

第一章 概 述

§ 1.1 dBASE ■ 数据库管理系统简介

1.1.1 数据库及数据库管理系统

数据库,是动态存储的有组织的关联数据的集合。它有这样一些特点:

1. 数据的共享性 数据库系统可以保存一个部门的所有各类数据,进行集中控制。这些数据可为多个不同类型的用户所共享,用于各种应用目的。同时,数据库系统能客观地反映出数据间存在的联系,对数据进行有效地组织,减少数据的冗余度,保证数据的相容性和正确性。

2. 数据的独立性 在数据库技术中,应用程序不依赖于数据的组织和物理存储位置。所以,当数据的结构需要修改时,不必修改相应的应用程序。同时,不同的应用程序可以按各自定义的逻辑数据结构去存取数据库中的数据,不需要改动数据的物理存取方式。

3. 数据的安全性 数据库管理系统对数据库中各个用户的数据采取不同的安全和保密措施。因此,在多用户共享数据库的情况下,保证了数据的安全性。

4. 数据的完整性 数据库系统通过转储、运行日志、修改、恢复等功能来保证数据的准确性,并对用户的操作加以适当的控制,保证存入的数据符合有关规定。

数据库管理系统,指的是对其数据库进行管理的软件系统。它包括数据描述语言及其翻译程序、数据操作语言及其编译程序、数据库

管理例行程序等。

1. 1. 2 dBASE Ⅲ 数据库

dBASE Ⅲ 是美国 Ashton-Tate 公司在原有 dBASE Ⅱ 的基础上发展出来的,适用于 IBM-PC/XT 及其全兼容机上的关系型数据库管理系统。dBASE Ⅲ 较 dBASE Ⅱ 在功能上,特性上有很大改进,它不仅具有 dBASE Ⅱ 的所有功能,而且在处理能力及范围上又有了加强,还增加了一些操作命令。dBASE Ⅲ 可方便地建立、修改和删除一个文件,灵活地进行数据的添加、删除、编辑、修改、打印和查询等,还可以进行一般的算术运算、总计和小计等。同时它支持全屏幕编辑,具有更完善的命令语言。

dBASE Ⅲ 引进到国内经汉化后,不仅保留了原来的全部功能特点,而且还可以处理汉字,如表格名称、栏目名称以及可直接用汉字表示数据。这就是中文的关系型数据库管理系统 dBASE Ⅲ。它为用户提供了极大的方便,为普及微型计算机开辟了广阔的道路。

§ 1. 2 dBASE Ⅲ 的规范及说明

用关系型数据库 dBASE Ⅲ 来建立、修改和使用自己的数据库时,必须首先了解该系统的一些基本问题。这些基本问题涉及到系统的环境要求和性能指标、数据库的基本结构、数据库的类型和范围、常用的各种函数、对数据进行有关操作的命令、与数据库有关的各种计算机文件的建立和使用等基本概念。

1. 2. 1 环境要求、系统结构、性能指标及运行

一、环境要求

1. 硬件环境

IBM-PC/XT 或任何其它的兼容机(如国产长城微型计算机等);

使用中文操作系统 CCDOS 软汉字库时,最少要求 512KB 字节的内存容量,使用硬汉字库时内存容量可相应减少;

至少两个 360KB 软盘驱动器或一个 360KB 软盘驱动器加一个硬盘驱动器。现在不少微机系统已配有高密软盘驱动器及大容量硬盘,为 dBASE Ⅲ 的运行提供了良好的环境;

行宽 80 列以上的打印机。

2. 软件环境

MS-DOS 或 PC-DOS (DOS 2.0 以上版本)以及各种版本的 CCDOS 系统基本上都适用。长城系列机也有其相应的 dBASE Ⅲ。

3. 微机系统配置

每当开机启动系统时,系统就要检查一个系统配置文件 CONFIG. SYS,此文件指明了微机运行的环境信息。dBASE Ⅲ 要求 CONFIG. SYS 文件中有以下两条命令:

FILES=20

BUFFERS=24

如果你的硬盘中没有 CONFIG. SYS,则可用文本编辑程序 EDLIN 或 WS 生成一个,也可以在操作系统下使用 COPY 命令从键盘上建立:

A)COPY└CON:CONFIG. SYS↙

FILES=20↙

BUFFERS=24↙

^Z↙

将此文件拷贝到启动盘上。尔后,每当人们使用 dBASE Ⅲ 时,机器必须是含有 CONFIG. SYS 的系统盘启动,否则在运行 dBASE Ⅲ 数据库管理系统时将无法正常工作。

PC 机系统正常情况下允许同一时刻打开至多八个文件,且本身用去其中的五个。dBASE Ⅲ 允许同时打开 10 个数据文件或 13 个其它类型的文件,因此,有可能出现较 PC 机正常情况下更多的文件,故必须设置 FILES 命令。FILES 命令并不占用很大空间,因此取

FILES=20 不会发生任何问题。

开辟缓冲区使得 dBASE III 在内存中保留更多的信息,因而运行速度更快。但缓冲区要占用更多的内存空间,如果发现运行程序时内存空间不够就应减少缓冲区的数目,但最少要保留 12 个。

二、系统结构

dBASE III 是用 C 语言编写的(只是控制台 I/O 部分及与操作系统的接口部分采用了汇编语言)。由四个基本模块和两个实用程序组成,四个基本模块是:

dBASE.EXE(117KB)

dBASE.OVL(149KB)

HELP.DBS(54KB)

ASSIST.HLP(15KB)

以上共约 335KB,由于 PC-DOS 系统本身在软盘上占用 41KB,命令文件 COMMAND.COM 占用 18KB,故若要在一张 360KB 的软盘上全部装上 dBASE III 的四个模块及 DOS 系统是做不到的。所以通常是将 dBASE III 的四个模块存放在一张不含 DOS 系统的盘片上,将 CONFIG.SYS 存放在 DOS 启动盘上,用 DOS 盘启动机器后,再将 dBASE III 盘插入机器供用户使用。

带有硬盘的机器就不存在这样的问题。

dBASE III 的两个实用程序如下:

dFORMAT.EXE(39KB)

dCONVERT.EXT(48KB)

实用程序 dFORMAT.EXE 主要用于帮助你生成在屏幕上输入输出或在打印机上输出的格式。

实用程序 dCONVERT 能帮助你把原来在 dBASE III 下运行的程序或文件转换成 dBASE III 下可用的新文件。

运行 dBASE III 应用程序时对微机的内存(RAM)也有一定的要求。

在西文工作状态下,运行 dBASE III 至少要求系统有 256KB 的

内存容量。而在最苛刻的情况下——用户同时打开了 10 个数据文件,每个数据库文件记录均由 128 个字段组成,且填满 4000 个字节,同时,每个数据库均建立了七个索引文件,这样 250KB 内存仍不够用,此时必须拥有 320KB 内存方可运行 dBASE III,倘若要在 dBASE III 内使用 RUN 命令运行 PC-DOS 的命令及其它程序,则要求系统有更大的内存容量。

在中文工作状态下,使用软汉字库需要更大的内存空间,对微机内存的要求应选择大于 640KB 为好,一般后期的微机内存都较大,能满足 dBASE III 的运行要求。

三、dBASE III 性能指标

关系型数据库管理系统 dBASE III 可用于交互式运行或应用程序开发,它具有完整的关系运算及可编程特性,使用方便,具有“自动运行方式”,有明确的 HELP 提示信息。

数据库文件的指标:

每库记录条数	≤10 亿条
每库字节总数	≤20 亿字节
每库记录最大长度	4000 个字符
字段个数	≤128 个

字段宽度的指标:

字符型字段	≤254 字节
日期型字段	=8 个字节
逻辑型字段	=1 个字节
明细型字段	≤4096 个字节
数值型字段	19 个字节(含小数点)

文件操作的指标:

最多能打开 15 个各种类型的文件;同时可打开 10 个数据库文件;如果在数据库中有明细型字段,则该数据库文件按两个文件计算,一个为数据库本身,另一个为明细文件。

每个现行的数据库文件可以打开相应的七个索引文件。

每个现行的数据库文件可以打开一个屏幕格式文件。

数值精度：

数字型字段最大宽度为 19 位(含小数点位)，整数部分最大宽度最多为 16 位。否则，16 位以后的位数不准确。就是说 dBASE III 系统的数值精度为 16 位。

最大数为 $1 \times 10^{+308}$

最小正数为 1×10^{-307}

有关内存变量的指标：

当前内存变量个数 ≤ 256 个

内存变量占用的总字节数 ≤ 6000

其它重要指标：

快速的多重字段排序；

对索引过的文件可进行快速的查找；

较完整的应用程序开发语言，具有过程调用，参数传输及其它灵活的特性。

四、dBASE III 的运行

1. 运行 dBASE III 的准备工作

初次使用 dBASE III 时，要检查 dBASE III 的系统盘上是否有 CONFIG.SYS 文件，并用 TYPE 命令显示其内容是否满足要求。若无则按“一”中微机系统配置一段中介绍的方法建立或修改 CONFIG.SYS 文件。

2. 运行 dBASE III

(1) 启动 CCDOS 或 GWBIOS 汉字操作系统。

(2) 在操作系统提示符 A>或 C>下键入命令：

A>dBASE 

dBASE III 调入内存并在屏幕上显示相应的提示信息之后出现提示符：

• ●

在圆点提示符之后有光标闪耀，此时 dBASE III 已调入内存可

以接受 dBASE III 的命令了。

比如：

- CREAT ✓ 建立库命令
- APPEND ✓ 添加记录命令
- LIST ✓ 显示记录命令
- SUM ✓ 求和命令

3. 退出 dBASE III

在 dBASE III 的圆点提示符下键入命令

- QUIT ✓

便可退出 dBASE III，回到操作系统提示符下：

A>

1.2.2 文件类型及有关约定

一、文件

我们向数据库输入的各种数据，最后都是以各种磁盘文件的形式存放于计算机系统中，在 dBASE III 中每建立一种关系，便形成一个文件。我们与数据库打交道，总离不开文件。

dBASE III 对文件的管理是在操作系统 DOS 支持下进行的，因此它的文件名的规定与 DOS 的规定一致，不再重复。

二、dBASE III 中的文件类型

dBASE III 中使用九种类型的文件：

1. 数据库文件(.DBF)

该文件是用户建立的数据库主要文件。其内容包括两部分：一部分是所建数据库的结构，包括有多少字段，什么名字，其宽度及类型规定等。另一部分是数据库的具体内容，一个记录相当于一行，一个字段相当于一列，形成一个二维表。

一个数据库文件最多可包括 1×10^9 个记录，每个记录最多可包括 4000 个字节，这些字节必须被包含在不多于 128 个字段中。

2. 数据库记忆(明细)字段文件(.DBT)

这是数据库的一个辅助文件,也称为数据库文本文件。用于存贮记忆型字段(MEMO)的内容。当用户建立的数据库文件有记忆型字段时,系统便自动生成一个同名的这类文件,其后缀部分为.DBT。记忆型字段可存贮 4000 个字符。

每个含有记忆型字段的数据库被打开时与其同名的记忆字段文件便同时被打开。如果用户把这个文件删掉了,则该数据库也不能被打开,必须建立同名记忆文件。

3. 索引文件(.NDX)

这是一个进行快速数据操作的辅助文件。它由(索引语句)INDEX 所产生。它是根据一个选定的关键字建立起来的,关键字是一个感兴趣的字段,或者是几个字段的某种组合,由于各记录均有与此关键字对应的项,便可根据这些项的值对各记录按逻辑顺序排列(而物理顺序是按输入记录的顺序排序)。

在 dBASE III 中,当对数据库的记录进行修改、删除、增加时,同时打开的索引文件也随之进行相应的修正。

4. 版式文件(.FMT)

此文件用于控制各种输出格式。由用户建立,由格式输出语句“@”及注释语句“*”组成。有了这个文件后,用户可以不使用系统提供的输入输出格式。使用中用 MODIFY COMMAND 命令建立和修改,用 SET FORMAT TO <文件名>命令识别,用 READ 命令启动。

5. 标签文件(.LBL)

可以根据数据的内容打印标签。例如,打印一个信封的内容,某种物品的品名及价格的标签等。打印的方法是先建立一个标签文件,然后根据此文件规定的内容和格式进行打印输出。它由 MODIFY LABEL 命令建立,用 LABEL 命令来使用。

6. 报表格式文件(.FRM)

该文件用来规定 dBASE III 产生报表时所应遵循的格式要求,包括标题、总计内容等信息,由 REPORT 命令产生,并可由文件编辑或字处理程序进行修改。

7. 内存变量文件(.MEM)

用户根据需要,可以把自己已经建立起来的内存变量的全部或某一部分,以文件的形式存在硬盘中形成一个内存变量文件。当需要这些变量时,还可以从磁盘中调到内存中来。用 SAVE 命令建立内存变量文件,用 RESTORE 命令从磁盘中调入内存。

8. 命令文件(.PRG)

dBASE III 的命令,提供了一套语言系统,可以进行编程。这种程序以一种文件形式存在,称为命令文件。这种文件是 ASCII 码文件,可以由系统提供的 MODIFY COMMAND 命令或 DOS 下的字处理程序来建立和编辑。当要执行这个文件时由 DO<文件名>命令来调用。

9. 文本输出文件(.TXT)

它主要作为 dBASE III 与其它软件之间的接口工具,是 ASCII 文件。通过一定的语句使 .DBF 文件和 .TXT 文件互相进行转换。
.TXT 文件由 APPEND FORM 命令指定的形式添加到 dBASE III 的 .DBF 文件中,COPY 命令的特定形式可将 .DBF 文件转换成 .TXT 文件。转换时若用“.”作为分界符,则转换后,将删去原字段中尾部的空白和数字字段中打头的空白,这样可以节省存储空间。

1.2.3 数据结构的约定

通过对数据库中各个字段的定义,可以建立数据库文件的结构框架。这是借助于命令 CREATE 和 MODIFY STRUCTURE 来完成的。字段的定义由下面三项组成:

字段名 字段名可达 10 个字符长。必须由一个字母打头,且不能含有嵌入的空格。字段名中允许出现字母、数字和下划线,也可使用汉字。

字段类型 用于 dBASE III 中的字段类型有五种:C——字符型;D——日期型;L——逻辑型;N——数字型;M——记忆型(或明细型)。

字段宽度 指字段中含有的字符或数字的最大个数。对 dBASE III 的五种字段类型简述如下：

字符型字段的最大宽度是 254 个字符。

日期型字段,用于存贮日期使用 PICTURE 函数可按多种形式键入和显示日期,缺省的格式是 mm/dd/yy,即月/日/年,日期字段的宽度总是 8。

数字型字段,有两种类型:整数型和小数型。数字字段的最大宽度是 19 位(含一位小数点),整数型数据的长度是 16 位,整数型数据中超过 16 位长度的部分是不可靠的。

逻辑型字段,只存放代表 True/False(真/假)值的一个单一字符。逻辑真为 T、t、Y 或 y4 个字符,逻辑假为 F、f、N 或 n4 个字符,宽度为 1 位。

记忆字段,是为了适应大块的文本信息而设计的。它们存放在数据库文件的辅助文件中,它在 .DBF 文件中用 Memo 表示。记忆型字段的大小是可变的,如果不键入数据,字段的大小为零。当键入数据时,字段空间每块指定为 512 字节,最大可为 4096 字节。在每个数据库文件(.DBF)记录中记忆字段占 10 个字节。

1.2.4 dBASE III 命令规范

dBASE III 的全部功能可通过调用提供给用户的 117 条命令来实现。这些命令大多数可以从键盘直接输入,使系统完成所规定的操作;也能够以命令文件形式输入,系统通过执行命令文件而得以执行它。也就是说,系统有两种工作模式。还有一小部分命令是专为程序模式提供的。dBASE 向用户提供一套相当于高级语言的程序语言系统。

一、dBASE III 命令结构的一般形式

一般形式为:

命令字[<范围>][<表达式>][FOR/WHILE<条件>]

说明:一般情况下命令结构分成四个部分:

1. 命令字 一个英文动词,dBASE III 提供的命令,它表达了命令要求执行的操作。当命令动词包含的字母超过 4 个时,第 5 个及其以后的字母可以省略。

2. 范围项 在范围中填入命令要求系统对其执行操作的记录范围。如果省略此项,不同的命令将有不同的含义,不省略时,一般有下列三种选择:

RECORD(n)——该操作仅施加于第 n 条记录。

NEXT(n)——该操作可施加于当前记录及其以后顺序的 n-1 个记录(共对 n 个记录进行操作)。

ALL——该操作施加于数据库的所有记录。

3. 表达式清单项 这一项中可列出一些表达式,中间以逗号“,”隔开。它表示命令所要涉及到的字段,也可以指出对这些字段进行何种操作。

4. 条件任选项 这一项可用“FOR”,也可用“WHILE”打头。“/”表示“或”的含义。然后是一个〈条件〉。一个条件就是一个表达式。这个表达式的返回值必须是逻辑型的。它表示只对那些能使这个条件为真(.T.)的记录进行命令所规定的操作。

当输入命令时,首先输入的是命令字,然后是任选项。但与上述三个任选项的顺序无关,可以任意排列。各部分之间和各英文单词之间(如 FOR 或 WHILE 后面)均应有空格。空格个数不限。但包括这些空格在内,整个命令的总长度不得超过 254 个字符。如果一个命令在一行显示不完时,系统就自动转入下一行,继续显示。

除了上述命令字外,命令中使用的其它由系统提供的英文文字(如 RECORD——记录,FIELDS——字段,STRUCTURE——结构,COMMAND——命令,REPORT——报告等),也可以自第五个字母开始省略。

二、命令结构举例

例如:显示命令

<u>DISP</u>	<u>NEXT 3</u>	<u>姓名,工资</u>	FOR	<u>工资>100</u>
命令字	范围	表达式		条件

此命令的四个部分是很明显的：第一部分“DISP”是“DISPLAY”命令的缩写，表示“显示”的意思；第二部分指明显示的范围，即当前记录以及后面的两个记录；第三部分说明要显示的字段仅有“姓名”和“工资”两项；最后一部分是条件，它指出，在规定的三个记录中，只显示工资字段的数值大于 100 的记录。

例：求平均命令

<u>AVERAGE</u>	<u>工资</u>	<u>ALL</u>	<u>TO AYG</u>
命令字	表达式	范围	结果存放

这个命令前三部分表示对所有记录的“工资”字段的数据求平均值，最后一部分表示将所求得的结果赋给内存变量 AYG。

三、命令书写规则

1. 每条命令必须以命令动词开头，动词后面的短语先后顺序是可以任意的，但必须符合规定的命令书写格式。

2. 应当用空格来分隔一条命令中的各个单词和短语，特别是两个关键字之间要用至少一个空格隔开。但应注意，T. 或 F. 两个逻辑值中，小圆点“.”和字母之间不容许有空格。

3. 一条命令的最大长度应少于 254 个字符。

4. 可以使用大写或小写字母。

5. dBASE ■ 没有保留字，但在给标识符命名时应该尽可能不使用 dBASE ■ 系统中的命令动词和关键字，以免引起不必要的混乱。

6. 当命令关键字很长时，可以缩写为前面四个字符。例如：STORE 可以写为 STOR.

1.2.5 表达式规则

表达式是 dBASE ■ 语句中的一个重要组成部分，不仅能反映出用户的各种要求，而且能反映出比字段自身操作更为复杂的运算关系，故比字段应用更普遍，它以各种形式出现。dBASE ■ 表达式可

以是字段、内存变量、常数、函数,也可以是用合法操作符将它们正确地连接起来的一个式子。表达式的类型取决于所含变量的类型,对于不同类型的变量能够进行的操作也不相同。

dBASE ■ 提供下述四种操作:

一、算术运算

这类运算主要对数字型变量进行操作,其运算结果也是一个数字型的量(日期型除外)。这类运算共包括六种操作:

1. 加法操作: + (此操作符在单目运算时表示“正”)
2. 减法操作: - (此操作符在单目运算时表示“负”)
3. 乘法操作: *
4. 除法操作: /
5. 指数操作: * * 或 ^
6. 括号操作: ()

二、关系运算

这种运算是对被操作的两个表达式进行比较,其结果必定是一个逻辑量。被操作的两个表达式可以是数字型的、字符型的,也可以是日期型的。对字符型表达式进行操作时,是比较它们的 ASCII 码值,此类运算有六种操作:

1. 小于操作: <
2. 大于操作: >
3. 等于操作: =
4. 不等于操作: <> 或 #
5. 小于等于操作: <=
6. 大于等于操作: >=

三、逻辑运算

这种运算是施加于一个或两个逻辑表达式的,得到的结果也是一个逻辑量。共四种操作:

1. 逻辑与操作: .AND.
2. 逻辑或操作: .OR.

3. 逻辑非操作：.NOT.

4. 括号操作：()

四、字符型运算

字符是指用双引号或单引号括起来的一个或一串字符,有以下常用运算:

1. 字符串连接运算符“+”和“-”

运算符“+”可将若干个字符串无条件地连接成一个字符串。

例如,下面的运算

“THIS”+“IS”+“A”+“BOOK”

其结果生成字符串“THISISABOOK”

运算符“-”也可以将若干个字符串连接起来,但是它要将每个字符串尾部的空格去掉后才进行连接而不管字符串前部是否有空格。

例如,下面的运算

“THIS”-“IS”-“A”-“BOOK”

其结果生成的字符串为“THISISA BOOK”

2. 字符串比较

dBASE III 命令中的条件,经常用到字符串比较,如何比较字符串的大小呢?

两个字符串比较是自左至右逐一比较字符的 ASCII 码值。前面相等比较后面的,直到有一个不相等或比较到最后仍相等为止。全部相等才说明字符串是相等的,否则为不相等。在机器内部每个字符有一个独立的 ASCII 码,编码值从低到高的顺序是:

〈空格〉! ” # \$ % & ' () * + , - . 0 1 2 3 4 5 6 7 8 9 ; : < = > ?

@ A—Z(大写)[\] ^ _ a—z(小写){|}

例如:

“ABC”<“ACB”

“ab”>“AB”

“aB”>“AB”

3. 子字符串匹配操作(\$)

这个运算是施加于两个字符串型表达式,但得到的结果是一个逻辑型的值。被操作的两个字符串表达式中,若\$后面的字符串包含有该操作符前面的字符串,则结果为.T.,若后面不包含有前面的,则结果为.F.。

例如:

“ABC”\$“NABCM”结果为(.T.)

“XYZ”\$“X=Y=Z”结果为(.F.)

五、表达式运算

所有上述操作在参加运算时,都须按一定的优先级别进行。

1. 综合表达式中的优先顺序:

(1) 算术运算

(2) 关系运算

(3) 逻辑运算

2. 算术运算中的优先顺序:

(1) 单目操作(+或-)

(2) 指数操作

(3) 乘法和除法操作

(4) 加法和减法操作

3. 逻辑运算中的优先顺序:

(1) 逻辑非运算

(2) 逻辑与运算

(3) 逻辑或运算

对于关系运算没有优先级差别。

在所有的各种操作中,括号操作是优先级别的最高者。同级操作自左至右顺序执行。

1.2.6 内存变量简介

内存变量用于暂时保存在程序运行中或命令执行所用的数据。

用户可任意定义自己的内存变量,内存变量为程序运行提供了更多方便。

内存变量的规定与字段变量的规定相同。

内存变量有四种类型:字符型、日期型、数字型和逻辑型。没有记忆型。内存变量的类型是在赋值过程中决定的。

dBASE III 最多允许定义 256 个内存变量,总的字节数不超过 6000 字节。

一般情形,在命令方式程序方式下均可以直接地使用内存变量名对其进行访问。但是,某一内存变量名有时可能与所使用的数据库中的字段变量名相同。为了防止混淆,在访问内存变量时,可以用下面的格式来使用内存变量。

M—>内存变量名

在程序方式下,内存变量还可分为全局型(PUBLIC)和局部型(PRIVATE)变量两种。

我们可以用 STORE 命令直接从键盘上定义内存变量。例如我们定一个内存变量 NAME。设它是一个字符型变量,可赋予它一个值“步兵”。则在命令方式下可输入下述命令:

• STORE “步兵” TO NAME↙

这时便定义了上述的内存变量,其宽度为 4,也可以由下面形式定义:

• NAME=“步兵”↙

为了查看某个内存变量的值可以使用命令“?”

例如:

• ? NAME↙

步兵

或

• ? M—>NAME↙

步兵

系统将 NAME 的值显示下一行上。

1.2.7 常用函数介绍

dBASE ■ 向用户提供了 37 种函数,以便对数据进行运算,有四种基本类型:数值型、字符型、逻辑型和日期型。函数一般出现在表达式中,也可以由“?”命令直接调用。这里介绍几种常用函数,其它函数将在以后各章节中介绍。

一、数据型数据运算函数

1. 自然指数函数 EXP

功能:该函数是求 e (≈ 2.71828) 的以〈数字表达式〉值为幂的数。

格式:

EXP(〈数字表达式〉)

例如:

• F=2

2

• ? EXP(F)

7.39

2. 取整函数 INT

功能:对〈数字表达式〉的值取整

格式:INT(〈数字表达式〉)

例如:

• STORE 3.14159 * 10 TO S

31.41590

• ? INT(S)

31

3. 自然对数函数 LOG

功能:求自然对数函数,也就是 EXP 函数的逆运算函数。

格式:LOG(〈数字表达式〉)

例如: