German Ricardo Bermúdez Navarro

Edwin Reinel Perdomo Sedano
Brian David Lozano Guarín
Diego alexander Beltrán Hernández

Actividad #2 consultas lenguajes

Girardot-Cundinamarca Marzo 2023
Ingeniería de sistemas
Minería de datos

El lenguaje M

El lenguaje M es básicamente el idioma de programación con el que trabaja el editor de consultas Microsoft Power Query. Cada aplicación tiene su propio lenguaje de programación, seguro que os suena alguno de los lenguajes de programación más conocidos, como por ejemplo JavaScript o C++. Pues bien, el lenguaje M es el lenguaje que utiliza Power Query para poder conectar, combinar y refinar los orígenes de datos y así dejar una base de datos limpia que permita extraer y analizar la información que se desee.

No es necesario entender el lenguaje M ni saber programar con él. La principal ventaja de Power Query es que nos permite buscar orígenes de datos, hacer conexiones entre ellos y limpiar columnas de datos mediante una interfaz sencilla que te indica todos los pasos que está haciendo, para que finalmente tengas tu base de datos lista para extraerle toda la información. Todo esto sin movernos de nuestra hoja de Excel o de nuestro panel de Power BI.

Microsoft Power Query proporciona una potente experiencia de "obtención de datos" que abarca muchas características. Una de las funciones básicas de Power Query es filtrar y combinar, es decir, "mezclar" los datos de una o varias colecciones enriquecidas de orígenes de datos admitidos. Cualquier mashup de datos se expresa mediante el lenguaje de fórmulas de Power Query (conocido informalmente como "M"). Para habilitar el mashup repetible de datos, Power Query inserta documentos M en una amplia gama de productos de Microsoft, incluidos Excel, Power BI, Analysis Services, Dataverse.

Se describe el lenguaje de forma precisa en varios pasos progresivos:

1. La estructura léxica define el conjunto de textos que son léxicamente válidos.

2. Los valores, las expresiones, los entornos y variables, los identificadores y el modelo de evaluación conforman los conceptos básicos del lenguaje.

3. La especificación detallada de los valores, tanto primitivos como estructurados, define el dominio de destino del lenguaje.

4. Los valores tienen tipos que, a su vez, son un tipo especial de valor, que caracterizan los tipos fundamentales de valores y contienen metadatos adicionales específicos de las formas de valores estructurados.

5. El conjunto de operadores de M define qué tipos de expresiones se pueden formar.

6. Las funciones, otro tipo de valores especiales, proporcionan la base para una biblioteca estándar enriquecida para M y permiten agregar abstracciones nuevas.

7. Al aplicar operadores o funciones durante la evaluación de expresiones, se pueden producir errores. Aunque los errores no son valores, hay formas de controlar los errores que permiten volver a asignarlos a valores.

8. Las expresiones let permiten la introducción de definiciones auxiliares que se usan para crear expresiones complejas en pasos más pequeños.

9. Las expresiones if admiten la evaluación condicional.

10. Las secciones proporcionan un mecanismo de modularidad simple. (En Power Query todavía no se aprovechan las secciones).

11. Por último, en una gramática consolidada se recopilan los fragmentos de gramática de todas las demás secciones de este documento en una única definición completa.

Para los teóricos de los lenguajes de computación: el lenguaje de fórmulas que se especifica en este documento es un lenguaje funcional parcialmente diferido, principalmente puro, de orden superior y con tipos dinámicos.

Expresiones y valores

En M, la construcción central es la *expresión*. Una expresión se puede evaluar (calcular), lo que produce un único *valor*.

Aunque muchos valores se pueden escribir literalmente como una expresión, un valor no es una expresión. Por ejemplo, la expresión 1 se evalúa como el valor *1*, mientras que 1+1 se evalúa como el valor *2*. Esta distinción es sutil, pero importante. Las expresiones son las recetas de evaluación y los valores, los resultados de esa evaluación.

En los siguientes ejemplos se muestran los diferentes tipos de valores disponibles en M. Como convención, un valor se escribe con la forma literal en la que aparecería en una expresión que se evalúa solamente como ese valor. (Tenga en cuenta que // indica el inicio de un comentario que continúa hasta el final de la línea).

Un valor *primitivo* es un valor de una sola parte, como un número, un valor lógico, texto o un valor null. Se puede usar un valor null para indicar la ausencia de datos.

```
123     // A number

true    // A logical

"abc"   // A text

null    // null value
```

Un valor de *lista* es una secuencia ordenada de valores. M admite listas infinitas, pero si se escriben como un literal, las listas tienen una longitud fija. Los caracteres de llave { y } indican el principio y el final de una lista.

```
{123, true, "A"}   // list containing a number, a logical, and
                   //     a text
{1, 2, 3}          // list of three numbers
```

Un *registro* es un conjunto de *campos*. Un campo es un par nombre-valor en el que el nombre es un valor de texto único dentro del registro del campo. La sintaxis literal de los valores de registro permite escribir los nombres sin comillas, formato que también se conoce como *identificadores*. El registro siguiente contiene tres campos denominados "A", "B" y "C", que tienen los valores 1, 2 y 3.

```
[
    A = 1,
    B = 2,
    C = 3
]
```

Una *tabla* es un conjunto de valores organizados en columnas (que se identifican por nombre) y filas. No hay ninguna sintaxis literal para crear una tabla, pero se pueden usar varias funciones estándar para crear tablas a partir de listas o registros.

```
#table( {"A", "B"}, { {1, 2}, {3, 4} } )
```

Esto crea una tabla con la forma siguiente:

A	B
1	2
3	4

Una *función* es un valor que, cuando se invoca con argumentos, genera un nuevo valor. Para escribir funciones, se enumeran los *parámetros* de la función en entre paréntesis, seguidos del signo igual =>, seguido de la expresión que define la función. Esa expresión normalmente hace referencia a los parámetros (por nombre).

```
(x, y) => (x + y) / 2`
```

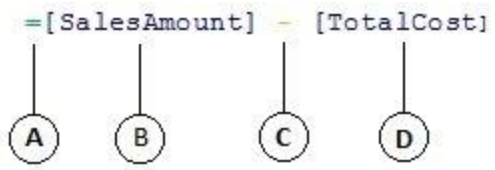
Lenguaje DAX:

DAX (Data Analysis Expressions) es un lenguaje específico para análisis de datos creado por Microsoft en el año 2010 para ser usado con un modelo de datos tabular y que se puede usar en Excel, Analysis Services y Power BI.

DAX es una recopilación de funciones, operadores y constantes que se pueden usar en una fórmula o expresión para calcular y devolver uno o varios valores. Dicho más fácilmente, DAX ayuda a crear información de datos nueva que ya está en un modelo. Las fórmulas de DAX proporcionan esta capacidad y muchas otras funciones importantes. Aprender a crear fórmulas DAX eficaces le ayudará a sacar el máximo partido de los datos. Cuando obtiene la información que necesita, puede empezar a solucionar los problemas empresariales reales que afectan a la base. Se trata de Business Intelligence y DAX ayuda entrar y comprender los conceptos relacionados aplicados.

Sintaxis:

Antes de crear sus propias fórmulas, echemos un vistazo a la sintaxis de las fórmulas de DAX. La sintaxis incluye varios elementos que conforman una fórmula o, más simplemente, cómo se escribe. Por ejemplo, examinemos una fórmula DAX simple utilizada para crear datos nuevos (valores) para cada fila de una columna calculada, denominada Margin, en una tabla FactSales: (los colores del texto de la fórmula únicamente son para fines ilustrativos)



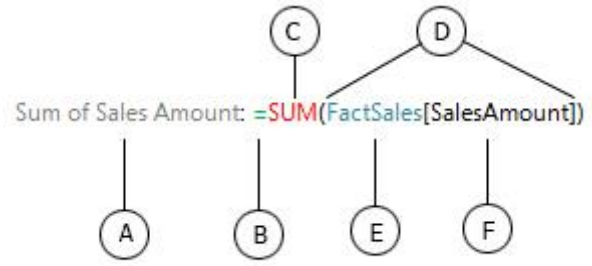
La sintaxis de esta fórmula incluye los elementos siguientes:

1. El operador del signo igual (=) indica el principio de la fórmula y cuando esta fórmula se calcule, devolverá un resultado o un valor. Todas las fórmulas que calculan un valor empezarán con un signo de igual.
2. La columna a la que se hace referencia [SalesAmount] contiene los valores de los que se resta. Una referencia de columna de una fórmula siempre viene entre corchetes []. A diferencia de las fórmulas de Excel que hacen referencia a una celda, una fórmula DAX hace referencia siempre a una columna.
3. El operador matemático de resta (-).
4. La columna a la que se hace referencia [TotalCost] contiene los valores que deseamos restar de los valores de la columna [SalesAmount].

Al intentar entender cómo leer una fórmula de DAX, suele ser útil analizar cada uno de los elementos de un idioma en el que piense y hable cada día. Por ejemplo, puede leer esta fórmula como:

En la tabla *FactSales,* para cada fila de la columna calculada Margen, calcule (=) un valor restando *(-)*valores de la columna [*TotalCost*] de los valores de la*columna [SalesAmount].*

Echemos un vistazo a otro tipo de fórmula, una que se usa en una medida:



Esta fórmula incluye los elementos de sintaxis siguientes:

1. El nombre de la medida Suma del importe de ventas. Las fórmulas para medidas pueden incluir el nombre de la medida, seguido de dos puntos, seguido de la fórmula de cálculo.
2. El operador del signo igual (=) indica el principio de la fórmula de cálculo. Cuando se calcule, devolverá el resultado.
3. La función SUM suma todos los números de la columna [SalesAmount]. Obtendrá más información sobre características más adelante.
4. Los paréntesis () alrededor de uno o más argumentos. Todas las funciones requieren al menos un argumento. Un argumento pasa un valor a una función.
5. La tabla a la que se hace referencia FactSales.
6. La columna a la que se referencia [SalesAmount] en la tabla FactSales. Con este argumento, la función SUM sabe qué columna agregar a SUM.

Bibliografía:

https://aglaia.es/blog/power-bi/lenguaje-m/

https://learn.microsoft.com/es-es/powerquery-m/

https://learn.microsoft.com/es-es/powerquery-m/m-spec-introduction

https://es.linkedin.com/learning/power-bi-avanzado/breve-introduccion-a-lenguaje-m-componentes-y-conceptos#:~:text=El%20lenguaje%20M%20est%C3%A1%20compuesto,M%20son%20expresiones%20y%20valores.

https://interactivechaos.com/es/manual/tutorial-de-power-bi/el-lenguaje-m

https://es.linkedin.com/learning/power-bi-esencial/introduccion-al-lenguaje-dax

https://aglaia.es/blog/power-bi/que-es-dax/

https://platzi.com/clases/2676-powerbi/44821-lenguaje-dax/

https://support.microsoft.com/es-es/office/tutorial-r%C3%A1pido-aprenda-los-fundamentos-de-dax-en-30-minutos-51744643-c2a5-436a-bdf6-c895762bec1a

https://www.dataxbi.com/blog/2018/11/13/introduccion-lenguaje-dax/

https://datascientest.com/es/saber-todo-dax-power-bi

https://verneacademy.com/blog/articulos-data/que-es-dax-power-bi/