DAX

Intermediate

Alish Bista - 10/10/2025

Evaluation Context

Evaluation Context

The Important Fundamental

- Context determines which data DAX uses when calculating
- Answers: “What rows should I include in this calculation?”
- Two types: Filter Context and Row Context
- Every calculation in DAX happens inside a context

Filter Context

Filter Context Filters

- Filter Context = which rows are visible to your calculation
- Automatically created by visuals when you add fields
- Rows, columns, slicers, and filters all create filter context
- Each cell in a visual has its own unique filter context
- This is why the same measure shows different values in different places

Year	Product	[Total Sales]
2024	Clothing	Where Year=2024 & Product = Clothing
2025	Electronics	Where Year=2025 & Product = Electronics
Total		No filters

Row Context

Row Context Iterates

- Row Context = processing one specific row at a time
- Created in calculated columns (each row calculated individually)
- Created by iterator functions like SUMX
- Lets you reference column values directly without SUM, MAX, etc.
- Does NOT filter the model - just points to current row

```
Line Total =
Sales[Quantity] *
Sales[Price]
```

CALCULATE

CALCULATE

Most Common Yet Most Powerful

- CALCULATE is the most important function in DAX
- Modifies the filter context before calculating a measure
- **Syntax**: CALCULATE(<expression>, <filter1>, <filter2>, ...)
- Can add filters, remove filters, or replace filters
- Essential for percentages, comparisons, and time intelligence

```
Electronics Sales =
CALCULATE(
    [Total Sales],
    Products[Category] =
"Electronics"
)
```

```
% of Total =
DIVIDE(
    [Total Sales],
    CALCULATE([Total Sales],
ALL(Products))
)
```

CALCULATE

Modifying Filter Context

Without CALCULATE:

- Measure uses whatever filters the visual provides
- If visual shows Year=2024 & Category=Electronics, measure sees only that data

With CALCULATE:

- Takes the existing filter context from the visual
- Modifies it based on your filter arguments
- Then evaluates the measure with the NEW filter context

CALCULATE doesn’t ignore existing filters - it modifies them

```
Electronics Sales =
CALCULATE(
    [Total Sales],
    Products[Category] =
"Electronics"
)
```

```
% of Total =
DIVIDE(
    [Total Sales],
    CALCULATE([Total Sales],
ALL(Products))
)
```

Table Functions

EVALUATE

Evaluating Tables

- EVALUATE is used to run table expressions in DAX Studio or Tabular Editor
- Returns a table result (not a single value)
- Must be the first statement in a DAX query
- Used for testing and debugging table functions

FILTER

Filtering Tables

- FILTER returns a table with rows that meet a specific condition
- Creates row context to evaluate each row individually
- Often used inside CALCULATE to create complex filters
- Can be expensive performance-wise - use simple filters when possible

```
High Value Sales =
CALCULATE(
    [Total Sales],
    FILTER(Products,
Products[Price] > 1000)
)
```

ALL

Everything!

- ALL removes filters from specified table or columns
- Returns all rows in a table, ignoring current filter context
- Most commonly used inside CALCULATE
- Essential for percentage calculations and comparisons

```
% of Total =
DIVIDE([Total Sales],
CALCULATE([Total
Sales],
ALL(Products)))
```

ALLSELECTED

All Those are Selected

- ALLSELECTED removes filters from the visual but keeps external filters
- Respects slicers, page filters, and report filters
- Ignores only the row/column filters from the current visual
- Perfect for “% of visible total” calculations

Difference from ALL:

- ALL removes everything (including slicers)
- ALLSELECTED keeps external context (slicers stay active)

```
% of Visible =
DIVIDE([Total Sales],
CALCULATE([Total
Sales],
ALLSELECTED(Products)
))
```

VALUES

SWITCHing Among Options

- VALUES returns distinct values currently visible in filter context
- Respects all active filters from visuals and slicers
- Includes a blank row if there are missing relationships
- Essential for counting filtered items or creating dynamic lists

```
Product Count =
COUNTROWS(VALUES(Products[Produ
ctName]))


Selected Category =

IF(


COUNTROWS(VALUES(Products[Categ
ory])) = 1,

    VALUES(Products[Category]),

    "Multiple"

)
```

DISTINCT

SWITCHing Among Options

- DISTINCT returns unique values like VALUES
- Main difference: does NOT include the blank row
- Use when you don’t want blanks from missing relationships
- Slightly faster than VALUES in some scenarios

VALUES vs DISTINCT:

- VALUES: includes blank row for missing relationships
- DISTINCT: only actual values, no blank row

```
Actual Product Count
=
COUNTROWS(DISTINCT(Pr
oducts[ProductName]))
```

Q&A