Power BI

Basic to Advanced

Part - 6



Sreerag A

AGENDA

- Data Analysis Expressions (DAX)
- Calculated table
- Calculated column
- Measure
- Calendar table
- Aggregation functions

Date & Time functions



Data Analysis Expressions (DAX)

- DAX, which stands for Data Analysis Expressions, is a powerful formula language used in Power BI and other Microsoft products like Excel and Analysis Services.
- It is specifically designed for working with tabular data models, allowing users to create custom calculations, aggregations, and expressions to derive insights from their data.

DAX functions and formulas are used to manipulate and transform data within Power BI, making it a fundamental component for data modeling, analysis, and dynamic reports and dashboards.



- It is a user-defined table created within the data model using DAX formulas.
- It allows you to define custom tables by specifying criteria, conditions, or calculations, and it refreshes dynamically as data changes in the source tables.

Calculated column

It is a user-defined column that you create

within a table in your data model.

 Unlike regular columns in a table, which are populated with data from the data source, a

calculated column is derived from a calculation or expression that you define.

 It contains values computed based on a formula you specify using the DAX language.



- It is a user-defined calculation that is performed on the data within a data model. Unlike regular measures, which are typically based on straightforward aggregations like sums or averages, calculated measures allow you to create custom calculations using the DAX language.
- Calculated measures are particularly useful when you need to perform complex

calculations, create custom Key Performance Indicators (KPIs), or derive insights that are not available directly from the data source.



Calendar table

- A Calendar Table, also known as a Date Dimension or Time Table, is a specialized table in a data model that contains a continuous and comprehensive list of dates.
- It covers a range of dates, starting from the earliest date in your dataset to the latest date.
- Calendar tables are essential for time-based analysis, reporting, and data modeling.
- They provide a consistent framework for organizing and analyzing data related to dates, such as sales, events, and trends.
- To create calendar table:

Click on 'New table' button in the Modeling tab. Enter the formula: Calendar = CALENDARAUTO()

Click enter to create calendar table



1) Aggregation functions

- Aggregation functions calculate a (scalar) value such as count, sum, average, minimum, or maximum for all rows in a column or table as defined by the expression.
- These functions are often used in calculated columns, measures, and other DAX formulas to perform calculations on data.

➢ <u>Sum</u>

It calculates the total sum of numeric values in a column or a table.

$\mathbf{\frown}$		

#Here, 'Sales' is the table

Syntax: SUM(<column>)

E.g.: = SUM(Sales[Amt])

Average

name, and 'Amt' is the column in it. Similarly, you can identify column names in other equations using square braces.

Returns the average (arithmetic mean) of all the numbers in a column.

Syntax: AVERAGE(<column>)

E.g.: = AVERAGE(InternetSales[ExtendedSalesAmount])

► Max

Returns the largest numeric value or largest string in a column, or the larger value between two scalar expressions. Ignores logical values.

Syntax: MAX(<column>)</column>		or	MAX(<expression1>,</expression1>
<expression2>)</expression2>			

E.g.: = MAX(InternetSales[ExtendedAmount])

➢ Min

Returns the smallest numeric value or smallest string in a

column, or the smaller value between two scalar expressions.

Ignores logical values.

Syntax: MIN(<column>) MIN(<expression1>, or <expression2>)

E.g.: = MIN([TransactionDate]) or = Min([TotalSales],

[TotalPurchases])

➢ Count

Counts the number of rows in the specified column that

contain non-blank values.

Syntax: COUNT(<column>)

E.g.: = COUNT([ShipDate])



Distinctcount

Counts the number of distinct values in a column.

Syntax: DISTINCTCOUNT(<column>)

E.g.: =

DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])

➢ Sumx

Returns the sum of an expression evaluated for each row in a

table.

```
Syntax: SUMX(, <expression>)
```

```
E.g.: = SUMX(FILTER(InternetSales,
```

InternetSales[SalesTerritoryID]=5),[Freight])

There are other aggregated functions such as averagex,

counta, countblank, product, etc. To know more about them

refer to the following link. <u>Aggregation functions (DAX) -</u>

DAX | Microsoft Learn



2) Date & Time functions

 These functions help you create calculations based on dates and times. Many of the functions in DAX are similar to the Excel date and time functions. However, DAX functions use a datetime data type and can take values from a column as an argument.

Calendar

Returns a table with a single column named "Date" that contains a contiguous set of dates. The range of dates is from the specified start date to the specified end date,

inclusive of those two dates.

Syntax: CALENDAR(<start_date>, <end_date>)

E.g.: = CALENDAR (DATE (2015, 1, 1), DATE (2021, 12, 31))

➢ Date

Returns the specified date in datetime format.

Syntax: DATE(<year>, <month>, <day>)

E.g.: = DATE(2009,7,8)



Datediff

Returns the number of units (unit specified in Interval) between the input two dates. A positive result is returned if Date2 is larger than Date1. A negative result is returned if Date1 is larger than Date2.

Syntax: DATEDIFF(<Date1>, <Date2>, <Interval>)

E.g.: = DATEDIFF(StartDate, EndDate, YEAR)

StartDate and EndDate can be specified by DATE(). DAY, WEEK, MONTH, QUARTER, and YEAR can be given as intervals.

➢ Day

Returns the day of the month, a number from 1 to 31.

Syntax: DAY(<date>)

E.g.: = DAY([Birthdate])

➢ Edate

Returns the date that is the indicated number of months before or after the start date. Use EDATE to calculate maturity dates or due dates that fall on the same day of the

month as the date of issue.

Syntax: EDATE(<start_date>, <months>)

E.g.: = EDATE([TransactionDate],3)

Networkdays

Returns the number of whole workdays between two dates (inclusive). Parameters specify which and how many days are weekend days. Weekend days and days specified as holidays are not considered as workdays.

 Dates must be specified by using the DATE function or as the result of another expression. For example, DATE (2022, 5, 30), specifies May 30th, 2022. Dates can also be specified as a literal in format, (dt"2022-05-30"). Do not specify dates as text.

Syntax: NETWORKDAYS(<start_date>, <end_date>,

[<weekend>, <holidays>])

E.g.: = NETWORKDAYS (DATE (2022, 5, 28), DATE (2022, 5,

30), 1, {DATE (2022, 5, 30)})

- Weekend number values indicate the following weekend days:
- 1 or omitted: Saturday, Sunday 3: Monday, Tuesday 5: Wednesday, Thursday 7: Friday, Saturday 12: Monday only 14: Wednesday only 16: Friday only
- 2: Sunday, Monday 4: Tuesday, Wednesday 6: Thursday, Friday 11: Sunday only 13: Tuesday only 15: Thursday only 17: Saturday only



➤ Today

Returns the current date.

Syntax: TODAY()

Weekday

Returns a number from 1 to 7 identifying the day of the week

of a date. By default, the day ranges from 1 (Sunday) to 7

(Saturday).

Syntax: WEEKDAY(<date>, <return_type>)

E.g.: = WEEKDAY([HireDate], 1)

Return type: 1, the week begins on Sunday (1) and ends on Saturday (7). numbered 1 through 7.

Return type: 2, week begins on Monday (1) and ends on

Sunday (7).

Return type: 3, week begins on Monday (0) and ends on Sunday (6). numbered 1 through 7.



➢ Weeknum

Returns the week number for the given date according to the return_type value. The week number indicates where the week falls numerically within a year.

There are two systems used for this function:
System 1 - The week containing January 1 is the first week of the year and is numbered week 1.

System 2 - The week containing the first Thursday of the year is the first week of the year and is numbered as week 1. This system is the methodology specified in ISO 8601, which is commonly known as the European week numbering system.

Syntax: WEEKNUM(<date>[, <return_type>])

E.g.: = WEEKNUM("Feb 14, 2010", 2)

Return type: 1 or omitted, week begins on Sunday

Return type: 11, week begins on Monday

Return type: 11, week begins on Monday Return type: 12, week begins on Tuesday Return type: 13, week begins on Wednesday Return type: 14, week begins on Thursday Return type: 15, week begins on Friday Return type: 16, week begins on Saturday Return type: 17, week begins on Saturday Return type: 2, week begins on Monday Return type: 21, Based on the European week numbering system, the week begins on Monday

➤ Yearfrac

Calculates the fraction of the year represented by the number of whole days between two dates. Use the YEARFRAC worksheet function to identify the proportion of a whole year's benefits or obligations to assign to a specific term.

Syntax: YEARFRAC(<start_date>, <end_date>, <basis>)

E.g.: =

YEARFRAC(Orders[TransactionDate],Orders[ShippingDate])

- Basis (Optional) The type of day count basis to use. All arguments are truncated to integers.
 - 0 US (NASD) 30/360 (Default value)
 - 1 Actual/actual
 - 2 Actual/360
 - 3 Actual/365
 - 4 European 30/360

There are other date and time functions such as eomonth, year, quarter, etc. To know more about them refer to the following link. <u>Date and time functions (DAX) - DAX | Microsoft</u> Learn

Thank you!

