INSIGHTSMASTERY ACADEMY

LEARNING HANDBOOK

Time Intelligence with DAX

Complete reference of DAX time intelligence functions
with practical examples for Power BI

Aligned with Microsoft Learn documentation
Time intelligence functions (DAX)

https://learn.microsoft.com/en-us/dax/time-intelligence-functions-dax

Power BI Desktop • Analysis Services • DAX

www.insightsmastry.in

Table of Contents

1. Introduction & Prerequisites

Data Analysis Expressions (DAX) includes **time intelligence functions** that enable you to manipulate data using time periods—days, weeks, months, quarters, and years—and then build and compare calculations over those periods.

This handbook is aligned with the official Microsoft Learn documentation: **Time intelligence functions (DAX)**.

Before using any time-intelligence function you **must** mark a table that contains a continuous date column as a **Date Table** in the model.

■ **Prerequisite checklist**

- A dedicated Date table with one row per day and no gaps
- Date column of Date (or DateTime at 00:00) type with unique values
- Table marked as Date Table (Table tools → Mark as date table)
- Relationship from fact date columns to the Date table
- Auto date/time disabled for cleaner models

2. Date Table Requirements

Microsoft requires the following for time intelligence to work correctly:

- All dates must be present for the years required (contiguous, no gaps).
- The Date table should start on January 1 and end on December 31 of the relevant years (or cover full fiscal years).
- A column of Date or DateTime type with unique values (usually named Date).
- The table must be marked as a date table when the relationship is not based on the Date column itself.

Recommended Date table (DAX)

```
DateTable =
ADDCOLUMNS (
    CALENDAR ( DATE ( 2020, 1, 1 ), DATE ( 2026, 12, 31 ) ),
    "Year", YEAR ( [Date] ),
    "Month Number", MONTH ( [Date] ),
    "Month Name", FORMAT ( [Date], "MMMM" ),
    "Quarter", "Q" & FORMAT ( [Date], "Q" ),
    "Year-Month", FORMAT ( [Date], "YYYY-MM" ),
    "Weekday", FORMAT ( [Date], "dddd" )
)
```

After creating the table: mark it as Date Table, create a many-to-one relationship from Sales[OrderDate] (or similar) to DateTable[Date], and use DateTable[Date] in all time intelligence expressions.

3. Function Categories Overview

Microsoft groups time intelligence functions by purpose. Understanding the groups helps you choose the right function quickly.

Category	Key Functions	Purpose
Period-to-Date	TOTALYTD, TOTALQTD, TOTALMTD, TOTALWTD, DATESYTD, DATESQTD, DATESMTD, DATESWTD	Cumulative values from the start of the period up to the current date in context.
Period Shift	DATEADD, SAMEPERIODLASTYEAR, PARALLELPERIOD	Move the current set of dates forward or backward by a number of intervals.
Previous / Next	PREVIOUSYEAR/QUARTER/MONTH/WEEK/DAY, NEXTYEAR/…	Return the entire previous or next calendar period relative to the current context.
Start / End of Period	STARTOFYEAR/QUARTER/MONTH/WEEK, ENDOFYEAR/…	Return the first or last date of the period that contains the current context.
Opening / Closing Balance	OPENINGBALANCEYEAR/…, CLOSINGBALANCEYEAR/…	Evaluate an expression on the first or last date of a period (typical for inventory, balances).
Custom Ranges	DATESBETWEEN, DATESINPERIOD, FIRSTDATE, LASTDATE	Build arbitrary continuous date ranges or extract boundary dates.

Table 1 — Function categories (Microsoft Learn)

4. Period-to-Date Functions

These functions return either a table of dates from the start of the period to the current date, or directly evaluate an expression over that range.

TOTALYTD / DATESYTD

TOTALYTD evaluates an expression for the year-to-date. **DATESYTD** returns the table of dates that can be used inside CALCULATE.

```
// Syntax (Microsoft Learn)
TOTALYTD ( ,  [, ] [, ] )
DATESYTD (  [, ] )

// Example — Sales Year-to-Date
Sales YTD =
TOTALYTD ( [Total Sales], DateTable[Date] )

// Equivalent with CALCULATE
Sales YTD =
CALCULATE ( [Total Sales], DATESYTD ( DateTable[Date] ) )

// Fiscal year ending 30 June
Sales Fiscal YTD =
TOTALYTD ( [Total Sales], DateTable[Date], "6/30" )
```

TOTALQTD / DATESQTD and TOTALMTD / DATESMTD

```
Sales QTD = TOTALQTD ( [Total Sales], DateTable[Date] )
Sales MTD = TOTALMTD ( [Total Sales], DateTable[Date] )

// Week-to-date (when week start is configured)
Sales WTD = TOTALWTD ( [Total Sales], DateTable[Date] )
```

■ When to use TOTAL* vs DATES*

Use TOTALYTD / TOTALQTD / TOTALMTD when you only need the aggregated value. Use DATESYTD / DATESQTD / DATESMTD inside CALCULATE when you need to combine the period filter with other filters or modifiers.

5. Period Shift Functions

DATEADD

Returns a table of dates shifted forward or backward by a specified number of intervals (DAY, MONTH, QUARTER, YEAR) from the dates in the current context.

```
// Syntax
DATEADD ( , ,  )

// Sales in the same period last year (shift -1 year)
Sales Prior Year =
CALCULATE ( [Total Sales], DATEADD ( DateTable[Date], -1, YEAR ) )

// Sales previous month
Sales Prior Month =
CALCULATE ( [Total Sales], DATEADD ( DateTable[Date], -1, MONTH ) )

// Sales two quarters ago
Sales 2Q Ago =
CALCULATE ( [Total Sales], DATEADD ( DateTable[Date], -2, QUARTER ) )
```

SAMEPERIODLASTYEAR

Specialized shortcut: returns dates shifted exactly one year back. Equivalent to DATEADD (…, -1, YEAR) for most calendar scenarios.

```
Sales SPLY =
CALCULATE ( [Total Sales], SAMEPERIODLASTYEAR ( DateTable[Date] ) )

// Year-over-Year growth
Sales YoY Growth % =
DIVIDE (
    [Total Sales] - [Sales SPLY],
    [Sales SPLY]
)
```

PARALLELPERIOD

Returns a period parallel to the dates in the current context, shifted by a number of intervals. The result always covers full months/quarters/years.

```
// Full previous quarter (not the same day range shifted)
Sales Parallel Prior Quarter =
CALCULATE (
    [Total Sales],
    PARALLELPERIOD ( DateTable[Date], -1, QUARTER )
)
```

■ DATEADD vs PARALLELPERIOD

DATEADD preserves the shape of the selection (e.g. 15 days stay 15 days). PARALLELPERIOD expands to the full parallel period (e.g. selecting March returns the whole previous quarter when shifting by quarter). Choose based on business need.

6. Previous / Next Period Functions

These helper functions return the entire previous or next calendar period based on the first (or last) date in the current context.

Previous period

```
Sales Previous Year =
CALCULATE ( [Total Sales], PREVIOUSYEAR ( DateTable[Date] ) )

Sales Previous Quarter =
CALCULATE ( [Total Sales], PREVIOUSQUARTER ( DateTable[Date] ) )

Sales Previous Month =
CALCULATE ( [Total Sales], PREVIOUSMONTH ( DateTable[Date] ) )

Sales Previous Week =
CALCULATE ( [Total Sales], PREVIOUSWEEK ( DateTable[Date] ) )

Sales Previous Day =
CALCULATE ( [Total Sales], PREVIOUSDAY ( DateTable[Date] ) )
```

Next period

```
Sales Next Year   = CALCULATE ( [Total Sales], NEXTYEAR ( DateTable[Date] ) )
Sales Next Quarter = CALCULATE ( [Total Sales], NEXTQUARTER ( DateTable[Date] ) )
Sales Next Month   = CALCULATE ( [Total Sales], NEXTMONTH ( DateTable[Date] ) )
Sales Next Week    = CALCULATE ( [Total Sales], NEXTWEEK ( DateTable[Date] ) )
Sales Next Day     = CALCULATE ( [Total Sales], NEXTDAY ( DateTable[Date] ) )
```

PREVIOUS* functions are useful when you want the complete prior calendar period rather than a shifted copy of the current selection.

7. Start / End of Period Functions

These return a single date: the first or last date of the week/month/quarter/year that contains the current context.

```
// Start of periods
Start of Year    = STARTOFYEAR ( DateTable[Date] )
Start of Quarter = STARTOFQUARTER ( DateTable[Date] )
Start of Month   = STARTOFMONTH ( DateTable[Date] )
Start of Week    = STARTOFWEEK ( DateTable[Date] )

// End of periods
End of Year    = ENDOFYEAR ( DateTable[Date] )
End of Quarter = ENDOFQUARTER ( DateTable[Date] )
End of Month   = ENDOFMONTH ( DateTable[Date] )
End of Week    = ENDOFWEEK ( DateTable[Date] )

// Example: Sales from start of year to end of selected month
Sales Full Months YTD =
CALCULATE (
    [Total Sales],
    DATESBETWEEN (
        DateTable[Date],
        STARTOFYEAR ( DateTable[Date] ),
        ENDOFMONTH ( DateTable[Date] )
    )
)
```

8. Opening & Closing Balance Functions

Ideal for inventory, account balances, headcount, or any semi-additive measure that should be evaluated on the first or last day of a period rather than summed across days.

```
// Closing balance of inventory at month end
Inventory Closing Month =
CLOSINGBALANCEMONTH (
    SUM ( Inventory[UnitsOnHand] ),
    DateTable[Date]
)

// Opening balance of the year
Inventory Opening Year =
OPENINGBALANCEYEAR (
    SUM ( Inventory[UnitsOnHand] ),
    DateTable[Date]
)

// Available functions:
// OPENINGBALANCEWEEK / MONTH / QUARTER / YEAR
// CLOSINGBALANCEWEEK / MONTH / QUARTER / YEAR
```

■ **Semi-additive measures**

Do not use SUM for balances across time. Use CLOSINGBALANCE* or LASTDATE patterns so the measure shows the value on the last day of the selected period.

9. Range Functions

DATESBETWEEN

Returns all dates from a start date to an end date (inclusive).

```
// Sales between two fixed dates
Sales In Range =
CALCULATE (
    [Total Sales],
    DATESBETWEEN ( DateTable[Date], DATE ( 2024, 3, 1 ), DATE ( 2024, 5, 31 ) )
)
```

DATESINPERIOD

Returns a set of dates that begins with a start date and continues for a specified number of intervals (DAY, MONTH, QUARTER, YEAR).

```
// Rolling 12 months ending at the last date in context
Sales Rolling 12M =
CALCULATE (
    [Total Sales],
    DATESINPERIOD (
        DateTable[Date],
        MAX ( DateTable[Date] ),
        -12,
        MONTH
    )
)

// Last 30 days
Sales Last 30 Days =
CALCULATE (
    [Total Sales],
    DATESINPERIOD ( DateTable[Date], MAX ( DateTable[Date] ), -30, DAY )
)
```

FIRSTDATE / LASTDATE

```
First Date in Context = FIRSTDATE ( DateTable[Date] )
Last Date in Context  = LASTDATE ( DateTable[Date] )
```

10. Practical Patterns & Complete Examples

Start with a simple base measure, then layer time intelligence on top. Always prefer measures over calculated columns for these calculations.

Base measure

```
Total Sales = SUM ( Sales[SalesAmount] )
```

Core time intelligence set

```
Sales YTD =
TOTALYTD ( [Total Sales], DateTable[Date] )

Sales PY =
CALCULATE ( [Total Sales], SAMEPERIODLASTYEAR ( DateTable[Date] ) )

Sales YoY % =
DIVIDE ( [Total Sales] - [Sales PY], [Sales PY] )

Sales MTD =
TOTALMTD ( [Total Sales], DateTable[Date] )

Sales PM =
CALCULATE ( [Total Sales], DATEADD ( DateTable[Date], -1, MONTH ) )

Sales Rolling 12 Months =
CALCULATE (
    [Total Sales],
    DATESINPERIOD ( DateTable[Date], MAX ( DateTable[Date] ), -12, MONTH )
)
```

Comparing multiple periods in one visual

```
// Useful for KPI cards or tables
Sales Current = [Total Sales]
Sales Prior Year = [Sales PY]
Sales YTD Current = [Sales YTD]
Sales YTD Prior Year =
CALCULATE ( [Sales YTD], SAMEPERIODLASTYEAR ( DateTable[Date] ) )
```

11. Common Pitfalls & Best Practices

Pitfalls

- Using a fact table date column instead of a marked Date table → blank or incorrect results.
- Gaps in the Date table (missing days) → time intelligence returns unexpected blanks.
- Forgetting to mark the table as Date Table.
- Leaving Auto Date/Time enabled (creates hidden tables and confusion).
- Applying time intelligence to calculated columns instead of measures.
- Mixing fiscal and calendar logic without explicit year_end_date or fiscal columns.

Best practices

- Always create and mark a dedicated Date table; disable Auto Date/Time.
- Write a base measure first, then build time intelligence measures on top of it.
- Prefer TOTALYTD / SAMEPERIODLASTYEAR / DATEADD over equivalent FILTER(ALL(…)) patterns for performance.
- Use VAR / RETURN to avoid recalculating the same measure multiple times.
- Test with known values (e.g. a single month and the same month last year) before publishing.
- Document fiscal year-end conventions in the model.

■ Blank results checklist

If a time intelligence measure returns blank: (1) Date table marked? (2) No gaps in dates? (3) Date column type is Date? (4) Relationship exists between fact and Date table?

12. Quick Reference Table

Function	Returns	Typical use
TOTALYTD / DATESYTD	YTD value / dates	Year-to-date totals
TOTALQTD / DATESQTD	QTD value / dates	Quarter-to-date
TOTALMTD / DATESMTD	MTD value / dates	Month-to-date
TOTALWTD / DATESWTD	WTD value / dates	Week-to-date
DATEADD	Shifted dates table	Any relative period
SAMEPERIODLASTYEAR	Dates –1 year	YoY comparison
PARALLELPERIOD	Full parallel period	Prior full quarter/year
PREVIOUSYEAR / MONTH …	Entire prior period	Full previous calendar period
NEXTYEAR / MONTH …	Entire next period	Full next calendar period
STARTOF* / ENDOF*	Single boundary date	Period boundaries
OPENINGBALANCE* / CLOSINGBALANCE*	Value on first/last day	Inventory, balances
DATESBETWEEN	Custom date range	Fixed start–end filter
DATESINPERIOD	Interval-based range	Rolling N months/days
FIRSTDATE / LASTDATE	First / last date in context	Boundary helpers

Table 2 — Quick reference (aligned with Microsoft Learn)

13. Official Reference

This handbook is designed to complement the official documentation. Always refer to Microsoft Learn for the latest syntax, parameter details, and remarks.

Primary source

Time intelligence functions (DAX)
https://learn.microsoft.com/en-us/dax/time-intelligence-functions-dax

Individual function pages (DATEADD, TOTALYTD, SAMEPERIODLASTYEAR, etc.) contain additional examples, remarks on calendar-based time intelligence, and behavior notes.

Related Microsoft Learn topics: Create date tables in Power BI Desktop, Use DAX time intelligence functions (training module), Mark as date table.

InsightsMastery Academy

Time Intelligence Handbook — DAX

Build accurate period comparisons. Trust your numbers.

www.insightsmastry.in