控件)	标题	关闭
	单击	[事件过程]

下面为按钮编写代码。

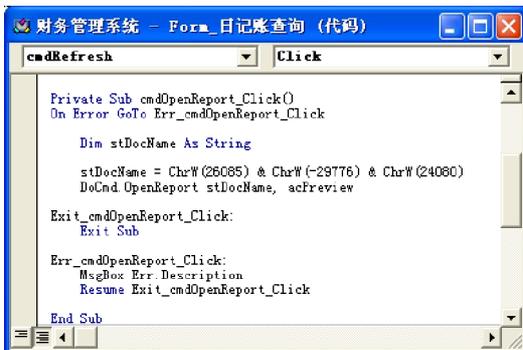
(1) 窗体初始化代码设计。窗体初始化代码用于为日记账查询设定起始和结束日期。



(2) “刷新”按钮 (cmdRefresh) 代码设计。cmdRefresh 按钮的功能是，单击“刷新”按钮时，根据设定的起始和结束日期生成新的日记账查询结果。



(3) “打开报表”按钮 (cmdOpenReport) 代码设计。cmdOpenReport 按钮的功能是，单击“打开报表”按钮时，打开日记账查询报表。



(4) “关闭”按钮 (cmdExit) 代码设计。cmdExit 按钮的功能是，单击“关闭”按钮时，关闭日记账查询窗体。





### 8. 分类账查询窗体的设计

分类账查询窗体中有文本框、标签、组合框和按钮等对象，表20-13所示为系统的分类账查询窗体的控件属性设置。

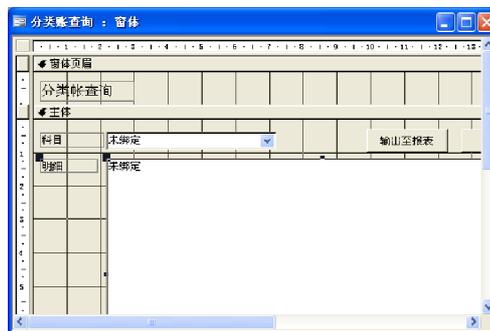


表 20-13 分类账查询窗体属性值

对象名称	属性名称	属性值
Label16 (标签控件)	标题	分类账查询
Label14 (标签控件)	标题	明细
cboTableName_标签 (标签控件)	标题	科目
cboTableName (组合框控件)	行来源类型	值列表
	单击	[事件过程]
cmdReport (按钮控件)	标题	输出至报表
	单击	[事件过程]
cmdExit (按钮控件)	标题	关闭
	单击	[事件过程]

下面为按钮编写代码。

(1) 窗体初始化代码设计。窗体初始化用于对科目标签后的组合框进行初始化。



(2) WriteDetail 过程代码设计。WriteDetail 过程用于格式输出分类账明细，代码如下。

```

Private Sub WriteDetail(iTableIndex As Integer)
    Dim crcLeft As Currency
    'crcLeft 记录账户的余额
    txtDetail = "日期" & Space(15) & "借" & Space(12) & "贷" & Space(5)
    & "摘要" & vbCrLf ' 首行
    strCurAccountNo = LeftB(cboTableName.ItemData(iTableIndex), InStrB
(cboTableName.ItemData(iTableIndex), " ") - 1)
    'strCurAccountNo 取“abcd-wxyz”格式的日记账中的“abcd”部分，即账户编号
    '处理账户建立时余额
    OpenRS "SELECT * FROM 帐户列表 WHERE 帐户编号 = '" & strCurAccountNo &
    
```



```
"" , rs
    crcLeft = rs("借方余额") - rs("贷方余额")
    ' 根据借贷来决定输出位置和内容
    If crcLeft > 0 Then
        txtDetail = txtDetail & Format(rs("建立日期"), "yyyy-mm-dd")
    & Space(2) & Format(crcLeft, "#####") & Space(16) & "期初余额" &
vbCrLf
    Else
        txtDetail = txtDetail & Format(rs("建立日期"), "yyyy-mm-dd")
    & Space(16) & Format(Abs(crcLeft), "#####") & Space(2) & "期初余额"
    " & vbCrLf
    End If
    ' 处理分类账
    strTableName = "分类账" & strCurAccountNo
    OpenRS "SELECT * FROM " & strTableName, rs
    Do Until rs.EOF
        If rs("借") > 0 Then
            txtDetail = txtDetail & Format(rs("日期"), "yyyy-mm-dd") &
Space(2) & Format(rs("借"), "#####") & Space(16) & rs("摘要") &
vbCrLf
        Else
            txtDetail = txtDetail & Format(rs("日期"), "yyyy-mm-dd") &
Space(16) & Format(rs("贷"), "#####") & Space(2) & rs("摘要") &
vbCrLf

        End If
        crcLeft = crcLeft + rs("借") - rs("贷")
        rs.MoveNext
    Loop
    ' 输出汇总结果
    txtDetail = txtDetail + String(77, "-") + vbCrLf
    If crcLeft > 0 Then
        txtDetail = txtDetail & Space(12) & Format(crcLeft,
"#####") & Space(16) & "期末余额" & vbCrLf
    Else
        txtDetail = txtDetail & Space(26) & Format(Abs(crcLeft),
"#####") & Space(2) & "期末余额" & vbCrLf
    End If
End Sub
```

(3) “科目”组合框(cboTableName)的Change事件代码设计。cboTableName组合框的Change事件用于在改变cboTableName(当前分类账)的内容同时改变txtDetail的内容。





(4) “输出至报表”按钮 (cmdReport) 代码设计。cmdReport 按钮的功能是，单击“输出至报表”按钮时，将分类账查询的结果输出至报表显示。

```

财务管理系统 - Form_分类账查询 (代码)
cmdReport Click
Private Sub cmdReport_Click()
    ' 打开分类账报表。这个报表的内容和txtDetail一样
    DoCmd.OpenReport "分类账", acViewPreview
End Sub
    
```

(5) “关闭”按钮 (cmdExit) 代码设计。cmdExit 按钮的功能是，单击“关闭”按钮时，关闭分类账查询窗体。

```

财务管理系统 - Form_分类账查询 (代码)
cmdExit Click
' 命令按钮向导生成代码
Private Sub cmdExit_Click()
    On Error GoTo Err_cmdExit_Click

    DoCmd.Close

Exit_cmdExit_Click:
    Exit Sub

Err_cmdExit_Click:
    MsgBox Err.Description
    Resume Exit_cmdExit_Click

End Sub
    
```

### 9. 财务报表窗体的设计

财务报表窗体中有文本框、标签和按钮等对象，表 20-14 所示为系统的财务报表窗体的控件属性设置。这里只是列出了一些重要的属性设置，没有列出的均为默认设置。



表 20-14 输出报表窗体属性值

对象名称	属性名称	属性值
Label 3 (标签控件)	标题	起始日期
Label 5 (标签控件)	标题	结束日期
Label 8 (标签控件)	标题	损益表日期
cmdBS (按钮控件)	标题	生成资产负债表
	单击	[事件过程]
cmdExit (按钮控件)	标题	关闭
	单击	[事件过程]
cmdIS (按钮控件)	标题	生成损益表
	单击	[事件过程]

下面为按钮编写代码。

(1) 窗体初始化代码设计。窗体初始化用于设置报表的默认时间。

```

财务管理系统 - Form_财务报表 (代码)
cmdBS Click
Private Sub Form_Load()
    ' 设置报表的默认时间
    txtBegin = GetLastDate(Date) ' 本月第一天
    txtEnd = Date ' 当前时间
End Sub
    
```

(2) “生成资产负债表”按钮 (cmdBS) 代码设计。cmdBS 按钮的功能是，单击“生成资产负债表”按钮时，打开资产负债表报表。

```

财务管理系统 - Form_财务报表 (代码)
cmdBS Click
Private Sub cmdBS_Click()
    ' 打开资产负债表报表
    ' 为全局变量 dDateStart、dDateEnd赋值
    dDateStart = txtBegin ' 实际不需要
    dDateEnd = txtEnd
    DoCmd.OpenReport "资产负债表", acViewPreview
End Sub
    
```

(3) “生成损益表”按钮 (cmdIS) 代码设计。cmdIS 按钮的功能是，单击“生成损益表”按钮时，打开损益表报表。

```
Private Sub cmdIS_Click()
    '打开损益表报表
    '为全局变量dDateStart, dDateEnd赋值
    dDateStart = txtBegin
    dDateEnd = txtEnd
    DoCmd.OpenReport "损益表", acViewPreview
End Sub
```

### 账务结算窗体模块的设计

在账务结算窗体中有文本框、框架对象和按钮等对象，表20-15所示为系统的账务结算窗体的控件属性设置。

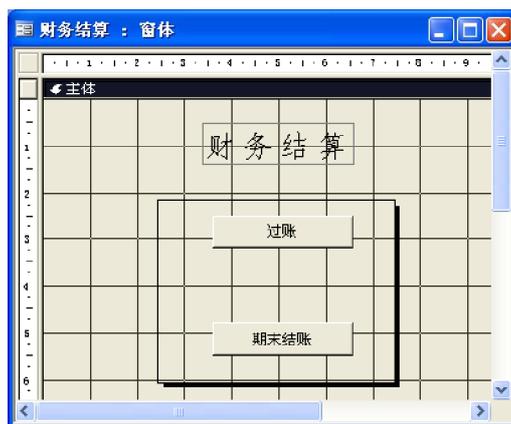


表 20-15 账务结算窗体属性值

对象名称	属性名称	属性值
主体	特殊效果	平面
cmdPosting (按钮控件)	标题	过账
	单击	[事件过程]
cmdCloseAccount (按钮控件)	标题	期末结账
	单击	[事件过程]

下面为按钮编写代码。

(1) “过账”按钮 (cmdPosting) 代码设计。cmdPosting 按钮的功能是，单击“过账”按钮时，将日记账中所有未过账的记录过账到分类账。

```
Private Sub cmdPosting_Click()
    If MsgBox("即将将日记账中所有未过帐的记录过帐到分类账") = vbYes Then
        Posting
    End Sub
End Sub
```

(2) “期末结账”按钮 (cmdCloseAccount) 代码设计。cmdCloseAccount 按钮的功能是，单击“期末结账”按钮时，将收益和费用账户结转至收益汇总账户。

```
Private Sub cmdCloseAccount_Click()
    If MsgBox("即将将收益和费用帐户结转至收益汇总帐户") = vbYes Then
        MakeIncomeSum
    End If
    If MsgBox("即将将收益汇总帐户结转至未分配利润", vbYesNo) = vbYes Then
        MakeProfit
    End If
End Sub
```



## 🔑 账务分析窗体模块的设计

### 1. 账务分析主窗体设计

“账务分析”窗体中有文本框、框架和按钮等对象，表20-16所示为系统的账务分析窗体的控件属性设置。

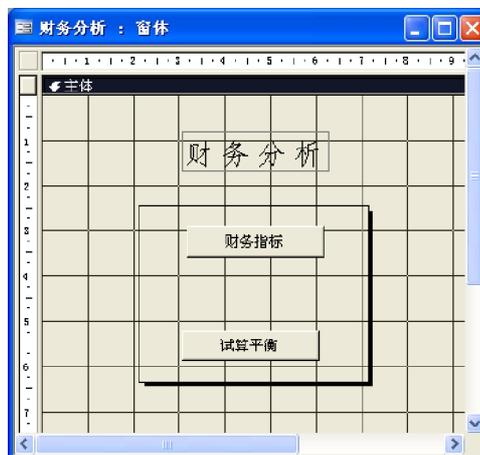


表 20-16 账务分析窗体属性值

对象名称	属性名称	属性值
主体	特殊效果	平面
cmdShowFinance (按钮控件)	标题	财务指标
	单击	[事件过程]
cmdTrialBalance (按钮控件)	标题	试算平衡
	单击	[事件过程]

下面为按钮编写代码。

(1) “财务指标”按钮 (cmdShowFinance) 代码设计。cmdShowFinance 按钮的功能是，单击“财务指标”按钮时，打开财务指标报表。



(2) “试算平衡”按钮 (cmdTrialBalance) 代码设计。cmdTrialBalance 按钮的功能是，单击“试算平衡”按钮时，打开试算平衡窗体。



### 2. 试算平衡窗体设计

在试算平衡窗体中有标签和按钮等对象，表20-17所示为系统的试算平衡窗体的控件属性设置。

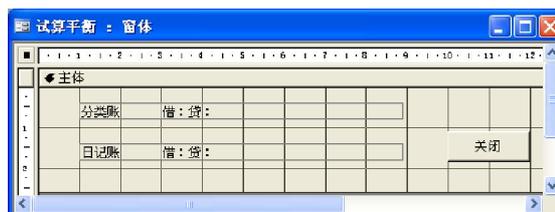




表 20-17 试算平衡窗体属性值

对象名称	属性名称	属性值
Label 0 (标签控件)	标题	分类账
Label 2 (标签控件)	标题	日记账
Lbl LedgerBalance (标签控件)	标题	借: 贷:
Lbl JournalBalance (标签控件)	标题	借: 贷:
cmdExit (按钮控件)	标题	关闭
	单击	[事件过程]

下面为按钮编写代码。

#### (1) 窗体初始化代码设计

窗体初始化用于在打开该窗体时，自动计算分类账和日记账借贷信息，代码如下。

```
Private Sub Form_Load()
    Dim rs As ADODB.Recordset, rs2 As ADODB.Recordset
    Dim crcLeft As Currency, crcRight As Currency
    ' crcLeft、crcRight 分别保存借、贷余额汇总
    ' 统计所有分类账
    OpenRS "SELECT * FROM 账户列表", rs
    Do Until rs.EOF
        crcLeft = crcLeft + rs("借方余额")
        crcRight = crcRight + rs("贷方余额")

        OpenRS "SELECT SUM(借),SUM(贷) FROM 分类账" & rs("账户编号"), rs2
        On Error Resume Next ' 防止下两个赋值语句当rs2()=Null时产生错误提示
        crcLeft = crcLeft + rs2(0)
        crcRight = crcRight + rs2(1)
        rs.MoveNext
    Loop
    LblLedgerBalance.Caption = "借:" & crcLeft & " 贷:" & crcRight
    If crcLeft = crcRight Then
        LblLedgerBalance.Caption = LblLedgerBalance.Caption & " 平衡"
    Else
        ' 不平衡时以红字显示
        LblLedgerBalance.ForeColor = vbRed
        LblLedgerBalance.Caption = LblLedgerBalance.Caption & " 不平衡"
    End If

    ' 统计所有日记账
    crcLeft = 0
    crcRight = 0
    OpenRS " SELECT SUM(借),SUM(贷) FROM 日记账", rs
    crcLeft = rs(0)
    crcRight = rs(1)
    LblJournalBalance.Caption = "借:" & crcLeft & " 贷:" & crcRight
End Sub
```



```

If crcLeft = crcRight Then
    lblJournalBalance.Caption = lblJournalBalance.Caption & " 平衡 "
Else
    lblJournalBalance.ForeColor = vbRed
    lblJournalBalance.Caption = lblJournalBalance.Caption & " 不平衡 "
End If
End Sub
    
```

(2) “关闭”按钮 (cmdExit) 代码设计。  
cmdExit 按钮的功能是，单击“关闭”按钮时，  
关闭试算平衡窗体。





Access 2003



# part

# 21

## 项目管理系统

- ◆ 系统总体设计
- ◆ 数据库设计
- ◆ 系统模块设计

本章将介绍项目管理系统的开发和设计过程，通过参与此系统的开发可以使读者了解管理信息系统的需求分析、设计和实现的完整过程，掌握Access数据库的基本设计和管理方法、常用VBA语言的使用方法，从而积累有效的数据库应用系统实战经验。



在本章中只介绍了本系统中包含数据表的逻辑设计，并没有详细介绍其设计过程，读者可根据前面介绍的方法进行操作。

## 1

## 系统总体设计

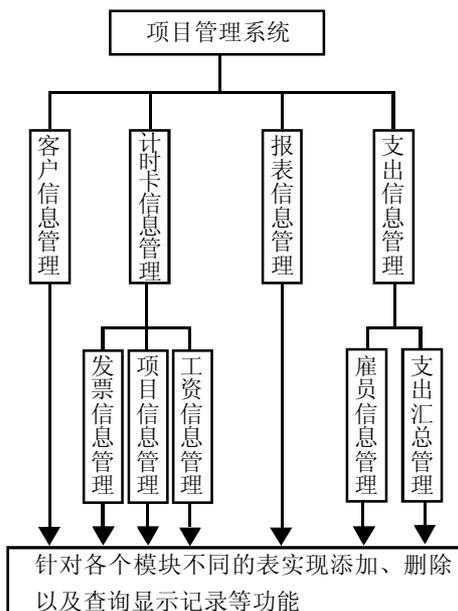
Access

要开发一个管理信息系统，首先需要进行需求分析和总体设计，分析系统的使用对象和用户需求，设计系统的体系结构和数据库结构，决定使用的开发工具和后台数据库，规划项目开发进度。在实际的项目开发过程中，这些工作是非常重要的。

项目管理系统的用户是企业的财务人员，实例包括客户信息管理、计时卡基本信息管理、雇员信息管理、其他信息管理等主要功能，具体功能如下。

- 客户信息的添加、修改、删除和查询。此功能为客户信息的分类管理提供依据。
- 计时卡信息的添加、修改、删除和查询。
- 支出信息的添加、修改、删除和查询。
- 付款方式的添加、修改、删除和查询。
- 报表信息管理。

本节将根据需求分析中总结的用户需求设计系统的体系结构。系统的功能模块图如下图所示。



## 2

## 数据库设计

Access



## 数据表逻辑结构设计

根据实际情况分别确定数据表中各字段的名称、数据类型和值域范围等，并对各表进行数据结构



设计、关键字设计和约束设计等。

### (1) 客户表的设计

客户表的逻辑结构设计如表21-1所示。

表 21-1 客户表

字段名	数据类型	字段宽度	格式	索引	说明
ClientID	自动编号	长整型	标准	有(无重复)	企业自定义格式的项目编号
CompanyName	文本	50	标准	有(有重复)	客户所在公司名称
Address	文本	255	标准	无	客户住址
City	文本	50	标准	无	客户所在的城市
StateOrProvince	文本	20	标准	无	客户所在的省/自治区
Postal Code	文本	20	标准	有(有重复)	邮政编码
Country	文本	50	标准	无	客户所在国家
ContactFirstName	文本	30	标准	无	公司联系该客户的联系人姓名
ContactLastName	文本	50	标准	有(有重复)	公司联系该客户的联系人姓氏
ContactTitle	文本	50	标准	无	该联系人的职务
PhoneNumber	文本	30	标准	无	联系人电话号码
FaxNumber	文本	30	标准	无	联系人传真号
ReferredBy	文本	50	标准	无	推荐人
Notes	备注		标准	无	备注信息

### (2) 项目表的设计

项目表的逻辑结构设计如表21-2所示。

表 21-2 项目表

字段名	数据类型	字段宽度	格式	索引	说明
ProjectID	自动编号	长整型	标准	有(无重复)	企业自定义格式的项目编号
ProjectName	文本	50	标准	有(有重复)	项目名称
ProjectDescription	备注		标准	有(有重复)	项目说明
ClientID	数字	长整型	标准	无	项目客户编号
PurchaseOrderNumber	文本	30	标准	无	订单编号
ProjectTotalBillingEstimate	货币		货币	无	项目预计总费用
EmployeeID	数字	长整型	标准	有(有重复)	负责项目的雇员编号
ProjectBeginDate	日期/时间		短日期	无	项目开始日期
ProjectEndDate	日期/时间		短日期	无	项目结束日期

### (3) 计时卡表的设计

计时卡表的逻辑结构设计如表21-3所示。

表 21-3 计时卡表

字段名	数据类型	字段宽度	格式	索引	说明
TimeCardID	自动编号	长整型	标准	有(无重复)	企业自定义格式的计时卡编号
EmployeeID	数字	长整型	标准	有(有重复)	雇员编号
DateEntered	日期/时间		短日期	无	计时卡输入日期

### (4) 付款表的设计

付款表的逻辑结构设计如表21-4所示。



表 21-4 付款表

字段名	数据类型	字段宽度	格式	索引	说明
PaymentID	自动编号	长整型	标准	有(无重复)	企业自定义格式的付款编号
ProjectID	数字	长整型	标准	有(有重复)	项目的编号
PaymentAmount	货币		货币	无	项目付款的金额
PaymentDate	日期/时间		短日期	无	项目付款的日期
CreditCardNumber	文本	30	标准	无	付款信用卡号
CardholdersName	文本	50	标准	无	付款信用卡持有者姓名
CreditCardExpDate	日期/时间		短日期	无	付款信用卡到期日期
CreditCardAuthorizationNumber	文本	30	标准	无	信用卡授权号
PaymentMethodID	数字	长整型	标准	有(有重复)	付款方式编号

### (5) 付款方式表的设计

付款方式表的逻辑结构设计如表21-5所示。

表 21-5 付款方式表

字段名	数据类型	字段宽度	格式	索引	说明
PaymentMethodID	自动编号	长整型	标准	有(无重复)	企业自定义格式的付款方式编号
PaymentMethod	文本	50	标准	无	付款方式
CreditCard	是/否		是/否	无	选择是否使用信用卡

### (6) 雇员表的设计

雇员表的逻辑结构设计如表21-6所示。

表 21-6 雇员表

字段名	数据类型	字段宽度	格式	索引	说明
EmployeeID	自动编号	长整型	标准	有(无重复)	企业自定义格式的雇员编号
FirstName	文本	50	标准	无	雇员名
LastName	文本	50	标准	有(有重复)	雇员姓
Title	文本	50	标准	无	雇员的职务
EmailName	文本	50	标准	有(有重复)	雇员的电子邮件地址
Extension	文本	30	标准	无	雇员电话分机号
Address	文本	255	标准	无	雇员所住住址
City	文本	50	标准	无	雇员所处城市
StateOrProvince	文本	20	标准	无	雇员所在省/自治区
Postal Code	文本	20	标准	有(有重复)	邮政编码
Country	文本	50	标准	无	雇员所在国家
HomePhone	文本	30	标准	无	雇员的家庭电话
WorkPhone	文本	30	标准	无	雇员的工作电话
BillingRate	货币		货币	无	费率

### (7) 支出代码表的设计

支出代码表的逻辑结构设计如表21-7所示。

表 21-7 支出代码表

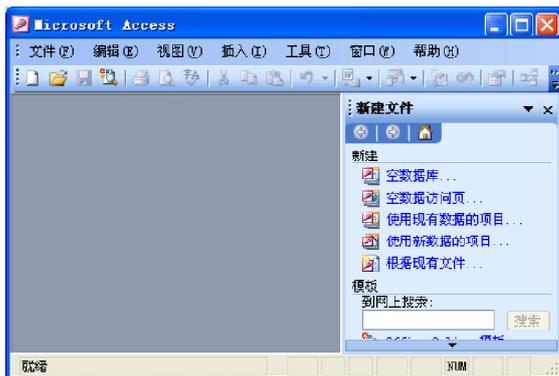
字段名	数据类型	字段宽度	格式	索引	说明
ExpenseCodeID	自动编号	长整型	标准	有（无重复）	企业定义格式支出代码编号
ExpenseCode	文本	30	标准	无	支出代码

## 数据表设计

创建或打开“项目管理系统”数据库以后，就可以开始在其中创建和设计相应的数据表。在建立数据库后，本节以项目表的设计为例进行介绍，其他数据表的设计方法类似。

### 1. 建立“项目管理系统”数据库

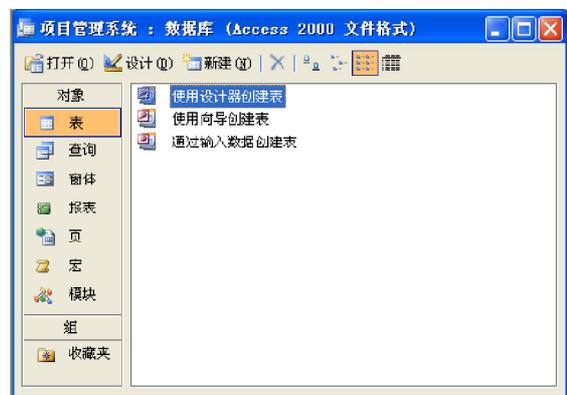
- 1 运行 Access，单击“新建文件”任务窗格的“新建”选项区中的“空数据库”选项。
- 2 在弹出的“文件新建数据库”对话框中进行相应设置，保存后“项目管理系统”数据库便建好了，下面就可以根据不同的工作需要为数据库创建数据表、查询等数据库对象。



### 2. 项目表的设计

在Access中收集来的信息都存储在数据表中，每一个表都由数据字段和数据记录（行和列）组成。表中的字段内容设计得好与差，对于日后表的维护、查询以及窗体的制作等有着直接的影响。

- 1 运行 Access，打开“项目管理系统”数据库。



- 在“项目管理系统：数据库”窗口中，选择“对象”列表框中的“表”选项，双击窗口右侧的“通过输入数据创建表”选项创建表。



- 向表中输入字段，输入方法是：双击图中的“字段1”、“字段2”……此时它们会呈反黑显示，这样就可以输入字段名称，输入完所有的字段后的“项目”表如下图所示。



- 保存以后，单击工具栏中的“设计视图”按钮回到表的“设计”视图中为字段设置属性。



### 3

## 系统模块设计

### 设计主界面

本系统的主界面是一个切换面板，具体的设计过程参见第18章的最后一节。下图所示为系统的主切换面板。

单击主切换面板中相应的选项即可进入不同界面，例如，单击“输入 / 编辑其他信息”选项可以打开其子切换面板。





## 客户管理窗体模块设计

客户管理模块用来管理客户详细信息,并且通过单击窗体中的功能按钮可以查看和当前客户相关的项目、计时、付款和发票信息。

### 1. 设计项目窗体

项目窗体用于管理与主窗体中客户相关的项目信息,并且用户还可以通过单击“查看支出”按钮查看该项目的支出信息。

#### (1) 设计支出子窗体

在设计项目窗体之前需要首先设计项目支出子窗体,项目支出子窗体用于管理项目的支出信息。相应的文本框属性如表 21-8 所示。



表 21-8 文本框属性说明

文本框名称	标签标题	“控件来源”属性内容
Total Expenses	总支出	=Sum([ExpenseAmount])
ExpenseDate	日期	ExpenseDate
Employee Name	雇员	Employee Name
ExpenseCode	支出代码	ExpenseCode
ExpenseAmount	金额	ExpenseAmount
ExpenseDescription	支出说明	ExpenseDescription

#### (2) 设计项目子窗体

项目子窗体设计视图如右图所示,相应的文本框属性如表 21-9 所示。



表 21-9 文本框属性说明

文本框名称	标签标题	“控件来源”属性内容
Total Hours Worked	已工作总小时数	=Sum([BillableHours])
Total Billed	已付总额	=Sum([BillableHours]*[BillingRate])
DateWorked	工作日期	DateWorked
Employee Name	雇员	Employee Name
WorkCode	工作代码	WorkCode
BillingRate	费率	BillingRate
BillableHours	可付款工时数	BillableHours
WorkDescription	工作说明	WorkDescription

### (3) 设计项目主窗体

项目窗体设计视图如右图所示，相应的控件属性如表 21-10 所示。



表 21-10 控件属性说明

对象名称	属性名称	属性值
Toggle View (按钮控件)	标题	查看支出
	单击	[事件过程]
项目支出子窗体	源对象	项目支出子窗体
Total Hours Worked (文本框控件)	控件来源	=IIf([项目子窗体]. [Form]. [RecordsetClone]. [RecordCount]=0, 0, [项目子窗体]. [Form]! [Total HoursWorked])
Total Expenses (文本框控件)	控件来源	=IIf([项目支出子窗体]. [Form]. [RecordsetClone]. [RecordCount]=0, 0, [项目支出子窗体]. [Form]! [Total Expenses])
Total Payments (文本框控件)	控件来源	=IIf(IsNull ([ProjectID]), 0, DSum("[PaymentAmount]", "付款", "[ProjectID]=" & [ProjectID]))
Total HourlyBillings (文本框控件)	控件来源	=IIf([项目子窗体]. [Form]. [RecordsetClone]. [RecordCount]=0, 0, [项目子窗体]. [Form]! [Total Billed])
Total Billable Amount (文本框控件)	控件来源	=[Total HourlyBillings]+[Total Expenses]
Amount Due (文本框控件)	控件来源	=[Total Billable Amount]-nz([Total Payments])

下面为该主窗体编写事件代码。主窗体的事件代码主要包括窗体的初始化事件代码和按钮的单击事件。

窗体初始化事件代码如下。

```

Private Sub Form_Current()      ' 当项目编号为空时将焦点至于项目名称
    If IsNull (Me! [ProjectID]) Then
        DoCmd. GoToControl "ProjectName"
    End If
End Sub

Private Sub Form_Open(Cancel As Integer)      ' 提示用户通过单击"客户"窗体
中的"项目"按钮打开窗体
    If Not IsLoaded("客户") Then
        MsgBox "使用"客户"窗体上的"项目"按钮可打开"项目"窗体。"
        Cancel = True
    End If

```



```

End Sub

Private Sub Form_Activate()      ' 根据项目主窗体中的项目编号来显示相应子窗体信息
On Error GoTo Err_Form_Activate
    Me.Requery
    If IsLoaded("客户") Then
        If Forms![客户]![客户子窗体].Form.RecordsetClone.RecordCount > 0 Then
            DoCmd.GoToControl "ProjectID"
            DoCmd.FindRecord Forms![客户]![客户子窗体].Form![ProjectID]
            DoCmd.GoToControl "ProjectName"
        End If
    End If
Exit_Form_Activate:
    Exit Sub
Err_Form_Activate:
    MsgBox Err.Description
    Resume Exit_Form_Activate
End Sub

Private Sub Form_BeforeInsert(Cancel As Integer)
On Error GoTo Err_Form_BeforeInsert
    Me![ClientID] = Forms![客户]![ClientID]
Exit_Form_BeforeInsert:
    Exit Sub
Err_Form_BeforeInsert:
    MsgBox Err.Description
    Resume Exit_Form_BeforeInsert
End Sub

Private Sub Form_Load()
    If Me.OpenArgs = "GotoNew" And Not IsNull([ProjectID]) Then
        DoCmd.DoMenuItem acFormBar, 3, 0, , acMenuVer70
    End If
End Sub

```

EmployeeID 组合框事件代码如下。

```

Private Sub EmployeeID_NotInList(NewData As String, Response As Integer)
    MsgBox " 双击此自字段可添加记录到列表。"
    Response = acDataErrContinue
End Sub

Private Sub EmployeeID_DbClick(Cancel As Integer)      ' 双击可打开相应的"雇员"窗体

```



```

On Error GoTo Err_EmployeeID_DbClick
    Dim lngProjectManagerID As Long
    If IsNull (Me![EmployeeID]) Then
        Me![EmployeeID].Text = ""
    Else
        lngProjectManagerID = Me![EmployeeID]
        Me![EmployeeID] = Null
    End If
    DoCmd.OpenForm "雇员", , , , , acDialog, "GotoNew"
    Me![EmployeeID].Requery
    If lngProjectManagerID <> 0 Then Me![EmployeeID] = lngProjectManagerID
Exit_EmployeeID_DbClick:
    Exit Sub
Err_EmployeeID_DbClick:
    MsgBox Err.Description
    Resume Exit_EmployeeID_DbClick
End Sub

```

单击窗体中的按钮可以在支出和工时两个窗体之间切换，代码如下。

```

Private Sub Toggle_View_Click()
    If Me![Toggle View].Caption = "查看支出(&V)" Then
        Me![项目子窗体].Visible = False
        Me![项目支出子窗体].Visible = True
        Me![Toggle View].Caption = "查看工时(&V)"
    Else
        Me![项目支出子窗体].Visible = False
        Me![项目子窗体].Visible = True
        Me![Toggle View].Caption = "查看支出(&V)"
    End If
End Sub

```

## 2. 设计付款窗体

付款窗体用于管理与主窗体中客户相关的付款信息。付款窗体设计视图如右图所示，相应的控件属性如表 21-11 所示。

控件名称	属性值
项目 ID	ProjectID
项目	= [Forms]![客户]![客户子窗体]
付款方式	PaymentMethodID
付款日期	PaymentDate
付款金额	PaymentAmount
信用卡持有者姓名	CardholdersName
信用卡号	CreditCardNumber
信用卡到期日期	CreditCardExpDate
信用卡授权号	CreditCardAuthorizationNu



表 21-11 控件属性说明

对象名称	属性名称	属性值
ProjectID (文本框控件)	控件来源	ProjectID
Project Name (文本框控件)	控件来源	= [Forms]![客户]![客户子窗体].[Form]![ProjectName]
PaymentMethodID (组合框控件)	控件来源	PaymentMethodID
PaymentDate (文本框控件)	控件来源	PaymentDate
PaymentAmount (文本框控件)	控件来源	PaymentAmount
CardholdersName (文本框控件)	控件来源	CardholdersName

下面为该主窗体编写事件代码。主窗体的事件代码主要包括窗体的初始化事件代码和按钮的单击事件。

### 1. 窗体初始化事件代码

窗体初始化事件代码如下。

```
Private Sub Form_Open(Cancel As Integer) ' 提示用户通过单击“客户”窗
    体中的“项目”按钮打开窗体
    If Not IsLoaded("客户") Then
        MsgBox "使用“客户”窗体上的“付款”按钮可打开“付款”窗体。"
        Cancel = True
    End If
End Sub

Private Sub Form_Activate()
On Error GoTo Err_Form_Activate
    Me.Requery
Exit_Form_Activate:
    Exit Sub
Err_Form_Activate:
    MsgBox Err.Description
    Resume Exit_Form_Activate
End Sub

Private Sub Form_BeforeInsert(Cancel As Integer)
On Error GoTo Err_Form_BeforeInsert
    Me![PaymentDate] = Date
    Me![PaymentAmount] = [Forms]![客户]![客户子窗体].Form![Amount Due]
Exit_Form_BeforeInsert:
    Exit Sub
Err_Form_BeforeInsert:
    MsgBox Err.Description
    Resume Exit_Form_BeforeInsert
End Sub
```



## 2. PaymentMethodID 组合框事件代码

PaymentMethodID 组合框事件代码如下。

```
Private Sub PaymentMethodID_AfterUpdate()  
On Error GoTo Err_PaymentMethodID_AfterUpdate  
    If Not IsNull(Me![PaymentMethodID]) Then  
        If Me![PaymentMethodID].Column(2) = True Then  
            Me![CardholdersName] = [Forms]![客户]![ContactFirstName] & " "  
& [Forms]![客户]![ContactLastName]  
        End If  
    End If  
Exit_PaymentMethodID_AfterUpdate:  
    Exit Sub  
Err_PaymentMethodID_AfterUpdate:  
    MsgBox Err.Description  
    Resume Exit_PaymentMethodID_AfterUpdate  
End Sub  
  
Private Sub PaymentMethodID_NotInList(NewData As String, Response As Integer)  
    MsgBox " 双击此自字段可添加记录到列表。"  
    Response = acDataErrContinue  
End Sub  
  
Private Sub PaymentMethodID_DbClick(Cancel As Integer)  
On Error GoTo Err_PaymentMethodID_DbClick  
    Dim lngPaymentMethodID As Long  
    If IsNull(Me![PaymentMethodID]) Then  
        Me![PaymentMethodID].Text = ""  
    Else  
        lngPaymentMethodID = Me![PaymentMethodID]  
        Me![PaymentMethodID] = Null  
    End If  
    DoCmd.OpenForm " 付款方式", , , , , acDialog, "GotoNew"  
    Me![PaymentMethodID].Requery  
    If lngPaymentMethodID <> 0 Then Me![PaymentMethodID] =  
lngPaymentMethodID  
Exit_PaymentMethodID_DbClick:  
    Exit Sub  
Err_PaymentMethodID_DbClick:  
    MsgBox Err.Description  
    Resume Exit_PaymentMethodID_DbClick  
End Sub
```



### 3. 设计打印发票窗体

打印发票窗体用于预览、打印与主窗体中客户相关的项目发票信息，并且用户还可以通过单击窗体中的按钮在工时和支出窗体间进行切换，同时还可以预览发票报表。

#### (1) 设计打印支出发票子窗体

在设计预览发票窗体之前需要首先设计打印支出发票子窗体。打印支出发票子窗体用于打印项目的支出信息，该子窗体设计视图如右图所示，相应的文本框属性如表 21-12 所示。



表 21-12 文本框属性说明

文本框名称	标签标题	“控件来源”属性内容
Total Expenses	总支出	=Sum([ExpenseAmount])
ExpenseDate	支出日期	ExpenseDate
Employee Name	雇员	Employee Name
ExpenseCode	支出代码	ExpenseCode
ExpenseAmount	金额	ExpenseAmount
ExpenseDescription	支出说明	ExpenseDescription

#### (2) 设计打印计时发票子窗体

打印计时发票子窗体设计视图如右图所示，相应的文本框属性如表 21-13 所示。



表 21-13 文本框属性说明

文本框名称	标签标题	“控件来源”属性内容
Total HoursWorked	总小时数	=Sum([BillableHours])
Total Billed	已付总额	=Sum([BillableHours]*[BillingRate])
DateWorked	工作日期	DateWorked
Employee Name	雇员	Employee Name
WorkCode	工作代码	WorkCode
BillingRate	费率	BillingRate
BillableHours	可付款工时数	BillableHours
WorkDescription	工作说明	WorkDescription



### (3) 设计发票报表

在设计预览发票窗体之前需要设计发票报表，发票报表用于显示支出和计时信息。发票报表的设计视图如右图所示，相应的文本框属性如表 21-14 所示。



表 21-14 文本框说明

文本框名称	标签标题	“控件来源”属性内容
MyCompany	控件来源	=DLookup("[CompanyName]","我的公司信息")
MyAddress	控件来源	=DLookup("[Address]","我的公司信息")
MyAddress2	控件来源	=DLookup("[City]","我的公司信息") & " " & DLookup("[StateOrProvince]","我的公司信息") & " " & DLookup("[Postal Code]","我的公司信息")
myPhoneandFax	控件来源	="电话: " & DLookup("[PhoneNumber]","我的公司信息") & " 传真: " & DLookup("[FaxNumber]","我的公司信息")
Invoice Date	控件来源	=[Forms]![打印发票]![Invoice Date]
Invoice Number	控件来源	=[Forms]![打印发票]![Invoice Number]
Payment Terms	控件来源	=[Forms]![打印发票]![Payment Terms]
Project Name	控件来源	=[Forms]![打印发票]![Project Name]
Project ID	控件来源	=[Forms]![打印发票]![Project ID]
Invoice Expense Subreport	源对象	报表. 计时发票子报表
InvoiceTotal	控件来源	=[Forms]![打印发票]![Total Expenses]+[Forms]![打印发票]![Total Hourly Billings]
InvoiceDescription	控件来源	=[Forms]![打印发票]![Invoice Description]

### 4. 设计打印发票主窗体

打印发票窗体设计视图如右图所示，相应的控件属性如表 21-15 所示。





表 21-15 控件属性说明

对象名称	属性名称	属性值
Toggle View	标题	查看支出
	单击	[事件过程]
Preview Invoice	标题	预览发票
	单击	[事件过程]
打印支出发票子窗体	源对象	打印支出发票子窗体
Total Hours	控件来源	=IIf([打印计时发票子窗体].[Form].[RecordsetClone].[RecordCount]=0,0,[打印计时发票子窗体].[Form]![Total Hours])
Total Expenses	控件来源	=IIf([打印支出发票子窗体].[Form].[RecordsetClone].[RecordCount]=0,0,[打印支出发票子窗体].[Form]![Total Expenses])
Total Hourly Billings	控件来源	=IIf([打印计时发票子窗体].[Form].[RecordsetClone].[RecordCount]=0,0,[打印计时发票子窗体].[Form]![Total Billed])
Amount Due	控件来源	=[Total Hourly Billings]+[Total Expenses]

下面为该窗体编写事件代码。窗体的事件代码主要包括窗体的初始化事件代码和按钮的单击事件。

窗体初始化事件代码如下图所示。

```

Private Sub Form_Open(Cancel As Integer)
    If Not IsLoaded("客户") Then
        MsgBox "使用“客户”窗体上的“预览发票”按钮可打  
Cancel = True
    End If
End Sub
Private Sub Begin_Date_AfterUpdate()
    Me! [打印计时发票子窗体]. Requery
    Me! [打印支出发票子窗体]. Requery
End Sub
Private Sub End_Date_AfterUpdate()
    Me! [打印计时发票子窗体]. Requery
    Me! [打印支出发票子窗体]. Requery
End Sub
    
```

“查看支出”按钮单击事件代码如下图所示。

```

Private Sub Toggle_View_Click()
    If Me! [Toggle View].Caption = "查看支出 (&V)" Then
        Me! [打印计时发票子窗体]. Visible = False
        Me! [打印支出发票子窗体]. Visible = True
        Me! [Toggle View].Caption = "查看计时 (&V)"
    Else
        Me! [打印支出发票子窗体]. Visible = False
        Me! [打印计时发票子窗体]. Visible = True
        Me! [Toggle View].Caption = "查看支出 (&V)"
    End If
End Sub
    
```

“预览发票”按钮单击事件代码如下图所示。

```

Private Sub PreviewInvoice_Click()
    On Error GoTo Err_PreviewInvoice_Click
    DoCmd.OpenReport "发票", acPreview, "[ProjectID]=" & Me.Visible = False
Exit_PreviewInvoice_Click:
    Exit Sub
Err_PreviewInvoice_Click:
    MsgBox Err.Description
    Resume Exit_PreviewInvoice_Click
End Sub
    
```

### 5. 设计客户主窗体

#### (1) 设计客户子窗体

在设计客户窗体之前需要首先设计客户子窗体，客户子窗体用于管理客户所负责的项目信息、工时信息、总付款以及未付款信息。客户子窗体设计视图如右图所示，相应的文本框属性如表 21-16 所示。

名称	属性值
项目名称	ProjectName
总工时	总工时
总金额	=[应付金额]+[总支出]
付款	总付款
未付金额	=[Total Amount]-nz([Payme



表 21-16 文本框属性说明

文本框名称	标签标题	“控件来源”属性内容
Total Hours	总工时	总工时
Total Amount	总金额	=[应付金额]+[总支出]
Payments	付款	总付款
Amount Due	未付金额	=[Total Amount]-nz([Payments])

下面介绍客户子窗体的事件过程，当双击该窗体的文本框时，将打开相应项目的详细信息。这里为客户子窗体编写一个过程，在双击各文本框时都将调用这个过程来打开项目窗体，事件代码如下。

```
Option Compare Database
Option Explicit

Private Sub ViewProject()
On Error GoTo Err_ViewProject
    DoCmd.OpenForm "项目"
Exit_ViewProject:
    Exit Sub
Err_ViewProject:
    MsgBox Err.Description
    Resume Exit_ViewProject
End Sub

Private Sub ProjectName_DblClick(Cancel As Integer)
    ViewProject
End Sub

Private Sub Payments_DblClick(Cancel As Integer)
    DoCmd.OpenForm "付款"
End Sub
```

(2) 设计客户主窗体

客户窗体可以编辑与客户相关的所有信息。系统用户可以根据自己的需要，单击相应的功能按钮打开不同窗体进行操作。客户主窗体设计视图如右图所示，相应的控件属性如表21-17所示。





表 21-17 控件属性说明

对象名称	属性名称	属性值
Projects (按钮控件)	标题	项目
	单击	[事件过程]
TimeCards (按钮控件)	标题	计时卡
	单击	[事件过程]
Payments (按钮控件)	标题	付款
	单击	[事件过程]
PrintInvoice (按钮控件)	标题	预览发票
	单击	[事件过程]
客户子窗体	源对象	客户子窗体

下面为该主窗体编写事件代码。主窗体的事件代码主要包括窗体的初始化事件代码和按钮的单击事件。

窗体初始化事件代码完成根据窗体中的客户编号来初始化客户子窗体中的项目信息。即当窗体中显示某个客户时，子窗体即对应显示相应的项目信息。代码及注释如下。

```
Private Sub Form_Current() ' 当客户编号为空时将焦点至于姓氏文本框
    If IsNull(Me![ClientID]) Then
        DoCmd.GoToControl "ContactFirstName"
    End If
End Sub

Private Sub Form_Activate() ' 根据主窗体中的客户信息来给出子窗体信息
On Error GoTo Err_Form_Activate
    Me![客户子窗体].Requery

Exit_Form_Activate:
    Exit Sub

Err_Form_Activate:
    MsgBox Err.Description
    Resume Exit_Form_Activate
End Sub
```

按钮单击事件代码用于打开和当前客户信息相关的其他（项目、计时卡）信息。

“项目”按钮事件代码如下。

```
Private Sub Projects_Click()
On Error GoTo Err_Projects_Click
    If IsNull(Me![ClientID]) Then
        MsgBox " 在查看项目窗体请输入客户信息。"
    Else
        DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
acMenuVer70
```



```
        DoCmd.OpenForm "项目", , , "[ClientID]=" & [ClientID]
    End If

Exit_Projects_Click:
    Exit Sub

Err_Projects_Click:
    MsgBox Err.Description
    Resume Exit_Projects_Click
End Sub
```

“计时卡”按钮事件代码如下。

```
Private Sub TimeCards_Click()
On Error GoTo Err_TimeCards_Click
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
    DoCmd.OpenForm "计时卡"
Exit_TimeCards_Click:
    Exit Sub

Err_TimeCards_Click:
    MsgBox Err.Description
    Resume Exit_TimeCards_Click
End Sub
```

“付款”按钮事件代码如下。

```
Private Sub Payments_Click()
On Error GoTo Err_Payments_Click
    If Forms![客户]![客户子窗体].Form.RecordsetClone.RecordCount = 0 Then
        MsgBox "在查看付款窗体前请输入该用户的项目。"
    Else
        DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
acMenuVer70
        DoCmd.OpenForm "付款"
    End If

Exit_Payments_Click:
    Exit Sub

Err_Payments_Click:
    MsgBox Err.Description
    Resume Exit_Payments_Click
End Sub
```



“预览发票”按钮事件代码如下。

```
Private Sub PrintInvoice_Click()
On Error GoTo Err_PrintInvoice_Click
    If Forms![客户]![客户子窗体].Form.RecordsetClone.RecordCount = 0 Then
        MsgBox "在查看发票前请输入客户和项目信息。"
    Else
        DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
acMenuVer70
        DoCmd.OpenForm "打印发票", , , , , acDialog
    End If

Exit_PrintInvoice_Click:
    Exit Sub

Err_PrintInvoice_Click:
    MsgBox Err.Description
    Resume Exit_PrintInvoice_Click
End Sub
```

## 🔑 计时卡管理模块设计

计时卡管理模块用来管理雇员工时和支出的详细信息,通过单击窗体中的功能按钮可以查看和当前雇员相关的其他信息。

### 1. 设计计时卡子窗体

在设计计时卡主窗体之前需要首先设计计时卡子窗体,计时卡子窗体用于管理雇员的工时信息。计时卡子窗体设计视图如右图所示,相应的控件属性如表 21-18 所示。



表 21-18 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
TotalHoursWorked	已工作总小时数	=Sum([BillableHours])
DateWorked	工作日期	DateWorked
ProjectID	项目	ProjectID
WorkCode	工作代码	WorkCode
BillingRate	费率	BillingRate
BillableHours	可付款工时数	BillableHours\
WorkDescription	工作说明	WorkDescription





表 21-19 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
Total Expenses	总支出	=Sum([ExpenseAmount])
ExpenseDate	日期	ExpenseDate
ProjectID	项目	ProjectID
ExpenseCode	支出代码	ExpenseCode
ExpenseAmount	金额	ExpenseAmount
ExpenseDescription	说明	ExpenseDescription

### 3. 设计时间报表

在设计计时卡窗体之前需要设计时间报表，时间报表用于显示各雇员所记录的支出和计时信息。时间报表的设计视图如右图所示，相应的控件属性如表 21-20 所示。



表 21-20 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
Employee Name	控件来源	Employee Name
DateEntered	控件来源	DateEntered
Time Sheet Subreport	源对象	报表. 时间表子报表
Time Sheet Expenses Subreport	源对象	报表. 时间表支出子报表
Text9	控件来源	=Now()

### 4. 设计计时卡主窗体

计时卡主窗体设计视图如右图所示，相应的控件属性如表 21-21 所示。





表 21-21 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
PreviewTimeCard (按钮控件)	标题 单击	预览计时卡 [事件过程]
计时卡子窗体	源对象	计时卡子窗体
计时卡支出子窗体	源对象	计时卡支出子窗体
Total BillableHours	控件来源	=[计时卡子窗体].[Form]![Total Hours Worked]
Total Expenses	控件来源	=[计时卡支出子窗体].[Form]![Total Expenses]

下面为该主窗体编写事件代码。主窗体的事件代码主要包括窗体的初始化事件代码和按钮的单击事件。

窗体初始化事件代码完成根据窗体中的计时卡编号来初始化计时卡和计时卡支出子窗体中的工时和支出信息。代码及注释如下。

```
Private Sub Form_Current()
    If IsNull(Me![TimeCardID]) Then
        DoCmd.GoToControl "EmployeeID"
    End If
End Sub

Private Sub Form_Activate()
On Error GoTo Err_Form_Activate
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
    Me![计时卡子窗体].Form![ProjectID].Requery
    Me![计时卡支出子窗体].Form![ProjectID].Requery
Exit_Form_Activate:
    Exit Sub
Err_Form_Activate:
    MsgBox Err.Description
    Resume Exit_Form_Activate
End Sub
```

EmployeeID组合框事件代码如下。

```
Private Sub EmployeeID_NotInList(NewData As String, Response As Integer)
    MsgBox " 双击此自字段可添加记录到列表。"
    Response = acDataErrContinue
End Sub

Private Sub EmployeeID_DbClick(Cancel As Integer)
On Error GoTo Err_EmployeeID_DbClick
    Dim lngEmployeeID As Long
    If IsNull(Me![EmployeeID]) Then
        Me![EmployeeID].Text = ""
    Else
```



```

        lngEmployeeID = Me![EmployeeID]
        Me![EmployeeID] = Null
    End If
    DoCmd.OpenForm "雇员", , , , , acDialog, "GotoNew"
    Me![EmployeeID].Requery
    If lngEmployeeID <> 0 Then Me![EmployeeID] = lngEmployeeID
Exit_EmployeeID_DblClick:
    Exit Sub
Err_EmployeeID_DblClick:
    MsgBox Err.Description
    Resume Exit_EmployeeID_DblClick
End Sub

Private Sub EmployeeID_Click()
    Me![计时卡子窗体].Requery
End Sub

```

按钮单击事件代码如下。

```

Private Sub PreviewTimeCard_Click()
    On Error GoTo PreviewTimeCard_Err
    If IsNull(Me![TimeCardID]) Then
        MsgBox "预览"计时卡"报表前请输入计时卡信息。"
    Else
        DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
acMenuVer70
        DoCmd.OpenReport "时间表", acPreview, , "[TimeCardID]=" &
[TimeCardID]
    End If
PreviewTimeCard_Exit:
    Exit Sub
PreviewTimeCard_Err:
    MsgBox Err.Description
    Resume PreviewTimeCard_Exit
End Sub

```

## 账单查询管理模块设计

根据用户输入的开始日期和结束日期，系统可以分别按照项目、工作代码给出雇员账单、客户账单以及项目账单报表。因此账单查询管理模块的设计包括一个管理界面和三个报表。下面首先设计报表。

### 1. 设计雇员账单（按项目）报表

雇员账单（按项目）报表用于显示给出一段时期内某雇员的总支出信息，其设计视图如右图所示，相应的控件属性如表 21-22 所示。



表 21-22 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
Rpt Date Range	控件来源	=“执行工作的时间范围: 从 ” & [Forms]![报表日期范围]![BeginDate] & ” 到 ” & [Forms]![报表日期范围]![EndDate]
Text14	控件来源	=“项目名称为 ” & [ProjectName] & ” 的汇总 (” & Count(*) & ” ” & IIf(Count(*)=1,“条详细记录”,“条详细记录”) & ” ) ”
Text12	控件来源	=“公司名称为 ” & [CompanyName] & ” 的汇总 (” & Count(*) & ” ” & IIf(Count(*)=1,“条详细记录”,“条详细记录”) & ” ) ”
Sum Of Hours Worked	控件来源	=Sum([Hours Worked])
Sum Of Hourly Billings	控件来源	=Sum([Hourly Billings])
Text28	控件来源	=Now()

雇员账单（按项目）报表必须在输入开始日期和结束日期后才能给出报表结果，因此需要编写报表的Open事件来完成这一操作，如下所示。

```
Private Sub Report_NoData(Cancel As Integer)
    MsgBox " 此报表没有数据。取消报表..."
    Cancel = -1
End Sub
Private Sub Report_Close()
    DoCmd.Close acForm, " 报表日期范围 "
End Sub
Private Sub Report_Open(Cancel As Integer)
    DoCmd.OpenForm " 报表日期范围", , , , acDialog, " 雇员账单 (按项目) "
    If Not IsLoaded(" 报表日期范围 ") Then
        Cancel = True
    End If
End Sub
```



### 2. 设计客户账单（按项目）报表

客户账单（按项目）报表用于显示给出一段时期内某客户的总支出信息，其设计视图如右图所示，相应的控件属性如表 21-23 所示。



表 21-23 控件属性说明

文本框名称	标签标题	“控件来源”属性内容
Rpt Date Range	控件来源	=“执行工作的时间范围：从” & [Forms]![报表日期范围]![BeginDate] & “到” & [Forms]![报表日期范围]![EndDate]
Text10	控件来源	=“公司名称为” & [CompanyName] & “的汇总(“ & Count(*) & “” & IIf(Count(*)=1,“条详细记录”,“条详细记录”)& “”
Sum Of Hours Worked	控件来源	=Sum([Hours Worked])
Sum Of Hourly Billings	控件来源	=Sum([Hourly Billings])
Text25	控件来源	=Now()

客户账单（按项目）报表必须在输入开始日期和结束日期后才能给出报表结果，因此需要编写报表的Open事件来完成这一操作，如下所示。

```

Private Sub Report_NoData(Cancel As Integer)
    MsgBox "此报表没有数据。取消报表..."
    Cancel = -1
End Sub
Private Sub Report_Close()
    DoCmd.Close acForm, "报表日期范围"
End Sub
Private Sub Report_Open(Cancel As Integer)
    DoCmd.OpenForm "报表日期范围", , , , , acDialog, "客户账单（按项目）"
    If Not IsLoaded("报表日期范围") Then
        Cancel = True
    End If
End Sub
    
```

### 3. 设计项目账单（按工作代码）报表

项目账单（按工作代码）报表用于显示给出一段时期内项目的总支出、工时以及工时费用等信息，其设计视图如右图所示，相应的控件属性如表 21-24 所示。





表 21-24 控件属性说明

文本框名称	属性名称	属性内容
Rpt Date Range	控件来源	= "执行工作的时间范围: 从 " & [Forms]![报表日期范围]![BeginDate] & " 到 " & [Forms]![报表日期范围]![EndDate]
Text14	控件来源	= "项目名称为 " & [ProjectName] & " 的汇总 (" & Count(*) & " " & IIf(Count(*)=1, "条详细记录", "条详细记录") & ")"
Text12	控件来源	= "公司名称为 " & [CompanyName] & " 的汇总 (" & Count(*) & " " & IIf(Count(*)=1, "条详细记录", "条详细记录") & ")"
Sum Of Hours Worked	控件来源	= Sum([Hours Worked])
Sum Of Hourly Billings	控件来源	= Sum([Hourly Billings])
Text28	控件来源	= Now()

项目账单（按工作代码）报表必须在输入开始日期和结束日期后才能给出报表结果，因此需要编写报表的Open事件来完成这一操作，如下所示。

```
Private Sub Report_NoData(Cancel As Integer)
    MsgBox " 此报表没有数据。取消报表..."
    Cancel = -1
End Sub
Private Sub Report_Close()
    DoCmd.Close acForm, " 报表日期范围 "
End Sub
Private Sub Report_Open(Cancel As Integer)
    DoCmd.OpenForm " 报表日期范围", , , , , acDialog, " 项目帐单（按工作代码）"
    If Not IsLoaded(" 报表日期范围 ") Then
        Cancel = True
    End If
End Sub
```

#### 4. 设计报表日期范围窗体

报表日期范围窗体设计视图如右图所示，相应的控件属性如表 21-25 所示。



表 21-25 控件属性说明

文本框名称	属性名称	属性值
BeginDate（文本框控件）	格式	短日期
	输入掩码	0000-99-99;0;
EndDate（文本框控件）	格式	短日期
	输入掩码	0000-99-99;0;
Preview（按钮控件）	标题	预览
	单击	事件过程



下面为该窗体编写事件代码。主窗体的事件代码即按钮的单击事件。当单击“预览”按钮时将显示输入日期范围内的报表信息，具体代码如下所示。

```
Private Sub Preview_Click()  
    If IsNull([BeginDate]) Or IsNull([EndDate]) Then  
        MsgBox "必须输入开始日期和结束日期。"  
        DoCmd.GoToControl "BeginDate"  
    Else  
        If [BeginDate] > [EndDate] Then  
            MsgBox "结束日期必须大于开始日期。"  
            DoCmd.GoToControl "BeginDate"  
        Else  
            Me.Visible = False  
        End If  
    End If  
End Sub
```



Access 2003



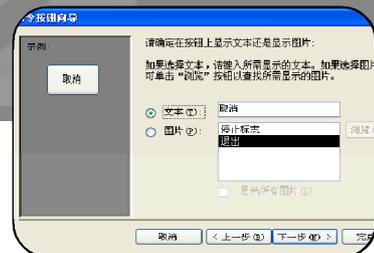
# part

# 22

## 资产管理系统

- ◆ 系统总体设计
- ◆ 数据库设计
- ◆ 系统模块设计

本章所设计的资产管理系统能够完成资产信息和资产借出信息的增加、删改、查询以及打印功能。该系统使用Access 2003进行系统开发，读者应该具备基本的数据库设计知识以及掌握Access 2003基本的操作方法，才可顺利进行设计和开发。





在本章中只介绍了本系统中包含数据表的逻辑设计，并没有详细介绍其设计过程，读者可根据前面介绍的方法进行操作。

## 1 系统总体设计

### 系统需求分析

资产管理系统开发的总体任务是实现资产管理和借还的信息化操作。在资产管理系统中需要完成以下基本功能。

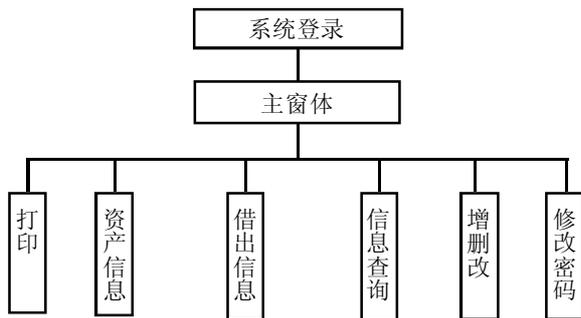
- 购入、调入的资产按照实际支付的买价或调拨价、包装费、运杂费、安装费和海关税费等记账，购置车辆按规定支付的车辆购置附加费计入购价之内。自行建造的资产，按照建造过程中实际发生的全部支出入账。融资租入、接受捐赠或盘盈的资产按有关规定入账。
- 增加的资产按月提取折旧，统一实行平均年限法。资产折旧年限（使用年限）由总经理办公会议在研究购建资产时一并确定。例如，2002年1月1日起购建的资产，按企业会计制度核算的公司（中心）一律以年限平均法计提折旧，按高等学校会计制度核算的公司（中心）不计提折旧。
- 当月增加的资产，下月计提折旧，当月减少的资产当月照提折旧，下月起不提折旧，资产提足折旧后，不管是否继续使用均不再提取折旧，提前报废的资产也不补提折旧。
- 资产要有专人保管，定期维护，以延长其使用寿命，对暂时闲置的资产应安全存放保养，对长期闲置的资产应进行有偿调配，以充分发挥其使用价值。

### 系统设计

根据以上需求分析，系统主要功能模块分析如下。

- 登录功能（登录窗体）：允许注册用户输入用户名和口令，从而登录系统。
- 主模块（主窗体）：显示各条资产信息记录，并且为用户提供数据记录浏览的按钮，以及其他所有主要模块窗体的链接按钮；显示所有资产信息（资产信息）；该窗体以表格的形式显示了所有资产信息记录。
- 资产信息查询（信息查询）：该窗体提供了一系列选项，用户选择想要查询的条件。
- 添加、删除及修改数据信息：添加、删除及修改主窗口中显示的数据信息。
- 打印和打印预览：打印和打印预览资产信息报表；打印和打印预览借出信息报表；打印和打印预览借出到期信息报表；打印和打印预览信息查询结果。
- 修改密码：修改某个特定用户的密码。
- 报废产品：报废一种资产。

根据前面对用户需求的分析，依据系统功能设计原则，对整个系统进行了模块划分，系统的模块结构图如下图所示。



## 2 数据库设计 Access

### 数据表逻辑结构设计

根据功能要求在 Access 里建立各数据表的结构如下所述。

(1) 系统用户表的逻辑结构设计如表 22-1 所示。

表 22-1 注册用户的资料表（系统用户）

字段名	数据类型	说明
用户ID	自动编号	用户ID
用户名	文本	用户的真实姓名
密码	文本	用户密码

(2) 为了资产管理以及查询的方便，需要建立资产的资料表，资产信息表的逻辑结构设计如表 22-2 所示。

表 22-2 资产的资料表（资产信息）

字段名	数据类型	说明
资产编号	文本	资产编号
资产类型	文本	资产类型
资产名称	文本	资产名称
型号	文本	资产型号
配置	文本	资产配置
购买日期	日期/时间	资产购买日期
使用期限	数字	资产使用期限（单位：年）
购买价格	货币	资产使用期限（单位：元）
存放位置	文本	资产存放位置
资产状态	文本	资产状态（在库中、已借出、报废）
可行操作	文本	对应资产状态的操作（借出、还入、回收）

(3) 为了管理以及查询资产的借出和归还信息，需要建立资产的借出资料表，资产借出信息表的逻辑结构设计如表22-3所示。

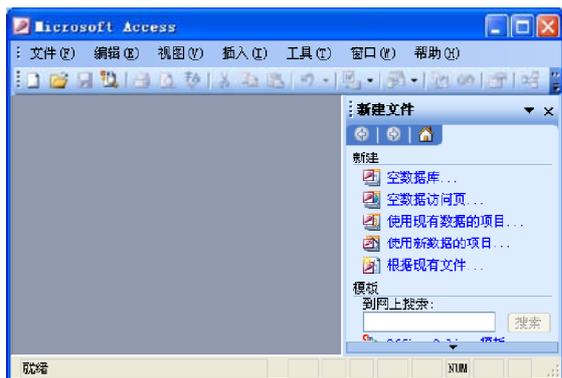
表 22-3 资产的借出资料表 (借出信息)

字段名	数据类型	说明
资产编号	文本	资产编号
借用人	文本	借用人ID
出借人	文本	出借人ID
批复人	文本	批复人ID
出借日期	日期/时间	出借日期
借用理由	文本	借用理由
拟还日期	日期/时间	拟还日期

## 创建数据库

### 1. 建立“资产管理系统”数据库

- 1 运行 Access 2003, 单击“新建文件”任务窗格的“新建”选项区中的“空数据库”选项。



- 2 在弹出的“文件新建数据库”对话框中进行相应设置, 保存后“资产管理系统”数据库便建好了, 下面就可以根据不同的工作需要为数据库创建数据表、查询等数据库对象。



创建或打开“资产管理系统”数据库以后, 就可以开始创建和设计相应的数据表。在建立数据库后, 本节以项目表的设计为例进行介绍, 其他数据表的设计方案类似。

### 2. 建立关系

“资产管理系统”数据库中的表之间是存在一定关系的。需要根据数据表的逻辑设计, 来建立数据库表之间的关系, 具体操作步骤如下。

- 1 在数据库窗口中, 单击工具栏上的“关系”按钮, 弹出“关系”窗口。如果数据库没有定义任何关系, 将会显示一个空白的“关系”窗口, 并且在空白“关系”窗口上弹出“显示表”对话框。





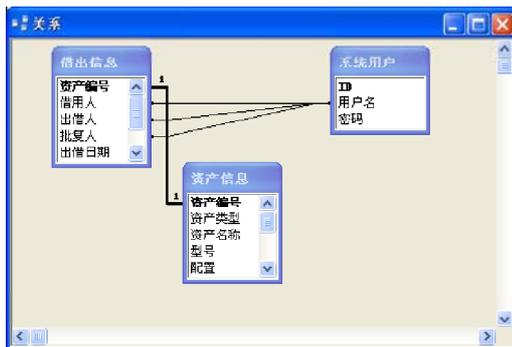
2 在“显示表”对话框的“表”选项卡中双击选取要作为定义关系的表的名称，然后关闭“显示表”对话框，在“关系”窗口中会增加选中的表。这里将数据库中的所有表都添加到“关系”窗口中。



3 定义其他表之间的关系。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，例如，将借出信息表中的“资产编号”字段拖到资产信息表中的“资产编号”字段，会出现“编辑关系”对话框。



4 单击“新建”按钮关闭“编辑关系”窗口，就创建了教师档案表和教师授课信息表之间的关系。教师档案表和教师授课表、教师授课表和课程名表之间的关系可按以上方法创建，最终创建的关系如右图所示。



## 3 系统模块设计

### 登录窗体的设计

当系统运行时，首先打开登录窗体，只有有权限的用户才能进入系统。登录成功后，将显示系统主界面。

#### 1. 登录窗体设计

系统登录窗体的功能是得到用户名和密码，并在“系统用户”数据表中核对用户名和密码是否匹配，然后再进行相关操作。在该窗体的设计视图中包含标签、文本框和按钮等控件，其中的主要控件属性如表22-4所示。





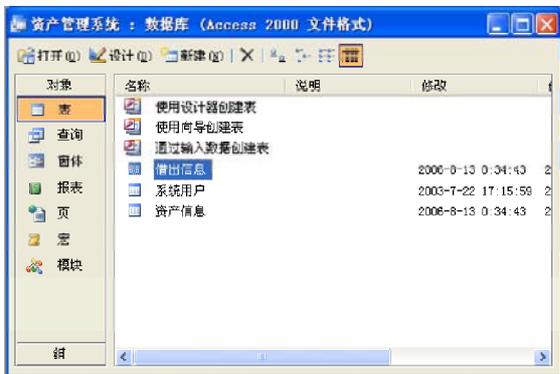
表 22-4 登录窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	用户名
标签5	类型	标签
	标题	密码
com用户	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [系统用户].[用户名], [系统用户].[用户名] FROM 系统用户
txt密码	类型	文本框
	输入掩码	密码
logi_n_ok	类型	按钮
	标题	确定
logi_n_cancel	类型	按钮
	标题	取消

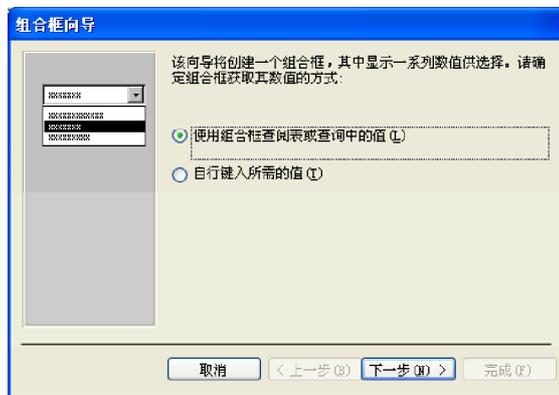
## 2. 添加组合框

需要说明的是，组合框的数据行来源是可以利用 Access 自动向导功能实现的，具体操作步骤如下。

- 1 双击打开数据库，在“对象”列表框中选择“表”选项。
- 2 双击“在设计视图中创建窗体”选项，打开窗体设计视图。



- 3 在窗体设计视图中单击“工具箱”上的“控件向导”按钮，添加一个“组合框”到窗体上，系统会自动弹出“组合框向导”对话框。





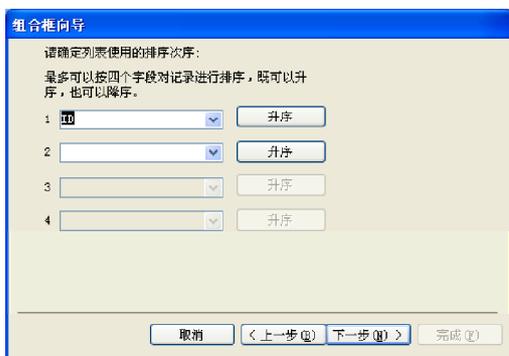
- 保持默认设置后，单击“下一步”按钮，进入下一界面。此时要求用户选择为组合框提供数据的表或查询，这里选择“系统用户”表。



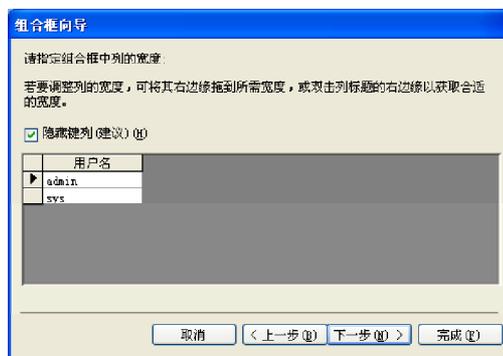
- 单击“下一步”按钮，进入下一界面，此时，要求用户选择包含到组合框中的字段，这里选择“用户名”和“ID”这两个字段。



- 单击“下一步”按钮，进入下一界面，此时需要选择排序的字段，这里选择对ID字段进行升序排序。



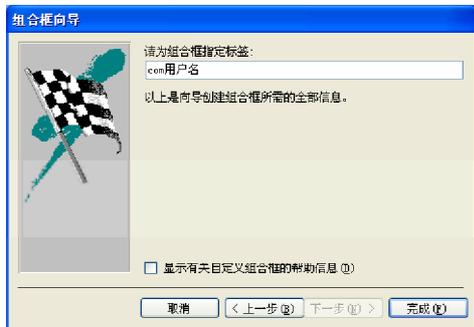
- 单击“下一步”按钮，进入下一界面，此时要求用户指定组何况中列的宽度，另外还要求用户选择是否隐藏列。这里选择不隐藏列，因为如果隐藏了“用户名”列，那么在组合框中只显示ID字段，对用户来说比较难以判断究竟选择哪一项，加上“用户名”字段，可以帮助用户更方便地进行操作。



- 单击“下一步”按钮进入下一界面，此时要求用户从ID和“用户名”两者中选择一个字段，系统将把该字段的值存储到数据库中，或者使用该字段的值来执行某项操作。如果前面没有选择两个字段包含到组合框内，那么向导就不会出现这一界面。



- 9 单击“下一步”按钮进入下一界面，此时要求用户为组合框指定标签，这里输入“com 用户名”。



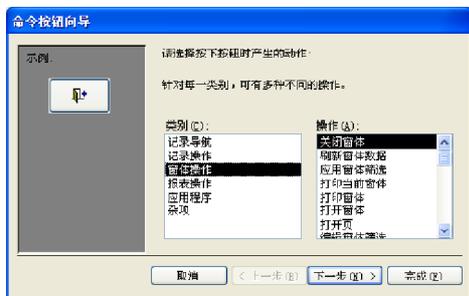
- 10 单击“完成”按钮，最终的效果如下图所示。



### 3. 添加按钮事件处理

下面开始处理按钮的单击事件。首先利用按钮的事件向导实现“取消”按钮的功能，该按钮的功能用于关闭登录窗体，具体操作步骤如下。

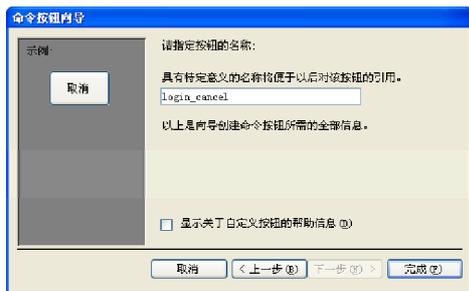
- 1 当用户将一个“命令按钮”控件添加到窗体中时，就会出现“命令按钮向导”对话框。在“类别”列表框中，选择“窗体操作”选项，在“操作”列表框中，选择“关闭窗口体”选项，然后单击“下一步”按钮。



- 2 此时进入的界面如下图所示，这里选中“文本”单选按钮，并在旁边的文本框中输入“取消”，然后单击“下一步”按钮。



- 3 此时进入的界面如下图所示，更改文本框中的按钮名称为 login\_cancel，然后单击“完成”按钮。



- 4 用鼠标右键单击步骤3创建的按钮，在弹出的快捷菜单中选择“事件生成器”命令，出现VBA 代码编辑器，为该按钮的单击事件输入的代码如下图所示。





- 按同样方法制作“确定”按钮，并设置“确定”按钮的单击事件的处理函数。用鼠标右键单击该按钮，在弹出的快捷菜单中选择“事件生成器”命令，在出现的VBA代码编辑器中输入如右图所示的代码。

```

login_ok Click
Private Sub login_ok_Click()
Dim userID As Integer
If IsNull(com用户) = False Then
If DLookup("[密码]", "系统用户", "[用户名]='" & com用户 &
userID = DLookup("[id]", "系统用户", "[用户名]='" & com
DoCmd.Close
DoCmd.OpenForm "主界面"
Form_主界面.userID = userID
Else
txt密码 = ""
txt密码.SetFocus
MsgBox "密码错误!", vbCritical
End If
End If
End Sub
    
```

### 主界面窗体设计

本节要新建各种窗体对象，用以建立主窗体，在主窗体中有许多功能按钮，用户只需按窗体上的按钮，就会启动命令按钮的单击过程，运行过程代码，如资产出借或打印报表等。

在主窗体中有标签、框架和按钮等对象，表 22-5 所示为系统的主窗体的对象属性设置。这里只是列出了一些重要的属性设置，没有列出的均为默认设置。

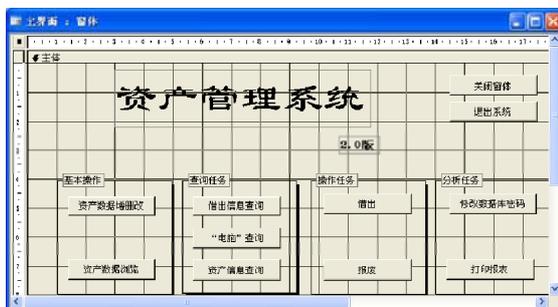


表 22-5 “主界面”窗体属性值

对象名称	属性名称	属性值
Lbl ShowBoard	标题	资产管理系统
Label 15	标题	基本操作
Label 17	标题	查询任务
Label 19	标题	分析任务
Label 21	标题	操作任务
CmdCl assi fyQuery	标题	“电脑”查询
	单击	[事件过程]
CmdCl oseAccount	标题	报废
	单击	[事件过程]
CmdCreateReports	标题	资产信息查询
	单击	[事件过程]
CmdDaybookI nput	标题	资产数据增删改
	单击	[事件过程]
CmdDaybookQuery	标题	借出信息查询
	单击	[事件过程]
CmdExi t	标题	退出系统
	单击	[事件过程]
CmdHi de	标题	关闭窗口
	单击	[事件过程]



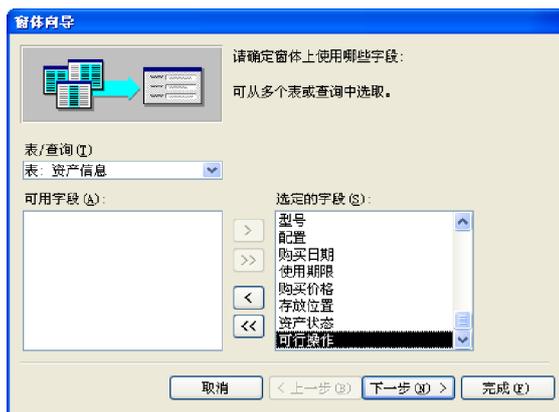
(续表)

对象名称	属性名称	属性值
CmdPosting	标题	借出
	单击	[事件过程]
CmdShowFinance	标题	修改数据库密码
	单击	[事件过程]
CmdTrialBalance	标题	打印报表
	单击	[事件过程]

### 1. 通过窗体向导完成主窗体设计

在这里需要说明的是，由于主窗体中包括了资产信息表的数据字段，如果完全手动加入每个文本框和标签，将会是一件很麻烦的事情，因此在这里可以利用 Access 的窗口向导功能，下面介绍如何利用窗口向导来创建系统主窗体。

- 1 在数据库窗口中选择“窗口”选项并单击“新建”按钮，将会弹出“新建窗体”对话框。
- 2 在对话框中选择“窗体向导”选项，然后单击“确定”按钮，将会弹出“窗体向导”对话框。按照如下图所示进行设置。

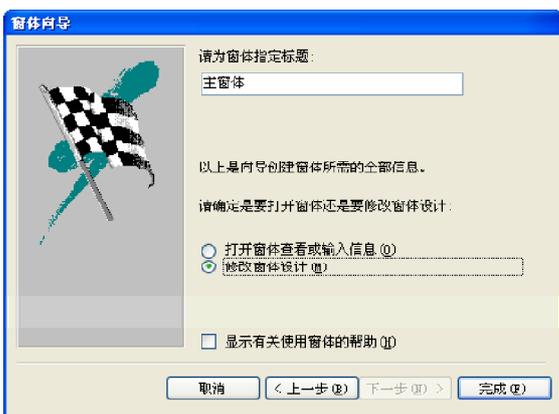


- 3 单击“下一步”按钮进入下一界面，这里选中“纵栏表”单选按钮。
- 4 单击“下一步”按钮进入下一界面，这里选择“混合”样式。





- 单击“下一步”按钮进入下一界面，在其中输入窗体的名称“主窗体”，然后选择“修改窗体设计”选项，进入窗口设计视图，得到窗体初始设计视图。



- 调整控件的位置和大小，然后加入一个“选项组”控件，将所有控件都放到其中。然后在这个“选项组”中删除标签“可行操作”，并将文本框“可行操作”的“特殊效果”属性设定为“凸起”。



## 2. 添加数据浏览按钮事件处理

现在要加入主窗口的数据浏览按钮。首先要确定这个窗口的“记录源”属性是“资产信息”表，然后再按照下面的方法加入4个相关按钮。

加入“转到首记录”按钮。方法是向窗体中添加一个按钮控件，然后在弹出的对话框中选择“记录导航”和“转至第一项记录”选项，然后单击“完成”按钮。在事件生成器中可以查看 Access 自动生成的 VBA 代码。



加入“转到下一条记录”按钮。方法是向窗体中添加一个按钮控件，然后在弹出的对话框中选择“记录导航”和“转至下一项记录”选项，然后单击“完成”按钮。在事件生成器中可以查看 Access 自动生成的 VBA 代码。



加入“转到上一条记录”按钮。方法是向窗体中添加一个按钮控件，然后在弹出的对话框中选择“记录导航”和“转至前一项记录”选项，然后单击“完成”按钮。在事件生成器中可以查看 Access 自动生成的 VBA 代码。



加入“转到最后记录”按钮。方法是向窗体中添加一个按钮控件，然后在弹出的对话框中选择“记录导航”和“转至最后一项记录”选项，然后单击“完成”按钮。在事件生成器中可以查看 Access 自动生成的 VBA 代码。



```

private Sub next_Click()
    On Error GoTo Err_next_Click
    DoCmd GoToRecord , acFirst
    If 资产状态 = "在库中" Then
        btn可行操作.Caption = "借出"
    ElseIf 资产状态 = "出借中" Then
        btn可行操作.Caption = "归还"
    ElseIf 资产状态 = "报废" Then
        btn可行操作.Caption = "回收"
    End If
Exit_sub:
    Exit Sub
Err_next_Click:
    MsgBox Err.Description
    Resume Exit_next_Click
End Sub

```

### 3. 添加其他按钮事件

最后向窗体中加入功能按钮，下面介绍这些功能按钮的事件实现方法。

#### (1) “查询”按钮事件

首先加入“查询”按钮，用户单击这个按钮触发的事件是打开信息查询窗体（这个窗体还没有创建，后面将会详细介绍它的创建方法）。



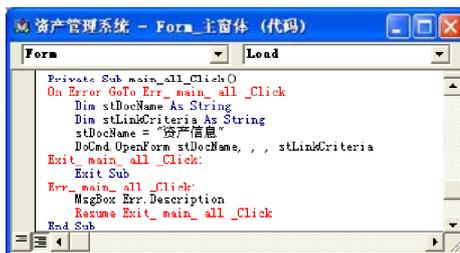
```

private Sub main_query_Click()
    On Error GoTo Err_main_query_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "信息查询"
    DoCmd OpenForm stDocName, , stLinkCriteria
Exit_sub:
    Exit Sub
Err_main_query_Click:
    MsgBox Err.Description
    Resume Exit_main_query_Click
End Sub

```

#### (2) “全部”按钮事件

下面加入“全部”按钮，用户单击这个按钮触发的事件是打开“资产信息窗体”。



```

private Sub main_all_Click()
    On Error GoTo Err_main_all_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "资产信息"
    DoCmd OpenForm stDocName, , stLinkCriteria
Exit_sub:
    Exit Sub
Err_main_all_Click:
    MsgBox Err.Description
    Resume Exit_main_all_Click
End Sub

```

#### (3) “已借出”按钮事件

下面加入“已借出”按钮，用户单击这个按钮触发的事件是打开“借出信息窗体”。



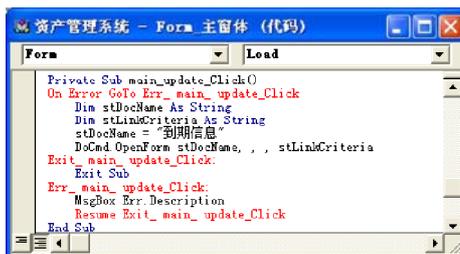
```

private Sub main_lend_Click()
    On Error GoTo Err_main_lend_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "借出信息"
    DoCmd OpenForm stDocName, , stLinkCriteria
Exit_sub:
    Exit Sub
Err_main_lend_Click:
    MsgBox Err.Description
    Resume Exit_main_lend_Click
End Sub

```

#### (4) “已到期”按钮事件

下面加入“已到期”按钮，用户单击这个按钮触发的事件是打开“到期信息窗体”。



```

private Sub main_update_Click()
    On Error GoTo Err_main_update_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "到期信息"
    DoCmd OpenForm stDocName, , stLinkCriteria
Exit_sub:
    Exit Sub
Err_main_update_Click:
    MsgBox Err.Description
    Resume Exit_main_update_Click
End Sub

```



### (5) “打印”按钮事件

下面加入“打印”按钮，用户单击这个按钮触发的事件是打开“打印窗体”。

```
Private Sub main_print_Click()
On Error GoTo Err_main_print_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "定期信息"
DoCmd.OpenForm stDocName, , stLinkCriteria
Exit_main_print_Click:
Exit Sub
Err_main_print_Click:
MsgBox Err.Description
Resume Exit_main_print_Click
End Sub
```

### (6) “密码”按钮事件

下面加入“密码”按钮，用户单击这个按钮触发的事件是打开“修改密码窗体”。

```
Private Sub main_pass_Click()
On Error GoTo Err_main_pass_Click
Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "定期高息"
DoCmd.OpenForm stDocName, , stLinkCriteria
Exit_main_pass_Click:
Exit Sub
Err_main_pass_Click:
MsgBox Err.Description
Resume Exit_main_pass_Click
End Sub
```

### (7) “退出”按钮事件

下面加入“退出”按钮，用户单击这个按钮触发的事件是退出应用程序。

```
Private Sub main_quit_Click()
On Error GoTo Err_命令0_Click
DoCmd.Quit
Exit_命令0_Click:
Exit Sub
Err_命令0_Click:
MsgBox Err.Description
Resume Exit_命令0_Click
End Sub
```

### (8) “可行操作”按钮事件

为“可行操作”文本框加入双击事件处理函数。该按钮的作用是借出或者归还某项资产。方法是在该文本框的事件属性中选择“双击”选项，然后单击  按钮，进入事件生成器，由于这项功能比较复杂，需要直接编辑 VBA 代码，输入下面的代码。

```
Private Sub txtOpt_DblClick(Cancel As Integer)
' 首先定义变量ID和r，其中ID是一个临时字符串，用来记录当前的用户ID，r是一个QueryDef
' 类型的对象，用来实现一个sql操作
    Dim ID As String
    Dim r As QueryDef
' 需要说明的是要利用QueryDef类型的对象必须加入MSDAO的连接库，方法是在VBA编辑器的菜
' 单中选择“工具”下的“引用”，
' 如果“可行操作”按钮上的文本是“借出”，那么就可以借出该项资产。
    If txtOpt = "借出" Then
' 打开“出借资产”窗体，该窗口现在还没有实现，在后面回将接如何创建这个窗体。
        DoCmd.OpenForm "出借资产"
' 更改“出借资产”窗体的各种控件的属性。
        Form_出借资产.txtId = Form_主窗口.文本8
' for debug!
```



```
user = 1
Form_出借资产.ok.Caption = "借出"
' 通过用户ID 查询用户名。
Form_出借资产.txtConfirm = DLookup("[用户名]", "系统用户", "[id]= " & CStr(user))
Form_出借资产.txtOutDate = Date
Form_出借资产.txtInDate = Date
End If
' 如果“可行操作”按钮上的文本是“还入”，那么就可以还入该项资产。
If txtOpt = "还入" Then
' 打开“出借资产”窗体。
DoCmd.OpenForm "出借资产"
' 更改“出借资产”窗体的各种控件的属性。
Form_出借资产.ok.Caption = "归还"
Form_出借资产.Caption = "归还资产"
ID = Form_主窗口.文本8
Form_出借资产.txtId = ID
Form_出借资产.txtUseMan = DLookup("[借用人]", "借出信息", "[资产编号]= "" & ID & """)
Form_出借资产.txtOutMan = DLookup("[出借人]", "借出信息", "[资产编号]= "" & ID & """)
Form_出借资产.txtConfirm = DLookup("[批复人]", "借出信息", "[资产编号]= "" & ID & """)
Form_出借资产.txtOutDate = DLookup("[出借日期]", "借出信息", "[资产编号]= "" & ID & """)
Form_出借资产.txtInDate = DLookup("[拟还日期]", "借出信息", "[资产编号]= "" & ID & """)
Form_出借资产.txtRes = DLookup("[借用理由]", "借出信息", "[资产编号]= "" & ID & """)
End If
```

如果“可行操作”文本框上显示的是“回收”，那么就直接更改该资产的“资产状态”属性，方法是执行一条SQL语句。

#### (9) “报废”按钮事件

最后为“报废”按钮添加事件处理函数。进入代码生成器界面，在其中加入如右图所示的VBA代码。这里同样用到了上面创建的“出借资产状态更新”查询，通过执行一条SQL更新命令实现功能。

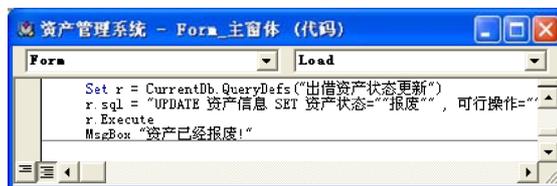




表 22-6 所示为主窗体中控件的主要属性。

表 22-6 主窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号
标签1	类型	标签
	标题	资产类型
标签2	类型	标签
	标题	型号
标签3	类型	标签
	标题	资产名称
标签4	类型	标签
	标题	购买价格
标签5	类型	标签
	标题	配置
标签6	类型	标签
	标题	购买日期
标签7	类型	标签
	标题	使用期限
标签8	类型	标签
	标题	存放位置
标签9	类型	标签
	标题	资产状态
主窗体	类型	窗体
	记录源	资产信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否
	分隔线	否
	自动调整	否
自动居中	是	
文本0	类型	文本框
	控件来源	资产编号
文本1	类型	文本框
	控件来源	资产类型
文本2	类型	文本框
	控件来源	型号
文本3	类型	文本框
	控件来源	资产名称
文本4	类型	文本框
	控件来源	购买价格
文本5	类型	文本框
	控件来源	配置
文本6	类型	文本框
	控件来源	购买日期



(续表)

控件名称	控件属性名称	控件属性值
文本7	类型	文本框
	控件来源	使用期限
文本8	类型	文本框
	控件来源	存放位置
文本9	类型	文本框
	控件来源	资产状态
文本10	类型	文本框
	控件来源	可行操作
	特殊效果	凸出



### 基本操作模块设计

基本操作模块主要完成资产信息的添加、删除、修改及查询等操作，该模块的设计即设计“资产信息”窗体，窗体以表格的形式显示所有资产信息。表 22-7 所示为资产信息窗体中控件的主要属性。



表 22-7 资产信息窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号
标签1	类型	标签
	标题	资产类型
标签2	类型	标签
	标题	型号
标签3	类型	标签
	标题	资产名称
标签5	类型	标签
	标题	配置
标签6	类型	标签
	标题	购买日期
标签7	类型	标签
	标题	使用期限
标签8	类型	标签
	标题	存放位置
标签9	类型	标签
	标题	资产状态
资产信息	类型	窗体
	记录源	资产信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否



(续表)

控件名称	控件属性名称	控件属性值
	分隔线	否
	自动调整	否
	自动居中	是
文本0	类型	文本框
	控件来源	资产编号
文本1	类型	文本框
	控件来源	资产类型
文本2	类型	文本框
	控件来源	型号
文本3	类型	文本框
	控件来源	资产名称
文本4	类型	文本框
	控件来源	购买价格
文本5	类型	文本框
	控件来源	配置
文本6	类型	文本框
	控件来源	购买日期
文本7	类型	文本框
	控件来源	使用期限
文本8	类型	文本框
	控件来源	存放位置
文本9	类型	文本框
	控件来源	资产状态
文本10	类型	文本框
	控件来源	可行操作

### 查询任务模块设计

信息查询窗体的作用是，读入用户输入的查询条件，根据这些条件进行查询。表 22-8 所示为信息查询窗体中控件的主要属性。

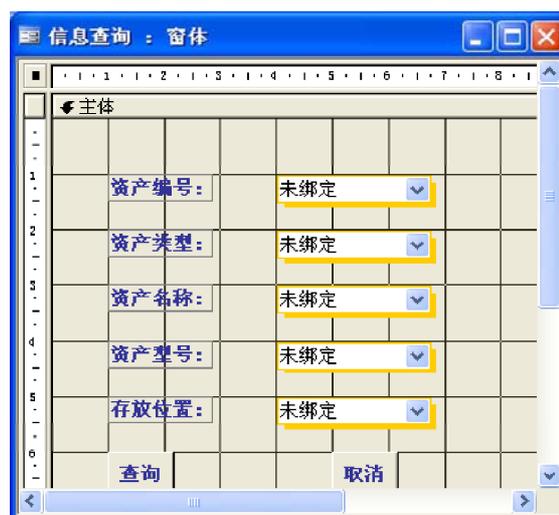




表 22-8 信息查询窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号:
标签1	类型	标签
	标题	资产类型:
标签2	类型	标签
	标题	资产型号:
标签3	类型	标签
	标题	资产名称
主窗体	类型	窗体
	记录源	资产信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否
	分隔线	否
	自动调整	否
自动居中	是	
comID	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [资产信息].[资产编号] FROM资产信息 GROUP BY [资产信息].[资产编号];
comType	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [资产信息].[资产类型] FROM资产信息 GROUP BY [资产信息].[资产类型];
comModel	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [资产信息].[型号] FROM资产信息 GROUP BY [资产信息].[型号];
comName	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [资产信息].[资产名称] FROM资产信息 GROUP BY [资产信息].[资产名称];
comPos	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [资产信息].[存放位置] FROM资产信息 GROUP BY [资产信息].[存放位置];

下面详细说明“查询”按钮的单击事件的处理函数。用鼠标右键单击该按钮，在弹出的快捷菜单中选择“事件生成器”，在出现VBA代码编辑器键入下面的代码。

```
Private Sub query_ok_Click()  
' 首先定义变量。其中R1是一个QueryDef类型的变量，用来实现SQL查询操作；sql和s(5)
```



用来存放临时字符串

```

Dim R1 As QueryDef
Dim sql As String
Dim s(5) As String
Dim i, j As Integer
' 设置sql的初始值为"select * from 资产信息 where "，从“资产信息”表中进行查询。
' 资产信息.资产名称, 资产信息.使用期限
sql = "select * from 资产信息 where "
' 新建查询“资产信息查询”。在数据库管理器中选择“查询”标签，然后单击“新建”按钮，利用“资产信息”表的所有字段创建查询。现在就可以将当前的数据查询实例定义为“资产信息查询”。
Set R1 = CurrentDb.QueryDefs("资产信息查询")
' 根据文本框的内容来实现SQL查询语句。
If IsNull(comID) = False Then
    s(0) = "资产信息.资产编号 = "" + comID + "" "
End If
If IsNull(comType) = False Then
    s(1) = "资产信息.资产类型 = "" + comType + "" "
End If
If IsNull(comName) = False Then
    s(2) = "资产信息.资产名称 = "" + comName + "" "
End If
If IsNull(comModel) = False Then
    s(3) = "资产信息.型号 = "" + comModel + "" "
End If
If IsNull(comPos) = False Then
    s(4) = "资产信息.存放位置 = "" + comPos + "" "
End If
' 设置SQL查询语句为select * from * where * + * + ... + *的形式
For i = 0 To 4
    If s(i) <> "" Then
        For j = i + 1 To 4
            If s(j) <> "" Then
                s(j) = "and " + s(j)
            End If
        Next j
        Exit For
    End If
Next i

For i = 0 To 4
    sql = sql + s(i)
Next i
R1.sql = sql

```



' 打开 " 资产信息 " 窗体，并且重新设置查询 R1 的数据集。

```
DoCmd.OpenForm " 资产信息 "
```

```
Form_ 资产信息. Form. Recordset. Close
```

```
Set Form_ 资产信息. Form. Recordset = R1.OpenRecordset
```

```
End Sub
```



## 操作任务模块设计

### 1. 设计借出信息窗体

借出信息窗体的作用是以表格的形式显示所有资产的借出信息。表 22-9 所示为借出信息窗体中控件的主要属性。



表 22-9 借出信息窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号
标签1	类型	标签
	标题	借用人
标签2	类型	标签
	标题	出借人
标签3	类型	标签
	标题	批复人
标签4	类型	标签
	标题	出借日期
标签5	类型	标签
	标题	拟还日期
标签6	类型	标签
	标题	借用理由
借出信息	类型	窗体
	记录源	借出信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否
	分隔线	否
	自动调整	否
自动居中	是	
文本0	类型	文本框
	标题	资产编号
文本1	类型	文本框
	标题	借用人



(续表)

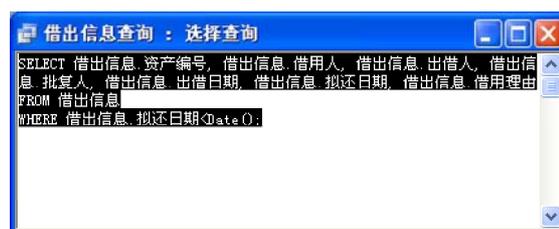
控件名称	控件属性名称	控件属性值
文本2	类型	文本框
	标题	出借人
文本3	类型	文本框
	标题	批复人
文本4	类型	文本框
	标题	出借日期
文本5	类型	文本框
	标题	拟还日期
文本6	类型	文本框
	标题	借用理由

## 2. 设计到期信息窗体

到期信息窗体的作用是以表格的形式显示所有已经借出但是到期未还的资产的信息。下面开始创建该窗体。

### (1) 设计借出信息查询

首先要创建借出信息查询，得到所有已经借出但是到期未还的资产的信息。按照如下方法创建这个查询。首先在数据库管理器中选择“查询”选项，再单击“新建”按钮。然后直接进入设计视图。在弹出的窗口中单击鼠标右键，在弹出的快捷菜单中选择“SQL 视图”命令，在其中输入如右图所示的 SQL 语句。



### (2) 设计窗体

到期信息窗体的作用是以表格的形式显示所有资产的到期信息。表 22-10 所示为到期信息窗体中控件的主要属性。



表 22-10 到期信息窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号
标签1	类型	标签
	标题	借用人
标签2	类型	标签
	标题	出借人
标签3	类型	标签
	标题	批复人



(续表)

控件名称	控件属性名称	控件属性值
标签4	类型	标签
	标题	出借日期
标签5	类型	标签
	标题	拟还日期
标签6	类型	标签
	标题	借用理由
到期信息	类型	窗体
	记录源	借出信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否
	分隔线	否
	自动调整	否
自动居中	是	
文本0	类型	文本框
	标题	资产编号
文本1	类型	文本框
	标题	借用人
文本2	类型	文本框
	标题	出借人
文本3	类型	文本框
	标题	批复人
文本4	类型	文本框
	标题	出借日期
文本5	类型	文本框
	标题	拟还日期
文本6	类型	文本框
	标题	借用理由

### 3. 设计出借信息窗体

出借信息窗体的作用是以表格的形式显示所有资产的出借信息。表 22-11 所示为出借信息窗体中控件的主要属性。





表 22-11 出借资产窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	资产编号
标签1	类型	标签
	标题	借用人
标签2	类型	标签
	标题	出借人
标签3	类型	标签
	标题	批复人
标签4	类型	标签
	标题	出借日期
标签5	类型	标签
	标题	拟还日期
标签6	类型	标签
	标题	借用理由
出借资产	类型	窗体
	记录源	借出信息
	滚动条	两者均无
	记录选择器	否
	浏览按钮	否
	分隔线	否
	自动调整	否
自动居中	是	

下面详细说明“借出”按钮的单击事件的处理函数。用鼠标右键单击该按钮，在弹出的快捷菜单中选择“事件生成器”命令，在出现的VBA代码编辑器中输入下面的代码。

```
Private Sub ok_Click()
' 定义查询变量和临时字符串变量
    Dim R1, R2 As QueryDef
    Dim sql As String
' 如果"确定"按钮的标题是"借出",那么应该完成借出一项资产的功能,方法是执行一条SQL插入语句
    If ok.Caption = "借出" Then
        sql = "INSERT INTO 借出信息([资产编号], [借用人], [出借人], [批复人], [出借日期], [拟还日期], [借用理由]) " + _
            "VALUES( "" + txtId + """" _
            + ", "" + txtUseMan + """" _
            + ", "" + txtOutMan + """" _
            + ", "" + txtConfirm + """" _
            + ", "" + Str(txtOutDate) + """" _
            + ", "" + Str(txtInDate) + """" _
            + ", "" + txtRes + """" _
            + ") "
```



' 现在需要新建一个"出借查询",方法是直接根据"借出信息"表新建一个查询。当"出借查询"建立好之后,就可以通过这个查询执行SQL语句了

```
Set R1 = CurrentDb.QueryDefs("出借查询")
```

```
R1.sql = sql
```

```
R1.Execute
```

' 通过执行SQL更新语句更新资产状态

```
Set R2 = CurrentDb.QueryDefs("出借资产状态更新")
```

```
R2.sql = "UPDATE 资产信息 SET 资产状态="" 已借出 "" , 可行操作="" 还入 ""  
WHERE 资产编号="" + """" + txtId + """"
```

```
R2.Execute
```

```
Else
```

' 如果"确定"按钮的标题不是"借出",那么应该完成归还一项资产的功能,方法是执行一条SQL删除语句。这里首先还需要创建"归还查询",方法是直接根据"借出信息"表新建一个查询。当"归还查询"建立好之后,就可以通过这个查询执行SQL语句了

```
sql = "delete from 借出信息 where 资产编号="" + """" + txtId + """"
```

```
Set R1 = CurrentDb.QueryDefs("归还查询")
```

```
R1.sql = sql
```

```
R1.Execute
```

' 通过执行SQL更新语句更新资产状态

```
Set R2 = CurrentDb.QueryDefs("出借资产状态更新")
```

```
R2.sql = "UPDATE 资产信息 SET 资产状态="" 在库中 "" , 可行操作="" 借出 ""  
WHERE 资产编号="" + """" + txtId + """"
```

```
R2.Execute
```

```
End If
```

```
End Sub
```

## 分析任务模块设计

### 1. 设计修改密码窗体

修改密码窗体的功能是得到用户名和原密码,并在“系统用户”数据表中核对用户名和原密码是否匹配,然后更改密码。表 22-12 所示为修改密码窗体中控件的主要属性。





表 22-12 修改密码窗体控件属性

控件名称	控件属性名称	控件属性值
标签0	类型	标签
	标题	用户名:
标签1	类型	标签
	标题	原密码:
标签2	类型	标签
	标题	新密码:
com用户	类型	组合框
	行来源类型	表/查询
	行来源	SELECT [系统用户].[用户名], [系统用户].[用户名]. FROM. 系统用户;
Old_pass	类型	文本框
	输入掩码	密码
New_pass	类型	文本框
	输入掩码	密码

下面详细说明“确定”按钮的单击事件的处理函数。用鼠标右键单击该按钮，在弹出的快捷菜单中选择“事件生成器”命令，在出现的VBA代码编辑器中输入下面的代码。

```
Private Sub 命令6_Click()
' 首先定义一个查询变量和一个临时字符串变量
    Dim r As QueryDef
    Dim s As String
' 利用DLookup函数找到与当前用户名相应的密码
    s = DLookup("[密码]", "系统用户", "[用户名]= "" & userName & """)
' 判断输入密码的正确性。
    If old_pass = s Then
' 利用SQL更新语句修改密码。这里与前面类似的要先建立一个查询“修改密码”。
        Set r = CurrentDb.QueryDefs("修改密码")
        r.sql = "UPDATE 系统用户 SET 密码 = "" + new_pass + "" where 用户  
名 = "" + userName + """
        r.Execute
        MsgBox "已经成功修改了密码！"
    Else
        MsgBox "密码错误！"
    End If
End Sub
```

## 2. 设计打印窗体

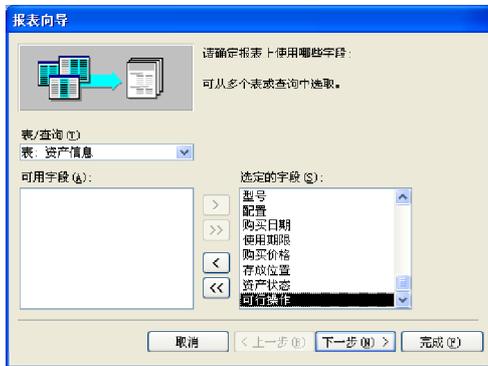
### (1) 设计报表

打印窗体的功能是进行打印和打印预览。首先需要创建4个报表，具体操作步骤如下。

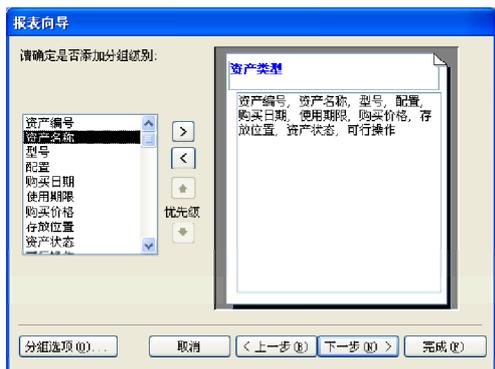
- 1 双击打开“资产管理系统”数据库，在“对象”列表框中选择“报表”选项，并双击“使用向导创建报表”选项。



- 2 在弹出的“报表向导”对话框中选择“资产信息”表的所有字段。单击“下一步”按钮，进入下一界面。



- 3 在该界面中要求选择一个或多个字段，作为分组的依据，这里选择“资产类型”字段。单击“下一步”按钮进入下一界面。



- 4 在该界面中要求为记录排序选择字段，这里选择按“资产编号”进行升序排列。单击“汇总选项”按钮，弹出“汇总选项”对话框，要求选择汇总的方式。



- 5 勾选“使用期限”和“购买价格”这两个字段的“汇总”复选框，表示希望在报表上显示“使用期限”和“购买价格”的总计统计。



- 6 单击“确定”按钮，并单击“下一步”按钮，进入下一界面。这里确定报表的版式，采用“递阶”式布局和“横向”方向。





- 7 单击“下一步”按钮，进入下一界面，这里确定报表的样式，选择“组织”型的样式。



- 8 单击“下一步”按钮，进入下一界面，这里确定报表的标题，将标题设定为“资产信息”，然后单击“完成”按钮即可完成整个报表的设计过程。



- 9 打开报表设计视图，可以在其中修改字段的名称，并将字段的位置调整至最佳。



- 10 按照与上面相同的方法创建借出信息报表、资产信息查询报表以及借出到期信息报表，这里不再赘述。

(2) 设计打印报表窗体

右图所示为“打印报表”窗体的设计视图。



下面为 4 个按钮控件添加事件处理。  
首先加入“资产信息报表”按钮，用户单击这个按钮触发的事件是打开“资产信息报表”。



```
Private Sub 资产信息报表_Click()
    On Error GoTo Err_preview_items_Click

    Dim stDocName As String

    stDocName = "资产信息"
    DoCmd.OpenReport stDocName, acPreview

Exit_preview_items_Click:
    Exit Sub

Err_preview_items_Click:
    MsgBox Err.Description
    Resume Exit_preview_items_Click

End Sub
```

现在加入“借出信息报表”按钮，用户单击这个按钮触发的事件是打开“借出信息报表”。



```
Private Sub 借出到期报表_Click()
    Dim stDocName As String

    stDocName = "借出到期信息"
    DoCmd.OpenReport stDocName, acPreview

End Sub
```

现在加入“资产信息查询报表”按钮，用户单击这个按钮触发的事件是打开“资产信息查询报表”。



```
Private Sub 资产信息查询结果_Click()
    Dim stDocName As String

    stDocName = "资产信息查询"
    DoCmd.OpenReport stDocName, acPreview

End Sub
```