

Función COUNT



AVERAGE se puede crear como medida implícita. Evitarlo



Int. Contexto

Contexto de Filtro

Tiene en cuenta el contexto de filtro

★★★★★

Contexto de Fila

Ignora el contexto de fila



Categorías

Según Proceso Interno

Cálculo directo

★★★★★

Según Resultado

Escalar

Descripción

La función COUNT cuenta los valores no BLANK de una columna en una tabla.

Sintaxis

1

Primer Parámetro

COUNT (< Nombre De Columna >)

1

Nombre de Columna

Columna que contiene los valores a contar.

Tipo

Obligatorio

Atributo

No Repetible



Valor Que Retorna



Retorna un valor de tipo INTEGER el cual es: el número de valores que no son BLANK arrojados del conteo de la columna. Donde BLANK para el lenguaje DAX es todo *valor vacío* y *valor blanco* que no sea producto de una expresión DAX.

- Valores Vacíos: Representa una cadena de caracteres vacía.
- Valores en Blanco: Representa valore o casillas con sólo espacios.

Es decir, lo que sería representado como *null* en Power Query.

Observaciones

1. Aquellos valores que se hereden al motor DAX de manera nativa como *ausencia absoluta de información*, representados en *Power Query* como una casilla completamente vacía y que no se puede operar dado que no tiene ningún tipo de definición, no son contados como BLANK de manera directa.

- Ejemplo:

Si una columna de datos compuesta por 18 filas tiene 12 valores *null*, 4 numéricos y 2 de *ausencia absoluta*, entonces la siguiente medida:

NOTA



Es imperativo notar que la descripción de Valor Que Retorna y la Observación 1 se basa en valores 100% nativos.

Dicho de otro modo, se basa en los valores recibidos por Power Query desde un origen de datos, puesto que, si el valor vacío o blanco es generado por una expresión DAX, estos se cuentan como una fórmula y no como BLANK, por lo que son tenidos en cuenta por la función COUNT.

1. `Conteo = -- Retorna 6.`
2. `COUNT (Datos[Datos])`

Retornaría 6 si es puesta en una tarjeta de Power BI, a pesar de que en la vista de datos se pueden visualizar sólo 4 valores y todo lo demás como casillas sin nada.

Para resolver este escenario podemos apoyarnos en el hecho de que cuando BLANK se involucra en una operación de comparación, este se propaga, por lo que aquí toca recurrir a la función COUNTX:

1. `ConteoConValoresAusentes = -- Retorna 4.`
2. `COUNTX (Datos ,`
3. `IF (Datos[Datos] = BLANK () ,`
4. `BLANK () , 1`
5. `)`
6. `)` – El motivo para el funcionamiento de esta expresión recae
7. – En que (ausencia) = BLANK arroja TRUE.

Caso diferente si se utiliza la función la función ISBLANK, ya que esta realiza una comparación de tipo: *estrictamente igual* (`=`), por lo que esta medida:

1. `ConteoConValoresAusentes = -- Retorna 6.`
2. `COUNTX (Datos ,`
3. `ISBLANK (Datos[Datos]) ,`
4. `BLANK () , 1`
5. `)`

También devolvería 6, a consecuencia de que es exactamente igual la siguiente expresión:

```

1. ConteoConValoresAusentes = -- Retorna 6.
2. COUNTX ( Datos,
3.     Datos[Datos] == BLANK () , -- El operador == significa
4.     BLANK () , 1 -- Estrictamente igual.
5. )

```

- I. La diferencia entre las funciones [COUNT](#) y [COUNTA](#) sólo se hace vivida cuando se trabaja con tipo de dato TRUE/FALSE, puesto que la función [COUNT](#) no puede trabajar con tipos booleanos.
- II. La función [COUNTA](#) retorna [BLANK](#) si ninguno de los valores a contar es no [BLANK](#).
- III. La función [COUNT](#) cuando se utiliza con una sola columna ejecuta de manera interna la función [COUNTX](#), sin ninguna diferencia en el rendimiento del modelo de datos.

```

1. =
2. COUNT ( Datos[Datos] )

```

Se corresponde con:

```

1. =
2. COUNTX ( Datos, Datos[Datos] )

```