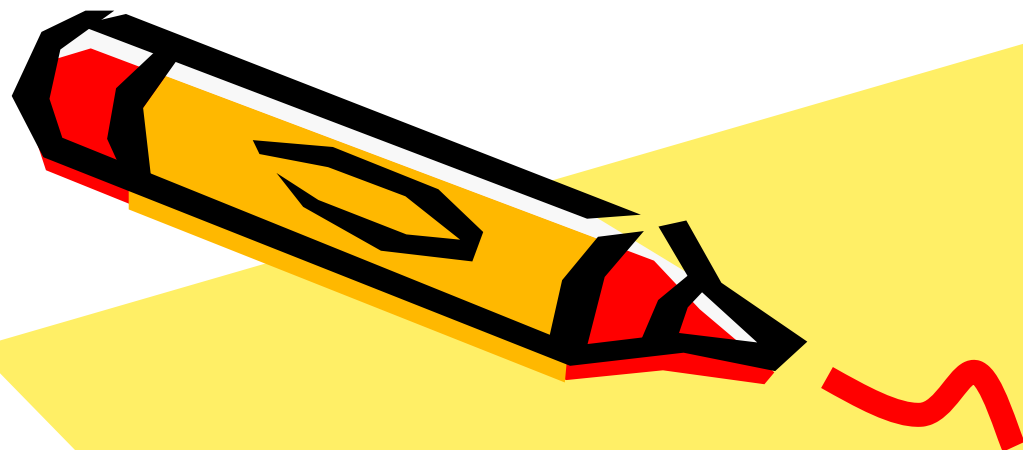
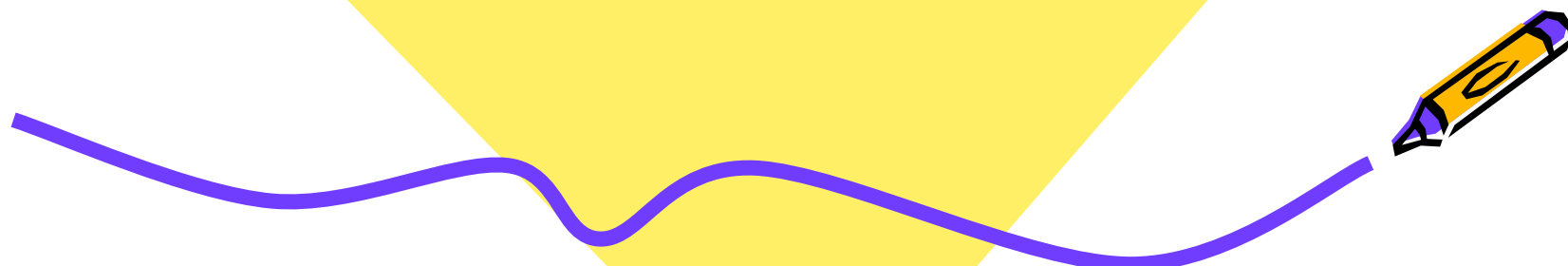




Access实例教程

潘明寒 赵义霞 主编

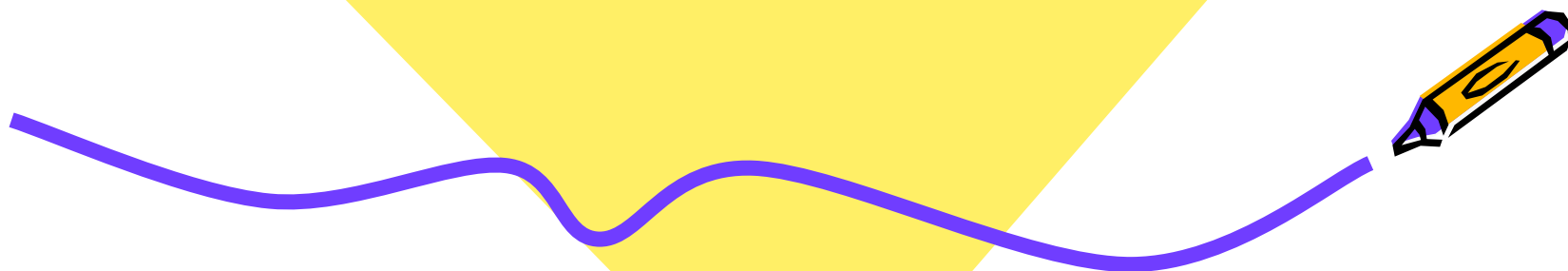


中国水利水电出版社



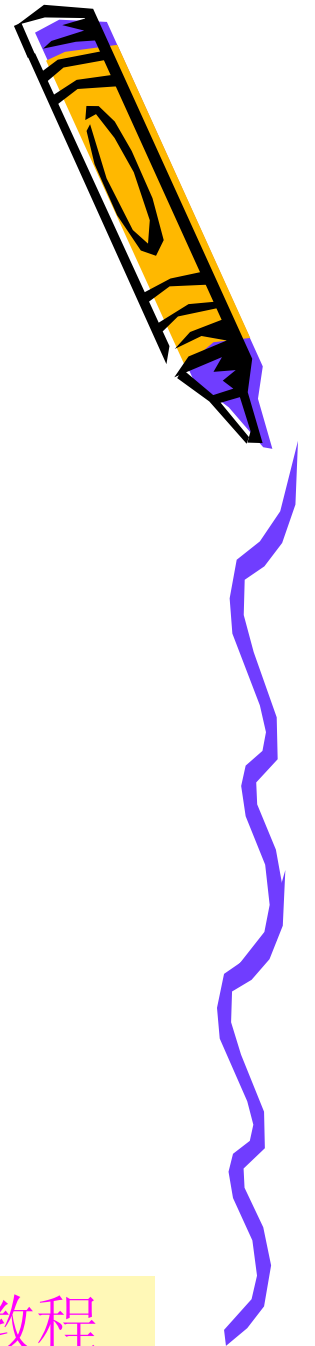
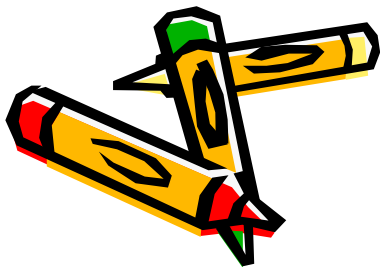
ACCESS实例教程

——ACCESS概述（第1章）



本章要点:

- 1, 启动并退出 Access 2003
- 2, 认识 Access 2003 工作窗口
- 3, 了解 Access 的 7 种数据库对象
- 4, 创建 Access 数据库
- 5, 数据库的基本概念



1.1 Access简介

Access 是一种关系型的桌面数据库管理系统，作为Microsoft Office软件的组件之一，在世界范围内得到广泛使用。Access具有功能强大和操作简单的特点，特别适合数据库技术的初学者。

以下将以Access 2003为教学背景介绍Access的使用方法。



ACCESS实例教程

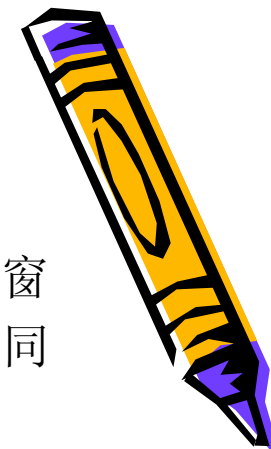
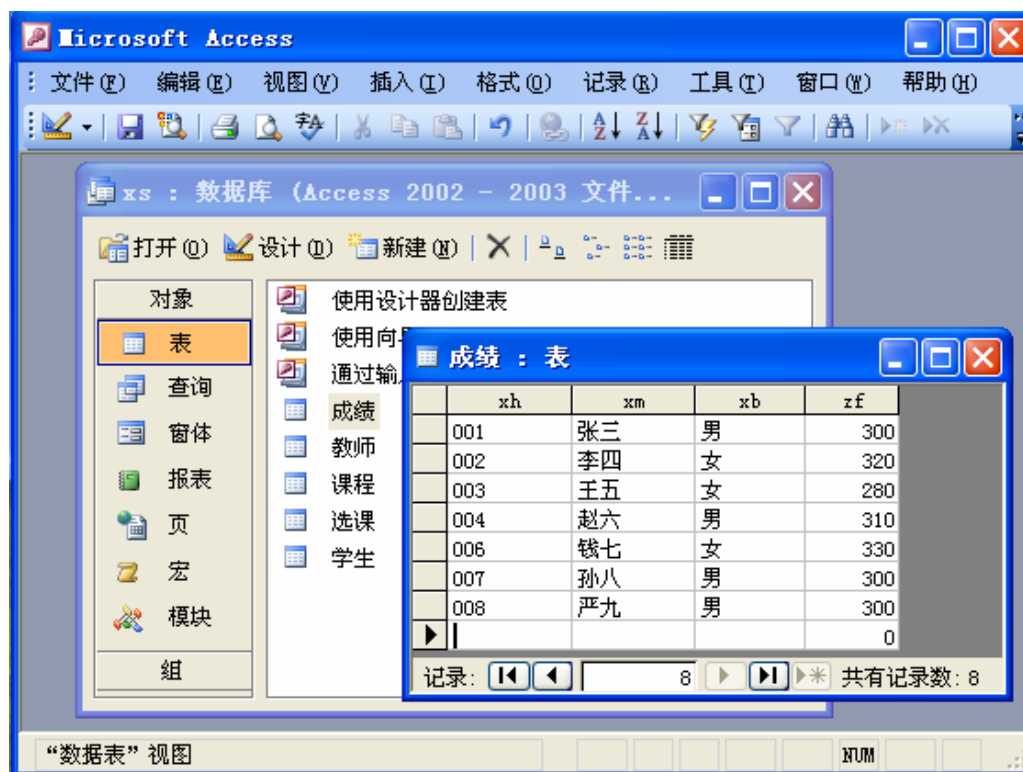
1.1.1 Access的特点

1. 有强大的开发工具VBA，可以编写数据库应用程序。
2. 能访问多种格式的数据，如：Excel数据表和text文本文件。
3. 支持ODBC标准的SQL数据库的数据。
4. 方便的向导功能使设计过程自动化。
5. 可以使用Internet功能发布信息。
6. 采用OLE技术支持对象的嵌入与链接。
7. 具有较强的安全性。



1.1.2 Access的工作窗口

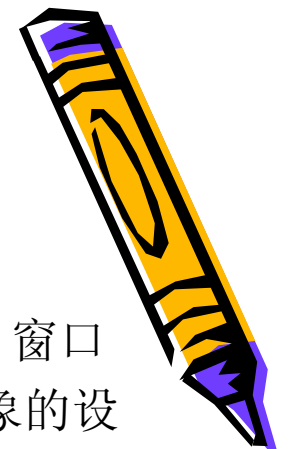
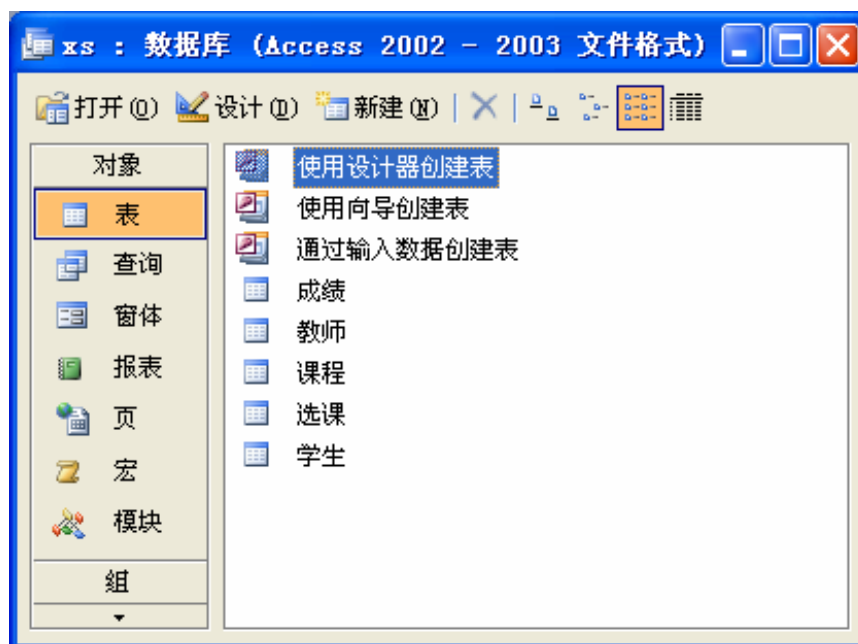
Access 2003的工作窗口中包含一个或多个Access对象的视图窗口或设计窗口，除了数据库窗口一次只能打开一个，其他对象可以同时打开多个。



ACCESS实例教程

1.1.3 Access 2003的数据库窗口

数据库窗口是数据库的设计视图，也是其他对象窗口的基础。窗口左边显示数据库的7个对象，选取一个对象类别，然后进行该类对象的设计与编辑。窗口右边是某类对象的成员列表。

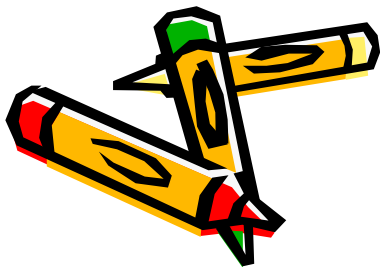
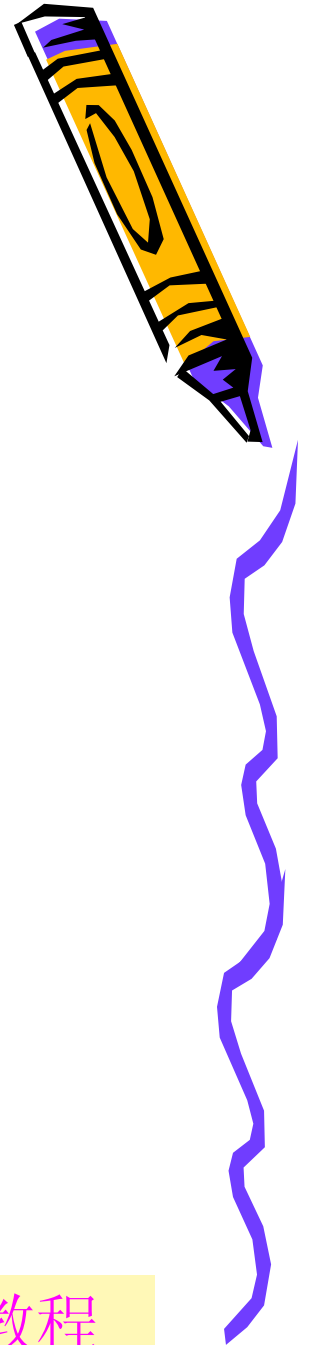


ACCESS实例教程

1.2 关于数据库的基本操作

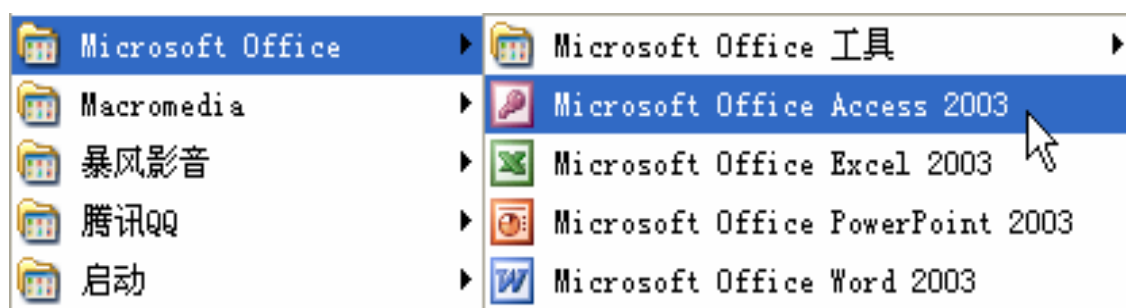
1.2.1 Access数据库文件

Access数据库文件的扩展名为“mdb”，除了页对象之外的其他数据库对象都是该数据库文件中的一项内容，不能作为单独的文件保存在磁盘上。如果关闭数据库窗口，所有基于该数据库的对象窗口都将被关闭。



1.2.2 启动Access 2003

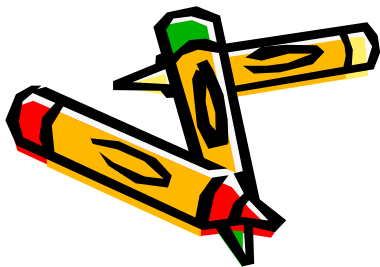
access是office的组件。



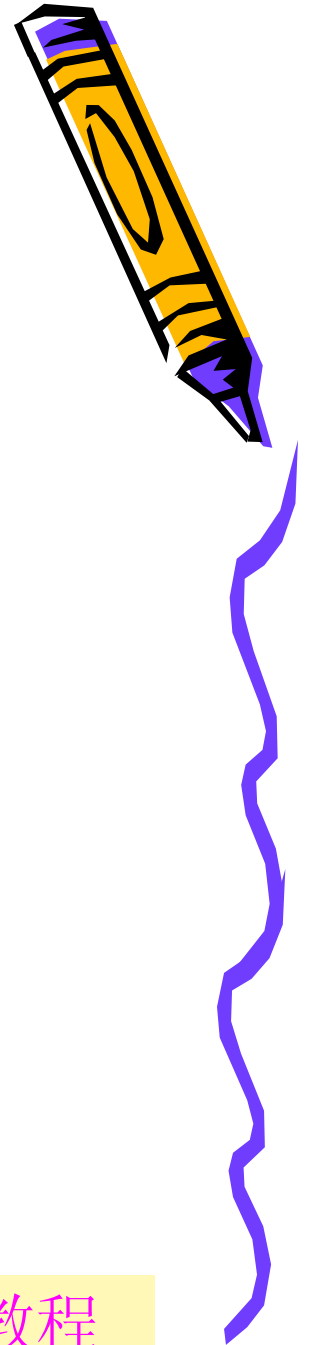
ACCESS实例教程

1.2.3 打开数据库与创建数据库

1. 单击“新建文件”选项将创建一个空的数据库。
2. 双击一个扩展名为mdb的Access数据库文件，可启动Access并打开该数据库。



ACCESS实例教程



1.2.4 关闭数据库与退出Access

单击数据库窗口标题栏的关闭图标，或使用“文件”菜单下的“关闭”选项，只关闭数据库，不退出Access。

退出Access 2003可以用如下4种方法。

- 1，单击Access 2003工作窗口标题栏右端的关闭按钮。
- 2，打开Access 2003工作窗口的“文件”菜单→选“退出”项。
- 3，用Alt+F+X组合键。（先按下Alt键不松手，敲F键再敲X键。）
- 4，用Alt+F4组合键。（先按下Alt键不松手，敲F4键。）



1.3 ACCESS的7种数据库对象

Access有7种数据库对象，分别是：表、查询、窗体、报表、页、宏、模块。

1.3.1 表

表由字段和记录组成。

字段是表中的列，每个字段代表一条信息在某一方面的属性，字段有类型，如“姓名”字段是字符型的，“年龄”字段是数字型的。字段的基本属性有：字段名称、数据类型、字段大小、默认值等。

记录是数据表中的行，由一个或多个字段的值组成，一条记录是一条完整的信息，显示一个对象的所有属性。如：001、张三、男、21，可以作为一条记录。



表是整个数据库系统的基础。一个数据库中 can 包含多个数据表，一个表应围绕一个主题建立，如学籍表、成绩表。表之间可以建关系，建立了关系的多个表可以像一个表一样使用。

成绩：表				
	学号	姓名	性别	总分
▶	001	张三	男	300
	002	李四	女	320
	003	王五	女	280
	004	赵六	男	310
	006	钱七	女	330
	007	孙八	男	300
	008	严九	男	300
米				0



1.3.2 查询

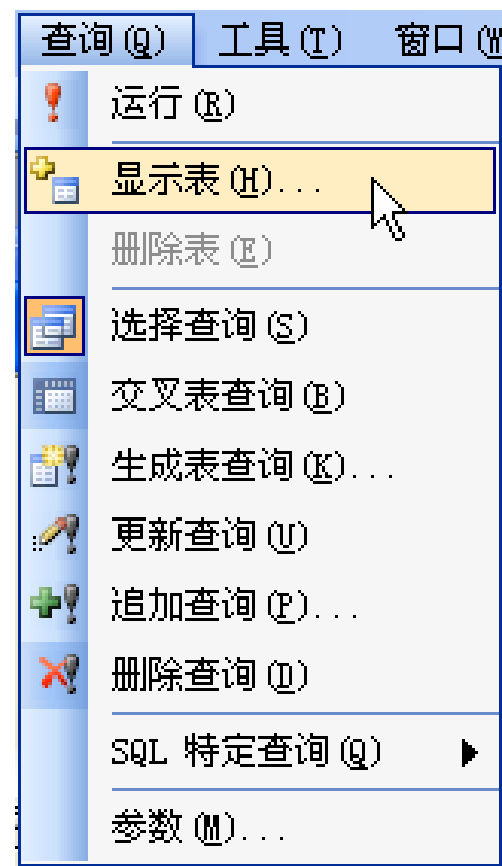
查询是数据库的核心操作，根据指定条件从数据表或其他查询中筛选出符合条件的记录。查询结果以二维表的形式显示，是动态数据集合，每执行一次查询操作都会显示数据源中最新数据。

300分以上的人：选择查询				
	学号	姓名	性别	总分
	002	李四	女	320
	004	赵六	男	310
	006	钱七	女	330
▶				0



查询类型有：

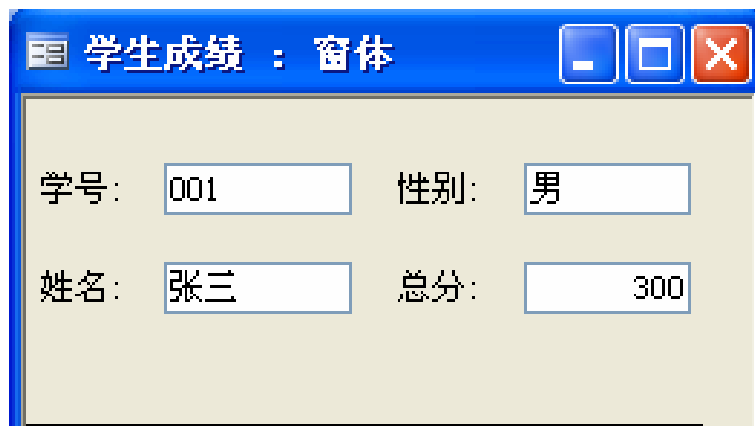
- 1, 选择查询
- 2, 交叉表查询
- 3, 生成表查询
- 4, 更新查询
- 5, 追加查询
- 6, 删除查询
- 7, SQL查询
- 8, 参数查询



ACCESS实例教程

1.3.3 窗体

窗体用来显示和修改表，是用户与Access应用程序之间的主要接口，可以简化数据库的操作。窗体的数据源来自表或查询，利用窗体将整个应用程序组织起来，形成一个完整的应用系统。

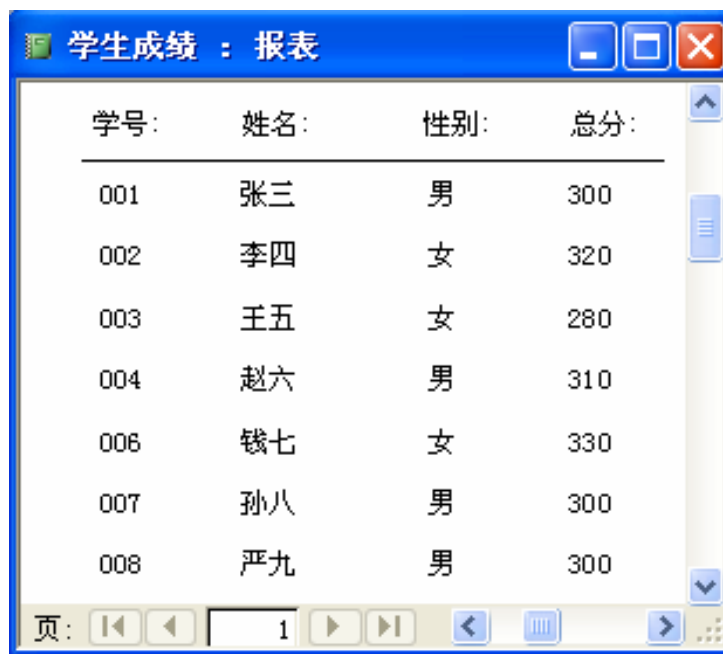


学号	性别	姓名	总分
001	男	张三	300



1.3.4 报表

报表用来以格式化方式显示并打印数据。利用报表可以整理和计算基本表中的数据，有选择的显示指定信息。报表的数据源来自表、查询或SQL语句，在报表中不能输入数据。

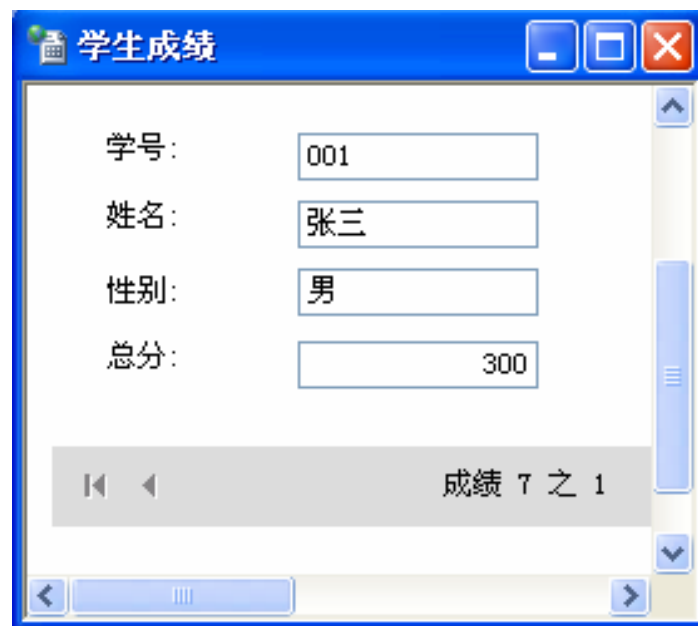


学号:	姓名:	性别:	总分:
001	张三	男	300
002	李四	女	320
003	王五	女	280
004	赵六	男	310
006	钱七	女	330
007	孙八	男	300
008	严九	男	300



1.3.5 页

页又称为数据访问页，是一种特殊的Web页，为通过网络发布数据提供方便。数据访问页直接与数据库相连，用户通过数据访问页查看和编辑Access数据库中的数据。对页中数据进行修改、添加或删除操作，结果会保存在数据库中。



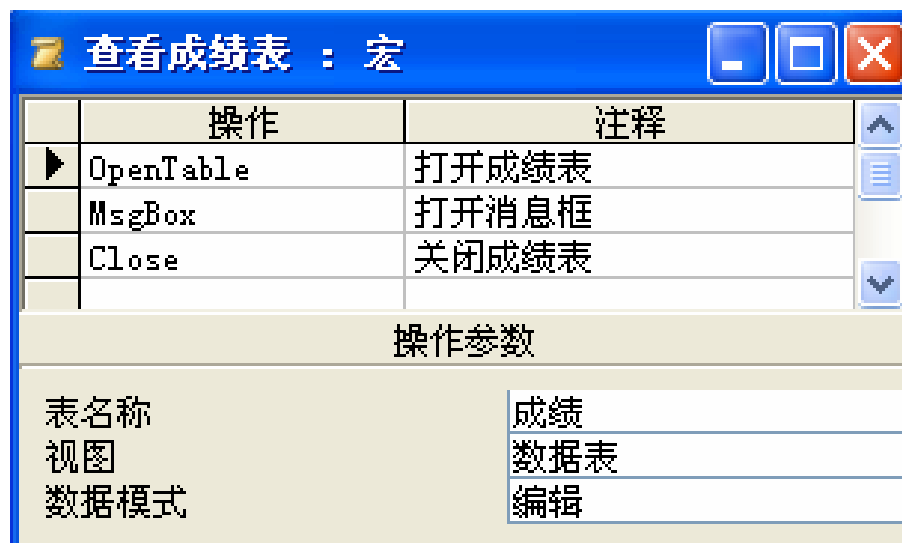
学号:	001
姓名:	张三
性别:	男
总分:	300

成绩 7 之 1



1.3.6 宏

宏是一系列操作的集合，如打开窗体、打印报表，完成大量的重复性工作，快速实现数据库的简单操作。可以单击某个命令按钮时运行该宏。



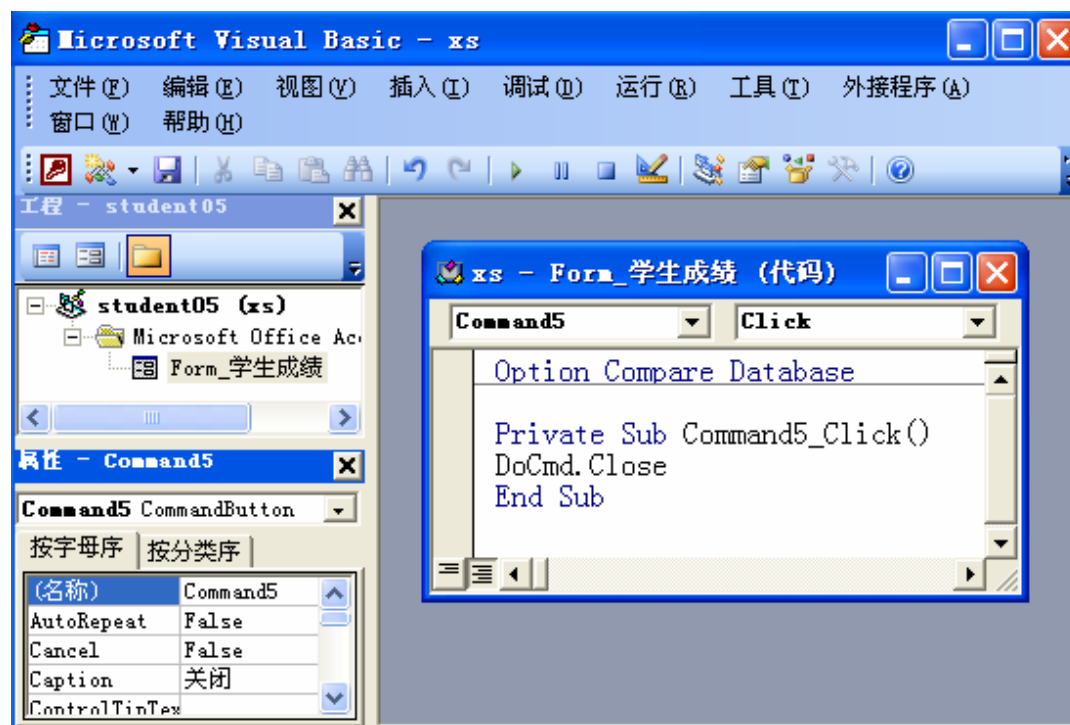
操作	注释
OpenTable	打开成绩表
MsgBox	打开消息框
Close	关闭成绩表

操作参数	
表名称	成绩
视图	数据表
数据模式	编辑



1.3.7 模块

模块是VBA（Visual Basic for Applications）程序的集合，实现数据库较为复杂操作。模块将声明和过程作为一个单元保存，完成宏不能完成的任务。模块有两个基本类型：类模块和标准模块。类模块与某个窗体或报表相关联，标准模块存放供其他Access数据库对象使用的公共过程。



ACCESS实例教程

1.4 数据库基本知识

1.4.1 基本概念

1. 数据

从计算机角度看，数据是能被计算机识别、存储和加工的信息载体。例如：“张三很高”是信息，“张三的身高为2米”是数据。在计算机中，文字、图形、图像、声音、动画、影像等都是数据。

2. 数据库

数据库（DataBase，DB）是存储在计算机存储设备中的、结构化的、可共享的数据集合。数据库中的数据面向多种应用，可以被多个用户或多个应用程序共享。

3. 数据库应用系统

数据库应用系统是用数据库系统开发的面向某类实际应用的计算机应用软件。如：学生管理系统、图书管理系统。



4. 数据库管理系统

数据库管理系统（DataBase Management System, DBMS）是用来建立、使用、维护数据库的数据管理软件，位于用户与操作系统之间，属于计算机系统软件的范畴。

5. 数据库系统

数据库系统（DataBase System, DBS）是引入数据库技术后的计算机系统，包括：硬件系统、数据库集合、数据库管理系统及相关软件、数据库管理员、用户。

6. 数据库、数据库管理系统、数据库系统三者的关系

数据库系统包括数据库和数据库管理系统。



1.4.2 数据模型

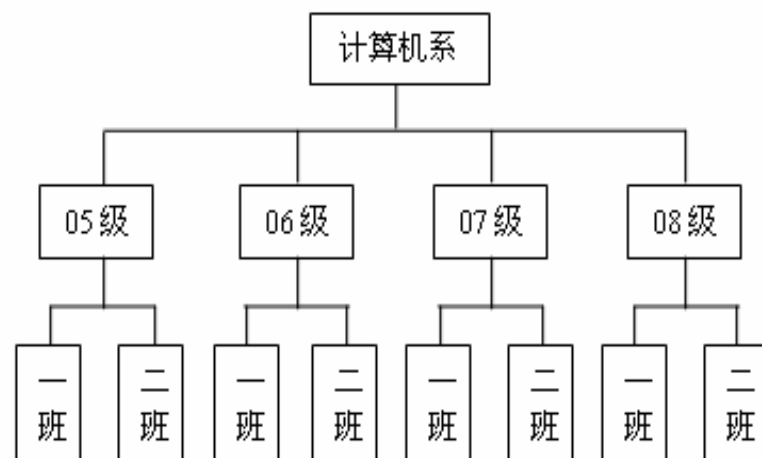
数据模型有3种：层次模型、网状模型、关系模型。

Access数据库管理系统所使用的数据模型是关系数据模型。

1. 层次模型

层次模型是树型结构, 自顶向下, 层次分明。层次模型要满足以下两个条件:

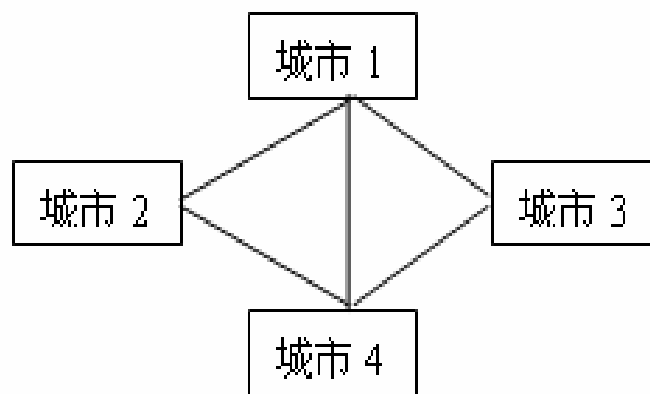
- 有且只有一个根节点, 根节点没有双亲。
- 其他节点有且只有一个双亲。



2，网状模型

网状模型是用无向图结构，是一种交叉关系，是关系模型的扩展。网状模型要满足以下两个条件：

- 允许一个以上的节点没有双亲。
- 一个节点有多于一个的双亲。



3, 关系模型

关系模型是用二维表结构，在关系模型中，操作对象和操作结果都是二维表。关系模型是目前最重要的数据模型，被几乎所有数据库管理系统支持。

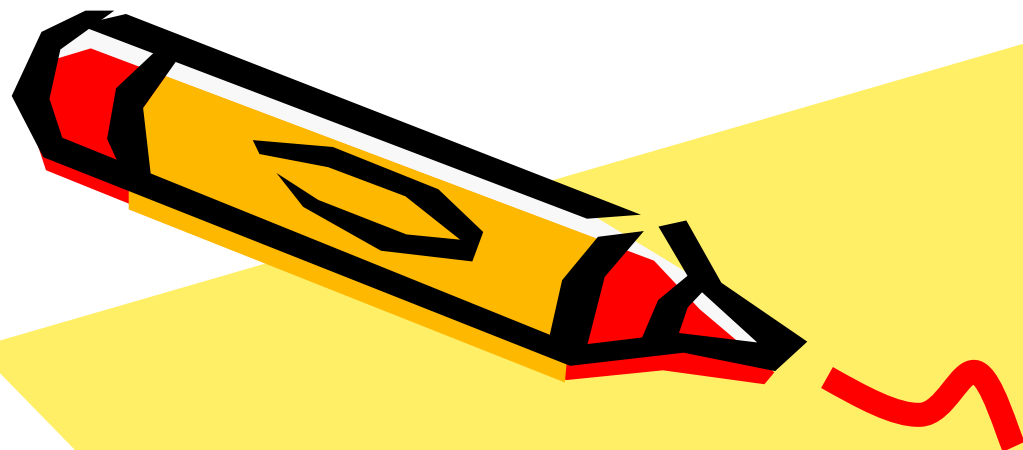
学号	姓名	性别	总分
001	张三	男	300
002	李四	女	320
003	王五	女	280
004	赵六	男	310
006	钱七	女	330
007	孙八	男	300
008	严九	男	300



关系模型要满足以下几个性质：

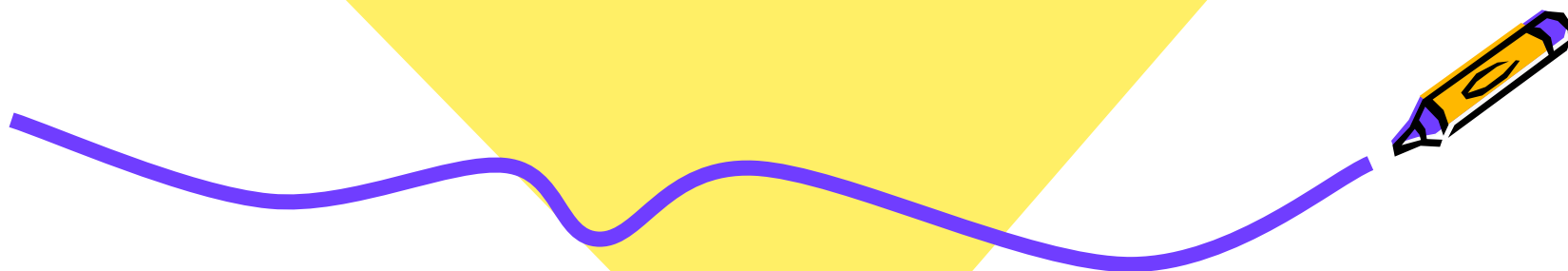
- 元组个数有限性：元组的个数是有限的。
- 元组惟一性：每个元组都是惟一的。
- 元组次序无关性：元组的次序可以任意交换。
- 元组分量的原子性：元组的分量是不可分割的基本数据项。
- 属性名惟一性：属性的名称各不相同。
- 属性次序无关性：属性的次序可以任意交换。
- 属性分量值域同一性：属性的分量与属性值域相同。





ACCESS实例教程

——数据表的操作（第2章）



本章要点:

- 1, 建立表结构
- 2, 向表中输入数据
- 3, 建立表对象之间的关联
- 4, 操作表中记录
- 5, 设置数据表格式



2.1 建立表结构

2.1.1 表的设计视图

数据表由表结构和表记录两部分组成，建立表结构在表的设计视图中完成，表的全部字段和每个字段的属性在设计视图中确定。

字段名称	数据类型	说明
姓名	文本	
性别	文本	
▶ 年龄	数字	
班级	文本	
出生年月	日期/时间	

字段属性	
常规	查阅
字段大小	字节
格式	
小数位数	自动
输入掩码	
标题	年龄
默认值	20
有效性规则	>=15 And <=35
有效性文本	否
必填字段	无
索引	
智能标记	



2.1.2 字段的命名规则

一个表要围绕一个主题设计字段，每个字段都应该是最小的逻辑部分，计算字段或推导字段不要作为表中的字段。

字段的命名规则如下：

1. 字段名可以包含字母、汉字、数字、空格和其他字符，第一个字符不能是空格。
2. 字段名不能包含小数点、叹号、方括号、西文单引号、西文双引号。
3. 字段长度为1~64个字符，在Access中一个汉字当作一个字符看待。



2.1.3 字段大小

字段大小用来定义字段所使用的存储空间大小，是字段值所占的字节数。只有文本型字段和数字型字段需要指定字段大小，其他类型的字段由系统分配字段大小，例如，“出生日期”是日期/时间类型，字段大小为8，“婚否”是逻辑类型，字段大小为1。一个字符和一个汉字字段大小都是1。

2.1.4 字段的数据类型

数据类型决定用户能保存在该字段中值的种类。Access字段的数据类型有10种，分别是：文本、备注、数字、日期/时间、货币、自动编号、是/否、OLE对象、超链接、查阅向导。





1. 文本型

文本型字段用来存放文本或作为文本看待的数字。如：学号、姓名、性别等字段。如果设置字段大小为5，则该字段的值最多只能容纳5个字符。

文本型字段的默认大小为50，最多可达255个字符。

文本型数字的排序按照字符串排序方法进行。

如文本型数字按升序排序：1、10、100、2、20、200

2. 备注型

备注型字段用来存放较长的文本和文本型数字。如：备忘录、简历等字段都是备注型。当字段中存放的字符个数超过255时，应该定义该字段为备注型。

备注型字段大小是不定的，由系统自动调整，最多可达64K。Access不能对备注型字段进行排序、索引、分组。



3. 数字型

数字型字段存放数字。如：工资、年龄等，数字型字段可以与货币型字段做算术运算。数字型字段的大小由数字类型决定，常用数字类型有以下几种：

- (1) 字节，存放0~255之间的整数，字段大小为1。
- (2) 整型，存放-32768~32767之间的整数，字段大小为2。
- (3) 长整型，存放-2147483648~2147483647之间的整数，字段大小为4。
- (4) 单精度型，存放-3.4E38~3.4E38之间的实数，字段大小为4。
- (5) 双精度型，存放-1.79734E308~1.79734E308之间的实数，字段大小为8。



4. 日期/时间型

日期/时间型字段存放日期、时间、或日期时间的组合。

如：出生日期、入校日期等字段都是日期/时间型字段。字段大小为8个字节，由系统自动设置。

日期/时间型的常量要用一对#号括起来。

5. 货币型

货币型字段存放具有双精度属性的数字。系统自动将货币字段的数据精确到小数点前15位及小数点后4位。字段大小为8，由系统自动设置。

向货币型字段输入数据时，系统会自动给数据添加2位小数，并显示美元符号与千位分隔符。



6. 自动编号型

自动编号型字段存放系统为记录绑定的顺序号，长整型，字段大小为4，由系统自动设置。一个表只能有一个自动编号型字段，该字段中的顺序号永久与记录相联，不能人工指定或更改自动编号型字段中的数值。删除表中含有自动编号字段的记录以后，系统将不再使用已被删除的自动编号字段中的数值。

例如，输入10条记录，自动编号从1到10，删除前3条记录，自动编号从4到10，删除第7条记录，自动编号中永远设有7。

与财务、税务有关的数据表通常设自动编号型字段，增加数据的安全性。



7. 是/否型

是/否型字段存放逻辑数据，字段大小为1，由系统自动设置。逻辑数据只能有2种不同的取值。如：婚否、团员否。所以，是/否型数据又被称为“布尔”型数据。

是/否型字段内容通过画“√”输入，带“√”的为“真”，不带“√”的为“假”，“真”值用true或on或yes表示，“假”值用false或off或no表示。



8. OLE对象型

OLE (Object Linking and Embedding) 的中文含义是“对象的链接与嵌入”，用来链接或嵌入OLE对象，如：文字、声音、图像、表格等。表中的照片字段应设为OLE对象类型。

OLE对象型字段的字段大小不定，最多可达到1GB。OLE对象只能在窗体或报表中用控件显示。不能对OLE对象型字段进行排序、索引或分组。

9. 超链接型

超链接型字段存放超链接地址，如：网址、电子邮件。超链接型字段大小不定。



10. 查阅向导型

查阅向导型字段仍然显示为文本型，所不同的是该字段保存一个值列表，输入数据时从一个下拉式值列表中选择。值列表的内容可以来自表或查询，也可以来自定义的一组固定不变的值。例如，将“性别”字段设为查阅向导型以后，只要在“男”和“女”2个值中选择一个即可。

查阅向导型字段大小不定。



2.1.5 设置字段属性

字段属性是字段特征值的集合，分为常规属性和查阅属性2种，用来控制字段的操作方式和显示方式。

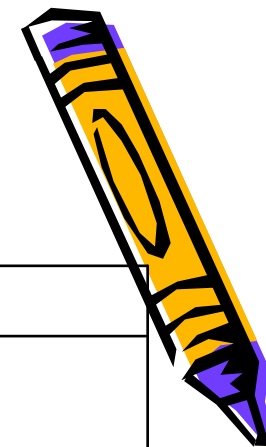
不同字段类型有不同的属性集合。

常规	查阅
字段大小	50
格式	
输入掩码	
标题	性别
默认值	"男"
有效性规则	"男" Or "女"
有效性文本	只能输入"男"或"女"
必填字段	是
允许空字符串	否
索引	无
Unicode 压缩	是
输入法模式	开启
IME 语句模式 (仅日文)	无转化
智能标记	

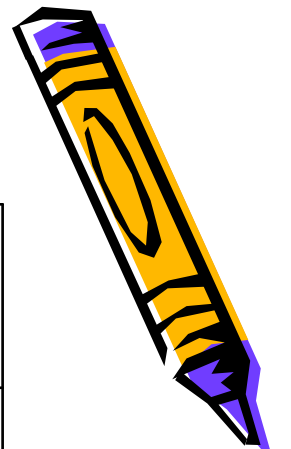


1, 输入掩码：字段的输入模板

字符	字符含义
0	在掩码字符位置必须输入数字。 例如，掩码：(00)00-000，示例：(12)55-234
9	在掩码字符位置输入数字或空格，保存数据时保留空格位置。 例如，掩码：(99)99-999，示例：(12)55-234，()55-234
#	在掩码字符位置输入数字、空格、加号或减号。 例如，掩码：####，示例：1+，9+999
L	在掩码字符位置必须输入英文字母，大小写均可。 例如，掩码：LLLL，示例：aaaa，AaAa
?	在掩码字符位置输入英文字母或空格，字母大小写均可。 例如，掩码：????，示例：a a，Aa
A	在掩码字符位置必须输入英文字母或数字，字母大小写均可。 例如，掩码：(00)AA-A，示例：(12)55-a，(80)AB-4



a	在掩码字符位置输入英文字母、数字或空格，字母大小写均可。 例如，掩码：aaaa，示例：5a5b，A 4
&	在掩码字符位置必须输入空格或任意字符。 例如，掩码：&&&&，示例：\$5A%
C	在掩码字符位置输入空格或任意字符。 例如，掩码：CCCC，示例：\$5A%
. , : ; - /	句点、逗号、冒号、分号、减号、正斜线，用来设置小数点、千位、日期时间分隔符。
<	将其后所有字母转换为小写。 例如，掩码：LL<LL，输入AAAA，显示AAaa
>	将其后所有字母转换为大写。 例如，掩码：LL>LL，输入aaaa，显示aaAA
密码	以*号显示输入的字符。



ACCESS实例教程

2. 有效性规则

有效性规则是一个条件，用来为字段的值定义数据范围和数值要求。如果输入的数据不符合有效性规则，将给出提示信息，并且光标停在原处，直到输入正确数据为止。

注意，有效性规则的设置不能与默认值冲突。

例如：性别字段只能输入文字男或女，

用：“男” or “女”。

例如：年龄字段的范围是大于0，

用：>0。

例如：出生日期的字段范围是2001年，

用：>=#2001-1-1# and <=#2001-12-31#。



2.1.6 建立表结构

一个“学生信息”表的结构



学号	姓名	性别	籍贯	奖学金	团员否	入校日期	email	照片
		男		0.00	☒	2008-9-10		

记录: 1 共有记录数: 1



2.1.7 主键

若一个字段的值可以惟一标识表中的记录，则该字段所代表的信息称为主键。如字段“学号”能惟一标识一条记录，可以将“学号”设置为主键。设置为主键的字段名又被称为主关键字。

主键可以保证数据输入的安全性，作为主键的字段禁止重复值，也不能为空。主键还用于在表之间建立关系，建立了关系的多个表使用起来就像一个表一样。

主键类型有3种：自动编号、单字段、多字段。

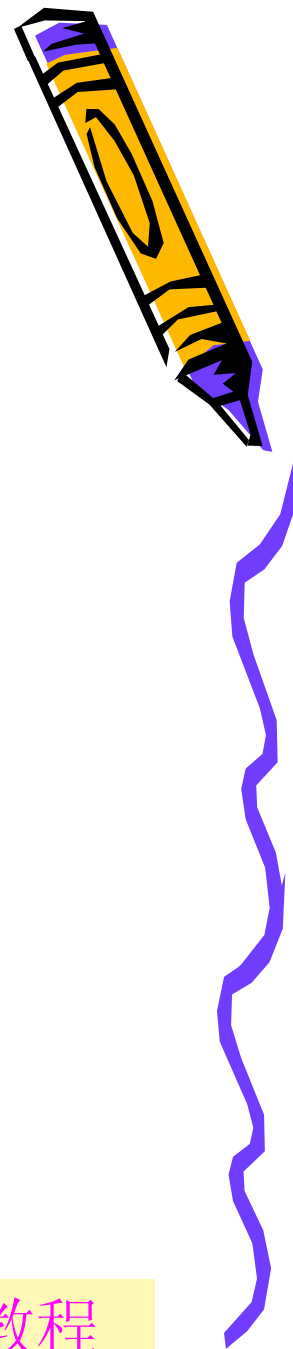


2.2 向表中输入数据

向表中输入数据可以直接输入或导入外部数据。从外部获取数据有导入和链接两种类型，导入的数据一旦操作完毕就与外部数据源无关。链接的数据只在当前数据库形成一个链接表对象，其内容随着数据源的变化而变化。

2.2.1 打开表的数据表视图

1. 在表的设计视图状态下，单击“数据表视图”按钮。
2. 在表的设计视图状态下，“视图”菜单→“数据表视图”。
3. 在库中选取表→单击“打开”按钮。
4. 在库中双击一个表的名字。



2.2.2 向表中直接输入数据

输入OLE型数据

单击某记录的“照片”字段→“插入”菜单→“对象”→对象类型选“画笔图片”→单击“确定”按钮→在画笔窗口单击“编辑”菜单→“粘贴来源”→在磁盘中选图片→双击图片将图片粘入画笔窗口→关闭窗口。



	学号	姓名	性别	籍贯	奖学金	团员否	入校日期	email	照片
✎	0801	张三	男	山东	300.00	☐	2008-9-10	aa@163.com	位图图像
	0802	李四	女	山西	320.00	☒	2008-9-12	bb@126.com	位图图像
	0803	王五	男	河南	280.00	☒	2008-9-11	cc@sohu.com	位图图像
*			男		0.00	☒	2008-9-10		

记录:   1    共有记录数: 3

2.2.3 向库中导入txt文件作为表

例：导入txt文件作为表



ACCESS实例教程

(1) 在数据库窗口中单击表对象→单击“新建”按钮→选“导入表”→“确定”→选文件位置→选文件类型为“文本文件”→选文件→单击“导入”按钮。显示：

☒ 带分隔符 - 用逗号或制表符之类的符号分隔每个字段 (D)

☐ 固定宽度 - 字段之间使用空格使所有字段在列内对齐 (W)

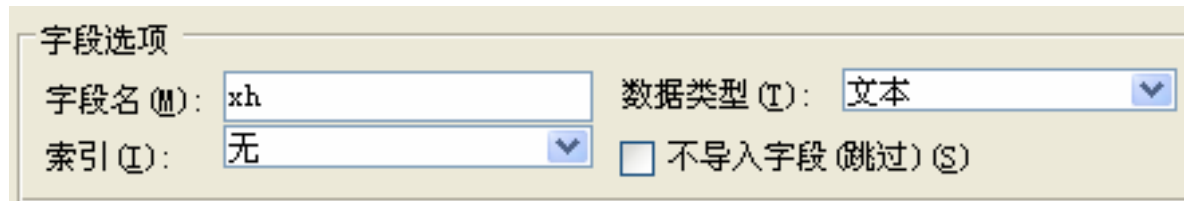
来自文件 D:\ACCESSLX\CJ.TXT 的示例数据。

1	xh, sx, yy, yw
2	0801, 90, 87, 96
3	0802, 85, 90, 79
4	0803, 94, 90, 88



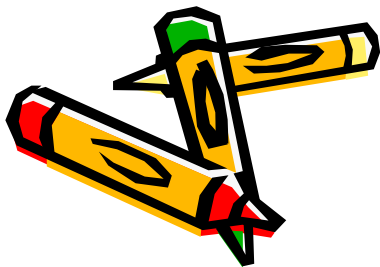
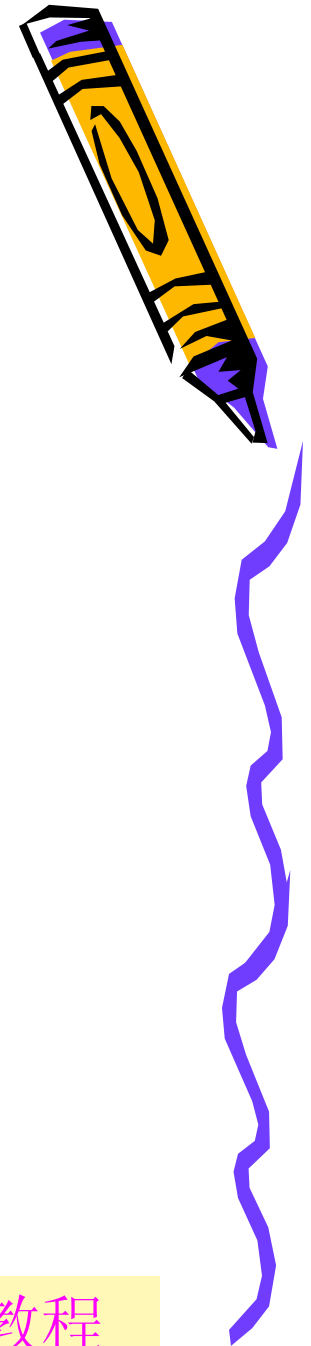
ACCESS实例教程

(2) 单击“下一步”→勾选“第一行包含字段名称”→单击“下一步”→选“新表中”→单击“下一步”→给xh字段的数据类型选“文本”。



字段选项	
字段名 (N):	xh
数据类型 (T):	文本
索引 (I):	无
☐ 不导入字段 (跳过) (S)	

(3) 单击“下一步”→选“不要主键”→单击“下一步”→给表起名为“成绩”→单击“完成”按钮。





2.2.4 向库中导入电子表格作为表

例：导入xls文件作为表

	A	B	C	D	E
1	学号	身高cm	体重kg	视力	
2	0801	1.75	66	2.00	
3	0802	1.60	52	1.50	
4	0803	1.80	70	1.70	
5					

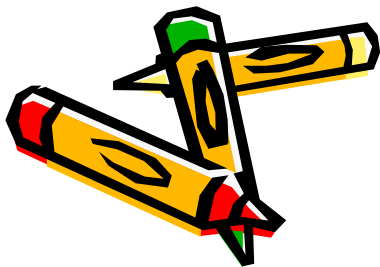
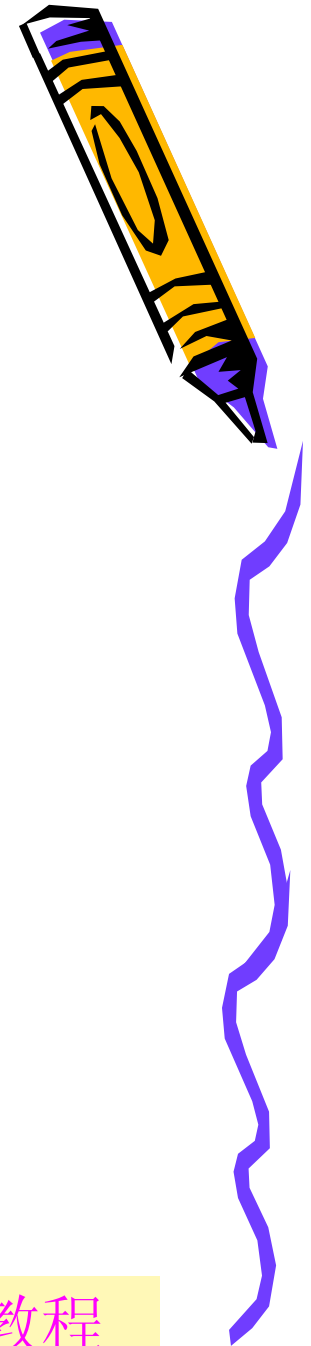


(1) 在数据库窗口中单击表对象→“文件”菜单→选“获取外部数据”→选“导入”→选文件位置→文件类型选“Microsoft Excel”→选文件→单击“导入”按钮。显示：

工作表“Sheet1”的示例数据。

1	学号	身高cm	体重kg	视力
2	0801	1.75	66	2.00
3	0802	1.60	52	1.50
4	0803	1.80	70	1.70

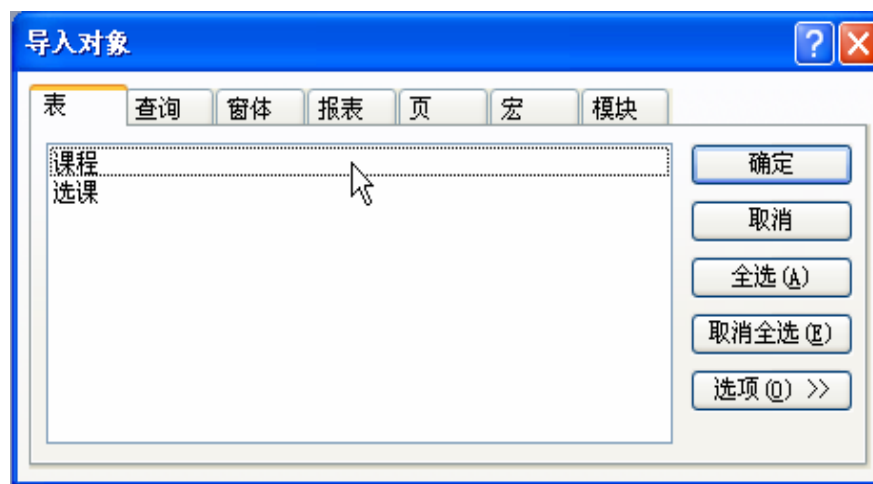
(2) 单击“下一步”→勾选“第一行包含列标题”→“下一步”→选“新表中”→两次单击“下一步”→选“不要主键”→“下一步”→给表起名为“体检”→“完成”。



2.2.5 将另一个库中的表导入到当前库中

例：将另一个库中的表导入到当前库中

(1) 在数据库窗口中单击表对象→“文件”菜单→“获取外部数据”→“导入”→选文件位置→文件类型选*.mdb→选文件→单击“导入”按钮。显示：



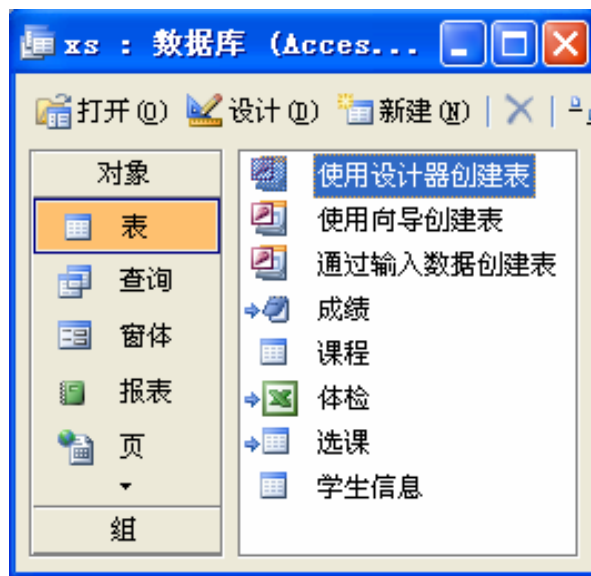
(2) 单击一个表（如：“课程”表）→“确定”。



2.2.6 链接外部数据

链接的文件在当前库像表一样显示，但不是库中真正的表。
单击表对象→“文件”菜单→“获取外部数据”→“链接表”→选文件位置→选文件类型→选文件→单击“链接”按钮。

链接文件的图标前都有链接标记。



2.2.7 将数据表导出为其他类型文件

例：将表导出为文本文件、电子表格

“文件”菜单→“导出”→选保存位置→选导出的文件类型→为文件起名。

文件名 (N):	公共课1. txt	▼
保存类型 (T):	文本文件 (*.txt;*.csv;*.tab;*.a	▼



2.2.8 表对象的复制、删除与重命名

用系统的“编辑”菜单中选取相应操作项。其中，剪切、复制、粘贴、删除、重命名都可以使用快捷键。

分别是：Ctrl+X、Ctrl+C、Ctrl+V、del、F2。



2.3 表记录的操作

对表记录的基本操作包括选定记录、添加记录、修改记录、删除记录等。

2.3.1 选定记录

1, 拖动鼠标选记录

学生信息：表				
	学号	姓名	性别	籍贯
▶	0801	张三	男	山东
■	0802	李四	女	山西
⇒	0803	王五	男	河南
	0802	李四	女	山西
*			男	

2, 用记录定位器

记录:      共有记录数: 3



3, 用菜单



2.3.2 添加记录

新添加的记录只能放在表的末尾，实际上是追加记录。将光标定位在表记录下面的第一个空行，然后输入新记录的各项数据项。



2.3.3 修改记录

将光标移到数据所在位置直接进行编辑修改即可。在数据表中移动光标除了用鼠标，还可以用快捷键。

上箭头	上一条记录的当前字段
下箭头	下一条记录的当前字段
左箭头, Shift+ Tab	当前记录当前字段的前一个字段
右箭头, Tab, 回车	当前记录当前字段的后一个字段
Ctrl+上箭头	第一条记录的当前字段
Ctrl+下箭头	最后一条记录的当前字段
Home	选中一个字段值, Home键使光标移到当前记录的第一个字段
End	选中一个字段值, End键使光标移到当前记录的最后一个字段
Ctrl+ Home	选中一个字段值, Ctrl+ Home键使光标移到第一条记录的第一个字段
Ctrl+ End	选中一个字段值, Ctrl+End键使光标移到最后记录的最后一个字段



2.3.4 删除记录

- 1, 右击选取的记录→快捷菜单中选“删除记录”。
- 2, 选取记录→按Del键。
- 3, 选取记录→“编辑”菜单→“删除记录”。

说明：删除记录的操作是不能撤消的。



2.3.5 查找与替换数据

查找和替换是同一个对话框中两个不同选项卡。

查找：用鼠标单击某列→“编辑”菜单→“查找”→在“查找内容”中输入字符串→选“搜索”范围→单击“查找下一个”按钮。

替换：用鼠标单击某列→“编辑”菜单→“替换”→在“查找内容”中输入字符串→在“替换为”中输入要替换的字符串→单击“查找下一个”按钮→找到查找目标后单击“替换”按钮。



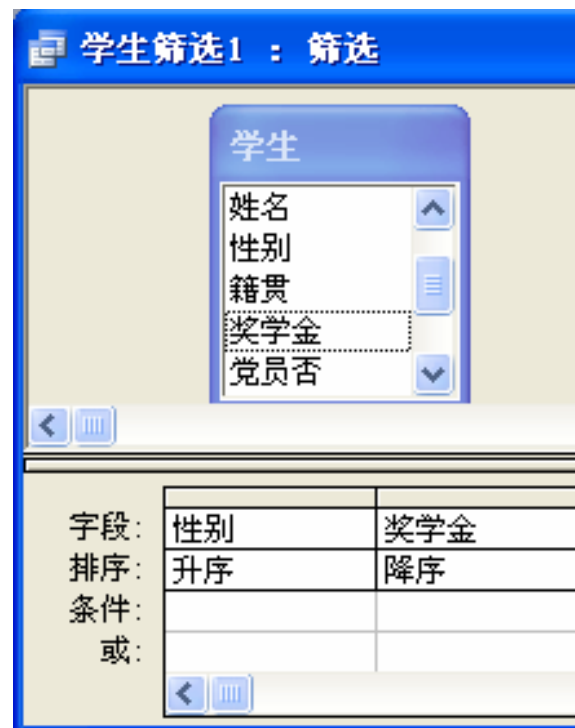
2.3.6 排序记录

简单排序：

在数据表视图选一个字段，单击“升序排序”或“降序排序”按钮，字段的值被排序。

高级排序：

“记录”菜单→“筛选”→“高级筛选/排序”→在筛选窗口选字段和排序方式。



例：筛选记录

设置条件：

字段：	性别	籍贯	奖学金
排序：			升序
条件：	"男"	"山东"	
或：			

< []

筛选结果：

学生：表						
	学号	姓名	性别	籍贯	奖学金	党员否
▶	080101	张红叶	男	山东	300.00	☐
	080204	张峰	男	山东	310.00	☒
*					0.00	☐

记录： [] 1 [] [] [] 共有记录数：2（已筛选的）



ACCESS实例教程

2.4 数据表的格式化

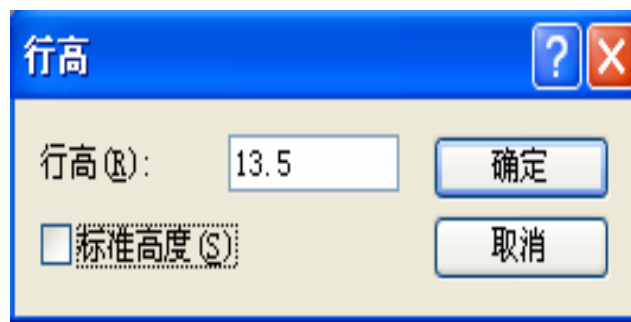
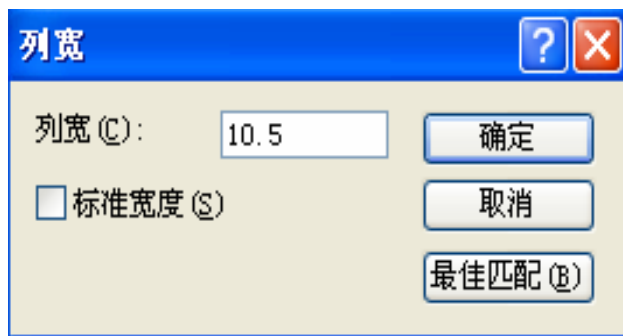
2.4.1 设置数据表格样式

“格式”菜单
→ “数据表”，打开
“设置数据表格式”
对话框，可以在对话框
中更改数据表的显示
样式。如背景色、
网格线颜色、单元格
效果、网格线显示方
式，等。



2.4.2 设置行高列宽

“格式”菜单→“列宽”→在“列宽”对话框中输入所需的列宽值→单击“确定”按钮。“格式”菜单→“行高”→在“行高”对话框中输入所需的行高值→单击“确定”按钮。



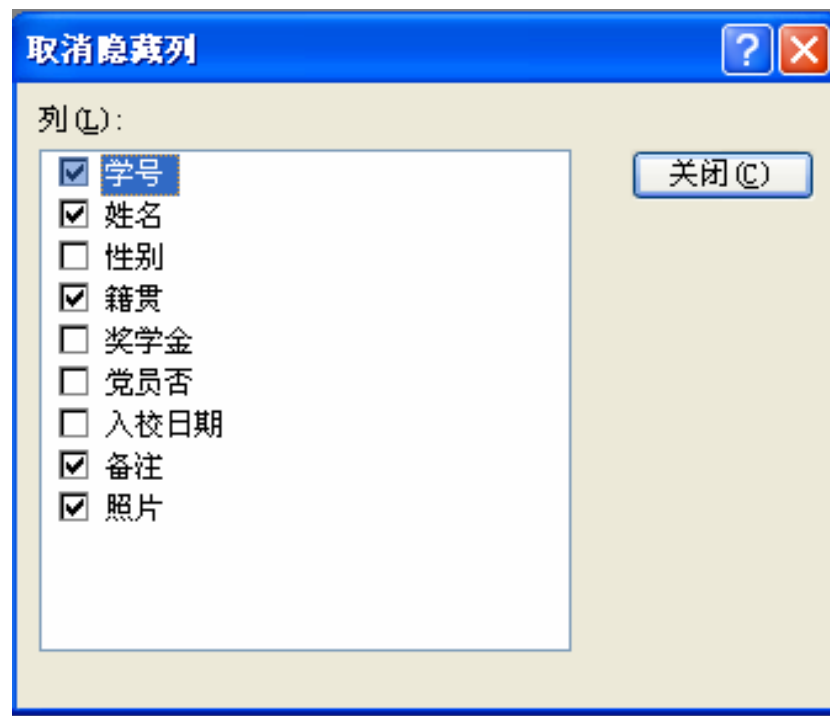
2.4.3 设置字体

“格式”菜单→“字体”→在“字体”对话框做设置，可以改变数据表的字体、字形、字号、字颜色等。



2.4.4 隐藏列与取消隐藏列

选取一列或几列→“格式”菜单→“隐藏列”，选中的列被隐藏。“格式”菜单→“取消隐藏列”→在隐藏列字段前打对勾→单击“关闭”按钮，该列被取消隐藏。



2.4.5 冻结列与取消冻结列

选中一列或几列→“格式”菜单→“冻结列”，或右击选中的列→快捷菜单中选“冻结列”，冻结的列显示数据表最左边。

拖动水平滚动条查看数据表，无论怎样水平移动数据表，被冻结的列始终显示在窗口最左边。

“格式”菜单→“取消对所有列的冻结”，数据表中不再有冻结列。

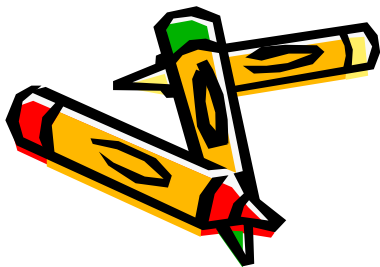


2.5 建立表对象之间的关联

2.4.1 建立表的基本原则

1. 一个表围绕一个主题，一事一地，避免大而全。
2. 表中的字段代表原子数据，不可再分。象总分、平均分这样的字段，是计算得到的二次数据，不要作为表中字段。
3. 表之间减少重复字段，只保留做连接用的公共字段即可。
4. 设置关键字和外部关键字，用于表之间建立联系。

说明：关键字是当前表的主键字段，外部关键字是在其他表做主键字段。



2.4.2 表之间的关系

表之间有3种关系：一对一、一对多、多对多。其中，多对多关系都被拆分成几个一对多关系。所以，只须建立表之间的一对一关系和一对多关系即可。

最常用的关系是一对多关系，要满足如下几点要求：

1. 建立关系的两个表要有公共字段，两字段类型相同，名称可以不同。
2. 建立关系以后，“一”方的表称为主表，“多”方的表称为子表。
3. 子表中公共字段的值在主表公共字段中有匹配的值，主表公共字段的值不必与子表匹配。子表可以是空表。



2.4.3 参照完整性

参照完整性是一个规则系统，用来确保相关表行之间关系的有效性，并确保不会在无意之间删除或更改相关数据。

在定义两个表之间关系时，如果实施了参照完整性，子表将不能随意添加记录，也不能随意更改公共字段中的值。如果同时勾选了“级联更新”和“级联删除”选项，在主表中更改公共字段中的值，子表对应的值自动被更改，在主表中删除某条记录，子表中所有对应记录自动被删除。

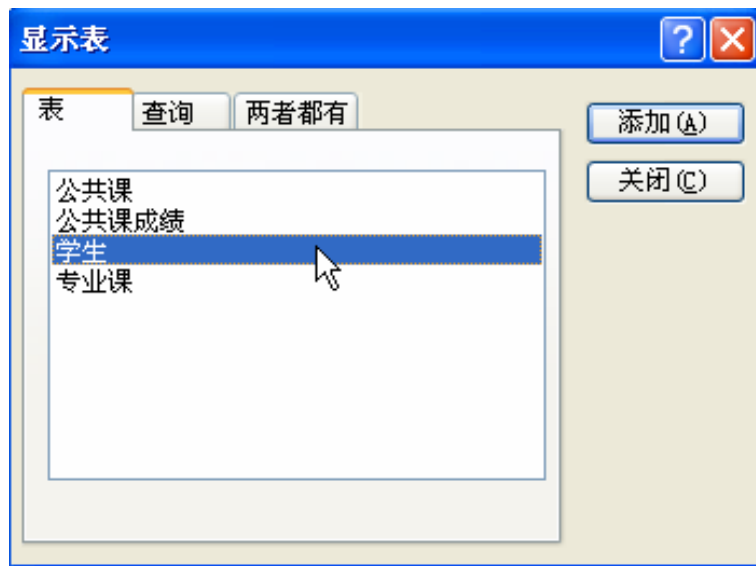


2.4.4 建立表之间的关联

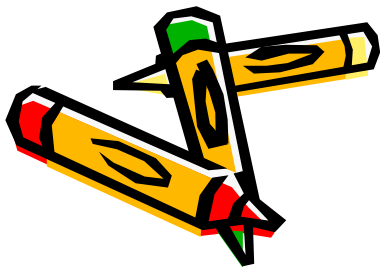
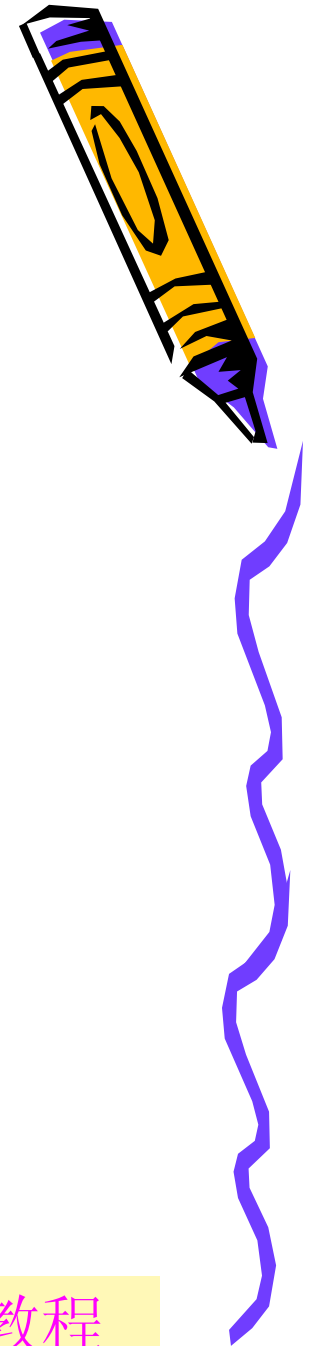
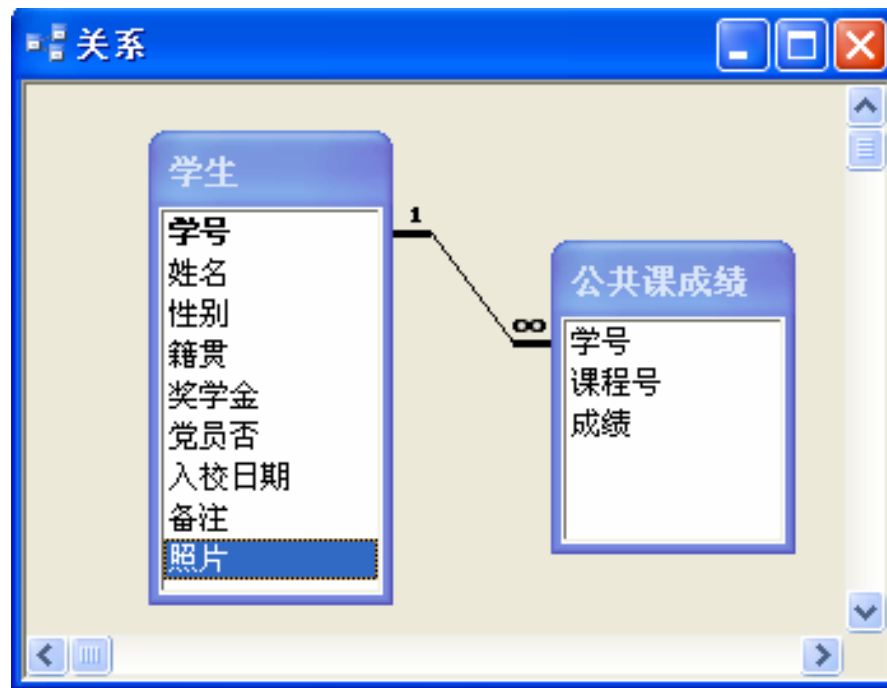
例：建立学生表与公共课成绩表一对多关联，实施参照完整性。

(1) 在数据库窗口→“工具”菜单→“关系”，或单击“关系”按钮。

(2) 在“显示表”窗口选取表→单击“添加”按钮。



- (3) 用鼠标在相关字段之间划一条线→在“编辑关系”窗口勾选“实施参照完整性”、“级联更新相关记录”和“级联删除相关记录”→单击“创建”按钮。
- (4) 两表之间建立了关系。



ACCESS实例教程

2.4.5 拆分表

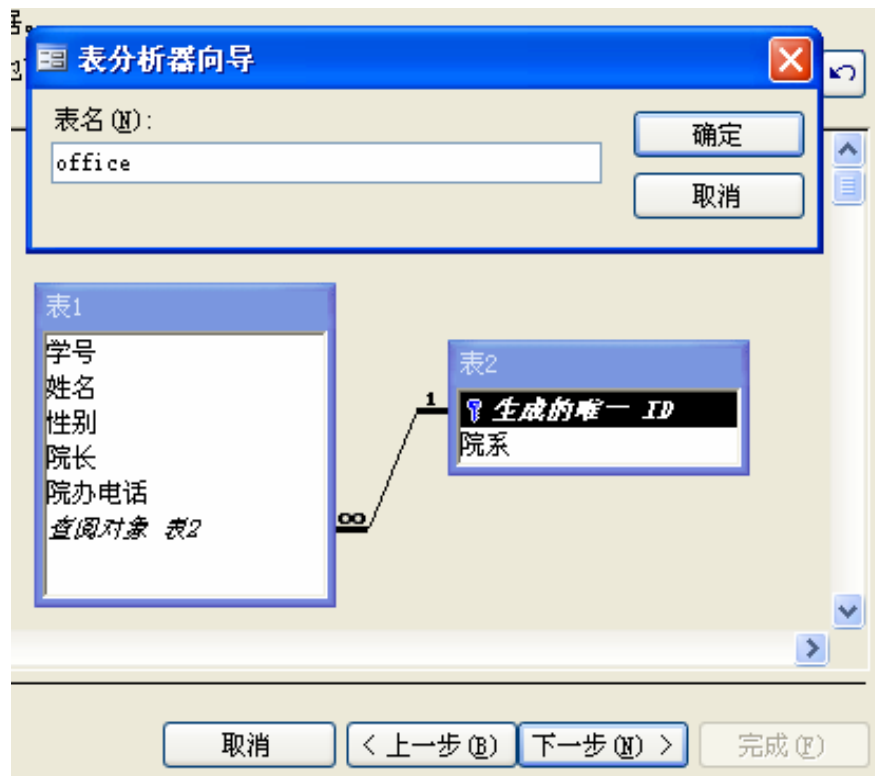
例：将“学生名单”表拆分为两个新表

(1) “学生名单”表

学生名单 : 表						
	学号	姓名	性别	院系	院长	院办电话
	080101	张红叶	女	英语系	王之元	8877991
	080102	李丁	男	英语系	王之元	8877991
	080103	王伟	男	英语系	王之元	8877991
	080201	赵大山	女	法律系	孙子山	8877992
	080202	钱小红	女	法律系	孙子山	8877992
	080203	孙力伟	男	法律系	孙子山	8877992
	080204	张峰	男	法律系	孙子山	8877992
▶	080301	李超群	男	中文系	周永波	8877993
	080302	王艳艳	女	中文系	周永波	8877993
	080303	赵娟	女	中文系	周永波	8877993
*						
记录: 8 共有记录数: 10						



- (2) “工具” 菜单→“分析”→“表”。
- (3) 按向导指示向下操作。
- (4) 将“院系”字段拖到窗口空白处产生新表→在表名框中输入office。



ACCESS实例教程

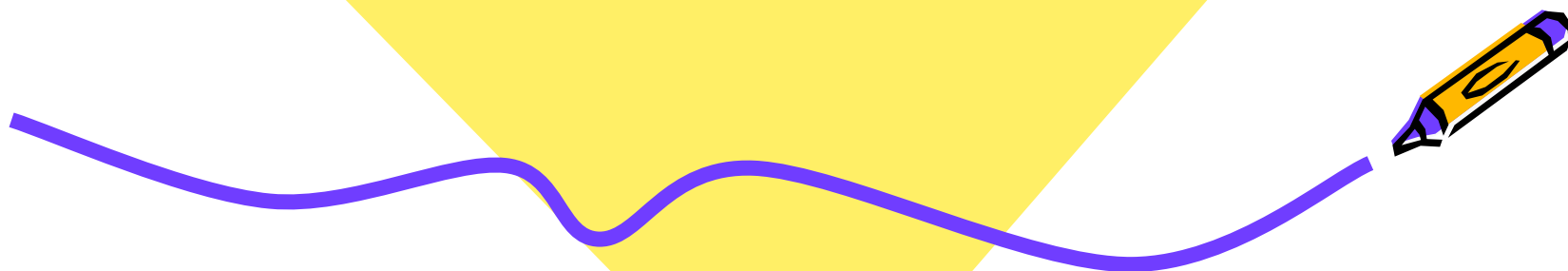
- 5, 将“院长”字段和“院办电话”字段拖到office表中→选中“院系”字段→单击窗口右上方按钮,“院系”字段被设为主键。
- 6, 选中“学号”字段→单击按钮将其设为主键→单击按钮将“学号”字段所在的表命名为“xs”。
- 拆分出来的两个新表:





ACCESS实例教程

——查询的操作（第3章）



本章要点:

- 1, 建立各种类型的查询
- 2, 建立计算字段
- 3, 使用统计函数
- 4, SQL语句与子查询



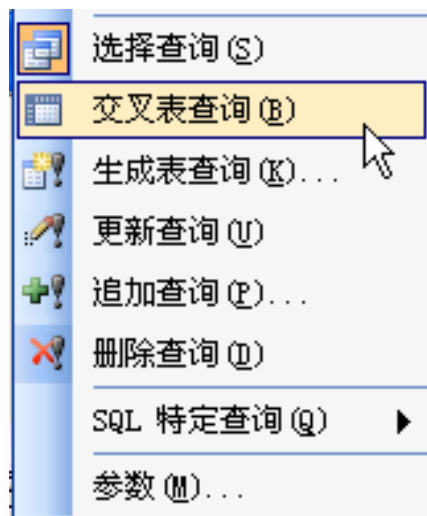
3.1 认识查询对象

查询是Access数据库的一个重要对象，用来查看、处理和分析数据。查询的数据源可以是一个或多个数据表或已存在的查询。查询产生的操作结果形式上看类似于数据表，实际上是一个动态的数据集合，每次打开查询，都会显示数据源的最新变化情况。查询与数据源表是相通的，在查询中对数据所做的修改可以在数据源表中得到体现。



3.1.1 查询对象的基本类型

查询对象有5类：选择查询、交叉表查询、操作查询、参数查询、SQL查询。其中操作查询又包括4种：生成表查询、更新查询、追加查询、删除查询。操作查询需要用“运行”命令使查询生效。



3.1.2 查询的设计视图

常用的查询视图有3种：设计视图、数据表视图、SQL视图。查询的设计视图窗口分上下两部分，上半部分是“字段列表”区，放置查询的数据源。下半部分是“设计网格”区，放置在查询中显示的字段和在查询中做条件的字段。

字段:	姓名	课程名称	成绩	性别
表:	学生	公共课	公共课成绩	学生
排序:			升序	
显示:	☒	☒	☒	☐
条件:			>=60	"男"
或:				



3.2 通配符、运算符与常用函数

3.2.1 通配符

1. *, 代表任意多个任何字符。

例如：李*, 通配所有第一个字符为“李”的字符串。

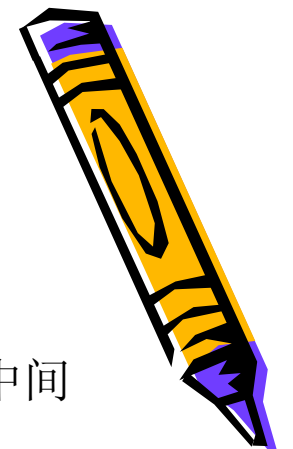
2. ?, 代表任意一个任何字符。

例如：李?, 通配所有第一个字符为“李”且只有2个字符的字符串。

3. [], 通配方括号内的任何单个字符。

例如：淡[红绿黄]色, 代表的字符串有：淡红色、淡绿色、淡黄色。





4. !，通配不在方括号内的任何单个字符。

例如：淡[!红绿黄]色，代表的字符串有：淡蓝色、淡紫色、等，中间的字不能是红、绿、黄3个字中的任何一个。。

5. -，通配指定范围内的任何字符，该范围必须是升序，通常针对英文字母。

例如：a[e-g]b，代表的字符串有：aeb、afb、agb。

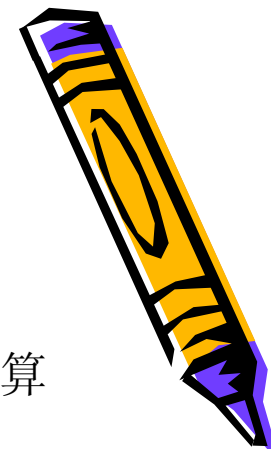
6. #，代表任意一个数字字符。

例如：2##2，代表的字符串有：2112、2802、等，第一个和最后一个必须是2，中间2个可以是任何数字。



3.2.2 普通运算符

1. **算术运算符：** +、-、*、/，乘、除同级，加、减同级，同级运算从左到右，乘、除运算优先于加、减运算。
2. **关系运算符：** >、>=、<、<=、!=、==，Access系统用true或-1表示“真”，用false或0表示“假”。
3. **逻辑运算符：** not、and、or，运算结果是逻辑值。
4. **连接运算符：** +、&，用于字符连接，+号要求两边必须是字符型，连接后得到新字符串。&号不论两边的操作数是字符串还是数字，都按字符串连起来，得到新字符串。



3.2.3 特殊运算符

1. like, 为文本字段设置查询模式, 支持通配符。
如: like "李*", Like "*红*"
2. in, 指定一个值列表作为查询的匹配条件, 不支持通配符。
如: in("张三","李四","王五")
3. between, 指定数据范围, 用and连接起始数据和终止数据。
如: between 10 and 30, 相当于: ≥ 10 and ≤ 30
4. is Null, 查找为空的数据。
5. is not Null, 查找非空的数据。



3.2.4 字符函数

1. left函数，从字符串左边取n个字符，得到左子串。

格式：left(“字符串”,n) 或 left(string型变量名,n)

2. right函数，从字符串右边取n个字符，得到右子串。

格式：right(“字符串”,n) 或 right(string型变量名,n)

3. mid函数，从字符串第n1个字符开始取n2个字符，得到子字符串。

格式：mid(“字符串”,n1,n2) 或 right(string型变量名,n1,n2)

说明：如果省略n2，则从字符串第n1个字符开始一直取到最后。



3.2.5 日期函数

1. date函数，返回系统当前日期。

格式：date() 或 date

2. now函数，返回系统当前日期和时间。

格式：now() 或 now

3. year函数，返回日期数据中的年份。

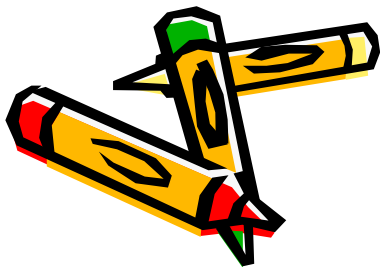
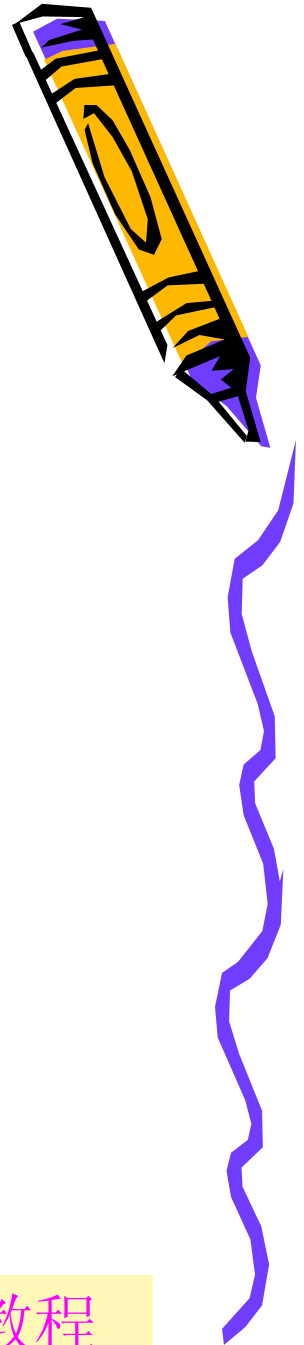
格式：year(日期常量) 或 year(日期/时间型变量名)

4. month函数，返回日期数据中的月份。

格式：month(日期常量) 或 month(日期/时间型变量名)

5. day函数，返回日期数据中日子的号码。

格式：day(日期常量) 或 day(日期/时间型变量名)



3.2.6 统计函数

1. sum函数, 对数字型表达式求和。

格式: sum(数字型表达式), 如: sum([工资]+[奖金])

2. avg函数, 对数字型表达式求和。

格式: avg(数字型表达式), 如: avg([奖金])

3. count函数, 对表达式统计个数。

格式: count(表达式), 如: count([编号])

4. max函数, 求数字型表达式的最大值。

格式: max(数字型表达式), 如: max([奖金])

5. min函数, 求数字型表达式的最小值。

格式: min(数字型表达式), 如: min([年龄])



3.3 建立简单的选择查询

简单的选择查询包括从一个或多个表中提取字段、按一定条件提取记录、对查询结果进行排序等操作。

3.3.1 从一个或多个表中提取字段

按条件从一个或多个表中提取字段，得到的查询结果就象组建了新数据表。



3.3.2 从一个或多个数据源中提取记录

例：以查询为源建立简单查询

1，已知数据表：

student : 表			
	身份证号	姓名	家长身份证号
▶	101101196001012370	张春节	
	370706198011035670	李强	370707196201012370
	370707196201012370	李永飞	
	370707196410015670	王元国	
	370707198011025660	王好一	370707196410015670
	370708198001013760	王媛	371101196001012370
	370708198011015770	张天	370708196510015760
	371101196001012370	王中华	
*			

记录: 1 共有记录数: 8



2, 建立查询1, 显示家长的身份证号与家长的姓名

查询1 : 选择查询

student

*
身份证号
姓名
家长身份证号

字段:	身份证号	姓名	家长身份证号
表:	student	student	student
排序:			
显示:	☒	☒	☐
条件:			Is Null
或:			



ACCESS实例教程

3, 建立查询2, 显示学生的身份证号、姓名和该学生家长的身份证号。

查询2 : 选择查询

student

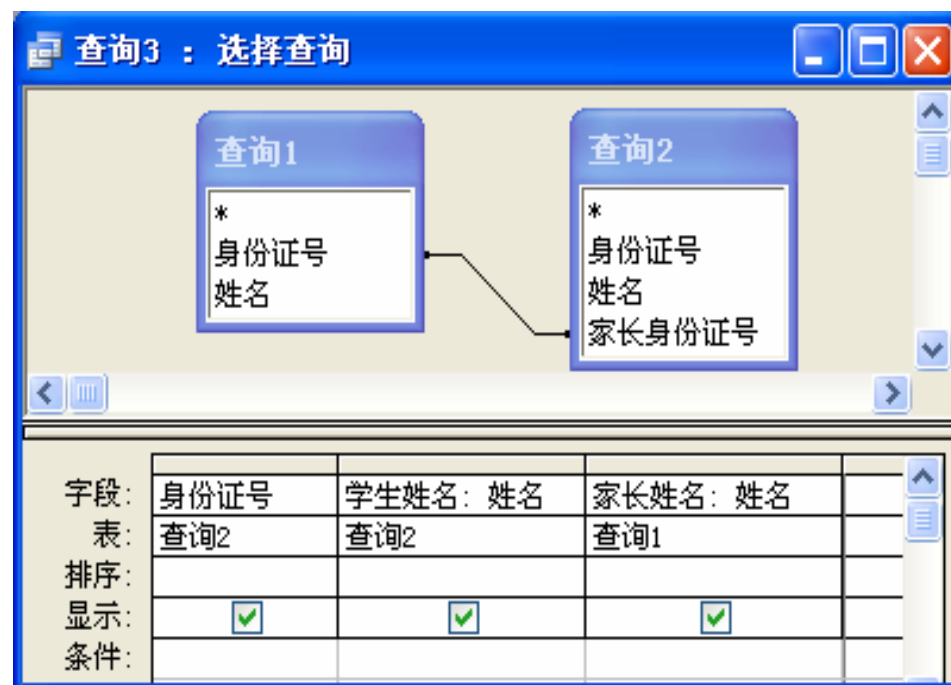
*
身份证号
姓名
家长身份证号

字段:	身份证号	姓名	家长身份证号
表:	student	student	student
排序:			
显示:	☒	☒	☒
条件:			Is Not Null
或:			



ACCESS实例教程

4, 建立查询3, 显示了家长是本校校友的学生。



ACCESS实例教程



3.3.3 用函数和特殊运算符设置查询条件

例：查找姓李的学生和有绘画特长的学生，显示字段为：年级、姓名、爱好。其中年级取学号的前4位。

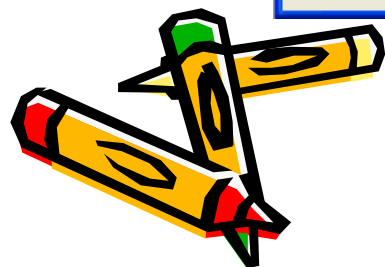
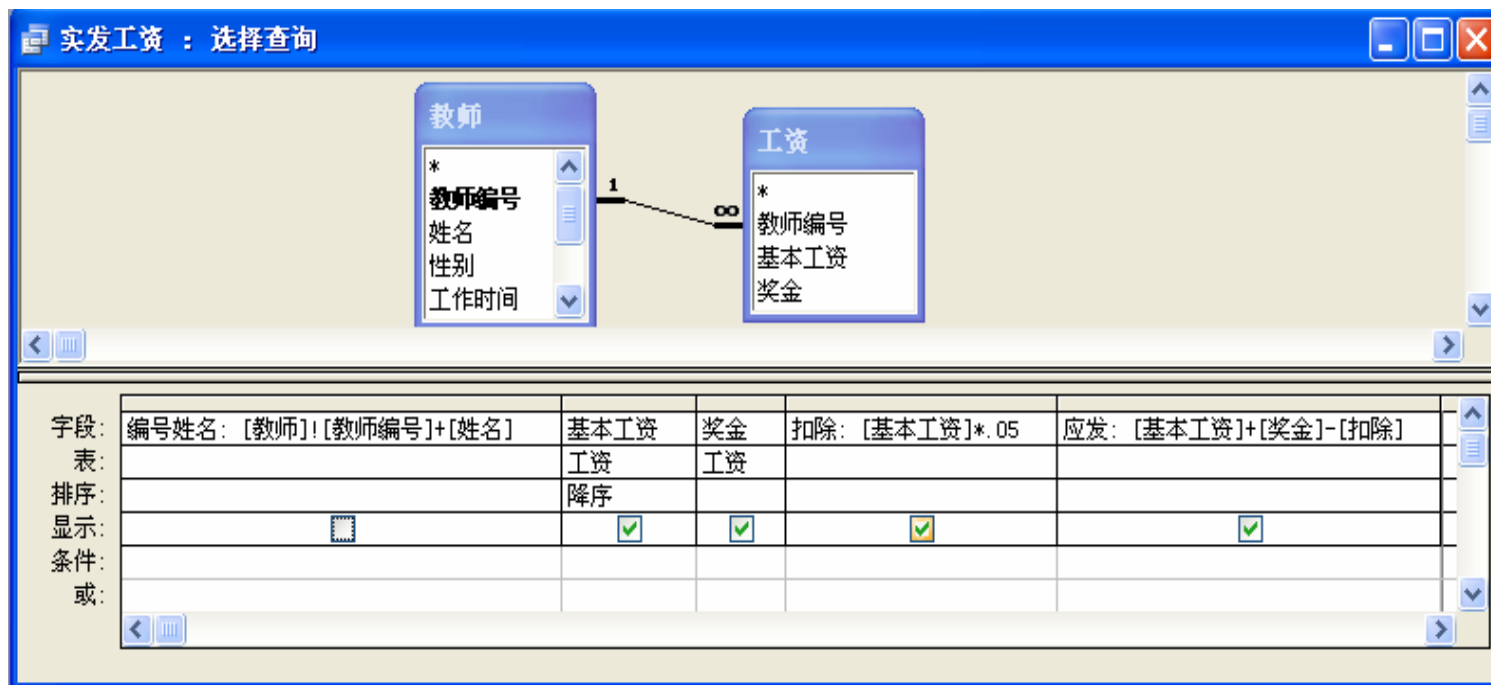
字段:	年级: Left([学号], 4)	姓名	爱好: 备注
表:		学生	学生
排序:			
显示:	☒	☒	☒
条件:		Left([姓名], 1) = "李"	
或:			Like "*绘画*"



3.4 建立带计算的选择查询

3.4.1 用表达式建立计算字段

例：计算教师工资



3.4.2 用统计函数建立计算字段

统计函数	功能
分组	对记录按字段值分组，字段值相同的记录只显示一个。
总计	对数字型字段的值求和。
平均值	对数字型字段的值求平均值。
最小值	求字段的最小值。
最大值	求字段的最大值。
计数	求字段值的个数。
标准差	对数字型字段的值求标准差。
方差	对数字型字段的值求方差。
第一条记录	求第一条记录的值。
最后一条记录	求最后一条记录的值。
表达式	定义表达式，不分组执行统计函数时用此选项。
条件	定义条件，执行统计函数时做条件的字段不显示在结果中。



例：求最高奖金、最低奖金、奖金最大差额

字段:	最高奖金: Max([奖金])	最低奖金: Min([奖金])	奖金最大差额: Max([奖金])-Min([奖金])
表:			
排序:			
显示:	☒	☒	☒
条件:			
或:			

奖金查询 : 选择查询

	最高奖金	最低奖金	奖金最大差额
▶	2250	1125	1125

记录: 共有记录数: 1



ACCESS实例教程

例：统计教师人数与奖金总额

字段:	教师人数: 教师编号	奖金合计: 奖金	
表:	工资	工资	
总计:	计数	总计	
排序:			
显示:	☒	☒	
条件:			
或:			

 教师人数 : 选择查询



	教师人数	奖金合计
▶	15	12350

记录:





共有记录数: 1



ACCESS实例教程

例：统计男、女讲师人数

字段：	性别	职称	人数：教师编号	职称	
表：	教师	教师	教师	教师	
总计：	分组	分组	计数	条件	
排序：					
显示：	☒	☒	☒	☐	
条件：				"讲师"	
或：					

男女讲师人数：选择查询			
	性别	职称	人数
▶	男	讲师	5
	女	讲师	3

记录： * 共有记录数



ACCESS实例教程

例：查询平均分低于班平均分的学生

(1) 先求班平均分

班平均：选择查询

学生

- * 学号
- 姓名
- 性别
- 籍贯

公共课成绩

- * 学号
- 课程号
- 成绩

1 — ∞

字段：	班级：Left([学生]![学号],4)	班平均：成绩
表：		公共课成绩
总计：	分组	平均值
排序：		
显示：	☒	☒
条件：		



ACCESS实例教程

(2) 再求人平均
均分

字段:	班级: Left([学生]![学号], 4)	姓名	人平均: 成绩
表:		学生	公共课成绩
总计:	分组	分组	平均值
排序:			
显示:	☒	☒	☒
条件:			

(3) 最后求小
于班平均
的人

小子班平均的人 : 选择查询

班平均

*

班级

班平均

人平均

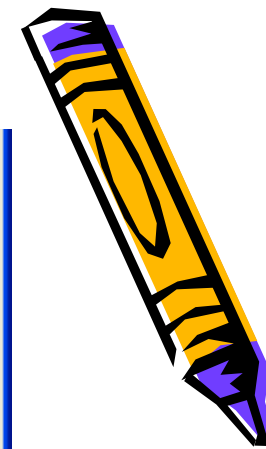
*

班级

姓名

人平均

字段:	班级	姓名	表达式1: [人平均]-[班平均]
表:	人平均	人平均	
排序:			
显示:	☒	☒	☐
条件:			<0



ACCESS实例教程

3.5 建立交叉表查询

交叉表查询是一种从水平和垂直两个方向对数据表进行分组统计的查询方法，用独特的概括形式返回表的统计数字。

建立交叉表查询至少要指定3个字段，一个字段用来分组作为行标题（行标题最多可以有3个），一个字段用来分组作为列标题（列标题只能有1个），一个字段放在行与列交叉位置作为统计项（统计项只能有1个）。



例：查询学生各门公共课的成绩

公共课成绩交叉表：交叉表查询

学生

- * 学号
- 姓名
- 性别
- 籍贯

公共课成绩

- * 学号
- 课程号
- 成绩

公共课

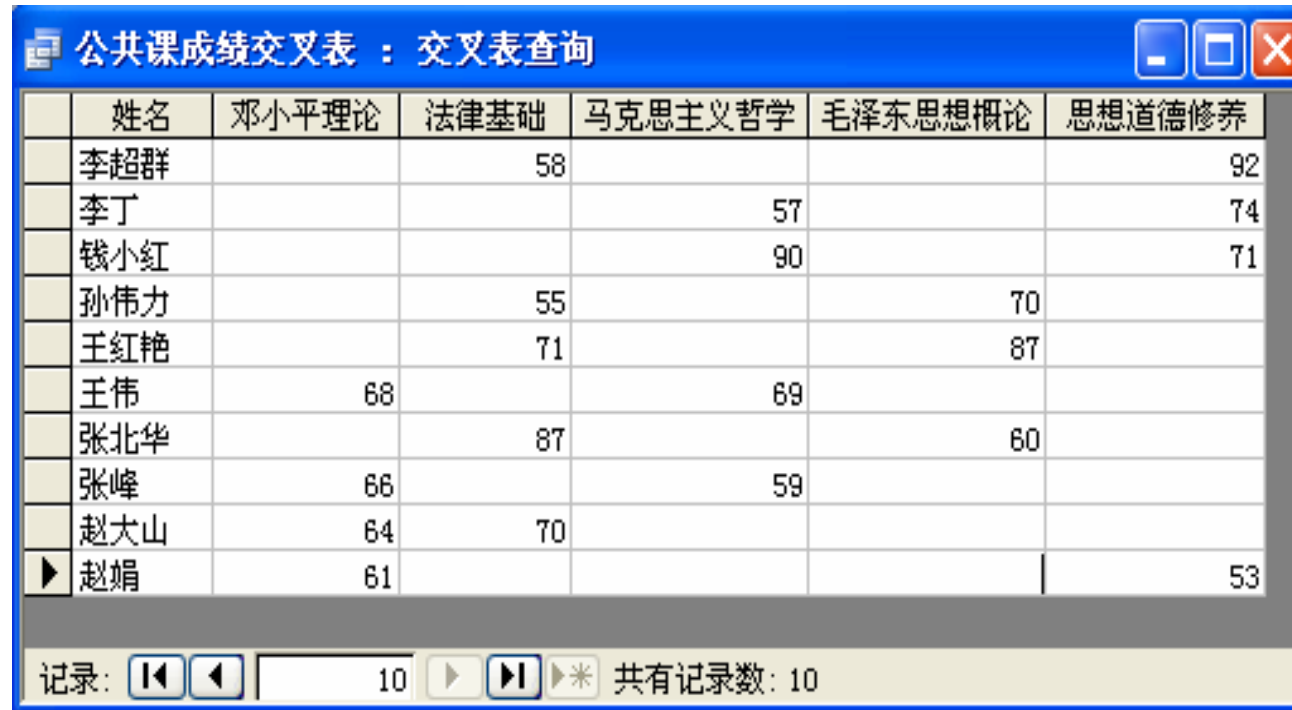
- * 课程编号
- 课程名称

字段：	姓名	课程名称	成绩
表：	学生	公共课	公共课成绩
总计：	分组	分组	第一条记录
交叉表：	行标题	列标题	值
排序：			
条件：			



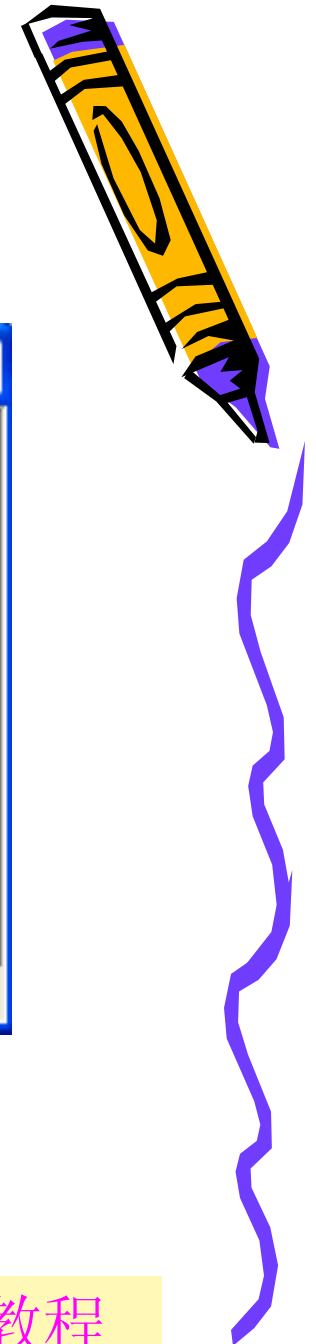
ACCESS实例教程

显示结果：



姓名	邓小平理论	法律基础	马克思主义哲学	毛泽东思想概论	思想道德修养
李超群		58			92
李丁			57		74
钱小红			90		71
孙伟力		55		70	
王红艳		71		87	
王伟	68		69		
张北华		87		60	
张峰	66		59		
赵大山	64	70			
赵娟	61				53

记录: 10 共有记录数: 10



ACCESS实例教程

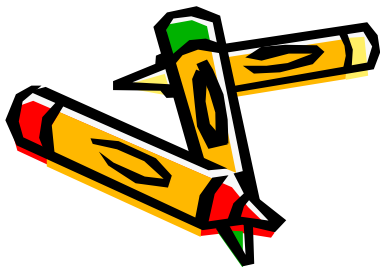
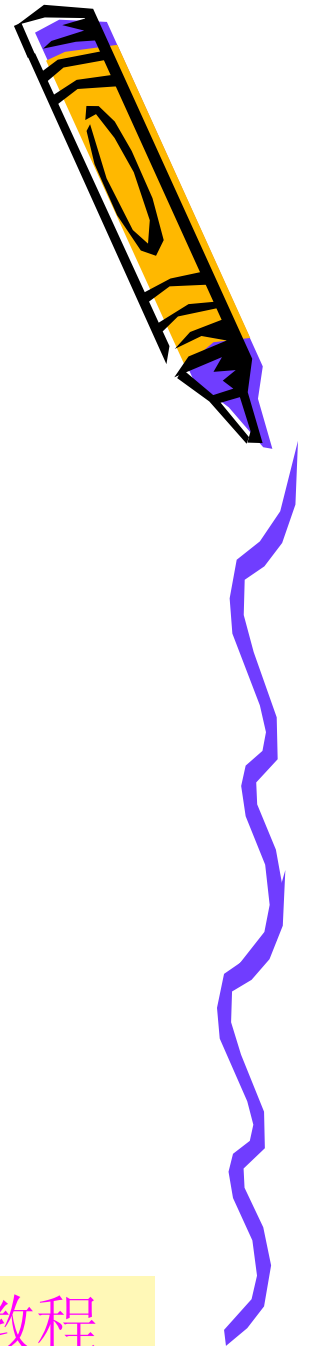


显示结果：



	班级	男	女
▶	0801	1	
	0802	2	
	0803	1	1

记录: [Navigation Buttons] 共有记录



3.6 建立操作查询

操作查询包括：生成表查询、更新查询、追加查询、删除查询。

3.6.1 生成表查询

生成表查询能将查询结果保存成数据表，使查询结果由动态数据集合转化为静态的数据表。新表不继承数据源表的关键字属性。

生成表查询通常用几个表中的数据组合起来生成新表，如果仅用一个表的数据生成新表，可以在数据库窗口用复制、粘贴的方法实现。



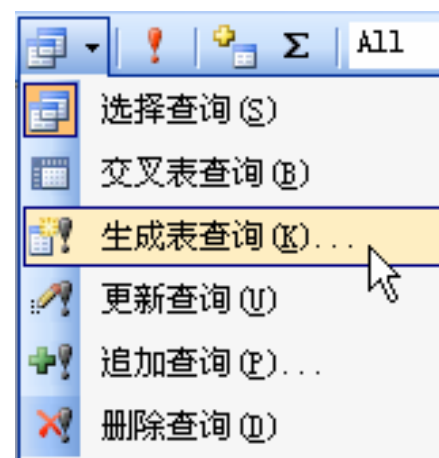
例： 将男学生成绩查询转换成数据表

(1) 先建查询



ACCESS实例教程

(2) 将查询转换为生成表查询

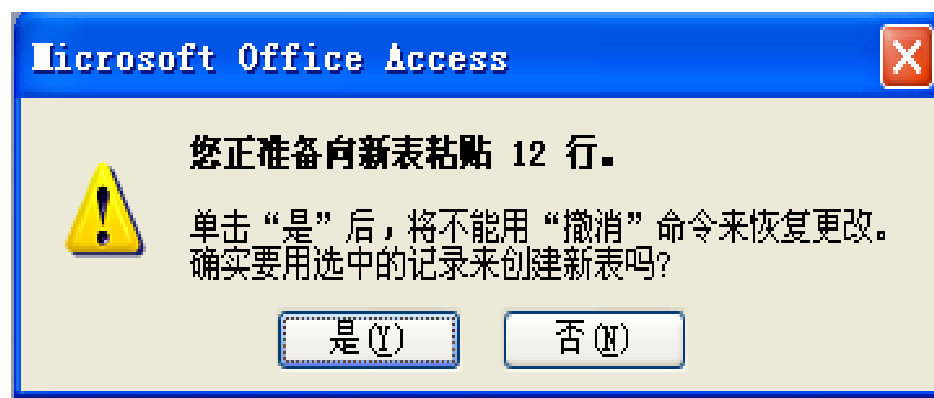


(3) 为新表起名



ACCESS实例教程

(4) 单击“运行”按钮



ACCESS实例教程

3.6.2 更新查询

更新查询能成批修改记录。用更新查询更改记录的数据项以后，无法用“撤消”命令取消操作。更新查询可以同时更新多个数据源和多个字段的值。在设计网格中，只将要更新的字段和做条件的字段放入即可。

例：将职称为讲师的“教师编号”字段前加“讲师”字符串，并将讲师的奖金增加50%。

字段：	教师编号	奖金	职称	
表：	教师	工资	教师	
更新到：	"讲师"+[教师]![教师编号]	[奖金]+[奖金]*.5		
条件：			"讲师"	
或：				



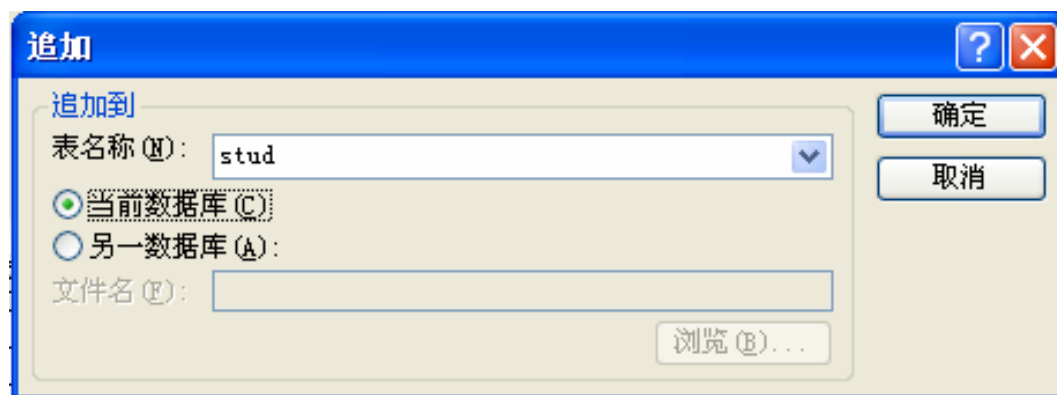
3.6.3 追加查询

追加查询能将数据源中符合条件的记录追加到另一个表尾部。数据源可以是表或查询，追加的去向是一个表。字段列表区只放提供字段的表或查询，被追加的表不能放入查询的字段列表区。数据源与被追加表对应的字段之间要类型匹配。



例：将籍贯为河北的记录追加到空表stud中，追加的字段为：班级、姓、名、性别。

(1) 选择追加去向



ACCESS实例教程

(2) 确定追加字段和条件

追加查询练习 : 追加查询

学生

- * 学号
- 姓名
- 性别
- 籍贯

公共课成绩

- * 学号
- 课程号
- 成绩

字段:	表达式1: Left([姓名],1)	表达式2: Mid([姓名],2)	成绩	籍贯
表:			公共课成绩	学生
排序:				
追加到:	姓	名	成绩	
条件:				"河北"
或:				



(3) 运行查询



	姓	名	成绩
▶	张	北华	60
	钱	小红	90
	李	超群	58
	张	北华	87
	钱	小红	71
	李	超群	92
*			0

记录: [Navigation Buttons] 共有记录数:



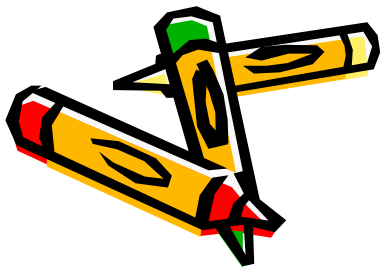
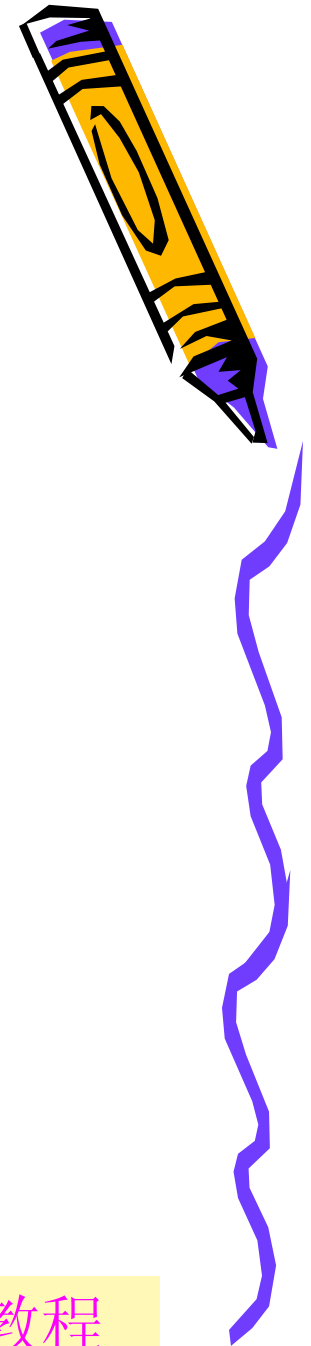
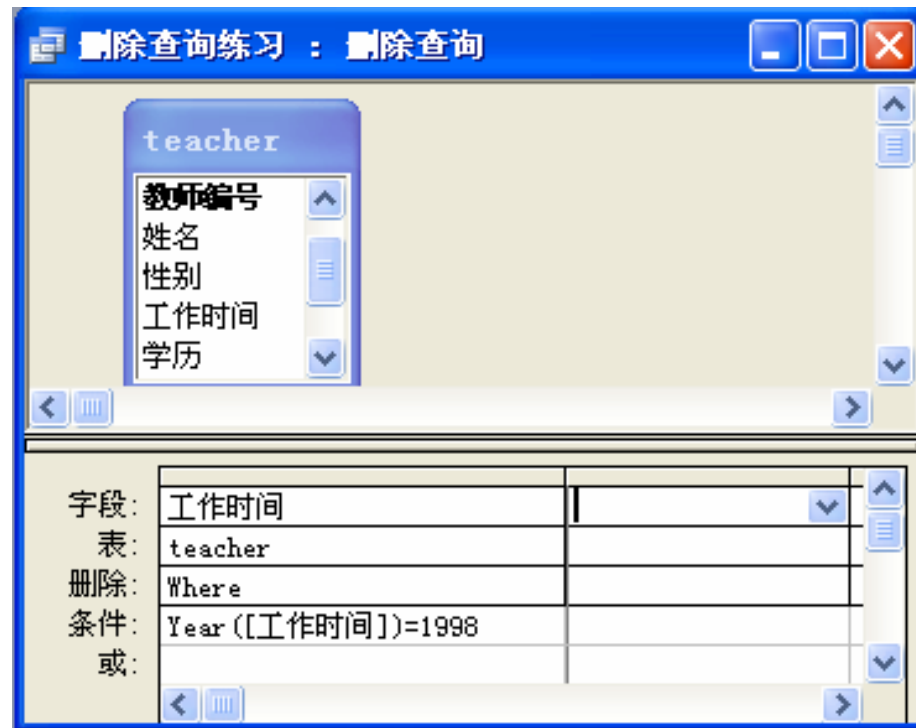
3.6.4 删除查询

删除查询能将数据表中符合条件的记录成批的删除。删除查询可以给单个表删除记录，也可以给建立了关系的多个表删除记录，多个表之间要建立参照完整性，并选择了“级联删除”选项。

在删除查询的设计网格中，只放入做删除条件的字段即可。运行删除查询后，被删除的表中记录不能用“撤消”命令恢复。



例：删除1998年参加工作的记录



ACCESS实例教程

3.7 建立参数查询

参数查询是按用户给定的字段值来查找记录的查询。参数查询提供一个对话框，提示用户输入一个参数值，然后查找并显示符合参数值的记录。

参数查询有单参数查询和多参数查询两种。

3.7.1 单参数查询

单参数查询只需要在字段中指定一个参数，参数通常是完整的字段值，也可以是字段的部分值，还可以是窗体控件中的值。



例：以输入的姓名为参数，显示该同学的公共课成绩

字段：	姓名	课程名称	成绩
表：	学生	公共课	公共课成绩
排序：			
显示：	☒	☒	☒
条件：	[请输入姓名：]		
或：			

运行时提示

输入参数值

请输入姓名：

王伟

确定 取消



ACCESS实例教程

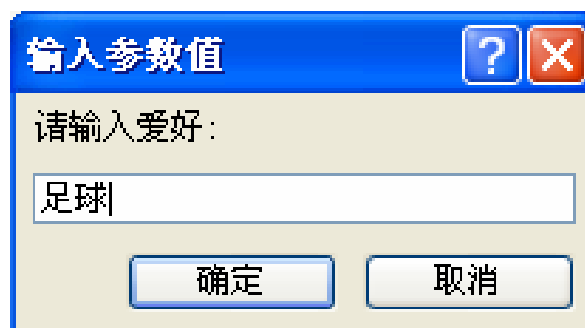
例：用部分字段值做参数

字段:	班级: Left([学号], 4)	姓名	备注
表:		学生	学生
排序:			
显示:	☒	☒	☒
条件:			Like "*" + [请输入爱好:] + "*"
或:			



ACCESS实例教程

运行时提示:



输入参数值

请输入爱好:

足球

确定 取消

显示结果:



	班级	姓名	备注
▶	0801	李丁	足球, 排球, 篮球
	0801	王伟	长跑, 绘画, 足球
	0802	张峰	篮球, 足球, 排球
	0803	李超群	武术, 绘画, 足球
*			

记录: 4 共有记录数: 4

ACCESS实例教程

3.7.2 多参数查询

多参数查询在多个字段中指定参数，执行时提示对话框从左到右依次显示，用户则依次输入多个参数，最后根据多个参数显示查询结果。

字段:	班级: Left([学生]![学号],4)	姓名	成绩
表:		学生	公共课成绩
排序:			
显示:	☒	☒	☒
条件:	[请输入班级:]	[请输入姓名:]	
或:			



3.8 建立SQL查询与子查询

SQL (Structure Query Language) 称为结构化查询语言，被所有关系型数据库支持。它包括了数据定义 (Data Defintion)、数据查询 (Data Query)、数据操纵 (Data Manipulation)、数据控制 (Data Control)。

SQL语句在Access中的应用分为4个方面：联合查询、子查询、数据定义查询、传递查询。

这里我们只介绍前三个查询。



3.8.1 创建联合查询

联合查询将两个select语句的查询结果合成一个查询结果，是集合的并运算。在介绍联合查询之前，先看一下select语句的语法。

1. select语句的语法

select 字段列表 from 表名 [where 查询条件]

[order by 排序项] [group by 分组项]

例：select * from 学生 where 性别="男“

例：select 姓名,成绩 from 学生,公共课成绩

where 学生.学号=公共课成绩.学号

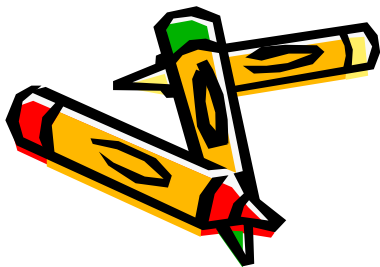
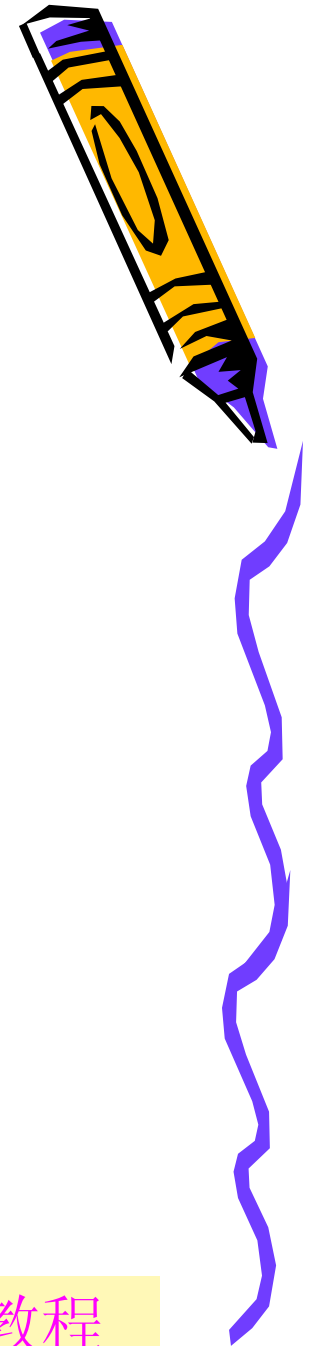


例: select 姓名,成绩 from 学生,公共课成绩
where 学生.学号=公共课成绩.学号
order by成绩 desc

例: select left([学号],4) as 班级,姓名 from 学生

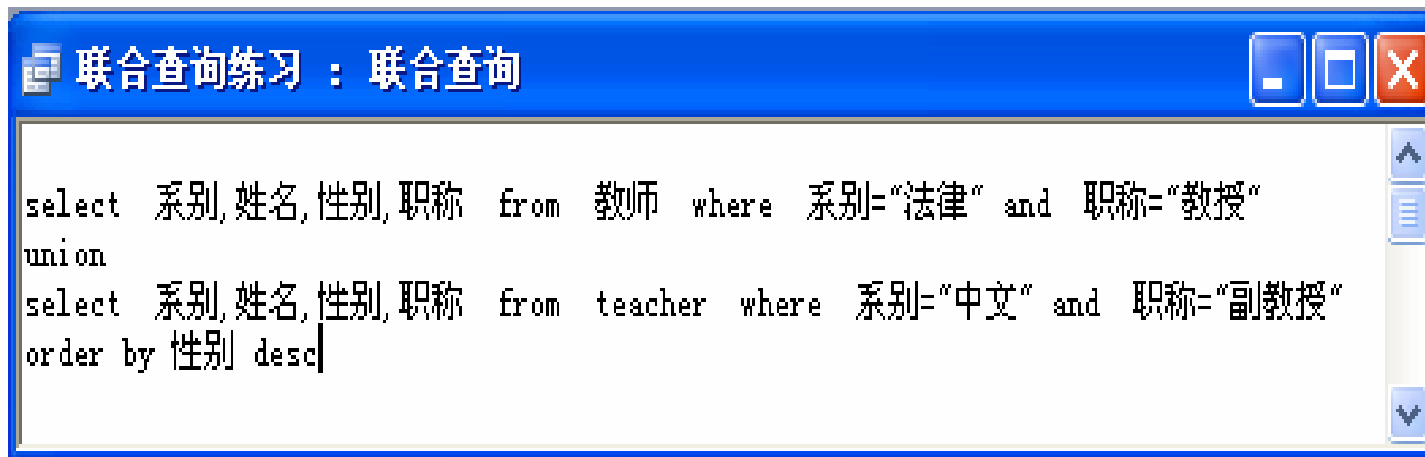
2. 联合查询的语法

```
select 语句1  
union [all]  
select 语句2
```

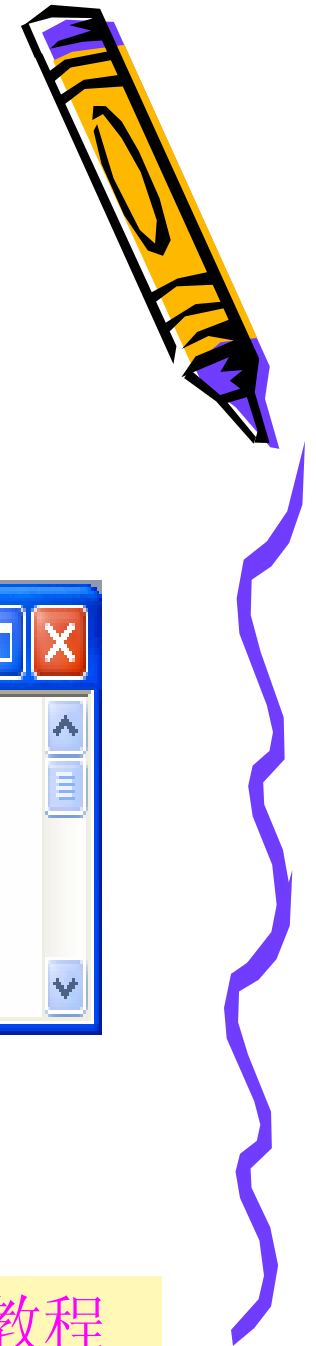


例：教师表法律系的教授与teacher表中文系的副教授
显示在一个查询结果中

- 1, “查询” 菜单 → “SQL特定查询” → “联合”
- 2, 写SQL命令



```
select 系别,姓名,性别,职称 from 教师 where 系别="法律" and 职称="教授"  
union  
select 系别,姓名,性别,职称 from teacher where 系别="中文" and 职称="副教授"  
order by 性别 desc
```



3.8.2 创建子查询

在select语句使用统计函数返回一个值，然后用这个值当条件建立选择查询，称为子查询。子查询不能单独作为一个查询，必须与其他查询相结合。做子查询的select语句只能返回一个值，而不是一组值。

1. 在select语句使用统计函数

- (1) count, 统计个数。
- (2) sum, 求和。
- (3) avg, 求平均值。
- (4) max, 求最大值。
- (5) min, 求最小值。

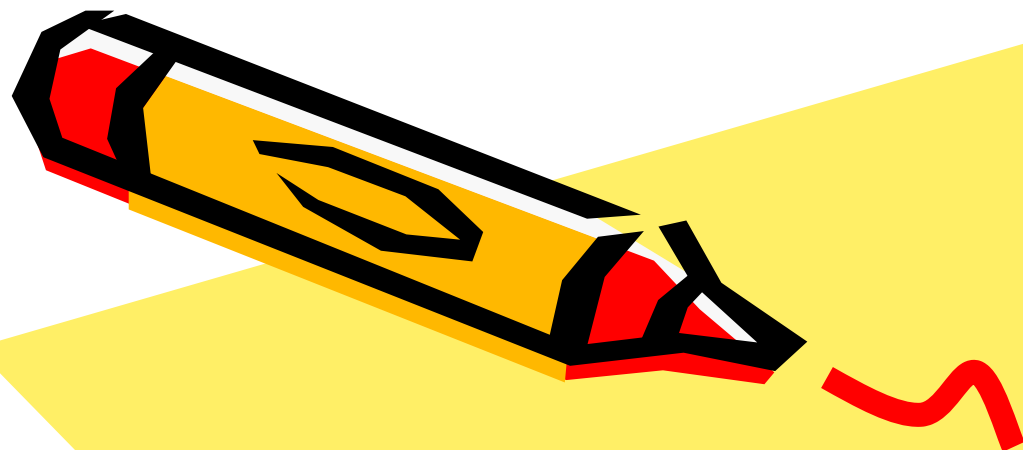


2. 建立子查询

例：显示奖金小于平均奖金的记录

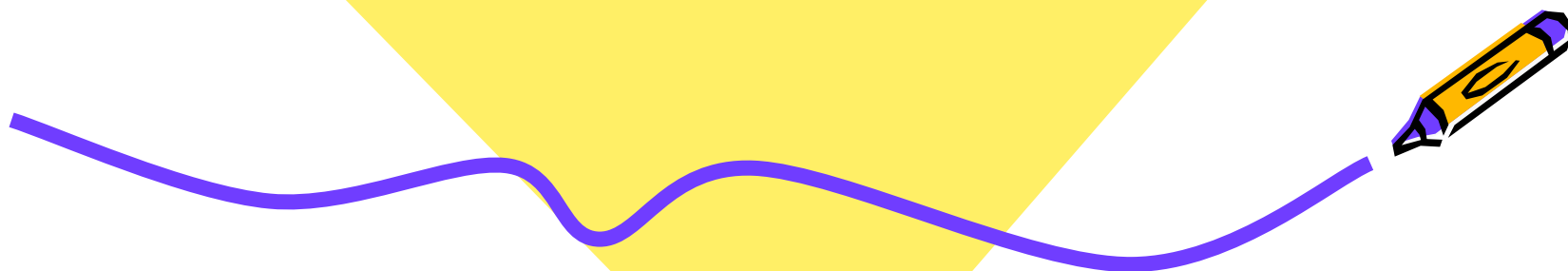
字段:	姓名	奖金	
表:	教师	工资	
排序:			
显示:	☒	☒	☐
条件:		<(select avg(奖金) from 工资)	
或:			





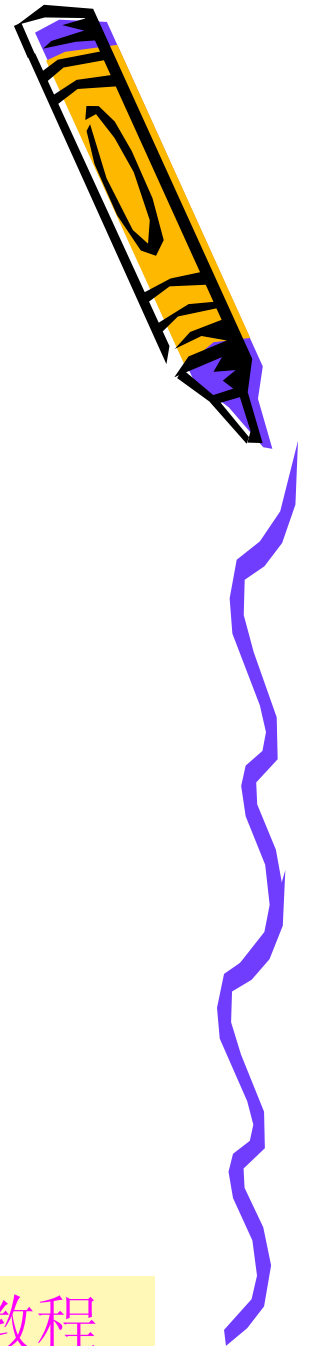
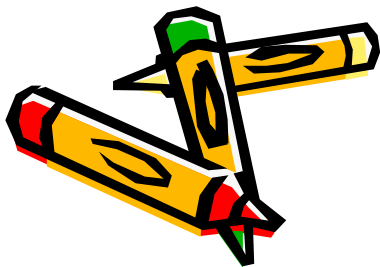
ACCESS实例教程

——窗体的操作（第4章）



本章要点:

- 1, 使用窗体控件
- 2, 建立窗体
- 3, 建立主/子窗体
- 4, 添加计算控件



4.1 认识窗体对象

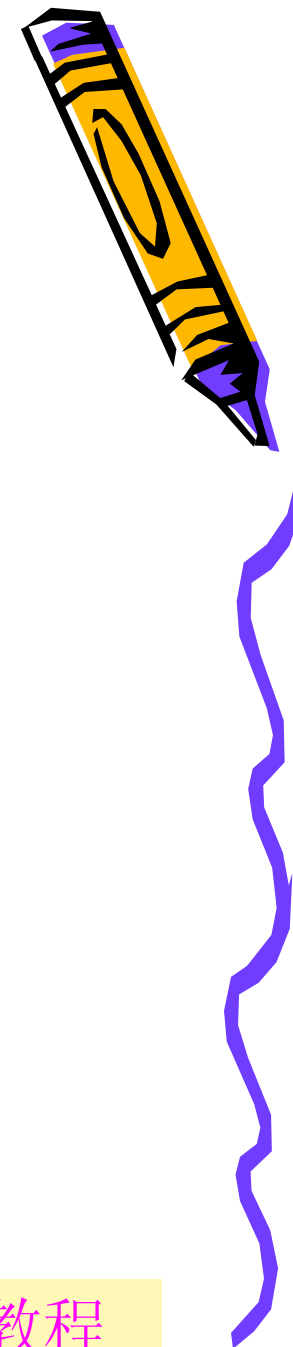
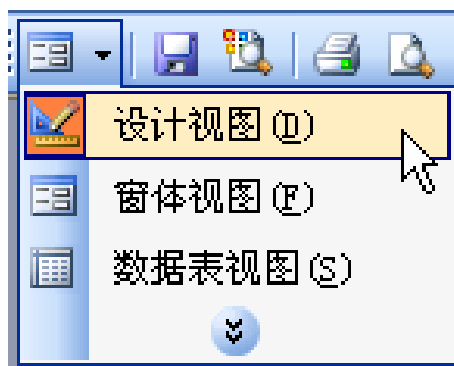
窗体对象是Access提供的最主要的操作界面对象，数据源是表或查询。窗体的主要作用是构造方便美观的输入输出界面，接收用户输入的命令，查看、编辑和追加数据。

窗体可以使数据的显示和操作按设计者的意愿实现，增加了数据操作的安全性和便捷性。数据库应用系统的用户对数据的任何操作只能在窗体中进行。



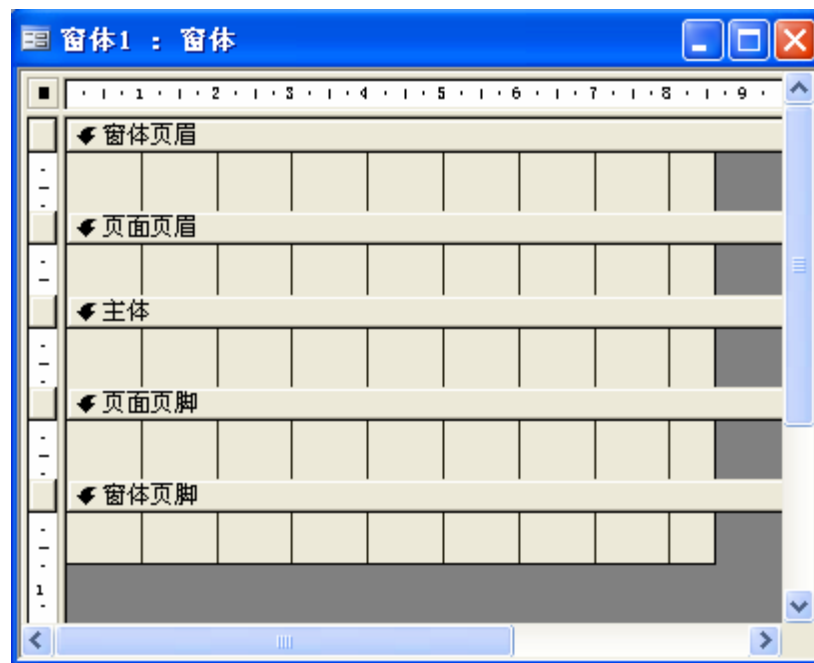
4.1.1 窗体的视图

最常用窗体视图有3种：设计视图、窗体视图、数据表视图。单击窗口左上角的视图按钮进行视图切换。



4.1.2 窗体的结构

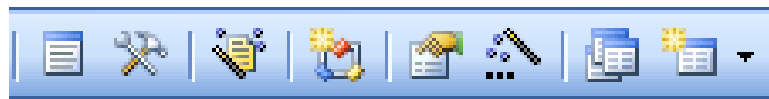
一个完整的窗体对象包含5个节，分别是：窗体页眉节、页面页眉节、主体节、页面页脚节、窗体页脚节。默认情况下窗体设计视图窗口只有主体节。



4.1.3 窗体的类型

窗体类型实际上是窗体布局，共有6种，分别是：纵栏式窗体、表格式窗体、数据表窗体、主/子窗体、图表窗体、数据透视表窗体。

4.1.4 窗体的几个重要按钮



分别是：字段列表，工具箱，自动套用格式，代码，属性，生成器，数据库，新对象



4.2 窗体的工具箱与属性对话框

窗体是一个容器对象，可以包含其他对象，包含的对象称为控件。在窗体设计视图中，Access提供了一个工具箱，用来生成窗体的常用控件，进行可视化的窗体设计。Access还提供了一个属性对话框，用来设置窗体本身和窗体内各控件的一系列属性。

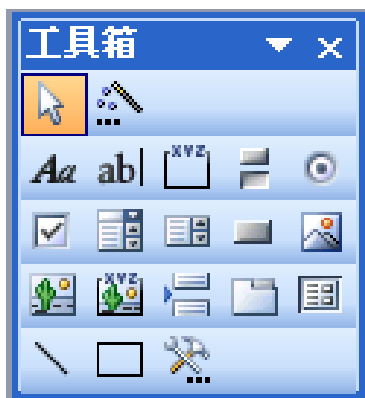
使用工具箱和属性对话框是可视化设计中最基本的操作。



4.2.1 工具箱的使用方法 with 按钮功能

工具箱与窗体设计界面是一体的，隐藏窗体的设计界面，工具箱就不可见。

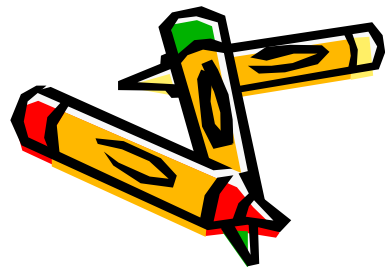
单击一个工具箱按钮，在设计视图中用鼠标画出一块区域，即生成与按钮对应的窗体控件。双击一个按钮将其锁定，在设计窗口重复画出多个相同控件，再次单击该按钮或用Esc键解除锁定。

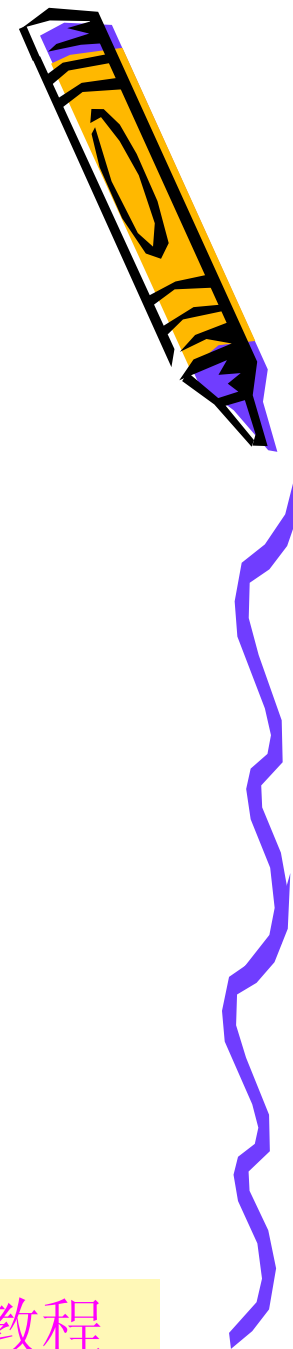


4.2.2 属性对话框

属性决定窗体和控件的结构、外观以及数据特性。任何对象都有一系列属性，不同对象拥有的属性也不相同。

先选取对象，然后再给选取的对象定义属性。





- 1, 格式属性卡, 定义控件的显示格式和外观样式。
如: 前景色、背景色、宽度、高度、上边界、左边界等。
- 2, 数据属性卡, 定义控件的数据来源和操作数据的规则。
如: 控件来源、输入掩码、有效性规则、默认值等。
- 3, 事件属性卡中显示当前控件所能识别的动作,
如: 单击、双击等。
- 4, 其他属性卡, 定义控件的附加属性。
如: 名称。控件的名称是一个字符串, 在VBA程序用名称调用控件。窗体中每一个控件都有名称, 名称是惟一的。



4.2.3 窗体的常用属性

窗体的常用属性有：标题、记录选择器、导航按钮、分隔线、图片、记录源等。

4.2.4 窗体的自动套用格式

在设计视图中打开一个窗体→“格式”菜单→“自动套用格式”→在列表选取一个样式→单击“确定”。

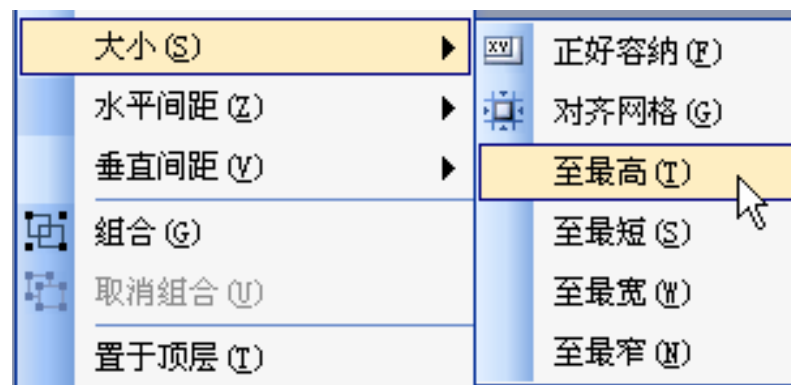


4.2.5 控件布局

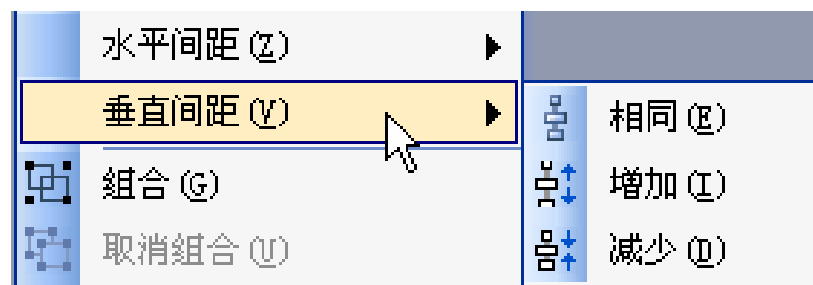
1, 对齐多个控件



2, 使多个控件大小相同



3, 设置多个控件的水平间距或垂直间距



设置多个控件的水平间距或垂直间距还可以用各控件的左边距、上边距、宽度、高度属性联合设置。

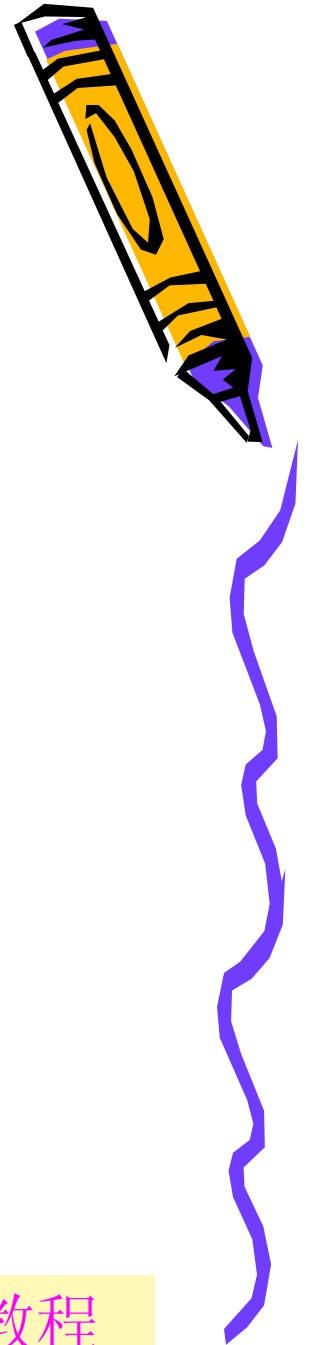
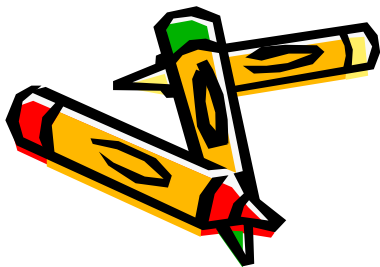
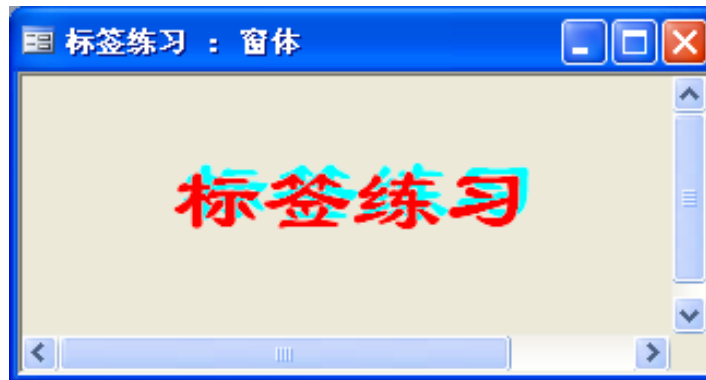


4.3 创建窗体控件

4.3.1 标签控件

用来显示说明性文字，没有控件源，文字不随记录的更换而改变。系统会为其他控件自动附加一个标签，也可以用标签按钮创建独立标签。

例：制作立体字



4.3.2 文本框控件

是交互式控件，用来输入、编辑和显示数据，有结合、非结合、计算三种类型。

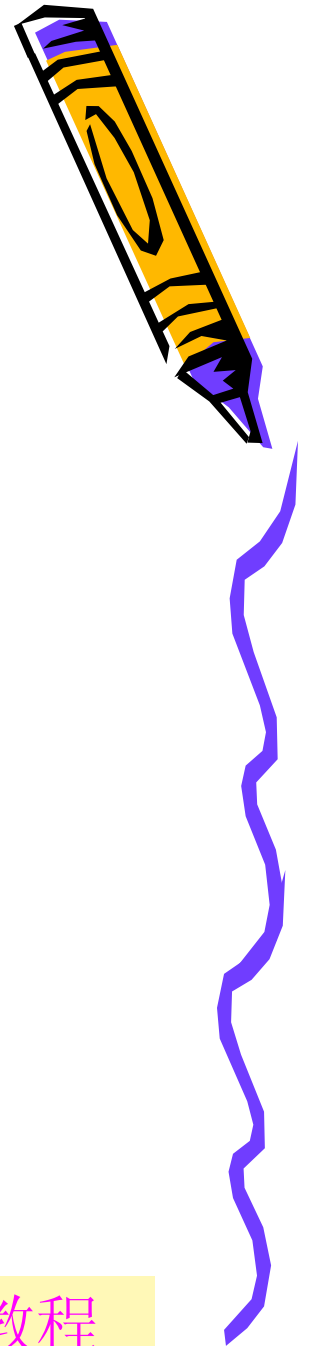
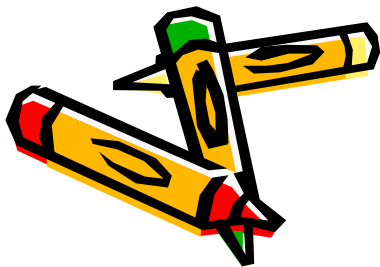
结合型文本框有控件来源，与表或查询中的字段相连。非结合型文本框没有控件来源，用来显示提示信息、接受用户输入的数据。计算型文本框用来显示表达式的值。表达式使用的数据可以是字段的值、其他控件的值、或函数值。计算型文本框内容从等号开始。



例：文本框练习

- (1) 从字段列表向窗体拖入姓名、系别、工资、奖金4个字段，生成了4个结合型文本框。
- (2) 建3个文本框→起名t1、t2、t3。
- (3) t1中写表达式：=[工资]+[奖金]
- (4) t3中写表达式：=t1-t2

	1	2	3	4	5	6	7	8	9
主体									
1	姓名:	姓名	请输入捐款:						
2	系别:	系别	未绑定						
3	工资:	工资							
4	奖金:	奖金	实发:						
5	应发:	=[工资]+[奖金]	=[t1]-[t2]						



转到窗体视图→在t2中输入200，回车后显示结果：



姓名:	马淑芬	请输入捐款:	
系别:	英语		200
工资:	2100		
奖金:	157	实发:	
应发:	2257		2057

记录: 1 共有记录数: 15



4.3.3 选项组控件

显示一组选项值，用户只需单击所需值即可完成数据输入。选项组中的值一次只能选择一个。可将选项组控件与一个字段绑定。

例：用选项组控件显示“党员否”字段的值

(1) 按下工具箱中的“控件向导”按钮→单击“选项组”按钮→在窗体中画一个矩形→在打开的对话框中依次输入“党员”和“非党员”。

请为每个选项指定标签：

	标签名称：
	党员
	非党员
	



(2) 单击“下一步”→选“不需要默认选项”→单击“下一步”→给“党员”对应的值填-1→给“非党员”对应的值填0

请为每个选项赋值:

	标签名:	值:
	党员	-1
	非党员	0

(3) 单击“下一步”→选“在此字段中保存该值”→单击旁边输入框的向下按钮选“党员否”字段。

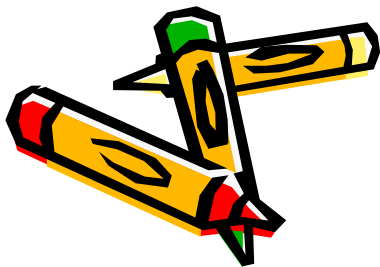
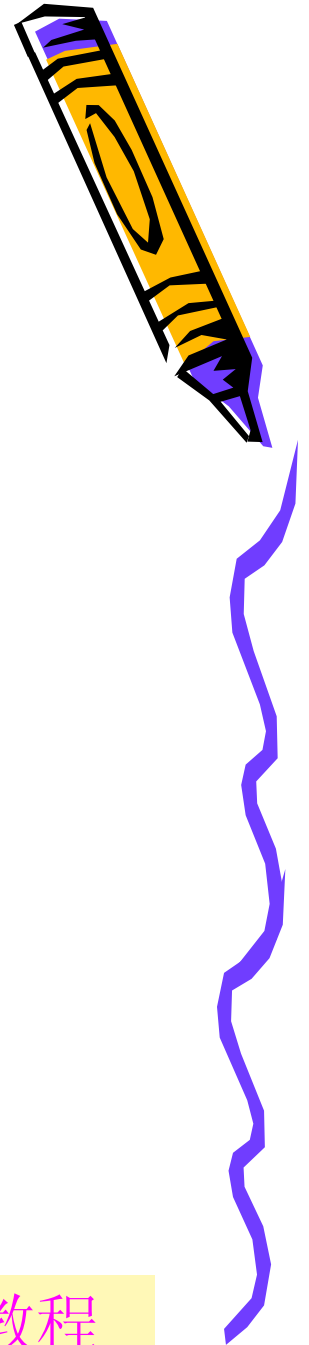
☐ 为稍后使用保存这个值 (S)

☒ 在此字段中保存该值 (V):



(4) 单击“下一步”→选按钮类型为“选项按钮”→“下一步”→为选项组指定标题为“是否党员”→单击“完成”。

(5) 整理选项组各控件位置→转到窗体视图，当前选项随记录的改变而不断变化。



4.3.4 复选框、切换按钮、选项按钮控件

复选框、切换按钮和选项按钮都显示“是/否”型数据的值。复选框中有对勾为“是”，无对勾为“否”。切换按钮按下状态为“是”，抬起状态为“否”。选项按钮中有圆点为“是”，无圆点为“否”。

例：将三种控件的控件来源设置为“党员否”字段。



4.3.5 组合框与列表框控件

组合框和列表框都提供一组值供用户选择，通过点击鼠标完成数据输入。

组合框只显示一行数据，在窗体中占的区域较小，既能进行选择，又能输入文本。列表框显示多行数据，在窗体中占的区域较大，只能进行选择，不能输入新值。组合框和列表框都分为结合型与非结合型2种。

例：用组合框显示“职称”字段，用列表框显示“系别”字段

(1) 按下“控件向导”按钮→单击“组合框”按钮→在窗体中画一个矩形→在打开的对话框中选“自行键入所需值”。



(2) 单击“下一步”→依次输入“教授”、“副教授”、“讲师”、“助教”。

列数(C):

	第 1 列
	教授
	副教授
	讲师
	助教
*	

(3) 单击“下一步”→选“将该数值保存在这个字段中”
→在右边的输入框中选“职称”字段。

☐ 记忆该数值供以后使用(R)

☒ 将该数值保存在这个字段中(S)



(3) 单击“下一步”→为组合框指定标签为“职称”→单击“完成”。

(4) 类似方法建立列表框→自行键入的值依次为：计算机、英语、中文、法律→将列表框数值保存在“系别”字段中。

(5) 整理各控件位置→转到窗体视图。显示：



姓名	系别	职称
江小洋	英语	教授



4.3.6 命令按钮

用来执行特定的操作，例如，单击命令按钮关闭窗体。
通常给命令按钮附加一段VBA代码或一个宏，也可以用“向导”给命令按钮指定一些特殊操作。

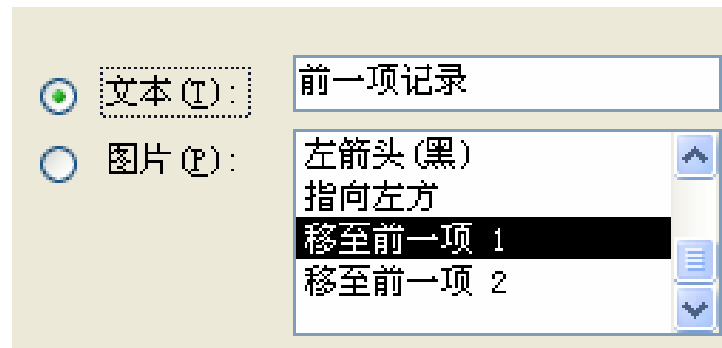
例：用向导建立命令按钮

(1) 按下“控件向导”→单击“命令按钮”控件→在窗体中画按钮大小→在对话框中选类别为“记录导航”→选操作为“转至前一项记录”。

类别 (C):	操作 (A):
记录导航	查找下一项
记录操作	查找记录
窗体操作	转至下一项记录
报表操作	转至前一项记录
应用程序	转至最后一项记录
杂项	转至第一项记录



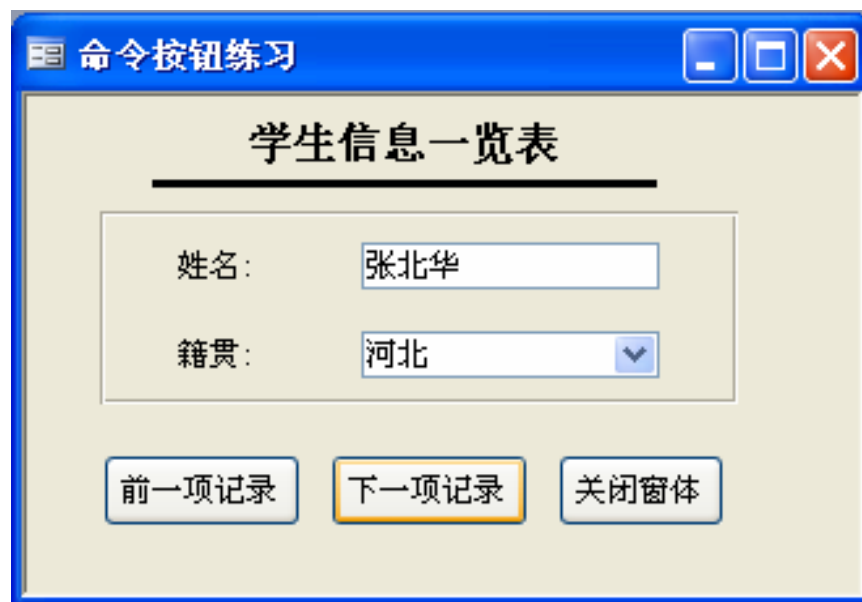
(2) 单击“下一步”→单击“文本”项→单击“下一步”→单击“完成”按钮。



(3) 类似方法建“转至下一项记录”按钮和“关闭窗口体”按钮。



(4) 转到窗体视图。显示：



命令按钮练习

学生信息一览表

姓名: 张北华

籍贯: 河北

前一项记录 下一项记录 关闭窗体



4.3.7 选项卡控件与其他控件

选项卡用来在一个窗体内显示多页信息，单击选项卡的标签进行页面切换。

例：建立选项卡

(1) 选取“选项卡控件”在窗体单击，窗体自动生成有2个页的选项卡。

(2) 右击第2个卡→在快捷菜单中选“插入页”→同样方法再插入一页，插入的页依次排在最后。

(3) 将4个页的“标题”分别设置为：基本信息、照片、成绩、日历→名称分别为：p1、p2、p3、p4。



- (4) 单击“基本信息”页→拖入学号、姓名、性别字段。
- (5) 单击“照片”页→在页中建立“绑定对象框”→“控件来源”属性选“照片”字段。
- (6) 单击“成绩”页→将数据库窗口公共课成绩表拖入页中。
- (7) 单击“日历”页→单击工具箱“其他控件”按钮→选“日历控件”→在页中画一个矩形作为日历大小。

主体

基本信息 照片 成绩 日历

2008 九月 九月 2008

星期日	星期一	星期二	星期三	星期四	星期五	星期六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11



(8) 转到窗体视图，显示：



The screenshot shows an Access form window titled '选项卡练习'. It has four tabs: '基本信息', '照片', '成绩', and '日历'. The '成绩' tab is active, displaying a table with three columns: '学号' (Student ID), '课程号' (Course ID), and '成绩' (Grade). The table contains three rows of data. Below the table is a record navigation bar with buttons for first, previous, next, last, and a search icon, along with a text box for the current record number and the total number of records. The bottom of the form shows a similar navigation bar for the entire dataset, indicating 10 total records.

学号	课程号	成绩
080101	101	60
080101	102	87
080101		

记录: [Navigation Buttons] 共有记录数: 2

记录: [Navigation Buttons] 1 共有记录数: 10



4.3.8 子窗体/子报表控件

如果2个表建立了一对多关系，并且2个表都建了窗体，可以用“子窗体/子报表”控件将子表窗体显示在主表窗体中。子表窗体通常是数据表窗体，利用主/子窗体使主表与子表数据同时显示，两表数据之间的关系一目了然。

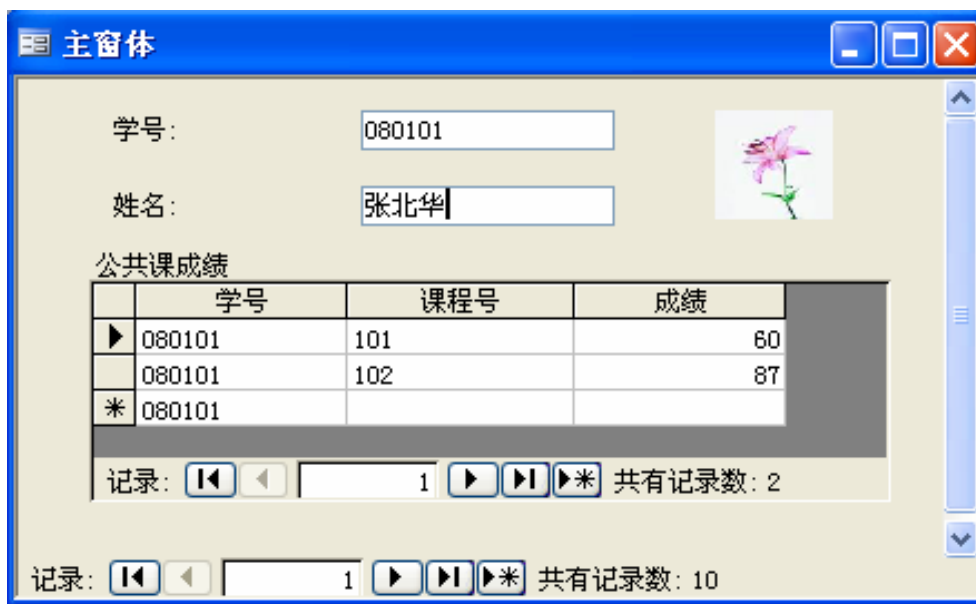
例：用主/子窗体方法显示学生表与公共课成绩表

(1) 将“学生”表与“公共课成绩”表建立关系并实施参照完整性，分别给两个表建纵栏式窗体和数据表式窗体。



(2) 在设计视图中打开主窗体→将子窗体拖入主窗体中→整理布局。

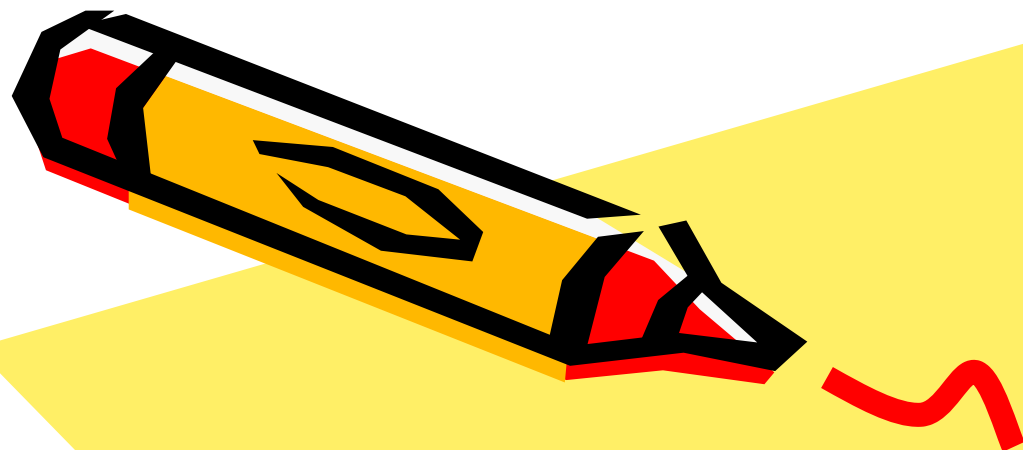
(3) 转到窗体视图，显示主/子窗体。



学号	课程号	成绩
080101	101	60
080101	102	87

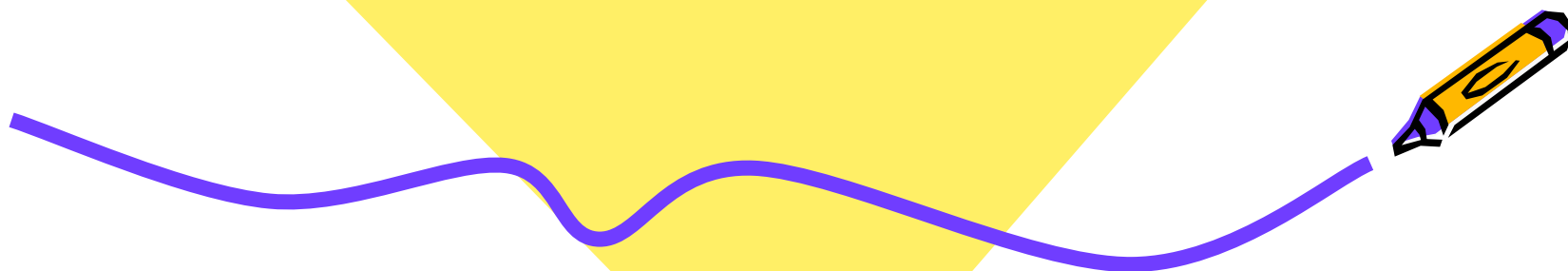


ACCESS实例教程



ACCESS实例教程

——报表的操作（第5章）



本章要点:

- 1, 使用报表控件
- 2, 建立报表
- 3, 建立主/子报表
- 4, 添加计算控件



5.1 认识报表对象

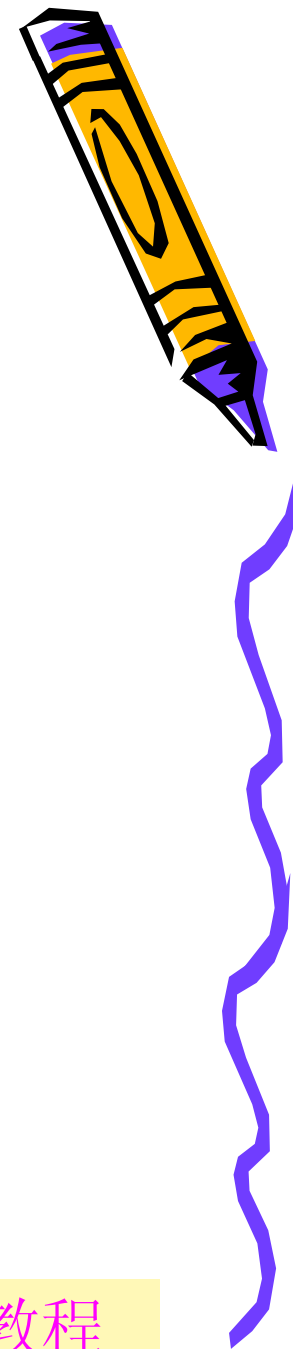
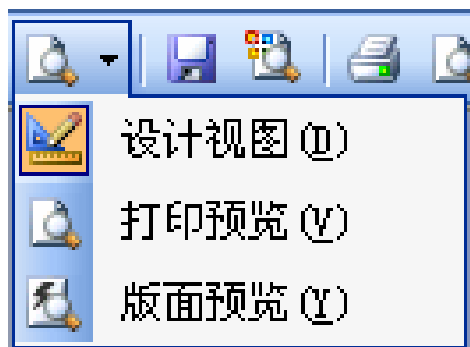
报表是按指定格式输出数据的对象，数据源是表、查询或SQL语句。报表主要用于数据库数据的打印，没有输入数据的功能。

报表是容器对象，包含数据源和其他对象，在报表中的对象称为报表控件。



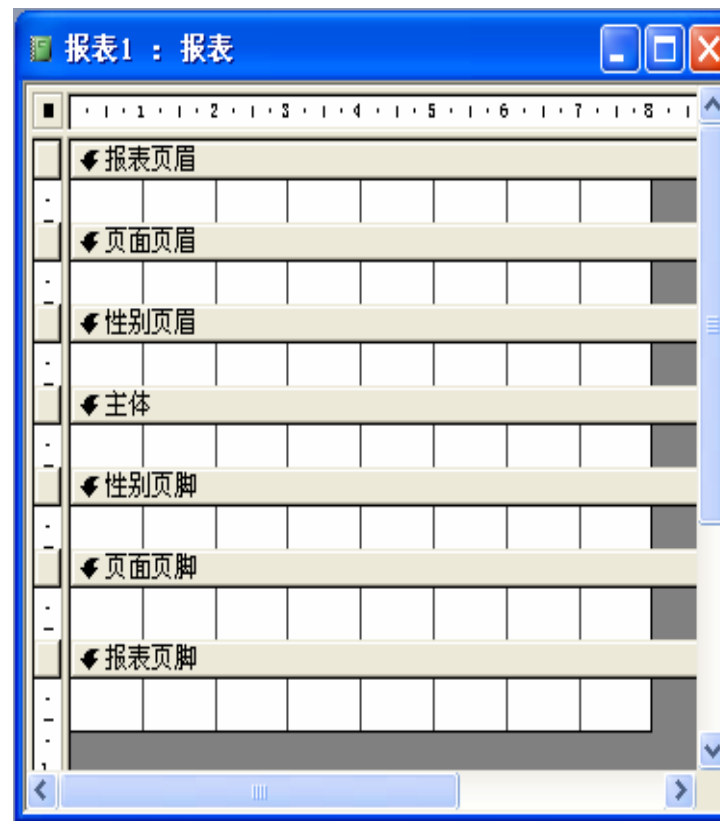
5.1.1 报表的视图

报表有3种视图：设计视图、打印预览、版面预览。单击窗口左上角的视图按钮进行视图切换。



5.1.2 报表的结构

一个完整的报表对象包含7个节：报表页眉节、页面页眉节、组页眉节、主体节、组页脚节、页面页脚节、报表页脚节。默认情况下设计视图窗口只显示页面页眉节、主体节、页面页脚节。只有在“视图”菜单的“排序与分组”中设置，才能显示组页眉节和组页脚节。



5.1.3 报表的类型

报表有4种类型，分别是：纵栏式报表、表格式报表、图表报表、标签报表。

纵栏式报表以垂直方式显示记录，字段标签与字段值一起显示在主体节内。

表格式报表以行、列形式显示记录，一页显示多条记录，字段标签不在主体节区域，而是在页面页眉节中。分组字段在表格式报表中设置。

图表报表用图表方式显示数据，直观地显示数据之间的关系。

标签报表是特殊类型的报表，将数据做成标签形式，一页中显示许多标签。



5.1.4 报表的几个重要按钮



分别是：字段列表，工具箱，排序分组，自动套用格式，代码，属性，生成器，数据库，新对象。

单击“排序分组”按钮，选取一个字段，可以对字段进行排序和分组操作，实现报表数据的排序、分组输出、分组统计。

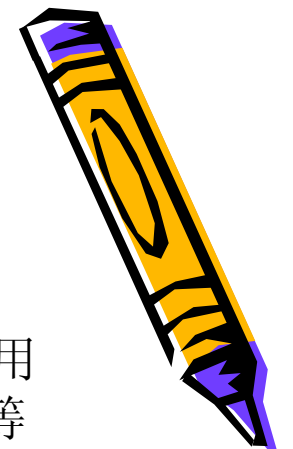


5.1.5 报表的节

报表的内容以节划分，节代表不同带区，每个节都有特定用途，并按一定顺序打印。报表有惟一宽度，改变一个节的宽度等于改变整个报表的宽度。

用“视图”菜单中的命令可以为报表添加“报表页眉/页脚”节、“页面页眉/页脚”节、“组页眉/页脚”节。

选中一个节，“可见性”属性设置为“否”，或删除节中控件，将节的“高度”属性设置为0，可以隐藏选中的节。



5.2 在报表中使用控件

5.2.1 用文本框控件显示页码

[page] 计算当前页， [pages]计算总页数。

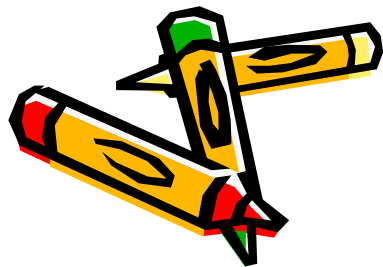
页码主要有2种显示格式：

(1) 显示格式为“当前页/总页数”， 如“3/10”

表达式：= [page] & "/" & [pages]

(2) 显示格式为“第n页/共m页”， 如“第3页/共10页”

表达式：= "第" & [page] & "页/总" & [pages] & "页"



5.2.2 用文本框控件在报表中添加新字段

例：在报表中添加“应发工资”字段

页面页眉中添加标签，主体节中添加计算文本框。

表达式：=[工资]+[奖金]



5.3.3 用复选框控件在报表中添加新字段

例：用复选框控件添加“通过否”字段

在主体节添加复选框控件，名称为fxk，“控件来源”属性写表达式：`=iif([人平均]>=70,true,false)`

在“录取否”对应的计算文本框表达式：

`=iif(fxk=-1,"录取","未录取")`



显示结果：



班级:	姓名:	成绩:	通过否:	录取否:
0801	李丁	65.5	☐	未录取
0801	王伟	68.5	☐	未录取
0801	张北华	73.5	☒	录取
0802	钱小红	80.5	☒	录取
0802	孙伟力	62.5	☐	未录取



5.2.4 在报表中显示非记录源字段

用 DLookup 函数在报表中显示非记录源（又称外部表）中的字段值，外部表与当前表之间无须建立关系，在函数中以共有字段作为连接条件即可。

DLookup（“外部表字段名”，“外部表名”，“条件表达式”）

说明：

- （1）函数中的各部分要用引号括起来。
- （2）条件表达式格式：外部表字段名=' "&当前表字段名& "'
注意其中单、双引号和&号的使用。
- （3）如果有多个字段符合条件表达式，DLookup 函数只返回第一个字段值。



例：当前表是“员工”表，外部表是“部门”表，用
DLookup 函数显示外部表中“名称”字段的值。

员工：表						
	编号	姓名	性别	年龄	职务	所属部门
▶	000001	李四	男	24	职员	04
	000002	张三	女	23	职员	04
	000003	程鑫	男	20	职员	03
	000004	刘红兵	男	25	主管	03
	000005	钟舒	女	35	经理	02
	000006	江滨	女	30	主管	04
	000007	王建钢	男	19	职员	01
	000008	璐娜	女	19	职员	04
	000009	李小红	女	23	职员	03
记录： 1 共有记录数：58						

部门：表		
	部门编号	名称
▶	01	生产部
	02	开发部
	03	人力部
	04	财务部
*		

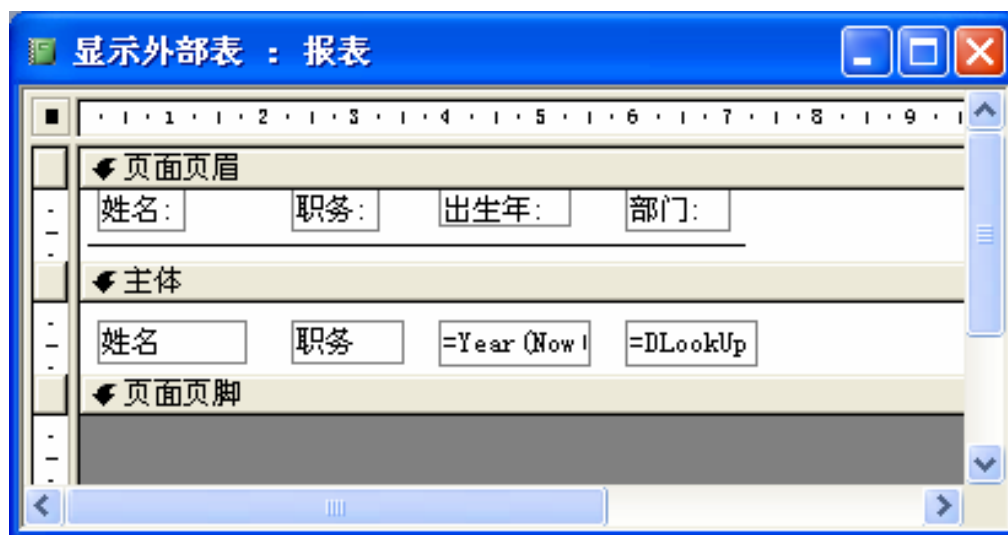


计算出生年：

=Year(Now())-[年龄]

计算部门：

=dlookup("名称","部门","部门编号=' "&所属部门&"' ")

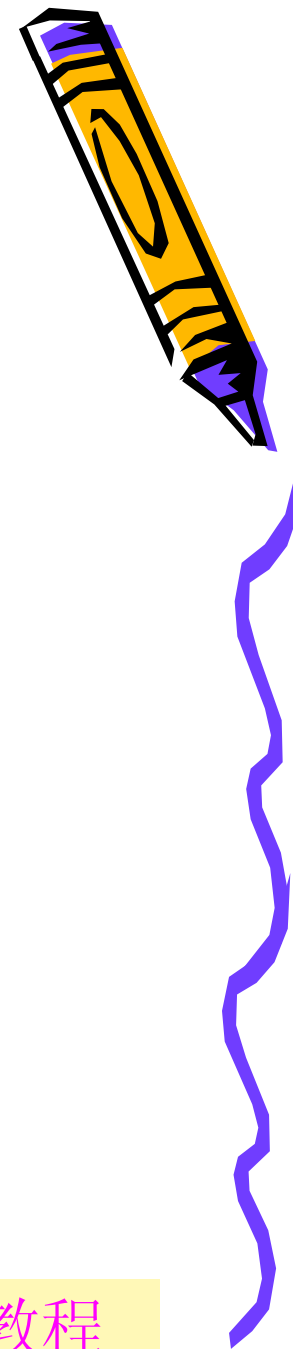


ACCESS实例教程

显示结果:



姓名:	职务:	出生年:	部门:
李四	职员	1984	财务部
张三	职员	1985	财务部
程鑫	职员	1988	人力部
刘红兵	主管	1983	人力部
钟舒	经理	1973	开发部



5.2.5 统计报表数据

在报表页脚节或组页脚节可以使用函数对整个报表或组做统计操作，常用的统计函数有：count（统计个数）、sum（求和）、avg（求平均值）。

例：用count、sum、avg函数统计报表数据

计算总人数：=count([姓名])

计算平均工资：=avg([工资])

计算奖金合计：=sum([奖金])

说明：“格式”属性选“固定”，值显示2位小数。



报表布局:

统计函数报表 : 报表

1 2 3 4 5 6 7 8 9

报表页眉

页面页眉

姓名: 系别: 工资: 奖金:

主体

姓名 系别 工资 奖金

页面页脚

报表页脚

总人数: =Count([姓名])

平均工资: =Avg([工资])

奖金合计: =Sum([奖金])



ACCESS实例教程

显示结果：

统计函数报表：报表			
张成	法律	4000	225
李鹏举	法律	3200	135
钟小于	法律	2600	157
总人数：		15	
平均工资：		2733.33	
奖金合计：		2266	



ACCESS实例教程

5.3 数据排序与分组

5.3.1 数据排序

“视图”菜单→“排序与分组”→在左边列中选字段→在右边列中选排序方式。如果对多个字段排序，则选取字段的顺序就是排序次序。首先对第一个字段排序，当第一个字段的值相同时，再对第二个字段排序。



字段/表达式	排序次序
▶ 班级	升序
人平均	降序

组属性

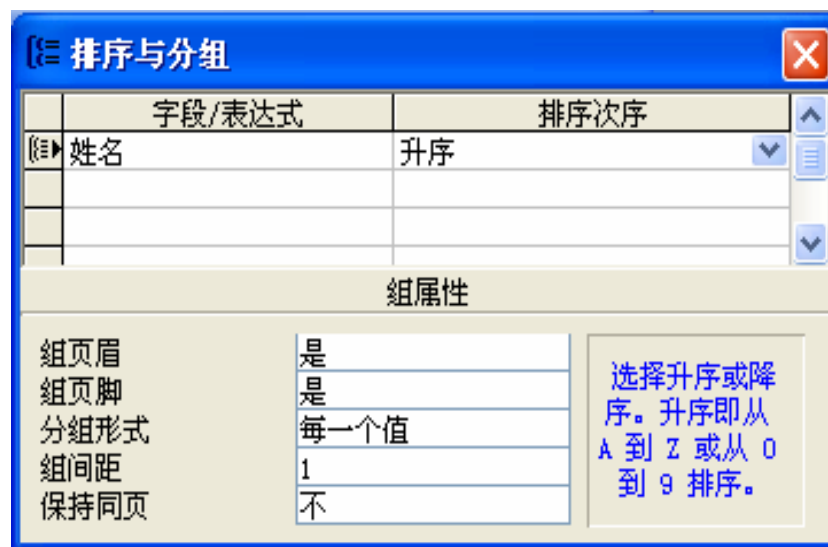
组页眉	否	选择一个字段或输入一个表达式以便进行排序或分组
组页脚	否	
分组形式	每一个值	
组间距	1	
保持同页	不	



5.3.2 数据分组

例：按照“姓名”字段分组，显示所选课程的名称、成绩，计算平均成绩。

“视图”菜单→“排序与分组”→分组字段选“姓名”→组页眉选“是”→组页脚选“是”，设计视图中显示组页眉和组页脚。



字段/表达式	排序次序
姓名	升序

组属性	
组页眉	是
组页脚	是
分组形式	每一个值
组间距	1
保持同页	不

选择升序或降序。升序即从A到Z或从0到9排序。



在组页脚添加计算字段。

表达式：=avg([成绩])

分组统计报表：报表

1 2 3 4 5 6 7 8 9 10

页面页眉		
姓名：	课程名称：	成绩：
姓名页眉		
姓名		
主体		
	课程名称	成绩
姓名页脚		
	平均成绩：	=Avg([成绩])
页面页脚		



ACCESS实例教程

显示结果：

分组统计报表：报表		
姓名：	课程名称：	成绩：
李超群		
	法律基础	58
	思想道德修养	92
	平均成绩：	75
李丁		
	马克思主义哲学	57
	思想道德修养	74
	平均成绩：	65.5



ACCESS实例教程

5.4 主/子报表与标签报表

5.4.1 主子报表

主/子报表类似于主/子窗体，是对建立了关系的两个表的操作。两个表都已单独建立了报表，然后将子表对应的报表插入到主表对应的报表中。主报表可以包含一个或多个子报表，也可以包含一个或多个子窗体。



例：用主/子报表的方法显示学生表与公共课成绩表

(1) 将“学生”表与“公共课成绩”表建立关系并实施参照完整性，并分别建立报表。

(2) 将子报表拖入主报表的设计视图，调整大小。

主报表：报表

页面页眉

学号: 姓名: 性别:

主体

学号 姓名 性别

1 2 3 4 5 6 7

1 2 3 4 5 6 7

学号 课程号 成绩

页面页脚



显示结果：

主报表：报表

学号：	姓名：	性别：
080102	李丁	男
080102	103	57
080102	105	74
080103	王伟	男
080103	103	69
080103	104	68

页： 1



ACCESS实例教程

5.4.2 标签报表

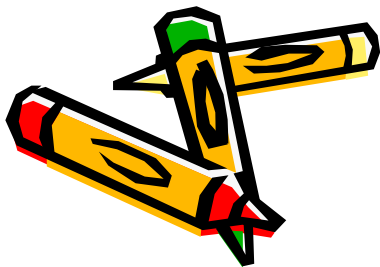
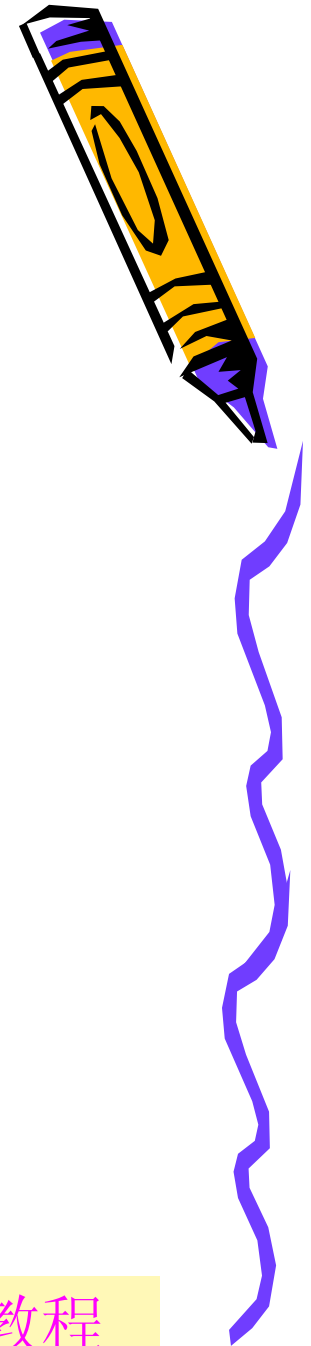
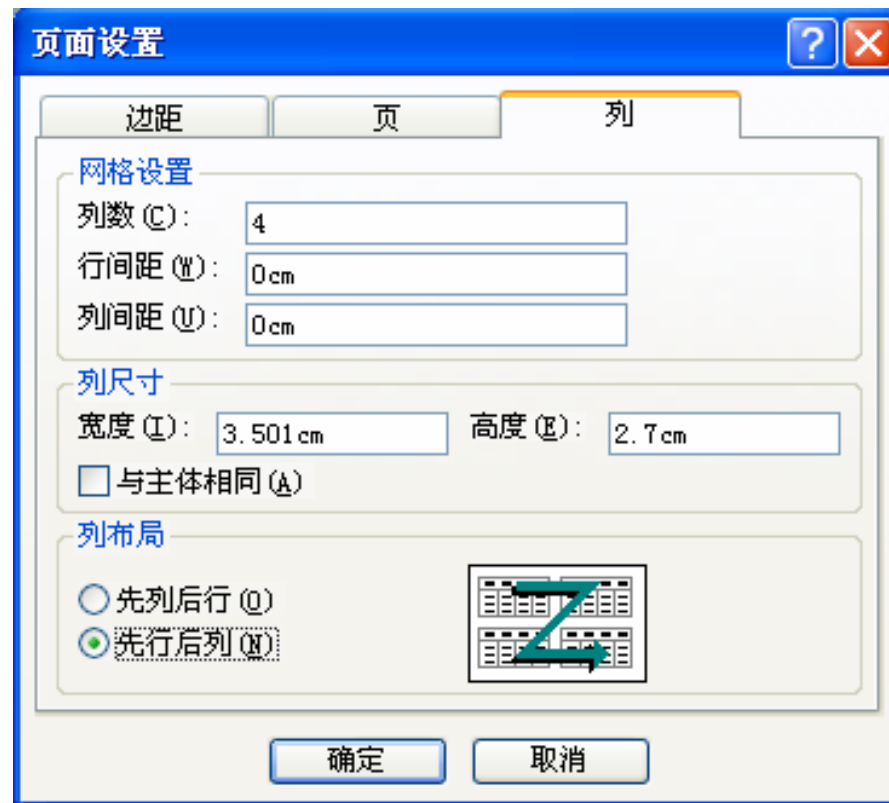
标签报表是一种多列报表，在一页中显示多列数据。

例：制作可以显示4列的标签报表。

(1) 添加字段，用矩形控件修饰。



“文件”菜单→“页面设置”→“列”选项卡中列数选4→行间距和列间距都选0→宽度3.5cm→高度2.7cm→去掉“与主体相同”的对勾。



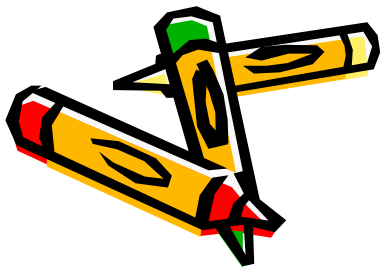
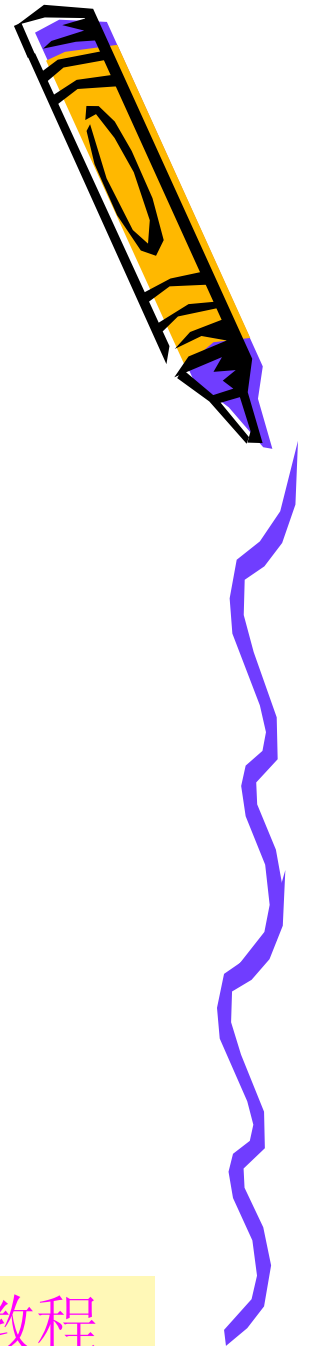
ACCESS实例教程

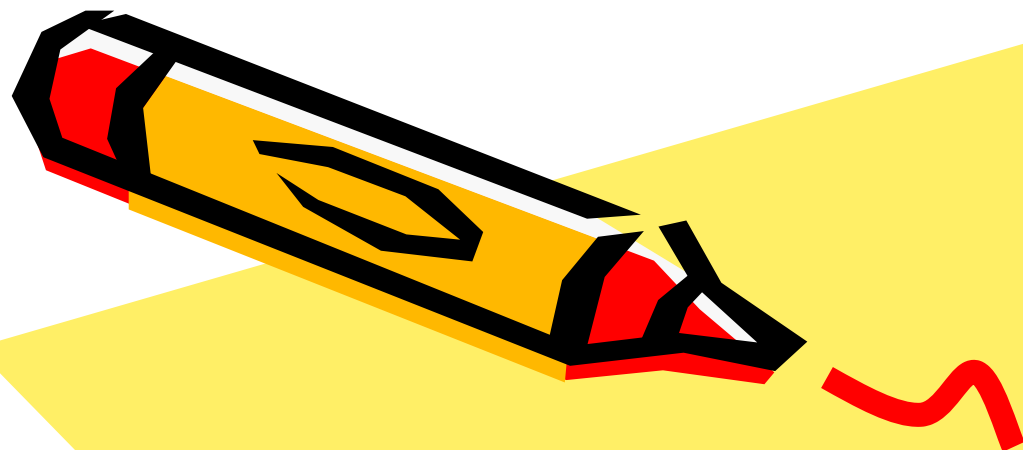
显示结果：

标签报表 : 报表

张宏 男 讲师 中文	刘立丰 女 副教授 中文	张晓芸 男 讲师 计算机	麻城风 男 副教授 英语
钟小于 女 讲师 法律	王成里 男 讲师 英语	李达成 男 副教授 计算机	李鹏举 男 副教授 法律

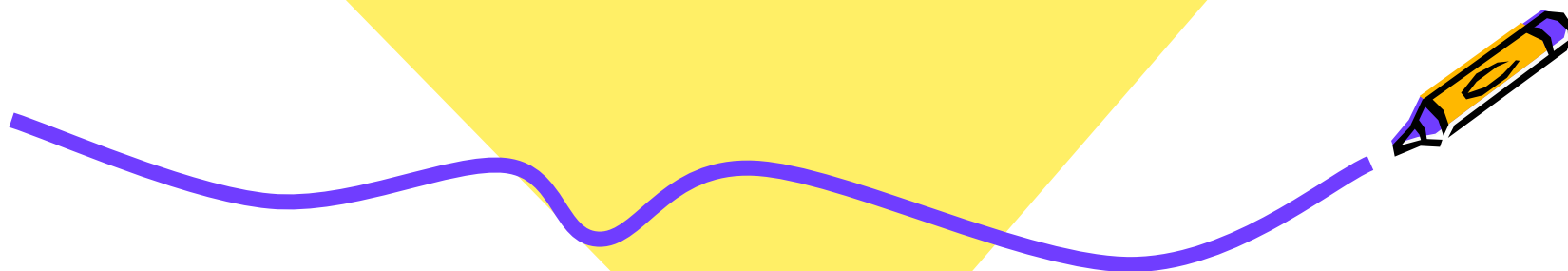
页: 1





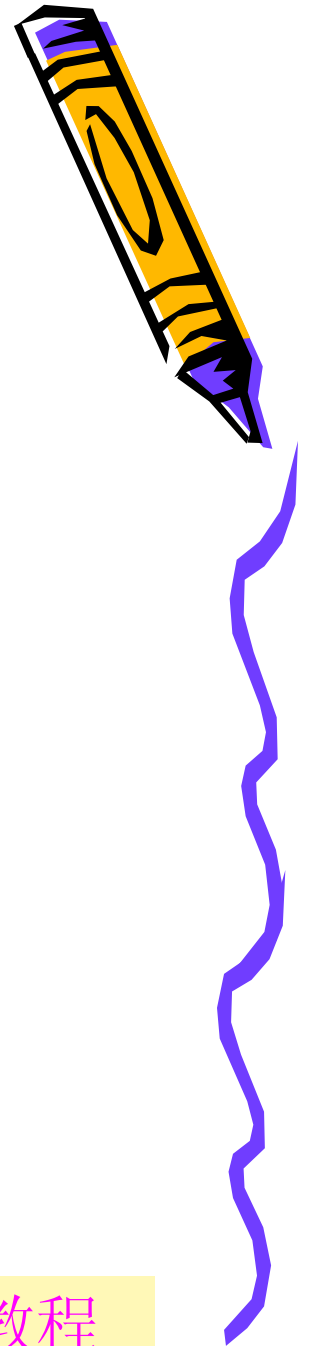
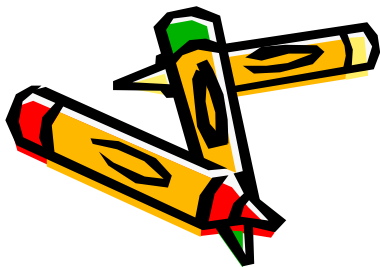
ACCESS实例教程

——页的操作（第6章）



本章要点:

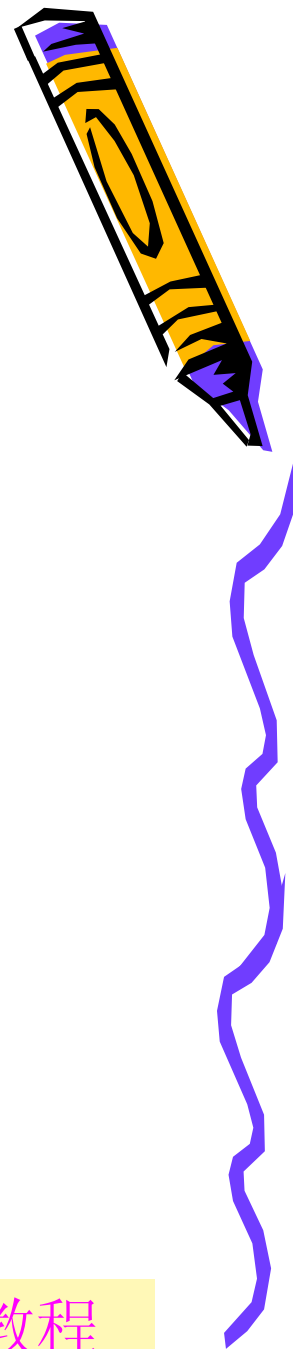
- 1, 使用页控件
- 2, 建立数据访问页
- 3, 编辑数据访问页



6.1 认识页对象

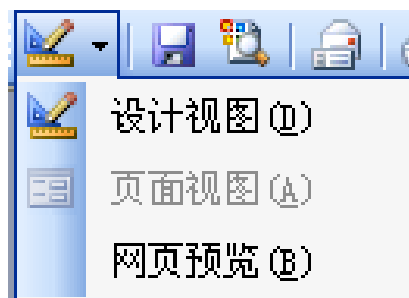
页对象是直接连接Access数据库数据的Web页，交互式的，可以通过网络对数据进行实时访问，并能查看、编辑、更新数据。所以，数据访问页具有网上数据库功能。

页在Access中设计，但页本身是存贮在Access之外的独立文件，文件扩展名为.htm或.html，数据库中保留的是指向该文件的快捷方式。正是由于数据访问页有这一特性，所以，使用数据访问页的用户不必安装Access也能访问数据库数据。

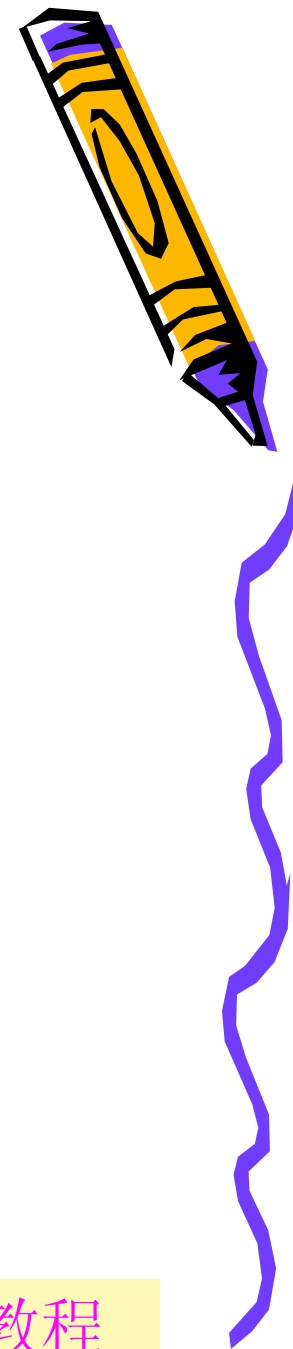


6.1.2 页的视图

页有3种视图：设计视图、页面视图、网页预览。



设计视图用来创建和编辑页结构，放置页的控件。页面视图用来查看页的输出效果。网页预览用浏览器查看页的设计，带有超链接的页只能用网页预览查看效果。

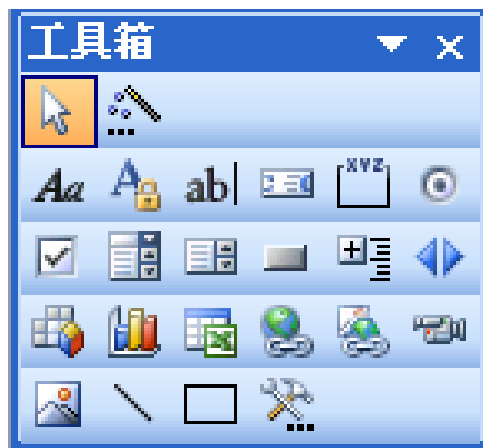


6.1.2 页的设计视图

页的设计视图是创建与编辑数据访问页的一个可视化集成界面，所有与字段有关的控件都要放在网格区域中。



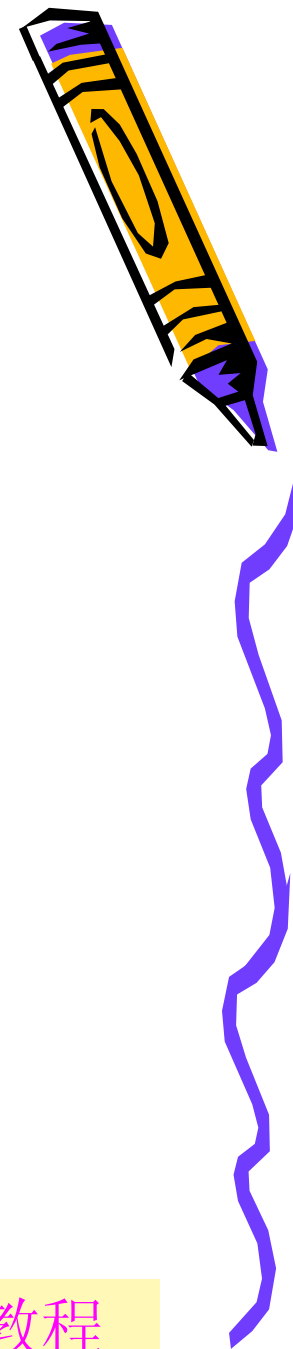
6.1.3 页的工具箱



页的工具箱有一些专用于网页的工具和专用于网上浏览数据的工具。



ACCESS实例教程



- (1) 滚动文字按钮，在页中插入一段移动文本。
- (2) 展开按钮，在页中插入一个展开/收缩按钮。
- (3) 记录浏览按钮，在页中插入一个工具栏，包含记录浏览的各种按钮。
- (4) 超链接按钮，在页中插入一个包含超链接地址的文本，单击文本打开超链接指向的目标端点。
- (5) 图像超链接按钮，在页中插入一个包含超链接地址的图片，单击图片打开超链接指向的目标端点。
- (6) 影片按钮，在页中插入一个影片控件，并指定影片。



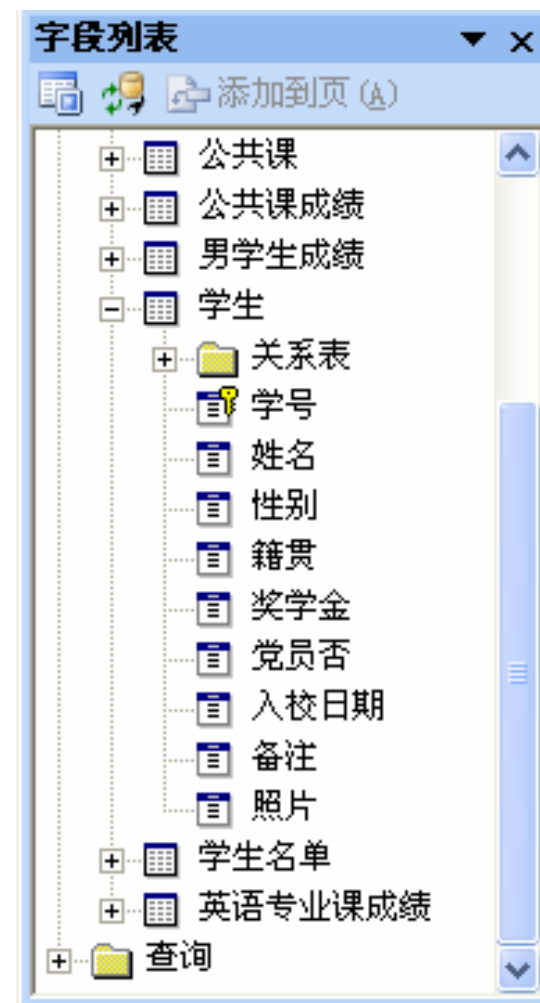
6.1.4 页的属性窗口

页的属性窗口与窗体、报表的属性窗口稍有不同，属性窗口中没有事件选项卡。



6.1.5 页的字段列表

页的字段列表框与窗体、报表的字段列表框也有所不同，页的字段列表框显示数据库中所有表和查询，当前数据源处于展开状态，显示所有字段。其他数据源则处于折叠状态。



例：建立一个数据访问页

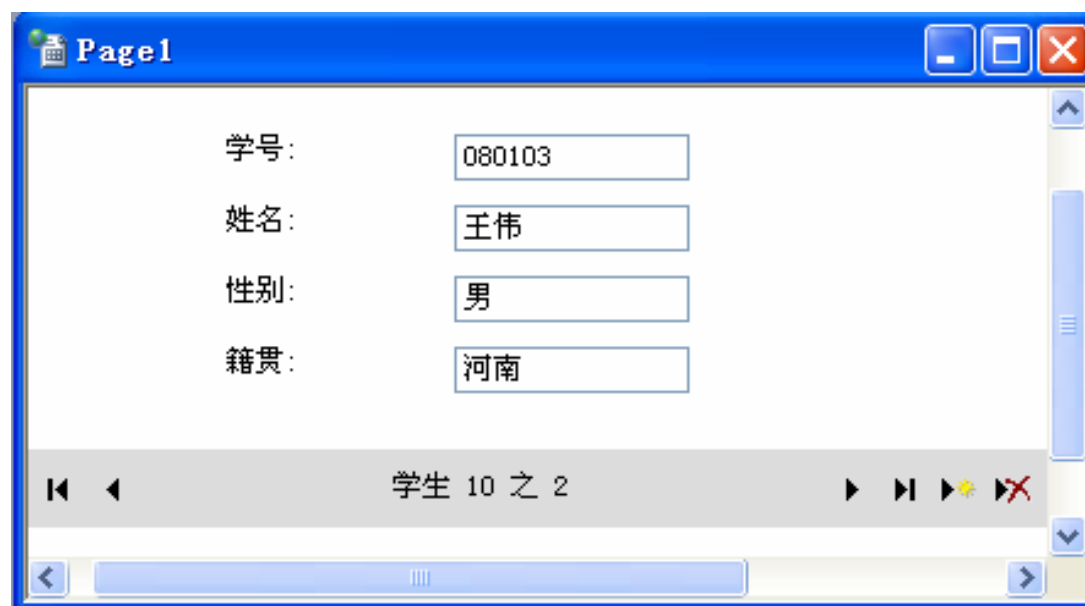
(1) 在库窗口中单击页对象→单击“新建”按钮→选“设计视图”→数据源选“学生”表→单击“确定”按钮。

(2) 从字段列表框向设计窗口拖入4个字段：学号、姓名、性别、籍贯→将控件纵向排整齐。

页眉：学生 ▼					
		学号：			
		姓名：			
		性别：			
		籍贯：			
导航：学生					



(3) 转到“页面视图”查看显示结果。



Page1

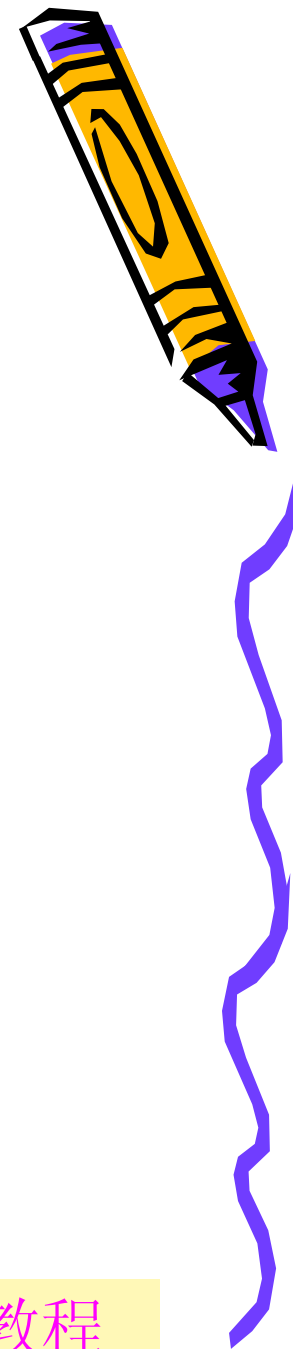
学号: 080103

姓名: 王伟

性别: 男

籍贯: 河南

学生 10 之 2



6.2 在页中使用控件

6.2.1 添加滚动文字

单击工具箱的“滚动文字”控件→在窗体中画一个矩形
→写文字→在常用工具栏中设字体、字号。



The screenshot shows a Microsoft Access form titled "Page1 : 数据访问页". The form has a scrollable text area at the top containing the text "欢迎光临!". Below this, there is a section titled "页眉: 学生" (Page Header: Student) which contains a table with four rows and two columns. The first column contains labels: "学号:" (Student ID), "姓名:" (Name), "性别:" (Gender), and "籍贯:" (Hometown). The second column contains empty text boxes for data entry. At the bottom of the form, there is a section titled "导航: 学生" (Navigation: Student) which also contains a table with two columns and one row, with the first column containing the label "学号:" and the second column containing an empty text box.



6.2.2 添加命令按钮

与字段有关的按钮添加在网格区域，添加方法与窗体中的命令按钮相同。



The screenshot shows a Microsoft Access form titled "Page1 : 数据访问页". The form has a header section labeled "页眉: 学生" and a footer section labeled "导航: 学生". The main content area is a grid with four rows and four columns. The first column contains labels: "学号:", "姓名:", "性别:", and "籍贯:". The second column contains empty text boxes. The third and fourth columns are empty. Below the grid, there are four command buttons, each with a hand cursor icon. The form also has a scroll bar on the right and a navigation bar at the bottom.



6.3 页的修饰

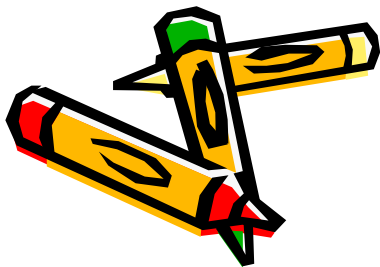
6.3.1 添加背景色

选中一个区域→“格式”菜单→“背景”→“颜色”→在显示的颜色盒中选取一种颜色，该区域显示指定颜色。



6.3.2 添加背景图片

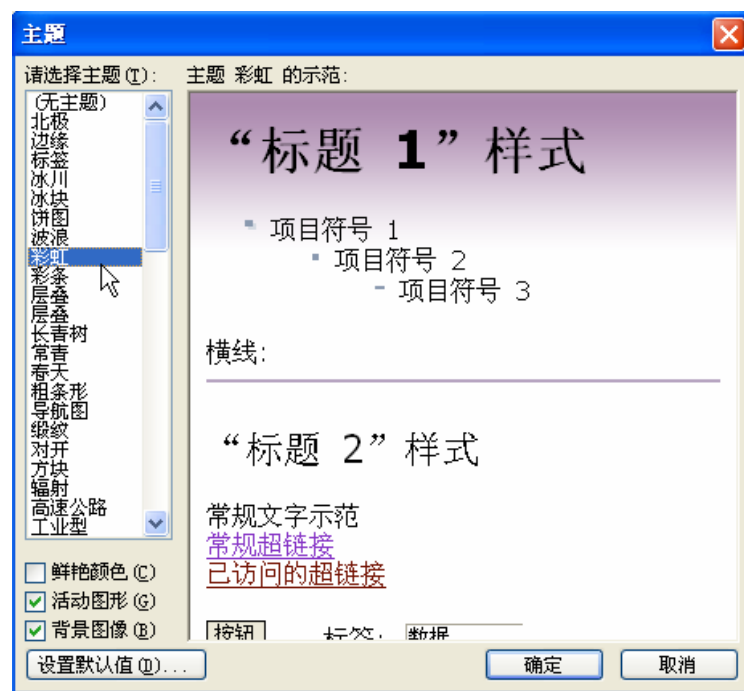
选中一个区域→“格式”菜单→“背景”→“图片”→在显示的对话框中选取一个图片文件，该图片就成为页的背景图片。



6.3.3 使用主题

主题是将字体、背景图像等元素进行统一设计、统一配色的方案集合。使用主题就是使用系统提供的设计与配色方案，快速创建具有专业水平的数据访问页。

“格式”菜单→“主题”→在对话框中选主题→“确定”。



ACCESS实例教程

转到页面视图，按主题显示页。



Page1

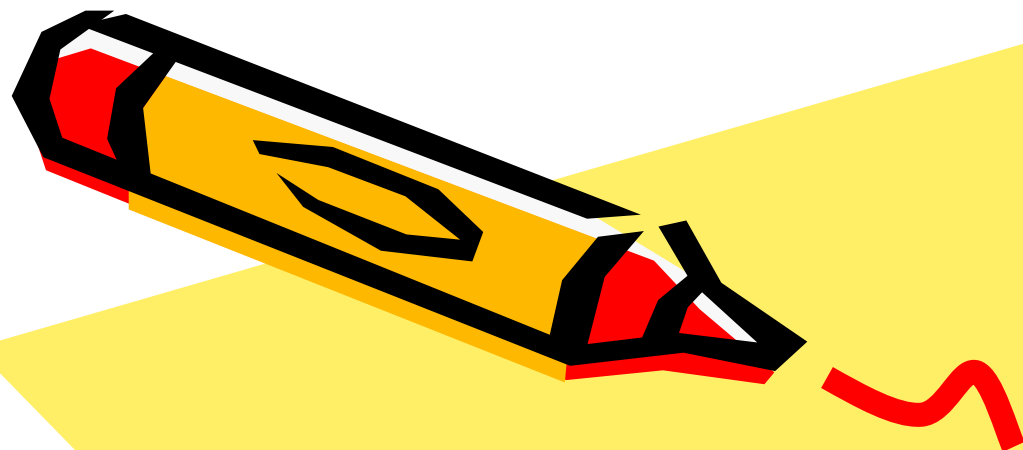
学生信息一览表

学号:	080102
姓名:	李丁
性别:	男
籍贯:	山西

学生 10 之 1

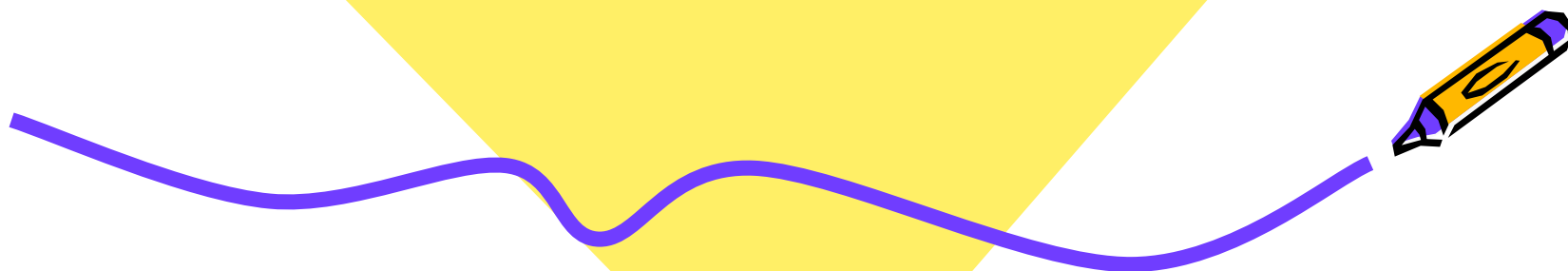


ACCESS实例教程



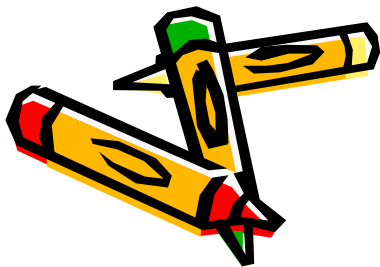
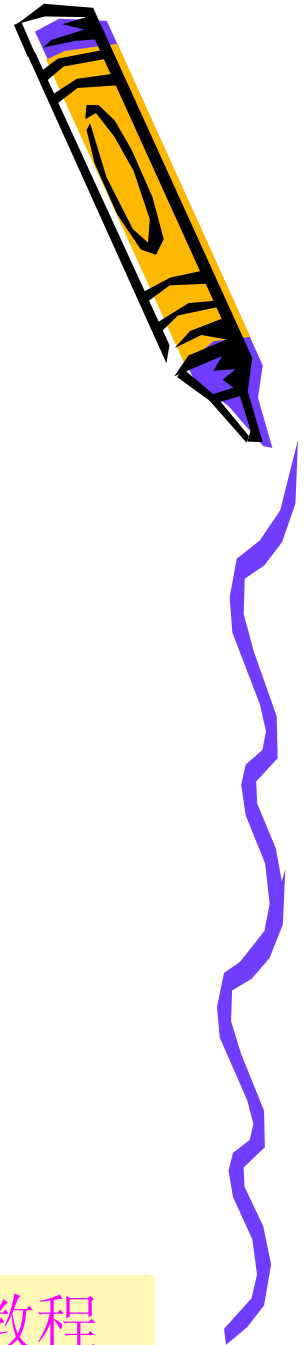
ACCESS实例教程

——宏的操作（第7章）



本章要点:

- 1, 建立宏
- 2, 建立宏组
- 3, 建立条件宏
- 4, 将宏附加给控件



7.1 认识宏对象

宏是一系列操作的集合，每个操作都自动完成特定功能，Access提供宏对象的目的是让操作能自动执行。

宏对象有3种类型：宏、宏组、条件宏。其中，宏是操作的集合，宏组是宏的集合，条件宏是带条件的操作序列，只在条件成立时才执行。



7.1.1 宏窗口

宏对象在宏窗口中建立。宏窗口分为上、下两部分，上边是设计网格，用来添加宏操作。下边是操作参数表，用来设置每个宏操作的参数。宏操作不同，相对应的参数表也不同。



7.1.2 常用宏操作

(1) 打开数据库中的对象

OpenTable命令，打开数据表。

OpenForm命令，打开窗体。

OpenReport命令，打开报表。

OpenQuery命令，打开查询。

(2) 关闭数据库对象和退出Access

Close命令，关闭指定的数据库对象。

Quit 命令，退出Access。

(3) 设置值和刷新值

SetValue命令，设置属性值。

Requery命令，刷新控件数据。



(4) 窗口操作

Maximize命令，最大化窗口。

Minimize命令，最小化窗口。

Restore命令，将最大化或最小化窗口恢复至初始大小。

(5) 运行操作

RunCommand命令，运行Access指定的内置语句。

RunSQL命令，运行指定的SQL语句。

RunMacro命令，运行指定的宏。

(6) 提示操作

Beep命令，使计算机发出“嘟嘟”声。

MsgBox命令，显示消息框。



7.1.3 宏的几个重要按钮



宏名：在宏窗口设计网格中显示“宏名”列。

条件：在宏窗口设计网格中显示“条件”列。

插入行：在当前宏操作行上方插入一个空行。

删除行：将当前的宏操作行删除。

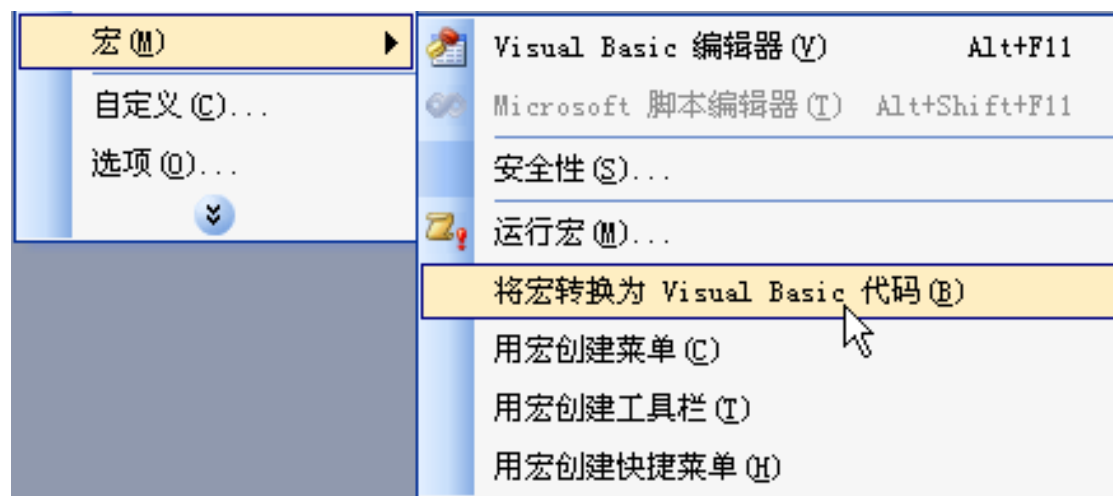
运行：运行宏，如果是宏组，只运行宏组中的第一个宏。

单步：单步运行宏。



7.1.4 将宏转换为VBA代码

选中一个宏→“工具”菜单→“宏”→“将宏转换为Visual Basic代码”。



7.2 创建宏对象

7.2.1 创建宏

例：建立宏。打开一个表和一个查询，先关闭查询，再关闭表，用消息框提示。

- (1) 新建宏
- (2) 第1个操作选OpenTable→表名称选“教师”。

操作参数	
表名称	教师
视图	数据表
数据模式	编辑

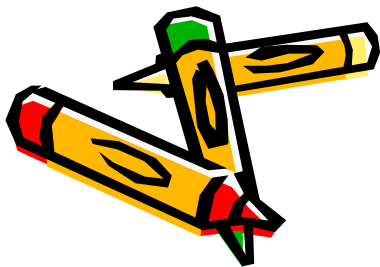
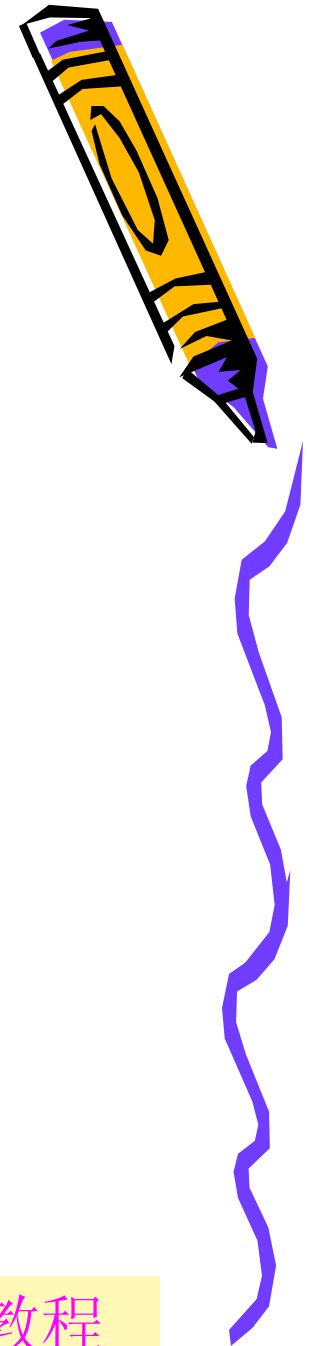


(3) 第2个操作选OpenQuery→查询名称选“教师人数”。

操作参数	
查询名称	教师人数
视图	数据表
数据模式	编辑

(4) 第3个操作选MsgBox→消息框输入“关闭教师人数查询吗?” →标题输入“提示信息”。

操作参数	
消息	关闭教师人数查询吗?
发嘟嘟声	是
类型	无
标题	提示信息



(5) 第4个操作选Close→对象类型选“查询”→对象名称选“教师人数”。

(6) 类似方法再加入一个MsgBox操作，提示“关闭教师表吗？”，加入一个Close操作，关闭教师表。

(7) “运行”菜单→“运行”，系统按顺序自动运行宏里面的6个操作。

打开关闭表和查询：宏

操作	注释
OpenTable	打开教师表
OpenQuery	打开教师人数查询
MsgBox	提示信息1
Close	关闭教师人数查询
MsgBox	提示信息2
Close	关闭教师表

操作参数

对象类型	表
对象名称	教师
保存	提示



ACCESS实例教程

7.2.2 创建宏组

例：建立并运行宏组

- (1) 新建并保存宏→“视图”菜单→“宏名”，窗口显示“宏名”列。
- (2) 在宏组中依次给两个宏起名并写入操作。

宏组练习 : 宏

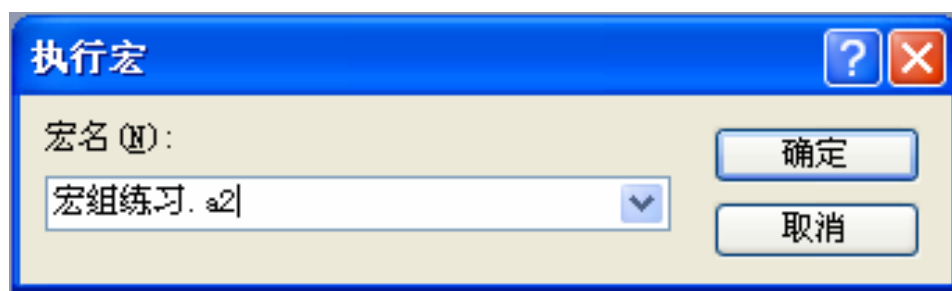
宏名	操作	注释
a1	OpenForm	打开“文本框练习”窗体
	MsgBox	提示
	Close	关闭“文本框练习”窗体
a2	OpenReport	打开“工资报表”
	MsgBox	提示
	Close	关闭“工资报表”

操作参数

报表名称	工资报表
视图	打印预览
筛选名称	
Where 条件	
窗口模式	普通



(3) “工具”菜单→“宏”→“运行宏”→在对话框中输入“宏组练习.a2”→单击“确定”按钮，宏组中的第2个宏被执行。



(4) 对宏组用“运行”命令，只运行宏组中的第1个宏。
也可以用上述方法运行宏组中第1个宏。



7.2.3 创建条件宏

条件宏根据条件结果的“真”或“假”，选择执行或不执行相应操作。当表达式的值为“真”时，运行对应的操作，当表达式值为“假”时，忽略对应的操作。

如果相邻操作的条件表达式相同，条件可以用省略号（…）代替。省略号输入3个西文句点即可。

（1）引用窗体控件的语法为：

Forms![窗体名]![控件名] 或 [Forms]![窗体名]![控件名]

（2）引用报表控件的语法为：

Reports![报表名]![控件名] 或 [Reports]![报表名]![控件名]



例：简单条件的条件宏

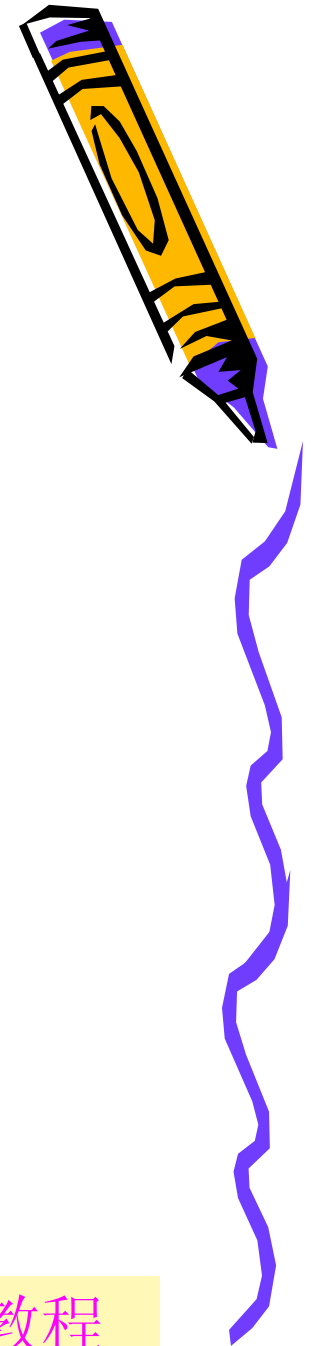
- (1) 在窗体中生成复选框控件→名称为fxk。
- (2) 新建宏→以“条件宏”为名保存→“视图”菜单→“条件”，宏窗口中显示“条件”列。
- (3) 在条件列第1行写条件表达式：[Forms]![ct]![fxk] →操作选MsgBox→消息写“你选了我了！”
- (4) 第2行条件相同→操作选Beep。
- (5) 第3行条件相同→操作选MsgBox→消息写“谢谢！”。
- (6) 第4行写条件：not [Forms]![ct]![fxk] →操作选MsgBox→消息写“你没有选我！”。
- (7) 第5行条件相同→操作选Beep。
- (8) 第6行条件相同→操作选MsgBox→消息写“再见！”。



(9) 用设计视图打开ct窗体→选取控件fxk→属性窗口单击“事件”卡→在“更新后”属性框中选“条件宏”。



显示结果:



ACCESS实例教程

例：复杂条件的条件宏

- (1) 新建窗体ct1 →生成2个复选框控件fxk1和fxk2→附加标签的标题分别为“唱歌”和“跳舞”。
- (2) 生成命令按钮控件c1→标题为“确定”。
- (3) 新建宏“条件宏2”→显示“条件”列。
- (4) 条件1: [Forms]![ct1]![fxk1] and [Forms]![ct1]![fxk2]
→操作选MsgBox→消息写“又唱歌又跳舞”。
- (5) 条件2: [Forms]![ct1]![fxk1] and not [Forms]![ct1]![fxk2]
→操作选MsgBox→消息写“只唱歌不跳舞”。
- (6) 条件3: not [Forms]![ct1]![fxk1] and [Forms]![ct1]![fxk2]
→操作选MsgBox→消息写“不唱歌只跳舞”。
- (7) 条件4: not [Forms]![ct1]![fxk1] and not [Forms]![ct1]![fxk2]
→操作选MsgBox→消息写“不唱歌不跳舞”。



条件宏2 : 宏

条件	操作	注释
[Forms]![ct1]![fxk1] And [Forms]![ct1]![fxk2]	MsgBox	都选中
[Forms]![ct1]![fxk1] And Not [Forms]![ct1]![fxk2]	MsgBox	选中一个
Not [Forms]![ct1]![fxk1] And [Forms]![ct1]![fxk2]	MsgBox	选中一个
▶ Not [Forms]![ct1]![fxk1] And Not [Forms]![ct1]![fxk2]	MsgBox	都没选中

操作参数

消息	不唱歌不跳舞
发嘟嘟声	是
类型	无
标题	消息提示

在此列中输入注释。



ACCESS实例教程

(8) ct1窗体中选取命令按钮→属性窗口单击“事件”卡
→在“单击”属性框中选“条件宏2”。



执行结果：



ACCESS实例教程

7.2.4 自动运行宏

名为autoexec的宏称为“自动运行宏”，打开数据库的同时会自动执行“自动运行宏”中包含的操作。自动运行宏的名字autoexec是专用的。

如果打开数据库时不想运行“自动运行宏”，可以在打开数据库时按住shift键，这样就能取消autoexec宏的自动运行。



7.3 宏的编辑与调试

7.3.1 宏的编辑

1. 更改宏组中的宏名和更改条件表达式

直接在宏名或条件表达式处输入新内容即可。

2. 更换宏的操作

单击操作框的向下箭头，在系统提供的52个操作中重新选一个。

3. 更换操作的参数

单击参数框的向下按钮，在系统提供的参数项中重新选一个。

4. 插入行与删除行

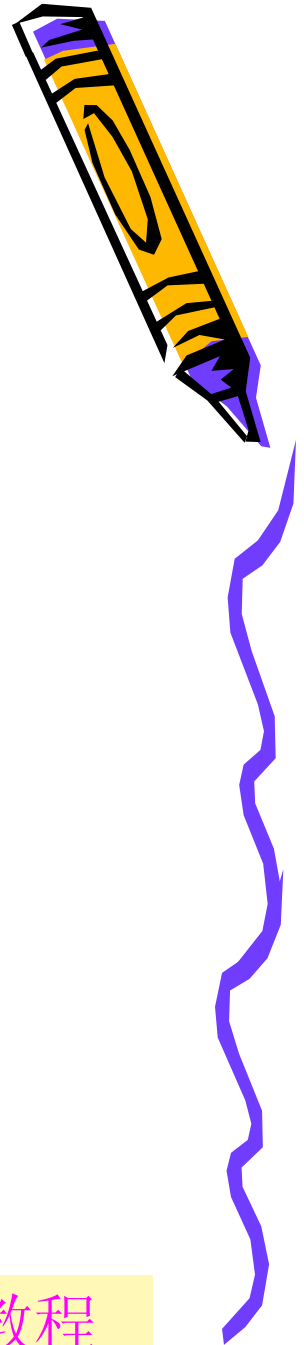
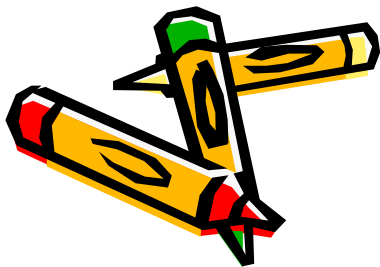
在设计网格中选定一行，单击插入行按钮，在当前行上方插入一个空行。选定一行，单击删除行按钮，删除当前行。



7.3.2 宏的调试

系统为宏提供了单步执行的调试工具，操作步骤如下：

- (1) 打开要调试的宏。
- (2) 按下单步运行按钮，或者打开“运行”菜单→选“单步”，使运行进入单步步跟踪状态。
- (3) 单击运行按钮，或者打开“运行”菜单→选“运行”，系统显示“单步执行宏”对话框。



(4) 单击对话框中“单步执行”按钮，依次执行宏的操作，每步操作完毕后都会暂停，显示操作信息或操作结果。

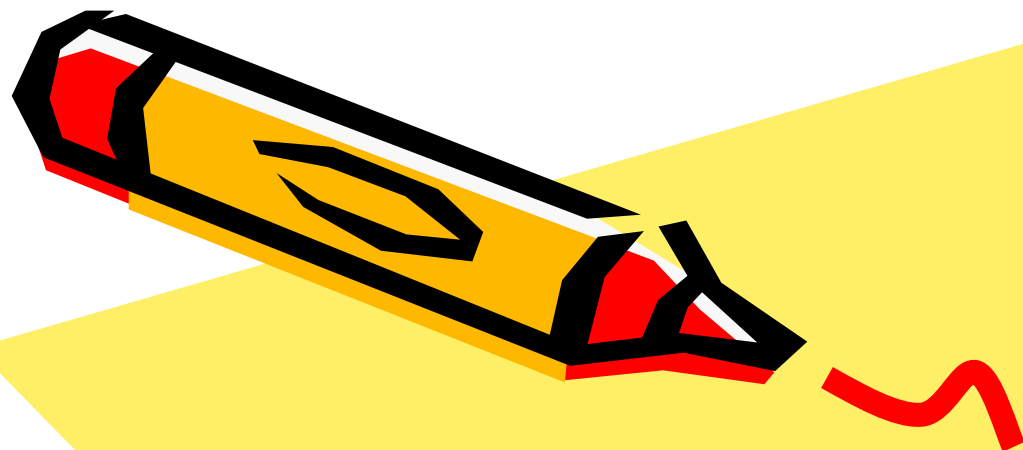
(5) 单击对话框中“继续”按钮，执行宏的下一个操作。

(6) 单击对话框中“停止”按钮，停止宏的执行，关闭对话框。

(7) 如果宏操作有问题，显示“操作”失败对话框。

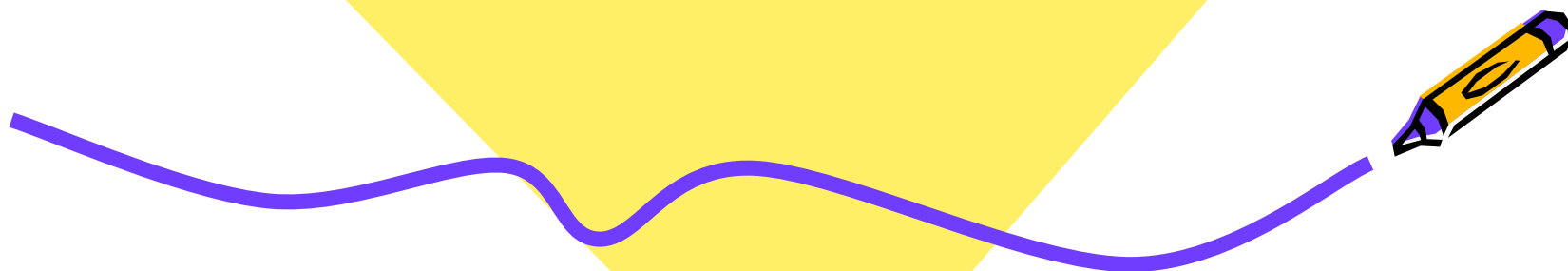
(8) 用组合键Ctrl+Break可以在宏的执行过程中暂停宏的执行。





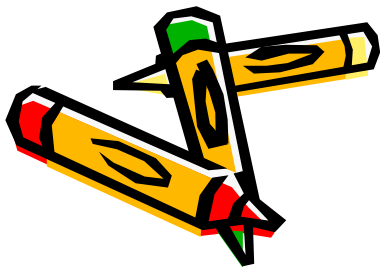
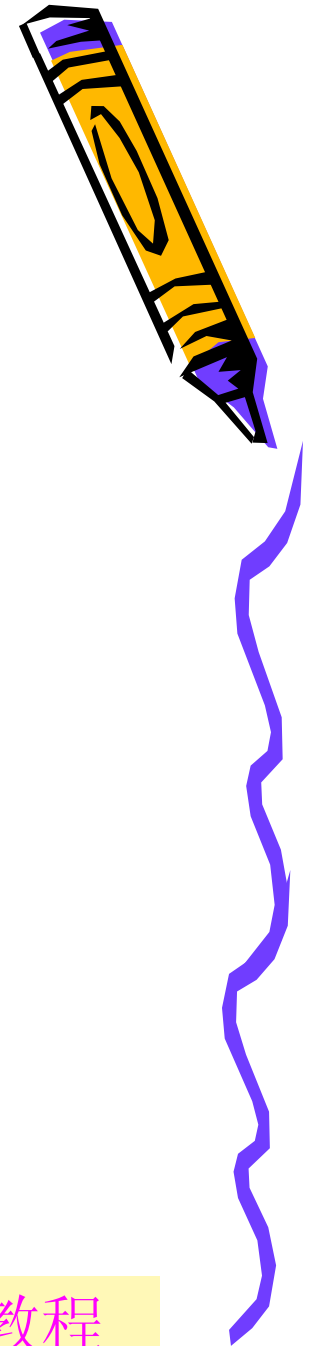
ACCESS实例教程

——模块的操作（第8章）



本章要点:

- 1, 模块的基本概念
- 2, VBA程序设计基础
- 3, VBA函数的使用
- 4, 模块的建立与使用
- 5, 过程与参数传递



8.1 认识模块对象

模块是由声明、语句和过程组成的集合，以VBA（Visual Basic for Application）语言编写，作为一个已命名的单元存储在一起。

模块有2种类型：类模块和标准模块。使用模块可以在实际开发中实现较为复杂的功能。



8.1.1 模块的基本概念

(1) 模块

模块由一个或多个过程组成，每个过程实现一个或几个功能。模块的执行通过特定事件激发相应的事件过程实现。

(2) 事件过程

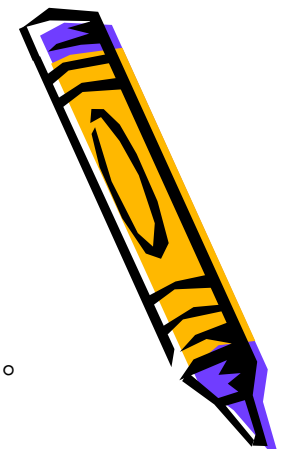
事件过程是一组代码，用于响应窗体事件或报表事件。

(3) 类模块

类模块是包含代码和数据的集合，可以看作是没有物理表示的控件，总是与某一特定的窗体或报表相关联。窗体模块和报表模块都属于类模块。

(4) 标准模块

标准模块是代码的集合，包含的过程不与任何其他对象相关联，是数据库对象使用的公共过程，保存在数据库窗口中。



8.1.2 启动Access的编程界面

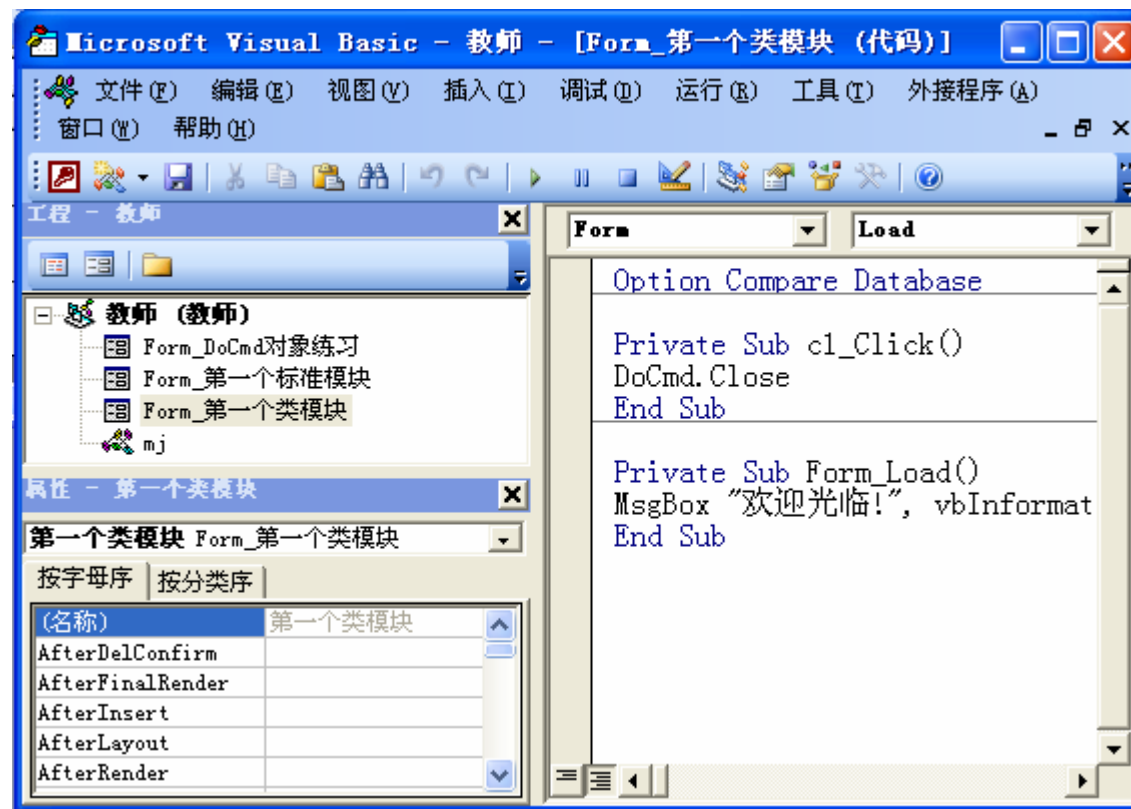
Access的编程界面称为VBE（Visual Basic Editor），是Microsoft Office所有组件公用的程序编辑系统。

方法1：在窗体或报表设计视图中选控件→单击标准工具栏的“生成器”按钮→在“选择生成器”对话框中选择“代码生成器”。

方法2：右击窗体或报表设计视图中控件→快捷菜单中选“事件生成器”→“选择生成器”对话框中选择代码生成器”。



8.1.3 VBE窗口



ACCESS实例教程



编写模块对象的代码在代码窗口进行。

(1) 窗口顶部有2个组合框，左边对象框显示所有对象名称，右边过程框显示当前对象能识别的所有事件名称。选定一个对象，再选定一个事件，系统会自动生成相应事件过程的起始行与结束行，只须在两行中间添加过程代码即可。

“通用”过程可以被所有控件调用。某控件的过程只在关联事件中有效。

(2) 窗口中央代码区有声明区和过程区，声明区中声明模块使用的变量，过程区显示一个或多个过程，之间用一条灰线分隔。

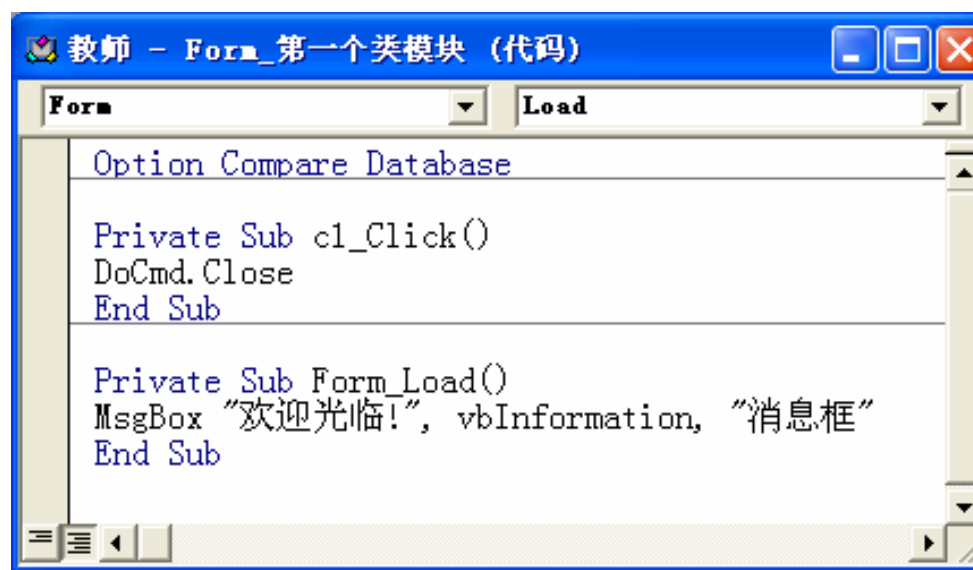
(3) 窗口底部有2个按钮，单击“过程视图”按钮，只显示当前过程。单击右边“完整的模块视图”按钮，窗口显示全部过程。



8.1.4 标准工具栏

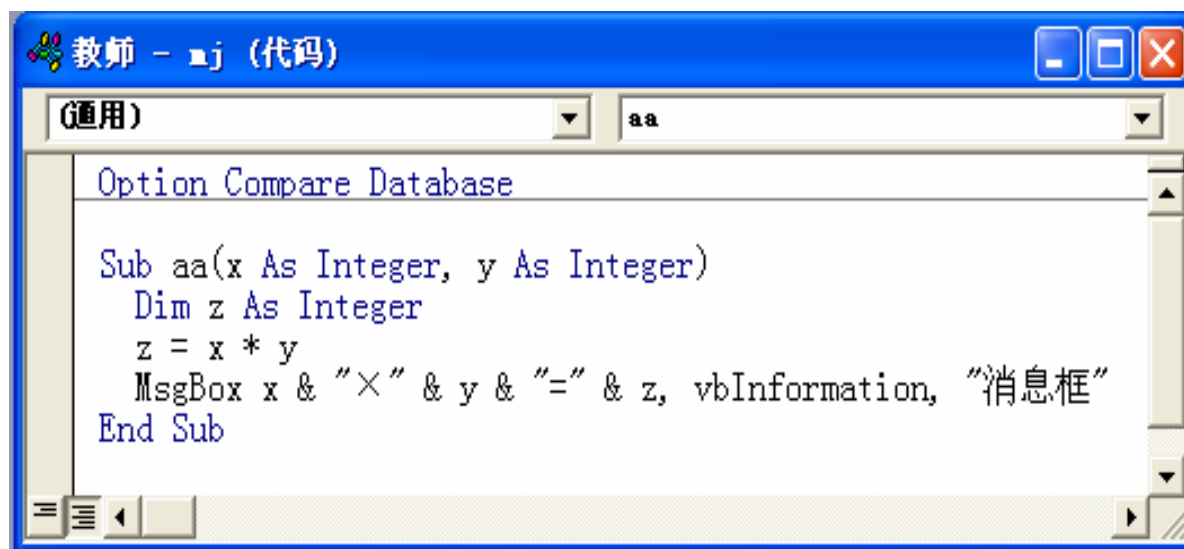


例:一个类模块



ACCESS实例教程

例：一个标准模块



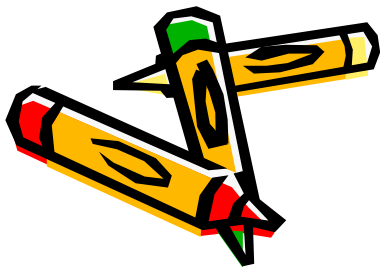
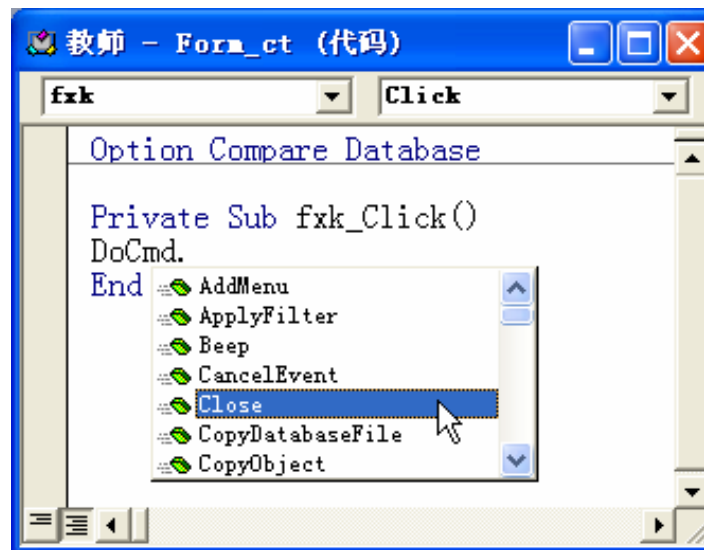
```
Option Compare Database

Sub aa(x As Integer, y As Integer)
    Dim z As Integer
    z = x * y
    MsgBox x & "×" & y & "=" & z, vbInformation, "消息框"
End Sub
```



8.1.7 使用提示与帮助

在代码窗口输入代码时，系统会自动显示提示信息，包括关键字列表、属性列表、过程参数列表等，在列表选取一项后双击鼠标，所选值会自动添加到当前光标处。



8.2 VBA程序设计基础

8.2.1 面向对象的程序设计的基本概念

(1) 对象

在自然界中，一个对象就是一个实体，如一辆汽车就是一个对象。在面向对象的程序设计中，对象代表应用程序中的元素，如表、窗体、按钮等。

(2) 属性

属性是对象的特征。如汽车有颜色和型号属性，按钮有标题和名称属性。对象的类别不同，属性会有所不同。同类别对象的不同实例，属性也有差异。例如，同是命令按钮，名称属性不允许相同。



(3) 事件

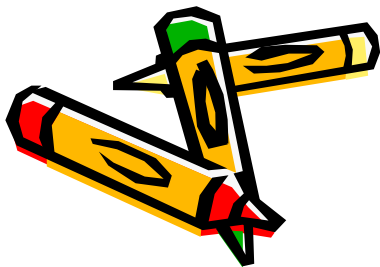
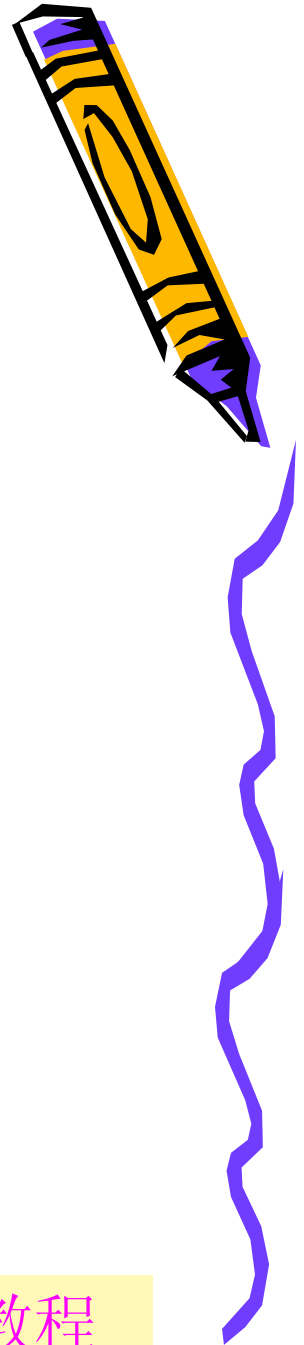
事件是对象能够识别的动作。如按钮可以识别单击事件、双击事件等。在类模块每一个过程的开始行，都显示对象名和事件名。如：Private Sub cl_Click()。

(4) 方法

方法是对象能够执行的动作，决定了对象能完成什么事。不同对象有不同的方法。如close方法能关闭一个窗体。

(5) 集合

集合由许多与对象有关的键和值组成，其中的键和值是配对的。如一本书是一个对象，书的页码是键，页码对应的内容是值。所有配对的页码和内容组成了书对象的集合。



(6) 过程

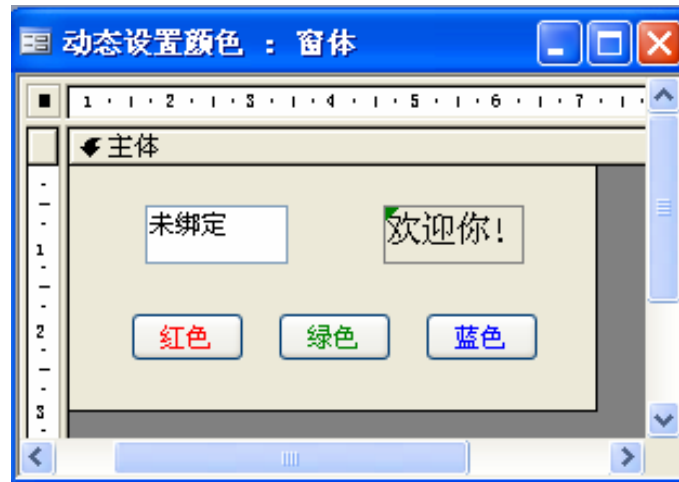
过程是由代码组成的单元，包含一系列计算语句和执行语句。每一个过程都有名字，过程名不能与所在模块的模块名相同。过程有两种类型：sub过程（无返回值），function过程（有返回值）。

例：动态设置控件属性

- (1) 在窗体中建1个文本框→名称为t1。
- (2) 在窗体中建1个标签→名称为b1→标题为“新年好！”。
- (3) 在窗体中建3个命令按钮→名称分别为c1、c2、c3→标题分别为“红色”、“绿色”、“蓝色”。



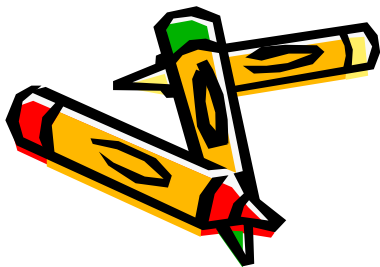
(4) 设置各按钮的前景色，在属性窗口可查看颜色值。



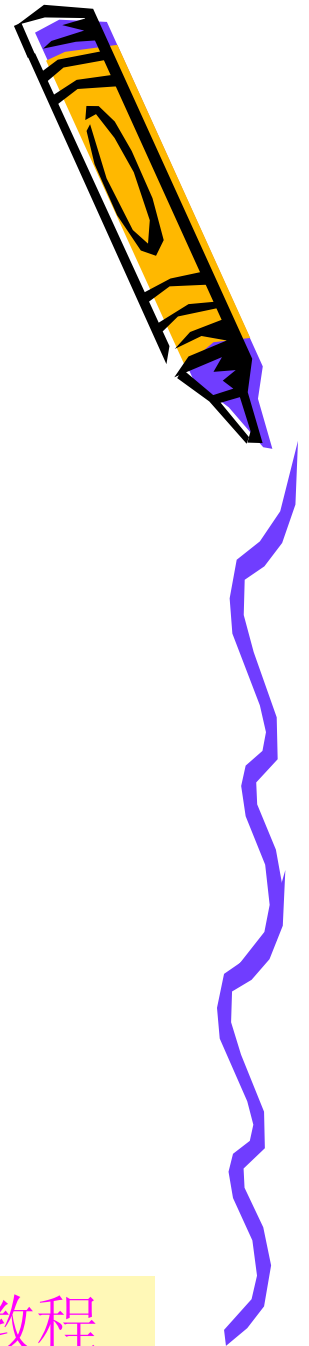
(5) c1的click事件代码：

```
t1.BackColor = 255
```

```
b1.ForeColor = 255
```



ACCESS实例教程



(6) c2的click事件代码:

```
t1.BackColor = 33792
```

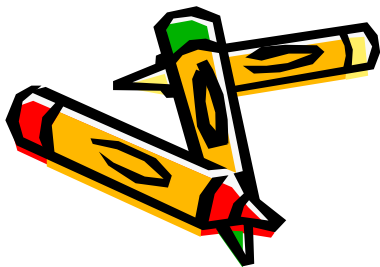
```
b1.ForeColor = 33792
```

(7) c3的click事件代码:

```
t1.BackColor = 16711680
```

```
b1.ForeColor = 16711680
```

执行结果:



ACCESS实例教程

8.2.2 Docmd对象

Docmd是Access的一个特殊对象，用来调用内置方法，在程序中实现对Access的操作，诸如打开窗口、关闭窗体、打开报表、关闭报表等。

DoCmd 对象的大多数方法都有参数，有些参数是必需的，有些则是可选的。若省略可选参数，参数将采用默认值。

(1) 用DoCmd 对象打开窗体

格式：DoCmd.OpenForm “窗体名”

功能：用默认形式打开指定窗体。

例如：DoCmd.OpenForm “文本框练习”



(2) 用DoCmd 对象关闭窗体

格式1: DoCmd.Close acForm, "窗体名"

功能: 关闭指定窗体。

例如: DoCmd.Close acForm, "文本框练习"

格式2: DoCmd.Close

功能: 关闭当前窗体。

(3) 用DoCmd 对象打开报表

格式: DoCmd.OpenReport "报表名", acViewPreview

功能: 用预览形式打开指定报表。

例如: DoCmd.OpenReport "工资报表", acViewPreview



(4) 用DoCmd 对象关闭报表

格式1: DoCmd.Close acReport, "报表名"

功能: 关闭指定报表。

例如: DoCmd.Close acReport, "工资报表"

格式2: DoCmd.Close

功能: 关闭当前报表。

(5) 用DoCmd 对象运行宏

格式: DoCmd.RunMacro "宏名"

功能: 运行指定宏。

例如: DoCmd.RunMacro "宏1"

(6) 用DoCmd 对象退出Access。

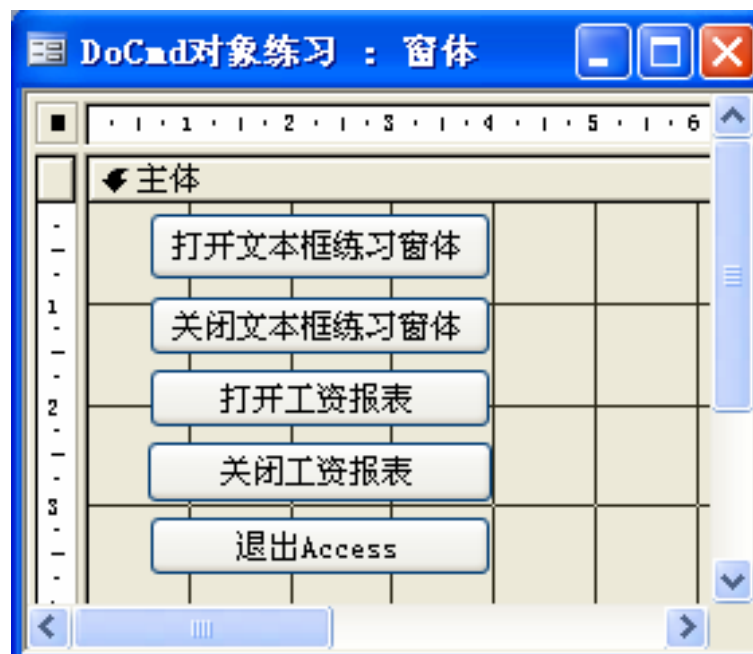
格式: DoCmd.Quit

功能: 关闭所有Access对象和Access本身。



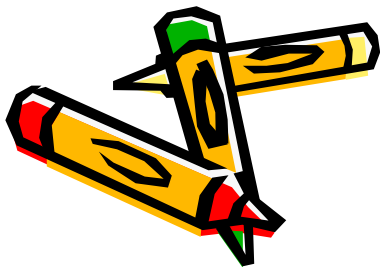
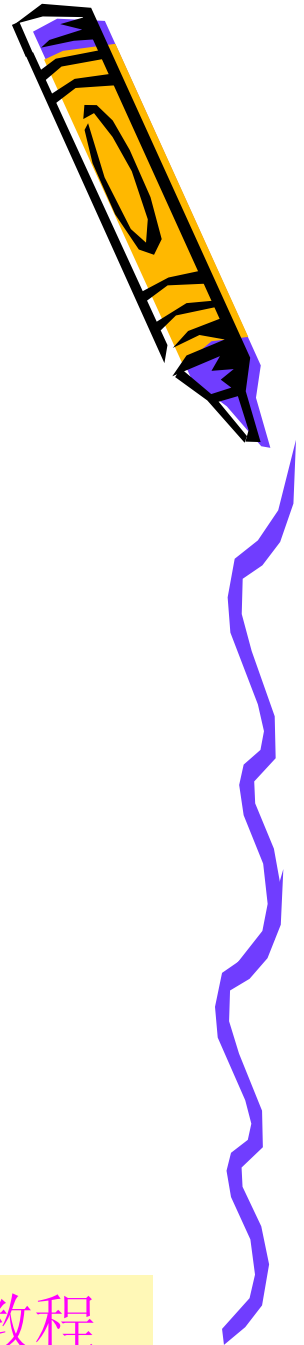
例：使用DoCmd对象

(1) 在窗体中建立5个按钮→名称分别为c1、c2、c3、c4、c5→标题分别为“打开文本框练习窗体”、“关闭文本框练习窗体”、“打开工资报表”、“关闭工资报表”、“退出Access”。

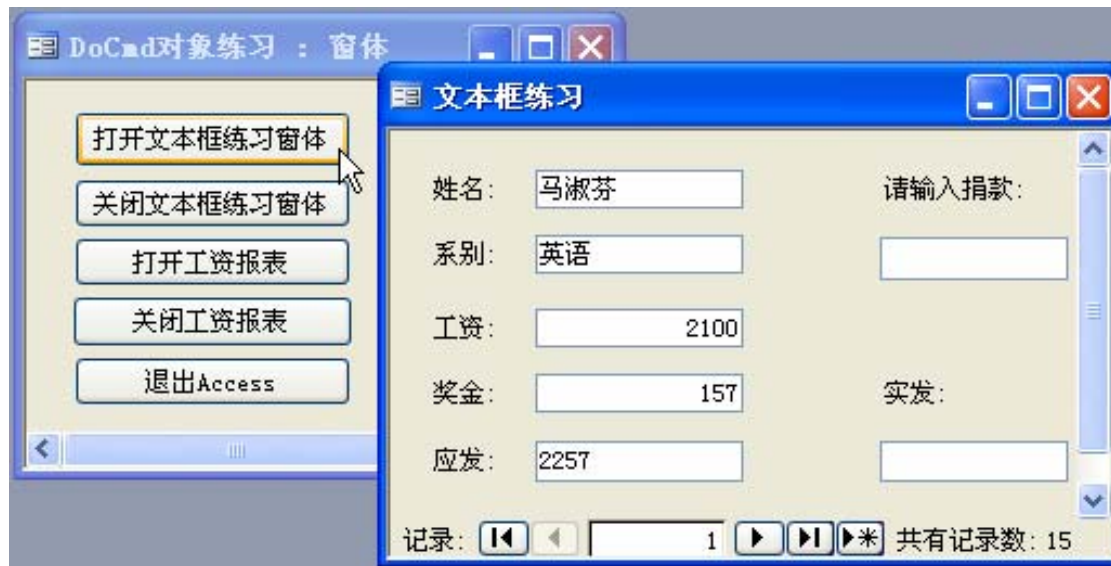


ACCESS实例教程

- (2) c1代码: DoCmd.OpenForm "文本框练习"
- (3) c2代码: DoCmd.Close acForm, "文本框练习"
- (4) c3代码: DoCmd.OpenReport "工资报表", acViewPreview
- (5) c4代码: DoCmd.Close acReport, "工资报表"
- (6) c5代码: DoCmd.Quit



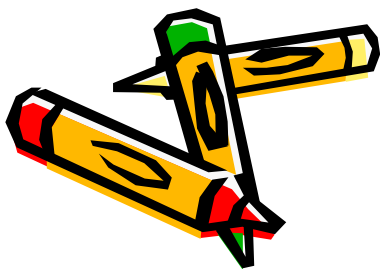
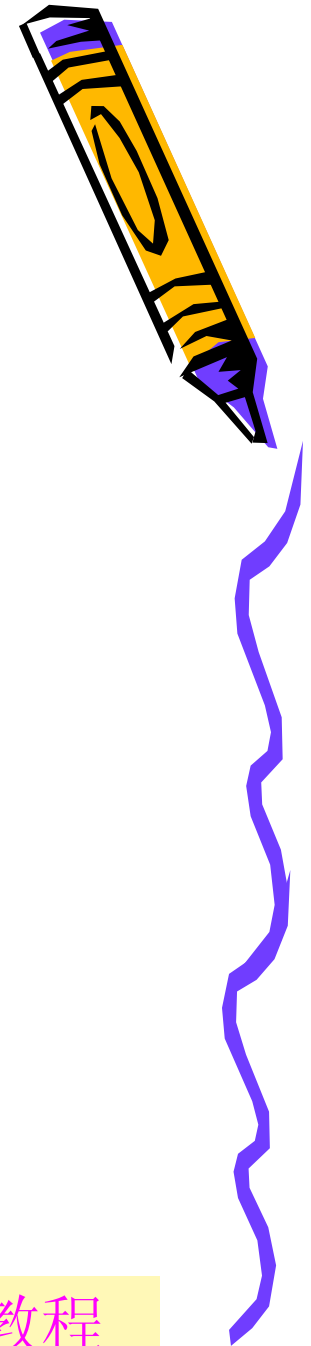
(7) 转到窗体视图→单击不同按钮执行不同操作。



The screenshot displays an Access application window titled "DoCmd对象练习：窗体". Inside, there is a sub-form titled "文本框练习". The sub-form contains several text boxes for data entry:

- 姓名: 马淑芬
- 系别: 英语
- 工资: 2100
- 奖金: 157
- 应发: 2257

There are also labels for "请输入捐款:" and "实发:" next to empty text boxes. At the bottom of the sub-form, there is a record navigation bar showing "记录: 1" and "共有记录数: 15".

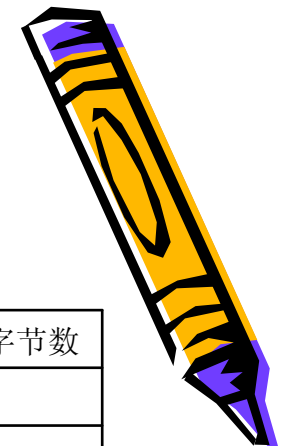


ACCESS实例教程

8.2.3 数据类型与数据类型转换

(1) 标准数据类型

VBA类型	符号	字段类型	取值范围	字节数
Byte		字节	0~255之间的整数	1
Integer	%	整型	-32768~32767之间的整数	2
Long	&	长整型	-2147483648~2147483647之间的整数	4
Single	!	单精度	负数: -3.402823E38~-1.401298E-45 正数: 1.401298E-45~3.402823E38	4
Double	#	双精度	负数: -1.79769313486232E308~ -4.94065645841247E-324 正数: 4.94065645841247E-324~ 1.79769313486232E308	8
Currency	@	货币	-922337203685477.5808~922337203685477.5807	8
String	\$	文本	0~65535个字符	
Boolean		是/否	true或false	1
Date		日期/时间	January 1,100到December 31,9999	8
Variant		无		



ACCESS实例教程

(2) 数据类型转换函数

函数	转换后类型	说明
CByte(x)	Byte	x取值范围同Byte，小数部分四舍五入，超出范围显示“溢出”
CInt(x)	Integer	x取值范围同Integer，小数部分四舍五入
CLng(x)	Long	x取值范围同Long，小数部分四舍五入
CSng(x)	Single	x取值范围同Single
Cdbl	Double	x取值范围同Double
CCur	Currency	x取值范围同Currency
CBool(x)	Boolean	x取值范围是任何有效数字或字符串
CDate(x)	Date	x取值范围是任何有效日期表达式
CStr(x)	String	无
CVar(x)	Variant	x如果是数值范围同Double，否则，与String相同

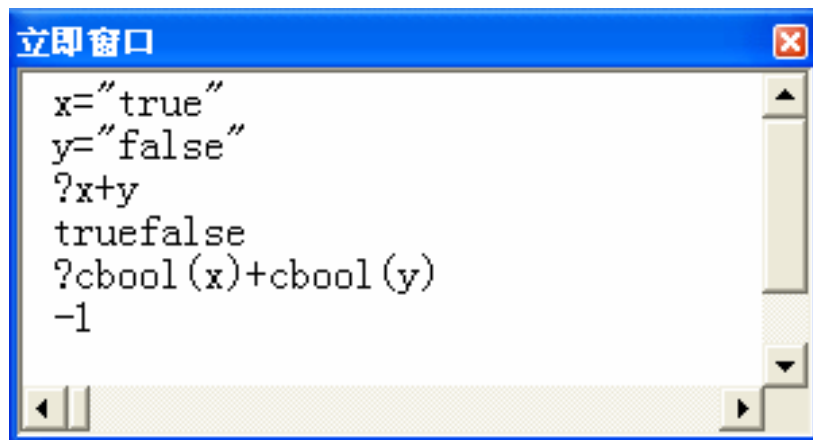


ACCESS实例教程

例：在立即窗口中计算函数值

- (1) 在VBE界面→“视图”菜单→“立即窗口”，显示立即窗口。
- (2) 输入表达式x=“true”→回车→输入表达式y=“false”→回车→输入表达式 ?x+y→回车，显示结果为：truefalse。
- (3) 输入表达式 ?cbool(x)+cbool(y)→回车，显示结果为：-1。

说明：每行写完后必须回车才能生效。问号与print语句相同，在立即窗口显示计算结果。



```
立即窗口
x="true"
y="false"
?x+y
truefalse
?cbool(x)+cbool(y)
-1
```



(3) 用户定义的数据类型

定义格式: type 数据类型名
 域名1 as 数据类型
 域名2 as 数据类型
 ...
 end type

例如: type student
 xh as string*6
 xm as string
 xb as string*1
 nl as integer
end type



8.2.4 变量

变量是指程序运行时值会发生变化的数据。系统会按照变量的数据类型在内存中为变量分配一定数量的存储单元，程序中用变量名调用存储的数据。所以，变量实际上是一个符号地址，代表了命名的存储位置。

(1) 变量命名规则

- 由字母、数字、下划线组成，用英文字母开头。
- 不能包含空格以及除下划线之外的标点符号。
- 不能用VBA的关键字（如：dim）和保留字（如：str）。
- 字符个数不得多于255个。
- 变量名不区分大小写。





(2) 用dim语句声明变量

格式: `dim 变量名 as 类型`

说明:

- 如果不用as定义类型, 则变量为变体类型。变体类型变量比其他类型变量占用更多的内存资源。
- 可以用dim同时定义多个变量, 变量之间用西文逗号分隔, 每一个变量都应该用as声明类型。

例如: `dim a1 as integer, a2 as boolean`

- 用dim定义的变量是局部变量, 会按照数据类型自动设置默认值。



(3) 用类型说明符声明变量类型

在变量名后加VBA的类型说明符，隐式的声明了变量类型。

例如：b1%=125

变量b1是整型，值是125。

既没有显式声明，也没有加类型说明符，隐式声明了变体型变量。

例如：c1=125

变量c1是变体型，值是125。



(4) 一个特殊的模块变量

本地窗口打开时，自动生成一个名为“Me”的特殊模块变量。对于类模块，定义为Me。Me是对象的引用，引用当前模块中当前类的实例。

Me变量不需要专门定义，直接使用即可。

例如，用代码定义“学生信息”窗体中“Lab”标签的标题属性，以下两种方法效果是一样的：

标准方法：Forms! 学生信息!Lab.Caption=“学生信息浏览”

常用方法：Me!Lab.Caption=“学生信息浏览”



8.2.5 变量的作用域和生命周期

1. 变量的作用域

变量的作用域是变量在程序中起作用的范围。分3个层次，从低到高依次为：局部、模块、全局。

(1) 局部变量

又称为本地变量，仅在声明变量的过程中有效。在过程和函数内部用dim声明或不用声明接使用的变量，都是局部变量。局部变量在本地拥有最高级，当存在同名的模块级变量时，模块级变量被屏蔽。

(2) 模块变量

模块变量在所声明模块的所有函数和所有过程都有效，变量定义在模块所有过程之外的起始位置，通常是窗体变量或标准模块变量。

(3) 全局变量

全局变量又称为公共变量，定义在标准模块所有过程之外，在所有模块的所有过程和函数中都有效。

定义格式：public 变量名 as 数据类型



2. 变量的生命周期

变量的生命周期是指变量从首次出现到变量消失的代码执行时间。变量首次出现是指声明变量并为其分配存储空间，变量消失是指变量所在的程序执行完毕。

局部变量的生命周期从过程或函数被调用到运行结束。全局变量的生命周期从声明到Access应用程序结束。

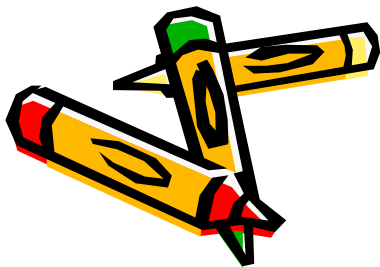
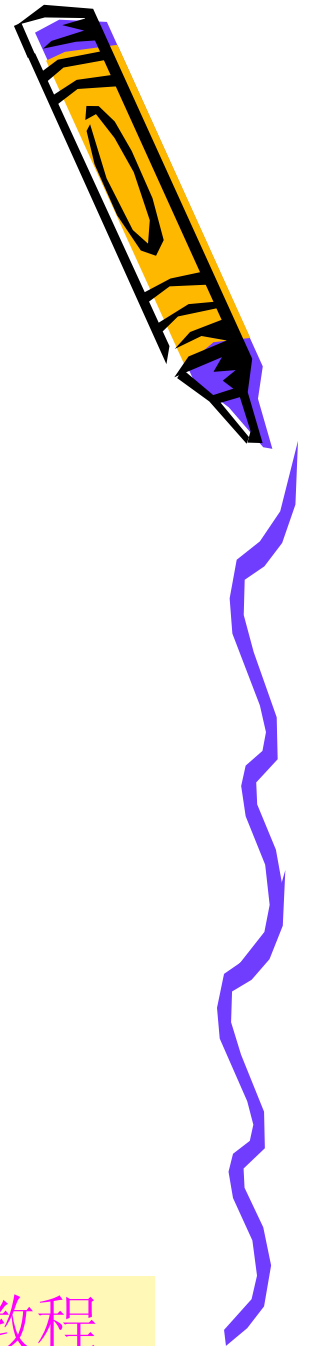
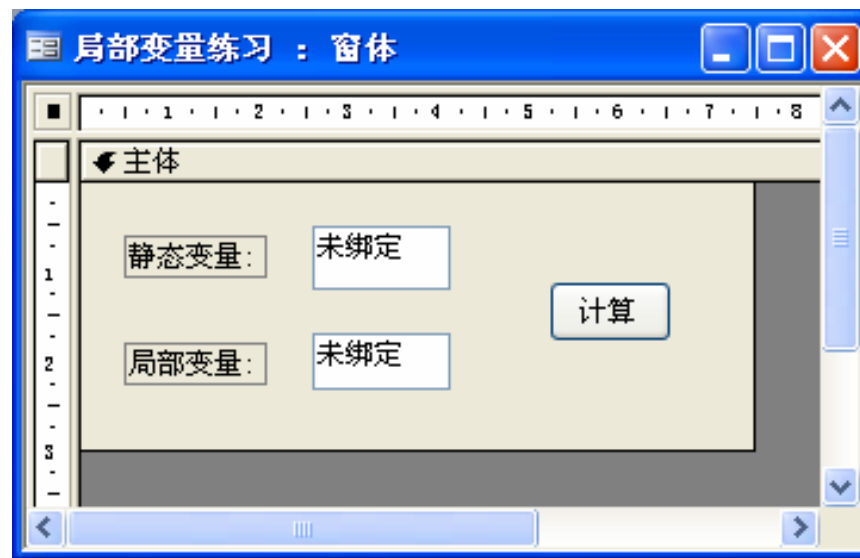
过程中用dim定义的变量，每次调用过程时都重新开始，过程结束时立即消失。用static代替dim定义变量，可以在过程实例间保留局部变量的值。用static定义的变量称为静态变量，作用范围与dim相同，在整个模块执行时一直存在。



例：用static和dim定义局部变量

(1) 在窗体中建立2个文本框→分别命名为t1和t2→附加标签的标题分别为“静态变量”和“局部变量”。

(2) 在窗体中建立命令按钮→名称为c1→标题为“计算”。



(3) c1的click代码:

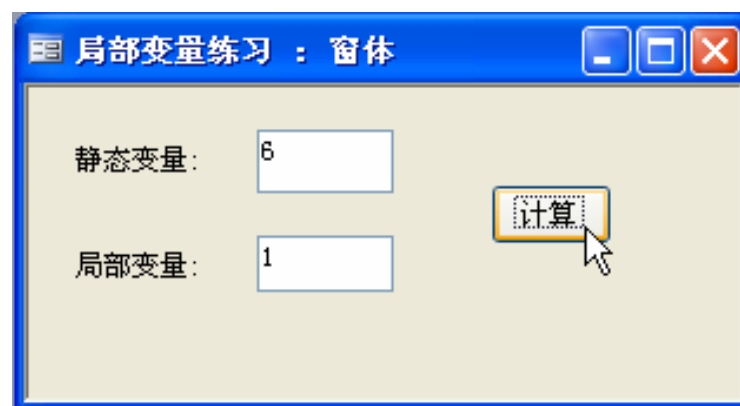
```
Static x As Integer
```

```
Dim y As Integer
```

```
x = x + 1: y = y + 1
```

```
t1 = x: t2 = y
```

(4) 执行时不断单击命令按钮，静态变量文本框中的值每次增加1，而局部变量文本框中的值总是1。



ACCESS实例教程

8.2.6 常量

常量是在程序运行过程中其值不能改变的量，如数字常量123、字符串常量"abc"、日期常量#2008-8-8# 等。这样的常量是直接常量，也称为字面常量。

系统还提供3种常量：符号常量、系统常量、内部常量。





1. 符号常量

用符号常量表示在编程中反复使用的相同值。符号常量名一般用大写字母，以便与变量名区分。

定义格式：Const 符号常量名=常量值

说明：

(1) 如果符号常量定义在模块声明区，所有模块的过程都能使用该变量，通常在前面加上Global或Public。

如：Public Const PI=3.14

(2) 如果符号常量定义在事件的过程中，该符号常量只在本过程中可用。

(3) 符号常量定义时不用指明数据类型。





2. 系统常量

系统常量是Access启动时就建立的常量，可以在所有程序编码时直接使用。

系统常量有：true、false、yes、no、on、off、null。

3. 内部常量

内部常量又称为固有常量，是VBA提供的一些预定义的内部符号常量，主要用来作为DoCmd对象的参数。

内部常量用前两个字母指明该常量的对象库：以ac开头的是Access的库常量，以vb开头的是VBA的库常量，以db开头的是DAO的库常量，以ad开头的ADO的库常量。

这些内部常量名不能作为用户自定义的变量名或符号常量名。



8.2.7 数组

1. 数组的概念

VBA中的数组是一组具有相同数据类型、同一名字、不同下标的变量集合。数组变量由变量名和数据下标组成，数组下标用圆括号括起来。

说明：

- (1) 数组要先定义后使用，VBA不允许隐式声明数组。
- (2) 同一过程中数组名不能与其他变量重名。





2. 声明一维数组

格式1: `dim 数组名(下标上限) as 数据类型`

格式2: `dim 数组名(下标下限 to 下标上限) as 数据类型`

例如: `dim a(6) as integer`

功能: 声明了有7个元素的数组a, 元素下标从0到6, 默认值均为0。

例如: `dim b(1 to 6) as string`

功能: 声明了有6个元素的数组b, 下标从1到6, 默认值均为空串。

说明:

- (1) 如果不定义数组下标的下限, 默认下标下限为0。
- (2) 数组定义中的参数必须是常数。
- (3) 如果使用as语句定义数组类型, 同一数组只能存放相同类型数据。





3. 声明多维数组

格式: `dim 数组名(下标上限1, 下标上限2, ...)` as 数据类型

例如: `dim c(3, 4) as integer`

功能: 声明有20个元素的数组c, 行下标从0到3, 列下标从0到4。

例如: `dim d(1 to 3, 2 to 4) as integer`

功能: 声明有9个元素的数组d, 行下标从1到3, 列下标从2到4。

4. 使用数组

声明数组后, 每个数组元素都被当作单个变量使用。

一维数组元素的引用格式: 数组名 (下标)

二维数组元素的引用格式: 数组名 (下标1, 下标2)



例：使用符号常量和数组

- (1) 在窗体建文本框→名称为t1。
- (2) 在窗体建命令按钮→名称为c1→标题为“确定”。
- (3) c1的代码：

```
const AA = “欢迎你!”          ‘ 定义符号常量
```

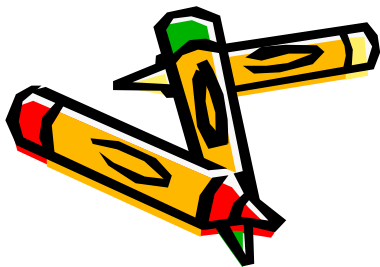
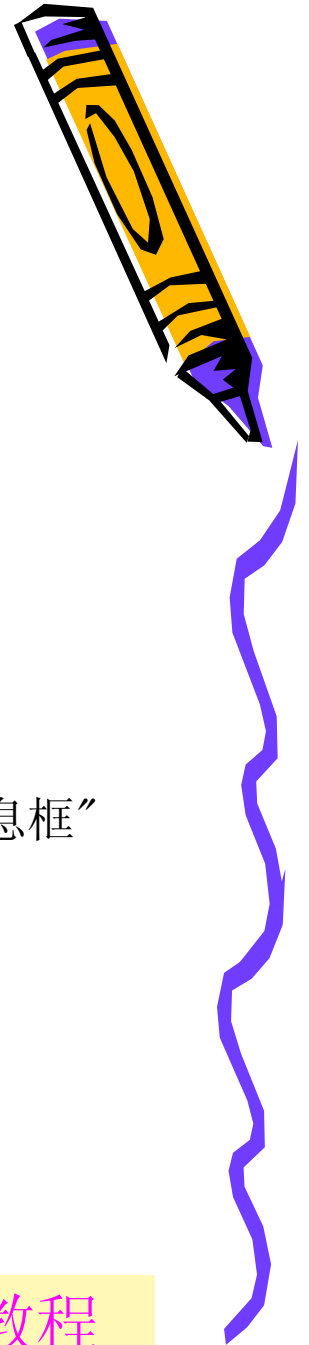
```
dim b(1) As String             ‘ 定义数组
```

```
b(0) = AA
```

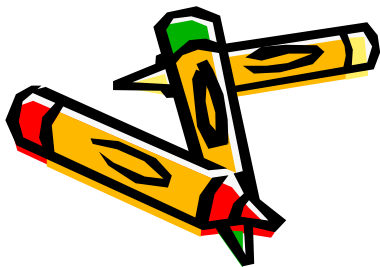
```
b(1) = t1
```

```
MsgBox b(0) & Chr(13) & Chr(13) & b(1), vbInformation, “消息框”
```

说明：Chr(13)生成一个回车，用单引号开始的字符串为注释语句。



(4) 转到窗体视图→在文本框中输入“张三”→单击“确定”按钮。



8.2.8 用户定义数据类型

用户定义数据类型包含几个分量，先用dim、public或static将变量声明为用户定义数据类型，然后给每个分量赋值。给分量赋值的格式为：变量名.分量名=值。

给用户定义数据类型的变量赋值，就是给变量的每个分量赋值。

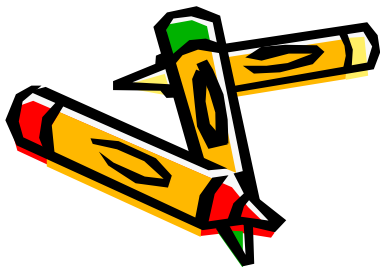
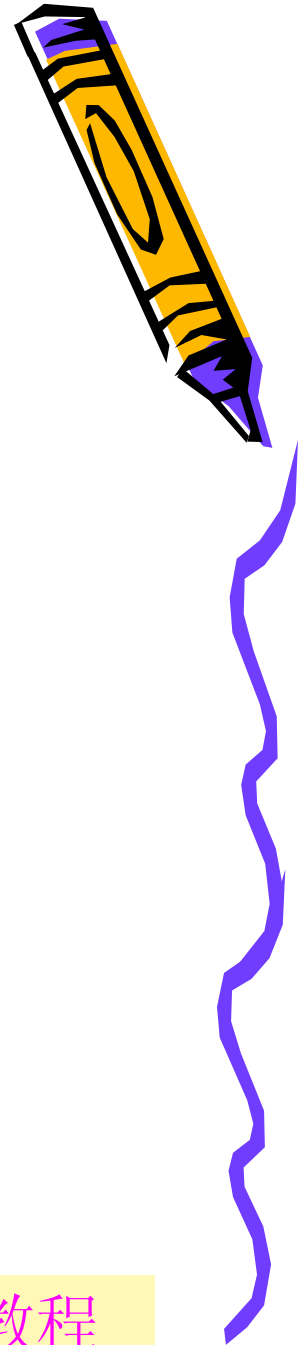
例如，用户定义数据类型为student，使用该变量代码如下：

```
dim stu as student  
stu.xh="080101"  
stu.xm="李四"  
stu.xb="女"  
stu.n1=19
```



用with语句简化变量的赋值，每个分量从点开始写，省略变量名。如：

```
dim stu as student
with stu
    .xh="080101"
    .xm="李四"
    .xb="女"
    .nl=19
end with
```



8.2.9 VBA运算符

VBA的运算符有4种：算术运算符、关系运算符、逻辑运算符、连接运算符。

1. 算术运算符：乘幂（^）、乘法（*）、除法（/）、整数除法（\）、求模（mod）、加法（+）、减法（-）。
2. 关系运算符：相等=、不相等<>、大于>、大于等于>=、小于<、小于等于<=。
对两个操作数进行比较，得到一个逻辑值。
3. 逻辑运算符
与and、或or、非not。
对两个逻辑值进行比较，结果仍然是逻辑值。
4. 连接运算符：&、+
&强制将两个操作数作为字符串连接。+只对字符串进行连接。





运算符的优先级：

括号>算术运算符>连接运算符>关系运算符>逻辑运算符。

优先级相同时，运算顺序从左到右。

5. 对象运算符

对象运算符用来指示随后出现的项目类型，有叹号运算符

(!) 和点运算符 (.)。

(1) 叹号运算符 !，引用一个窗体、报表或控件。

例如：forms!学生信息窗体!t1

(2) 点运算符 .，引用对象的属性。

例如：Me!t1.forecolor=255



8.3 VBA内置函数

第3章中介绍过的函数在此不作详细介绍。

8.3.1 VBA常用函数

1. 算术函数

(1) `abs(x)`，返回 x 的绝对值。

(2) `int(x)`，返回 x 的整数部分， $x < 0$ 时，返回值小于等于 x 。

`fix(x)`，返回 x 的整数部分， $x < 0$ 时，返回值大于等于 x 。

(3) `sqr(x)`，返回 x 的平方根。要求 x 大于等于0。

(4) `rnd(x)`，返回0~1之间的随机数，为单精度类型。

说明：若 $x > 0$ ，每次产生不同随机数；若 $x = 0$ ，产生最近生成的随机数；若 $x < 0$ ，每次产生相同随机数。当 $x > 0$ 时可直接写`rnd`，省略括号和参数。



例如：

`int (100*rnd)`，产生0～99之间的随机整数。

`int (101*rnd)`，产生0～100之间的随机整数。

`int (100*rnd+1)`，产生1～100之间的随机整数。

`int (100*rnd+100)`，产生100～199之间的随机整数。

`int (101*rnd+100)`，产生100～200之间的随机整数。

(5) `round(x, n)`，n是小数位数，对x的小数做四舍五入，返回有n位小数的x值。

例如：`round(12.735, 2)`，返回12.74。



2. 字符串函数

(1) `len(x)`，返回x的长度，即字符个数。要求x为字符串类型。

(2) `space(n)`，返回由n个空格组成的字符串，n为数字。

(3) `ucase(x)`，将x中的小写字母转大写。

`lcase(x)`，将x中的大写字母转小写。

(4) `trim(x)`，去掉x的首部和尾部空格。

`ltrim(x)`，去掉x的首部空格。

`rtrim(x)`，去掉x的尾部空格。

(5) `instr(起始位置, 字符串1, 字符串2, 比较方式)`，返回字符串2在字符串1中最早出现的位置，返回值是一个整数。





3. 日期时间函数

(1) weekday(x, n), 返回1~7的整数, 表示星期几。

说明: n为可选项, 默认值1。当n取值为1时, 星期天返回1, 星期一返回2, ..., 依次类推。若n取值为2, 则星期一返回1, 星期二返回2, ..., 星期天返回7。

(2) hour(x), 返回x的小时数。

(3) minute(x), 返回x的分钟数。

(4) second(x), 返回x的秒数。





4. 类型转换函数

(1) `asc(x)`，返回首字符的ASCII码，其中，`x`是字符串或字符串变量名。

说明：ASCII码是美国标准信息转换码，共128个字符，对应十进制数0~127。

(2) `chr(n)`，将数字`n`转换成相应字符，`n`的取值范围0~127。

(3) `str(n)`，将数字转换为字符串，`n`是数字或数字表达式。

说明：当数字转换为字符串时，总会在前头留一个符号位，如果是正数，符号位显示空格。

(4) `val(x)`，将数字型字符串转换为数字，`x`是数字型字符串。

说明：转换时自动将空格、制表符、换行符去掉，当遇到第一个不能识别为数字的字符时即停止读入。



8.3.2 输入输出函数

1. 输出函数MsgBox

格式：msgbox（显示信息，按钮数目+图标类型，标题栏字符串）

功是：在对话框中显示消息，等待用户单击按钮，返回一个整数，该值代表用户单击了哪个按钮。

返回值	单击的按钮
1	确定
2	取消
3	放弃
4	重试
5	忽略
6	是
7	否





2. 输入函数inputbox

格式：inputbox（提示信息，标题栏字符串，默认值）

功能：提示用户输入一个字符串或数值。

说明：

（1）“提示信息”是必选项，是一个字符串，显示在输入框中。

（2）“标题栏字符串”是可选项，省略此项，标题栏将显示应用程序名。

（3）默认值是可选项，是字符串或数字，省略此项，系统会自动匹配数据类型。

（4）函数返回值是在输入框中输入的数字或字符串。



8.3.3 计算外部数据源数据的函数

除了在报表中介绍过的Dlookup函数, 还有以下函数:

1. Davg函数

格式: Davg ("字段名", "表名", "条件表达式")

功能: 对指定字段求平均值。

说明: 若省略条件表达式, 对全体字段求平均值。

例如: Davg ("奖金", "工资", "基本工资>=2000")

2. Dsum函数

格式: Dsum ("字段名", "表名", "条件表达式")

功能: 对指定字段求和。

说明: 若省略条件表达式, 对全体字段求和。

例如: Dsum ("奖金", "工资")



3. Dcount函数

格式：Dcount（"字段名"，"表名"，"条件表达式"）

功能：对指定字段统计个数。

说明：若省略条件表达式，对全体字段统计个数。

例如：Dcount（"教师编号"，"教师"，"性别='女'"）



例:统计外部表数据

外部数据:

教师编号	基本工资	奖金
1001	2100.00	1575.00
1002	1900.00	1350.00
1003	2600.00	1200.00
1004	2200.00	1575.00
1005	2700.00	1200.00
2001	2000.00	1575.00
2002	2900.00	1200.00
2003	1800.00	1125.00

(1) 在窗体建3个文本框→名称为t1、t2、t3→附加标签的标题分别为“人数总计”、“工资合计”、“平均奖金”。

(2) 在窗体建立命令按钮→标题为“计算”→名称为c1。



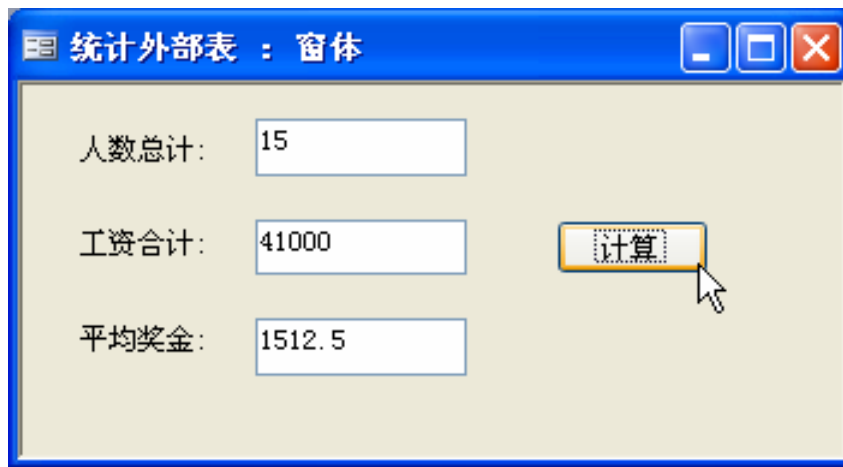
(3) c1的click事件代码:

```
t1 = DCount("教师编号", "工资")
```

```
t2 = DSum("基本工资", "工资")
```

```
t3 = DAvG("奖金", "工资")
```

(4) 转到窗体视图→单击命令按钮→各数据显示在文本框中。



人数总计:	15
工资合计:	41000
平均奖金:	1512.5

计算



8.3.4 处理空值的函数 Nz

Nz函数可以将Null值转换为数字0、空字符串或自定义的返回值。

格式：Nz（变量/表达式/字段属性名[，指定值]）

说明：指定值是可选项。若给出指定值，Nz函数返回指定值。
若省略指定值，当类型为数值型且值为Null，Nz函数返回数字0。当
类型为字符型且值为Null，Nz函数返回空字符串。

例：如果文本框为空，显示提示信息。

- (1) 在窗体建文本框→名称为t1→附加标签标题为“输入数值”。
- (2) 在窗体建标签→名称为b1→标题为“输入的内容为：”
- (3) 在窗体建命令按钮→标题为“验证”→名称为c1。



(4) c1的click事件代码:

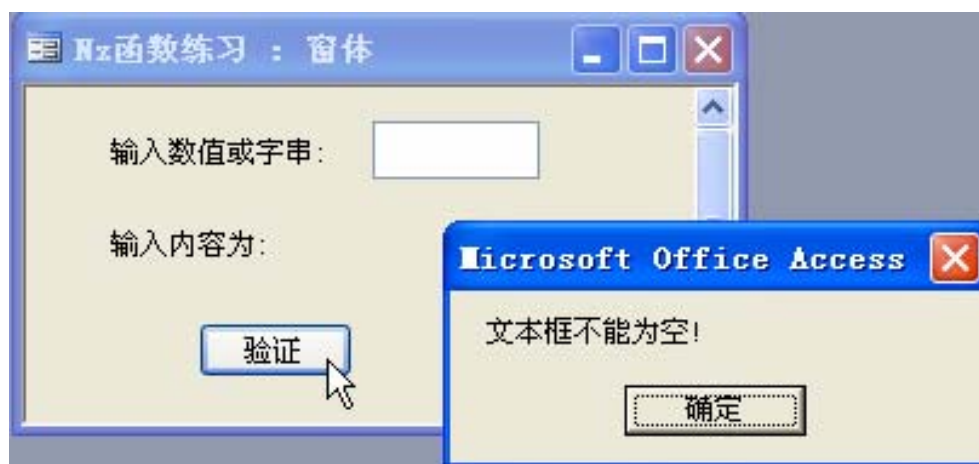
```
Dim aa As String
```

```
aa = Nz(t1.Value, "文本框不能为空!")
```

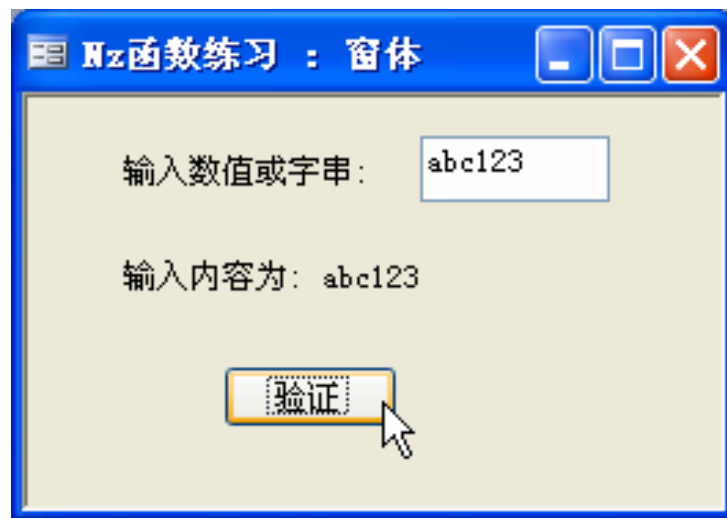
```
MsgBox aa
```

```
b1.Caption = "输入内容为: " & Nz(t1.Value)
```

(5) 转到窗体视图，文本框为空时单击命令按钮，显示消息。



(6) 在文本框中输入字串“abc123”，单击命令按钮，显示结果



8.4 VBA程序结构

VBA程序是语句的集合，语句是一条能够完成某项操作的命令，可以包含关键字、运算符、变量、常数和表达式，程序用来告诉计算机完成指定任务。

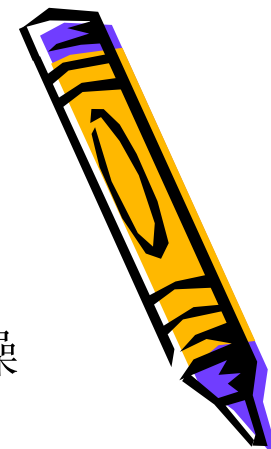
8.4.1 程序构成

1. 语句分类

语句按功能分为两大类：声明语句和执行语句。

声明语句用来定义变量、常量、过程，并指定数据类型。

执行语句进行赋值操作、调用过程、实现各种流程控制。





2. 程序的三种结构

程序有三种基本结构：顺序结构、选择结构、循环结构。

顺序结构，按语句排列顺序依次执行。

选择结构，又称为条件结构，根据不同条件选择执行不同操作。

循环结构，重复执行某段程序代码。

3. 程序书写规则

- (1) 一条语句一行写不下时，用续行符“_”作为第一行结尾，将剩余语句写在下一行。
- (2) 语句较短时可以几条语句写在一行，语句之间用冒号分隔。
- (3) 如果一行语句输入完成后显示为红色，表示该语句存在错误。
- (4) 尽量使用提示信息。
- (5) 代码不区分大小写。



4. 注释语句

注释语句是非执行语句，用来提高程序的可读性，不被解释和编译。注释语句显示为绿色。

格式1: rem 注释内容

说明：用rem引导的注释语句，如果放在其他语句后面，之间用冒号分隔。

格式2: ' 注释内容

说明：用单引号引导的注释语句，放在其他语句后面时无需使用冒号分隔。

例: dim a1 as integer, a2 as integer : rem 定义两个整型变量

a1=12 ' 给两个变量赋值

a2=34

a1=a1+a2 ' 将两个变量的和赋给变量a1

rem 用消息框显示结果

msgbox "a1+a2=" & a1 , vbInformation, "消息框"



5. 声明语句

声明语句是用来定义常量、变量、数组和过程。定义这些的同时，也定义了初始值、生命周期、作用域等内容。

初始值由数据类型决定，

如：integer, string。

生命周期由定义的位置决定，

如：局部、模块、全局。

作用域由定义时所使用的关键字决定。

如：dim、public。



6. 赋值语句

赋值语句用来为变量指定一个值。

格式：变量名=值或表达式

说明：

- (1) 格式中的等号(=)称为赋值号，与数学中等号意义不同。如表达式 $a=a+1$ 在数学中不能用，在赋值语句中常用。
- (2) 赋值号左边只能是变量名，不能是常量和表达式。
- (3) 赋值语句有计算和赋值双重功能，将赋值号右边的计算结果赋给赋值号左边的变量。
- (4) 赋值号两边要类型匹配。例如，表达式 $a\%="abc"$ 会返回错误提示，因为该操作把字符串赋给整型变量，类型不匹配。



8.4.2 条件语句

1. if - end if 语句

格式: if 条件 then

语句序列

end if

功能: 若条件成立, 执行语句序列, 否则, 什么也不做。

说明:

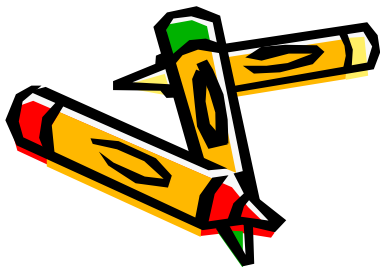
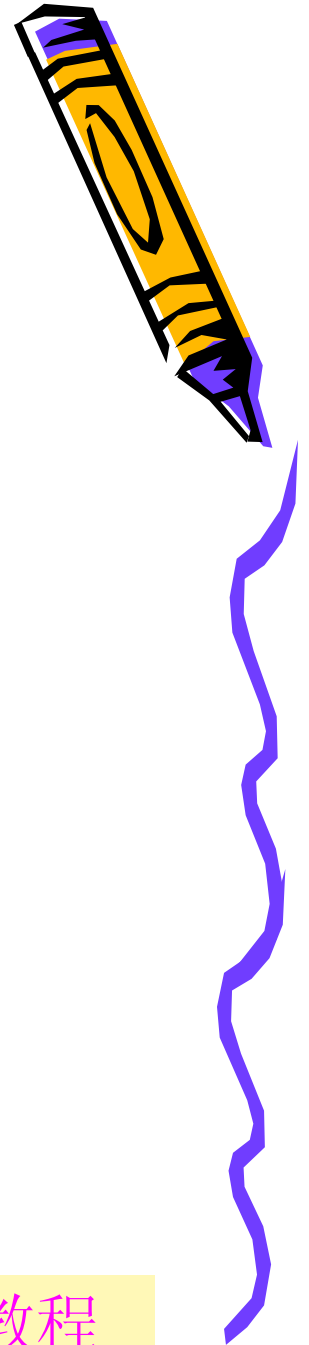
如果语句序列只有一条语句, 可简写在一行, 并省略end if语句。

例: dim a as integer, b as integer

a=12: b=8

if a<b then a=b

msgbox "最大值为: " & a, vbInformation, "消息框"



2. if - else - end if 语句

格式: if 条件 then

语句序列1

else

语句序列2

end if

功能: 若条件成立, 执行语句序列1, 否则, 执行语句序列2。

例: dim a as integer, b as integer, c as integer

a=12: b=8

if a<b then

c=b

' 用c存放最大值

else

c=a

end if

msgbox "最大值" & c, vbInformation, "消息框"



ACCESS实例教程

8.4.3 多项选择语句

1. 格式

```
select case 表达式
case 值1
    语句序列1
case 值2
    语句序列2
...
case 值n
    语句序列n
case else
    语句序列n+1
end select
```

2. 功能：首先计算表达式的值，然后将表达式值与每个case的值比较，如果找到匹配的值，执行该case后的语句块，如果没有相匹配的值，执行case else后面的语句。





3. 说明

(1) 多个分支中只能选择执行一个，执行了第一个符合条件的分支以后，即使有其他分支符合条件也不再执行。

(2) select case后面的表达式通常是一个变量的名字。

(3) select case与end select要成对出现，end select之间有空格。

(4) case后面的值有4种写法：

- 单一数值
- 一行并列数值，之间用逗号分隔。
- 数值1 to 数值2，前一个值必须比后一个值小。
- 用is开头的简单条件式，如is>10。不允许复杂条件式。

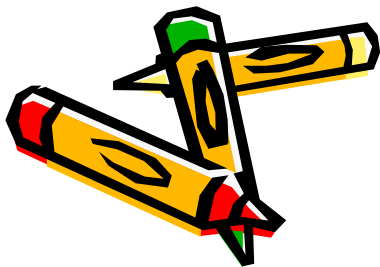


例：select case练习

(1) 建立文本框→名称为t1→建立命令按钮→名称分别为c1。

(2) c1的click事件代码：

```
Dim aa As String, bb As String  
aa = t1  
Select Case aa  
Case "a" To "z": bb = "英文字母"  
Case "0" To "9": bb = "数字"  
Case "!", "?", ":", ".", ",", ";": bb = "标点符号"  
Case Else: bb = "特殊字符"  
End Select  
MsgBox bb
```



(3) 转到窗体视图→文本框中输入字母→单击按钮
显示消息框 。



ACCESS实例教程

8.4.4 选择函数与计时器事件

1. iif函数

格式: `iif(条件, 表达式1, 表达式2)`

功能: 如果条件为真, 函数值为表达式1的值, 否则, 函数值为表达式2的值。

例: `y=iif(a>b, "a大", "b大")`

2. switch函数

格式: `switch(条件1, 表达式1, 条件2, 表达式, ..., 条件n, 表达式n)`

功能: 从左到右依次判断, 遇到第一个条件为真时返回对应表达式的值。

说明: 条件与表达式成对, 当多个条件为真时, 只取第一个条件的对应表达式。

例: `y=switch(x<0, -1, x=0, 1, x>0, 1)`



3. choose函数

格式: choose(索引式, 值1, 值2, ..., 值n)

功能: 根据索引项的值, 返回对应的值。

说明: 索引项通常是整型变量名, 取值范围1~n, 超出范围时返回NULL。

例如: y=choose(x, "优秀", "良好", "中等", "及格", "不及格")

4. 计时器事件

VBA没有直接提供时间控件, 通过timer事件实现定时功能。

方法: 首先设置窗体的计时器间隔属性TimerInterval, 然后给timer事件写过程代码。打开窗体时, 每隔一个时间间隔激发一次timer事件, 事件的过程就被执行一次, 从而实现“定时”处理功能。

计时器间隔的时间单位为毫秒, 1000毫秒=1秒。



例:显示电子表

- (1) 建立2个标签→名称为b1和b2→标题都为数字1。
- (2) 打开窗体的属性窗口→单击“事件”选项卡→设置计时器间隔属性为1000。
- (3) 在VBE窗口“通用-声明”写代码:

```
Public a As Boolean      ' a为逻辑型, 默认false
```

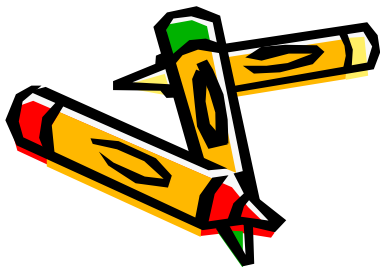
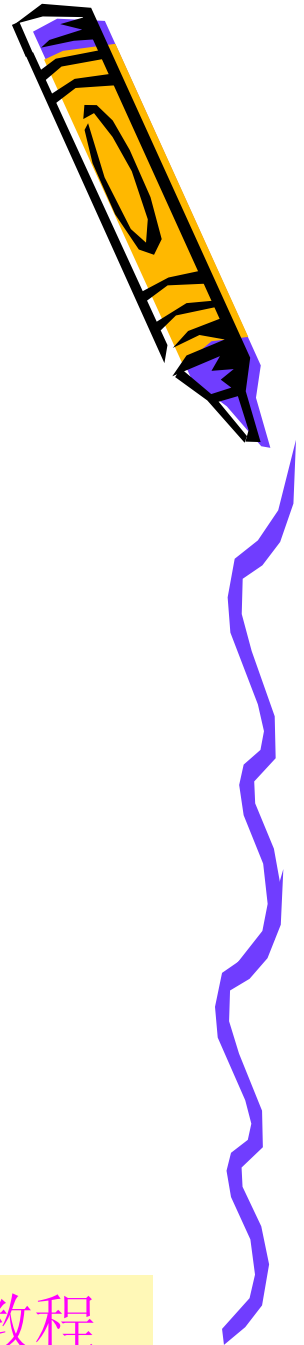
- (4) Form的Timer事件代码:

```
a = Not a
```

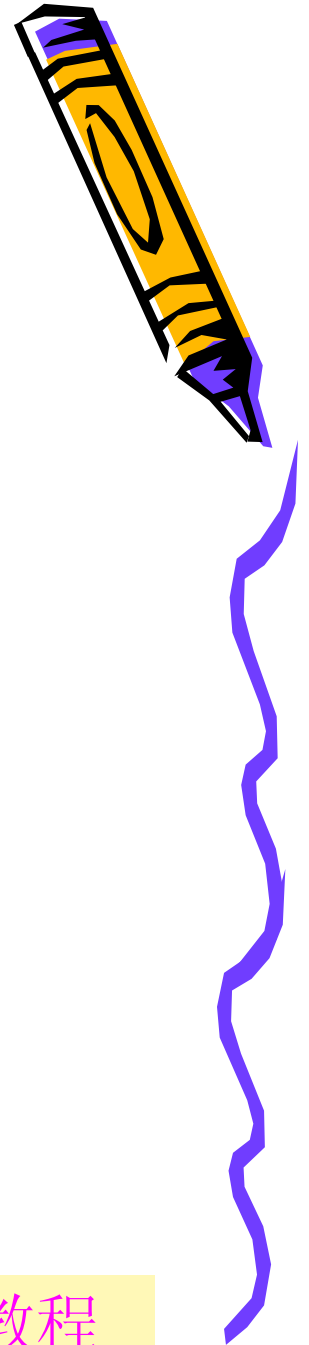
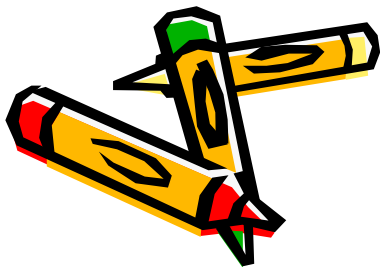
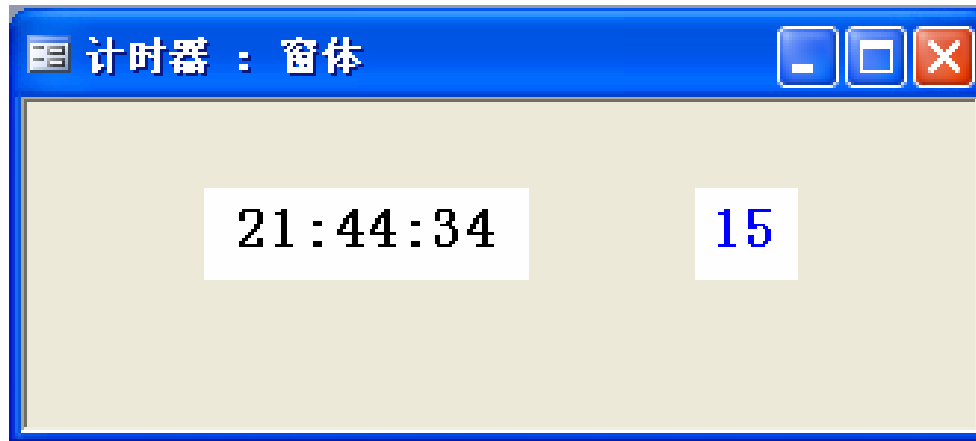
```
b1.Caption = Time()
```

```
b2.Caption = CInt(b2.Caption) + 1
```

```
b2.ForeColor = IIf(a = True, 255, 16711680)
```



(5) 转到窗体视图。b1中显示系统当前时间，b2中的数字每秒增加1，数字的颜色红、蓝交替，每秒换一次。



8.4.5 验证函数

VBA的常用验证函数

函数名	功能
IsNumeric(x)	验证x是否为数值，若返回true，是数值
IsDate(x)	验证x是否为日期值，若返回true，是日期或可识别的有效日期
IsNull(x)	验证x是否为无效值，若返回true，是无效值
IsEmpty(x)	验证x是否已被初始化，若返回true，未被初始化
IsArray(x)	验证x是否为数组，若返回true，是数组
IsError(x)	验证x是否为一个错误值，若返回true，有错误
IsObject(x)	验证x是否为对象变量，若返回true，是对象



例:检验字符串类型

- (1) 建文本框→名称为t1→附加标签标题为“输入年龄：”。
- (2) 建命令按钮→名称分别为c1→标题为“验证”。
- (3) c1的click事件代码：

```
If t1 = "" Or IsNull(t1) Then  
MsgBox "不能为空!", vbCritical, "提示"  
ElseIf IsNumeric(t1) = False Then  
MsgBox "必须是数字!", vbCritical, "提示"  
ElseIf t1 <= 0 Or t1 >= 150 Then  
MsgBox "超出范围!", vbCritical, "提示"  
Else  
MsgBox "验证通过!", vbInformation, "提示"  
End If
```



(4) 转到窗体视图→在文本框中输入一个字母→单击“验证”按钮→消息框显示消息“必须是数字!”。



8.4.6 循环语句

当某一程序段需要反复执行，用循环结构实现。
循环结构对应两类循环语句：

先判断后执行的循环语句（当型循环结构）

先执行后判断的循环语句（直到型循环结构）





1. for-next循环

for-next循环能使语句序列运行指定次数，循环中有一个计数器变量，变量的值随每一次循环增加或减少。

for-next是当型循环结构，先判断后执行。

(1) 格式：for 循环变量=初值 to 终值 step 步长
语句序列
next

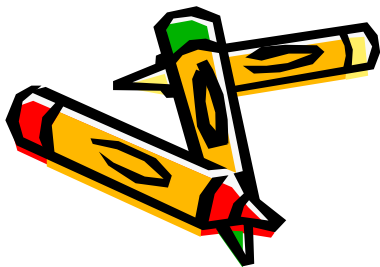
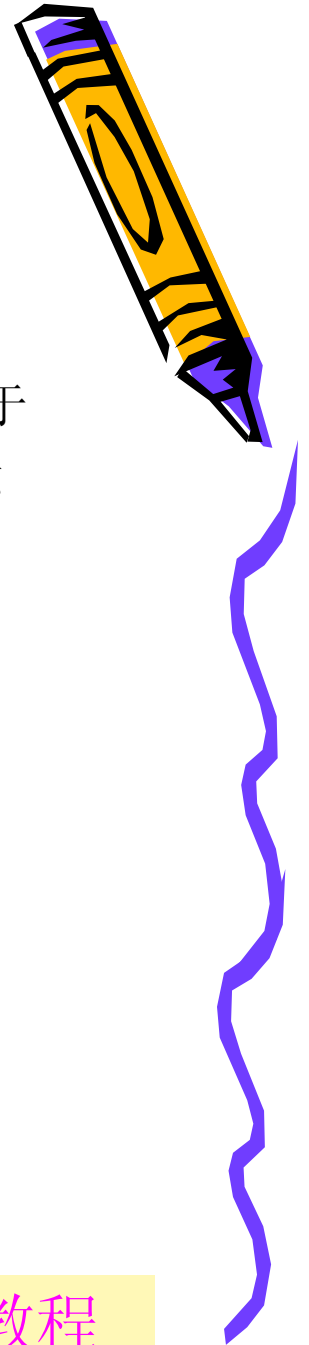
(2) 功能

先将初值赋给循环变量，再将循环变量的当前值与终值做比较，如果比较结果为真执行语句序列，增加一个步长，再进行比较，如果比较结果仍为真，继续循环……。如果比较结果为假，结束循环。



(3) 说明

- 步长大于0时判断循环变量的当前值是否大于终值，步长小于0判断循环变量的当前值是否小于终值。步长为0时导致循环无法结束，所以步长不要设置为0。
- 步长可以是整数或小数，步长为1时可以省略。
- 除第一次循环以外，其他循环增加一个步长后与终值比较。
- for循环可以嵌套。
- 在for循环中可以用exit for语句强行中止循环。



练习：阅读下面程序段，循环结束后，各变量的值是多少？

```
dim s as integer, i as integer  
s=0  
for i=1 to 10 step 2  
    s=s+i  
    i=i+2  
next
```

结论：循环结束后，变量i的值是13，变量 s 的值是15。



例：显示Fibonacci数列的第15个元素

(Fibonacci数列的前两个数都是1，从第3个数开始，每个数都是前两个数之和。)

(1) 建立文本框→名称为t1→建立命令按钮→名称为c1。

(2) c1的click事件代码：

```
Dim f(15) As Integer  
f(1) = 1: f(2) = 1  
For i = 3 To 15  
    f(i) = f(i - 1) + f(i - 2)  
Next  
t1 = f(15)
```



2. do-while-loop循环

do-while-loop循环是当型循环结构，先判断后执行。

(1) 格式

do while 循环条件

语句序列

loop

(2) 功能

先检查循环条件是否成立，若条件为真，执行语句序列。遇到loop语句时返回循环开始处重新判断，若条件仍然为真，再次执行语句序列……。当条件为假时退出循环。



(3) 说明

- 可以用exit do强行中止循环。
- 要考虑循环变量赋初值和给循环变量增加步长的问题。
- 如果将while换成until, 当条件为真时循环结束。

练习：阅读下面的程序（注：大写字母A的ASCII码是65。）

```
dim a(26) as string, b(26) as integer,
```

```
dim i as integer
```

```
i=1
```

’ 给循环变量赋初值

```
do while i<=26
```

```
a(i)=chr(i+64)
```

’ 将ASCII码变为字母

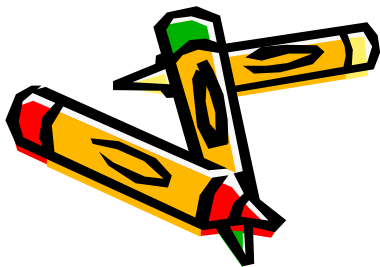
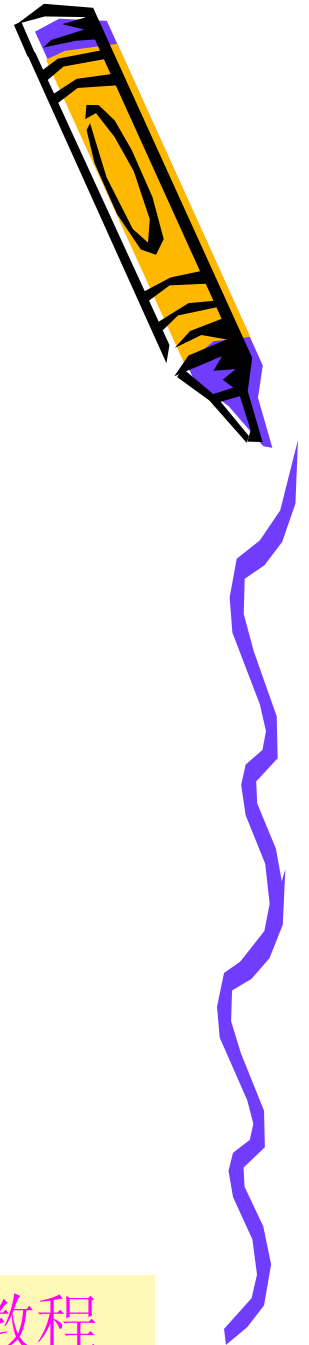
```
b(i)=Lcase(a(i))
```

’ 将大写字母变为小写字母

```
i=i+1
```

’ 使循环变量增加1

```
loop
```



3. do-loop-while循环

是直到型循环结构，先执行后判断。

(1) 格式

do

语句序列

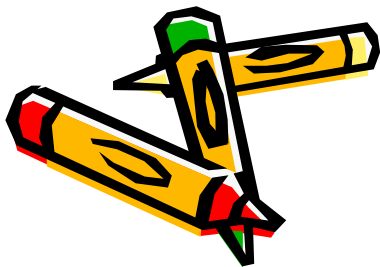
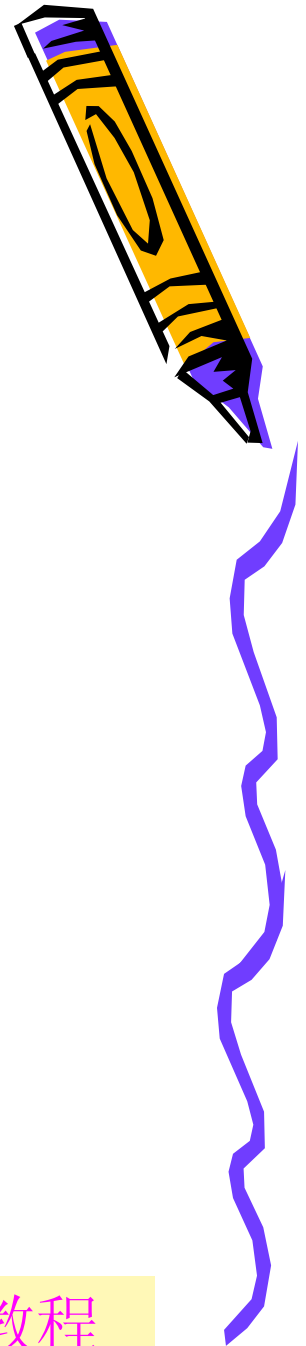
Loop while 循环条件

(2) 功能

先执行语句序列，遇到loop语句时判断循环条件，若条件为真，再次执行语句序列……。当条件为假时退出循环。

(3) 说明

- 可以用exit do强行中止循环。
- 要考虑给循环变量赋初值和给循环变量增加步长的问题。
- 如果将while换成until，当条件为真时循环结束。



4. for-each 循环语句

for each循环用来遍历数组或对象集合中每一个元素，不需要指定循环次数，每个元素只遍历一次。如果不知道一个集合有多少个元素，用for each循环非常方便。

(1) 格式

```
for each 循环变量 in 集合或数组  
    语句序列  
next
```

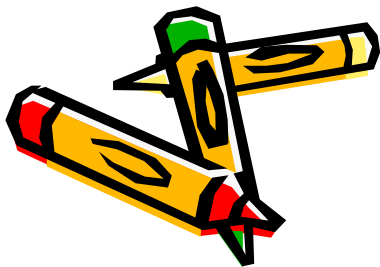
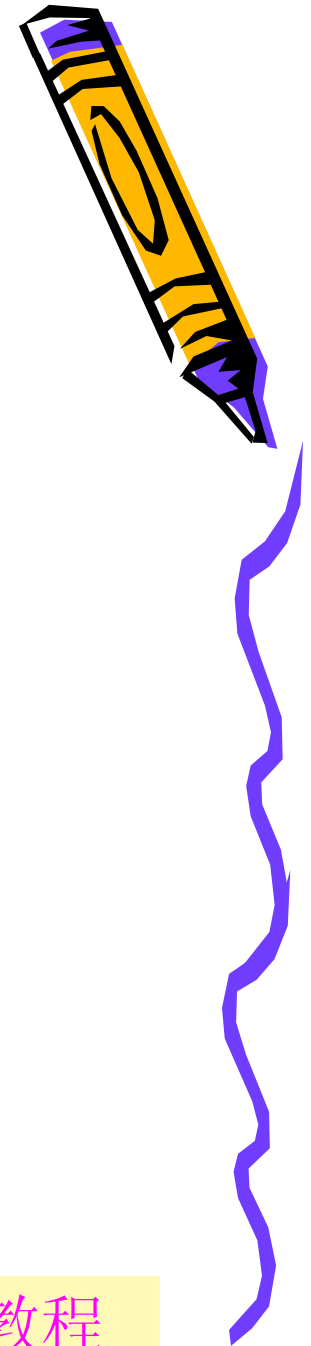
(2) 功能

用循环变量依次遍历数组或对象集合中每一个元素，直到遍历结束。



练习：阅读下面程序。

```
dim a(50) as integer, i as integer, s as integer
for i=1 to 50          ' 给数组赋值
    a(i)=i
next
s=0
for each i in a        ' 遍历a中元素
    if i mod 2<>0 then s=s+i ' 计算奇数之和
next
```



ACCESS实例教程

8.5 过程调用与参数传递

8.5.1 认识过程和参数

1. 过程

过程是用来执行特定任务的一段独立的程序代码，这段代码能被反复调用。VBA的模块以过程为单元组成。

VBA的过程根据是否返回值分为两类：Sub过程和Function过程。Sub过程只执行操作不返回值，不能用在表达式中，调用时就象使用基本语句一样。Function过程又称为用户自定义函数，执行操作后返回结果，常用在表达式中，调用时就象使用基本函数一样。

过程名是标识符，不要与模块名重名，否则调用时会出现混乱。同一模块中，Sub过程也不要与Function过程重名。

过程不能嵌套定义，但可以嵌套调用。



2. 参数

如果在调用过程中，主调方（调用过程的语句）与被调方（过程）存在数据传递关系，表现这种传递关系的数据就是参数。

参数分为形参（形式参数）和实参（实际参数）。形参用在被调方，只能是变量名或数组名。实参用在主调方，可以是常量、已赋值的变量、有计算结果的表达式。

当形参和实参都是变量时，存在两种参数传递方式：值传递与地址传递。值传递只能把实参的值传给形参，是“单向传递”。地址传递能在实参与形参之间实现参数的“双向传递”，不但将实参的值传给形参，形参的值也可以传给实参。



8.5.2 sub过程

sub 过程是包含在 sub 和 end sub 之间的一组代码，调用 sub 过程时只执行其中的操作，不返回值。

1. 定义sub过程的格式

```
sub 过程名 (形参1 as 数据类型, 形参2 as 数据类型, ...)  
    语句序列  
end sub
```

2. 调用sub过程

格式1: call 过程名 (实参1, 实参2, ...)

格式2: 过程名 实参1, 实参2, ...



3. 说明

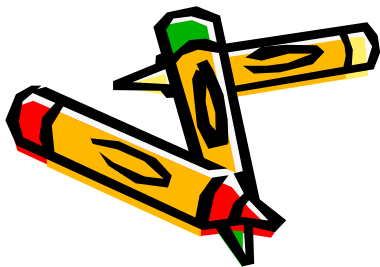
- (1) 参数之间用逗号分隔，对应的形参与实参之间要类型匹配。
- (2) 用格式1调用sub过程必须加括号，用格式2调用sub过程不加括号。
- (3) 用 `exit sub` 语句立即从 `sub` 过程中退出。
- (4) 定义sub过程时即使无任何参数，也必须包含空括号（）。
- (5) sub之前可以用public或private或static定义过程作用域。
- (6) 标准模块中的过程可以被所有对象调用，类模块中的过程只在本模块中有效。



例：sub过程练习

- (1) 建2个文本框→名称分别为t1和t2→附加标签标题分别为“输入矩形长：”和“输入矩形宽：”。
- (2) 建命令按钮→名称为c1→标题为“计算”。
- (3) 建立标准模块→以“过程模块”为名保存模块。
- (4) 在“过程模块”中定义jxmj过程，代码如下：

```
Public Sub jxmj (h As Integer, w As Integer)
    Dim s As Integer
    s = h * w
    MsgBox "矩形面积为:" & s
End Sub
```



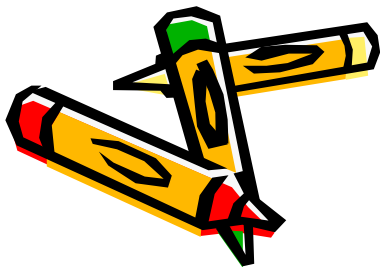
(5) 命令按钮c1的click事件代码:

```
Dim a1 As Integer, a2 As Integer
```

```
a1 = t1:a2 = t2
```

```
Call jxmj (a1, a2)          ' 用格式1调用sub模块
```

(6) 转到窗体视图→输入矩形长和宽→单击“计算”按钮。



ACCESS实例教程

8.5.3 function过程

1. 定义function过程的格式

```
function 过程名 (形参1, 形参2, ...) as 数据类型  
    语句序列  
    过程名=表达式  
    .....  
end function
```

2. 调用function过程

调用function过程只有一种方式，即直接引用过程名，而且过程名在赋值号右端或在表达式中。



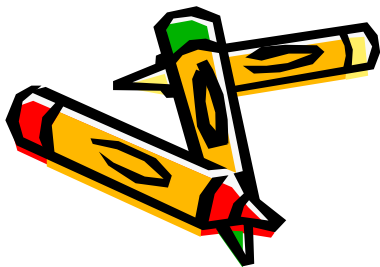
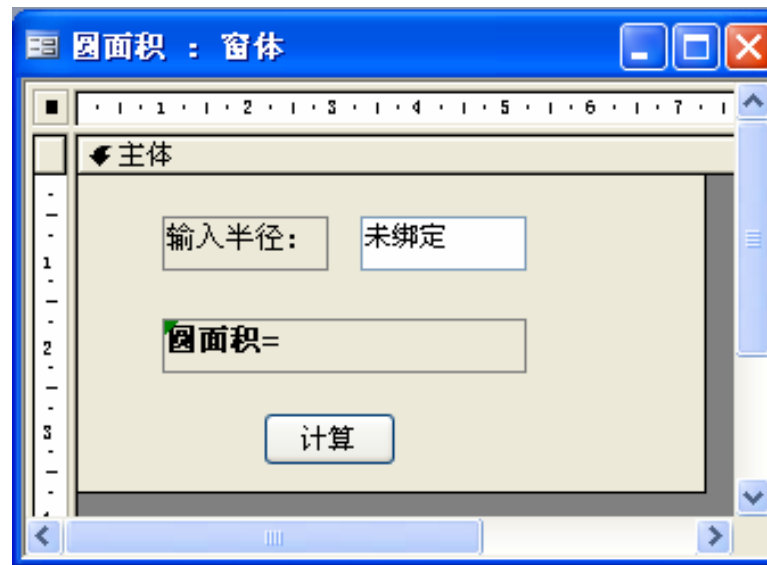
3. 说明

- (1) 参数之间用逗号分隔，对应形参与实参之间要类型匹配。
- (2) “过程名=表达式”是function过程中不可缺少的语句。
- (3) 用 `exit function` 语句立即从 `function` 过程中退出。
- (4) 若function过程无任何参数，也必须包含空括号（）。
- (5) function之前可以用`public`或`private`或`static`定义过程的作用域。



例：function过程练习

- (1) 建文本框→名称为t1。
- (2) 建标签→名称为Lab→标题为“圆面积=”
- (3) 建命令按钮→名称为c1→标题为“计算”。



ACCESS实例教程

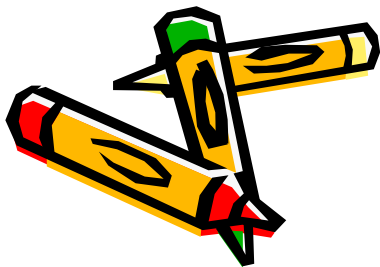
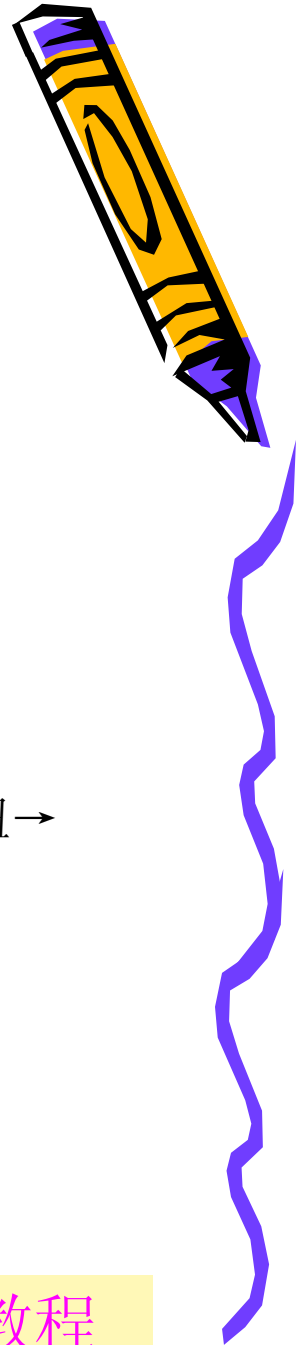
(4) 在已有的“过程模块”中添加ymj过程，代码如下：

```
Public Function ymj (r As Integer) As Single  
    ymj = r * r * 3.14  
End Function
```

(5) 命令按钮c1的click事件代码：

```
Dim a As Integer  
a = t1  
Lab.Caption = Lab.Caption & ymj (a)
```

(6) 转到窗体视图→在文本框中输入半径→单击“计算”按钮→
计算结果显示在标签中。



8.5.4 参数传递

当实参是变量名时，既可以用传值方式传递参数，也可以用传地址的方式传递参数。

1. 参数的值传递

如果在过程的形参前加 `byval` 说明符，参数的传递方式为“传值”。值传递的含义是指在过程中另外开辟存储单元存放从实参传过来的值，一旦过程结束，过程中开辟的存储单元被释放，该单元数据的改变不会保留下来。



2. 参数的地址传递

如果在过程的形参前加 `byref` 说明符，参数的传递方式为“传地址”，参数的地址传递是默认选项，不加任何说明符，默认就是传地址。

地址传递的含义是指过程中的形参与主调方的实参指向同一个存储单元，在过程中对存储单元所做的修改，当过程结束后成为实参的当前值。

地址传递通常用在sub过程中。



例：参数传递练习

- (1) 建立2个文本框→名称分别为t1和t2。
- (2) 建立命令按钮→名称为c1→标题为“调用过程”。
- (3) 在已有的“过程模块”中添加cscd过程，代码如下：

```
Public Sub cscd (ByRef a As Integer, ByVal b As Integer)
a = a + 10
b = b + 10
End Sub
```



(4) 命令按钮c1的click事件代码:

```
Dim x As Integer, y As Integer
```

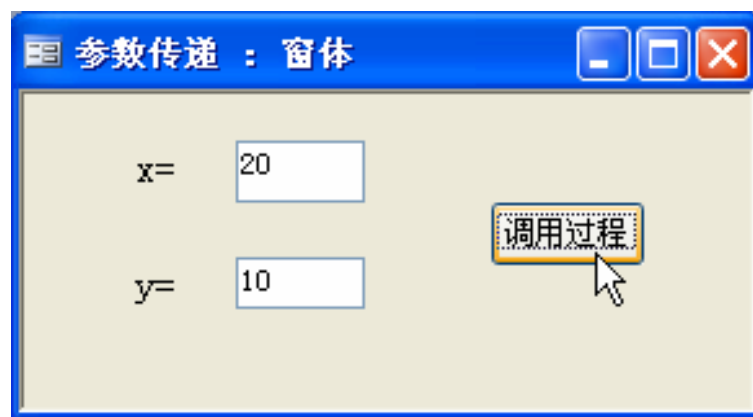
```
x = t1:y = t2
```

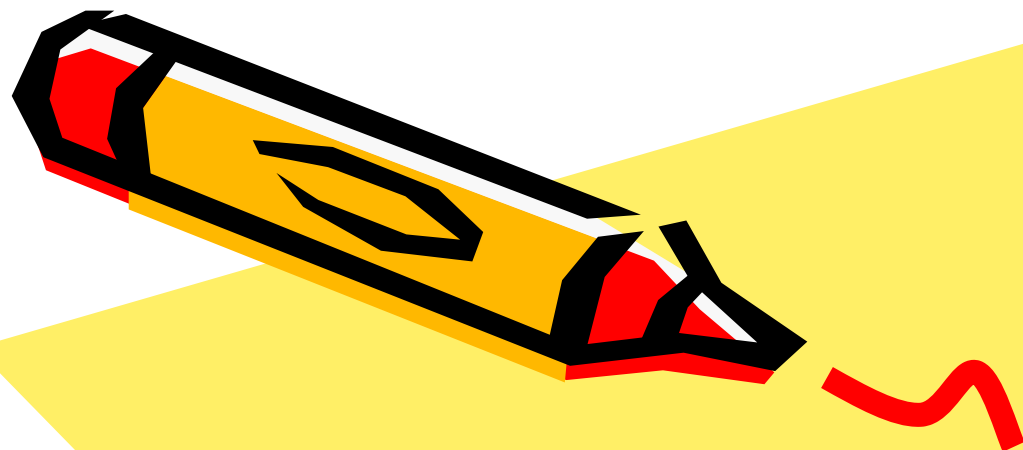
```
cscd x, y
```

’ 用格式2调用sub过程

```
t1 = x:t2 = y
```

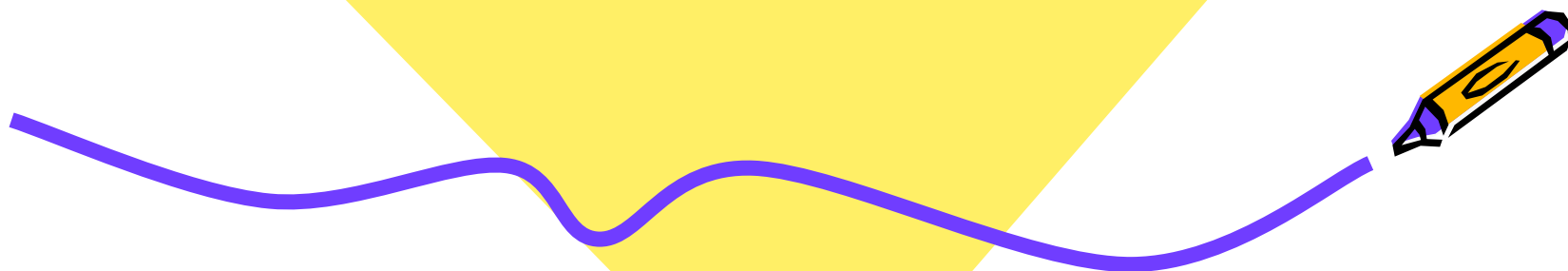
(5) 转到窗体视图→在两个文本框中都输入10→单击“调用过程”按钮→计算结果仍显示在文本框中。





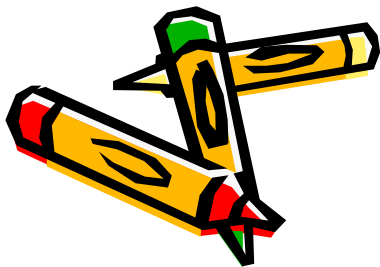
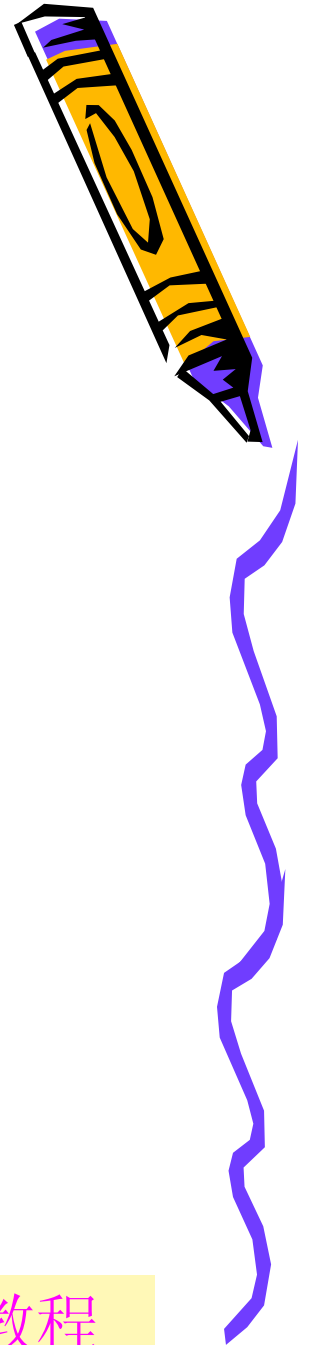
ACCESS实例教程

——数据库编程（第9章）



本章要点:

- 1, 用ADO访问数据库
- 2, 用DAO访问数据库
- 3, 程序运行错误处理



9.1 VBA数据库访问接口

VBA数据库访问接口是指VBA与后台数据库的连接部分，也就是VBA与Access数据库连接的方法。

9.1.1 三种数据库访问接口

1. ODBC (Open Database Connectivity) ,

ODBC称为“开放式数据库连接”，是一种关系数据源的接口界面。ODBC基于SQL (Structured Query Language) , 把SQL作为访问数据库的标准，一个应用程序通过一组通用代码访问不同的数据库管理系统。ODBC可以为不同的数据库提供相应的驱动程序。

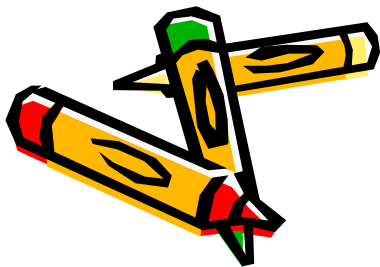
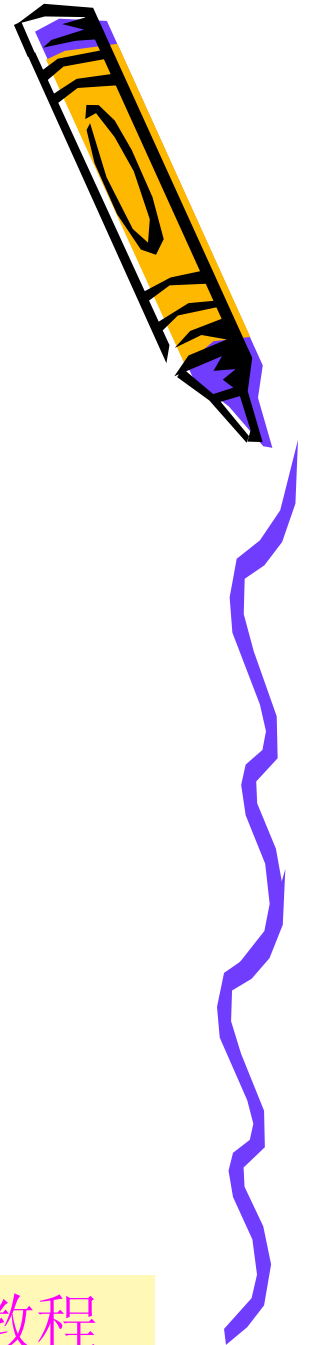


2. DAO (Data Access Objects)

DAO称为“数据访问对象”，是一种面向对象的界面接口，提供一个访问数据库的对象模型，用其中定义的一系列数据访问对象，实现对数据库的各种操作。使用DAO的程序编码非常简单。

3. ADO (ActiveX Data Objects)

ADO称为“Active数据对象”，是基于组件的数据库编程接口。ADO实际是一种提供访问各种数据类型的连接机制，是一个与编程语言无关的COM (Component Object Model) 组件系统。ADO设计为一种极简单的格式，可以方便地连接任何符合ODBC标准的数据库。



数据库引擎是一组动态链接库DLL（Dynamic Link Library），在程序运行时被连接到VBA，实现对数据库的数据访问功能。是应用程序与物理数据库之间的桥梁。

VBA通过DAO和数据引擎可以识别3类数据库：

1. 本地数据库：即Access数据库。
2. 外部数据库：所有索引顺序访问方法（ISAM）数据库，如VFP。也可以访问文本文件数据库和Microsoft Excel或Lotus1-2-3电子表格。
3. ODBC数据库：符合ODBC标准的C/S数据库，如SQL Server、Oracle。

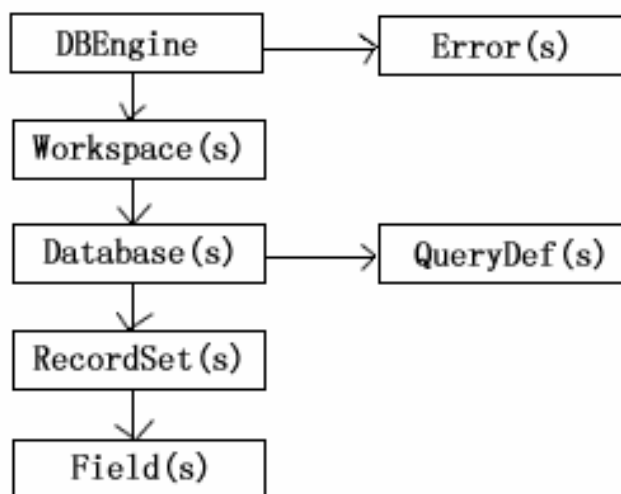


ACCESS实例教程

9.2 用DAO访问数据库

9.2.1 DAO模型结构

DAO模型是设计关系数据库系统结构的对象类的集合，它们提供了管理关系型数据库系统所需的全部操作的属性和方法，这其中包括创建数据库、定义表、字段和索引、建立表之间的关系、定位和查询数据库等。



ACCESS实例教程

3.2 统配符、运算符与常用函数

9.2.2 DAO对象简介

1. DBEngine对象，处于最顶层，表示数据库引擎，是模型中惟一不被其他对象所包含的对象，它包含并控制DAO模型中其他全部对象。
2. Workspace对象，表示工作区，可以使用隐含的workspace对象。
3. Database对象，代表到数据库的连接，表示操作的数据库对象。
4. RecordSet对象，代表一个数据记录的集合，该集合的记录来自于一个表、一个查询或一个SQL语句的运行结果。
5. Field对象，表示记录集中的字段。
6. QueryDef对象，表示数据库查询信息。
7. Error对象，表示数据提供程序出错时的扩展信息。



9.2.3 用DAO访问数据库

用DAO访问数据库时，先在程序中设置对象变量，然后通过对象变量调用访问对象的方法、设置访问对象的属性，从而实现对数据库的各种访问。定义DAO对象要在对象前面加上前缀“DAO”

用DAO访问数据库的一般语句和步骤。

```
Dim ws as DAO.Workspace      ' 定义Workspace对象变量
Dim db as DAO.Database       ' 定义Database对象变量
Dim rs as DAO.RecordSet      ' 定义RecordSet对象变量
Dim fd as DAO.Field          ' 定义Field对象变量
Set ws=DBEngine.Workspace(o) ' 打开默认工作区
Set db=ws.OpenDatabase(数据库的地址与文件名) ' 打开数据库
Set rs=db.OpenRecordSet(表名、查询名或SQL语句) ' 打开记录集
```



Do While not rs.EOF	’ 循环遍历整个记录集直至记录集末尾
.....	’ 对字段的各种操作
rs.MoveNext	’ 记录指针移到下一条
Loop	’ 返回到循环开始处
rs.close	’ 关闭记录集
db.close	’ 关闭数据库
set rs=nothing	’ 释放记录集对象变量所占内存空间
set db=nothing	’ 释放数据库对象变量所占内存空间

说明:

如果是本地数据库, 可以省略定义Workspace对象变量, 打开工作区和打开数据库两条语句用下面一条语句代替:

```
Set db = CurrentDb()
```

该语句是Access的VBA给DAO提供的数据库打开快捷方式。



ACCESS实例教程

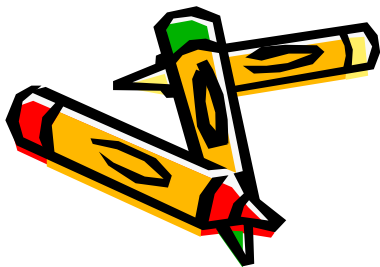
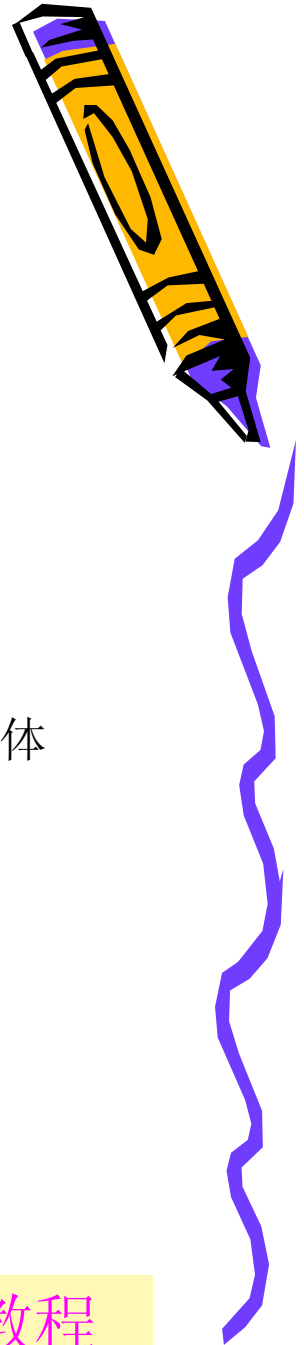
例：用DAO访问数据库

- (1) 在窗体建2个文本框→名称分别为t1和t2→附加标签的标题分别为“姓名”和“工资”。
- (2) 在窗体页眉添加标签→标题为“工资增加10%”。
- (3) 在窗体页脚建命令按钮→名称为c1→标题为“显示第一条记录”。



(4) 给命令按钮的单击事件写代码如下:

```
Dim db as DAO.Database      ' 定义Database对象变量
Dim rs as DAO.Recordset     ' 定义Recordset对象变量
Dim fd as DAO.Field         ' 定义Field对象变量
Set db = CurrentDb()        ' 建立与当前数据库的连接
Set rs = db.OpenRecordset("gz") ' 建立与gz表的连接
Set fd = rs.Fields("工资")   ' 设置对“工资”字段的引用
Do While Not rs.EOF          ' 如果指针没有到最后就执行循环体
    rs.Edit                  ' 使rs处于可编辑状态
    fd = fd + fd * 0.1       ' 给指定字段值增加10%
    rs.Update                 ' 更新表
```



```
rs.MoveNext
```

' 向下移动指针

```
Loop
```

' 返回到循环开始处

```
rs.MoveFirst
```

' 指针移到第一条记录

```
t1 = rs.Fields("姓名")
```

' 将姓名字段的值显示在t1中

```
t2 = rs.Fields("工资")
```

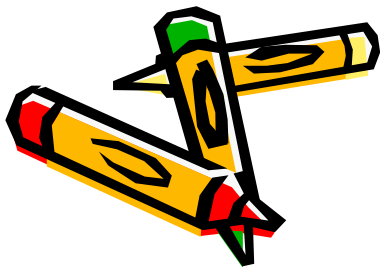
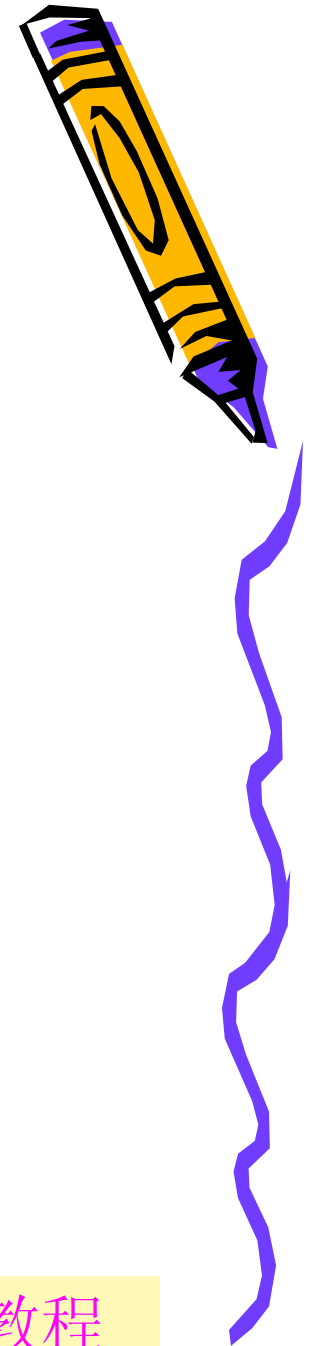
' 将工资字段的值显示在t2中

```
rs.Close
```

```
db.Close
```

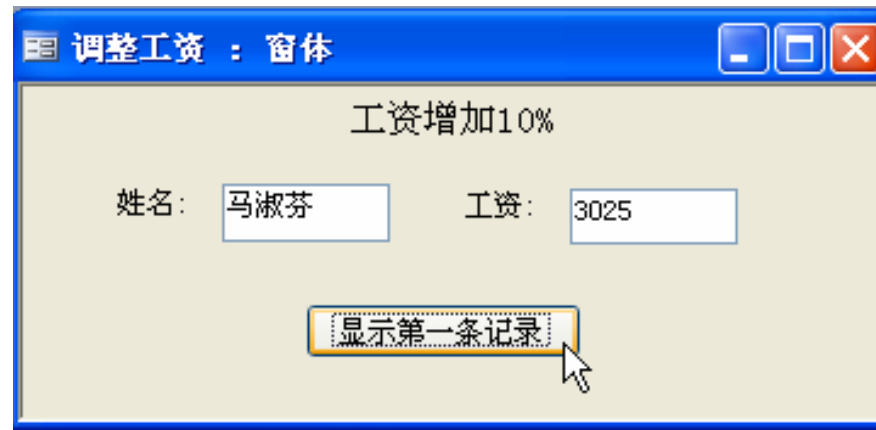
```
Set rs = Nothing
```

```
Set db = Nothing
```



ACCESS实例教程

(5) 转到窗体视图→单击命令按钮，可以看到t2文本框中显示第一条记录更新后的工资值。

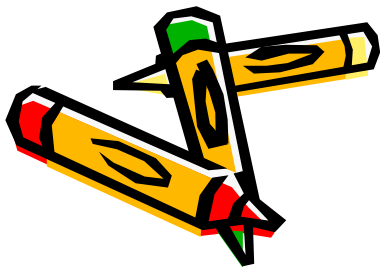
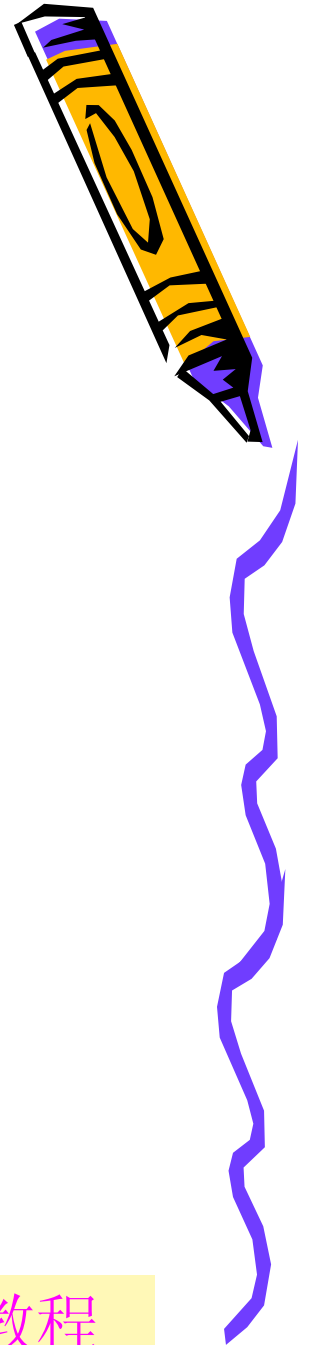


调整工资 : 窗体

工资增加10%

姓名: 马淑芬 工资: 3025

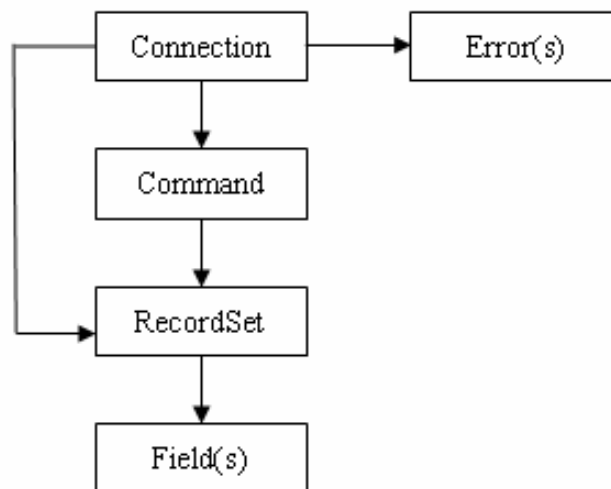
显示第一条记录



9.3 用ADO访问数据库

9.3.1 ADO模型结构

ADO对象模型是一系列对象的集合，对象不分级，除Field对象和Error对象之外，其他对象可直接创建。使用时，通过对象变量调用对象的方法、设置对象的属性，实现对数据库的访问。



ACCESS实例教程

9.3.2 ADO对象简介

1. Connection对象，建立到数据源的连接。
2. Command对象，表示一个命令。
3. RecordSet对象，表示数据操作返回的记录集合。
4. Field对象，表示记录集中的字段。
5. Error对象，表示数据提供程序出错时的扩展信息。

Connection对象与RecordSet对象是两个ADO中最重要的对象。RecordSet对象可以分别与Connection对象和Command对象联合使用。



9.3.3 用ADO访问数据库

首先创建对象变量，然后用对象的方法和属性访问数据库。为了与DAO中同名对象有所区分，ADO对象的前面要加上前缀“ADODB”。用ADO访问数据库的一般语句和步骤：

1. RecordSet对与Connection对象联合使用

```
Dim cn as new ADODB.Connection      ' 建立连接对象
Dim rs as new ADODB.RecordSet        ' 建立记录集对象
cn.Provider="Microsoft.Jet.OLEDB.4.0" ' 设置数据提供者
cn.Open 连接字符串                  ' 打开数据库
rs.Open 查询字符串                  ' 打开记录集
do while not rs.EOF                  ' 循环开始
    .....                           ' 对字段的各种操作
rs.movenext                          ' 记录指针移到下一条
```

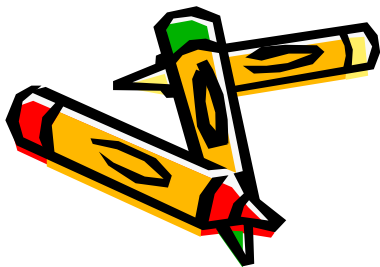


loop	' 返回到循环开始处
rs.close	' 关闭记录集
cn.close	' 关闭连接
set rs=nothing	' 释放记录集对象变量所占内存空间
set cn=nothing	' 释放连接对象变量所占内存空间

说明:

对于本地数据库, Access的VBA也给ADO提供了类似于DAO的数据库打开快捷方式, 可以将设置数据提供者和打开数据库两条语句用下面一条语句代替:

```
Set cn = CurrentProject.Connection()
```

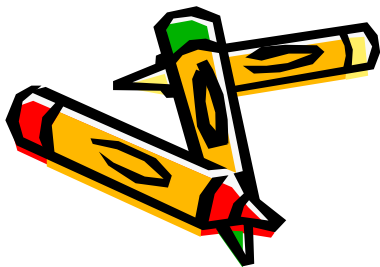
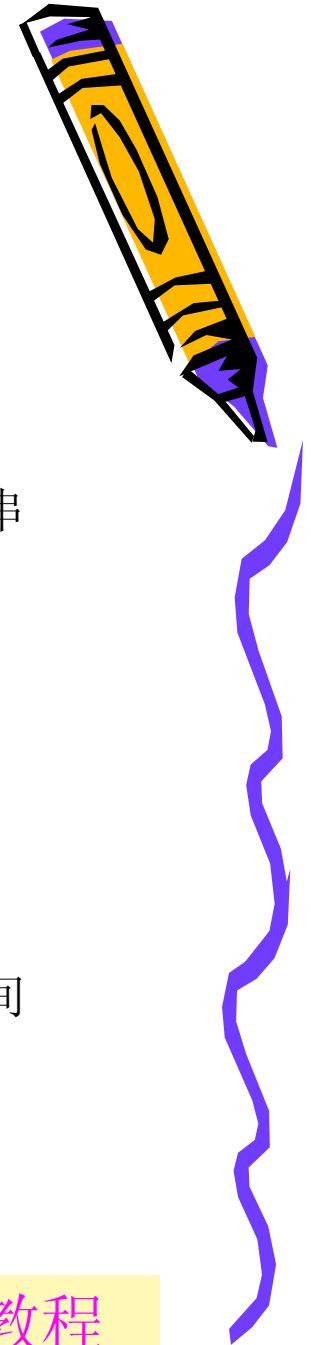


ACCESS实例教程

2. RecordSet对与Command对象联合使用

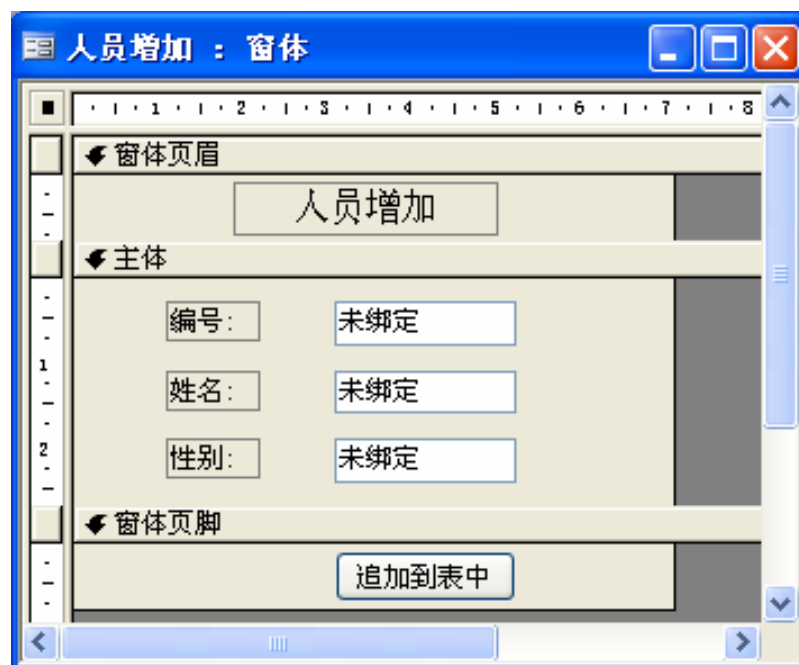
```
Dim cm as new ADOBD.Command
Dim rs as new ADOBD.RecordSet
cm.ActiveConnection=连接字符串
cm.CommandType=查询类型
cm.CommandText=查询字符串
rs. Open cm, 其他参数
do while not rs.EOF
    .....
rs.movenext
loop
rs.close
set rs=nothing
```

- ' 建立命令对象
- ' 建立记录集对象
- ' 建立命令对象的活动连接
- ' 指定命令对象的查询类型
- ' 建立命令对象的查询字符串
- ' 打开记录集
- ' 循环开始
- ' 对字段的各种操作
- ' 记录指针移到下一条
- ' 返回到循环开始处
- ' 关闭记录集
- ' 释放记录集对象变量所占内存空间



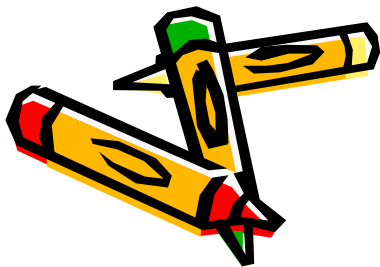
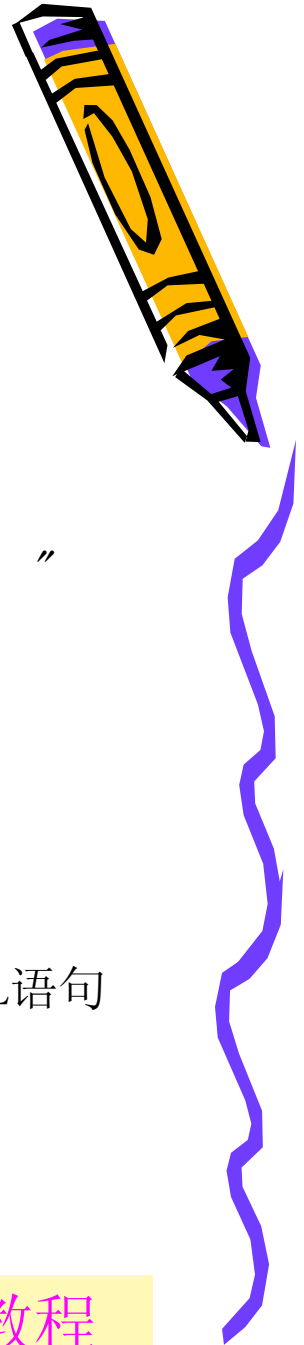
例：用ADO访问数据库

- (1) 在窗体建3个文本框→名称分别为t1、t2和t3。
- (2) 在窗体页眉添加标签→标题为“工员增加”。
- (3) 在窗体页脚建命令按钮→名称为c1→标题为“追加到表中”。



(4) 命令按钮c1的单击事件代码:

```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
Dim str1 As String
Set cn = CurrentProject.Connection          ' 建立本地连接
rs.ActiveConnection = cn
rs.Open "Select 教师编号 From 教师 Where 教师编号=' "+t1+" ' "
If rs.EOF = False Then
MsgBox "该编号已存在, 不能追加!"
Else
str1 = "Insert Into 教师 (教师编号, 姓名, 性别) "
str1 = str1 + "Values(' "+t1+" ', ' "+t2+" ', ' "+t3+" ')"
cn.Execute str1                             ' 执行指定的SQL语句
```



ACCESS实例教程

MsgBox “添加成功，请继续！”

End If

rs.Close

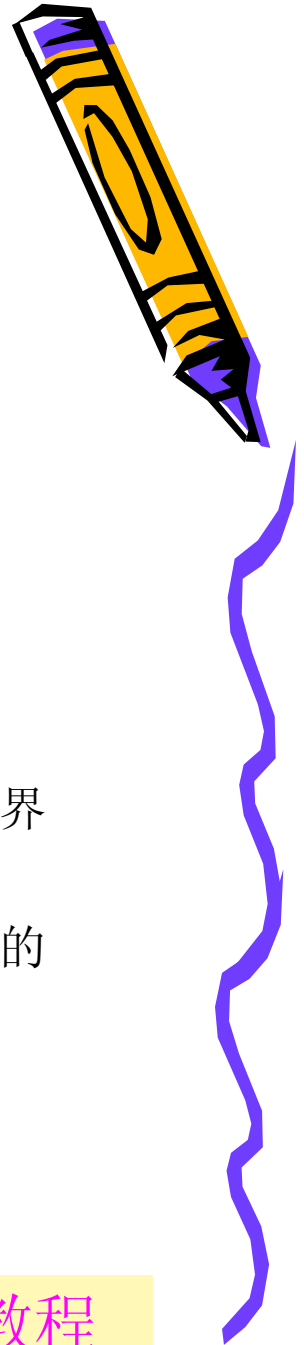
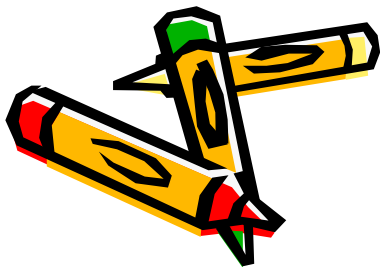
cn.Close

Set rs = Nothing

Set cn = Nothing

说明：

- 在程序中，用 “+t1+” 取得文本框t1中的值，原来用双引号定界的地方改为单引号。两个相同的引号不能连着写。
- Execute 是Command 对象和Connection对象的方法，执行指定的查询，并将执行产生的结果存储在 Recordset 对象中。

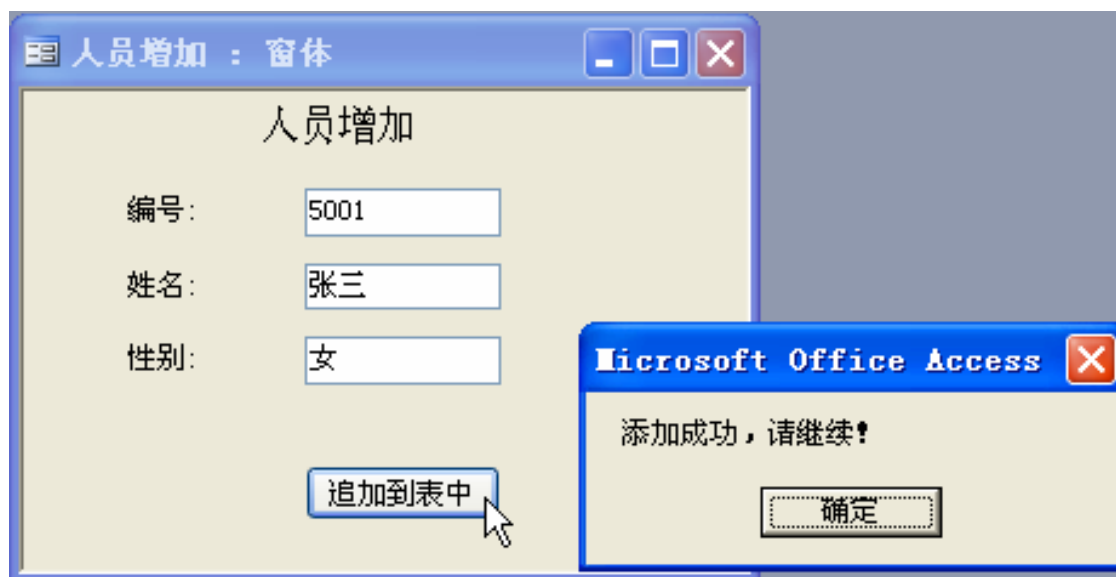


(5) 转到窗体视图，如果输入的编号与表中的编号有重复，单击命令按钮后显示消息框，输入的信息不能追加到表中。



ACCESS实例教程

(6) 如果输入的编号与表中的编号不重复，单击命令按钮后显示消息框，输入的信息被追加表中。



9.4 VBA程序错误处理

VBA提供了On Error语句，用来控制当有错误发生时程序的处理，启动一个错误处理程序并指定该子程序在一个过程中的位置。

一个错误处理程序不是 Sub 过程或 Function 过程，它是一段用行号标记的代码。

9.4.1 On Error语句的语法

On Error语句的语法通常有3种样式：



1. On Error GoTo 行号

启动错误处理程序，如果发生一个运行时错误，控件会转到行号所指位置，执行其中的代码。指定的行号必须在一个过程中，这个过程与 On Error 语句所在的过程相同。

例如：

On Error GoTo aa

’如果发生错误转到aa处

● ● ● ● ● ●

aa:

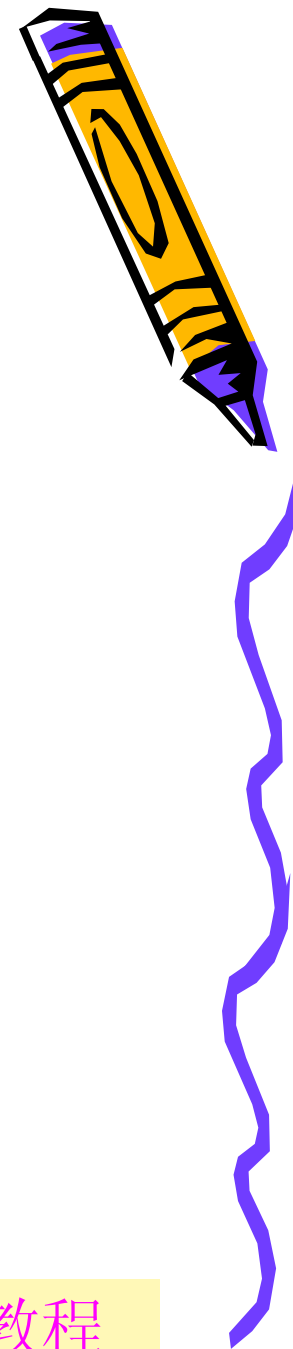
，行号aa

错误处理程序代码

’ 处理错误

● ● ● ● ● ●

功能：当程序发生错误，跳转到aa位置执行。



2. On Error Resume Next

当一个运行错误发生时，不考虑错误，继续执行下一条语句。访问对象时要使用这种形式而不使用 On Error GoTo 语句。

3. On Error GoTo 0

关闭错误处理，禁止当前过程中任何已启动的错误处理程序。



9.4.2 了解错误信息

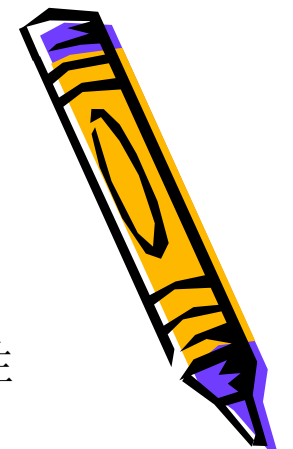
1. Err对象

Err是一个含有运行时错误信息的对象，Err对象的number属性返回错误代码。用Err.Source可以帮助确定是哪个对象产生错误，用 Err.Number 可以帮助确定是哪个对象将错误代码放在其中。Err 是全局范围的固有对象，在代码中不必建立这些对象的实例。

2. Rrror函数

格式：Rrror（错误号）

Rrror函数的自变量是一个有效的错误号，函数返回与已知错误号对应的错误信息。如果不是有效的错误号，会导致错误发生。错误号是可选项，如果省略，将返回与最近一次运行错误对应的消息。如果没有发生运行错误，或者错误号是 0，则 Error函数 返回一个长度为零的字符串（""）。



例：错误处理练习

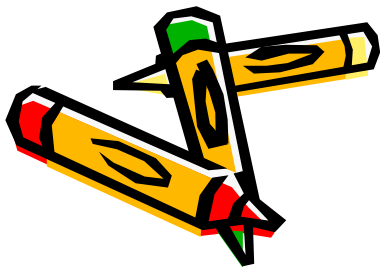
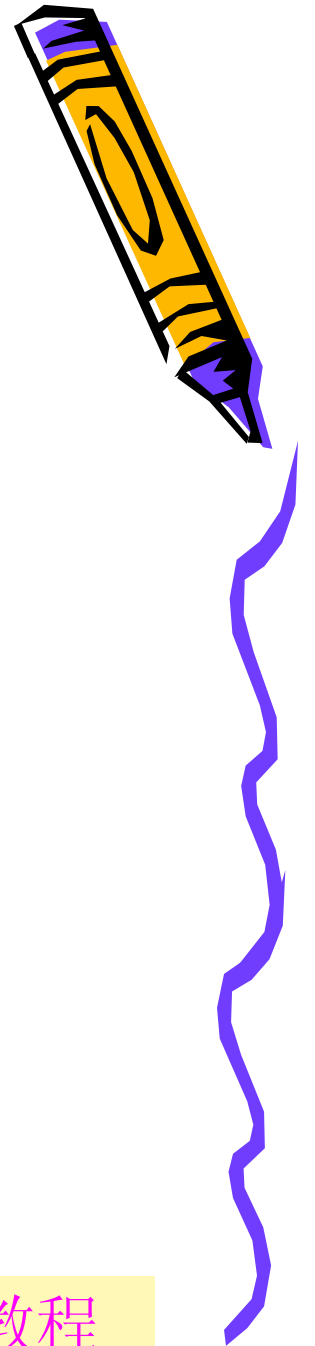
(1) 建文本框→名称为t1→建立命令按钮→名称为c1。

(2) 命令按钮c1的单击事件代码：

```
Private Sub c1_Click()  
On Error GoTo aa          ' 如发生错误转到行号aa  
Dim s1 As String  
s1 = t1  
DoCmd.OpenQuery s1, acViewNormal, acEdit    ' 打开查询  
bb:                                           ' 行号bb  
Exit Sub                                     ' 退出过程  
aa:                                           ' 行号aa  
MsgBox "没有此查询, 请重新输入!"  
Resume bb                                   ' 转到行号bb  
End Sub
```



(3) 转到窗体视图，在文本框输入“男讲师”，单击命令按钮，显示名为“男讲师”的查询。



(4) 在文本框输入“女讲师”，单击命令按钮，调用错误处理机制，用消息框提示错误。

