

ACCESS

在财务中的应用

杨密 杨乐 葛莹明 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Access 在财务中的应用

杨 密 杨 乐 葛莹明 编著

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书共分 10 章, 前 3 章从 Access 的基本用法着手, 以财务需求为例进行阐述; 后 7 章针对财务以及账务的要求, 逐一描述了用 Access 实现的方法。本书介绍了 Access 和 Excel 的结合应用, 其中第 6 章专门讨论在 Access 的窗体中直接生成 Excel 的财务报表。系统性和完整性是本书的一个重要特点, 为了方便读者使用, 最后一章为使用指南。

本书的读者对象为财务人员和软件开发人员。财务人员可以了解到什么是数据库软件, 为了方便数据处理, 财务人员应当准备什么样的数据; 另一方面, 数据库的开发人员可以了解一些必要的财务知识, 以及怎样才能满足财务数据处理的需求。

本书也可作为财务专业和软件应用专业在校学生的培训教材。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Access 在财务中的应用/杨密, 杨乐, 葛莹明编著. —北京: 电子工业出版社, 2006.9
ISBN 7-121-03122-1

I. A… II. ①杨…②杨…③葛… III. 关系数据库—数据库管理系统, Access—应用—企业管理: 财务管理 IV. F275-39

中国版本图书馆 CIP 数据核字 (2006) 第 100675 号

责任编辑: 孙延真

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 17.75 字数: 420 千字

印 次: 2006 年 9 月第 1 次印刷

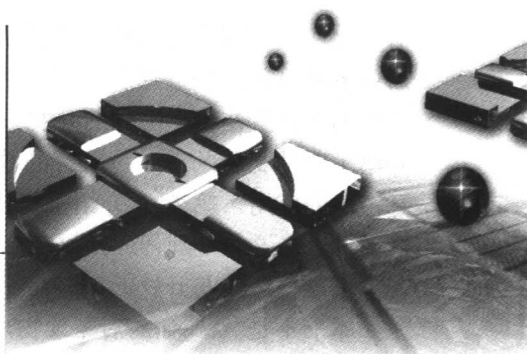
印 数: 5000 册 定价: 36.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言



财务管理在企业管理中非常重要，而现代企业的财务管理必须充分地结合数据库的应用。在财务管理（其他业务亦然）和数据库应用之间有个难点，即财务怎样和计算机应用相结合。一方面，财务人员怎样能用数据库的语言表达自己的需求；另一方面，软件开发人员怎样能尽快地掌握财务知识。本书针对这方面的难点进行了讨论，把财务管理的要求和数据库的应用密切联系在一起。

本书以 Access 为开发工具。虽然人们渐渐地开始了解 Access，但水平远远不够。1996 年，我们出版了《财务人员学用 Excel》一书，那时，计算机不像现在这样普及，熟悉办公软件的人也不太多。今非昔比，计算机普及了，软件功能增强了。然而，软件应用的发展并不如人们想像得那样快。很多人一定会提出我们早期遇到的问题：Excel 和 Access 各自的特点究竟是什么？简单地说，Excel 擅长处理局部的数据，可以进行局部数据的分析；Access 处理连续的数据，保持数据的连贯性。就财务应用来说，用 Access 进行凭证输入，全年的数据都存在于一个特定的 Access 文件（数据文件）中，然后，用 Excel 将财务报表输出。这样，两种软件的优点都体现了。

很多初学者会有这样的困惑：不知道怎样才能熟练掌握某种软件的使用方法。我们的经验是必须针对一种应用，持之以恒。可能不同的业务之间需要使用的技巧并不完全相同，然而，共性部分所占比例很大，否则就不会选择这种软件。例如比较财务处理和仓库管理，虽然两者的差异很大，但如果使用 Access，访问数据库的方法完全相同。针对一对多的数据结构，财务管理中会出现一个凭证多条借贷的记录；仓库管理中往往是一个入库单，多项入库物品。可见，无论从哪种业务切入，只要深入研究，就可以真正掌握。目前，熟悉 Excel 的人比了解 Access 的人多。我们认为，如果想处理连续的、常年的数据，Excel 并不适合。例如数据输入，在一个界面上完成总账、往来账等的数据输入，Excel 很难做到。如果在工作中需要和数据打交道，无论是否熟悉 Excel 都有必要学习一些 Access 的应用。

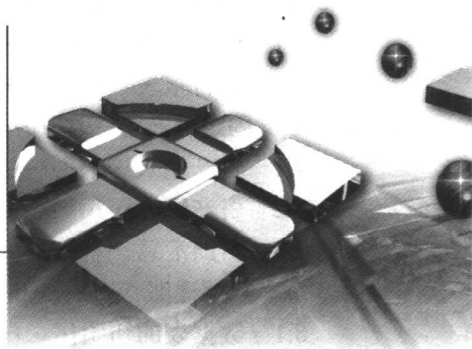
本书共分 10 章，第 1 章简要地介绍了数据库软件应用的一般方法，引出数据文件和窗体文件的概念。第 2 章和第 3 章分别就财务管理介绍了数据文件和窗体文件。数据文件看似简单，只是由若干数据表构成的文件，然而，数据文件是数据库软件的基础。针

对一项应用，需要什么样的数据表，数据表的字段怎样构成，都需要认真思考。只有精通业务并且熟悉数据库软件才能考虑周到，给出比较完善的结构设计。有了基础之后，主要的工作就在窗体文件的开发方面。从第 4 章到第 8 章，本书根据财务的需求及功能介绍了窗体文件的开发，所有的数据处理都包括在窗体文件中。实际上，软件的应用就是窗体的应用。第 9 章介绍了窗体文件所调用的数据表处理程序，为了保证正文部分的逻辑性，把这些处理程序安排在这一章。第 10 章则给出软件使用的示例，从而更好地展现本书软件应用与开发的完整性。

本书不是 Access 的启蒙读物，启蒙读物已经出版了很多。当然，我们也从启蒙读物中学习到很多知识。一方面，我们对这些作者表示感谢；另一方面，我们也把自己的经验与体会总结一下，并贡献出来。虽然我们的开发应用解决了一些实际问题，但不能说这是最好的方案。在软件应用中，套用一句广告语：“没有最好只有更好”。我们希望读者提出问题，共同研究、取长补短，提高该软件应用的能力。

作 者
2006 年 7 月

目 录



第 1 章 关于数据库	(1)
1.1 从 Excel 工作表谈起	(1)
1.2 关系型数据库	(2)
1.3 数据的输入与输出	(5)
1.4 数据表结构	(5)
1.5 数据的处理	(6)
1.6 数据库应用程序的构架	(11)
1.7 Access 数据库软件	(11)
第 2 章 财务管理中的数据表	(13)
2.1 数据表对象	(13)
2.2 按业务性质分类	(17)
2.3 按数据处理方式分类	(19)
2.4 数据表字段的说明	(20)
第 3 章 窗体文件	(31)
3.1 窗体文件中的对象	(31)
3.2 窗体的作用	(32)
3.3 财务软件的窗体说明	(34)
3.4 报表与数据表	(70)
第 4 章 凭证处理	(75)
4.1 凭证输入	(75)
4.2 凭证查询	(120)
4.3 凭证打印	(126)
第 5 章 账簿处理	(130)
5.1 日记账	(130)
5.2 明细账	(136)

5.3	总账	(155)
5.4	往来账	(163)
第 6 章	财务报表处理	(173)
6.1	财务报表	(173)
6.2	科目报表	(192)
第 7 章	固定资产	(195)
7.1	资产输入	(196)
7.2	资产账簿	(215)
7.3	折旧汇总	(219)
第 8 章	初期建账	(224)
8.1	数据表的初始化	(224)
8.2	年末结转	(228)
第 9 章	汇总程序	(230)
9.1	关于科目汇总表	(230)
9.2	关于往来账	(240)
9.3	关于部门费用	(245)
9.4	关于财务报表	(249)
第 10 章	使用手册	(256)
10.1	凭证处理	(257)
10.2	账簿处理	(262)
10.3	报表处理	(266)
10.4	固定资产处理	(268)
10.5	初期建账	(270)

第 1 章 关于数据库

本章阐述数据库的基本概念，为进一步的讨论打下基础。

1.1 从 Excel 工作表谈起

首先提出一个问题，一个仓库的材料保管员怎样用计算机进行材料管理呢？当然，前提是并没有现成的软件。也许，很多人都会想到使用 Office 中的软件之一——Excel。实际上，Excel 的确可以解决库存管理的问题。从库存业务的角度考虑，需要用 Excel 的工作表输入数据，把某种材料的入库情况和出库情况记录下来，参见图 1.1。



The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - 材料01'. The menu bar includes '文件(F)', '编辑(E)', '视图(V)', '插入(I)', '格式(O)', '工具(T)', '数据(D)', '窗口(W)', and '帮助(H)'. The toolbar contains various icons for file operations and formatting. The status bar at the bottom shows '宋体', '12', and 'B I U'.

	A	B	C	D	E	F	G
1	单证号	日期	代码	名称	入库数量	出库数量	
2	BH0001	2004-3-1	B-TJ-001	钛金板4*8*0.6	1000		
3	BH0002	2004-3-1	B-TJ-001	钛金板4*8*0.6		500	
4	BH0003	2004-3-1	B-TJ-003	钛金板4*8*1.0下	1000		
5	BH0004	2004-3-1	B-TJ-001	钛金板4*8*0.6		800	
6	BH0005	2004-3-1	B-TJ-003	钛金板4*8*1.0下	100		
7	BH0006	2004-3-1	B-TJ-001	钛金板4*8*0.6		1000	
8	BH0007	2004-3-1	B-TJ-003	钛金板4*8*1.0下	100		
9	BH0008	2004-3-1	B-TJ-001	钛金板4*8*0.6		50	
10	BH0009	2004-3-1	B-TJ-003	钛金板4*8*1.0下		100	

图 1.1 用 Excel 记录材料的入库出库情况

图 1.1 是 Excel 的工作表，其中有 9 条记录，每条记录包括 6 方面的内容，即单证号、日期、代码、名称、入库数量和出库数量。这样的记录为流水记录，即完全按物流发生的先后顺序输入的。虽然实际内容可能复杂得多，但此表已经完全可以解决数量的进出信息。

接下来的问题是，为了记录库存材料的信息，仅仅使用如图 1.1 所示的工作表是否就可以了？答案是否定的。图 1.1 仅仅记录了库存货物的发生情况，还必须有汇总情况。于是，需要第二个工作表。

图 1.2 是用 Excel 工作表建立的材料汇总表，该汇总表针对如图 1.1 所示的数据进行了汇总。在入库数量一列输入了公式“=SUMIF(code,\$A3,num_in)”，在出库数量一列输入了公式“=SUMIF(code,\$A3,num_ot)”，这样就可以自动将图 1.1 的数据汇总在图 1.2 上（注：

不在此讨论=SUMIF 函数的用法)。

材料汇总表						
代码	名称	单位	期初数量	入库数量	出库数量	期末数量
B-TJ-001	钛金板 4*8*0.6	张	2000	1000	2350	650
B-TJ-002	钛金板 4*8*0.8	张		0	0	0
B-TJ-003	钛金板 4*8*1.0下	张		1200	100	1100
B-TJ-004	钛金板 4*8*1.0上	张		n	n	n

图 1.2 材料入库出库汇总情况

实际上,图 1.1 和图 1.2 都是数据表,其特征是每列的信息属性相同,列标题做了说明,每行是一条记录。这两个工作表构成了一个小型的数据库。这样解释数据库或许有人会提出疑义,数据库似乎应该复杂得多。其实,数据库就是由数据表组成的。数据表的多少应该由需求决定。一个大型的数据库系统可以有上百个数据表,小型的系统有几个表就可以了。Excel 是由许多工作表组成的,最少必须有一个表,这完全符合数据库的特征。可以认为数据库是一种数据存放的形式,如果一个简单的系统只要用 Excel 就能解决,那就不需要采用更复杂的方案。然而,如果数据的内容太多,列数和记录数都很多,Excel 就不太适合了。Excel 的特点是表格形式,直接运算,比较适合信息量不大的数据表,若信息量太大,直接对数据表操作反而成了弊端。于是,针对大量的信息必须使用专门的数据库软件。

1.2 关系型数据库

数据库的概念和计算机产业的发展密切相关,早些时候,包括若干字段、许多记录的一个集合就被称为数据库。在使用 dBASE、FoxBASE 和 FoxPro 的时候,人们常常会对一个完整的数据处理系统问:“共有几个库?”现在,这样的认识应该修正。以前的数据库用现在的观点来认识只是一个表(为了明确也可以说成数据表),而现在认为一个数据库就是一个可以进行完整数据操作和处理的系统。就 Access 来说(数据库软件大都类似),一个数据库文件包括若干个表、查询、窗体、报表、模块等内容。这些内容(从专业角度应该说对象)各有各的用处,相互配合,组成一个有机的数据处理系统,这就是现在的数据库概念。

关系型数据库最基本的理解是指数据库中的数据表与表之间可以用相同的字段建立关联,这样做的好处是避免重复,减少数据量。例如,企业中销售人员和客户的关系,部门负责人和员工的关系。若用一个数据表来记录这样的信息当然可以,可是对应每一个客户的销售人员(或对应每一个员工的部门负责人)的信息有可能是相同的,必然会出现重复。若某个部门有 20 名员工,每一名员工对应的负责人的姓名、电话和家庭住址

都是相同的（当然应该假定每个部门一个负责人）。若用两个表，一个用来记录部门负责人的信息，一个用来记录员工的信息，并在这两个表之间用一个共同的字段建立关联。当然这个共同的字段也是一次重复，不过，这是最低限度的不可避免的重复。

在员工和负责人的例子中，我们只说了负责人的信息有姓名、电话和家庭住址（3个字段），若有更多的字段，减少的重复次数就十分可观了。

实际上，关系型数据库的功能远不止避免重复，可以用来保证数据的完整性，可以用来建立复杂的数据关系。这里所说的建立关系是指很容易地就可以把关系建立起来，例如使用鼠标单击就可以实现。对于非关系型的数据库，实现的手段就只有通过程序了。

表与表的关系是一个既简单又复杂的问题。说简单，指解释起来不复杂；说复杂，指使用起来不简单。很多书解释表关系的时候说关系有3种：即一对一、一对多和多对多。已经够简单了，却还可以再简化，实际上，可以认为只有一种：即一对多。

一对一的关系几乎不值得说明，既然是一对一，为何不放在一个表中，多设几个字段就可以了。有两种常见的解释：一是字段太多、一个表放不下；二是把需要做特别处理的字段信息（如保密）放在另一个表中。这两种解释或许有道理，不过在实际使用的时候有些牵强。对第一种解释很少会有人真正碰上，一个表可以最多容纳255个字段，意味着255种不同类型的信息，完全能满足一般用户的需求。第二种情况也未必常见，数据的访问一般不采用直接打开数据表的方法，而是通过窗体（即界面），这样处理就完全可以控制信息的显示或输出。

两个表之间多对多的关系在逻辑上是不能直接实现的，必须通过一个过渡表，由两组一对多的关系来实现。那么，就剩下一对多的关系了。

在图1.3中给出工资处理软件的表与表的关系，符号“∞”表示数据表多的一方。

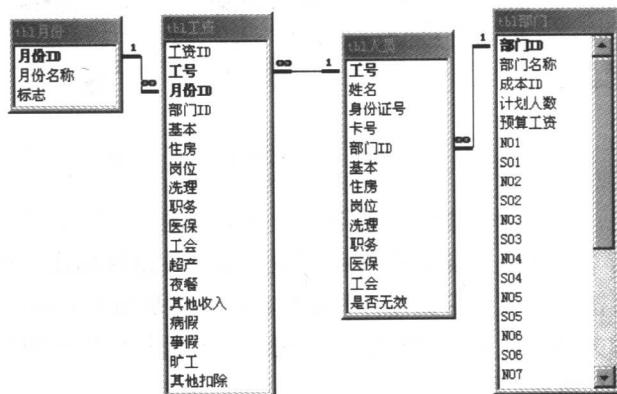


图 1.3 表与表之间的关系

图1.3中有4个表，“tbl 部门”、“tbl 人员”、“tbl 工资”和“tbl 月份”。部门表中存有部门的信息，每个部门只有（只应该有）一条记录，而人员表中每名员工有一条记录，

许多员工都可能属于同一个部门，在人员表中同一个部门是要重复出现的，这就是“多”的含义。

再看工资表，工资表和人员表以工号相关联。在人员表中，员工的记录是唯一的，每个员工一条记录；而工资表记录了发工资的次数（每人每次算一次），员工的记录一般每人每年发生 12 次，这就是“多”。

在数据库中，数据表与表的关系是根据数据处理的功能需要构建的。是否存在某种关系，是一对一还是多对多，完全按实际情况设定。

现在，我们来观察另一种数据表的关系：产品和订单的关系。

在产品和订单的关系中，每一种产品都会有多个订单，每一个订单都可能有多种产品，这也是一种典型的多对多关系。为了使订单和产品相关，就建立一个订单明细表，该表将记录每次订单的各种产品，参见图 1.4。

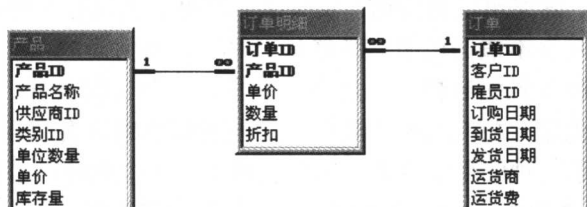


图 1.4 产品和订单形成多对多的关系

还有一个很值得一提的多对多关系，即在教学管理中老师、班级和学生的关系，参见图 1.5。

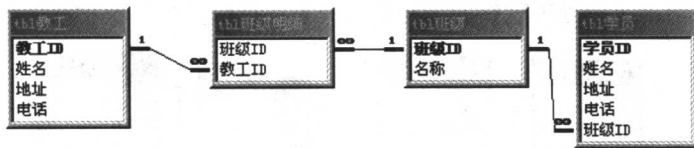


图 1.5 老师、班级和学生形成的关系

在实际情况中，老师和班级的关系不是一一对应的。每一个班级需要多个老师，而每一个老师又给若干个班级授课。班级和老师这样的关系就形成了多对多的关系，多对多的关系是不能直接关联的，需要有一个过渡表，这就是图 1.5 中的“tbl 班级明细”表。除了班级和老师的关系，还有班级和学生的关系。班级和学生构成了一对多的关系，一个学生只能隶属某个特定的班级。

通过图 1.5，我们很容易了解到学生和老师的关系也不是一一对应的，每一个班级的学生需要多个科目的老师；而每一个老师又有许多班级的学生。学生和老师的关系通过班级相关，也形成了多对多的关系，实际的情况也正是这样。

1.3 数据的输入与输出

前面提到,数据库指的是完整的数据处理系统,而不仅仅只是数据表,数据的输入与输出是数据库不可缺少的组成部分。

现在,随着计算机技术的发展,数据输入和输出的方法很多。然而,屏幕的输入与输出是最重要的手段。当然,打印输出也是很重要的。

屏幕的输入和输出涉及界面的设计,这是一项很复杂、却又很个性化的工作,往往需要反复多次才可以完成。在后面的章节中,介绍每一种应用都会涉及界面。可以说,在这样的处理中需要这样的界面,但却不能说提供的界面就是最好的。界面的设计除了考虑方便、适用外,还应考虑美观,正如现在的网页设计,是否美观,仁者见仁、智者见智。

1.4 数据表结构

数据表的结构必须符合要处理的数据,这是毫无疑问的。在实践中,我们认为有两个问题值得讨论:其一,一对多的关系中的重复数据;其二,字段与记录的关系问题。

1.4.1 一对多的关系中的重复数据

一对多的关系前面已经提到,避免重复数据是建立一对多关系的原因之一。然而,有时候,为了方便数据的使用,往往需要使数据重复,参见图 1.6。

图 1.6 可以见到两个数据表,“tbl 凭证明细”和“tbl 科目汇总”,两者之间的关系是通过科目 ID 字段建立的。这样可以保证 tbl 凭证明细中的科目 ID 必须输入,不可以有空缺,这正是财务系统中需要的,可是,当看到 tbl 凭证明细中同样也出现科目名称字段时就会认为不可理解,已经有了科目 ID,就不需要科目名称了,需要时可以根据 tbl 凭证明细中的科目 ID 向 tbl 科目汇总中查询获得。然而,在实际使用中往往需要更全面的考虑,查询是需要时间的,为此,不如多增加一个字段,数据虽然重复,却可以减少数据处理的时间。

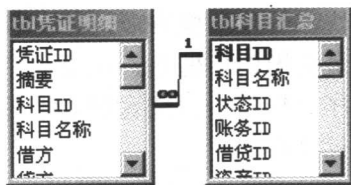


图 1.6 一对多关系中的重复数据

1.4.2 字段与记录的关系问题

在数据处理中,汇总的数据应该怎样保存是个值得研究的问题。在与财务有关的数

据处理中，这样的问题也会常常涉及。例如，某个科目借方金额和贷方金额的按月汇总数据应该怎样保存呢？可以有两种方法处理：多字段和多记录。

多字段的方法就是多建字段，每个月的借方和贷方都建立字段。那么，科目数据表中除了包括科目代码、科目名称等字段外，关于月汇总数据的字段就必须有 37 个，而且，这些字段数据的值是首尾相关的，前一个月的期末值就是后一个月的期初值，字段的名称也必须符合一定的规律。如表 1.1 所示为多字段的方案。

表 1.1 多字段的方案

科目代码	科目名称	M00	D01	C01	M01
D02	C02	M02	D03	C03	M03
D04	C04	M04	D05	C05	M05
D06	C06	M06	D07	C07	M07
D08	C08	M08	D09	C09	M09
D10	C10	M10	D11	C11	M11
D12	C12	M12			

若采用多记录的方法，情况就不一样了，多记录字段的方案如表 1.2 所示。

表 1.2 多记录字段的方案

科目代码	科目名称	月份	期初	借方	贷方	期末
------	------	----	----	----	----	----

那么，多字段和多记录究竟有怎样的区别呢？我们认为，多字段的处理速度较快，而且结构清晰。其缺陷是字段不可持续地增加。每年年末、年初都必须初始化。最关键的问题是，在用程序对数据处理时，程序的写法很不相同，多记录的方法程序比较复杂。究竟应当采用多记录的方法还是多字段的方法，应根据具体的情况进行分析。

1.5 数据的处理

数据处理一般指对数据表增加记录、删除记录或对已有的数据进行修改。这样的工作往往都是由程序完成的。要处理数据，首先必须进入数据表。可以说，在数据库软件应用中最重要的是对数据库的访问。访问数据库的目的可分成 3 种情况：读出数据、修改数据和增加数据。结合 Visual Basic 语言，可以用两种方式对数据库进行访问，即 DAO 和 ADO。ADO 方式比 DAO 方式先进，不过，一般的用户并不能体会。例如，我们需要访问数据表“tbl 月份”，参见图 1.7。

月份ID	月份	总账	往来	库存
200301	2003-01	☒	☒	☐
200302	2003-02	☐	☐	☐
200303	2003-03	☐	☐	☐
200304	2003-04	☐	☐	☐
200305	2003-05	☐	☐	☐
200306	2003-06	☐	☐	☐
200307	2003-07	☐	☐	☐
200308	2003-08	☐	☐	☐
200309	2003-09	☐	☐	☐
200310	2003-10	☐	☐	☐
200311	2003-11	☐	☐	☐
200312	2003-12	☐	☐	☐

图 1.7 tbl 月份数据表中的记录

访问 tbl 月份数据表的目的是要得出总账、往来账或库存账的首次使用月份。该月份值和需要进行数据处理月份进行对比。

程序 1-1 DAO 方式

```
Function mMonthCheck(mMon As Long, mFld As String) As Boolean
```

```
    Dim mDB As DAO.Database
```

```
    Dim mRst As DAO.Recordset
```

```
    Dim mStr As String
```

```
    Dim mM As Long
```

```
    Set mDB = CurrentDb
```

```
    mStr = "select * From tbl 月份 where " & mFld & " = -1"
```

```
    Set mRst = mDB.OpenRecordset(mStr)
```

```
    mM = 0
```

```
    mM = mRst!月份 ID
```

```
    mDB.Close
```

```
    mRst.Close
```

```
    Set mDB = Nothing
```

```
    Set mRst = Nothing
```

```
    '以上为访问数据库的 DAO 方式
```

```
    If mM = 0 Then
```

```
        MsgBox "总账的使用月份未设定!"
```

```
        mMonthCheck = True
```

```
        Exit Function
```

```
    End If
```

```
    If mMon - mM < 0 Then
```

```
        MsgBox "选择的月份不得小于" & mMon
```

```
        mMonthCheck = True
```

```
        Exit Function
```



```
End If
mMonthCheck = False
End Function
```

程序 1-2 ADO 方式

```
Function mMonthCheck(mMon As Long, mFld As String) As Boolean
    Dim mCnn As ADODB.Connection
    Dim mRstAdoM As ADODB.Recordset
    Dim mStrAdoM As String
    Dim mM As Long
    Set mCnn = CurrentProject.Connection
    mCnn.CursorLocation = adUseClient
    mCnn.Open mCnnStr
    Set mRstAdoM = New ADODB.Recordset
    mRstAdoM.ActiveConnection = mCnn
    mRstAdoM.CursorType = adOpenKeyset
    mRstAdoM.LockType = adLockOptimistic
    mStrAdoM = "select * From tbl 月份 where " & mFld & " = -1"
    mRstAdoM.Open mStrAdoM
    mM = 0
    mM = mRstAdoM!月份 ID
    mRstAdoM.Close
    Set mRstAdoM = Nothing
    '以上为访问数据库的 ADO 方式
    If mM = 0 Then
        MsgBox "总账的使用月份未设定!"
        mMonthCheck = True
        Exit Function
    End If
    If mMon - mM < 0 Then
        MsgBox "选择的月份不得小于" & mMon
        mMonthCheck = True
        Exit Function
    End If
    mMonthCheck = False
End Function
```

比较程序 1-1 (DAO 方式) 和程序 1-2 (ADO 方式) 可以知道, 访问数据的语句不同。就这两段程序而言, DAO 方式比 ADO 方式简单。不过, 无论哪种方式, 两者都有一些共同的法则。

1.5.1 定义变量

定义变量是编程的一般法则，然而针对 DAO 和 ADO，必须分清两者的不同。

```
' 定义 DAO 的对象变量
Dim mDB As DAO.Database
Dim mRst As DAO.Recordset

' 定义 ADO 的对象变量
Dim mCnn As ADODB.Connection
Dim mRstAdoM As ADODB.Recordset
```

1.5.2 打开数据文件

打开数据源实际上就是打开数据文件。访问数据是对某个文件的访问，例如，需要打开自身文件就应当采用如下的方法。

```
' 打开 DAO 的数据文件
Set mDB = CurrentDb

' 打开 ADO 的数据文件
Set mCnn = CurrentProject.Connection
```

1.5.3 打开数据表

访问数据最终是访问数据文件的某些数据表，打开数据文件后必须指定访问的数据表。

```
' 用 DAO 方式打开 tbl 月份数据表
mStr = "select * From tbl 月份 where " & mFld & "=-1"
Set mRst = mDB.OpenRecordset(mStr)

' 用 ADO 方式打开 tbl 月份数据表
mStrAdoM = "select * From tbl 月份 where " & mFld & "=-1"
mRstAdoM.Open mStrAdoM mCnn
```

1.5.4 多个数据文件和多个数据表

掌握了访问数据表的一般方法之后，可以根据需要访问多个数据表。可以在一个数据文件下，访问多个数据表，还可以访问多个数据文件。


```
' 访问 E 盘 Traders-2003 目录下的一个文件 Traders_BE.mdb
mCnnStr = "driver={microsoft access driver (*.mdb)};DBQ=" _
    "E:\Traders-2003\Traders_BE.mdb;password=12345"
Set mCnn = New ADODB.Connection
mCnn.CursorLocation = adUseClient
mCnn.Open mCnnStr

' 访问 Traders_BE.mdb 中的 tbl 科目汇总
Set mRstKM = New ADODB.Recordset
mRstKM.ActiveConnection = mCnn
mRstKM.CursorType = adOpenKeyset
mRstKM.LockType = adLockOptimistic
mStrKM = "Select * From tbl 科目汇总 Where 状态 ID > 0 Order by 科目 ID"
mRstKM.Open mStrKM, options:=adCmdText

' 访问 Traders_BE.mdb 中的 tbl 凭证记录
Set mRstAdo = New ADODB.Recordset
mRstAdo.ActiveConnection = mCnn
mRstAdo.CursorType = adOpenKeyset
mRstAdo.LockType = adLockOptimistic
mStrAdo = "select * from tbl 凭证记录 where 月份 ID=" & mMonth & " Order by 凭证号"
mRstAdo.Open mStrAdo

' 访问 Traders_BE.mdb 中的 tbl 凭证明细
Set mRstAd2 = New ADODB.Recordset
mRstAd2.ActiveConnection = mCnn
mRstAd2.CursorType = adOpenKeyset
mRstAd2.LockType = adLockOptimistic

' 访问自身的 tbl 凭证明细
Set mRstL = New ADODB.Recordset
mRstL.ActiveConnection = CurrentProject.Connection
mRstL.CursorType = adOpenKeyset
mRstL.LockType = adLockOptimistic
```

以上的指令可以在一个模块中同时出现。