

CASE

Obtener valores de campo basados en expresiones condicionales

Las instrucciones CASE usan la lógica condicional para determinar los valores de los campos calculados.

Este tipo de instrucciones suele utilizarse para crear categorías o grupos de datos. Por ejemplo, puede crear una instrucción CASE como la que se indica a continuación para agrupar valores de países seleccionados en una dimensión denominada *Región de ventas*:

```
CASE
  WHEN País IN ("EE. UU.", "Canadá", "México") THEN "Norteamérica"
  WHEN País IN ("Inglaterra", "Francia") THEN "Europa"
  ELSE "Otro"
END
```

Secciones de este artículo:

- [Sintaxis de las instrucciones CASE](#)
- [Condiciones WHEN](#)
- [Resultados THEN](#)
- [Resultados ELSE](#)
- [Operadores de comparación](#)
- [Operadores lógicos](#)
- [Funciones compatibles](#)
- [Recursos relacionados](#)

Crear una instrucción CASE

En una fuente de datos:

1. Edite la fuente de datos.
2. En la parte superior derecha, haga clic en **+ Añadir un campo**.

En un informe o exploración:

1. Edite el informe o la exploración.
2. Seleccione un gráfico.
3. A la derecha, en el panel de propiedades, haga clic en **+ Añadir dimensión** o **+ Añadir métrica**.
4. Haga clic en **+ CREAR CAMPO**.

A continuación:

1. Asigne un **nombre** al campo.
2. En el campo **Fórmula**, introduzca la instrucción CASE.

En una fuente de datos, haga clic en **DAR FORMATO A FÓRMULA** para diseñar automáticamente la fórmula y facilitar su lectura.

3. Haga clic en **GUARDAR** o en **APLICAR**.

Sintaxis de las instrucciones CASE

```
CASE
  WHEN condición THEN resultado
  [WHEN condición THEN resultado]
  [...]
  [ELSE resultado]
END
```

Las instrucciones CASE comienzan con la palabra clave CASE y terminan con la palabra clave END. Entre ambas palabras clave se incluyen varias secciones o "cláusulas":

- **WHEN:** una condición lógica que quiere evaluar. Puede incluir varias cláusulas WHEN en una sola instrucción CASE.
- **THEN:** el resultado que devuelve la expresión si se cumple la condición de la cláusula WHEN. La instrucción CASE debe contener una cláusula THEN por cada cláusula WHEN.
- **ELSE:** Esta cláusula es opcional. Si no se cumple ninguna de las condiciones de las cláusulas WHEN, CASE devuelve el valor de la cláusula ELSE, o **NULL** si no se ha especificado ninguna.

Condiciones WHEN

Las condiciones WHEN evalúan los datos y devuelven el valor "true" si se cumple la condición especificada o, si no se cumple, el valor "false". Las condiciones WHEN válidas pueden hacer las siguientes funciones:

- Comparar una dimensión o métrica con un valor literal utilizando un [operador](#) compatible.
- Aplicar una [coincidencia de expresiones regulares](#) a una dimensión o métrica.
- Determinar si el valor de una dimensión o métrica [es nulo \(is null\)](#).
- Determinar si una dimensión o métrica [contiene](#) un valor especificado en una lista.
- Determinar si una dimensión o métrica [no contiene](#) un valor especificado en una lista.

Límites de las condiciones WHEN

WHEN debe compararse con un valor literal o con una función compatible.

No se pueden comparar las dimensiones ni las métricas entre sí. En los ejemplos siguientes, las dos primeras instrucciones son condiciones WHEN válidas, mientras que las dos últimas NO son válidas:

```
CASE WHEN País = "Inglaterra" ... // fórmula válida
CASE WHEN Conversiones > 100 ... // fórmula válida
```

```
CASE WHEN Dimensión1 = Dimensión2 ... // fórmula no válida  
CASE WHEN Métrica1 > Métrica2 ... // fórmula no válida
```

Funciones compatibles

Las condiciones WHEN pueden incluir una o varias funciones de comparación compatibles. [Consulte los detalles más abajo](#)

Las condiciones WHEN no pueden mezclar dimensiones y métricas.

Por ejemplo, no se admite lo siguiente:

```
CASE WHEN País = "Inglaterra" AND Conversiones > 100 THEN ... // fórmula no válida
```

Las condiciones WHEN deben contener una dimensión o una métrica.

Las condiciones WHEN no pueden incluir únicamente un valor literal. Por ejemplo, no se admite lo siguiente:

```
CASE WHEN 1 THEN ... // fórmula no válida
```

Las condiciones WHEN no pueden hacer cálculos.

No es posible calcular ni usar funciones de campo calculado en las condiciones WHEN. Por ejemplo, no se admite lo siguiente:

```
CASE WHEN métrica1 + métrica2 > 4 THEN ...// fórmula no válida
```

Puede solucionar esta limitación creando antes un campo calculado con la fórmula que quiera usar. A continuación, utilice ese campo en la instrucción CASE. Por ejemplo:

1) Cree un campo calculado (*Mi campo*) con la siguiente fórmula:

```
métrica1 + métrica2
```

2) Utilice *Mi campo* en la instrucción CASE:

```
CASE WHEN Mi campo > 4 THEN ...
```

Resultados THEN

Las condiciones WHEN deben corresponderse con una cláusula WHEN, que especifica los resultados si la condición se cumple. Si hay varias condiciones WHEN, la instrucción CASE devuelve el resultado de la primera condición que se cumple.

Las cláusulas THEN pueden devolver los siguientes elementos:

- Dimensiones
- Métricas
- Campos calculados
- Valores literales
- **NULL**

Límites de los resultados THEN

Las cláusulas THEN no pueden devolver una expresión o una fórmula.

Por ejemplo, no se admite lo siguiente:

```
CASE WHEN País = "Inglaterra" THEN métrica1 + métrica2 END // fórmula no válida
```

Todos los posibles resultados de una instrucción CASE deben ser del mismo tipo.

Por ejemplo, si la primera cláusula THEN devuelve texto, las demás cláusulas THEN y la cláusula ELSE también deben devolver texto.

Resultados ELSE

La cláusula opcional ELSE especifica un resultado predeterminado, que se devuelve si no se cumple ninguna cláusula WHEN. Si una instrucción CASE no tiene ninguna cláusula ELSE y tampoco se cumple ninguna cláusula WHEN, la instrucción devuelve **NULL**. Las instrucciones CASE solo pueden contener una cláusula ELSE.

La cláusula ELSE puede devolver los siguientes elementos:

- Dimensiones
- Métricas
- Campos calculados
- Valores literales
- **NULL**

Más información sobre NULL

NULL no equivale a una cadena vacía (""). Si se intenta utilizar una cadena vacía en una operación aritmética, se suele producir un error; en cambio, si se usa NULL, se devuelve el valor nulo:

```
1+NULL = NULL
```

```
1*NULL = NULL
```

```
1/NULL = NULL
```

Etc.

Por otro lado, aunque puede concatenar valores con la cadena vacía, el resultado de concatenar cualquier valor con NULL es el valor nulo:

```
CONCAT("A", "", "B") = "AB"
```

```
CONCAT("A", NULL, "B") = null
```

Operadores de comparación

Puede incluir los siguientes operadores en las cláusulas WHEN:

- =
- !=
- >
- >=
- <
- <=

Los operandos que aparecen a ambos lados de los operadores de comparación deben ser del mismo tipo: de texto, numéricos o booleanos. Al menos uno de los operandos debe ser un campo, mientras que el otro puede ser de texto, numérico o booleano.

Ejemplos

```
CASE WHEN Medio != "CPC" THEN "gratis" ELSE "de pago" END
```

```
CASE WHEN Tiempo en la página <= 90 THEN 1 ELSE 0 END
```

Operadores lógicos

Puede añadir condiciones AND y OR a las condiciones WHEN. Utilice paréntesis para agrupar condiciones y definir el orden de precedencia.

Ejemplos

```
CASE WHEN Código ISO del país = "US" AND Medio = "CPC" THEN "US: de pago" ELSE "Otro" END
```

```
CASE WHEN REGEXP_MATCH(Título del vídeo, ".*Google Analytics*") AND  
is_livestream = TRUE OR Duración del vídeo > 120 THEN "GA LIVE or LONG" END
```

Funciones compatibles

Puede utilizar las siguientes funciones en las condiciones WHEN.

¿Por qué no puedo usar otras funciones en las instrucciones CASE?

Las condiciones WHEN deben ser expresiones booleanas. Excepto en el caso de REGEXP_MATCH, las funciones de campo calculado devuelven resultados no booleanos (es decir, números, texto, fechas o códigos geográficos).

IS NULL

Devuelve el valor "true" si el campo de entrada está vacío o es nulo.

Esta función solo está disponible dentro de las instrucciones CASE. No puede utilizar IS NULL en campos calculados independientes.

Sintaxis

`campo IS NULL`

Parámetros

`campo`: la dimensión o métrica que quiere evaluar.

Ejemplos

El campo contiene	Fórmula	Resultados
b	Campo IS NULL	falso
	Campo IS NULL	verdadero

IN

Devuelve el valor verdadero si el campo de entrada contiene alguno de los valores de la lista. Los valores del campo y de la lista deben ser del mismo tipo. La función IN devuelve valores completos y distingue entre mayúsculas y minúsculas. Para obtener coincidencias de valor parcial o que no distingan entre mayúsculas y minúsculas, utilice la función REGEXP_MATCH.

Esta función solo está disponible dentro de las instrucciones CASE. No puede utilizar IN en campos calculados independientes.

Sintaxis

`campo IN (valor1, valor2, ... valorN)`

Parámetros

`campo`: la dimensión o métrica que quiere evaluar.

`valor1, valor2, ... valorN`: valores que se deben buscar.

Ejemplos

El campo contiene	Tipo de campo	Fórmula	Resultados
Roberto	Texto	<i>Campo</i> IN ('Roberto', 'Alicia')	verdadero

Alicia	Texto	<i>Campo</i> IN ('Roberto', 'Alicia')	verdadero
Jorge	Texto	<i>Campo</i> IN ('Roberto', 'Alicia')	falso
20	Número	<i>Campo</i> IN (10, 20)	verdadero
10	Número	<i>Campo</i> IN (10, 20)	verdadero
100	Número	<i>Campo</i> IN (10, 20)	falso
Roberto	Texto	<i>Campo</i> IN ('Roberto', 20)	error, porque el campo es de texto y 20 es un número. Para solucionar el problema, escriba "20" entre comillas.

NOT IN

Devuelve el valor "true" (verdadero) si el campo de entrada no contiene ningún valor de la lista. Los valores del campo y de la lista deben ser del mismo tipo. La función IN devuelve valores completos y distingue entre mayúsculas y minúsculas. Para obtener coincidencias de valor parcial o que no distingan entre mayúsculas y minúsculas, utilice la función REGEXP_MATCH.

Esta función solo está disponible dentro de las instrucciones CASE. No puede utilizar NOT IN en campos calculados independientes.

Sintaxis

campo NOT IN (*valor1*, *valor2*, ...*valorN*)

Parámetros

campo: la dimensión o métrica que quiere evaluar.

valor1, *valor2*, ...*valorN*: valores que se deben buscar.

Ejemplos

El campo contiene	Tipo de campo	Fórmula	Resultados
Jorge	Texto	<i>Campo</i> NOT IN ('Roberto', 'Alicia')	verdadero
Alicia	Texto	<i>Campo</i> NOT IN ('Roberto', 'Alicia', 'David')	verdadero
Roberto	Texto	<i>Campo</i> NOT IN ('Roberto', 'Alicia')	falso
20	Número	<i>Campo</i> NOT IN (10, 20)	falso
10	Número	<i>Campo</i> NOT IN (10, 20)	falso
100	Número	<i>Campo</i> NOT IN (10, 20)	verdadero
Roberto	Texto	<i>Campo</i> NOT IN ('Roberto', 20)	error*

* El error se produce porque el tipo de *campo* es texto y 20 es un número. La fórmula sería válida si "20" estuviera entre comillas.

REGEXP_MATCH

Devuelve el valor "true" (verdadero) si el campo de entrada coincide con la expresión regular. El uso de esta función no se limita a las instrucciones CASE y se puede utilizar en otros tipos de campos calculados.

[Más información sobre REGEXP_MATCH](#)

Recursos relacionados

- [Campos calculados](#)
- [Lista de funciones de Data Studio](#)
- [Documentación de google/RE2 en GitHub](#)

¿Te ha resultado útil esta información?

Sí

No