

# DAX: Het gebruik van FILTER voorkomen als filterargument

28-1-2020 • 5 minutes to read • [Edit Online](#)

Als gegevensmodelbouwer is het gebruikelijk dat u DAX-expressies schrijft die moeten worden geëvalueerd in een aangepaste filtercontext. U kunt bijvoorbeeld een meetdefinitie schrijven om de verkoop van producten met een hoge marge te berekenen. Verderop in dit artikel wordt deze berekening beschreven.

## NOTE

Dit artikel is vooral relevant voor modelberekeningen waarmee filters worden toegepast op importtabellen.

De DAX-functies [CALCULATE](#) en [CALCULATETABLE](#) zijn belangrijke en nuttige functies. Hiermee kunt u berekeningen schrijven waarmee filters worden verwijderd of toegevoegd, of relatiepaden wijzigen. Dit wordt gedaan door het doorgeven van filterargumenten, die ofwel Booleaanse expressies, tabellexpressies of speciale filterfuncties zijn. In dit artikel bespreken we alleen Booleaanse en tabellexpressies.

Houd rekening met de volgende meetdefinitie die de verkoop van rode producten berekent met behulp van een tabellexpressie. Alle filters die kunnen worden toegepast op de tabel **Product** worden vervangen.

```
Red Sales =  
CALCULATE(  
    [Sales],  
    FILTER('Product', 'Product'[Color] = "Red")  
)
```

De functie [CALCULATE](#) accepteert een tabellexpressie die wordt geretourneerd door de DAX-functie [FILTER](#), waarmee de filterexpressie wordt geëvalueerd voor elke rij van de tabel **Product**. Het juiste resultaat wordt behaald: het verkoopresultaat voor rode producten. Het kan echter veel efficiënter worden bereikt met behulp van een Booleaanse expressie.

Hier volgt een verbeterde meetdefinitie, waarbij een Booleaanse expressie wordt gebruikt in plaats van de tabellexpressie. De DAX-functie [KEEPFILTERS](#) zorgt ervoor dat eventuele bestaande filters die zijn toegepast op de kolom **Kleur**, behouden blijven en niet worden overschreven.

```
Red Sales =  
CALCULATE(  
    [Sales],  
    KEEPFILTERS('Product'[Color] = "Red")  
)
```

U wordt aangeraden filterargumenten als Booleaanse expressies door te geven, wanneer dat mogelijk is. De reden hiervoor is dat importmodeltabellen kolomopslagplaatsen in het geheugen zijn. Ze zijn expliciet geoptimaliseerd om kolommen op deze manier efficiënt te filteren.

Er zijn echter beperkingen die van toepassing zijn op Booleaanse expressies wanneer ze worden gebruikt als filterargumenten. Deze:

- Kunnen geen kolommen vergelijken met andere kolommen
- Kunnen niet verwijzen naar een meting

- Kunnen geen geneste CALCULATE-functies gebruiken
- Kunnen geen functies gebruiken die een tabel scannen of retourneren

Dit betekent dat u tabellexpressies moet gebruiken voor complexere filtervereisten.

Overweeg nu een andere meetdefinitie.

```
High Margin Product Sales =
CALCULATE(
    [Sales],
    FILTER(
        'Product',
        'Product'[ListPrice] > 'Product'[StandardCost] * 2
    )
)
```

De definitie van een *product met hoge marge* is er een met een catalogusprijs die het dubbele van de standaardkosten overschrijdt. In dit voorbeeld moet de FILTER-functie worden gebruikt. De reden hiervoor is dat de filterexpressie te complex is voor een Booleaanse expressie.

Hier is nog een voorbeeld. De vereiste deze keer is om de omzet te berekenen, maar alleen voor maanden waarin winst is gemaakt.

```
Sales for Profitable Months =
CALCULATE(
    [Sales],
    FILTER(
        VALUES('Date'[Month]),
        [Profit] > 0)
    )
)
```

In dit voorbeeld moet ook de functie FILTER worden gebruikt. De reden hiervoor is dat de meting **Winst** moet worden geëvalueerd om die maanden te elimineren waarin geen winst is behaald. Het is niet mogelijk om een meting te gebruiken in een booleaanse expressie wanneer deze wordt gebruikt als een filterargument.

## Aanbevelingen

Voor de beste prestaties raden we u aan om waar mogelijk Booleaanse expressies als filterargumenten te gebruiken.

Daarom mag de functie FILTER alleen worden gebruikt wanneer dit nodig is. U kunt deze gebruiken om complexe kolomvergelijkingen te filteren. Deze kolomvergelijkingen kunnen bestaan uit:

- Metingen
- Overige kolommen
- Het gebruik van de DAX-functie [OR](#) of de logische operator OR (||)

## Volgende stappen

Bekijk de volgende bronnen voor meer informatie over dit artikel:

- [Naslaginformatie voor Data Analysis Expressions \(DAX\)](#)
- [Filterfuncties \(DAX\)](#)
- Vragen? [Misschien dat de Power BI-community het antwoord weet](#)