IntelliPaat

Microsoft Power BI
Certification Training (DA-100)

Module 4 – Data Modelling

Agenda

01 What are Relationships?

02 Creating Relationships

03 Cardinality

04 Cross Filter Direction

05 What is DAX?

06 Why DAX?

Agenda

07 DAX Syntax

08 DAX Functions

09 Context in DAX

10 Calculated Columns

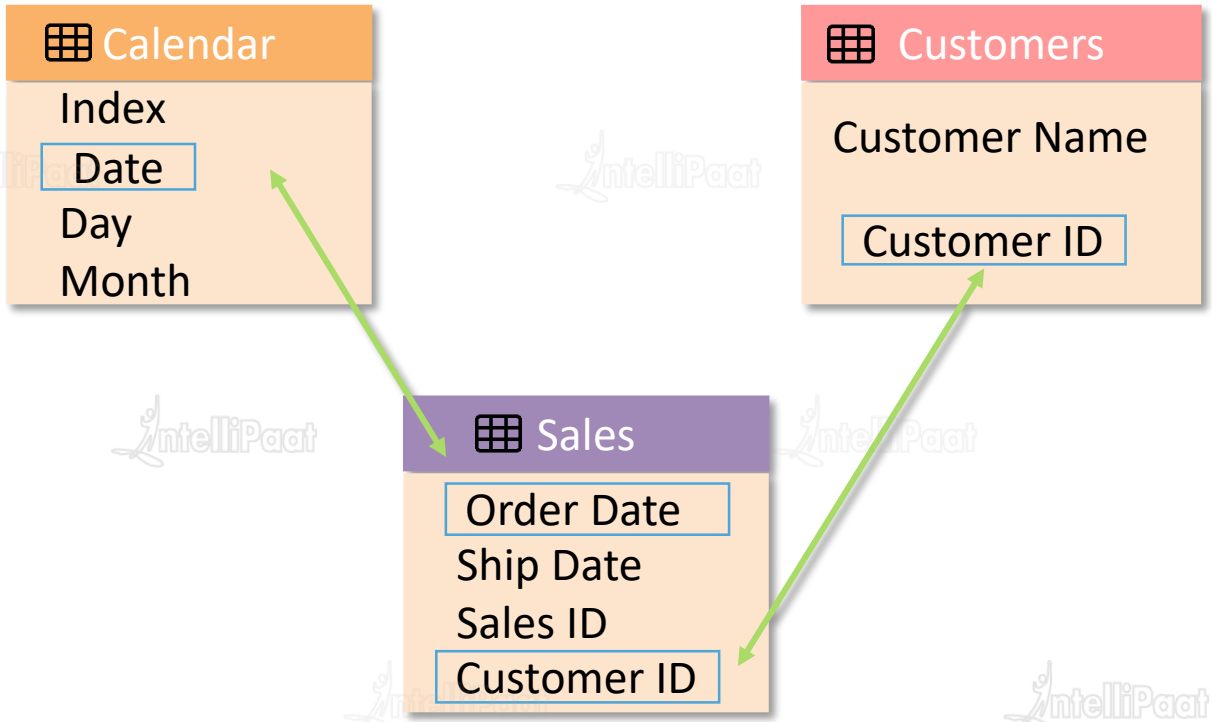
11 Calculated Tables

12 Measures

What are Relationships?

What are Relationships?

Relationships are used to join tables together so that we can work with them as if they are one



Relationships in Power BI are much similar to the relationships we create in relational databases, such as Microsoft SQL Server®, or data warehouse databases, such as SQL Server Analysis Services (SSAS)

What are Relationships?

Relationships in a relational database/OLTP, a data warehouse or Power BI are a part of normalization

Benefits of using relationships in a table are:

Benefits

Reduces Redundant data

It refines table structures and minimizes redundant data

Establishes a Connection

It establishes a connection between a pair of tables that are logically related to each other

Creating Relationships

Creating Relationships

There are two ways of creating a relationship in Power BI:

Creating Relationships Using Autodetect

When data is imported into the model, Power BI automatically creates relationships. If we then create calculated tables or use Enter Data to add new tables, relationships will not exist. However, Autodetect is the best guess, and it might need adjusting after it runs

Creating Relationships Manually

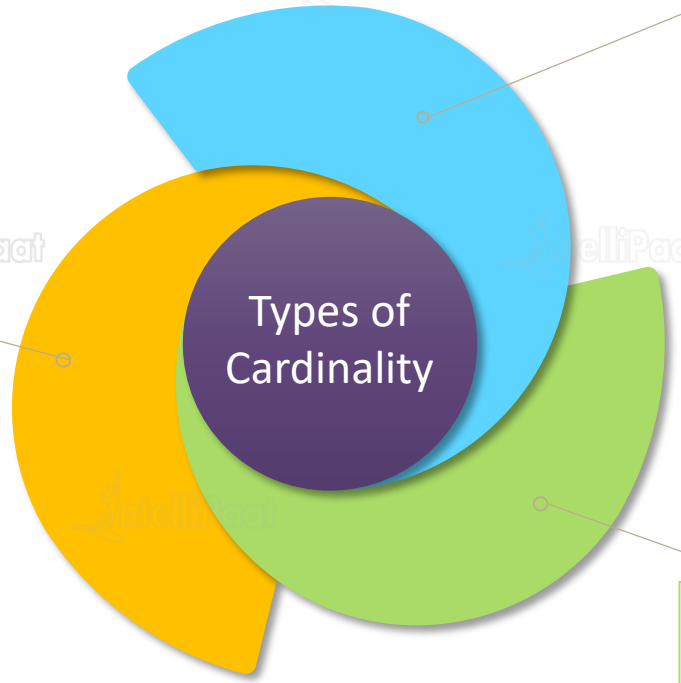
The easiest way to create a relationship between two tables is to drag a column from the first table to the related column in the second table where we want to join. If the data is valid for creating a relationship, the columns will be connected

Cardinality

Cardinality

In data modeling, cardinality refers to the relationship that one table has with another table

There are three types of cardinality in Power BI data modeling:



Many to One (*:1)

Many to one means that one table can have more than one instance of the value used in the column to join to the other table. The other table would have only one value. This is the most commonly used type in Power BI

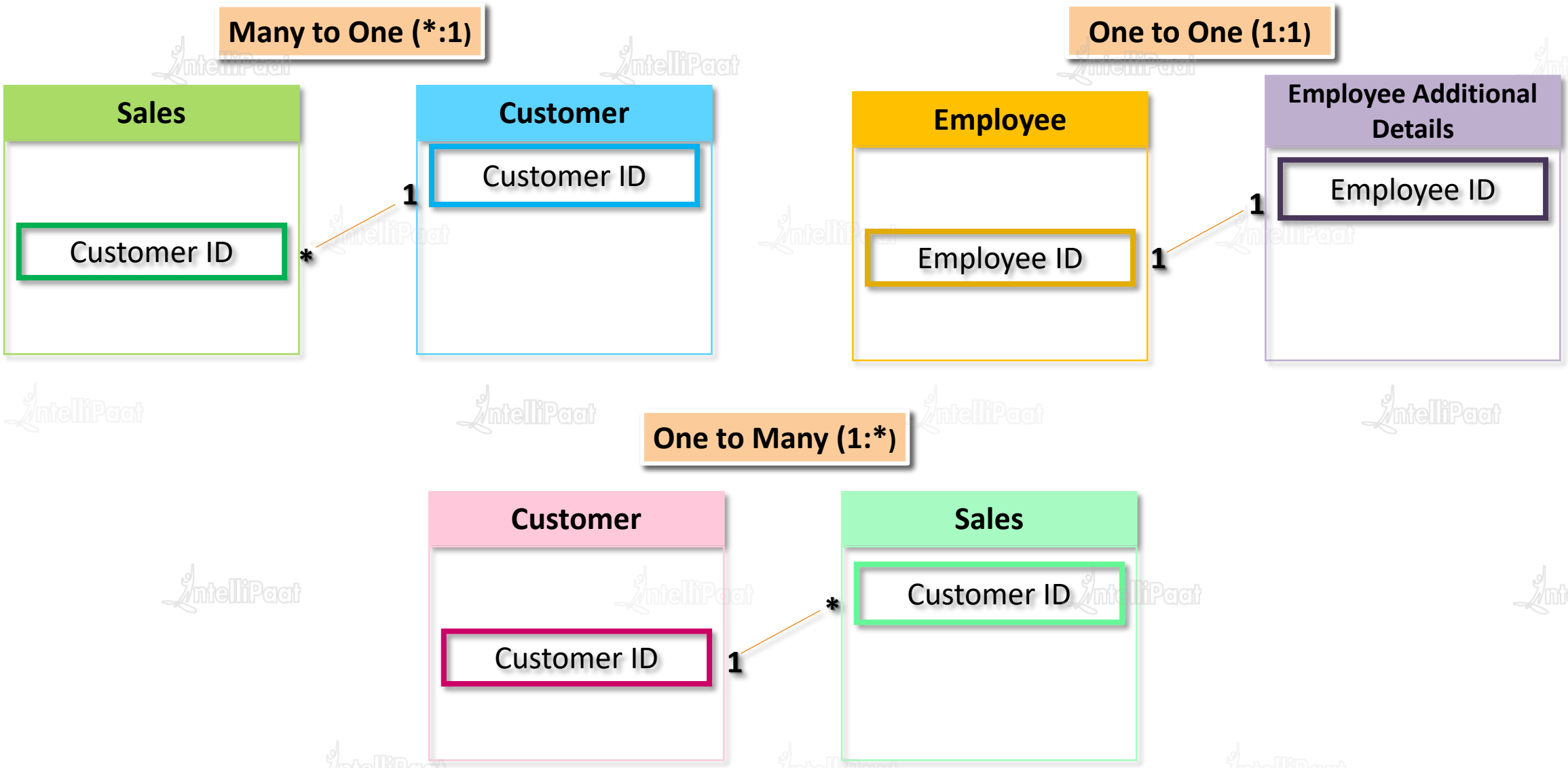
One to Many (1:*)

This is the same as many to one, except that the position of the tables is reversed in this relationship type

One to One (1:1)

In a one-to-one relationship, both tables in the relationship have one instance of the value

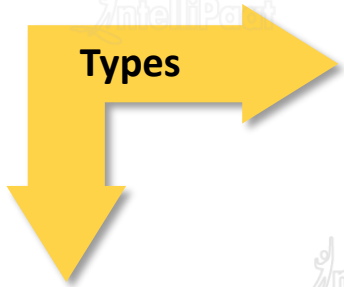
Cardinality



Cross Filter Direction

Cross Filter Direction

The cross filter direction of the relationships in our dataset decides how Power BI has to treat the tables in visualizations in our reports



Both

Both is the most common and the default. When we apply filtering, the two tables are considered as one for aggregating the data in a visualization

Single

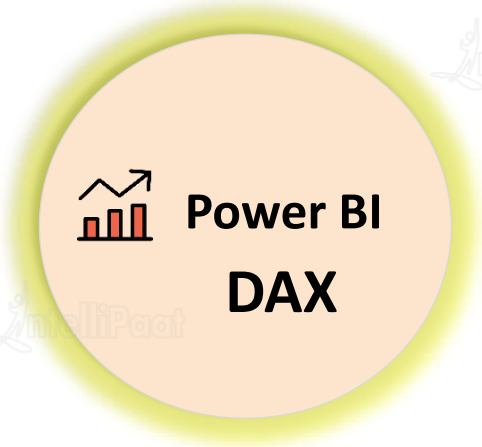
When we apply a single cross filter direction, the filters in related tables operate on the table where the values are aggregated

When we manually create a relationship, or the Autodetect feature generates the relationship for us, Power BI makes the best guess at the cross filter direction

Demo: Working with Relationships

What is DAX?

What is DAX?



Data Analysis Expressions (DAX) is a formula language that consists of a library of more than 200 functions, constants, and operators

DAX is used in a formula or expression to calculate and return a single value or multiple values

DAX is not a new feature. It is used in Power Pivot for Excel or SQL Server Analysis Services (SSAS)

Why DAX?

Why DAX?

01

Using DAX in a effective way will help us get the most of our data

02

Measures created using DAX help us solve real-world problems easily

03

DAX helps us perform calculations over large data easily to get valuable insights to make it more useful

Why DAX?

DAX helps us find the insights that we want to extract from our data to make it more useful

For example, suppose we want to compare the sales of this year so far, like-for-like with the last year. If the current month is May, we only want to compare till May of the previous year

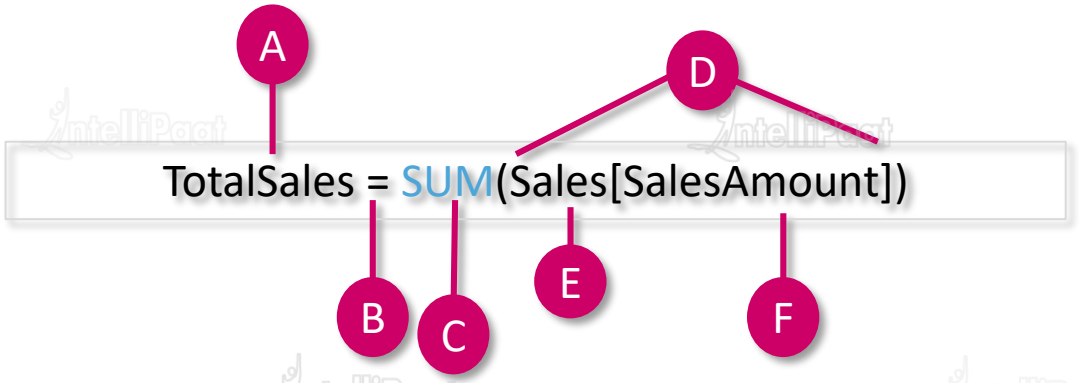
Last Year Sales = CALCULATE ([Total Sales], **SAMEPERIODLASTYEAR**('Date'[Date]))

DAX Syntax

DAX Syntax

The key for understanding and using DAX is learning the syntax for structuring formulas, the functions that we use to make calculations, and the context

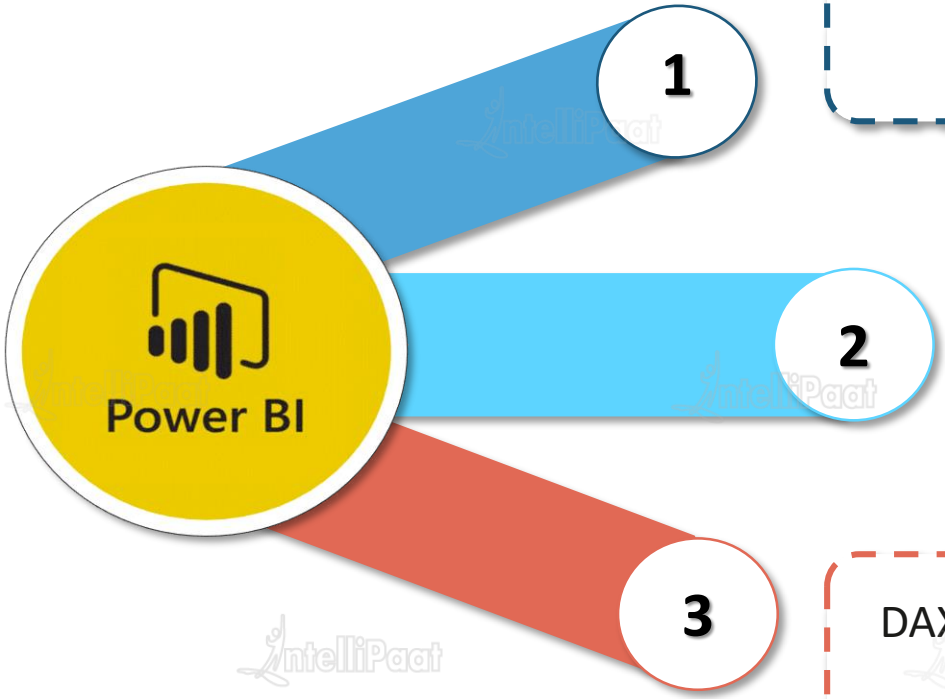
DAX formula syntax includes various elements that make up a formula



A. The measure name
B. The equal sign operator, which indicates the beginning of the formula
C. The DAX function, SUM
D. Parentheses(), which surround an expression that contains one or more arguments
E. The referenced table (Sales)
F. The referenced column (SalesAmount) in the referenced table

DAX Functions

DAX Functions



DAX functions are predefined formulas that perform calculations on one or more arguments

We can pass a column, function, expressions, formulas, constants, numbers, text, and TRUE or FALSE as an argument

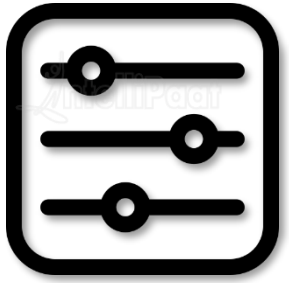
DAX is similar to Excel, but DAX refers to an entire column or a table. We can filter to refer to a particular value

DAX Functions

The DAX library of more than 200 functions, operators, and constructs is segmented into the following 10 categories:

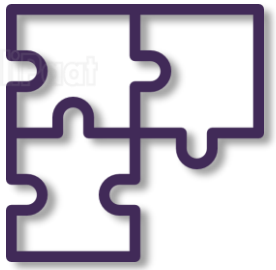
Date and Time

Time Intelligence

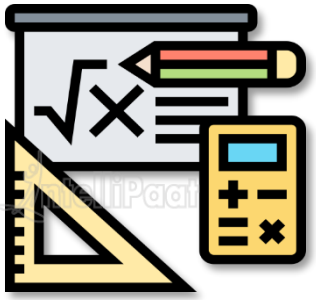


Filter

Information



Logical



Math and Trig

Parent-Child

Statistical

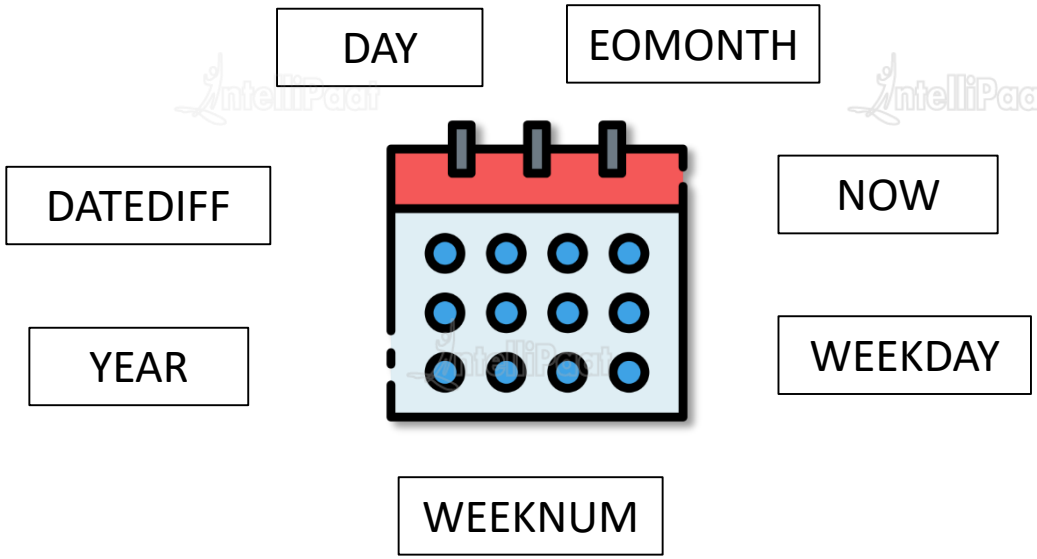
Text

Others

DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

The Date and Time DAX function is similar to the date and time functions used in Excel, but it is based on the datetime data types used by Microsoft SQL Server



DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

Using Time Intelligence functions, we can create date and time ranges combined with aggregations, and it is useful for building comparisons across time periods

NEXTYEAR

NEXTQUARTER

TOTALYTD

DATEADD

PREVIOUSMONTH

CLOSINGBALANCEMONTH

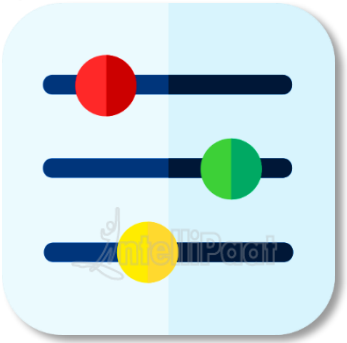
DAX Functions

- Date and Time
- Time Intelligence
- **Filter**
- Information
- Logical
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

Using Filter functions, we can get specific data types, look up values in related tables, or filter by related values

RELATED

CALCULATE



ISFILTERED

FILTER

VALUES

RELATEDTABLE

DAX Functions

- Date and Time
- Time Intelligence
- Filter
- **Information**
- Logical
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

Information functions evaluate a table or a column provided as an argument to another function and inform us if the value matches the expected type

ISTEXT

ISBLANK

ISEVEN

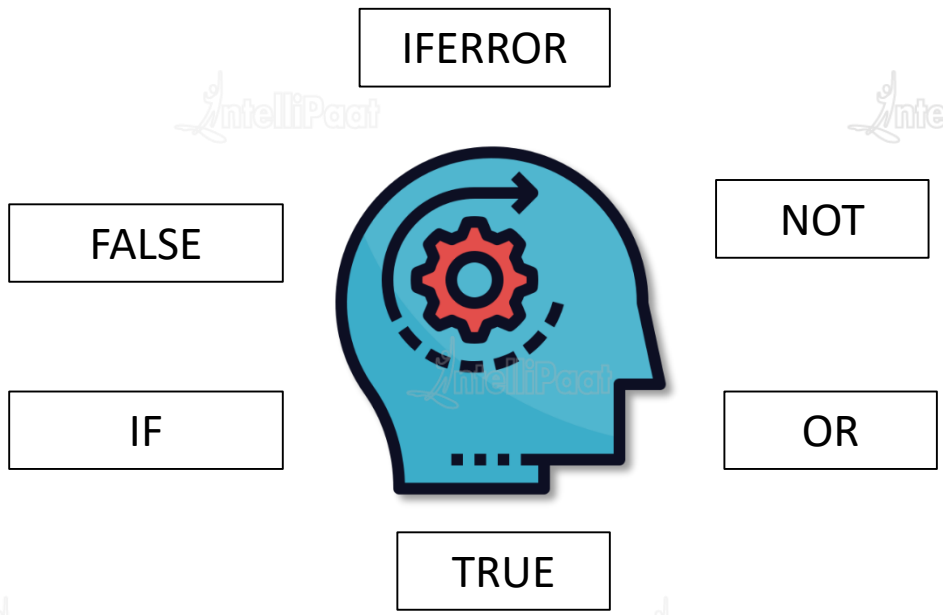
ISERROR

USERNAME

DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- **Logical**
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

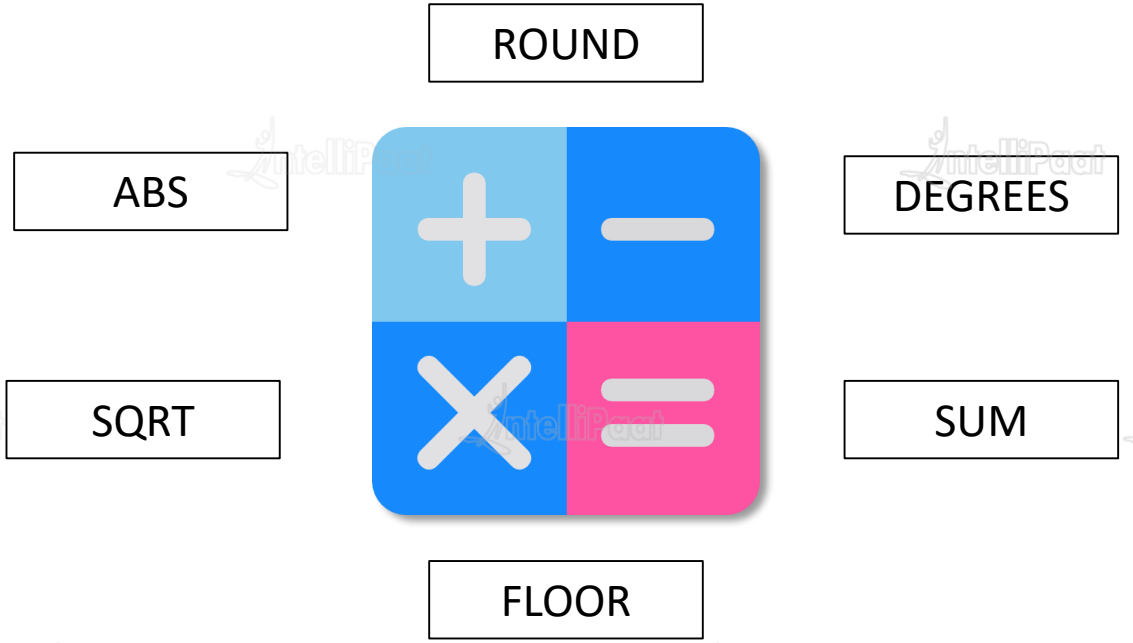
The Logical function returns information about the value in our expression



DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- **Math and Trig**
- Parent-Child
- Statistical
- Text
- Others

The Math and Trig functions perform a wide variety of mathematical calculations, and they are similar to the mathematical and trigonometrical functions in Excel



DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- **Parent-Child**
- Statistical
- Text
- Others

The Parent-Child functions work on the data that is presented in a parent/child hierarchy in the data model

PATHITEM

PATH

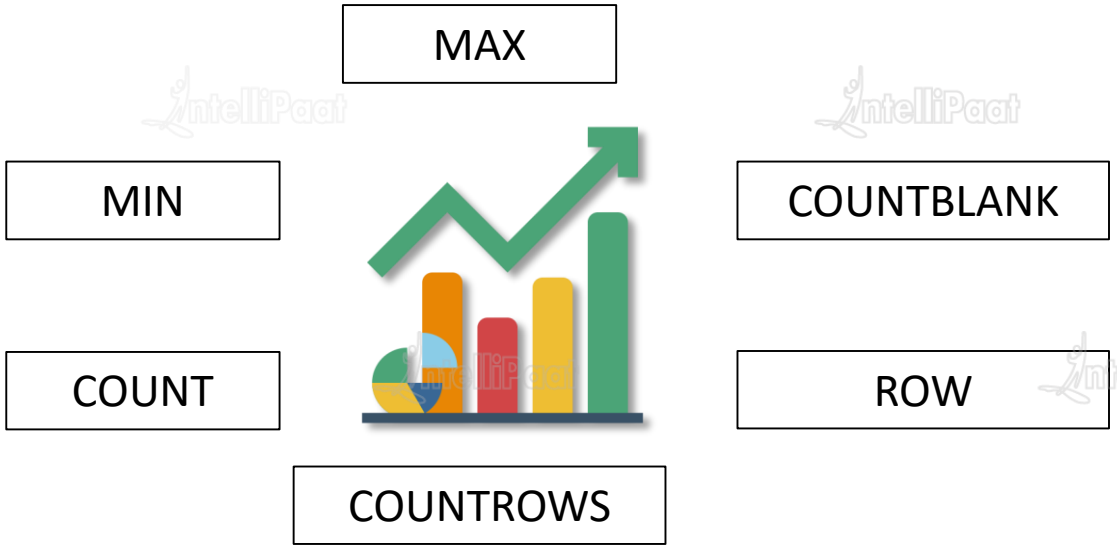
PATHLENGTH

PATHCONTAINS

DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- Parent-Child
- **Statistical**
- Text
- Others

Statistical functions are used to perform aggregations, and we can filter a column prior to aggregations



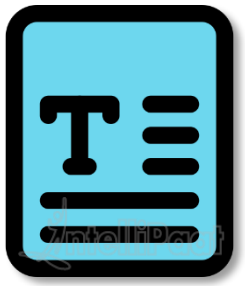
DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- Parent-Child
- Statistical
- **Text**
- Others

Text functions operate on string values. We can use Text functions for searching a text within a string, returning a substring, formatting dates, times, and numbers, and concatenating strings

SEARCH

CONCATENATE



LEN

FIND

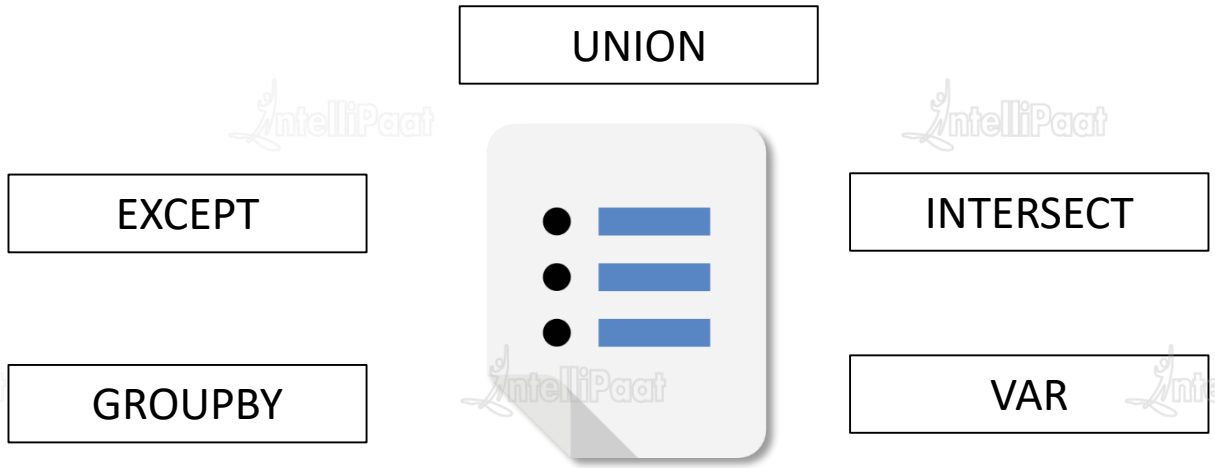
REPLACE

TRIM

DAX Functions

- Date and Time
- Time Intelligence
- Filter
- Information
- Logical
- Math and Trig
- Parent-Child
- Statistical
- Text
- Others

These are some unique functions that do not fall into any of the other categories



Context in DAX

Context in DAX

Context is an important concept to understand if we want to write expressions that return the results we expect

Types

1. A row context is a formula that includes a function that uses filters to identify a single row in a table. The function applies a row context to each Row in the table to which the filter is applied

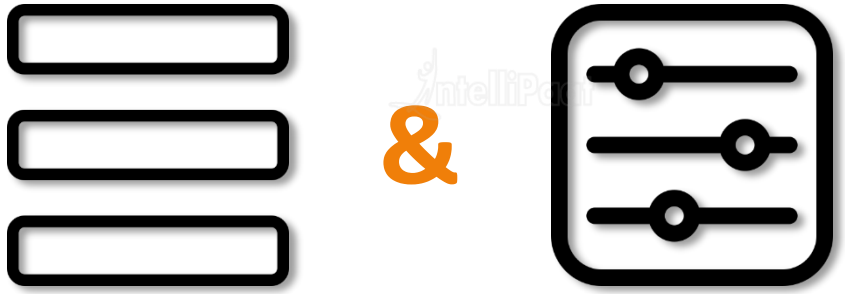
Row Context

2. A filter context is one or more filters applied in a calculation, which determine a single value or the result. We can use a filter context to reduce the values that are included in a calculation

Filter Context

Context in DAX

The following measure demonstrates how a row context and a filter context operate on a calculation in the formula



Using Row Context and Filter Context in a Measure

```
UK Sales = CALCULATE([Total Sales],
Customers[Country] = "UK")
```

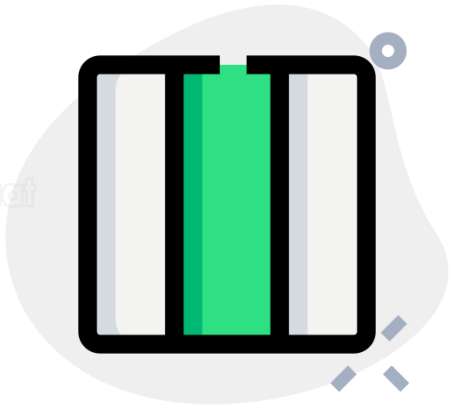
This formula uses **Total Sales** and applies a filter of **UK** so that only the sum of UK Sales will be returned in the result

Demo: Using Row Context and Filter Context

Calculated Columns

Calculated Columns

Calculated columns are added to tables by applying DAX formulas to the existing data



The DAX formula defines the values in the new column rather than querying the data source to create the column

We can create calculated columns by concatenating strings or multiple numbers together, combining the data from anywhere in the model

Calculated columns are similar to measures, but the difference lies in how they are used. Measures are used in the Values area of a visualization, and Calculated columns are used in the Axis, Legend, or Group fields

Calculated Columns

There are two ways in which we can create calculated columns in a data model

Creating a calculated column using the existing data

This code does not include table names since the columns exist within the same table, but it is a good practice to include the table name for clarity

Full Name = [First Name] & " " & [Last Name]

Creating a calculated column using the Related function

If we refer to a column in another table, then we must include the table name using the Related function

Location= RELATED(Countries[Region]) & ", " & [City]

Calculated Tables

Calculated Tables

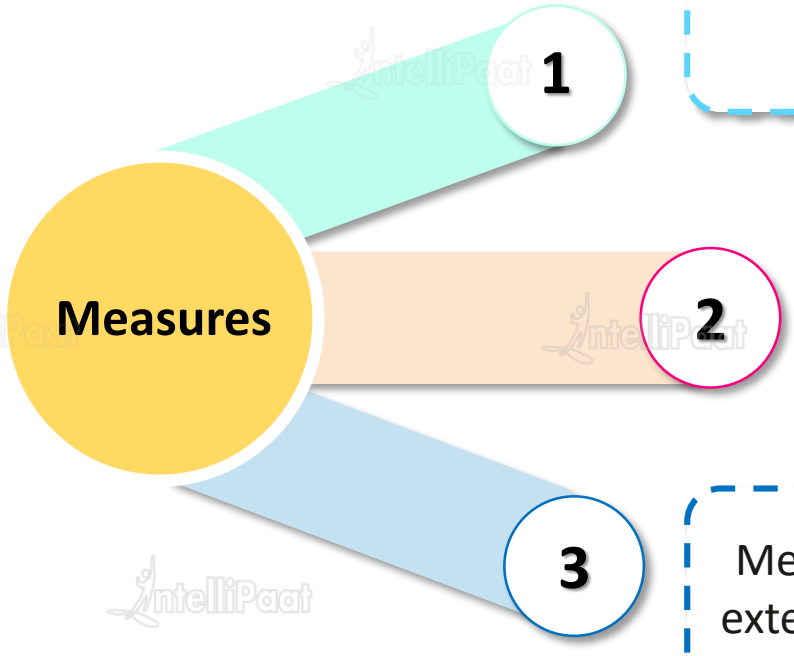
Like calculated columns, calculated tables are also created using the data that already exists in the model, which uses a DAX formula to define the values

Calculated tables are created in both the Report view and the Data view in Power BI Desktop

1. Combine the existing table using UNION
2. Create a calculated table using NATURALINNERJOIN
3. Create a calculated table using the DATATABLE function

Measures

Measures



Power BI measures help us discover insights into our data that might otherwise be hidden

Measures use aggregating functions such as average, minimum, maximum, count distinct, or more complex calculations that use a DAX function

Measures are created using DAX formulas and with an extensive library of functions, operators, and constructs

Demo: Using Calculated Columns and Measures

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24/7 Chat with Our Course Advisor