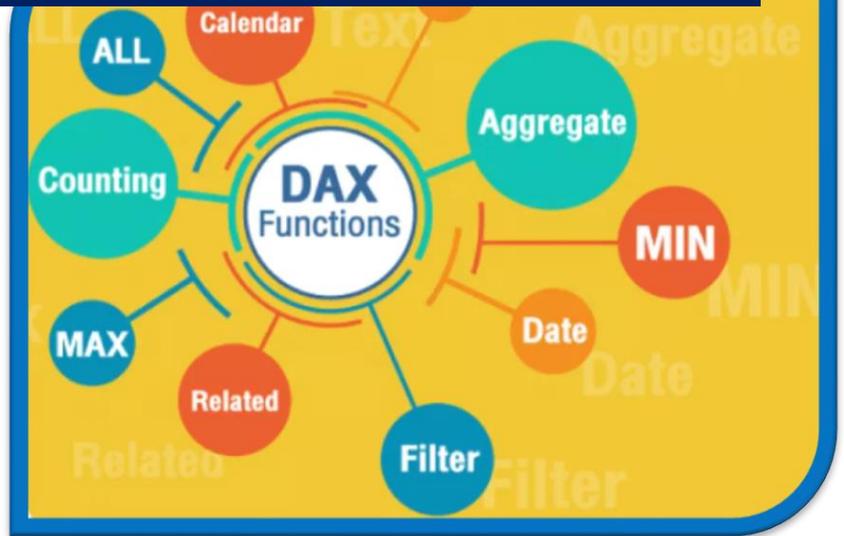


Power BI - Data Analysis Expressions (DAX)



Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Cognitive Convergence is Subject Matter Expert in Office 365, Dynamics 365, SharePoint, Project Server, Power Platform: Power Apps-Power BI-Power Automate-Power Virtual Agents.

We offer Power BI consulting services covering solution architecture refinement, customization, integration, transformation, visualization and analytics to uncover insights hidden within data and enhance data exploration.

Contents

Objectives	3
Background	3
Data Analysis Expression (DAX)	3
DAX Calculation type	3
<i>Calculated Columns</i>	4
<i>Creating a Column Formula using DAX</i>	4
<i>Calculated Measures</i>	5
<i>Creating a Measure Formula using DAX</i>	5
DAX Formula – Syntax	8
<i>DAX Functions</i>	9
Types of DAX functions	9
<i>Date and Time Functions</i>	9
<i>Time Intelligence Functions</i>	10
<i>Information Functions</i>	11
<i>Logical Functions</i>	11
<i>Mathematical and Trigonometric Functions</i>	12
<i>Statistical Functions</i>	13
<i>Text Functions</i>	14
<i>Parent-Child functions</i>	15
<i>Other functions</i>	16
DAX Context	16
DAX Statements	17
Power Query computed columns	17
comparisson between DAX calculated columns and Power Query computed columns	17
Challenges and solutions	18



Conclusion..... 19



OBJECTIVES

This paper will discuss Data Analysis Expression (DAX) which is a formula language, its use, functions and calculation types will discuss here.

BACKGROUND

DAX concepts are simple and straightforward, but DAX is powerful. DAX uses some unique programming concepts and patterns that can make it difficult to fully use and understand. Traditional ways of learning languages may not be the best approach to DAX, so the goal of this video is to teach you concepts and theory that will help later on in your Power BI work.



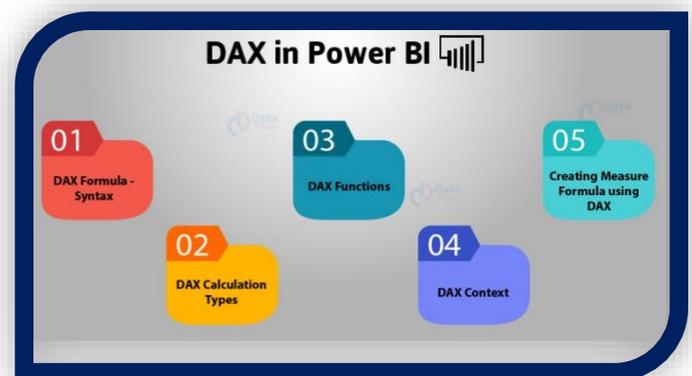
DATA ANALYSIS EXPRESSION (DAX)

DAX (Data Analysis Expressions) is a formula expression language and can be used in different BI and visualization tools. DAX is also known as function language, where the full code is kept inside a function.

In DAX, functions can contain other nested functions, conditional statements, and value references. Execution in DAX starts from the innermost function or parameter, and works outward. In Power BI, DAX formulas are written in a single line, so formatting your functions correctly is important for readability. DAX uses operator overloading, which means you can mix data types in your calculations and the results will change based on the type of data used in the inputs. Conversion happens automatically, which means you don't have to know the data types of the columns you're working with in Power BI, but it also means that sometimes conversion can happen in unexpected ways. It's good practice to understand the data you're using to ensure that your operators are behaving as anticipated.

DAX is designed to work with tables, DAX programming formula contains two data types:

- **Numeric:** integers, currency and decimals
- **Other:** string and binary object.



DAX CALCULATION TYPE

There are two primary calculations that can be created using DAX:



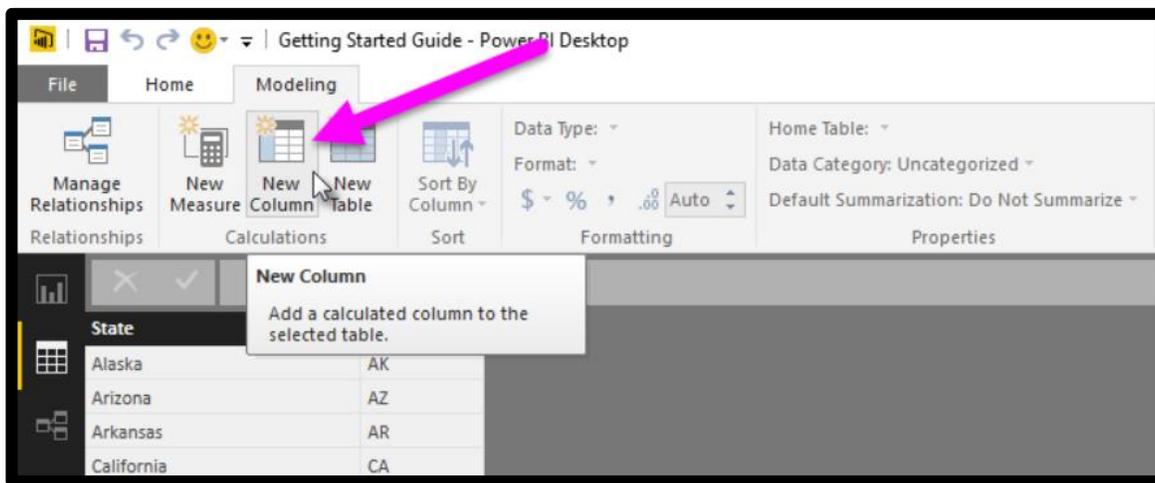
Calculated Columns

The calculated columns create a new column in your existing table. The only difference between a regular column and a calculated column is that it is necessary to have at least one function in the calculated column. These are used when you want to create a column with filtered or sorted information. To create a calculated column:

- Go to the Modeling tab in Power BI Desktop.
- Then select New Column option. A Formula bar will open showing "Column =". You can replace the "Column" word with the column name you want.
- After this, enter the expression for the calculated column on the right of the equals to sign.

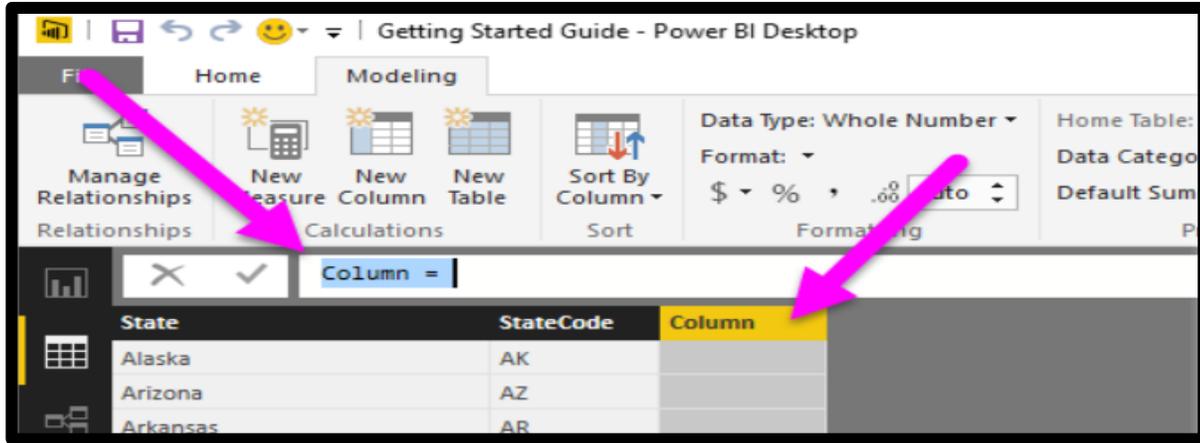
Creating a Column Formula using DAX

Step 1: Navigate to the Modeling tab, there is a New Column option at the top of the screen. This also opens the formula bar where you can enter DAX formula to perform the calculation, there is also an availability to rename the column by changing the Column text in the formula bar.



Step 2: Once you select the New Column button, the Formula Bar is populated with a basic column name (which you change to suit your formula, of course) and the = operator, and the new column appears in the data grid, as shown in the following image.





The required elements for a calculated column are the following:

- a new column name
- at least one function or expression

Calculated Measures

A calculated measure creates a field having aggregated values such a sum, ratios, percentages, averages, etc. To create a calculated measure:

- Go to the Modeling tab in Power BI Desktop.
- Then select New Measure option. A Formula bar will open showing "Measure =". You can replace the "Measure" word with the measure name you want.
- After this, enter the expression for the calculated measure on the right of the equals to sign.
- Once the measure is created, it can modify your measure name with a calculator icon next to it, under the table name of created the measure in.

Creating a Measure Formula using DAX

Step 1: To create a measure using the DAX formula, open Power BI Desktop app. In the Power BI Desktop, there are two ways of creating new measures in Power BI. One, right-click on the table's name in which to add the measure field and select New measure.

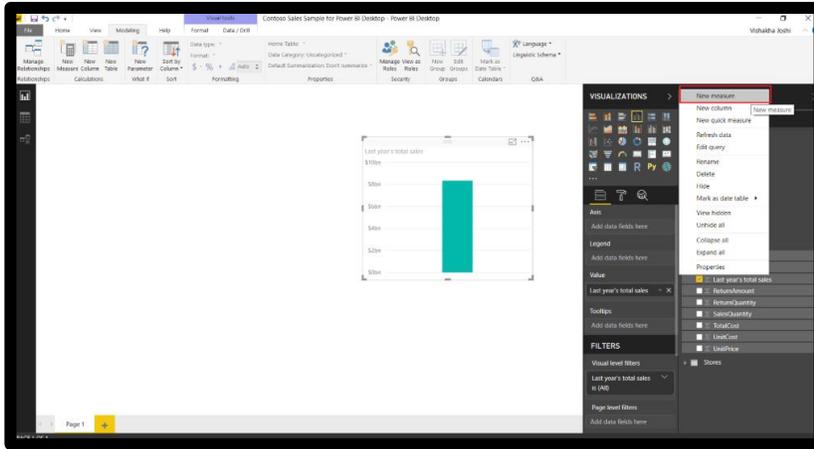
We use DAX to solve several calculations and data analysis problems that can help you create new information from data that is already in your model.

Cognitive Convergence

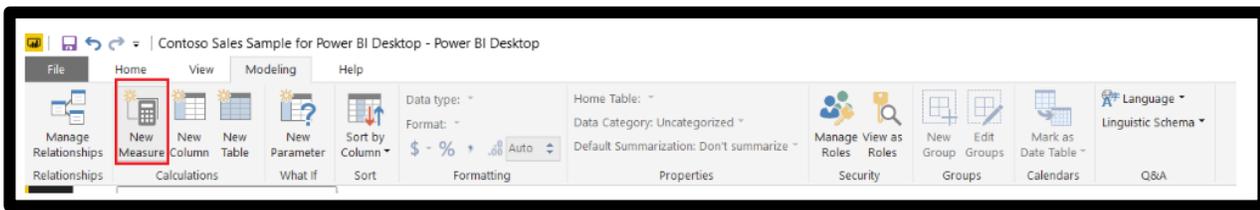
<http://www.cognitiveconvergence.com>

+1 4242530744

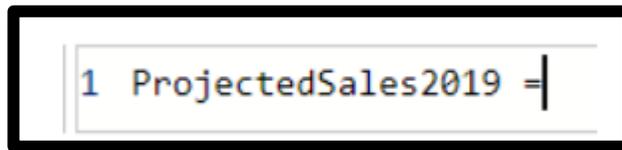
shahzad@cognitiveconvergence.com



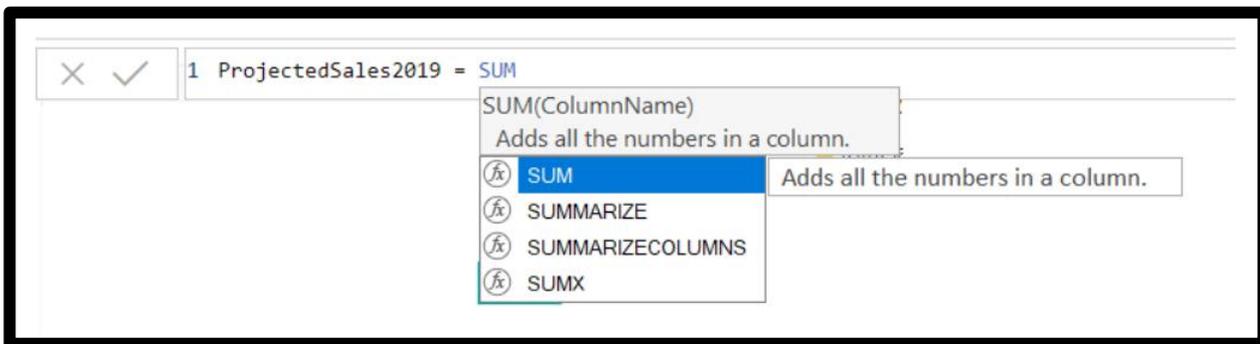
Or, go to the Modeling tab and click on “New measure” option.



Step 2: In the formula bar, start with entering the name of new measure. In the supposed scenario, we are using the name ProjectedSales2019.

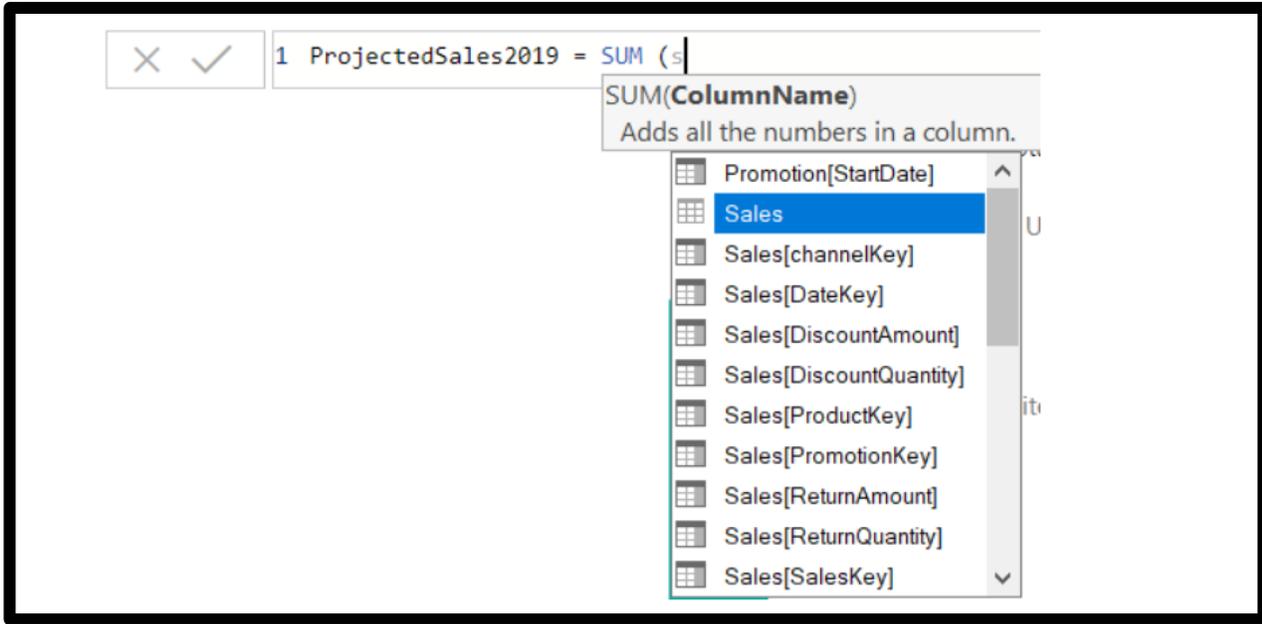


Step 3: Next, enter the name of the function. Start typing the name of the function and suggestions will appear in the list. A small section showing a description of the function selected will also appear. Notice in the image below where we have selected the SUM function.

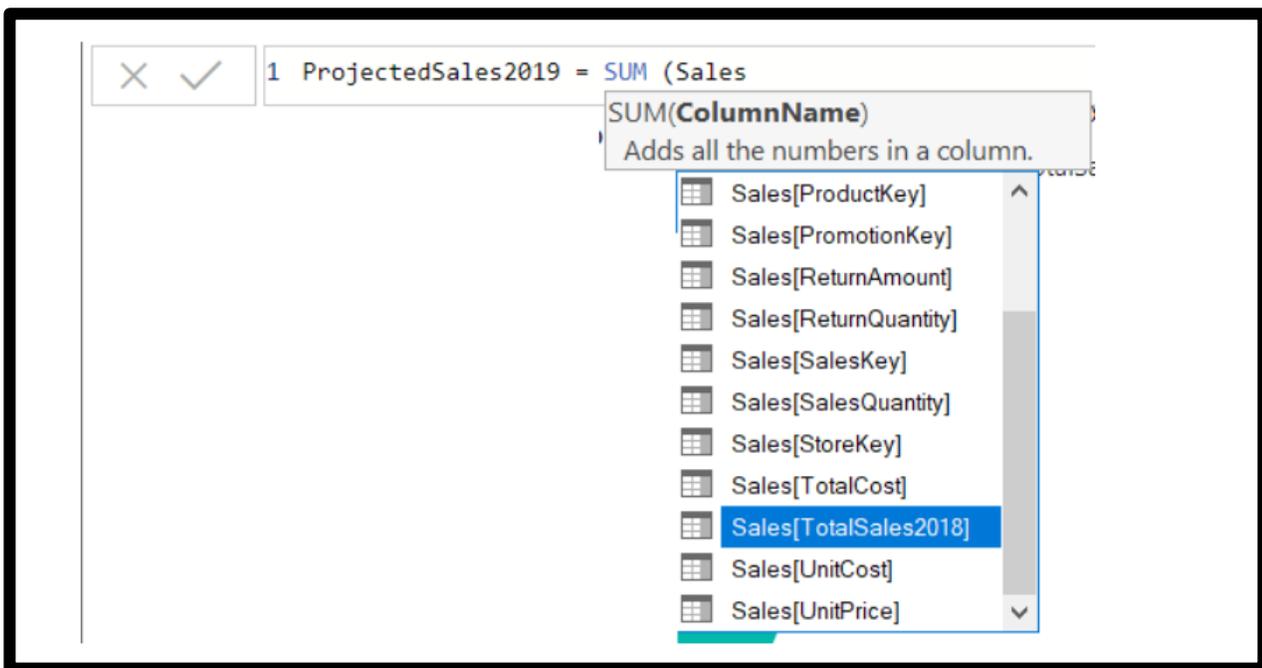


Step 4: Then, put a parenthesis “)” and write the name of the table or column referring to.

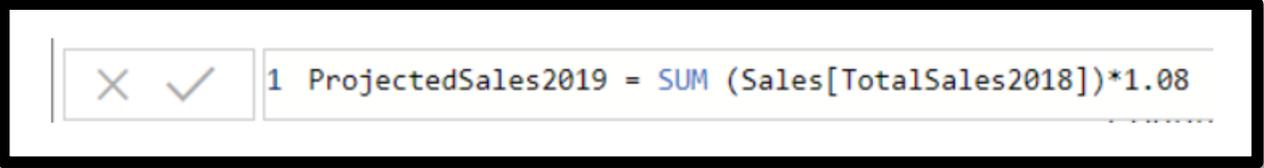




Step 5: The function will apply to the values within the selected column. Close the parenthesis.



Step 6: Further, you can add more operators, other functions, constants to increase the complexity of the formula.



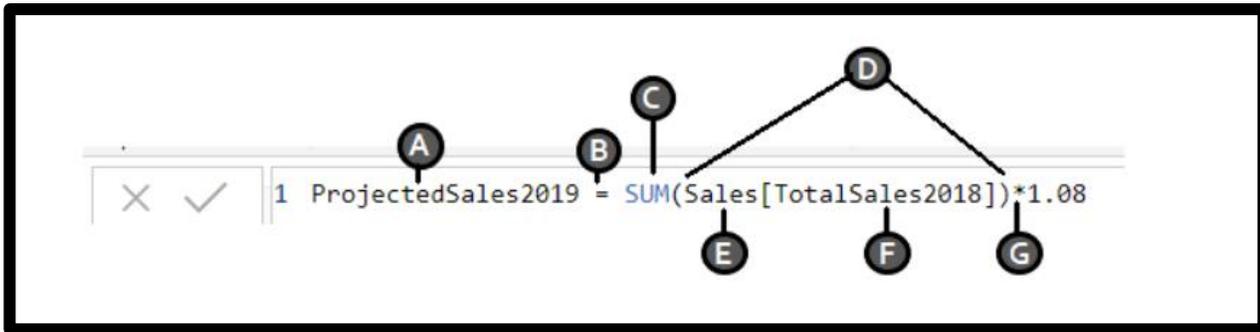
```
1 ProjectedSales2019 = SUM (Sales[TotalSales2018])*1.08
```

The DAX formula that we created will create a new measure named "ProjectedSales2019"

```
ProjectedSales2019 = SUM(Sales[TotalSales2018])* 1.08
```

DAX FORMULA - SYNTAX

The initial and most crucial step in learning any language is to break it down into definitive elements and understand its elements. And, that is why we study the syntax of a language.



- **A:** It is the name of the new measure (Projected Sales 2019).
- **B:** It is the equals sign (=) which is an operator indicating the start of the DAX formula and equating the two sides.
- **C:** It is the DAX function used to add the values of a given field (Total Sales 2018) from a table (Sales). The function used here is SUM.
- **D:** The parenthesis () is used to enclose and define arguments in an expression. Every function must have at least one argument.
- **E:** It is the name of the table from which a field or column is taken in the formula (Sales).
- **F:** It is the name of the field from which the formula will use the values. For instance, the function SUM will apply itself on the values of the column or field [Total Sales 2018] of the table Sales.
- **G:** It is another operator used for multiplication. Although, the syntax elements from A to F constitutes the basic syntax of DAX.

Thus, in simpler words, this DAX formula commands the system to calculate the product of sum of the values in Total Sales 2018 and 1.08 (8% increase) and store the value in a new field or column known as Projected Sales 2019.

We strive to get better at DAX taking the smarter approach to make your interactive dashboard.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

DAX Functions

A DAX function is a predefined formula which performs calculations on values provided to it in arguments. The arguments in a function need to be in a particular order and can be a column reference, numbers, text, constants, another formula or function, or a logical value such as TRUE or FALSE. Every function performs a particular operation on the values enclosed in an argument. You can use more than one argument in a DAX formula.

Key Points about DAX Functions

Here are some unique facts about DAX functions that you must know to understand them better:

- Any DAX function always refers to a complete column/field or a table. It will never refer to individual values. If you want to use the functions on separate values within a column, you need to apply filters in a DAX formula.
- DAX functions provide the flexibility to create a formula that is applied on a row-by-row basis. The calculations or formulas get applied as per the context of the values in each row.
- In some cases, DAX functions return a full table which can be used in other DAX formulas that need a complete set of values. However, you cannot display this table's contents.
- DAX functions have a category known as time intelligence functions. Such functions are used to calculate time/date ranges and periods.

TYPES OF DAX FUNCTIONS

Date and Time Functions

The date time functions carry out calculations on the date and time values. The data type of these values is always datetime data type.

Function	Description
CALENDAR	Returns a table with a single column named "Date" that contains a contiguous set of dates.
CALENDARAUTO	Returns a table with a single column named "Date" that contains a contiguous set of dates.
DATE	Returns the specified date in datetime format.
DATEDIFF	Returns the count of interval boundaries crossed between two dates.
DATEVALUE	Converts a date in the form of text to a date in datetime format.
DAY	Returns the day of the month, a number from 1 to 31.
EDATE	Returns the date that is the indicated number of months before or after the start date.
EOMONTH	Returns the date in datetime format of the last day of the month, before or after a specified number of months.
HOUR	Returns the hour as a number from 0 (12:00 A.M.) to 23 (11:00 P.M.).
MINUTE	Returns the minute as a number from 0 to 59, given a date and time value.
MONTH	Returns the month as a number from 1 (January) to 12 (December).
NOW	Returns the current date and time in datetime format.
QUARTER	Returns the quarter as a number from 1 to 4.
SECOND	Returns the seconds of a time value, as a number from 0 to 59.
TIME	Converts hours, minutes, and seconds given as numbers to a time in datetime format.
TIMEVALUE	Converts a time in text format to a time in datetime format.
TODAY	Returns the current date.
WEEKDAY	Returns a number from 1 to 7 identifying the day of the week of a date.
WEEKNUM	Returns the week number for the given date and year according to the return_type value.
YEAR	Returns the year of a date as a four digit integer in the range 1900-9999.
YEARFRAC	Calculates the fraction of the year represented by the number of whole days between two dates.

Time Intelligence Functions

The time-intelligence functions are used to evaluate values over a fixed period such as days, weeks, months, quarter, years, etc. You can specify a time period using these functions and compare two scenarios in your report.

Function	Description
CLOSINGBALANCEMONTH	Evaluates the expression at the last date of the month in the current context.
CLOSINGBALANCEQUARTER	Evaluates the expression at the last date of the quarter in the current context.
CLOSINGBALANCEYEAR	Evaluates the expression at the last date of the year in the current context.
DATEADD	Returns a table that contains a column of dates, shifted either forward or backward in time by the specified number of intervals from the dates in the current context.
DATESBETWEEN	Returns a table that contains a column of dates that begins with the start_date and continues until the end_date.
DATESINPERIOD	Returns a table that contains a column of dates that begins with the start_date and continues for the specified number_of_intervals.
DATESMTD	Returns a table that contains a column of the dates for the month to date, in the current context.
DATESQTD	Returns a table that contains a column of the dates for the quarter to date, in the current context.
DATESYTD	Returns a table that contains a column of the dates for the year to date, in the current context.
ENDOFMONTH	Returns the last date of the month in the current context for the specified column of dates.
ENDOFQUARTER	Returns the last date of the quarter in the current context for the specified column of dates.
ENDOFYEAR	Returns the last date of the year in the current context for the specified column of dates.
FIRSTDATE	Returns the first date in the current context for the specified column of dates.
FIRSTNONBLANK	Returns the first value in the column, column, filtered by the current context, where the expression is not blank.
LASTDATE	Returns the last date in the current context for the specified column of dates.
LASTNONBLANK	Returns the last value in the column, column, filtered by the current context, where the expression is not blank.
NEXTDAY	Returns a table that contains a column of all dates from the next day, based on the first date specified in the dates column in the current context.
NEXTMONTH	Returns a table that contains a column of all dates from the next month, based on the first date in the dates column in the current context.
NEXTQUARTER	Returns a table that contains a column of all dates in the next quarter, based on the first date specified in the dates column, in the current context.
NEXTYEAR	Returns a table that contains a column of all dates in the next year, based on the first date in the dates column, in the current context.
OPENINGBALANCEMONTH	Evaluates the expression at the first date of the month in the current context.
OPENINGBALANCEQUARTER	Evaluates the expression at the first date of the quarter, in the current context.
OPENINGBALANCEYEAR	Evaluates the expression at the first date of the year in the current context.
PARALLELPERIOD	Returns a table that contains a column of dates that represents a period parallel to the dates in the specified dates column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.
PREVIOUSDAY	Returns a table that contains a column of all dates representing the day that is previous to the first date in the dates column, in the current context.
PREVIOUSMONTH	Returns a table that contains a column of all dates from the previous month, based on the first date in the dates column, in the current context.
PREVIOUSQUARTER	Returns a table that contains a column of all dates from the previous quarter, based on the first date in the dates column, in the current context.
PREVIOUSYEAR	Returns a table that contains a column of all dates from the previous year, given the last date in the dates column, in the current context.

SAMEPERIODLASTYEAR	Returns a table that contains a column of dates shifted one year back in time from the dates in the specified dates column, in the current context.
STARTOFMONTH	Returns the first date of the month in the current context for the specified column of dates.
STARTOFQUARTER	Returns the first date of the quarter in the current context for the specified column of dates.
STARTOFYEAR	Returns the first date of the year in the current context for the specified column of dates.
TOTALMTD	Evaluates the value of the expression for the month to date, in the current context.
TOTALQTD	Evaluates the value of the expression for the dates in the quarter to date, in the current context.
TOTALYTD	Evaluates the year-to-date value of the expression in the current context.

Information Functions

The information functions are used to provide certain information on the data values contained in rows and columns. It evaluates the given condition in a function for the value given and return TRUE or FALSE. For instance, the function ISERROR will return TRUE if the value evaluated contains an error.

Function	Description
CONTAINS	Returns true if values for all referred columns exist, or are contained, in those columns; otherwise, the function returns false.
CUSTOMDATA	Returns the content of the CustomData property in the connection string.
ISBLANK	Checks whether a value is blank, and returns TRUE or FALSE.
ISERROR	Checks whether a value is an error, and returns TRUE or FALSE.
ISEVEN	Returns TRUE if number is even, or FALSE if number is odd.
ISINSCOPE	Returns true when the specified column is the level in a hierarchy of levels.
ISLOGICAL	Checks whether a value is a logical value, (TRUE or FALSE), and returns TRUE or FALSE.
ISNONTEXT	Checks if a value is not text (blank cells are not text), and returns TRUE or FALSE.
ISNUMBER	Checks whether a value is a number, and returns TRUE or FALSE.
ISONORAFTER	A boolean function that emulates the behavior of a Start At clause and returns true for a row that meets all of the condition parameters.
ISTEXT	Checks if a value is text, and returns TRUE or FALSE.
LOOKUPVALUE	Returns the value in <i>result_columnName</i> for the row that meets all criteria specified by <i>search_columnName</i> and <i>search_value</i> .
USERNAME	Returns the domain name and username from the credentials given to the system at connection time.
USEROBJECTID	Returns the current user's Object ID or SID.
USERPRINCIPALNAME	Returns the user principal name.

Logical Functions

The logical functions are used to evaluate an expression or argument logically and return TRUE or FALSE if the condition is met or not.

Function	Description
AND	Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE.
COALESCE	Returns the first expression that does not evaluate to BLANK.
FALSE	Returns the logical value FALSE.
IF	Checks if a condition provided as the first argument is met.
IF.EAGER	Uses eager execution to check if a condition provided as the first argument is met.
IFERROR	Evaluates an expression and returns a specified value if the expression returns an error
NOT	Changes FALSE to TRUE, or TRUE to FALSE.



OR	Checks whether one of the arguments is TRUE to return TRUE.
SWITCH	Evaluates an expression against a list of values and returns one of multiple possible result expressions.
TRUE	Returns the logical value TRUE.

Mathematical and Trigonometric Functions

The mathematical and trig functions are used to perform all sorts of mathematical functions on the referred values. Given below, is a list of all the available math and trig DAX functions in Power BI.

Function	Description
ABS	Returns the absolute value of a number.
ACOS	Returns the arccosine, or inverse cosine, of a number.
ACOSH	Returns the inverse hyperbolic cosine of a number.
ACOT	Returns the arccotangent, or inverse cotangent, of a number.
ACOTH	Returns the inverse hyperbolic cotangent of a number.
ASIN	Returns the arcsine, or inverse sine, of a number.
ASINH	Returns the inverse hyperbolic sine of a number.
ATAN	Returns the arctangent, or inverse tangent, of a number.
ATANH	Returns the inverse hyperbolic tangent of a number.
CEILING	Rounds a number up, to the nearest integer or to the nearest multiple of significance.
COMBIN	Returns the number of combinations for a given number of items.
COMBINA	Returns the number of combinations (with repetitions) for a given number of items.
COS	Returns the cosine of the given angle.
COSH	Returns the hyperbolic cosine of a number.
CURRENCY	Evaluates the argument and returns the result as currency data type.
DEGREES	Converts radians into degrees.
DIVIDE	Performs division and returns alternate result or BLANK() on division by 0.
EVEN	Returns number rounded up to the nearest even integer.
EXP	Returns e raised to the power of a given number.
FACT	Returns the factorial of a number, equal to the series $1*2*3*...*$, ending in the given number.
FLOOR	Rounds a number down, toward zero, to the nearest multiple of significance.
GCD	Returns the greatest common divisor of two or more integers.
INT	Rounds a number down to the nearest integer.
ISO.CEILING	Rounds a number up, to the nearest integer or to the nearest multiple of significance.
LCM	Returns the least common multiple of integers.
LN	Returns the natural logarithm of a number.
LOG	Returns the logarithm of a number to the base you specify.
LOG10	Returns the base-10 logarithm of a number.
MROUND	Returns a number rounded to the desired multiple.
ODD	Returns number rounded up to the nearest odd integer.
PI	Returns the value of Pi, 3.14159265358979, accurate to 15 digits.
POWER	Returns the result of a number raised to a power.
PRODUCT	Returns the product of the numbers in a column.
PRODUCTX	Returns the product of an expression evaluated for each row in a table.
QUOTIENT	Performs division and returns only the integer portion of the division result.
RADIANS	Converts degrees to radians.
RAND	Returns a random number greater than or equal to 0 and less than 1, evenly distributed.
RANDBETWEEN	Returns a random number in the range between two numbers you specify.
ROUND	Rounds a number to the specified number of digits.
ROUNDDOWN	Rounds a number down, toward zero.
ROUNDUP	Rounds a number up, away from 0 (zero).

SIGN	Determines the sign of a number, the result of a calculation, or a value in a column.
SQRT	Returns the square root of a number.
SUM	Adds all the numbers in a column.
SUMX	Returns the sum of an expression evaluated for each row in a table.
TRUNC	Truncates a number to an integer by removing the decimal, or fractional, part of the number.

Statistical Functions

These functions carry out statistical and aggregation functions on data values in a DAX expression in Power BI. The list of available statistical functions is given below.

Function	Description
ADDCOLUMNS	Adds calculated columns to the given table or table expression.
APPROXIMATEDISTINCTCOUNT	Returns the <i>approximate</i> number of rows that contain distinct values in a column.
AVERAGE	Returns the average (arithmetic mean) of all the numbers in a column.
AVERAGEA	Returns the average (arithmetic mean) of the values in a column.
AVERAGEX	Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.
BETA.DIST	Returns the beta distribution.
BETA.INV	Returns the inverse of the beta cumulative probability density function (BETA.DIST).
CHISQ.DIST	Returns the chi-squared distribution.
CHISQ.DIST.RT	Returns the right-tailed probability of the chi-squared distribution.
CHISQ.INV	Returns the inverse of the left-tailed probability of the chi-squared distribution.
CHISQ.INV.RT	Returns the inverse of the right-tailed probability of the chi-squared distribution.
CONFIDENCE.NORM	The confidence interval is a range of values.
CONFIDENCE.T	Returns the confidence interval for a population mean, using a Student's t distribution.
COT	Returns the cotangent of an angle specified in radians.
COTH	Returns the hyperbolic cotangent of a hyperbolic angle.
COUNT	Counts the number of cells in a column that contain numbers.
COUNTA	Counts the number of cells in a column that are not empty.
COUNTAX	Counts nonblank results when evaluating the result of an expression over a table.
COUNTBLANK	Counts the number of blank cells in a column.
COUNTROWS	Counts the number of rows in the specified table, or in a table defined by an expression.
COUNTX	Counts the number of rows that contain a number or an expression that evaluates to a number, when evaluating an expression over a table.
CROSSJOIN	Returns a table that contains the Cartesian product of all rows from all tables in the arguments.
DATATABLE	Provides a mechanism for declaring an inline set of data values.
DISTINCTCOUNT	Counts the number of distinct values in a column.
DISTINCTCOUNTNOBLANK	Counts the number of distinct values in a column.
EXPON.DIST	Returns the exponential distribution.
GENERATE	Returns a table with the Cartesian product between each row in <i>table1</i> and the table that results from evaluating <i>table2</i> in the context of the current row from <i>table1</i> .
GENERATEALL	Returns a table with the Cartesian product between each row in <i>table1</i> and the table that results from evaluating <i>table2</i> in the context of the current row from <i>table1</i> .
GEOMEAN	Returns the geometric mean of the numbers in a column.
GEOMEANX	Returns the geometric mean of an expression evaluated for each row in a table.
MAX	Returns the largest numeric value in a column, or between two scalar expressions.
MAXA	Returns the largest value in a column.
MAXX	Evaluates an expression for each row of a table and returns the largest numeric value.
MEDIAN	Returns the median of numbers in a column.
MEDIANX	Returns the median number of an expression evaluated for each row in a table.

MIN	Returns the smallest numeric value in a column, or between two scalar expressions.
MINA	Returns the smallest value in a column, including any logical values and numbers represented as text.
MINX	Returns the smallest numeric value that results from evaluating an expression for each row of a table.
NORM.DIST	Returns the normal distribution for the specified mean and standard deviation.
NORM.INV	The inverse of the normal cumulative distribution for the specified mean and standard deviation.
NORM.S.DIST	Returns the standard normal distribution (has a mean of zero and a standard deviation of one).
NORM.S.INV	Returns the inverse of the standard normal cumulative distribution.
PERCENTILE.EXC	Returns the k-th percentile of values in a range, where k is in the range 0..1, exclusive.
PERCENTILE.INC	Returns the k-th percentile of values in a range, where k is in the range 0..1, inclusive.
PERCENTILEX.EXC	Returns the percentile number of an expression evaluated for each row in a table.
PERCENTILEX.INC	Returns the percentile number of an expression evaluated for each row in a table.
POISSON.DIST	Returns the Poisson distribution.
RANK.EQ	Returns the ranking of a number in a list of numbers.
RANKX	Returns the ranking of a number in a list of numbers for each row in the <i>table</i> argument.
ROW	Returns a table with a single row containing values that result from the expressions given to each column.
SAMPLE	Returns a sample of N rows from the specified table.
SELECTCOLUMNS	Adds calculated columns to the given table or table expression.
SIN	Returns the sine of the given angle.
SINH	Returns the hyperbolic sine of a number.
STDEV.P	Returns the standard deviation of the entire population.
STDEV.S	Returns the standard deviation of a sample population.
STDEVX.P	Returns the standard deviation of the entire population.
STDEVX.S	Returns the standard deviation of a sample population.
SQRTPI	Returns the square root of (number * pi).
SUMMARIZE	Returns a summary table for the requested totals over a set of groups.
T.DIST	Returns the Student's left-tailed t-distribution.
T.DIST.2T	Returns the two-tailed Student's t-distribution.
T.DIST.RT	Returns the right-tailed Student's t-distribution.
T.INV	Returns the left-tailed inverse of the Student's t-distribution.
T.INV.2t	Returns the two-tailed inverse of the Student's t-distribution.
TAN	Returns the tangent of the given angle.
TANH	Returns the hyperbolic tangent of a number.
TOPN	Returns the top N rows of the specified table.
VAR.P	Returns the variance of the entire population.
VAR.S	Returns the variance of a sample population.
VARX.P	Returns the variance of the entire population.
VARX.S	Returns the variance of a sample population.
XIRR	Returns the internal rate of return for a schedule of cash flows that is not necessarily periodic.
XNPV	Returns the present value for a schedule of cash flows that is not necessarily periodic.

We build Dashboards that can Become Cryptic for Outsiders otherwise.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Text Functions

The text functions in Power BI are very similar to the string functions of Excel. These functions evaluate string values.

Function	Description
BLANK	Returns a blank.
CODE	Returns a numeric code for the first character in a text string.
CONCATENATE	Joins two text strings into one text string.
CONCATENATEX	Concatenates the result of an expression evaluated for each row in a table.
CONTAINSSTRING	Returns TRUE or FALSE indicating whether one string contains another string.
CONTAINSSTRINGEXACT	Returns TRUE or FALSE indicating whether one string contains another string.
EXACT	Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise.
FIND	Returns the starting position of one text string within another text string.
FIXED	Rounds a number to the specified number of decimals and returns the result as text.
FORMAT	Converts a value to text according to the specified format.
LEFT	Returns the specified number of characters from the start of a text string.
LEN	Returns the number of characters in a text string.
LOWER	Converts all letters in a text string to lowercase.
MID	Returns a string of characters from the middle of a text string, given a starting position and length.
REPLACE	REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.
REPT	Repeats text a given number of times.
RIGHT	RIGHT returns the last character or characters in a text string, based on the number of characters you specify.
SEARCH	Returns the number of the character at which a specific character or text string is first found, reading left to right.
SUBSTITUTE	Replaces existing text with new text in a text string.
TRIM	Removes all spaces from text except for single spaces between words.
UNICHAR	Returns the Unicode character referenced by the numeric value.
UNICODE	Returns the numeric code corresponding to the first character of the text string.
UPPER	Converts a text string to all uppercase letters.
VALUE	Converts a text string that represents a number to a number.

Parent-Child functions

The parent and child functions are used for data values that are a part of a parent-child hierarchy.

Function	Description
PATH	Returns a delimited text string with the identifiers of all the parents of the current identifier.
PATHCONTAINS	Returns TRUE if the specified <i>item</i> exists within the specified <i>path</i> .
PATHITEM	Returns the item at the specified <i>position</i> from a string resulting from evaluation of a PATH function.
PATHITEMREVERSE	Returns the item at the specified <i>position</i> from a string resulting from evaluation of a PATH function.
PATHLENGTH	Returns the number of parents to the specified item in a given PATH result, including self.



Other functions

There are a bunch of functions that do not fit in any particular category. These are also very useful functions.

Function	Description
CONVERT	Converts an expression of one data type to another.
DATATABLE	Provides a mechanism for declaring an inline set of data values.
ERROR	Raises an error with an error message.
EXCEPT	Returns the rows of one table which do not appear in another table.
GENERATESERIES	Returns a single column table containing the values of an arithmetic series.
GROUPBY	Similar to the SUMMARIZE function, GROUPBY does not do an implicit CALCULATE for any extension columns that it adds.
INTERSECT	Returns the row intersection of two tables, retaining duplicates.
ISEMPTY	Checks if a table is empty.
ISSELECTEDMEASURE	Used by expressions for calculation items to determine the measure that is in context is one of those specified in a list of measures.
NATURALINNERJOIN	Performs an inner join of a table with another table.
NATURALLEFTOUTERJOIN	Performs an inner join of a table with another table.
SELECTEDMEASURE	Used by expressions for calculation items to reference the measure that is in context.
SELECTEDMEASUREFORMATSTRING	Used by expressions for calculation items to retrieve the format string of the measure that is in context.
SELECTEDMEASURENAME	Used by expressions for calculation items to determine the measure that is in context by name.
SUMMARIZECOLUMNS	Returns a summary table over a set of groups.
Table Constructor (DAX)	Returns a table of one or more columns.
TREATAS	Applies the result of a table expression as filters to columns from an unrelated table.
UNION	Creates a union (join) table from a pair of tables.

DAX CONTEXT

The concept of DAX context is essential in gaining a complete understanding of DAX language and how it works in Power BI. There are two types of DAX contexts; Row context and Filter context.

Row context

Row context means taking into account a specific row that has been filtered in a DAX expression. The row context focuses on the operation of the formula on the current row. This type of context is most often applied to the measure rows.

Filter context

The filter context takes focusing on values one step ahead. In row context, we were filtering out and applying operations on specific rows. But in filter context, the expression applies a filter to focus on specific values within a row. Thus, the filter context is applied in addition to the row context to narrow down the scope of calculation to specific values. Filter context is applied when we use functions like CALCULATE, FILTER, RELATED, ALL, etc.

We use an ideal syntax language DAX that possesses a great capability to manage and manipulate data.

Cognitive Convergence

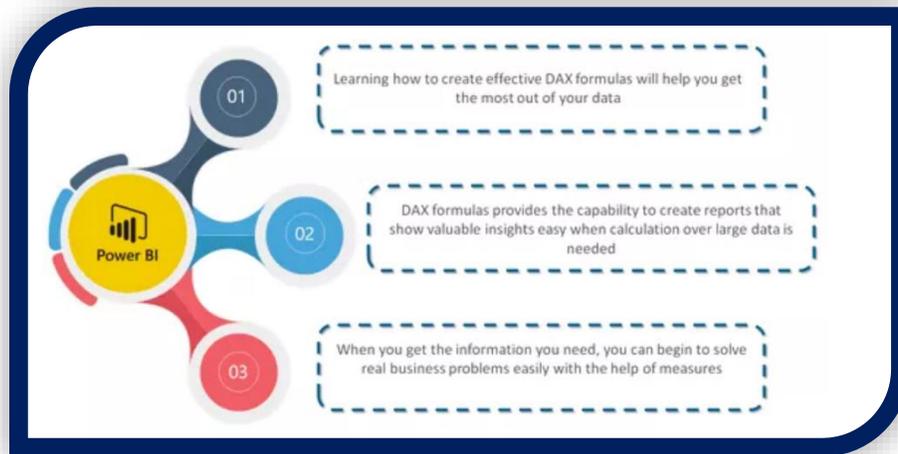
<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

DAX STATEMENTS

Statement	Description
DEFINE	(Keyword) Defines entities that exist only for the duration of a DAX query.
EVALUATE	(Keyword) A statement required to execute a DAX query.
ORDER BY	(Keyword) Defines one or more expressions used to sort results of a DAX query.
VAR	(Keyword) Stores the result of an expression as a named variable, which can then be passed as an argument to other measure expressions.



POWER QUERY COMPUTED COLUMNS

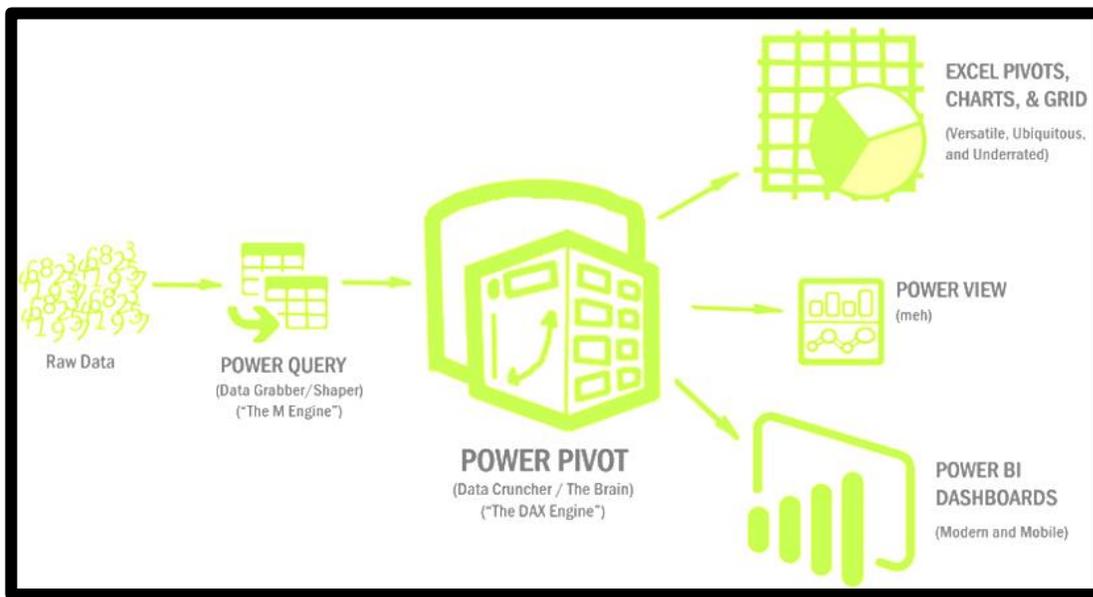
A Power Query computed column is an M expression that defines the transformations to apply to a data source before importing a table in the data model. Depending on the expression and on the data source, the M expression of a Power Query computed column can be translated into an equivalent request to the data source – like an expression of a column in a SQL query; it can also be computed by the Power Query engine, modifying the rows coming from the data source that are used to populate the table in the data model.

COMPARISSON BETWEEN DAX CALCULATED COLUMNS AND POWER QUERY COMPUTED COLUMNS

This is a list of the most important differences between DAX calculated columns and Power Query computed columns considering specific scenarios.

Features	DAX	Power Query
Adding a column to the model	A new DAX calculated column does not require a full refresh of the table.	A new Power Query computed column requires a full refresh of the table.
Full refresh of the table	The engine processes DAX calculated columns after the data of all the partitions has been loaded and compressed in memory.	A Power Query computed column is processed like any other column coming from the data source.

Incremental refresh and partition refresh	The engine processes the DAX calculated columns for all the rows of the table, even when only a few rows are added to or refreshed in the table.	The engine only computes values for the Power Query computed columns of the rows that are processed in any partial refresh.
Comparing compression	A DAX calculated column does not participate in the evaluation of the best sort order for the compression. As a result, the compression of the column might be far from ideal, especially for columns with a low number of unique values.	A Power Query computed column is compressed like any other column of the table.
Processing Time	The cost of processing a single DAX calculated column corresponds to a sequential evaluation of the DAX expression for each row included in a refresh operation. Multiple DAX calculated columns within the same table are processed sequentially, one after the other, with an order compatible with calculation dependencies.	The cost for processing a Power Query calculated column mostly depends on the data source when the expression is optimized using query folding. For example, if M expressions are translated into equivalent SQL expressions, performance and parallelism entirely depend on the SQL data source.
Query performance	Query performance of DAX calculated column is low.	Query performance of Power Query calculated column is low.



CHALLENGES AND SOLUTIONS

DAX is a very strong formula language and there are number of advantages in learning and using DAX while making dashboard i.e. Speed Up Your Dashboard. But there are also some challenges that can be faced because of DAX.

Challenges	Solutions
You Can Do a Lot With Alternatives	The 80/20 Rule can be used, 80% of dashboard can be built without DAX, 20% of DAX require to make an effective speedy dashboard.
Dashboards Can Become Cryptic for Outsiders	



CONCLUSION

In this paper we discuss Data Analysis Expression (DAX) which is a formula language, its use, functions and calculation types, formula and syntax of DAX is explained. DAX context, statements, and queries, keywords are also explained, Moreover, DAX is compared with Power Query language which is another formula language of Power BI.

Cognitive Convergence will provide consulting services that help in designing, deploying, managing, enhancing or troubleshooting on-premises, cloud-based or hybrid Power BI environment. Cognitive Convergence will provide help to start fresh with Power BI to modernize current business analytics solution or revamp existing Power BI deployment by incorporating new data sources or adding new services.

