



计算机专项辅导——access 讲义

第 1 章 数据库基础知识.....	1
第 2 章 数据库和表.....	6
第 3 章 查 询.....	12
第 4 章 窗体.....	18
第 5 章 报表.....	22
第 6 章 数据访问页.....	24
第 7 章 宏.....	26
第 8 章 模块与 VBA 编程基础.....	28

第 1 章 数据库基础知识

1.1 数据库基础知识

1.1.1 计算机数据管理的发展

数据(Data)

数据是描述现实世界事物的符号记录,是用物理符号记录的可以鉴别的信息。包括文字、图形、声音等,他们都是用来描述事物特性的。

数据处理

数据处理是对各种类型的数据进行收集、存储、分类、计算、加工、检索与传输的过程。包括:收集原始数据、编码转换、数据输入、数据处理、数据输出等。

数据处理的中心问题是数据管理。

数据管理技术的发展,与计算机硬件(主要是外部存储器)、系统软件及计算机应用的范围有着密切的联系。数据管理技术的发展经历了以下几个阶段:人工管理阶段、文件系统阶段、数据库系统阶段、分布式数据库系统阶段和对象数据库系统阶段。

1. 人工管理阶段
2. 文件系统阶段
3. 数据库系统阶段
4. 分布式数据库系统阶段
5. 面向对象数据库系统阶段

1.1.2 数据库系统

1. 数据

描述事物的物理符号序列称为数据(Data)。

描述事物的物理符号可以是用来表示长度、体积、重量之类的数字数值,也可以是人名或地名、图形、图像、动画、影像、声音等非数值数据。

2. 数据库

数据库(DataBase, DB),顾名思义,是存放数据的仓库。是长期存放在计算机内,有组织的、大量的、可共享的数据集合。

3. 数据库应用系统

数据库应用系统是为特定应用开发的数据库应用软件系统。

数据库管理系统为数据的定义、存储、查询和修改提供支持,而数据库应用系统是对数据库中的数据进行处理和加工的软件,它面向特定应用。



4. 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是位于用户与操作系统 (OS) 之间的数据管理软件, 是数据库系统的核心软件。其主要功能包括以下几个方面 (P5 页)

- (1) 数据定义
- (2) 数据操纵
- (3) 数据库的运行管理
- (4) 数据的组织、存储和管理
- (5) 数据库的建立和维护
- (6) 数据通信接口

5. 数据库系统

数据库系统 (DataBase System, DBS) 是指拥有数据库技术支持的计算机系统。它可以实现有组织地、动态地存储大量相关数据, 提供数据处理和信息资源共享服务。数据库系统由硬件系统、数据库、数据库管理系统及相关软件、数据库管理员 (DataBase Administrator, DBA) 和用户组成。

1.1.3 数据模型

模型是现实世界特征的模拟和抽象。要将现实世界转变为机器能够识别的形式, 必须经过两次抽象, 即使用某种概念模型为客观事物建立概念级的模型, 将现实世界抽象为信息世界, 然后再把概念模型转变为计算机上某一 DBMS 支持的数据模型, 将信息世界转变为机器世界。

1. 实体的概念

(1) 实体 (Entity)

客观存在并可相互区别的事物称为实体。

(2) 属性 (Attribute)

实体所具有的某一特性称为属性。

(3) 实体集和实体型

属性值的集合表示一个实体, 而属性的集合表示一种实体的类型, 称为实体型。同类型的实体的集合, 称为实体集。

(4) 联系 (Relationship)

实体集之间的对应关系称为联系, 它反映现实世界事物之间的相互关联。

实体间的联系可分为以下三种类型

- ① 一对一联系 例如, 一个学生只能有一个学号, 而一个学号只能指向一个学生, 则学生与学号之间具有一对一联系。
- ② 一对多联系 例如, 一个班级中有若干名学生, 而每个学生只在一个班级中学习, 则班级与学生之间具有一对多联系。
- ③ 多对多联系 例如, 一门课程同时有若干个学生选修, 而一个学生可以同时选修多门课程, 则课程与学生之间具有多对多联系。

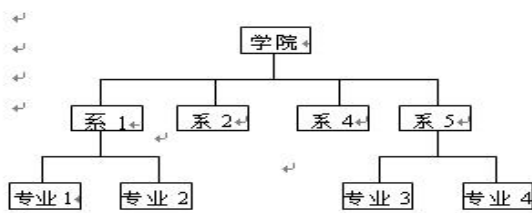
3. 数据模型

常用的数据模型有 3 种: 层次模型、网状模型和关系模型。

层次模型

层次模型是数据库系统中最早出现的数据模型它用树形结构表示各实体以及实体间的联系。

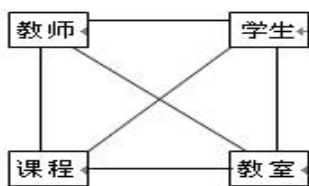
若用图来表示, 层次模型是一棵倒立的树。在数据库中, 满足以下条件的数据模型称为层次模型: ① 有且仅有一个结点无父结点, 这个结点称为根结点; ② 其他结点有且仅有一个父结点。



I 层次模型示例

网状模型

在现实世界中，事物之间的联系更多的是非层次关系的，用层次模型表示非树型结构是很不直接的，网状模型则可以克服这一弊病。网状模型是一个网络。在数据库中，满足以下两个条件的数据模型称为网状模型。① 允许一个以上的结点无父结点；② 一个结点可以有多于一个的父结点。



网状模型示例

1.2 关系数据库

1.2.1 关系模型

以二维表的形式表示实体与实体间联系的数据模型称为关系模型。

从模型的三要素角度看，关系模型的内容为：

数据结构：一张二维表格。

数据操作：数据表的定义，检索、维护、计算等。

数据约束条件：表中列的取值范围即域值的限制条件。

(1) 关系

在关系模型中，一个关系就是一张二维表，每个关系都有一个关系名。在数据库中，一个关系存储为一个数据表。

(2) 元组

表中的行称为元组，每一行是一个元组，对应数据表中的一个具体记录，元组的各分量分别对应于关系的各个属性。

(3) 属性

表（关系）中的列称为属性，每一列有一个属性名，对应数据表中的一个字段。

(4) 域

一个属性的取值范围是该属性的域。

(5) 关键字

表中的一个属性（组），它的值可以唯一地标识一个元组。如：学号。

(6) 外关键字

如果一个关系中的属性或属性组并非该关系的关键字，但它们是另外一个关系的关键字，则称其为该关系的外关键字。

关系的特点：

(1) 关系必须规范化。规范化是指关系模型中的每一个关系模式都必须满足一定的要求。最基本的要求是每个属性必须是不可分割的数据单元，即表中不能在包含表。

(2) 在同一个关系中不能出现相同的属性名。

(3) 关系中不允许有完全相同的元组，即冗余。

(4) 在一个关系中元组的次序无关紧要。



(5) 在一个关系中列的次序无关紧要。

1.2.2 关系运算

在关系数据库中，关系的基本运算有两类：传统的集合运算和专门的关系运算。

一 传统的集合运算

1. 并

设有两个关系 R 和 S，它们具有相同的结构。R 和 S 的并是由属于 R 或属于 S 的元组组成的集合，运算符为 \cup 。

R			S			R \cup S		
A	B	C	A	B	C	A	B	C
a1	b1	c1	a1	b2	c2	a1	b1	c1
a1	b2	c2	a1	b3	c2	a1	b2	c2
a2	b2	c1	a2	b2	c1	a2	b2	c1
						a1	b3	c2

(a) (b) (c)

2. 差

设有两个关系 R 和 S，它们具有相同的结构。R 和 S 的差是由属于 R 但不属于 S 的元组组成的集合。运算符为 $-$ 。

R			S			R - S		
A	B	C	A	B	C	A	B	C
a1	b1	c1	a1	b2	c2	a1	b1	c1
a1	b2	c2	a1	b3	c2			
a2	b2	c1	a2	b2	c1			

(a) (b) (d)

3. 交

设有两个关系 R 和 S，它们具有相同的结构。R 和 S 的交是由既属于 R 又属于 S 的元组组成的集合，运算符为 \cap 。

R			S			R \cap S		
A	B	C	A	B	C	A	B	C
a1	b1	c1	a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2			
a2	b2	c1	a2	b2	c1	a2	b2	c1

(a) (b) (e)

二 传统的集合运算

1. 选择

选择运算是在关系中选择满足某些条件的元组。也就是说，选择运算是在二维表中选择满足指定条件的行。



【例1-1】关系 R 如图所示，求 $\sigma_{A='a'}(R)$ 。

A	B	C
a	b	c
d	e	f
g	b	e

A	B	C
a	b	c

2. 投影

投影运算是从关系模式中指定若干个属性组成新的关系。即在关系中选择某些属性列。

【例1-2】关系 R 如图所示，求 $\Pi_{A,C}(R)$ 。

A	B	C
a	b	c
d	e	f
g	b	e

A	C
a	c
d	f
g	e

3. 联接

联接运算将两个关系模式通过公共的属性名拼接成一个更宽的关系模式，生成的新关系中包含满足联接条件的元组。

【例1-3】关系 R 和关系 S 分别如图所示，求 $R \bowtie S$ 。

A	B	C
a	b	c
d	a	f
c	b	d

B	C	D
b	c	d
a	f	a

A	B	C	D
a	b	c	d
d	a	f	a

1.3 数据库设计基础

1.3.2 数据库设计的步骤

1. 需求分析
2. 确定所需要的表
3. 确定所需字段
4. 确定联系
5. 设计求精



第2章 数据库和表

2.1 数据库的创建

创建数据库方法有两种,一是先建立一个空数据库,然后向其中添加表、查询、窗体和报表等对象,这是创建数据库最灵活的方法;二是使用“数据库向导”,利用系统提供的模板进行一次操作来选择数据库类型,并创建所需的表、窗体和报表,这是操作最简单的方法。

2.1.1 创建空数据库

例 2-1 建立“教学管理”空数据库,并将建好的数据库保存在 C 盘 Access 文件夹中。

注意:数据库创建好后,数据库容器中还不存在任何其他数据库对象,此时可以根据需要在该数据库容器中创建数据库对象。另外,在创建数据库之前,最好先建立用于保存该数据库文件的文件夹,以便今后的管理。

2.1.2 使用向导创建数据库

例 2-2 在 C 盘“教学管理”文件夹下创建“教学管理”数据库。

利用“数据库向导”创建数据库对象,在所建的数据库对象容器中包含了表、查询、窗体、报表、宏和模块等 Access 对象。但是,由于“数据库向导”创建的表可能与需要的表不完全相同,表中包含的字段可能与需要的字段不完全一样。因此通常使用“数据库向导”创建数据库后,还需要对其进行补充和修改。

2.1.3 打开数据库

1. 通过“开始工作”任务窗格打开
2. 使用“打开”命令打开

例 2-3 打开 C 盘“教学管理”文件夹中“教学管理”数据库。

2.1.4 关闭数据库

关闭数据库的方法有 3 种。

- (1) 单击“数据库”窗口右上角“关闭”按钮
- (2) 双击“数据库”窗口左上角“控制”菜单图标
- (3) 单击“数据库”窗口左上角“控制”菜单图标,从弹出快捷菜单中选择“关闭”命令

2.2 建立表

2.2.1 表的组成

表是 Access 数据库中最基本的对象,是具有结构的某个相同主题的数据集合。

表由两部分构成:表结构和表内容。

1. 字段名称

每个字段应具有唯一的名字,称为字段名称。

字段名称的命名规则为:

- (1) 长度为 1~64 个字符。
- (2) 可以包含字母、汉字、数字、空格和其他字符,但不能以空格开头。
- (3) 不能包含句号 (.)、惊叹号 (!)、方括号 ([]) 和重音符号 (')。
- (4) 不能使用 ASCII 为 0~32 的 ASCII 字符。

2. 数据类型



数据类型表

数据类型	用途	字符长度
文本	字母、汉字和数字。如：学号、姓名、性别、民族	0~255 个字符
备注	字母、汉字和数字（和文本型数据相似，但容量更大）	0~64000 个
数字	数值。一般参与算术计算。如：入学成绩	1、2、4、或 8 字节
日期/时间	日期/时间，如，出生日期	8 字节
货币	数值	8 字节
自动编号	每次添加新记录时 Access2003 自动添加的连续数字	4 字节
是/否	是/否、真/假或开/关	1 位（1/8 字
OLE 对象	可与 Visual Basic 交互作用的 OLE 对象（链接或嵌入对象）。如：照片	可达 1GB
超链接	Web 地址、Internet 地址或链接到其他数据库或应用程序	可达 65536 字符
查阅向导	来自其他表或者列表的值	通常为 4 字节

注意：“数字”、“日期 / 时间”、“货币”以及“是 / 否”，这些数据提供预先定义好的显示格式。可以从每一个数据类型可用的格式中选择所需的格式来设置“格式”属性。也可以为所有的数据类型创建自定义显示格式，但“OLE 对象”数据类型除外。

2.2.2 建立表结构

1. 使用“数据表”视图

“数据表”视图是按行和列显示表中数据的视图。在数据表视图中，可以进行字段的编辑、添加、删除和数据的查找等各种操作。

2. 使用设计视图

表设计视图是创建表结构以及修改表结构最方便、最有效的窗口。

3. 使用表向导

表设计视图是创建表结构以及修改表结构最方便、最有效的窗口。

4. 定义主键

定义主键的方法有两种，一是在建立表结构过程中定义主键；二是在建立表结构后，重新打开设计视图定义主键。

主键也称为关键字，是表中能够唯一标识记录的一个字段或多个字段的组合。

主键有三种，即自动编号、单字段和多字段。

自动编号主键的特点是，当向表中增加一个新记录时，主键字段值会自动加 1，如果在保存新建表之前未设置主键，则 Access 会询问是否要创建主键，如果回答“是”，Access 将创建自动编号类型的主键。

单字段主键是以某一个字段作为主键，来唯一标识记录，这类主键的值可由用户自行定义。

多字段主键是由两个或更多字段组合在一起来唯一标识表中记录。

2.2.3 设置字段属性

字段属性表示字段所具有的特性，它定义了字段数据的保存、处理或显示。

1. 字段大小

- 字段大小属性用于限制输入到该字段的最大长度，当输入的数据超过该字段设置的字段大小时，系统将拒绝接收。
- 字段大小属性只适用于文本、数字或自动编号类型的字段。
- 在设计视图的字段属性框的字段大小属性中选择字段大小的类型（数字）或输入字段大小的字符



个数（文本）。

- ❑ 如果文本字段中已经有数据，那么减小字段大小会造成数据丢失，Access 将截去超出新限制的字符。
- ❑ 如果在数字字段中包含小数，那么将字段大小属性设置为整数时，Access 自动将小数取整。
- ❑ 如果文本型字段的值是汉字，那么每个汉字占一位。

2. 格式

- ❑ **格式属性影响数据的显示格式。**
- ❑ 可确定数字、货币、日期等字段格式。
- ❑ 在设计视图的字段属性框的字段格式属性中选择字段格式的类型。

3. 输入掩码

- ❑ **用于指定数据的输入格式。**
- ❑ 可定义文本、数字、日期/时间、货币等数据类型的字段。
- ❑ 如果为某字段定义了输入掩码，同时又设置了它的格式属性，格式属性将在数据显示时优先于输入掩码的设置。
- ❑ **只为文本型和日期/时间型字段提供向导。**
- ❑ 可使用字符定义输入掩码属性。
- ❑ 输入掩码属性由字面字符（如空格、点、括号等）和决定输入数值的类型的特殊字符组成。

字符	说明
0	数字（0 到 9，必选项；不允许使用加号[+]和减号[-]）
9	数字或空格（非必选项；不允许使用加号和减号）
#	数字或空格（非必选项；空白将转换为空格，允许使用加号和减号）
L	字母（A 到 Z，必选项）
?	字母（A 到 Z，可选项）
A	字母或数字（必选项）
a	字母或数字（可选项）
&	任一字符或空格（必选项）
C	任一字符或空格（可选项）
. , : ; - /	十进制占位符和千位、日期和时间分隔符（实际使用的字符取决于 Windows “控制面板”的“区域设置”中指定的区域设置）
<	使其后所有的字符转换为小写
>	使其后所有的字符转换为大写
!	输入掩码从右到左显示，输入至掩码的字符一般都是从左到右的。可以在输入掩码的任意位置包含叹号
\	使其后的字符显示为原义字符。可用于将该表中的任何字符显示为原义字符（例如，\A 显示为 A）
密码	将“输入掩码”属性设置为“密码”，以创建密码输入项文本框。文本框中键入的任何字符都按原字符保存，但显示为星号（*）

4. 默认值

- ❑ 设置默认值可以减少数据输入量。
- ❑ 在设计视图的字段属性框的默认值属性中输入默认的值。
- ❑ 设置默认值属性时，必须与字段中所设的数据类型相匹配，否则会出现错误。

5. 有效性规则



- ❑ 为防止数据的输入错误而设置。
- ❑ 有效性规则允许定义一条规则，限制可以接受的内容。
- ❑ 有效性规则的实质是一个限制条件，完成对数据输入的检测。
- ❑ 在设计视图的字段属性框的有效性规则属性中该字段的有效性规则。

6. 有效性文本

- ❑ 当输入的数据违反了有效性规则时，显示的文本信息。
- ❑ 通常与有效性文本同时定义。

7. 索引

- ❑ 索引能根据键值加速在表中查找和排序的速度，并且能对表中的记录实施唯一性。
- ❑ 在 Access 中，可以创建基于单个字段的索引，也可以创建基于多个字段的索引。

2.2.4 建立表之间的关系

1. 定义关系的条件

- ❑ 表间关系分为一对一、一对多和多对多三种。
- ❑ 建立表间关系的条件：

- (1) 如果仅有一个相关字段是主键或具有唯一索引，则创建一对多关系。
- (2) 如果两个相关字段都是主键或唯一索引，则创建一对一关系。
- (3) 多对多关系实际上是某两个表与第三个表的两个一对多关系，第三个表的主键包含两个字段，分别是前两个表的外键。

2. 参照完整性

- ❑ 指数据的正确性、准确性和有效性。参照完整性是在输入或删除记录时，为维持表之间已定义的关系而必须遵循的规则。
- ❑ 如果实施了参照完整性，那么当主表中没有相关记录时，就不能将记录添加到相关表中，也不能在相关表中存在匹配的记录时删除主表中的记录，更不能在相关表中有相关记录时，更改主表中的主键值。

3. 建立表之间的关系

建立表间关系的方法包括：使用向导创建数据库同时创建表间关系；使用“关系”命令定义表间关系。

4. 子数据表

- ❑ 子数据表是指在一个数据表视图中显示已与其建立关系的数据表视图。
- ❑ 在建有关系的主数据表视图上，每条记录左端都有一个关联标记“□”。在未显示子数据表时，关联标记内为一个“+”号，此时单击某记录的关联标记，即可显示该记录对应的子数据表记录数据，而该记录左端的关联标记内变为一个“-”号。若需展开所有记录的子数据表，可选择“格式”→“子数据表”→“全部展开”命令；若需折叠展开的子数据表，可选择“格式”→“子数据表”→“全部折叠”命令。

3.2.5 向表中输入数据

1. 以数据表视图方式输入数据

2. 创建查阅列表字段

3. 获取外部数据

(1) 导入数据

导入数据是将其它系统的数据库数据导入到 Access 数据库中。

一般使用导入命令将外部数据导入至当前数据库中。

(2) 链接数据

链接数据是指在自己的数据库中形成一个链接表对象，每次在 Access 数据库中操作数据时，都是即时从外部数据源获取数据。

2.3 维护表



为使数据库中表结构更合理,内容使用更有效,需要对表进行维护。

2.3.1 打开/关闭表

1. 打开表

- 使用打开命令(改内容)
- 使用设计命令(改结构)

2. 关闭表

2.3.2 修改表结构

1. 添加字段

2. 修改字段

3. 删除字段

4. 重新设置主键

2.3.3 编辑表内容

1 定位记录

数据表中有了数据后,修改是经常要做的操作,其中定位和选择记录是首要的任务。常用的记录定位方法有两种:一是用记录号定位,二是用快捷键定位。快捷键及其定位功能见下表。

2 选择记录

选择记录是指选择用户所需要的记录。用户可以在“数据表”视图下使用鼠标或键盘两种方法选择数据范围。

3. 编辑表内容

- (1) 添加记录 在已经建立的表中,添加新的记录。
- (2) 删除记录 删除表中出现的不需要的记录。
- (3) 修改数据 在已建立的表中,修改出现错误的的数据。
- (6) 复制数据 在输入或编辑数据时,有些数据可能相同或相似,这时可以使用复制和粘贴操作将某些字段中的部分或全部数据复制到另一个字段中。

2.3.4 调整表外观

调整表的结构和外观是为了使表看上去更清楚、美观。调整表外观的操作包括:改变字段次序、调整字段显示宽度和高度、隐藏列和显示列、冻结列、设置数据表格式、改变字体显示等。

2.4 操作表

2.4.1 查找数据

查找指定内容

在“数据表”视图中,使用编辑菜单中的查找命令

查找空值或空字符串

空值是字段中还没有值,可用 Null 值代替

空字符串是用双引号括起来的空字符串,即"",中间没有空格,长度为 0



字符	用法	示例
*	与任何个数的字符匹配，它可以在字符串中， 当做第一个或最后一个字符使用	wh* 可以找到 what、white 和 why
?	与任何单字母的字符匹配	B?ll 可以找到 ball、bell 和 bill
[]	与方括号内任何单个字符匹配	B[ae]ll 可以找到 ball 和 bell， 但找不到 bill
!	匹配任何不在括号之内的字符	B[!ae]ll 可以找到 bill 和 bull， 但找不到 bell
-	与范围内的任何一个字符匹配。必须以 递增排 序次序来指定区域（A 到 Z，而不是 Z 到 A）	B[a-c]d 可以找到 bad、bbd 和 bcd
#	与任何单个数字字符匹配	1#3 可以找到 103、113、123

2.4.2 替换数据

在“数据表”视图中，使用编辑菜单中的替换命令。

- 1、替换指定内容
- 2、更改默认设置

2.4.3 排序记录

排序记录时，不同的字段类型，排序规则有所不同，具体规则如下：

- (1) 英文按字母顺序排序，大小写视为相同，升序时按 A 到 z 排列，降序时按 z 到 A 排列。
- (2) 中文按拼音的顺序排序，升序时按 A 到 z 排列，降序时按 z 到 A 排列。
- (3) 数字按数字的大小排序，升序时从小到大排列，降序按从大到小排列。
- (4) 使用升序排序日期和时间，是指由较前的时间到较后的时间；使用降序排序时，则是指由较后的时间到较前的时间。

排序时，要注意的事项如下：

- (1) 在“文本”字段中保存的数字将作为字符串而不是数值来排序。因此，如果要以数值的顺序来排序，必须在较短的数字前面加上零，使得全部文本字符串具有相同的长度。例如：要以升一序来排序以下的文本字符串“1”、“2”、“11”和“22”，其结果将是“1”、“11”、“2”、“22”。必须在仅有一位数的字符串前面加上零，才能正确地排序：“01”、“02”、“11”、“22”。对于不包含 Null 值的字段，另一个解决方案是使用 Val 函数来排序字符串的数值。例如：如果“年龄”列是包含数值的“文本”字段，在“字段”单元格指定 Val ([年龄])，并且在“排序”单元格指定排序次序后，才会以正确的顺序来放置记录。如果只在“文本”字段之中保存数字或日期，可以考虑将表的数据类型更改为数字、货币或日期 / 时间。这样在对此字段排序时，数字或日期将会以数值或日期的顺序来排序，而不需要加入前面的零。

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

2.4.4 筛选记录

1. 按选定内容筛选
 - ☐ 筛选符合单值的记录
2. 按窗体筛选
 - ☐ 筛选符合两个以上值的记录
3. 按筛选目标筛选



- 筛选符合某条件的记录

4. 高级筛选

- 筛选满足复杂条件的记录

第3章 查 询

3.1 查询概述

查询是 Access 数据库中的一个重要对象，是使用者按照一定条件从 Access 数据库表或已建立的查询中检索需要数据的最主要方法。

3.1.1 查询的功能

1. 选择字段 在查询中，选择表中的部分字段。
2. 选择记录 根据指定的条件查找所需的记录，并显示找到的记录。
3. 选择记录 利用查询添加、修改和删除表中记录。
4. 实现计算 在建立查询时进行各种统计计算。
5. 建立新表 利用查询得到的结果建立新表。
6. 为窗体、报表或数据访问页提供数据

查询对象不是数据的集合，而是操作的集合。查询的运行结果是一个数据集合，也称为动态集。它很像一张表，但并没有被存储在数据库中。创建查询后，保存的只是查询的操作，只有在运行查询时，Access 才会从查询数据源表的数据中抽取出来并创建它；只要关闭查询，查询的动态集就会自动消失。

3.1.2 查询的类型

在 Access 中，查询分为 5 种，分别是选择查询、交叉表查询、参数查询、操作查询和 SQL 查询。5 种查询的应用目标不同，对数据源的操作方式和操作结果也不同。

1. 选择查询

- 选择查询是最常用的查询类型。顾名思义，它是根据指定条件，从一个或多个数据源中获取数据并显示结果。也对记录进行分组，并且对分组的记录进行总计、计数、平均以及其他类型的计算。
- 例如，查找 1992 年参加工作的男教师，统计各类职称的教师人数等。

2. 交叉表查询

- 交叉表查询将来源于某个表或查询中的字段进行分组，一组列在数据表左侧，一组列在数据表上部，然后在数据表行与列的交叉处显示数据源中某个字段统计值。
- 例如，统计每个系男女教师的人数，要求行标题显示系名，列标题显示性别，表的交叉处显示统计的人数。

3. 参数查询

- 参数查询是一种根据使用者输入的条件或参数来检索记录的查询。
- 例如，可以设计一个参数查询，提示输入两个成绩值，然后 Access 检索在这两个值之间的所有记录。

4. 操作查询

- 操作查询与选择查询相似，都需要指定查找记录的条件，但选择查询是检查符合特定条件的一组记录，而操作查询是在一次查询操作中对所得结果进行编辑等操作。
- 操作查询有 4 种：生成表、删除、更新和追加。

5. SQL 查询

- SQL 查询是使用 SQL 语句来创建的一种查询。
- SQL 查询有 4 种：包括联合查询、传递查询、数据定义查询和子查询等。

联合查询是将一个或多个表、一个或多个查询的字段组合为查询结果中的一个字段，执行联合查询时，将返回所包含的表或查询中的对应字段记录；

传递查询是直接将命令发送到 ODBC 数据库，它使用服务器能接受的命令，利用它可以检索或更改记录；



数据定义查询可以创建、删除或更改表，或在当前的数据库中创建索引；

子查询是包含另一个选择或操作查询中的 SQL SELECT 语句，可以在查询设计网格的“字段”行输入这些语句来定义新字段，或在“准则”行来定义字段的准则。

3.1.3 查询的条件

1. 运算符

运算符是构成查询条件的基本元素。Access 提供了关系运算符，逻辑运算符和特殊运算符 3 种。

关系运算符	说明
=	等于
<>	不等于
<	小于
<=	小于等于
>	大于
>=	大于等于

逻辑运算符	说明
Not	当 Not 连接的表达式为真时，整个表达式为假
And	当 And 连接的表达式都是真时，整个表达式为真，否则为假
Or	当 Or 连接的表达式有一个为真时，整个表达式为真，否则为假

特殊运算符	说明
In	用于指定一个字段值的列表，列表中的任意一个值都可与查询的字段相匹配
Between	用于指定一个字段值的范围，指定的范围之间用 And 连接
Like	用于指定查找文本字段的字符模式。在所定义的字符模式中，用“?”表示该位置可匹配任何一个字符；用“*”表示该位置可匹配零或多个字符；用“#”表示该位置可匹配一个数字；用方括号描述一个范围，用于表示可匹配的字符范围
IsNull	用于指定一个字段为空
IsNotNull	用于指定一个字段为非空

2. 函数（简单介绍几个）

Access 提供了大量的标准函数，如数值函数、字符函数、日期时间函数和统计函数等。（详见附录）

函数	说明
Abs	返回数值表达式值的绝对值
Int	返回数值表达式值的整数部分
Sqr	返回数值表达式值的平方根
Sgn	返回数值表达式的符号值

3. 使用数值作为查询条件
4. 使用文本值作为查询条件
5. 使用计算或处理日期结果作为查询条件
6. 使用字段的部分值作为查询条件
7. 使用空值或空字符串作为查询条件



注意:

- 在条件中字段名必须用方括号括起来。
- 数据类型必须与对应字段定义的类型相符合。

3.2 创建选择查询

根据指定条件, 从一个或多个数据源中获取数据的查询称为选择查询。创建创建选择查询的方法有两种: 查询向导和设计视图。

3.2.1 使用查询向导

使用查询向导创建查询, 操作者可以在向导指示下选择一个或多个表、一个或多个字段, 但不能设置查询条件。

例 3-1 查找“教师”表中的记录, 并显示“姓名”、“性别”、“工作时间”和“系别”4 个字段。

注意:

- 1、在数据表视图显示查询结果时, 字段的排列顺序与在“简单查询向导”对话框中选定字段的顺序相同。故在选定字段时, 应考虑按照字段的显示顺序选取。
- 2、当所建查询的数据源来自于多个表时, 应建立表之间的关系。

3.2.2 使用“设计”视图

1. 查询“设计”视图

- 在 Access 中, 查询有 5 种视图: 设计视图、数据表视图、SQL 视图、数据透视表视图和数据透视图视图。
- 在设计视图中, 既可以创建不带条件的查询, 也可以创建带条件的查询, 还可以对已建查询进行修改。

3.2.3 在查询中进行计算

1 查询中的计算功能

在 Access 查询中, 可以执行两种类型的计算, 预定义计算和自定义计算。

(1) 预定义计算即“总计”计算, 是系统提供的用于对查询中的记录组或全部记录进行的计算, 它包括总计、平均值、计数、最大值、最小值、标准偏差或方差等。

(2) 自定义计算使用一个或多个字段的值进行数值、日期和文本计算。对于自定义计算, 必须直接在“设计网格”中创建新的计算字段, 创建方法是将表达式输入到“设计网格”中的空字段单元格, 表达式可以由多个计算组成。

2 总计查询

总计查询是在成组的记录中完成一定计算的查询。使用查询设计视图中的“总计”行, 可以对查询中全部记录或记录组计算一个或多个字段的统计值。

例 4-7 统计 1992 年参加工作的教师人数。

3 分组总计查询

将记录进行分组, 对每个组的值进行统计。分组统计时, 应在该字段的“总计”行上选择“分组”。

例 4-8 计算各类职称的教师人数。

4 添加计算字段

添加的新字段值是根据一个或多个表中的一个或多个字段并使用表达式计算得到, 也称为计算字段。

例 4-10 查找平均分低于所在班平均分的学生并显示其班级号、姓名和平均成绩。假设, 班级号为“学生编号”中的前 8 位。

3.3 交叉表查询

1. 认识交叉表查询

(1) 所谓交叉表查询, 就是将来源于某个表中的字段进行分组, 一组列在数据表的左侧, 一组列在数据表的上部, 然后在数据表行与列的交叉处显示表中某个字段的各种计算值

(2) 与 Excel 中的数据透视表一样



2. 使用“交叉表查询向导”

例 3-11 创建一个交叉表查询, 统计每班男女生人数。

注意:

使用向导创建交叉表的数据源必须来自于一个表或一个查询。如果数据源来自多个表, 可以先建立一个查询, 然后以此查询作为数据源。

3. 使用“设计”视图

例 3-12 创建交叉表查询, 使其统计各班男生平均成绩和女生平均成绩。

当所建“交叉表查询”数据来源于多个表或查询时, 使用设计视图。当所用数据源来自于一个表或查询, 使用“交叉表查询向导”。如果“行标题”或“列标题”需要通过建立新字段得到, 使用设计视图。

3.4 参数查询

参数查询利用对话框, 提示输入参数, 并检索符合所输参数的记录。可以创建一个参数提示的单参数查询, 也可以创建多个参数提示的多参数查询。

1. 单参数查询

创建单参数查询, 即指定一个参数。在执行单参数查询时, 输入一个参数值。

例 3-13 按学生姓名查找某学生的成绩, 并显示“学生编号”、“姓名”、“课程名称”及“成绩”等。

2. 多参数查询

创建多参数查询, 即指定多个参数。在执行多参数查询时, 需要依次输入多个参数值。

例 3-14 建立一个查询, 使其显示某班某门课的学生“姓名”和“成绩”。

3.5 操作查询

操作查询是指仅在一个操作中更改许多记录的查询。例如在一个操作中删除一组记录, 更新一组记录等。

操作查询包括生成表查询、删除查询、更新查询、追加查询等。

1. 生成表查询

生成表查询是利用一个或多个表中的全部或部分数据创建新表。

2. 删除查询

从一个或多个表中删除指定的数据。

3. 更新查询

对一个或多个表中的记录进行更新。

4. 追加查询

从一个或多个表中将一组记录追加到另一个或多个表中。

注意: 无论哪一种操作查询, 都可以在一个操作中更改许多记录, 并且在执行操作查询后, 不能撤消刚刚做过的更改操作。因此在执行操作查询之前, 最好单击工具栏上的“视图”按钮, 预览即将更改的记录。另外, 在使用操作查询之前, 应该备份数据。

3.6 创建 SQL 查询

3.6.1 查询与 SQL 视图

- 在 Access 中, 一个查询都对应着一个 SQL 语句, 查询对象的实质是一条 SQL 语句。
- 当使用设计视图建立一个查询时, Access 在后台就会构造一个等价的 SQL 语句。

3.6.2 SQL 语言简介

SQL(Structured Query Language, 结构化查询语言)是数据库领域中应用最为广泛的数据库查询语言。

SQL 的特点:

- SQL 是一种一体化语言, 包括数据定义、数据查询、数据操纵和数据控制等方面的功能, 可以完成数据库活动中的全部工作。
- SQL 是一种高度非过程化语言, 只需描述“做什么”, 不需说明“怎么做”。
- SQL 是一种非常简单的语言, 所用语句很接近于 SQL 是一个共享语言, 全面支持客户机



/服务器结构。

2. SQL 语句

SQL 语句的功能包括：查询、操纵、定义和控制。

SQL 功能	动 词	SQL 功能	动 词
数据定义	CREATE, DROP, ALTER	数据查询	SELECT
数据操作	INSERT, UPDATE, DELETE	数据控制	GRANT, REVOKE

1.CREATE 语句

格式：CREATE TABLE <表名>

(<字段名 1> <数据类型> [列完整性约束条件],

[<字段名 2> <数据类型> [列完整性约束条件]])

例 3.19 建立一个“雇员”表，包括雇员号、姓名、性别、出生日期、部门，备注字段。

CREATE TABLE 雇员

(雇员号 CHAR(8) SMALLINT Primary Key,

姓名 CHAR(4) Not Null, 性别 CHAR(1),

出生日期 DATE, 部门 CHAR(20)), 备注 MEMO;

2. ALTER 语句

格式：ALTER TABLE <表名>

[ADD <新字段名> <数据类型> [字段级完整性约束条件]]

[DROP [<字段名>]]

[ALTER <字段名> <数据类型>]

其中，<表名>是指需要修改的表的名字，ADD 子句用于增加新字段和该字段的完整性约束条件，DROP 子句用于删除指定的字段，ALTER 子句用于修改原有字段属性。

例 3.20 在“雇员”表中增加一个字段，字段名为“职务”，数据类型为“文本”；将“备注”字段删除；将“雇员号”字段的数据类型改为文本型，字段大小为 8。

(1) 添加新字段的 SQL 语句为：

ALTER TABLE 雇员 ADD 职务 CHAR(10);

(2) 删除“备注”字段的 SQL 语句为：

ALTER TABLE 雇员 DROP 备注;

(3) 修改“雇员号”字段属性的 SQL 语句为：

ALTER TABLE 雇员 ALTER 雇员号 CHAR(8);

3.DROP 语句

格式：DROP TABLE <表名>

例 3.21 删除已建立的“雇员”表

DROP TABLE 雇员;

4.INSERT 语句

格式：

INSERT

INTO <表名>[(<属性名 1>[,<属性名 2>...])]

VALUES (<常量 1>[,<常量 2>]...);

例 3.22 将一条新记录插入到“雇员”表中

INSERT INTO 雇员 VALUES (“0001”, “张磊”, “男”, #1960-1-1#, “办公室”);

例 3.23 将一条新记录插入到“雇员”表中，其中“雇员号”为“0002”，“姓名”为“王宏”，“性别”为“男”



INSERT INTO 雇员 (雇员号, 姓名, 性别) VALUES ("0002", "王宏", "男");

注意：文本数据应用双引号括起来。日期数据应用“#”号括起来。

5. UPDATE 语句

格式：

UPDATE <表名>

SET <列名>=<表达式> [<列名>=<表达式>]...

[WHERE <条件>];

例 3.24 将“雇员”表张磊的出生日期改为“1960-1-11”

UPDATE 雇员 SET 出生日期=#1960-1-11#

WHERE 姓名="张磊";

6. DELETE 语句

格式：

DELETE

FROM <表名>

[WHERE <条件>];

例 3.25 将“雇员”表中“雇员号”为“0002”的记录删除

DELETE FROM 雇员 WHERE 雇员号="0002";

7. SELECT 语句

格式：

SELECT [ALL|DISTINCT] *|<字段列表> FROM <表名 1>[,<表名 2>]...

[WHERE <条件表达式>]

[GROUP BY <字段名>[HAVING<条件表达式>]]

[ORDER BY <字段名>[ASC|DESC]];

其中 All (默认)：返回全部记录； Distinct：略去选定字段中重复值的记录； From：指明字段的来源，即数据源表或查询； Where：定义查询条件； Group By：指明分组字段， Having：指明分组条件，必须跟随 Group By 使用； Order By：指明排序字段， Asc | Desc：排序方式，升序或降序。

例 3.27 查找并显示“教师”表中“姓名”、“性别”、“工作时间”和“系别”4 个字段

SELECT 姓名, 系别, 工作时间, 系别 FROM 教师;

例 3.28 查找 1992 年参加工作的男教师，并显示“姓名”、“性别”、“学历”、“职称”、“系别”和“联系电话”

SELECT 姓名, 性别, 学历, 职称, 系别, 联系电话 FROM 教师 WHERE 性别="男" AND 工作时间>=#1/1/1992#;

3.6.3 创建 SQL 特定查询

SQL 特定查询分为联合查询、传递查询、数据定义查询和子查询等 4 种。其中联合查询、传递查询、数据定义查询不能在查询“设计”视图中创建，必须直接在“SQL”视图中创建 SQL 语句。对于子查询，要在查询设计网络的“字段”行或“条件”行中输入 SQL 语句。

1. 联合查询

联合查询是将一个或多个表、一个或多个查询组合起来，形成一个完整的查询。执行联合查询时，将返回所包含的表或查询中对应字段的记录。

2. 传递查询

传递查询是自己并不执行而是传递给另外一个数据库来执行的查询。传递查询可直接将命令发送到 ODBC 数据库服务器中，如 SQL Server。使用传递查询时，不必与服务器上的表链接，就可以直接使用相应的表。

例 3-33 查询 SQL Server 数据库（名为“教学管理系统”）中“教师”表和“课程”表的信息，显示姓名，系别和课程名称字段的值。

注意：如果将传递查询转换为另一种类型的查询，例如选择查询，将丢失输入的 SQL 语句。如果在“ODBC



连接字符串”属性中没有指定连接串,或者删除了已有字符串,Access 将使用默认字符串“ODBC”,并且在每次运行查询时,提示连接信息。

3. 数据定义查询

数据定义查询与其他查询不同,利用它可以直接创建、删除或更改表,或者在当前数据库中创建索引。在数据定义查询中要输入 SQL 语句,每个数据定义查询只能由一个数据定义语句组成。

4. 子查询

在对 Access 表中的字段进行查询时,可以利用子查询的结果进行进一步的查询。不能将子查询作为单独的一个查询,必须与其他查询相结合。

3.7 编辑和使用查询

3.7.1 运行已创建的查询

- ❑ 通过工具栏上的“执行”按钮
- ❑ 通过“数据库”窗口中“查询”对象下的“打开按钮”
- ❑ 鼠标双击要运行的查询

3.7.2 编辑查询中的字段

- ❑ 添加字段
- ❑ 删除字段
- ❑ 移动字段

3.7.3 编辑查询中的数据源

- ❑ 添加表或查询
- ❑ 删除表或查询

3.7.4 调整查询的列宽

调整列宽的方法是:打开要修改查询的设计视图,将鼠标指针移到要更改列的字段选择器的右边界,使鼠标指针变成双向箭头,拖动鼠标改变列的宽度,双击鼠标可将其调整为“设计网格”中可见输入内容的最大宽度。

3.7.5 排序查询的结果

通过排序,查询中的记录指定顺序排列,可使显示的记录清晰、一目了然。

例 4-36 对例 3.10 的查询结果按成绩从低到高的顺序排列。

第 4 章 窗体

4.1 认识窗体

4.1.1 窗体的作用

窗体是应用程序和用户之间的接口,是创建数据库应用系统最基本的对象。用户通过使用窗体来实现数据维护、控制应用程序流程等人机交互的功能。

- ◆ 数据输入和编辑
- ◆ 信息显示和数据打印
- ◆ 控制应用程序流程

4.1.2 窗体的类型

Access 提供了 7 种类型的窗体,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、数据透视表窗体和数据透视图窗体。

1 纵栏式窗体

纵栏式窗体将窗体中的一个显示记录按列分隔,每列的左边显示字段,右边显示字段内容。

2 表格式窗体

通常,一个窗体在同一时刻只显示一条记录的信息。如果一条记录的内容比较少,单独占用一个窗体的空间就显得很浪费。这时,可以建立一种表格式窗体,即在一个窗体中显示多条记录的内容。



3 数据表窗体

数据表窗体从外观上看与数据表和查询的界面相同。数据表窗体的主要作用是作为一个窗体的子窗体。

4 主 / 子窗体

窗体中的窗体称为子窗体，包含子窗体的基本窗体成为主窗体。主窗体和子窗体通常用于显示多个表或查询中的数据，这些表或查询中的数据具有一对多的关系。

主窗体只能显示为纵栏式的窗体，子窗体可以显示为数据表窗体，也可以显示为表格式窗体。

5 图表窗体

图表窗体是利用 Microsoft Graph 以图表方式显示用户的数据。可以单独使用图表窗体，也可以在子窗体中使用图表窗体来增加窗体的功能。

6 数据透视表窗体

数据透视表窗体是 Access 为了以指定的数据表，或查询为数据源产生一个 Excel 的分析表而建立的一个窗体形式。数据透视表窗体允许用户对表格内的数据进行操作；用户也可以改变透视表的布局，以满足不同的数据分析方式和要求。

7. 数据透视图窗体

数据透视图窗体用于显示数据表和窗体中数据的图形分析窗体。数据透视图窗体允许通过拖动字段和项或通过显示和隐藏字段的下拉列表中项，查看不同级别的详细信息或指定布局。

4.1.3 窗体的视图

表和查询有两种视图：“数据表”视图和“设计”视图，而窗体有 5 种视图，即“设计”视图、“窗体”视图、“数据表”视图、“数据透视表”视图和“数据透视图”视图。“设计”视图是用于创建窗体或修改窗体的窗口；“窗体”视图是显示记录数据的窗口，主要用于添加或修改表中的数据；“数据表”视图是以表格的形式显示表、查询或窗体数据的窗口。“数据透视表”视图使用“Office 数据透视表”组件，易于进行交互式数据分析；“数据透视图”视图使用“Office Chart 组件”，帮助用户创建动态的交互式图表。

4.2 创建窗体

创建窗体有两种途径：一种是在窗体的设计视图下手动创建，另一种是使用 Access 提供的各种向导快速创建。

本节介绍如何利用系统提供的向导创建各类窗体。

4.2.1 自动创建窗体

Access 提供了两种方法自动创建窗体。一种是在“表”对象下使用“自动创建窗体”功能；另一种是在“窗体”对象下使用“自动创建窗体”功能。

1. 使用“自动窗体”

使用“自动窗体”功能是创建数据维护窗体最快捷的方法，它可以快速创建基于选定表或查询中所有字段及记录的窗体，其窗体布局结构简单规整。区别于其他窗体创建方法的是，自动窗体创建时，需先选定表对象或查询对象，而不是在窗体对象的窗口下启动向导或进入窗体设计视图。

2. 使用“自动创建窗体”

利用 Access 提供的“自动创建窗体”向导，可以创建 3 种形式的数据操作窗体：

- 纵栏式窗体
- 表格式窗体
- 数据表窗体

4.2.2 使用向导创建窗体

使用“自动窗体”方便快捷，但是内容和形式都受到限制，不能满足更为复杂的要求。使用“窗体向导”就可以更灵活、全面地控制数据来源和窗体格式，因为“窗体向导”能从多个表或查询中获取数据。

1. 创建单一数据源窗体

在“窗体”对象窗口中，双击“使用向导创建窗体”，或从“新建窗体”对话框中选择“窗体向导”选项，弹出“窗体向导”第一个对话框。



2. 创建涉及多个数据源的窗体

使用向导创建窗体更重要的应用是创建涉及多个数据源的数据维护窗体，也称此类窗体为主/子窗体。如果这些不同数据源之间的数据存在关联，那么就可以创建带有子窗体的窗体。

4.2.3 创建图表窗体

1. 数据透视表

数据透视表是一种特殊的表，用于从数据源的选定字段中分类汇总信息。数据透视表的两个主要元素是“轴”和“字段列表”。

(1) 轴是数据透视表窗口中的一个区域，它可能包含一个或多个字段的数据。在用户界面中，因为可以向轴中拖放字段，所以它们也被称为“拖放区域”。数据透视表有 4 个主要轴，每个轴都有不同的作用。4 个主要轴分别为“行字段”、“列字段”、“筛选字段”和“汇总或明细字段”。

(2) 字段列表的功能与查询或窗体中使用的字段列表的功能很相似。它根据窗体的“数据来源”(Record Source) 属性来显示可供数据透视表使用的字段。

2. 数据透视图

数据透视图是一种交互式的图表，功能与数据透视表类似，只不过以图形化的形式来表现数据。数据透视图能较为直观地反映数据之间的关系。

3. 创建图表窗体

使用图表窗体能够更直观的显示表和查询中的数据。可以使用“图表向导”创建图表窗体。

4.3 窗体的设计

在创建窗体的各种方法中，更多的时候是使用窗体设计视图来创建窗体，这种方法更直观、更灵活。创建何种窗体依赖于用户实际需求。在设计视图下创建窗体时，用户可以完全控制窗体的布局和外观，准确地把控件放在合适的位置，设置它们的格式直到达到满意的效果。

4.3.1 窗体设计视图

1. 窗体的组成和结构

窗体设计视图是设计窗体的窗口，它是由 5 个节组成，分别是主体、窗体页眉、页面页眉、页面页脚和视图页脚。



2. 工具栏 (见书本 P123)



3. 工具箱 (见书本 P124)

工具箱是窗体设计时最重要的应用，通过工具箱可以向窗体添加各种控件。控件是窗体中的对象，它在窗体中起着显示数据、执行操作以及修饰窗体的作用。

4. 字段列表

通常窗体都是基于某一个表或查询建立起来的，因此窗体内控件显示的是表或查询中的字段值。在创建窗体过程中当需要某一字段时，单击工具栏中的“字段列表”按钮 ，即可显示“字段列表”窗口。例如，要在窗体内创建  控件来显示字段列表中的某一文本型字段的数据时，只需将该字段拖到窗体内，窗体便自动创建一个文本框控件与此字段关联。

4.3.2 常用控件的功能

控件是窗体上用于显示数据、执行操作、装饰窗体的对象。在窗体中添加的每一个对象都是控件。Access 包含的控件有：标签、文本框、选项组、切换按钮、选项按钮、复选按钮、组合框、列表框、命令按钮、图像、非绑定对象框、绑定对象框、分页符、选项卡控件、子窗体 / 子报表、直线和矩形等。

控件的类型可以分为：绑定型、未绑定型与计算型 3 种。绑定型控件主要用于显示、输入、更新数据库中的字段；未绑定型控件没有数据源，可以显示信息、线条、矩形或图像；计算型控件用表达式作数据源，表达式可以利用窗体或报表所引用的表或查询字段中的数据，也可以是窗体或报表上的其他控件中的数据。

4.3.3 常用控件的使用 (详见书本 P127)



在窗体“设计”视图中，用户可以直接将一个或多个字段拖曳到主体节区域中，Access 会自动地为字段结合适当的控件或结合用户控件。结合适当的控件的操作方法是：单击窗体设计工具栏中的“字段列表”按钮，Access 则显示窗体数据源的字段列表，然后从表中拖曳某一个字段到主体节区域中。创建控件的方式取决于是要创建结合控件、非结合控件、还是计算控件。

4.3.4 窗体和控件的属性

在 Access 中，属性决定表，查询，字段、窗体及报表的特征。窗体和窗体上控件都有自己的一系列属性，这些属性决定了控件的外观、它所包含的数据，以及对鼠标或键盘事件的响应。

在属性表中，单击要设置的属性，在属性框中输入一个设置值或表达式可以设置该属性。如果属性框中显示有箭头，也可以单击该箭头，从列表选择一个数值。如果属性框的旁边显示“生成器”按钮，单击该按钮可以显示一个生成器或显示一个可以选择生成器的对话框，通过生成器可以设置对话框。

窗体的格式属性包括默认视图、滚动条、记录选定器、浏览按钮、分隔线、自动居中、控制框、最大化/最小化按钮、关闭按钮、边框样式等。这些属性都可以在窗体的属性对话框中设置。

在窗体设计视图下双击窗体选择器按钮，可打开窗体的属性对话框。

1. 应用条件格式

条件格式允许用户编辑基于输入值的字段格式。

2. 为窗体添加状态栏

要添加状态栏，只需选中要添加帮助的字段控件，在属性对话框的“其他”选项卡中的“状态栏文字”属性中输入帮助信息。保存所作的操作后，在窗体视图下当焦点落在指定控件上时，状态栏中就会显示出帮助信息。

3. 使用背景位图

在使用背景位图时，应设置窗体有关图片的相关属性。

4.4 格式化窗体

4.4.1 使用自动套用格式

在使用向导创建窗体时，用户可以从系统提供的固定样式中选择窗体格式，这些样式就是窗体的自动套用格式。这与 Office 的其他组件中的自动套用格式功能一样，把所有的格式属性的设定全部完成。

4.4.2 使用条件格式

除了可以利用自动套用格式对话框对窗体进行美化，还可以根据需要对窗体的格式、窗体的显示元素等进行美化设置。

4.4.3 添加当前日期和时间

如果用户希望在窗体中添加当前日期和时间，操作步骤为：

- (1)在“数据库”窗口中单击“窗体”对象。
- (2)单击要选择的窗体，单击“设计”按钮。
- (3)单击“插入”菜单的“日期和时间”命令，显示“日期与时间”对话框。
- (4)如插入日期和时间，则在对话框中选择“包含日期”和“包含时间”复选框。
- (5)在选择了某一项后，再选择日期和时间格式，然后单击“确定”按钮即可。

5.4.4 对齐窗体中的控件

1. 改变控件大小和控件定位

可以在控件的属性对话框中修改宽度和高度属性，也可在设计视图下选中控件后，用鼠标拖曳控件边框上的控制点来改变控件尺寸。

控件的精确定位可以在属性对话框中设置，也可以用鼠标完成。方法是保持控件的选中状态，按住 Ctrl 键不放，然后按下方向箭头移动控件直到正确的位置。

2. 将多个控件设置为相同尺寸

操作步骤如下：

- (1) 按住 Shift 键连续单击要设置的多个控件。
- (2) 选择“格式”→“大小”→“至最短”命令。

3. 将多个控件对齐

操作步骤如下：

- (1) 选中需要对齐的控件。
- (2) 选择“格式”→“对齐”→“靠左”或“靠右”命令，这样保证了控件之间垂直方向对齐，如果选择“靠上”或“靠下命令”，则保证水平对齐。

在水平对齐或垂直对齐的基础上，可以进一步设定等间距。假设已经设定了多个控件垂直方向的对齐，操作步骤如下：

- (1) 选中需要对齐的控件
- (2) 选择“格式”→“垂直间距”→“相同”命令。



第5章 报表

5.1 报表简介

5.1.1 报表的基本概念

报表是 Access 数据库中的一个对象,它根据指定的规则打印输出格式化的数据信息。报表的功能包括呈现格式化的数据;分组组织数据,进行数据汇总;报表之中包含子报表及图表;打印输出标签、发票、订单和信封等多种样式;可以进行计数、求平均、求和等统计计算;在报表中嵌入图像或图片来丰富数据显示的内容。

Access 的报表操作提供了 3 种视图:“设计”视图、“打印预览”视图和“版面预览”视图。“设计”视图用于创建和编辑报表的结构;“打印预览”视图用于查看报表的页面数据输出形态;“版面预览”视图用于查看报表版面设置。

报表由以下几部分组成:

报表页眉:在报表的开始处,用来显示报表的标题、图形或说明性文字,每份报表只有一个报表页眉。

页面页眉:用来显示报表中的字段名或对记录的分组名称,报表的每一页只有一个页面页眉。

主体:打印表或查询中的记录数据,是报表显示数据的主要区域。

页面页脚:打印在每页的底部,用来显示本页的汇总说明,报表只有一个页面页脚。

报表页脚:用来显示整份报表的汇总说明,在所有记录都被处理后,只打印在报表的结束处。

5.1.2 报表设计区

设计报表时,可以将各种类型的文本和字段控件放在报表“设计”窗体中的各个区域内,一次处理一条数据记录。

在报表的“设计”视图中,区段被表示成带状形式,称为“节”。

根据主体节内字段数据的显示位置,报表可划分为 4 种类型:

- 纵栏式报表
- 表格式报表
- 图表报表
- 标签报表

1、纵栏式报表

纵栏式报表(一也称为窗体报表),一般是在一页中主体节内显示一条或多条记录,而且以垂直方式显示。纵栏式报表记录数据的字段标题信息与字段记录数据一起被安排在每页的主体节区内显示。

2、表格式报表

表格式报表是以整齐的行、列形式显示记录数据,通常一行显示一条记录、一页显示多行记录。表格式报表与纵栏式报表不同,其记录数据的字段标题信息不是被安排在每页的主体节区内显示,而是安排在页面页眉节区内显示。

3、图表报表

图表报表是指包含图表显示的报表类型。报表中使用图表,可以更直观地表示数据之间的关系。

4、标签报表


标签报表是一种特殊类型的报表在实际应用中,经常使用标签,可以通过标签报表来实现。

5.2 创建报表

创建报表与创建窗体非常类似。报表和窗体都是使用控件来组织和显示数据的,因此,在第 4 章中介绍过的创建窗体的许多技巧也适用于创建报表。一旦创建了一个报表,就能够在报表中添加控件(包括创建计算型控件)、修改报表的样式等。

5.2.1 使用“自动报表”创建报表

方法一:在数据库窗口的表对象下,选定作为数据源的表,单击数据库工具栏上的“新对象”按钮(即环境右侧的向下三角形按钮),在下拉菜单中选择“自动报表”自动生成报表;或选择“报表”选项启动“新建报表”对话框。

方法二:在数据库窗口的报表对象下,单击“新建”按钮,在弹出的“新建报表”对话框中选择“自动创建报表”。

5.2.2 使用“报表向导”创建报表

使用“报表向导”创建报表时,向导会提示用户选择数据源、字段、版面及所需的格式,根据用户的选择来创建报表。在向导提示的步骤中,用户可以从多个数据源中选择字段,可以设置数据的排序和分组,产生各种汇总数据,还可以生成带子报表的报表。

5.2.3 使用“图表向导”创建报表

图表报表是 Access 特有的一种图表格式的报表,它用图表的形式表现数据库中的数据,相对普通报表来说数据表现的形式更直观。

用 Access 提供的“图表向导”可以创建图表报表。“图表向导”的功能十分强大,它提供了多达 20 种的图表形式供用户选择。

应用“图表向导”只能处理单一数据源的数据,如果需要从多个数据源中获取数据,须先创建一个基于多个数据源的查询,再在“图表向导”中选择此查询作为数据源创建图表报表。



标签是 Access 提供的一个非常实用的功能，利用它可将数据库中的数据加载到控件上，按照定义好的标签的格式打印标签。创建报表“标签向导”。“标签向导”的功能十分强大，不但支持标准型号的标签，也可以自定义尺寸制作标签。

5.2.5 使用“设计”视图创建报表

除了可以使用自动报表和向导功能创建报表以外，Access 中还可以从“设计”视图开始创建一个新报表，主要操作过程有：创建空白报表并选择数据源；添加页眉页脚；布置控件显示数据、文本和各种统计信息；设置报表排序和分组属性；设置报表和控件外观格式、大小位置和对齐方式等。

5.3 编辑报表（详见书本 P158）

5.4 报表排序和分组

5.4.1 记录排序

使用“报表向导”创建报表时，最多可以对 4 个字段进行排序。“报表向导”中设置字段排序，除有最多一次设置 4 个字节的限制外，排序依据还限制只能是字段，不能是表达式。实际上，一个报表最多可以安排 10 个字段或字段表达式进行排序。

5.4.2 记录分组

以记录的某个或多个特征（字段）分组，可使具有共同特征的相关记录组成一个集合，在显示或打印报表时，它们将集中在一起。对分组产生的每个集合，可以设置计算汇总等信息。一个报表最多可以对 10 个字段或表达式进行分组。

分组后的报表设计视图下，增加了“组页眉”和“组页脚”节。一般在组页眉中显示和输出用于分组的字段的值；组页脚用于添加计算型控件，实现对同组记录的数据汇总、计算和显示输出。不同组的数据可以显示或打印在同一页上，也可以通过设置，使之显示或打印在不同页上。

5.5 使用计算控件

在报表的实际应用中，除了显示和打印原始数据，还经常需要包含各种计算用做数据分析，得出某些结论性的结果。报表的高级应用包括对在报表中使用计算型控件，对报表进行排序、分组、统计汇总等。

5.5.1 报表添加计算控件

报表中也能加入计算型控件用来计算包含在报表中的数据。与窗体一样，通过向未绑定的文本框中输入表达式，可以在报表中创建计算型控件。表达式的格式与窗体中使用的表达式格式相同。

文本框是最常用来显示计算数值的控件，但是也可以使用任何有“控件来源”属性的控件。

5.5.2 报表统计计算

在 Access 中利用计算控件进行统计计算并输出结果操作主要有两种形式：

1 主体节内添加计算控件

在主体节内添加计算控件对每条记录的若干字段值进行求和或求平均计算时，只要设置计算控件的控件源为不同字段的计算表达式即可。

2 组页眉 / 组页脚节区内或报表页眉 / 报表页脚区内添加计算字段

在组页眉 / 组页脚节区内或报表页眉 / 报表页脚节区内添加计算字段对某些字段的一组记录或所有字段进行求和或求平均计算时，这种形式的统计计算一般是对报表字段列的纵向记录数据进行统计，而且要使用 Access 提供的内置统计函数来完成相应的计算操作。

5.6 创建子报表

1、在已有的报表中创建子报表

2、已有报表添加到其他已有报表来创建子报表

3、链接主报表和子报表

在插入包含与主报表数据具有相关的信息的子报表时，子报表控件必须与主报表相链接。该链接可以确保在子报表中打印的记录与在主报表中打印的记录保持正确的对应关系。

在通过向导创建子报表，或者直接将报表或数据表由“数据库”窗口拖到其他报表中来创建子报表时，如果满足下列条件，Access 将自动使子报表与主报表保持同步。

5.7 创建多列报表

创建多列报表的步骤如下：

(1)创建报表。

(2)单击“文件”菜单中的“页面设置”命令。

(3)在“页面设置”对话框中，单击“列”选项卡。

(4)在“网格设置”标题下的“列数”编辑框中键入每一页所需的列数。

(5)在“行间距”对话框中，键入主体节中每个记录之间所需的垂直距离。



(6)在“列间距”对话框中,键入各列之间所需的距离:

(7)在“列尺寸”标题下的“宽度”编辑框中键入所需的列宽,例如,3英寸。在“高度”编辑框中键入所需的高度值,即可设置主体节。或者在“设计”视图中直接调整节的高度。

(8)在“列布局”标题下单击“先列后行”或“先行后列”选项。

(9)单击“页”选项卡。

(10)在“打印方向”标题下单击“纵向”或“横向”选项。

(11)最后单击“确定”按钮。

5.8 设计复杂的报表

5.8.1 报表属性

用户可以单击工具条中的“属性”按钮或单击“视图”菜单中“属性”命令显示报表属性对话框。“属性”对话框内常用的属性为:记录源、打开、关闭、网络线 X 坐标、网络线 Y 坐标、打印版式、页面页眉、页面页脚、记录锁定、宽度、帮助文件和帮助上下文 ID 等。

5.8.2 节属性

节“属性”对话框内常用的属性为:强制分页、新行或新列、保持同页、可见性、可以扩大、可以缩小、格式化和打印等。

第6章 数据访问页

6.1 数据访问页的基本概念

6.1.1 数据访问页的作用

数据访问页是一个能够动态显示、添加、删除以及修改记录内容的特殊网页。用户既可以在 Internet 和 Intranet 上使用数据访问页,在网络上发布数据库信息,又可以通过电子邮件发送数据访问页。

用户可以根据需要设计:交互式报表、数据输入、数据分析等不同用途的数据访问页。

6.1.2 数据访问页的视图

数据访问页有 2 种视图方式,分别是设计视图和页面视图。

(1)设计视图是创建与设计数据访问页的一个可视化的集成界面。

(2)页面视图是查看所生成的数据访问页的一种视图方式。

6.1.3 数据访问页的调用方式

数据访问页是保存在 Access 数据库或 Access 项目文件外的独立文件。其调用方式,除了可以在 Access 数据库中打开,还可以利用 Internet Explore 浏览器打开。

1. 在 Access 中打开数据访问页

在 Access 中打开数据访问页并不是为了应用,而是为了测试。打开的方法是:在 Access“数据库”窗口的“页”对象中,选中要打开的数据访问页,然后单击窗口工具栏“打开”按钮,或直接双击要打开的数据访问页。

2. 在 IE 浏览器中打开数据访问页

利用 Internet Explore 浏览器打开数据访问页的方法是:在存放数据访问页的文件夹下,双击数据访问页文件(.htm);或先打开 Internet Explore 浏览器,然后在其地址栏中输入数据访问页文件路径。

6.1.4 数据访问页的数据来源

1. Access 数据库或 SQL Server 数据库

创建数据访问页应为其绑定一个 Access 数据库或 SQL Server 数据库。

2. 其他数据来源

数据访问页还可以包含其他数据源的数据。例如,Excel 工作表、非 Access 或 SQL Server 数据库中的数据透视表、电子表格或图表。

6.2 创建数据访问页

6.2.1 自动创建数据访问页

自动创建数据访问页,即使用“自动创建数据页”向导来创建数据访问页,是最简单最快捷的创建方法。使用这种向导可以快速创建基于单一数据源(表或查询)中所有字段(存储图片的字段除外)和记录的数据访问页。数据访问页格式由 Access 系统自动规定,用户不需要做任何设置。采用这种方式只能针对一个表建立纵栏式数据页。

6.2.2 使用向导创建页

使用 Access 提供的“数据页向导”可通过对话方式,让用户在创建过程中,根据自己的需要选择多个表或查询中的字段,选择按一个或多个字段升序或降序排列记录,也可以选择按某个字段来分组记录,然后由系统根据用户要求完成数据访问页的创建。

6.2.3 使用设计视图创建数据访问页

在创建数据访问页时,可以使用类似报表的设计视图修改已有的数据访问页,或直接在设计视图中设计新的数据访问页。使用页的设计视



图建立数据访问页的操作方法和过程类似于使用报表设计视图。

6.3 编辑数据访问页

6.3.1 添加标签

在数据访问页上使用标签控件的目的是用其来显示说明文本，如标题、字段内容说明等。标签并不显示字段或表达式数值，它是一种未绑定型控件，记录移动时，它们的值都不会改变。

在数据访问页中，“标签”控件的使用及格式属性设置，与窗体中一致。

6.3.2 添加命令按钮

在生成的数据访问页中，虽然记录导航工具栏可以为用户提供方便的数据浏览、编辑、删除等操作工具，但当不需要进行这些操作时，记录导航栏上的工具按钮就没有用途，此时可以删除导航工具栏，添加需要的命令按钮。

在数据访问页中，“命令按钮”控件的使用及格式属性设置，与窗体中一致。

6.3.3 插入图像编辑

单击“工具箱”中的“图像”控件，在数据访问页中，在需要插入图片的位置单击，拖曳鼠标指针直至所需的大小。然后松开鼠标左键，在弹出的对话框中选择要插入的图像，再单击“确定”按钮即可。

6.3.4 插入超链接控件

在数据访问页中，超级链接也是以控件的形式出现。要插入一个超级链接，可以单击控件工具箱中的“超级链接”按钮，然后像插入其他控件那样在数据访问页中拖曳鼠标画出一个矩形，然后松开鼠标左键，系统将弹出“插入超链接”对话框，在该对话框中，可以选择链接到一个原有的 Web 页文件，或者链接到本数据库中的某个数据访问页，还可以链接到一个新建的页或链接到一个电子邮件地址。选择需要链接的目标，并在对话框上部的“要显示的文字”文本框中输入超级链接的显示内容，然后单击“确定”按钮。

6.3.5 office 组件

在数据访问页中添加 Office 组件，可以简化数据分析的操作，让数据间的互动关系以可视的方式显示出来。Office 组件包括 Office 电子表格控件、Office 图表控件和 Office 数据透视表控件等。

6.3.6 设置滚动文字

1. 设置绑定型滚动文字控件

单击工具箱中的“滚动文字”按钮，放入数据访问页的合适位置上。然后，单击页设计工具栏中的“字段列表”按钮，打开“字段列表”。从“字段列表”中，将选中的字段拖动到数据访问页中的滚动文字控件上，其定位方法是，将图标的左上角放置到滚动文字控件左上角所需的位置。

2. 设置未绑定型滚动文字控件

单击工具箱中的“滚动文字”按钮，然后在数据访问页中准备放置滚动文字的位置单击。Access 将创建默认尺寸的滚动文字控件。如果需要创建特定大小的滚动文字控件，则应在数据访问页上拖放控件，直到获取所需的尺寸大小为止。在滚动文字控件中输入相关文本及格式，就形成了该滚动文字控件显示的信息。

3. 设置滚动文字的运动

滚动文字的默认运动方式为从左到右的运动。如果需要设定与之不同的运动方式，可通过设置滚动文字控件的 Behavior 属性来实现。

- (1) 将滚动文字控件的 Behavior 属性值设定为 Scroll，文字在控件中连续滚动；
- (2) 将滚动文字控件的 Behavior 属性值设定为 Slide，文字从开始滑动到控件的另一边，然后保持在屏幕上；
- (3) 将滚动文字控件的 Behavior 属性值设定为 Alternate，文字从开始到控件的另一边来回滚动，并且总是保持在屏幕上。

4. 更改滚动文字重复次数

通过设置滚动文字控件的 Loop 属性来实现。

- (1) 将滚动文字控件的 Loop 属性值设定为-1，文字连续滚动显示；
- (2) 将滚动文字控件的 Loop 属性值设定为一个大于零的整数，文字滚动指定的次数。例如，如果将 Loop 属性值设置为 10，文字将滚动 10 次，然后停止不动。

5. 更改文字滚动的速度

滚动文字控件的 True Speed 属性设置为 True 时，允许通过设置 Scroll Delay 属性值和 Scroll Amount 属性值来控制控件中文字的运动速度。

- (1) Scroll Delay 属性值用来控制滚动文字每个重复动作之间延迟的毫秒数；
- (2) Scroll Amount 属性值用来控制滚动的文本在一定时间内（该时间在“滚动延迟”属性框中指定）移动的像素数。

6. 更改滚动文字移动的方向

滚动文字控件的 Direction 属性值用来控制滚动文字控件中文字的运动方向。

- (1) Direction 属性值设置为 Left，滚动文字在控件中从左到右移动；
- (2) Direction 属性值设置为 Right，滚动文字在控件中从右到左移动；



(4) Direction 属性值设置 Up, 滚动文字在控件中从上到下移动。

6.3.7 设置背景

1. 设置背景颜色

设置背景颜色的方法是: 在“设计视图”中打开设置的数据访问页, 然后选择“格式”→“背景”→“颜色”命令, 从系统打开的颜色选择界面中, 单击所需颜色, 该颜色将成为数据访问页的背景颜色。

2. 设置背景图片

设置背景图片的方法是: 在“设计视图”中打开设置的数据访问页, 然后选择“格式”→“背景”→“图片”命令, 在系统打开的“插入图片”对话框中, 查找并选择作为背景的图片文件, 选择的图片将成为数据访问页的背景图片。

3. 设置背景声音

设置背景声音的方法是: 在“设计视图”中打开设置的数据访问页, 然后选择“格式”→“背景”→“声音”命令, 在系统打开的“插入声音文件”对话框中, 查找并选择作为背景声音的文件。以后打开该数据访问页时, 将自动播放该背景音乐。

这里需要说明两点: 第一, 在使用自定义背景颜色、图片或声音之前, 必须删除已经应用的主题; 第二, 在数据访问页中自定义“背景颜色”或“图片”后, 有些已设计的控件, 如“标签”控件, 没有得到显示, 可以使用“格式”→“置于顶层”命令, 把“标签”控件置于顶层。

第 7 章 宏

7.1 宏的功能

7.1.1 宏的基本概念

宏是指一个或多个操作的集合, 其中每个操作实现特定的功能, 例如打开某个窗体或打印某个报表。宏可以使某些普通的任务自动完成。

在 Access 中, 共定义了近 50 种这样的基本操作, 也叫宏命令。

Access 中宏可以分为: 操作序列宏、宏组合含有条件操作的条件宏。

宏组是共同存储在一个宏名下的相关宏的集合。

对于一些复杂的操作, 还可以使用条件宏, 即在执行宏的过程中按照一定的逻辑条件来决定执行哪些宏命令。

使用宏的好处:

创建的过程简单。不需编程, 不需记住各种复杂的语法, 即可实现某些特定的自动处理功能。

7.1.2 设置宏操作

Access 中提供了一系列基本的宏操作, 每个操作都有自己的参数, 可以按需要进行设置。常见宏操作和功能参见附录 D

7.2 建立宏

7.2.1 创建操作序列宏

(1) 在“宏”窗口的“操作”列单击第一个空白行。假如要在两个操作行之间插入一个操作, 单击插入行下面的操作行的行选定器, 然后在工具栏上单击“插入行”按钮。

(2) 在“操作”列, 请单击箭头显示操作列表。

(3) 选择要使用的操作。

(4) 可以在“操作”列右侧的“备注”栏中为操作键入相应的说明, 说明是可选的。

(5) 假如需要, 在窗口的下部指定参数。

宏窗口的组成:

- “操作”列
- “宏名”列
- “条件”列
- “注释”列
- “操作参数”部分
- “说明”部分

7.2.2 创建宏组

- 宏组由若干彼此相关的宏组成
- 宏组中的每个宏有自己的宏名, 执行宏组中的宏时需要在宏名前加宏组名, 形式为: 宏组名. 宏名。
- 建立宏组的目的是方便管理

7.2.3 条件操作宏

条件宏是设置了宏命令的执行条件的宏。



运行宏时先测试条件, 如果条件成立, 则执行对应的宏命令。否则, 不执行。

关于创建条件宏:

操作时需要先执行菜单命令“视图”→“条件”, 调出“条件”列。

◆ 其中条件是什么计算结果为 True/False 或“是/否”的表达式

例如: [Forms][按性别浏览学生].[frm 性别]=1

7.2.4 设置宏的操作参数

在宏中添加了某个操作之后, 可以在“宏”窗口的下部设置这个操作的参数。

关于设置操作参数的一些提示如下:

- (1) 可以在参数框中输入数值, 也可以从列表中选择某个设置。
- (2) 假如通过从“数据库”窗口拖曳数据库对象的方式来向宏中添加操作, 系统会设置适当的参数。
- (3) 假如操作中有调用数据库对象名的参数, 则可以将对象从“数据库”窗口中拖曳到参数框, 从而设置参数及其对应的对象类型参数。
- (4) 可以用前面加等号的表达式来设置许多操作参数。

7.2.5 运行宏

运行宏时, 系统按照宏中宏命令的排列顺序由上向下依次执行各个宏命令。

1. 直接运行宏

- (1) 从“宏”窗口中运行宏, 请单击工具栏上的“运行”按钮。
- (2) 从“数据库”窗口中运行宏, 请单击“宏”, 然后双击相应的宏名。
- (3) 从“工具”菜单上选择“宏”, 单击“运行宏”命令, 再选择选择或输入要运行的宏。
- (4) 使用 Docomd 对象的 RunMacro 方法, 在 VBA 代码过程中运行宏。

2. 运行宏组中的宏

- (1) 将宏指定为窗体或报表的事件属性设置, 或指定为 RunMacro 操作的宏名(Macro Name)参数。引用宏组的格式: 宏组名.宏名
- (2) 从“工具”菜单中选择“宏”选项, 单击“运行宏”命令, 再选择或输入要运行的宏组中的宏。
- (3) 使用 Docomd 对象的 RunMacro 方法, 在 VBA 代码过程中运行宏。

3. 通过窗体、报表或控件的事件运行宏或事件过程

Access 可以对窗体、报表或控件中的多种类型事件做出响应, 包括鼠标单击、数据更改以及窗体或报表打开或关闭等。

将窗体、报表或控件的适当事件属性设为宏的名称; 如果使用的是事件过程, 可以设为“事件过程”。

7.2.6 宏的调试

在 Access 系统中提供了“单步”执行的宏调试工具。使用单步执行宏, 可以观察宏的流程和每个操作的结果, 并且可以排除导致错误或产生非预期结果的操作。

调试的步骤:

- (1) 打开相应的宏。
- (2) 在工具栏上单击“单步”按钮。
- (3) 在工具栏上单击“运行”按钮。
- (4) 单击“单步”按钮, 以执行显示在“单步执行宏”对话框中的操作。
- (5) 单击“暂停”按钮, 以停止宏的运行并关闭对话框。
- (6) 单击“继续”以关闭单步执行, 并执行宏的未完成部分。

如果要在宏运行过程中暂停宏的执行, 然后再以单步运行宏, 请按[Ctrl+Break]快捷键

7.3 透过事件触发宏

7.3.1 事件的概念

事件是在数据库中执行的一种特殊操作, 是对象所能辨识和检测的动作, 当发生于某一个对象上时, 其对应的事件就会被触发。

事件是预先定义好的活动, 也就是说一个对象拥有哪些事件是由系统本身定义的, 至于事件被引发后要执行什么内容, 则由用户为此事件编写的宏或事件过程决定的。事件过程是为响应由用户或程序代码引发的事件或系统触发的事件而运行的过程。

宏运行的前提是有触发宏的事件发生。

7.3.2 通过事件触发宏

常用的触发宏的操作有:

- (1) 将宏和某个窗体、报表相连。

当其中的数据被修改的前后, 或者该窗体失去/获得焦点, 或者在窗体中执行了鼠标或键盘操作, 都可以设置使其触发运行宏。

- (2) 用菜单或工具栏上的某个命令按钮触发宏。



(3) 将宏和窗体、报表中的某个控件相连。

当单击了该控件，或其中的数据发生改变，或该控件失去/获得焦点时运行宏。

(4) 用快捷键触发执行宏。

(5) 制作 autoexec 宏。使得打开数据库时自动运行宏。

附：常见操作：

1 打开或关闭数据库对象

OpenForm 命令用于打开窗体。

OpenReport 命令用于打开报表。

OpenQuery 命令用于打开查询。

Close 命令用于关闭数据库对象。

2 运行和控制流程

RunSQL 命令用于执行指定的 SQL 语句。

RunApp 命令用于执行指定的外部应用程序。

Quit 命令用于退出 Access。

3 设置值

SetValue 命令用于设置属性值。

4 刷新、查找或定位记录

Requery 命令用于实施指定控件重新查询及刷新控件数据。

FindRecord 命令用于查找满足指定条件的第一条记录。

FindNext 命令用于查找满足指定条件的下一条记录。

GoToRecord 命令用于指定当前记录。

5 控制显示

Maximize 命令用于最大化激活窗口。

Minimize 命令用于最小化激活窗口。

Restore 命令用于将最大化或最小化窗口恢复至原始大小。

6 通知或警告用户

Beep 命令用于使计算机发出“嘟嘟”声。

MsgBox 命令用于显示消息框。

SetWarnings 命令用于关闭或打开系统消息。

7 导入和导出数据

TransferDatabase 命令用于从其他数据库导入和导出数据。

TransferText 命令用于从文本文件导入导出数据。

第 8 章 模块与 VBA 编程基础

8.1 模块的基本概念

模块是 access 系统中的一个重要对象，它以 VBA (Visual Basic for Applications) 为基础编写，以函数过程 (Function) 和子过程(Sub)为单元的集合方式存储。

在 access 中，模块分为类模块和标准模块两种类型。

8.1.1 类模块

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

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

窗体或报表模块中的过程可以调用已经添加到标准模块中的过程。

窗体和报表模块具有局部特性，其作用范围局限在所属窗体和报表内部，而生命周期则是伴随着窗体和报表的打开而开始，关闭而结束。

8.1.2 标准模块

标准模块一般用于存放供其他 access 数据库对象使用的公共过程。在系统中可以通过创建新的模块对象而进入其代码设计环境。

标准模块通常安排一些公共变量或过程供类模块里的过程调用。在各个标准模块内部也可以定义私有变量和私有过程仅供本模块内



标准模块中的公共变量和公共过程具有全局特性,其作用范围在整个应用程序里,而生命周期则是伴随着应用程序的运行而关闭而结束。

8.1.3 将宏转换为模块 (所有的宏都可以转换成模块代码)

使用宏可以启动应用程序: `doccmd.runapp`

在 Access 系统中,根据需要可以将设计好的宏对象转换为模块代码的形式。

8.2 创建模块

过程是模块的组成单元,由 VBA 代码编写而成。过程分两种类型: Sub 子过程和 Function 函数过程。

8.2.1 在模块中加入过程

模块是装着 VBA 代码的容器。在窗体和报表的设计视图中,单击工具栏“代码”按钮或者创建窗体和报表的事件过程可以加入类模块的设计和编辑窗口。单击数据库窗体中的“模块”对象标签,然后单击“新建”按钮即可进入标准模块的设计和编辑窗口。

一个模块包含一个声明区域,且可以包含一个或多个子过程或函数过程。

1、Sub 过程 (子过程)

执行一系列操作,无返回值。定义格式如下:

Sub 过程名

[程序代码]

End Sub

可以引用过程名来调用该子过程。此外,VBA 提供了一个关键字 Call,可显示调用一个子过程。

2、Function 过程 (函数过程)

执行一系列操作,有返回值。定义格式如下:

Function 过程名 As (返回值) 类型

[程序代码]

End Function

函数过程不能使用 Call 来调用执行,需要直接引用函数过程名,并直接在函数过程名后的括号所辨别。

8.2.2 在模块中执行宏

在模块的过程定义中,使用 DoCmd 对象的 RunMacro 方法,可以执行设计好的宏。其调用格式为:

`DoCmd.RunMacro MacroName [, RepeatCount][, RepeatExpression]`

其中 MacroName 表示当前数据库中宏的有效名称;RepeatCount 为可选项,用于计算宏运行次数的整数值;RepeatExpression 也是可选项,为数组表达式,在每一次运行宏时进行计算,结果为 False(0)时,停止运行宏。

8.3 VBA 程序设计基础

VBA 是 Microsoft Office 内置的编程语言,是根据 Visual Basic 简化的宏语言,其基本语法、词法与 Visual Basic 基本相同,因而具有简单、易学的特点。

与 Visual Basic 不同的是,VBA 不是一个独立的开发工具,一般被嵌入到像 Word、Excel、Access 这样的软件中,与其配套使用,从而实现在其中的程序开发功能。

8.3.1 面向对象程序设计的概念

1. 对象和集合

对象:在采用面向对象程序设计方法的程序中,程序处理的目标被抽象成了一个个对象,每个对象具有各自的属性、方法和事件。

类:是对一类相似对象的定义和描述。因此类可看做是对象的模板,每个对象由类来定义。**对象是类的实例**

集合:是由一组对象组成的集合,这些对象的类型可以相同,也可以不同。

Access 有几十个对象,其中包括对象和对象集合。所有对象和对象集合按层次结构组织,处在最上层的是 Application 对象,即 Access 应用程序,其他对象或对象集合都处在它的下层或更下层。

2. 属性和方法

对象的特征用属性和方法描述。

属性:用来表示对象的状态,如窗体的 Name (名称)属性、Caption (标题)属性等。

方法:用来描述对象的行为,如窗体有 Refresh 方法,Debug 对象有 Print 方法等。

引用对象的属性或方法时应该在属性名或方法名前加对象名,并用对象引用符“.”连接,即对象.属性或对象.行为

例如 `DoCmd.OpenReport “教师信息”`

是指利用 DoCmd 对象的 OpenReport 方法打开报表“教师信息”



3. 事件和事件过程

事件：是对象可以识别的动作，通常由系统预先定义

事件过程：对象在识别了所发生的事件后执行的程序

看懂例题并演示，下面的事件过程描述了单击按钮之后所发生的一系列动作。

```
Private Sub Command1_Click()  
    Me!Label1.Caption = "合肥领航教育"  
    Me!Text1 = ""  
End Sub
```

8.3.2 VBA 编程环境

一、Visual Basic 编辑器（详见书本 P202）

Visual Basic 编辑器 VBE（Visual Basic Editor）是编辑 VBA 代码时使用的界面。VBE 窗口主要由标准工具栏、工具窗口、属性窗口、代码窗口和立即窗口等组成。

二、进入 VBA 编程环境

Access 模块分成类模块和标准模块两种。

对于类模块，可以直接定位到窗体或报表，然后单击工具栏上的“代码”按钮进入；或定位到窗体、报表和控件上通过指定对象事件处理过程进入。其方法有两种：

（1）右键单击控件对象，单击快捷菜单上的“事件生成器命令”，打开“事件生成器”对话框，选择其中的“代码生成器”，单击“确定”按钮即可进入；

（2）单击属性窗口的“事件”选项卡，选中某个事件直接单击属性右侧的“...”按钮，打开“事件生成器”对话框，选择其中的“代码生成器”，单击“确定”按钮即可进入。

对于标准模块，有 3 种方法进入：

- （1）对于已存在的标准模块，只需从数据库窗体对象列表上选择“模块”，双击要查看的模块对象即可进入；
- （2）要创建新的标准模块，需要从数据库窗体对象列表上选择“模块”，单击工具栏上的“新建”按钮即可进入；
- （3）在数据库对象窗体中，选择“工具”菜单里“宏”子菜单的“Visual Basic 编辑器”选项即可进入。

三、VBE 环境中编写 VBA 代码

VBA 代码是由语句组成的，一条语句就是一行代码。例如

```
intCount=3                Rem 将 3 赋值给变量 intCount  
Debug. Print intCount    '在立即窗口打印变量 intCount 的值 3
```

Access 的 VBE 编辑环境提供了完整的开发和调试工具。其中的代码窗口顶部包含两个组合框，左侧为对象列表，右侧为过程列表。操作时，从左侧组合框选定一个对象后，右侧过程组合框中会列出该对象的所有事件过程，再从该对象过程列表选项中选择某个事件名称，系统会自动生成相应的事件过程模块，用户添加代码即可。

双击工程窗口中的任何类或对象都可以在代码窗口中打开相应代码进行编辑处理。

四、程序语句书写原则（详见书本 P206）

8.3.3 数据类型和数据库对象

一、标准数据类型

（1）布尔型数据（类似与我们所学的是否型数据类型）

布尔型数据只有两个值 True 或 False。布尔型数据转换为其他类型数据时，True 转换为-1，False 转换为 0；其他类型数据转换为布尔型数据时，0 转换为 False，其他类型转换为 True。

（2）日期型数据

“日期/时间”类型数据必须前后用“#”号封住。

如#2007-1-1#、#2002-5-4 14:30:00 PM#。

（3）变体类型数据

变体类型数据是特殊的数据类型。**VBA 中规定，如果没有显示声明或使用符号来定义变量的数据类型，则默认为变体类型。**

二、用户定义的数据类型（详见书本 P208）(要求选择填空会做题目)

应用过程中可以建立包含一个或多个 VBA 标准数据类型的数据类型，这就是用户定义数据类型。它不仅包含 VBA 的标准数据类型，还包含其他用户定义的数据类型。

用户定义数据类型可以在 Type ... End Type 关键字间定义，定义格式如下：

```
Type[数据类型名]
```



...

End Type

书上例题 8.3 定义一个学生数据类型

Type NewStudent

txtNo As String*7 ‘学号，7 位定长字符串

txtName As String ‘姓名，定长字符串

txtSex as String*1 ‘性别，1 位定长字符串

txtAge as Integer ‘年龄，整型

End Type

用户定义类型变量的取值，可以指明变量名以及分量名，两者之间用句点分隔，例如，定义一个学生信息类型变量 NewStud 并操作分量的例子如下：

Dim NewStud as NewStudent

NewStud.txtSno="980306"

NewStud.txtName="冯伟"

NewStud.txtSex="女"

NewStud.txtAge=20

可以用关键字 WITH 简化程序中重复的部分。例如，为上面 NewStud 变量赋值可以用：

With NewStud

.txtSno="980306"

.txtName="冯伟"

.txtSex="女"

.txtAge=20

End With

历年真题：

如下程序段定义了学生成绩的记录类型，由学号、姓名和三门课程成绩（百分制）组成。

Type Stud

no As Integer

name As String

score (1 to 3) As Single

End Type

若对某个学生的各个数据项进行赋值，下列程序段中正确的是（ ）。

A) Dim S As Stud

Stud.no=1001

Stud.name="舒宜"

Stud.score=78,88,96

B) Dim S As Stud

S.no=1001

S.name="舒宜"

S.score=78,88,96

C) Dim S As Stud

Stud.no=1001

Stud.name="舒宜"

Stud.score(1)=78

Stud.score(2)=88

Stud.score(3)=96

D) Dim S As Stud



```
S.no=1001
S.name="舒宜"
S.score(1)=78
S.score(2)=88
S.score(3)=96
```

有如下用户定义类型及操作语句：

```
Type Student
    No As String
    Name As String
    Age As Integer
End Type
Dim Stu As Student
With Stu
    .No = "200801001"
    .Name = "高强"
    .Age = 23
End With
```

执行 MsgBox Stu.Name 后，消息框输出结果是_____。

三、数据库对象（详见书本 P208）

8.3.4 变量与常量

变量是程序运行过程中值会发生变化的数据。如同一间旅馆客房，昨天可住旅客 A，今天住旅客 B，明天又有可能被闲置。变量的命名规则如下：

- (1) 以字母或汉字开头，后可跟字母、数字或下划线。
- (2) 变量名最长为 255 个字符。变量名不能含有空格。
- (3) 不区分变量名的大小写，不能使用关键字。
- (4) 字符之间必须并排书写，不能出现上下标。

以下是合法的变量名：

a, x, x3, BOOK_1, sum5

以下是非法的变量名：

3s s*T -3x bowy-1 if

常量是在程序中可以直接引用的实际值，其值在程序运行过程中不变。在 VBA 中，常量可以分为 3 种：直接常量、符号常量和系统常量

一、变量的声明

VBA 变量声明有两种方法。

1、显式声明

VBA 中定义变量的格式为：

Dim 变量名 [As 类型]

格式中 Dim 是一个 VBA 命令，此处用于定义变量；As 是关键字，此处用于指定变量的数据类型。

例如：Dim bAge as Integer 'bAge 为整型变量

2、隐含声明

VBA 允许用户在编写应用程序时，不声明变量而直接使用，这就是隐式声明。所有隐式声明的变量都是 Variant 数据类型。例如：

Dim m,n, 'm, n 为变体 Variant 变量 NewVar=528 'NewVar 为 Variant 类型变量，其值为 258。

二、强制声明

在默认情况下，VBA 允许在代码中使用未声明的变量，如果在模块设计窗口的顶部“通用—声明”区域中，加入语句：

Option Explicit

强制要求所有变量必须定义才能使用。这种方法只能为当前模块设置了自动变量声明功能，如果想为所有模块都启用此功能，可以单击菜单命令“工具”下“选项”对话框中，选中“要求变量声明”选项即可。

三、变量的作用域

(1) 局部范围 (Local)



变量定义在模块的过程内部，过程代码执行时才可见。在子过程或函数过程中定义的或直接使用的变量作用范围都是局部的。在子函数内部使用 Dim、Static...As 关键字说明的变量就是局部范围的。

(2) 模块范围 (Module)

变量定义在模块的所有过程之外的起始位置，运行时在模块所包含的所有子过程或函数过程中可见。在模块的通用说明区，用 Dim、Static、Private...As 关键字定义的变量作用域都是模块范围。

(3) 全局范围 (Public)

变量定义在标准模块的所有过程之外的起始位置，运行时在类模块和标准模块的所有子过程或函数过程中都可见。在标准模块的变量定义区域，用 Public...As 关键字说明的变量就属于全局的范围。

变量的持续时间 (生命周期) 是从变量定义语句所在的过程第一次运行，到程序代码执行完毕并将控制权交回调用它的过程为止的时间。

四. 数据库对象变量

Access 建立的数据库对象及其属性，均可被看成是 VBA 程序代码中的变量及其指定的值来加以引用。例如，Access 中窗体和报表对象的引用格式为：

Forms ! 窗体名称 ! 控件名称 [属性名称]

或 **Reports ! 报表名称 ! 控件名称 [属性名称]**

关键字 Forms 或 Reports 分别表示窗体或报表对象集合。感叹号“!”分隔开对象名称和控件名称。“属性名称”部分缺省，则为控件基本属性。

如果对象名称中含有空格或标点符号，就要用方括号把名称括起来。

五. 数组

数组是在有规则的结构中包含一种数据类型的一组数据，也称作数组元素变量。数组变量由变量名和数组下标构成，使用数组必须先定义数组。通常用 Dim 语句来定义数组，定义格式为：

Dim 数组名([<下标下限>to<下标上限>)[As <数据类型>]

缺省情况下，下标下限为 0，数组元素从“数组名(0)”至“数组名(下标上限)”；如果使用 to 选项，则可以安排非 0 下限。

例如，dim score (10) as Integer

定义了 11 个整形数构成的数组，数组元素为 score(0) 至 score(10)

再如，dim score (1 to 10) as Integer

定义了 10 个元素的整形数组，数组元素为 score(1)至 score(10)

注：(1) 所有数组元素在内存连续存放

(2) 根据下标区分数组元素

关于数组的定义，还有下面的几点说明：

(1) 定义数组时数组名的命名规则与变量名的命名规则相同。

(2) 一般在定义数组时应给出数组的上界和下界。但也可以省略下界，<下界>缺省 为 0。

例如，Dim a (10) As Single

默认情况下，数组 a 由 11 个元素组成。

若希望下标从 1 开始，可在模块的通用声明段使用 Option Base 语句声明。其使用格式为

Option Base 0! ! 后面的参数只能取 0 或 1

(3) <下界>和<上界>不能使用变量，必须是常量，常量可以是字面常量或符号常量，一般是整型常量。

(4) 如果省略 As 子句，则数组的类型为 Variant 变体类型。

二维数组的定义

格式为：

Dim 数组名([<下界>to<上界>, [<下界>to<上界>][As <数据类型>]

例如，Dim c (1 To 3, 1 To 4) As Single

c (1, 1)	c (1, 2)	c (1, 3)	c (1, 4)
c (2, 1)	c (2, 2)	c (2, 3)	c (2, 4)
c (3, 1)	c (3, 2)	c (3, 3)	c (3, 4)

七. 符号常量

如果程序中多处用到某个常量，将其定义成符号常量可增加代码的可读性，也便于维护。

符号常量用关键字 Const 来定义，格式如下：

Const 符号常量名称=常量值

例如：Const PI as single=3.1415926



定义了一个符号常量 PI，其值为 3.1415926

若是在模块的声明区中定义符号常量，则建立一个所有模块都可以用的全局符号常量。一般是 Const 前加 Global 或 Public 关键字。

八. 系统常量

这是一类特殊的符号常量，Access 系统内部包含若干个启动时就建立的系统常量，有 True、False、Yes、No、On、Off 和 Null 等。在编码时可以直接使用。

例如：

VBA 类库中，vbRed 代表红色，vbBlue 代表蓝色。

这里开始的两个字母表示所在的类库

Access 类库的常量以 ac 开始，如 acForm 等。

ADO 类库的常量以 ad 开始，如 adOpenKeyset。

8.3.5 常用标准函数 (本节请详见书本 P213)

在 VBA 中，除在模块创建中可以定义子过程与函数过程完成特定功能外，又提供了近百个内置的标准函数，可以方便完成许多操作。

标准函数一般用于表达式中，有的能和语句一样使用。其使用形式如下：

函数名(<参数 1><，参数 2>[，参数 3][，参数 4][，参数 5]...)

其中，函数名必不可少，函数的参数放在函数名后的圆括号中，参数可以是常量、变量或表达式，可以有一个或多个，少数函数为无参函数。

8.3.6 运算符和表达式 (详见书本 P219)

在 VBA 编程语言中，提供了许多运算符来完成各种形式的运算和处理。根据运算不同，可以分成 4 种类型的运算符：算术运算符、关系运算符、逻辑运算符和连接运算符。

8.4 VBA 流程控制语句

一个语句是能够完成某项操作的一条命令。VBA 程序的功能就是由大量的语句串命令构成。

VBA 程序语句按照其功能不同分成两大类型：

(1)声明语句，用于给变量、常量或过程定义命名；

(2)执行语句，用于执行赋值操作，调用过程，实现各种流程控制。

执行语句分为 3 种结构：

(1)顺序结构，按照语句顺序顺次执行。

(2)条件结构，又称为选择结构，根据条件选择执行路径。

(3)循环结构，重复执行某一段程序语句。

8.4.1 赋值语句

赋值语句是最基本的语句。它的功能是给变量或对象的属性赋值。其格式为

<变量名>=<表达式> 或 <对象名.属性>=<表达式>

例如：

Rate = 0.1 '给变量 Rate 赋值 0.1

Me!Text1.Value = "欢迎来到领航教育" '给控件的属性赋值

8.4.2 条件语句

1. If...Then 语句

语句格式为

If <表达式>

Then

<语句块 1>

End If

例如：

'输入一个数并在立即窗口输出其值

Dim x As Integer

x = InputBox("请输入 x 的值:")

If x Then

Debug.Print x

End If

例题：在窗体中添加一个名称为Command1的命令按钮，然后编写如下事件代码：

Private Sub Command1_Click()

A=75

If A<60 Then x=1



```
If A<70 Then x=2
```

```
If A<80 Then x=3
```

```
If A<90 Then x=4
```

```
MsgBox x
```

```
End Sub
```

窗体打开运行后，单击命令按钮，则消息框的输出结果是（ ）。

A) 1 B) 2 C) 3 D) 4

2. If...Then...Else 语句

语句格式为

```
If <表达式> Then
```

```
    <语句块 1>
```

```
Else
```

```
    <语句块 2>
```

```
End If
```

说明：

(1) <表达式>可以是任何表达式，一般为关系表达式或布尔表达式。如果是其他表达式，则非 0 认为是 True，0 认为是 False。

(2) 执行时，先判断表达式的值，为 True 则执行语句块 1，否则执行语句块 2。

例如：输入两个数并在立即窗口输出其中较大的数

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

```
x = InputBox("请输入 x 的值:")
```

```
y = InputBox("请输入 y 的值:")
```

```
If x > y Then
```

```
    MsgBox x
```

```
Else
```

```
    MsgBox y
```

```
End If
```

3. If...Then...ElseIf 语句

语句格式为

```
If <表达式 1> Then
```

```
    <语句块 1>
```

```
ElseIf <表达式 2> Then
```

```
    <语句块 2>
```

```
...
```

```
[ElseIf <表达式 n> Then
```

```
    <语句块 n>
```

```
Else
```

```
    <语句块 n+1> ]
```

```
End If
```

运行时，从表达式 1 开始逐个测试条件，当找到第一个为 True 的条件时，即执行该条件后所对应的语句块。

4. Select Case—End Select 语句

语句格式为



Select Case <变量或表达式>

Case <表达式 1>

语句块 1

Case <表达式 2>

语句块 2

...

[Case Else

语句块 n+1]

End Select

说明:

(1) Select Case 后的变量或表达式只能是数值型或字符型表达式。

(2) 执行过程是先计算 Select Case 后的变量或表达式的值, 然后从上至下逐个比较, 决定执行哪一个语句块。如果有多个 Case 后的表达式列表与其相匹配, 则只执行第一个 Case 后的语句块。

(3) 语句中的各个表达式列表应与 Select Case 后的变量或表达式同类型。各个表达式列表可以采用下面的形式

- 表达式: $a + 5$
- 用逗号分隔的一组枚举表达式: 2, 4, 6, 8
- 表达式 1 To 表达式 2 60 to 100
- Is 关系运算符表达式 $Is < 60$

5. 条件函数

除了上述条件语句外, VBA 还提供了 3 个函数来完成相应选择操作:

(1) If 函数: If(条件式, 表达式 1, 表达式 2)。

该函数根据“条件式”的值来决定函数返回值。“条件式”值为真, 函数返回“表达式 1”的值, “条件式”值为假, 函数返回“表达式 2”的值。

(2) Switch 函数: Switch(条件式 1, 表达式 1[, 条件式 2, 表达式 2][, 条件式 3, 表达式 3]...[, 条件式 n, 表达式 n])

该函数是分别根据“条件 1”, “条件 2”直至“条件 n”的值来决定函数的返回值。

(3) Choose 函数: Choose(索引式, 选项 1[, 选项 2]...[, 选项 n])

该函数式根据“索引式”的值来返回选项列表中的某个值。

8.4.3 循环语句

循环控制结构也叫重复控制结构。特点是程序执行时, 该语句中的一部分操作即循环体被重复执行多次。

循环语句可以实现重复执行一行或几行程序代码。VBA 支持以下循环语句结构:

(1) For...Next 语句。

(2) Do...Loop 语句。

(3) While...Wend 语句。

1. For--Next 循环语句

语句格式为

For <循环变量>=<初值> to <终值> [Step <步长>]

<循环体>

Exit For

<语句块>

Next <循环变量>

说明:

(1) 循环控制变量的类型必须是数值型。



- (2) 步长可以是正数，也可以是负数。如果步长为 1，Step 短语可以省略。
- (3) 根据初值、终值和步长，可以计算出循环的次数，因此 For 语句一般用于循环次数已知的情況。
- (4) 使用 Exit For 语句可以提前退出循环。

例 编程用 For 语句求 $1+2+3+\dots+10$ 之和

```
Public Sub gc2()  
    Dim s As Integer, i As Integer  
    s=0  
    For i = 1 To 10 Step 1  
        s = s + i  
    Next i  
    Debug.Print s  
End Sub
```

历年题目：

下面程序运行后，输出结果为（ ）。

```
Dim a()  
a=Array(1,3,5,7,9)  
s=0
```

```
For i=1 To 4  
    s=s*10+a(i)
```

```
Next i
```

```
Print s
```

A) 1357 B) 3579 C) 7531 D) 9753

历年真题：下列程序的输出结果是_____。

```
Private Sub Command1_Click()
```

```
Dim arr(1 To 10)
```

```
For i=1 To 10
```

```
    arr(i)=i
```

```
Next i
```

```
For Each i In arr()
```

```
    arr(i)=arr(i)*22+1
```

```
Next i
```

```
MsgBox arr(7)
```

```
End Sub
```

历年真题：在窗体上画一个命令按钮，其名称为Command1，然后编写如下事件过程：

```
Private Sub Command1_Click()
```

```
Dim a1(4,4),a2(4,4)
```

```
For i=1 To 4
```

```
    For j=1 To 4
```

```
        a1(i,j)=i+j
```

```
        a2(i,j)=a1(i,j)+i+j
```

```
    Next j
```

```
Next i
```

```
MsgBox(a1(3,3)*a2(3,3))
```

```
End Sub
```



2. Do While...Loop 语句

形式如下：

Do While <条件>

 循环体

Exit Do

 语句块

Loop

说明：

(1) 这里的条件可以是任何类型的表达式，非 0 为真，0 为假。

(2) 执行过程是：在每次循环开始时测试条件，对于 Do While 语句，如果条件成立，则执行循环体的内容，然后回到 Do While 处准备下一次循环；如果条件不成立，则退出循环。

(3) Exit Do 语句的作用是提前终止循环。

例如：下面的程序用 Do While...Loop 语句求 $1+2+3+\dots+10$ 之和。

```
Dim s As Integer, i As Integer
```

```
s = 0
```

```
i = 1
```

```
Do While i <= 10
```

```
    s = s + i
```

```
    i = i + 1
```

```
Loop
```

```
Debug.Print s
```

3. Do Until...Loop 语句

形式如下：

Do Until <条件>

 循环体

Exit Do

 语句块

Loop

说明：

(1) 这里的条件可以是任何类型的表达式，非 0 为真，0 为假。

(2) 执行过程是：在每次循环开始时测试条件，对于 Do Until 语句，如果条件不成立，则执行循环体的内容，然后回到 Do Until 处准备下一次循环；如果条件成立，则退出循环。

(3) Exit Do 语句的作用是提前终止循环。

例如：下面的程序用 Do Until...Loop 语句求 $1+2+3+\dots+10$ 之和。

```
Dim s As Integer, i As Integer
```

```
s = 0
```

```
i = 1
```

```
Do Until i > 10
```

```
    s = s + i
```

```
    i = i + 1
```

```
Loop
```

```
Debug.Print s
```

4. Do—Loop While 语句



格式如下：

Do

 循环体

Exit Do

 语句块

Loop While<条件>

说明：

和 Do While...Loop 不同的是，Do—Loop While 语句在每次循环结束时测试条件。因此，二者的区别是如果一开始循环条件就不成立，则 Do While...Loop 中的循环体部分一次也不执行，而 Do—Loop While 中的循环体部分被执行一次。

5. Do—Loop Until 语句

格式如下：

Do

 循环体

Exit Do

 语句块

Loop Until<条件>

说明：

和 Do Until...Loop 不同的是，Do—Loop Until 语句在每次循环结束时测试条件。因此，二者的区别是如果一开始循环条件就不成立，则 Do Until...Loop 中的循环体部分一次也不执行，而 Do—Loop Until 中的循环体部分被执行一次。

6. While---Wend 语句

格式如下：

While 条件式

 循环体

Wend

说明：

(1) While---Wend 循环与 Do While ...Loop 结构类似，但不能在 While---Wend 循环中使用 Exit Do 语句。

(2)在 VBA 中，尽量不要使用 While---Wend 循环。

8.4.4 其他语句--标号和 GoTo 语句

GoTo 语句用于实现无条件转移。

使用格式为：GoTo 标号。

程序运行到此结构，会无条件转移到其后的“标号”位置，并从那里继续执行。GoTo 语句使用时，“标号”位置必须首先在程序中定义好，否则转移无法实现。

8.5 过程调用和参数传递

- 过程是一段可以实现某个具体功能的代码
- 与函数不同，过程没有返回值
- 既可以在类模块中，也可以在标准模块中创建过程

1. 子过程的定义和调用

可以用 Sub 语句声明一个新的子过程、接收的参数和子过程代码。其定义格式为：

[Public|Private][Static] Sub 子过程名 ([<形参>])

 [<子过程语句>]

 <语句块>

 [Exit Sub]



<子过程语句>

End Sub

使用 **Public** 关键字可以使该过程适用于所有模块中的所有其他过程;使用 **Private** 关键字可以使该子过程只适用于同一个模块中的其他过程。

创建过程的方法是:

打开模块, 选择菜单“插入”→“过程”命令, 在“添加过程”对话框中输入过程名。

调用过程有以下两种格式:

格式 1: call 过程名 ([实参列表])

格式 2: 子过程名 [实参列表]

这里过程名后的参数叫实际参数。

2. 函数过程的定义和调用

可以使用 **Function** 语句定义一个新函数过程、接收参数、返回变量类型及运行函数过程的代码。其格式如下:

[Public | Private][Static]Function 函数过程名[<形参>][As 数据类型]

[<函数过程语句>]

[函数过程语句=<表达式>]

[Exit Function]

[<函数过程语句>]

[函数过程语句=<表达式>]

End Function

函数过程的调用格式只有一种:

函数过程名(<实参>)

由于函数过程返回一个值, 实际上, 函数过程的上述调用形式主要有两种方法: 一是将函数过程返回值作为赋值成分赋予某个变量, 其格式为“变量=函数过程名(<实参>)”;二是将函数过程返回值作为某个过程的实参成分使用。

3. 参数传递

过程定义时可以设置一个或多个形参(形式参数的简称), 多个形参之间用逗号分隔。其中, 每个形参的完整定义格式:

[Optional][ByVal | ByRef][ParamArray]varname[()][As type][=defaultvalue]

含参数的过程被调用时, 主调过程中的调用式必须提供相应的实参(实际参数的简称), 并通过实参向形参传递的方式完成过程操作。

在函数的调用过程中, 一般会发生数据的传递, 即将主调过程中的实参传给被调过程的形参。

在参数传递过程中, 可以有传址和传值两种形式。

1. 传址

如果在定义过程或函数时, 形参的变量名前不加任何前缀, 即为传址;若加前缀, 须在形参变量名前加 **ByRef**。

传递过程是: 调用过程时, 将实参的地址传给形参。因此如果在被调过程或函数中修改了形参的值, 则主调过程或函数中实参的值也跟着变化。

例如: 如果输入 5 和 7, 程序的结果会是多少?

```
Public Sub swap(x As Integer, y As Integer)
```

```
    Dim t As Integer
```

```
    t = x: x = y: y = t
```

```
End Sub
```

按钮的单击事件如下:



```
Private Sub Command0_Click()  
    Dim x As Integer, y As Integer  
    x = InputBox("x=")  
    y = InputBox("y=")  
    Debug.Print x, y  
    swap x, y  
    Debug.Print x, y  
End Sub
```

2. 传值

如果在定义过程或函数时，形参的变量名前加 ByVal 前缀，即为传值。这时主调过程将实参的值复制后传给被调过程的形参，因此如果在被调过程或函数中修改了形参的值，则主调过程或函数中实参的值不会跟着变化。

例如：如果输入 5 和 7，程序的结果会是多少？

```
Public Sub swap1(ByVal x As Integer, ByVal y As Integer)
```

```
    Dim t As Integer  
    t = x: x = y: y = t
```

```
End Sub
```

按钮的单击事件如下：

```
Private Sub Command0_Click()  
    Dim x As Integer, y As Integer  
    x = InputBox("x=")  
    y = InputBox("y=")  
    Debug.Print x, y  
    swap1 x, y  
    Debug.Print x, y  
End Sub
```

8.6 VBA 程序运行错误处理

VBA 中提供 On Error GoTo 语句来控制当有错误发生时程序的处理。

On Error GoTo 指令的一般语法如下：

On Error GoTo 标号

On Error Resume Next

On Error GoTo 0

“On Error GoTo 标号”语句在遇到错误发生时程序转移到标号所指定位置的代码处执行。

“On Error Resume Next”语句在遇到错误发生时不会考虑错误，并继续执行下一条语句。

“On Error GoTo 0”用于关闭错误处理。

8.7 VBA 程序的调试（详见书本 P238）

8.7.1 设置断点

所谓“断点”就是在过程的某个特定语句上设置一个位置点以中断程序的执行。“断点”的设置和使用贯穿在程序调试运行的整个过程中。

在程序中人为设置断点，当程序运行到设置了断点的语句时，会自动暂停运行并进入中断状态。

设置断点的方法是：在代码窗口中单击要设置断点的那一行语句左侧的灰色边界标识条。

再次单击边界标识条可取消断点。

8.7.2 单步跟踪

也可以单步跟踪程序的运行，即每执行一条语句后都自动进入中断状态。



单步跟踪程序的方法是：

将光标置于要执行的过程内，单击“调试”工具栏的“逐语句”按钮，或选择“调试”→“逐语句”命令。

8.7.3 设置监视点

即设置监视表达式。一旦监视表达式的值为真或改变，程序也会自动进入中断模式。

设置监视点的方法如下。

- (1) 选择“调试”→“添加监视”命令，弹出“添加监视”对话框。
- (2) 在“模块”下拉列表框中选择被监视过程所在的模块，在“过程”下拉列表框中选择要监视的过程，在“表达式”文本框中输入要监视的表达式。
- (3) 最后在“监视类型”栏中选择监视类型。