dnc.group

IMERSÃO DE PROJETOS

LEAN + SCRUM + EXCEL + POWER BI

SE TRANSFORME EM UM PROFISSIONAL DESEJADO POR CONSULTORIAS, EMPRESAS MULTINACIONAIS E STARTUPS!

Expressões, valores e expressão Let

Uma consulta de linguagem de fórmula Power Query M é composta pelas etapas da **expressão** da fórmula que criam uma consulta de mashup. Uma expressão de fórmula pode ser avaliada (computada), produzindo um valor. A expressão **let** encapsula um conjunto de valores a serem computados, nomes atribuídos, então usados em uma expressão subsequente que segue a instrução **in**. Por exemplo, uma expressão let pode conter uma variável **Fonte** igual ao valor de **Text.Proper()** e produz um valor de texto com o uso correto de maiúsculas e minúsculas.

Expressão Let

powerquery-mCopiar

```
let

Source = Text.Proper("hello world")

in

Source
```

No exemplo acima, Text.Proper("olá, mundo") é avaliado como "Olá, Mundo".

As seções a seguir descrevem os tipos de valor no idioma.

Valor primitivo

Um valor **primitivo** é um valor de parte única, como um número, lógico, texto ou nulo. Um valor nulo pode ser usado para indicar a ausência de dados.

TABELA 1	
Tipo	**Valor de exemplo**
Binário	00 00 00 02 // número de pontos (2)
Data	23/5/2015
DateTime	5/23/2015 12:00:00 AM
DateTimeZone	5/23/2015 12:00:00 AM -08:00
Duração	15:35:00
Logical	true e false
Nulo	nulo
Número	0, 1, -1, 1,5 e 2.3e-5
Texto	"abc"
Hora	12:34:12 PM

Valor da função

Uma **Função** é um valor que, quando invocado com argumentos, produz um novo valor. As funções são gravadas pela listagem dos **parâmetros** da função entre parênteses, seguidos pelo símbolo de ir para =>, seguido pela expressão que define a função. Por exemplo, para criar uma função chamada "MyFunction" que tem dois parâmetros e executa um cálculo no parâmetro1 e parâmetro2:

powerquery-mCopiar

```
let

MyFunction = (parameter1, parameter2) => (parameter1 + parameter2) / 2

in

MyFunction
```

Calling the MyFunction() returns the result:

```
let

Source = MyFunction(2, 4)

in

Source
```

Esse código produz o valor de 3.

Valores de dados estruturados

A linguagem M dá suporte aos seguintes valores de dados estruturados:

- Lista
- Registro
- Tabela
- Exemplos de dados estruturados adicionais

Observação

Os dados estruturados podem conter qualquer valor de M. Para ver alguns exemplos, confira Exemplos de dados estruturados adicionais .

Lista

Uma lista é uma sequência ordenada com base em zero de valores entre chaves { }. As chave { } também são usadas para recuperar um item de uma lista por posição de índice. Confira [List value](#_List_value).

Observação

O Power Query M dá suporte a um tamanho de lista infinito, mas se uma lista for gravada como um literal, ela terá um comprimento fixo. Por exemplo, {1, 2, 3} tem um comprimento fixo de 3.

A seguir estão alguns exemplos de **Lista**.

TABELA 2	
Valor	Tipo
{123, true, "A"}	Lista que contém um número, um lógico e um texto.
{1, 2, 3}	Lista de números
{{1, 2, 3},{4, 5, 6}}	Lista da lista de números
{[CustomerID = 1, Name = "Bob", Phone = "123-4567"],[CustomerID = 2, Name = "Jim", Phone = "987-6543"]}	Lista de Registros
{123, true, "A"}{0}	Obtenha o valor do primeiro item em uma lista. Essa expressão retorna o valor 123.
{{1, 2, 3},{4, 5, 6}}{0}{1}	Obtenha o valor do segundo item do primeiro elemento List. Essa expressão retorna o valor 2.

Registro

Um **Registro** é um conjunto de campos. Um **campo** é um par de nome/valor em que o nome é um valor de texto exclusivo dentro do registro do campo. A sintaxe para valores de registro permite que os nomes sejam gravados sem aspas, uma forma também conhecida como **identificadores**. Um identificador pode ter as duas formas a seguir:

- identifier_name como OrderID.
- #"nome do identificador", como #"A data de hoje é: ".

A seguir está um registro que contém os campos chamados "OrderID", "CustomerID", "Item" e "Price" com os valores 1, 1, "Vara de pescar" e 100,00. Os caracteres de colchete [] denotam o início e o fim de uma expressão de registro e são usados para obter um valor de campo de um registro. Os exemplos a seguir mostram um registro e como obter o valor do campo item.

Aqui está um exemplo de registro:

```
powerquery-mCopiar

let Source =

[

OrderID = 1,

CustomerID = 1,

Item = "Fishing rod",

Price = 100.00

]

in Source
```

Para obter o valor de um Item, use colchetes como Source[item]:

```
powerquery-mCopiar

let Source =

[
```

```
OrderID = 1,

CustomerID = 1,

Item = "Fishing rod",

Price = 100.00

]

in Source[Item] //equals "Fishing rod"
```

Tabela

Uma **Tabela** é um conjunto de valores organizados em colunas e linhas nomeadas. O tipo de coluna pode ser implícito ou explícito. Você pode usar #table para criar uma lista de nomes de coluna e lista de linhas. Uma **Tabela** de valores é uma Lista em uma **Lista**. Os caracteres de chave {} também são usados para recuperar uma linha de uma **Tabela** por posição de índice (confira Exemplo 3 – Obter uma linha de uma tabela pela posição de índice).

Exemplo 1 – Criar uma tabela com tipos de coluna implícitos

```
powerquery-mCopiar

let

Source = #table(

{"OrderID", "CustomerID", "Item", "Price"},

{

{1, 1, "Fishing rod", 100.00},

{2, 1, "1 lb. worms", 5.00}

})

in

Source
```

Exemplo 2 – Criar uma tabela com tipos de coluna explícitos

powerquery-mCopiar

```
let

Source = #table(

type table [OrderID = number, CustomerID = number, Item = text, Price =
number],

{

{1, 1, "Fishing rod", 100.00},

{2, 1, "1 lb. worms", 5.00}

}

)

in

Source
```

Os dois exemplos acima criam uma tabela com a seguinte forma:

TABELA 3			
OrderID	CustomerID	Item	Preço
1	1	Vara de pescar	100,00
2	1	1 lb de minhocas	5,00

Exemplo 3 – Obter uma linha de uma tabela por posição de índice

powerquery-mCopiar

```
let

Source = #table(

type table [OrderID = number, CustomerID = number, Item = text, Price =
number],

{

{1, 1, "Fishing rod", 100.00},

{2, 1, "1 lb. worms", 5.00}

}

)

in

Source{1}
```

Essa expressão retorna o registro a seguir:

TABELA 4	
OrderID	2
CustomerID	1
Item	1 lb de minhocas
Preço	5

Exemplos de dados estruturados adicionais

Os dados estruturados podem conter qualquer valor de M. Aqui estão alguns exemplos:

Exemplo 1 – Lista com valores [Primitive](#_Primitive_value_1), [Function](#_Function_value) e [Record](#_Record_value)

powerquery-mCopiar

```
let

Source =

{

1,

"Bob",

DateTime.ToText(DateTime.LocalNow(), "yyyy-MM-dd"),

[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0]

}

in

Source
```

Avaliar essa expressão pode ser visualizado como:

A List containing a Record	
1	
"Bob"	
2015-05-22	
OrderID	1
CustomerID	1
Item	"Fishing rod"
Price	100.0

Exemplo 2 – Registro que contém valores Primitivos e Registros aninhados

powerquery-mCopiar

```
let

Source = [CustomerID = 1, Name = "Bob", Phone = "123-4567", Orders =

{

[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0],

[OrderID = 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0]

}]

in

Source
```

Avaliar essa expressão pode ser visualizado como:

A record containing a List of Records		
CustomerID	1	
Name	"Bob"	
Phone	"123-4567"	
Orders	OrderID	1
	CustomerID	1
	Item	"Fishing rod"
	Price	100.0
	OrderID	2
	CustomerID	1
	Item	"1 lb. worms"
	Price	5.0

Observação

Embora muitos valores possam ser escritos literalmente como uma expressão, um valor não é uma expressão. Por exemplo, a expressão 1 é avaliada como o valor 1; a expressão 1+1 é avaliada como o valor 2. Essa distinção é sutil, mas importante. As expressões são receitas para avaliação; os valores são os resultados da avaliação.

Expressão If

A expressão **if** seleciona entre duas expressões com base em uma condição lógica. Por exemplo:

powerquery-mCopiar

```
if 2 > 1 then

2 + 2

else

1 + 1
```

A primeira expressão (2 + 2) será selecionada se a expressão lógica (2 > 1) for verdadeira e a segunda expressão (1 + 1) será selecionada se for false. A expressão selecionada (nesse caso, 2 + 2) é avaliada e torna-se o resultado da expressão **if** (4).