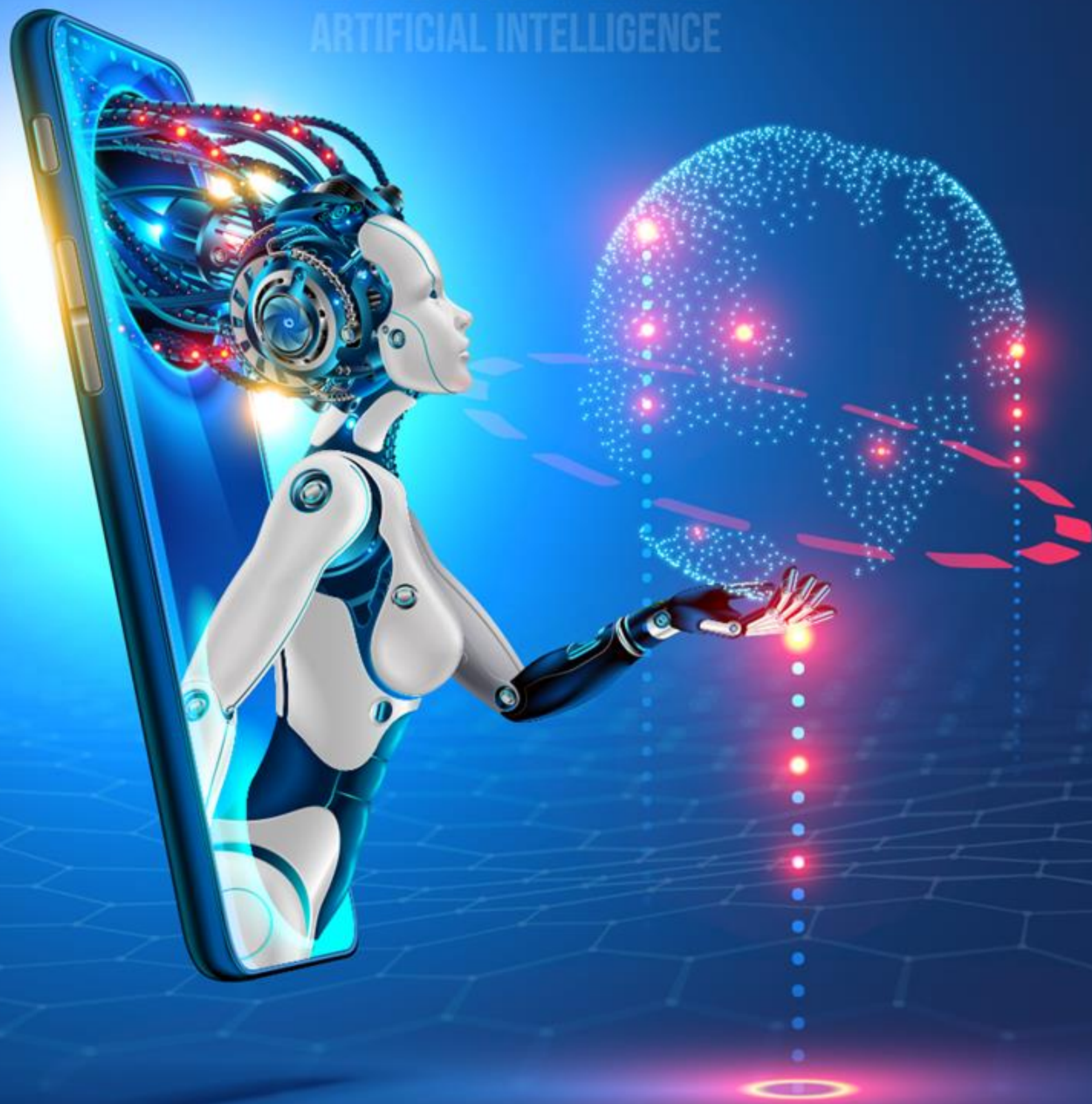


# DATA AND ARTIFICIAL INTELLIGENCE



## Data Manipulation and Reporting with Power BI

# DATA AND ARTIFICIAL INTELLIGENCE



**Using DAX**



# Learning Objectives

By the end of this lesson, you will be able to:

- 🕒 Get an overview of DAX functions
- 🕒 Learn to use the different types of DAX calculations
- 🕒 Learn how to use DAX functions to create new columns and measures



## Overview of DAX Functions

# Introduction to DAX Functions

- Data Analysis Expressions or DAX is a functional language which empowers you to modify or transform data in the data modeling tables by performing various actions
- Simple yet very powerful concept
- Numeric and Other are the two major data types in DAX
- The DAX functions are configured in such a way that if you work on any one type of number, the function is applied to other numeric values too

## Numeric data types can include:

- Integers
- Decimals
- Currency

## Other data types can include:

- Strings
- Binary objects

# Types of DAX Calculations

**Calculated Columns**

**Calculated Measures**



# Calculated Columns

When you want to apply calculation on every row in your table, you need the following elements in the column:

**Column name**



**At least one function or expression**

# Calculated Measures

When you want to calculate the percentages or ratios, or you need complex aggregations to be created in the column. To accomplish this, you need:

**Measure name**



**At least one function  
or expression**

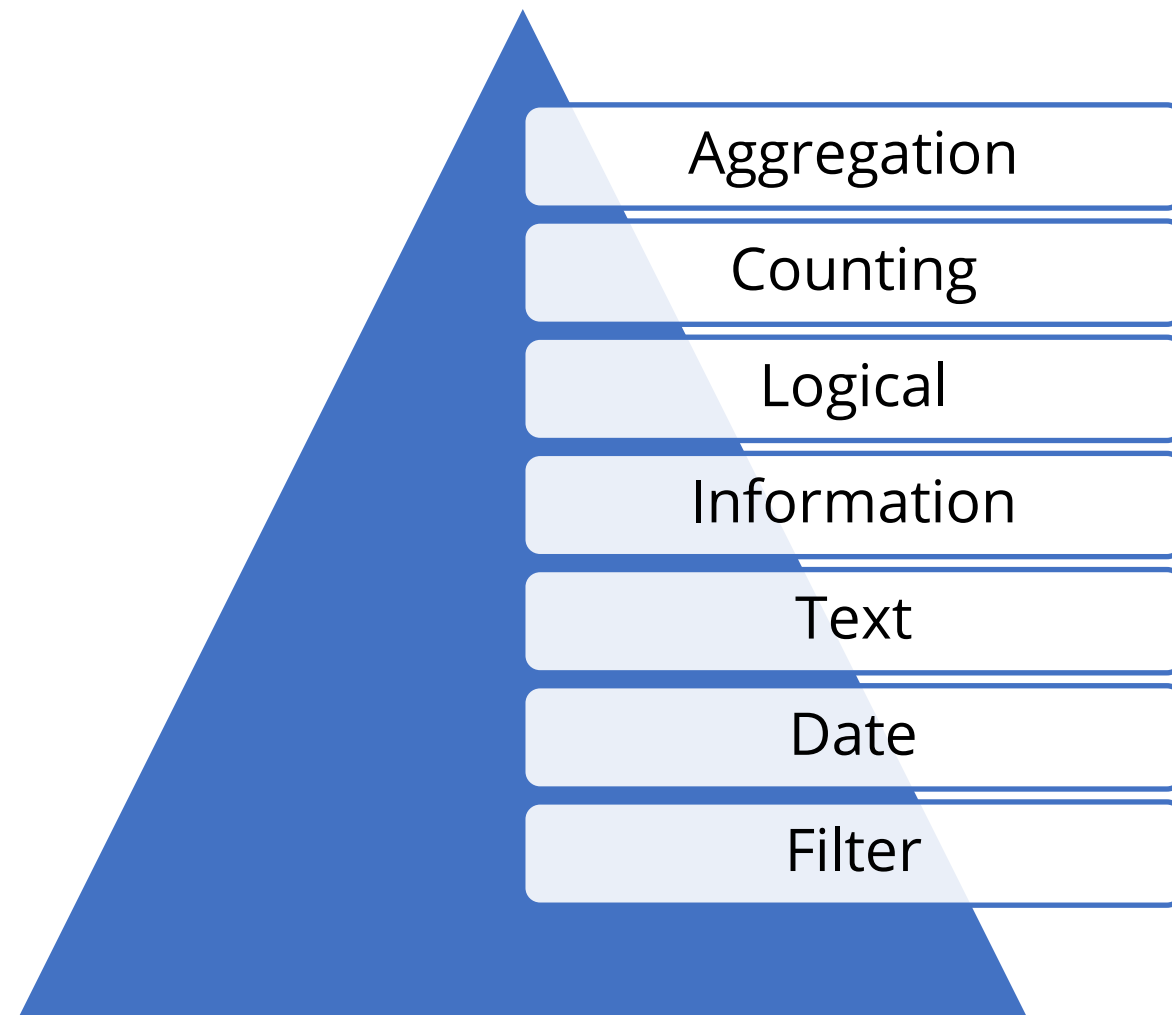
Example: `ColumnName = Function(TableName[ColumnName])`



## Using DAX Functions

# Using DAX Functions

The different types of DAX functions which we are going to learn are:



# Using DAX: Aggregation Functions

Aggregation functions work only on numeric columns. They can aggregate one column at a time. The following are the commonly used DAX aggregation functions:

SUM

MeasureName = SUM(TableName[ColumnName])

AVERAGE

MeasureName = AVERAGE(TableName[ColumnName])

MIN

MeasureName = MIN(TableName[ColumnName])

MAX

MeasureName = MAX(TableName[ColumnName])

SUMX (and other X functions)

MeasureName = SUMX(TableName,[ColumnName1]\*[ColumnName1])

# Using DAX: Counting Functions

Counting functions count distinct values, table rows, and non-empty values and provide the count of elements. The following are the commonly used DAX counting functions:

COUNT

MeasureName = COUNT(TableName[ColumnName])

COUNTA

MeasureName = COUNTA(TableName[ColumnName])

COUNTBLANK

MeasureName = COUNTBLANK(TableName[ColumnName])

COUNTROWS

MeasureName = COUNTROWS(TableName)

DISTINCTCOUNT

MeasureName = DISTINCTCOUNT(TableName[ColumnName])

# Using DAX: Logical Functions

Logical functions return logical values such as (TRUE/FALSE) based on the logical operations deployed on the relevant parameters. The following are the commonly used DAX logical functions:

AND

= AND([Country] = "USA",[Medal] = "Gold")

OR

= OR([Country] = "USA",[Medal] = "Gold")

NOT

= NOT([Country] = "USA")

IF

= IF([Country]="USA","American","Other")

IFERROR

= IFERROR (5/0,"Div by zero")



# Using DAX: Information Functions

Information functions evaluate the value or column that is provided as an argument and confirm whether the values match the expected type. The following are the commonly used DAX information functions:

ISBLANK

= ISBLANK('TableName'[ColumnName])

ISNUMBER

= ISNUMBER('TableName'[ColumnName])

ISTEXT

= ISTEXT('TableName'[ColumnName])

ISNONTEXT

= ISNONTEXT('TableName'[ColumnName])

ISERROR

= ISERROR(Calculation)

# Using DAX: Text Functions

Text functions are used to return a part of a string, concatenate string values, or search for text within a string. You can use it to control the numbers, time, and date formats. The following are the commonly used DAX text functions:

CONCATENATE

= CONCATENATE([ColumnName],"Value")

REPLACE

= REPLACE('New Products'[Product Code],1,2,"OB")

SEARCH

= SEARCH("com",[Subcategory],1,0)

UPPER

= UPPER([ColumnName])

FIXED

FIXED([Revenue Forecast],2,0)

# Using DAX: Date Functions

Date functions are used to extract relevant business information from the date values after applying some calculations. The following are the commonly used DAX date functions:

DATE

= DATE(2017,10,28)

HOUR

= HOUR([ColumnName])

NOW

= NOW()

EOMONTH

= EOMONTH("3/10/2017",1)

WEEKDAY

= WEEKDAY("3/10/2017",1)

## Using DAX: Filter Functions

- The filter functions are the most powerful DAX functions.
- The lookup functions operate using tables and relationships, like a database.
- To create dynamic calculations, the filtering functions let you manipulate data context.

CALCULATE

Books Revenue = CALCULATE(SUM([Revenue]),'Category Sales Report'[Category]="Books")

RELATED

Cost = RELATED('Subcategory Master'[Cost])

FILTER

High Revenue = FILTER('Category Sales Report','Category Sales Report'[Revenue]>10000)

## Creating Measures Using DAX



# What Are Measures?

1

Measures are most often created automatically by the system, when a column is included in a visual.

2

Measures help us to perform calculations on the values of a numeric column. Measures can be used in various visuals.

3

Power BI supports various aggregation functions to create a new Measure (Sum, Average, Count etc.)

# Creating Measures

- To create a new measure, click on the “New Measure” button from the “Modeling” tab.
- In the formulae bar, type the name of the measure you want to create and use a numeric column along with an aggregation function to create the measure.
- The screenshot below shows an example of creating a “Revenue” measure by applying a “Sum” aggregation function on the “Sales” column.

The screenshot displays the Power BI interface with the 'Modeling' tab selected. A red box highlights the 'New Measure' button in the ribbon, labeled with a blue circle containing the number 1. Another red box highlights the formula bar, labeled with a blue circle containing the number 2, showing the formula 'Revenue = SUM([Sales Value])'. A third red box highlights the 'Revenue' measure in the 'Fields' pane, labeled with a blue circle containing the number 3. The 'Fields' pane also shows other measures like 'Month ID', 'Profit', 'Sales Value', and 'Subcategory ID'. The background shows a data table with columns: Month ID, Month Description, Subcategory ID, Subcategory Description, Profit, and Sales Value.

Month ID	Month Description	Subcategory ID	Subcategory Description	Profit	Sales Value
201001	Friday, January 01, 2010	11	Art & Architecture	1762.514	6916
201001	Friday, January 01, 2010	12	Business	1410.222	5738
201001	Friday, January 01, 2010	13	Literature	937.304	4273
201001	Friday, January 01, 2010	14	Books - Miscellaneous	910.910999999999	4718
201001	Friday, January 01, 2010	15	Science & Technology	2705.05	10767
201001	Friday, January 01, 2010	16	Sports & Health	1163.743	4749
201001	Friday, January 01, 2010	21	Audio Equipment	121.1558	53300
201001	Friday, January 01, 2010	22	Cameras	50.7978	69310
201001	Friday, January 01, 2010	23	Computers	6530.124	32410
201001	Friday, January 01, 2010	24	Electronics - Miscellaneous	13764.624	69878
201001	Friday, January 01, 2010	25	TV's	10975.658	54325
201001	Friday, January 01, 2010	26	Video Equipment	15416.9776	75110

# Creating Measures

- Measures can be created out of one column or multiple columns. In this case, individual aggregation function needs to be applied to each column name.
- Similarly, a measure can be created based on the other measures as well. In this case, just including the measure names into the calculation is enough.

Manage Relationships		New Measure	New Column	New Table	Sort by Column	\$ % ' .00 Auto	Default Summarization: Don't summarize	
Relationships		Calculations		Sort		Formatting		Properties
		Profit Margin = SUM([Profit Earned])/SUM([Sales Value])						
Month ID	Month Description	Subcategory ID	Subcategory Description	Profit Earned	Sales Value			
201001	Friday, January 01, 2010	11	Art & Architecture	1762.514	6916			
201001	Friday, January 01, 2010	12	Business	1410.222	5738			
201001	Friday, January 01, 2010	13	Literature	937.304	4273			

		Profit Margin = [Revenue]/[Profit]						
Month ID	Month Description	Subcategory ID	Subcategory Description	Profit Earned	Sales Value			
201001	Friday, January 01, 2010	11	Art & Architecture	1762.514	6916			
201001	Friday, January 01, 2010	12	Business	1410.222	5738			
201001	Friday, January 01, 2010	13	Literature	937.304	4273			
201001	Friday, January 01, 2010	14	Books - Miscellaneous	910.910999999999	4718			
201001	Friday, January 01, 2010	15	Science & Technology	2705.05	10767			
201001	Friday, January 01, 2010	16	Sports & Health	1163.743	4749			
201001	Friday, January 01, 2010	21	Audio Equipment	10321.1558	53300			
201001	Friday, January 01, 2010	22	Cameras	14160.7978	69310			
201001	Friday, January 01, 2010	23	Computers	6530.124	32410			
201001	Friday, January 01, 2010	24	Electronics - Miscellaneous	13764.624	69878			
201001	Friday, January 01, 2010	25	TV's	10975.658	54325			

### Fields

Search

Subcategory

Month Description

Σ Month ID

Profit

Σ Profit Earned

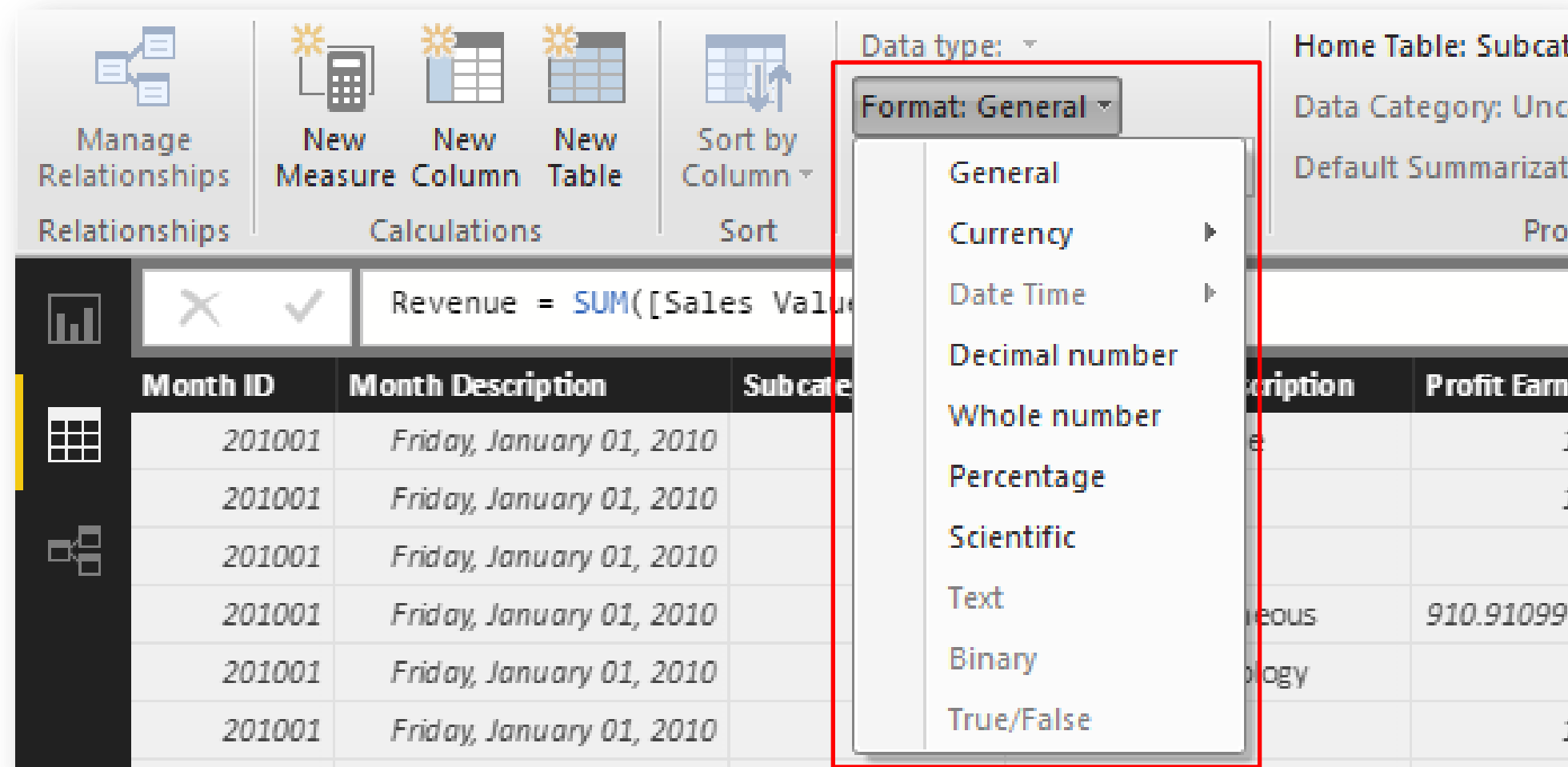
Profit Margin

Revenue

Σ Sales Value

# Creating Measures

After creating a new measure, you can format the object according to your requirement. The following screenshots show the formatting types supported by Power BI:





**Objective:** To perform various operations in the Power BI Desktop using DAX functions.

**Access:** To execute the practice, follow these steps:

- For SUM, we can use only one table, as shown below:

Start Date	End Date	Member	Personal Loan Balance
9/17/2019	11/20/2019	301	540000
9/18/2019	11/28/2019	301	650000
9/18/2019	11/28/2019	302	450000
9/19/2019	11/30/2019	303	760000
9/21/2019	12/17/2019	304	780000
9/27/2019	12/27/2019	305	940000



# DAX Functions



**Objective:** To perform various operations in the Power BI Desktop using DAX functions.

**Access:** To execute the practice, follow these steps:

Step 1: Import data to Power BI.

Step 2: Right-click on the table and select **add column** or add measure. Let us now add a column called **Total** that will use the function SUM and calculate the total loan for each member.

Step 3: Below is the function to be used:

✕ ✓

1 Total = CALCULATE(SUM('Personal Loan'[Personal Loan Balance]),ALL('Personal Loan'))

**DATEDIF Function:** This function calculates the difference between two dates.

Step 1: Create a new column and enter the DAX function as shown below:

✕ ✓

1 DIFF = DATEDIFF('Personal Loan'[Start Date],'Personal Loan'[End Date],DAY)

ASSISTED PRACTICE

# DAX Functions



**Objective:** To perform various operations in the Power BI Desktop using DAX functions.

**Access:** To execute the practice, follow these steps:

**FILTER Function:** This function can be used to filter values from a table. It creates a table which is a subset of another table.

Step 1: After importing the data, right-click on the table and click on **Create column**. Below is the example of a DAX filter function:

```
1 filterColumn = CALCULATE(SUM('Personal Loan'[Personal Loan Balance]), FILTER('Personal Loan', 'Personal Loan'[Member]=301))
```

**Logical DAX - IF Function:** This function is used to make logical decisions in a table.

Step 1: Create a new column, say **Loan Type** that specifies the type of loan as high or low. Below is the syntax for the same:

```
1 Loan Type = IF('Personal Loan'[DIFF] > 80, "HIGH LOAN", "LOW LOAN")
```

# DAX Functions



**Objective:** To perform various operations in the Power BI Desktop using DAX functions.

**Access:** To execute the practice, follow these steps:

**GROUPBY Function:** This function is used to reuse the data. It uses a function called CURRENTGROUP() which is used inside aggregation function.

Step 1: After importing the data, from the top menu bar, click on **Modeling**.

Step 2: Click on **New table**.

Step 3: Create a new table that uses the GROUPBY function to calculate the total loan for each member. Below is the syntax for the same:

```
Loan = GROUPBY('Personal Loan','Personal Loan'[Member],"Total Loan",SUMX(CURRENTGROUP(),'Personal Loan'[Personal Loan Balance]))
```

ASSISTED PRACTICE

# Creating Measures



**Objective:** To create measures on different datasets in Power BI.

**Access:** To execute the practice, follow these steps:

We will be creating measures in the same datasets which we used earlier Car Loan and House Loan. One without relation and one with relation.

Creating measures with only one dataset (without relation):

Step 1: Load the dataset to Power BI.

Step 2: From the top menu bar, click on the **New Measure** button to create a measure.

Step 3: A measure bar will appear as shown below. You have to type the formula or measure you want to calculate.



ASSISTED PRACTICE

# Creating Measures



**Objective:** To create measures on different datasets in Power BI.

**Access:** To execute the practice, follow these steps:

We will be creating measures in the same datasets which we used earlier Car Loan and House Loan. One without relation and one with relation.

Creating measures with only one dataset (without relation):

Considering the dataset being very simple, let us create a measure for calculating the total of all the car loan amount. Below is the function to be used:

Total = SUM(Sheet\_name[Column\_name])

Below is the measure created for the dataset used:

```
1 Total =SUM(Sheet1[Car Loan Balance])
```

**Step 4:** On the left panel, click on the model view to verify the creation of the measure. It will show the measure created named **Total**.



# Creating Measures



**Objective:** To create measures on different datasets in Power BI.

**Access:** To execute the practice, follow these steps:

Creating measures with only one dataset (without relation):

Considering the dataset being very simple, let us create a measure for calculating the total of all the car loan amount. Below is the function to be used:

Total = SUM(Sheet\_name[Column\_name])

Below is the measure created for the dataset used:

```
1 Total =SUM(Sheet1[Car Loan Balance])
```

**Step 4:** On the left panel, click on the model view to verify the creation of the measure. It will show the measure created named **Total**.

**Step 5:** This measure will calculate the total. To verify, create a card for the field total. It will show the total of all the loan amounts.

ASSISTED PRACTICE

# Creating Measures



**Objective:** To create measures on different datasets in Power BI.

**Access:** To execute the practice, follow these steps:

Creating measures with two dataset (with relation):

**Step 1:** Import the datasets in Power BI.

**Step 2:** Click on “New Measure”. Now, let us create a measure for calculating the total loan for each person, which includes the addition of car loan and house loan. Below will be the DAX function for the same:

```
1 Total = SUMX('Car Loan', 'Car Loan'[Car Loan Balance] + RELATED('House Loan'[House Loan Balance]))
```

This will create a measure that will calculate the total loan for each person with relationships in both datasets.

**Step 3:** Click on the model view to see the measure created and view the relationship.



## Knowledge Check

## Knowledge Check

1

The COUNTA function counts the number of cells in a column that are not empty.

- a. True
- b. False



## Knowledge Check

1

The COUNTA function counts the number of cells in a column that are not empty.

- a. True
- b. False



The correct answer is **a.**

**Yes, the COUNTA function counts the number of cells in a column that are not empty.**

## Knowledge Check

2

Is it possible to derive the month or year from a data column using a DAX function?

- a. Yes
- b. No





## Knowledge Check

2

Is it possible to derive the month or year from a data column using a DAX function?

- a. Yes
- b. No



The correct answer is **a.**

**Yes, it is possible to derive the month or year from a data column using a DAX function.**

## Knowledge Check

3

If multiple DAX functions are nested inside each other, which level would be resolved first?

- a. Innermost
- b. Outermost



**Knowledge  
Check**

**3**

If multiple DAX functions are nested inside each other, which level would be resolved first?

- a. Innermost
- b. Outermost



The correct answer is **a.**

**If multiple DAX functions are nested inside each other, the innermost level would be resolved first.**

# Key Takeaways

You are now able to:

- 🕒 Get an overview of DAX functions
- 🕒 Learn to use the different types of DAX calculations
- 🕒 Learn how to use DAX functions to create new columns and measures

