

第 5 章



逻辑与数值函数

在众多的数值类函数中,大多数函数的使用频率不高。使用频率较高的数值类函数主要有 `Number.From()`、`Number.FromText()`、`Number.Round()`、`Number.ToText()` 等几个。

5.1 逻辑函数

Power Query 的逻辑函数目前有 3 个: `Logical.From()`、`Logical.FromText()`、`Logical.ToText()`。

5.1.1 `Logical.From()`

`Logical.From()` 函数用于从某个值返回逻辑值,语法如下:

```
Logical.From(value as any) as nullable logical
```

说明:当给定值为 `null` 时,返回 `null`;当给定值为逻辑值、数值、文本型 "`true`" 或 "`false`" 时(不区分大小写),则返回值为 `true` 或 `false`。

应用举例,代码如下:

```
= Logical.From(null) // null

= Logical.From(Text.Length("2021")) //true

= Logical.From(2000 + 21) //true

= Logical.From(Logical.From(2021) and true) //true

= Logical.From("A" = "a" and 3 = 3) //false

= Logical.From("A" = "a" or 3 = 3) //true
```

```
= Logical.From("False")           // false
= Logical.From("True")             // true
```

"true" 或 "false" 之外的文本是不可以转换为逻辑值的。当逻辑值来自 "true" 或 "false" 之外的其他文本时,返回值会报错,代码如下:

```
Logical.From("A")
```

报错提示:

```
Expression.Error: 无法转换为 Logical。
详细信息:
    A
```

5.1.2 Logical.FromText()

Logical.FromText() 函数用于从文本值返回逻辑值 true 或 false,语法如下:

```
Logical.FromText(text as nullable text) as nullable logical
```

说明: 文本只能是文本型 "true" 或 "false" (不区分大小写),代码如下:

```
= Logical.FromText("False")       // false
= Logical.FromText("True")        // true
```

如果 text 为其他字符串,则会报错,代码如下:

```
= Logical.FromText("A")
```

报错提示:

```
Expression.Error: 无法转换为 Logical。
详细信息:
    A
```

5.1.3 Logical.ToText()

Logical.ToText() 函数用于从逻辑值返回文本值,语法如下:

```
Logical.ToText(logicalValue as nullable logical) as nullable text
```

应用举例,代码如下:

```
= Logical.ToText(true) //true
```

5.2 常用数值函数

5.2.1 判断

1. Number.IsNaN()

NaN 是 Not a Number 的英文简写,它是计算机语言中的数据类型之一,表示未定义或不可表示的值。例如,对 NaN 值的四则运算;对正负无穷大值的四则运算;对负数的开偶次方等,这些运算都将返回 NaN 值,代码如下:

```
= Number.IsNaN(0/0) //true  
  
= Number.IsNaN(1/0) //false  
  
= Number.IsNaN(Number.NaN + Number.NaN) //true  
  
= Number.IsNaN(2021 + Number.NaN) //true
```

2. Number.IsEven()

Even 是能被 2 整除的偶数,例如,个位数为 0、2、4、6、8 的自然数都是偶数。IsEven 表示用以判断它是不是偶数,返回的值为 true 或 false,代码如下:

```
= Number.IsEven(2021) //false  
  
= Number.IsEven(2022) //true
```

3. Number.IsOdd()

Odd 是不能被 2 整除的奇数,例如,个位数为 1、3、5、7、9 的自然数都是奇数。IsOdd 表示用以判断它是不是奇数,返回的值为 true 或 false,代码如下:

```
= Number.IsOdd(2021) //true  
  
= Number.IsOdd(2022) //false
```

4. Number.Sign()

Sign 中文是“符号”的意思。在很多计算机语言中,Sign 函数常用来表示“数学符号”函数。如果数值为正数,则返回 1;如果数值为负数,则返回 -1;如果数值为 0,则返回 0;如果值为 null,则返回 null,代码如下:

```
= Number.Sign(-2021)           //-1  
= Number.Sign(2021)            //1  
= Number.Sign(0)               //0  
= Number.Sign(null)            //null
```

5.2.2 随机数

1. Number.Random()

Random 中文是“随机”的意思。在大多数计算机语言中,Random 函数常用来生成“介于 0~1 的随机数”,代码如下:

```
= Number.Random()              //0.39867571527076684
```

2. Number.RandomBetween()

Between 中文是“介于”的意思,意味着其有“起、止”数据范围;Number.RandBetween() 函数用于返回介于“起、止”范围间的随机数,代码如下:

```
= Number.RandomBetween(6, 12)   //9.53216271732569
```

注意: Excel 的 Randbetween() 函数生成的是随机整数,它与 Number.RandBetween() 函数是有区别的。

5.2.3 计算

1. Number.Abs()

Abs 是 Absolute 的英文简写,中文是“绝对”的意思,Number.Abs() 函数返回的是数据的绝对值,代码如下:

```
= Number.Abs(-2021)            //2021
```

2. Number.IntegerDivide()

Integer 是“整数”的意思,Divide 在数学中表示的是“除、除以”的意思,所以 IntegerDivide 代表的是“整除”。Number.IntegerDivide() 函数共有三个参数:第一个参数为被除数,第二个参数为除数,第三个参数为可选参数。如果第一个参数或第二个参数为 null,则函数返回的值为 null,代码如下:

```
= Number.IntegerDivide(8, 4)    //2
```

```
= Number.IntegerDivide(9, 4)           //2  
= Number.IntegerDivide(9.3, 4)         //2  
= Number.IntegerDivide(9.3, null)      //null
```

3. Number.Mod()

Number.Mod()函数用于返回整除后的余数,代码如下:

```
= Number.Mod(8, 4)                     //0  
= Number.Mod(9, 4)                     //1  
= Number.Mod(9.3, 4)                   //1.3000000000000007  
= Number.Mod(9.3, null)                //null
```

4. Number.Power()

Power 在数学中代表的是“幂、乘方”的意思。Number.Power()函数共两个参数,第一个参数为基数,第二个参数为指数,代码如下:

```
= Number.Power(2,4)                    //16  
= Number.Power(2.05,4)                 //17.661006249999993
```

5. Number.Sqrt()

Sqrt 是 Square Root 的英文简写,是“平方根”的意思。Number.Sqrt()函数只有一个参数,代码如下:

```
= Number.Sqrt(25)                      //5  
= Number.Sqrt(36)                      //6
```

5.2.4 舍入

1. Number.Round()

在计算机语言中,Round()函数多用于数值的四舍五入。Number.Round()函数有三个参数。第一个参数为必选参数,当第一个参数为 null 时,返回的值为 null;第二个和第三个参数为可选参数。第二个参数为小数点位数的指定,当小数点位数未指定时,返回的值为整数;第三个参数为 RoundingMode(舍入方向的指定),代码如下:

```

= Number.Round(1.23) //1
= Number.Round(1.2345,2) //1.23
= Number.Round(Number.Power(2.05,4),2) //17.66,函数嵌套
= Number.Round(null) //null

```

以下是第三个参数 RoundingMode 的参数值说明：

1) RoundingMode.Up

参数值 RoundingMode.Up 表示向上舍入,也可以用代码 0 表示,代码如下:

```

= Number.Round(1.2345,3,0) //1.235
= Number.Round(1.2345,3,RoundingMode.Up) //1.235
= Number.Round(-1.2345,3,0) //-1.234

```

2) RoundingMode.Down

参数值 RoundingMode.Down 表示向下舍入,也可以用代码 1 表示,代码如下:

```

= Number.Round(1.2345,3,1) //1.234
= Number.Round(1.2345,3,RoundingMode.Down) //1.234
= Number.Round(-1.2345,3,1) //-1.235

```

3) RoundingMode.AwayFromZero

参数值 RoundingMode.AwayFromZero 表示向远离 0 的方向舍入,也可以用代码 2 表示,代码如下:

```

= Number.Round(1.2345,3,2) //1.235
= Number.Round(1.2345,3,RoundingMode.AwayFromZero) //1.235
= Number.Round(-1.2345,3,2) //-1.235

```

4) RoundingMode.TowardZero

参数值 RoundingMode.TowardZero 表示向靠近 0 的方向舍入,也可以用代码 3 表示,代码如下:

```

= Number.Round(1.2345,3,3) //1.234

```

```
= Number.Round(1.2345,3,RoundingMode.TowardZero) //1.234  
= Number.Round(-1.2345,3,3) // -1.234
```

2. Number.RoundUp()

Number.RoundUp()函数为向上舍入函数,共两个参数。如果第一个参数为 null,则返回的值为 null;第二个参数为可选参数,用于指定小数点的位数,代码如下:

```
= Number.RoundUp(1.2345,3) //1.235  
= Number.RoundUp(-1.2345,3) //-1.234
```

Number.RoundDown()函数等效于 Number.Round()函数,有使用第三个参数 RoundingMode.Down 的情形,代码如下:

```
= Number.Round(1.2345,3,0) //1.235, 第三个参数的 0 代表 RoundingMode.Up  
= Number.Round(-1.2345,3,0) //-1.234
```

3. Number.RoundDown()

Number.RoundDown()函数为向下舍入函数,共两个参数。如果第一个参数为 null,则返回的值为 null;第二个参数为可选参数,用于指定小数点的位数,代码如下:

```
= Number.RoundDown(1.2345,3) //1.234  
= Number.RoundDown(-1.2345,3) //-1.235
```

Number.RoundDown()函数等效于 Number.Round()函数,有使用第三个参数 RoundingMode.Down 的情形,代码如下:

```
= Number.Round(1.2345,3,1) //第三个参数的 1 代表 RoundingMode.Down  
= Number.Round(-1.2345,3,1)
```

4. Number.RoundAwayFromZero()

Number.RoundAwayFromZero()函数为向远离 0 的方向舍入,“向上舍入正数、向下舍入负数”。当值 ≥ 0 时,返回 Number.RoundUp(value);当值 < 0 时,返回 Number.RoundDown(value)。该函数共有两个参数。如果第一个参数为 null,则返回的值为 null;第二个参数为可选参数,用于指定小数点的位数,代码如下:

```
= Number.RoundAwayFromZero(1.2345,3)    //1.235

= Number.RoundAwayFromZero(-1.2345,3)    //-1.235
```

该函数等效于 Number.Round() 函数,有使用第三个参数 RoundingMode.AwayFromZero 的情形。

5. Number.RoundTowardZero()

Number.RoundTowardZero() 函数为向靠近 0 的方向舍入。该函数共有两个参数。如果第一个参数为 null,则返回的值为 null; 第二个参数为可选参数,用于指定小数点的位数,代码如下:

```
= Number.RoundTowardZero(1.2345,3)        //1.234

= Number.RoundTowardZero(-1.2345,3)        //-1.234
```

该函数等效于 Number.Round() 函数,有使用第三个参数 RoundingMode.TowardZero 的情形。

5.3 不常用数值函数

5.3.1 常量

1. Number.E

e 是数学中自然对数的底数,是一个无限不循环小数。Number.E 的常量值约为 2.7182818284590451。

2. Number.PI

Pi 是数学中的圆周率(π),是一个无限不循环小数。Number.PI 的常量值约为 3.1415926535897931。

3. Number.Epsilon

Number.Epsilon 的常量值约为 4.94065645841247E-324。

4. Number.PositiveInfinity

正无穷大为大于 0 的有理数或无理数(符号为 $+\infty$),没有具体的数值,表示的是比任何一个数字都要大的值,它来自正数除以 0 的值(例如 1/0)。

5. Number.NegativeInfinity

负无穷大为小于 0 的有理数或无理数(符号为 $-\infty$),没有具体的数值,表示的是比任何一个数字都要小的值,它来自负数除以 0 的值(例如 -1/0)。

5.3.2 计算

在数学中:ln 是以 e 为底的自然对数;lg 是以 10 为底的常用对数;log 是以任意数为

底的一般对数,可以以任何大于 0 且不等于 1 的数为底; $\log_{10}=\lg_{10},\ln=\log_e$ 。

在 M 语言中,与对数相关的函数有 Number.Ln()、Number.Exp()、Number.Log()、Number.Log10()。应用与示例如下:

```
= Number.Log(10) = Number.Ln(10) //true

= Number.Log(Number.E) = Number.Ln(Number.E) //true
```

5.3.3 三角函数

正弦(Number.Sin())、余弦(Number.Cos())、正切(Number.Tan())等是最常用的三角函数。其他与正弦定理相关的扩展函数有 Number.Asin()、Number.Sinh()、Number; 与余弦定理相关的扩展函数有 Number.Acos()、Number.Cosh(); 与正弦定理相关的扩展函数有 Number.Atan()、Number.Atan2()、Number.Tanh()。

在日常工作与生活中,在依据起止地的经纬度而计算出两地的距离时需要用到三角函数。如图 5-1 所示,依据“发货地经度、发货地纬度、收货地经度、收货地纬度”计算两地的距离。

	A	B	C	D	E
1	发货地经度	发货地纬度	收货地经度	收货地纬度	两地距离(千米)
2	93.341639	44.246048	99.82576	39.383646	761.8
3	104.143809	30.800075	116.666784	23.391161	1486.7
4	105.946251	27.1679	101.853877	27.273446	404.8
5	108.658567	19.101105	114.340055	37.750333	2145.6
6	111.776976	30.480651	112.239019	30.401957	45.2

图 5-1 计算两地的距离

在 Excel 工作表的 E2 单元格中,依据经纬度测算两地直线距离(单位:千米)的公式如下(为了便于阅读与理解,公式已被格式化):

```
E2 = ROUND(
    6371 *
    ACOS(1 -
        (
            POWER(
                (
                    SIN((90 - B2) * PI()/180) * COS(A2 * PI()/180) -
                    SIN((90 - D2) * PI()/180) * COS(C2 * PI()/180)
                ),
                2)
            +
            POWER(
                (
                    SIN((90 - B2) * PI()/180) * SIN(A2 * PI()/180) -
                    SIN((90 - D2) * PI()/180) * SIN(C2 * PI()/180)
```

```
),
2)
+
POWER(
(
COS((90 - B2) * PI()/180) - COS((90 - D2) * PI()/180)
),
2)
)/2
),
1
)
```

在 Excel 中,通过“数据”→“从表格”将数据(“表 1”)导入 Power Query,如图 5-2 所示。

	A	B	C	D
1	发货地经度	发货地纬度	收货地经度	收货地纬度
2	93.341639	44.246048	99.82576	39.383646
3	104.143809	30.800075	116.666784	23.391161
4	105.946251	27.1679	101.853877	27.273446
5	108.658567	19.101105	114.340055	37.750333
6	111.776976	30.480651	112.239019	30.401957

图 5-2 数据源

在 Power Query 编辑器中,选择“添加列”→“自定义列”。在“自定义列”对话框中输入计算公式,在高级编辑器中查看到的完整代码如下:

```
//ch503 - 022
let
    源 = Excel.CurrentWorkbook(){[Name = "表 1"]}[Content],

    两地距千米 = Table.AddColumn(
        源,
        "两地距离", each
        Number.Round(
            6371 * //地球的半径为 6371 千米
            Number.Acos(1 -
                (
                    Number.Power(
                        (
                            Number.Sin((90 - [发货地纬度]) * Number.PI/180) *
                            Number.Cos([发货地经度] * Number.PI/180)
                        -
                            Number.Sin((90 - [收货地纬度]) * Number.PI/180) *
                            Number.Cos([收货地经度] * Number.PI/180)
                        ),
                    2)
                ),
            2)
    )
```

```

+
Number.Power(
(
Number.Sin((90 - [发货地纬度]) * Number.PI/180) *
Number.Sin([发货地经度] * Number.PI/180)
-
Number.Sin((90 - [收货地纬度]) * Number.PI/180) *
Number.Sin([收货地经度] * Number.PI/180)
),
2)
+
Number.Power(
(
Number.Cos((90 - [发货地纬度]) *
Number.PI/180) - Number.Cos((90 - [收货地纬度]
)
* Number.PI/180)
),
2)
)/2
),
1
)
)
in
两地距千米
```

两地距离的单位为千米,返回值如图 5-3 所示。

	A	B	C	D	E
1	发货地经度	发货地纬度	收货地经度	收货地纬度	两地距离
2	93.341639	44.246048	99.82576	39.383646	761.8
3	104.143809	30.800075	116.666784	23.391161	1486.7
4	105.946251	27.1679	101.853877	27.273446	404.8
5	108.658567	19.101105	114.340055	37.750333	2145.6
6	111.776976	30.480651	112.239019	30.401957	45.2

图 5-3 两地距离

5.4 数值转换函数

5.4.1 Number.From()

Number.From()函数用于从某个值返回一个数值。该函数共有两个参数,第二个参数为可选参数,用于区域指定,语法如下:

```
Number.From(  
    value as any,  
    //第一个参数可为逻辑值、文本数值、日期、日期时间、日期时区时间、时间、持续时间  
    optional culture as nullable text  
) as nullable number
```

该函数的用法较为简单,例如 Number.From("12")返回的值为 12。

5.4.2 Number.FromText()

Number.FromText()函数用于从文本值返回一个数值。第一个参数只能为文本数值,不可以为“逻辑值、日期、日期时间、日期时区时间、时间、持续时间”。

5.4.3 Number.ToText()

Number.ToText()函数用于根据 format 指定的格式,将数值 number 格式化为文本值。该函数共有三个参数,其中第二个参数和第三个参数为可选参数。第二个参数的格式是单个字符代码,后面可能带一个数字精度说明符,第三个参数是区域语言选项,语法如下:

```
Number.ToText(  
    number as nullable number,  
    optional format as nullable text,  
    optional culture as nullable text  
) as nullable text
```

第二个参数的格式代码及用途如表 5-1 所示。

表 5-1 格式代码与用途说明

参 数 代 码	用 途
"D"或"d"	(十进制) 将结果格式化为整数。精度说明符控制输出中的位数
"E"或"e"	(指数/科学) 指数表示法。精度说明符控制最大小数的位数(默认值为 6)
"F"或"f"	(固定点) 整数和小数位
"G"或"g"	(常规) 固定点或科学记数法的最简洁形式
"N"或"n"	(数字) 带组分隔符和小数分隔符的整数和小数位
"P"或"p"	(百分比) 乘以 100 并显示百分号的数字
"R"或"r"	(往返) 可往返转换同一数字的文本值。将忽略精度说明符
"X"或"x"	(十六进制) 十六进制文本值

1. 参数为"D"

第二个参数以代码 D 为例,将结果格式化为整数,代码如下:

```
= Number.ToText(10, "D") //10
```

第一个参数必须为整数数值,如果值为文本型数值,则会报错,代码如下:

```
= Number.ToText("10","d")
```

错误提示如下:

Expression.Error: 无法将值 "10" 转换为类型 Number。

详细信息:

```
Value = 10  
Type = [Type]
```

如果第一个参数为带小数点的数值,则将返回错误提示,代码如下:

```
= Number.ToText(10.6, "D")
```

错误提示如下:

Expression.Error: 应为整数值。

详细信息:

```
10.6
```

2. 参数为"f"

```
= Number.ToText(12.345, "f1") //12.3
```

```
= Number.ToText(12.345, "f2") //12.35
```

3. 参数为"P"

```
= Number.ToText(0.12345, "P1") //12.3 %
```

```
= Number.ToText(0.12345, "P2") //12.35 %
```

4. 参数为"n"

```
= Number.ToText(12345.6789, "n") //12,345.68
```

```
= Number.ToText(12345.6789, "n1") //12,345.7
```

```
= Number.ToText(12345.6789, "n3") //12,345.679
```

5. 参数为"g"

```
= Number.ToText(12345.6789, "g") //12345.6789
```

```
= Number.ToText(12345.6789, "g1") //1e+04  
= Number.ToText(12345.6789, "g2") //1.2e+04
```

6. 手动指定格式

```
= Number.ToText(12345.6789, "0.0") //12345.7,一位小数  
= Number.ToText(12345.6789, "0.0 %") //1234567.9 %,一位小数的百分比  
= Number.ToText(12345.6789, "0000.00")  
//12345.68,数位补全(如整数不足4位,则用0补充显示)
```