**Power BI: LINGUAGEM DAX (Data Analysis Expressions)**

**Funções de agregação**

**AVERAGE -** Calcula a média de uma expressão em um contexto especificado. Ela itera por cada linha de uma tabela, aplicando a expressão a cada uma delas e, em seguida, retorna a média desses resultados.

**SUM** - Calcula a soma de uma expressão em um contexto especificado. Ela itera por cada linha de uma tabela, aplicando a expressão a cada uma delas e, em seguida, retorna a soma desses resultados.

**ALL** - Remove todos os filtros aplicados a uma expressão, permitindo que você a avalie em um contexto mais amplo.

**MAX** – Quantidade Máxima de uma expressão

**MIN** – Calcula o mínimo de uma expressão

***Toda vez que for trabalhar com MARGEM (%) sempre trabalhar com MEDIDA ao invés de COLUNA CALCULADA.***

***Uma medida sempre deve está atrelada a uma tabela. As medidas são criadas usando funções agregadas, como soma, média, máximo, mínimo, entre outras, para fornecer informações resumidas sobre os dados usadas em visualizações interativas como gráficos e tabelas e cartões.***

* **KPIs (Indicadores-chave de desempenho.**
* **Agregação versus Valor Individual:**As medidas são calculadas para fornecer um valor agregado, como soma, média ou contagem, com base em um conjunto de linhas ou colunas. Por outro lado, as colunas calculadas retornam um valor individual para cada linha da tabela, calculado com base em uma fórmula específica.
* **Reutilização:** As medidas podem ser reutilizadas em várias visualizações e relatórios. Elas são criadas uma vez e podem ser usadas em diferentes partes do projeto. Por outro lado, as colunas calculadas são específicas para a tabela em que são criadas e não podem ser diretamente reutilizadas em outras tabelas.

**VAR = Variável**

**Funções Iteradora**

**AVERAGEX -** Calcula a média de uma expressão em um contexto especificado. Ela itera por cada linha de uma tabela, aplicando a expressão a cada uma delas e, em seguida, retorna a média desses resultados.

**SUMX** - Calcula a soma de uma expressão em um contexto especificado. Ela itera por cada linha de uma tabela, aplicando a expressão a cada uma delas e, em seguida, retorna a soma desses resultados.

**MAXX** – Quantidade Máxima de uma expressão

**MINX** – Calcula o mínimo de uma expressão

**COUNTAX:** Conta o número de linhas em uma tabela que atendem a uma condição específica.

**FILTER:** Filtra as linhas de uma tabela com base em uma condição e retorna a tabela filtrada.

**ALLSELECTED:** Retorna todas as linhas de uma tabela, ignorando quaisquer filtros aplicados nas visualizações.

**Função de Conversão**

* **A função CONVERT no DAX (Data Analysis Expressions)** é usada para converter um valor de um tipo de dados para outro. Ela é útil quando você precisa alterar a representação de um valor para atender às suas necessidades de análise e visualização. A sintaxe básica da função CONVERT é a seguinte: **CONVERT(<valor>, <tipo\_de\_dados>)**Onde <valor> é o valor a ser convertido e <tipo\_de\_dados> é o tipo de dados para o qual você deseja converter o valor, incluindo os seguintes tipos: **INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME.**

**CONVERT(3.14, INTEGER) -** Este exemplo converte o número decimal 3.14 em um número inteiro.

**CONVERT(100, STRING)** - Aqui, o número 100 é convertido em uma string.

**CONVERT("2023-06-15", DATETIME)   
CONVERT(50, CURRENCY) -** Aqui, o número 50 é convertido em um valor monetário.

**Funções BLANK e ISBLANS**

**As funções BLANK e ISBLANK no DAX (Data Analysis Expressions)** são úteis para trabalhar com valores vazios ou nulos em suas fórmulas. Elas desempenham um papel importante na manipulação e tratamento de dados ausentes ou indefinidos.  
**A função BLANK** é usada para representar um valor vazio ou nulo em uma expressão. Vocpode usá-la para atribuir um valor vazio a uma coluna calculada ou a uma medida.

Por exemplo:

***MinhaColunaCalculada = IF(ISBLANK([MinhaColuna]), BLANK(), [MinhaColuna])***

Nesse caso, a função ISBLANK é usada para verificar se a coluna [MinhaColuna] está vazia ou nula. Se sim, a função BLANK é aplicada para atribuir um valor vazio à coluna calculada MinhaColunaCalculada. Caso contrário, o valor da coluna [MinhaColuna] é mantido.  
A função ISBLANK, por sua vez, é usada para verificar se um valor está vazio ou nulo. Ela retorna VERDADEIRO se o valor for vazio e FALSO caso contrário.   
***IF(ISBLANK([MinhaColuna]), "Valor Vazio", "Valor Preenchido")****Nele, a função ISBLANK é utilizada para verificar se a coluna [MinhaColuna] está vazia. Se sim, a expressão retorna "Valor Vazio", caso contrário, retorna "Valor Preenchido".*

**Função IF, IFERROR e ISERROR  
  
IF (SE)**

* **Função:**Testa uma condição e retorna um valor se for verdadeira e outro se for falsa. ***IF(condição, valor\_se\_verdadeiro, valor\_se\_falso)***

**IFERROR (SEERRO)**

* **Função:** Verifica se uma fórmula resulta em erro e retorna um valor alternativo caso ocorra.  
  ***IFERROR(fórmula, valor\_se\_erro)***

**ISERROR (ÉERRO)**

**Função**: Verifica se uma célula ou fórmula resulta em qualquer tipo de erro (ex: #N/A, #DIV/0!, #VALUE!), retornando VERDADEIRO ou FALSO.  
***ISERROR(valor)*Diferença entre IFERROR e ISERROR**

***IFERROR substitui o erro por um valor definido.***

***ISERROR apenas identifica se há erro (sem substituição***

**Função SWITCH  
A função SWITCH no DAX** é usada para avaliar uma expressão e retornar um valor com base em correspondências específicas, funcionando como uma alternativa mais limpa e eficiente a múltiplas condições aninhadas com IF.  
**Sintaxe:  
Comparação com TRUE() para Condições Complexas**  
***FaixaPreço =***

***SWITCH(***

***TRUE(),***

***Produtos[Preço] > 1000, "Premium",***

***Produtos[Preço] > 500, "Intermediário",***

***"Econômico" -- Valor padrão***

***)***

**Tratamento de Erros com SWITCH**Pode ser combinado com **ISBLANK() ou ISERROR():  
  
Sintaxe:  
*StatusVenda =***

***SWITCH(***

***TRUE(),***

***ISBLANK(Vendas[Data]), "Não Concluída",***

***Vendas[Valor] <= 0, "Inválida",***

***"Válida"***

***)***

**Função RELATED  
A função RELATED** é uma das funções mais úteis e poderosas no contexto do DAX (Data Analysis Expressions). Ela é usada em modelos de dados relacionais, como aqueles criados no Power BI e no Excel, para recuperar valores de uma tabela relacionada com base em uma relação estabelecida.  
A função **RELATED** permite que você acesse informações de uma tabela relacionada a partir de uma tabela atual. Ela é frequentemente usada em colunas calculadas ou medidas, onde você deseja trazer informações de outra tabela com base em uma relação.  
  
**A sintaxe básica da função RELATED é a seguinte:**

**RELATED(TabelaRelacionada[ColunaRelacionada])  
Onde:**

* **"TabelaRelacionada" é o nome da tabela da qual você deseja recuperar os valores.**
* **"ColunaRelacionada" é o nome da coluna na tabela relacionada que contém os valores que você deseja trazer.**

**Função VALUES**

**A função VALUES** é usada para retornar uma tabela de valores únicos de uma coluna específica em um contexto de filtro. Ela é frequentemente utilizada para criar uma lista distinta de valores com base nas seleções ou filtros aplicados em outras colunas.   
  
***A sintaxe básica da função VALUES é:  
VALUES(tabela[coluna])  
  
Onde "tabela" é a tabela que contém a coluna desejada, e "coluna" é o nome da coluna da qual se deseja extrair os valores únicos. O resultado é uma nova tabela contendo os valores únicos da coluna especificada.***

**Função DISTINCT**

**A função DISTINCT** é semelhante à função VALUES, pois também retorna uma tabela de valores únicos de uma coluna específica. No entanto, a função DISTINCT pode ser usada sem a necessidade de especificar uma tabela, tornando-a mais flexível quando você precisa de uma lista de valores únicos sem se referir a uma tabela específica.

***A sintaxe básica da função DISTINCT é:  
DISTINCT(tabela[coluna])  
Produtos Distintos = DISTINCT(Vendas[Produto])***

***Ambas as funções são especialmente úteis em cenários de criação de relatórios e análises, quando você precisa gerar listas de valores exclusivos para criar visualizações específicas ou para realizar cálculos em um contexto de filtro mais refinado.***

**Funções de modificações de filtros**

**ALL:** É uma das funções mais usadas nesta categoria. Ela remove os filtros de uma ou mais colunas especificadas, ou mesmo de toda a tabela, garantindo que todas as linhas sejam consideradas no cálculo. Isso é especialmente útil quando você quer realizar cálculos ignorando os filtros aplicados em uma visualização específica.  
  
**ALLEXCEPT:** É utilizada para remover os filtros de todas as colunas de uma tabela, exceto aquelas que você deseja manter no contexto de avaliação. Ela permite que você controle quais colunas terão impacto no resultado da fórmula, enquanto outras serão ignoradas.  
  
**FILTER:** É uma função versátil que permite criar filtros personalizados em tempo de cálculo. Ela permite que você especifique condições complexas para filtrar uma tabela e obtenha resultados dinâmicos com base nessas condições.  
  
**VALUES:** É frequentemente utilizada para criar uma lista distinta de valores de uma coluna, removendo qualquer filtro que possa estar sendo aplicado a ela. Ela retorna todos os valores únicos da coluna, independentemente dos filtros nas outras colunas.  
  
**ISBLANK -** Um valor booliano de TRUE se o valor estiver em branco; caso contrário, FALSE.  
  
**CALCULATE** - é uma das funções mais importantes no DAX e é usada para modificar ou estender o contexto de avaliação padrão em uma fórmula. Em essência, a função **CALCULATE** permite que você crie cálculos que consideram diferentes níveis de filtros ou adicionem/removam filtros específicos para atender às necessidades da sua análise.

A função **CALCULATE** no Power BI é uma das funções mais poderosas da linguagem DAX, pois ela permite não apenas realizar cálculos, mas também controlar o contexto de filtro em que esses cálculos são aplicados.

* **Sobrescrever um Filtro:**

Às vezes, você pode querer realizar um cálculo específico em um contexto de filtro diferente do que está sendo aplicado naturalmente. A função CALCULATE() permite fazer isso. Para sobrescrever um filtro, você pode passar uma ou mais expressões para a função, que serão usadas como critérios para o filtro. Isso basicamente cria um novo contexto de filtro temporário.

Por exemplo, suponha que você tenha uma tabela com vendas e queira calcular a soma das vendas apenas para um determinado ano, independentemente de quaisquer outros filtros aplicados. Você pode fazer o seguinte:

**TotalVendasPorAno =**

**CALCULATE(**

**SUM(Vendas[Valor]),**

**FILTER(**

**ALL(Vendas),**

**Vendas[Ano] = 2023**

**)**

**)**

**Nesse exemplo, a função CALCULATE() está sendo usada para sobrescrever o filtro naturalmente aplicado aos dados da tabela "Vendas" e, em vez disso, aplicar um filtro específico para o ano de 2023.**

* **Agregar em um Filtro Já Existente:**

Além de sobrescrever filtros, a função CALCULATE() também pode ser usada para agregar cálculos em um filtro já existente. Isso permite que você adicione cálculos às condições de filtro existentes.

Suponha que você tenha uma medida para calcular a média das vendas para um determinado produto, mas deseja calcular essa média apenas para produtos que tiveram mais de 100 unidades vendidas:

**MediaVendasPorProduto =**

**CALCULATE(**

**AVERAGE(Vendas[Valor]),**

**FILTER(Vendas, Vendas[Quantidade] > 100) )**

**Nesse caso, a função CALCULATE() está agregando o cálculo da média das vendas à condição de filtro existente (mais de 100 unidades vendidas) sem alterar o filtro original que pode estar aplicado a outros aspectos do relatório.**

**Contexto de Linha:** O contexto de linha se refere ao cálculo que é realizado em cada linha individual de uma coluna. No entanto, ao criar medidas, o cálculo não é efetuado em uma linha específica, mas sim em um contexto mais amplo da coluna como um todo. Duas maneiras são apresentadas para tratar esse contexto de linha:

**Coluna Calculada:** Uma coluna é criada com um cálculo para cada linha da tabela. Isso permite que você realize um cálculo específico para cada linha e retorne o resultado. Essa abordagem é útil quando você precisa de uma coluna adicional com informações específicas para cada linha.

**Funções Iteradoras:** Estas são funções DAX que permite iterar por cada linha da tabela e realizar uma operação, mesmo antes de calcular uma soma, média ou outro tipo de agregação. Um exemplo mencionado é a função SUMX(), que realiza a soma dos valores após iterar por cada linha.

**Contexto de Filtro:** O contexto de filtro é influenciado pelos filtros aplicados nas tabelas. Isso significa que um cálculo agregado, como uma soma, é afetado pelo filtro que foi aplicado à tabela. Por exemplo, a função SUM() aplicada a uma coluna de valores Total será calculada considerando os filtros ativos na tabela.

**KEEPFILTER:** A função DAX **KEEPFILTERS (ManterFiltro, em português)** é uma função poderosa no contexto de fórmulas e expressões no Power BI, Power Pivot e outras ferramentas que utilizam a linguagem de fórmula DAX. Essa função é usada para preservar os filtros aplicados em uma coluna ou tabela, enquanto você executa cálculos em outra coluna ou tabela. Isso é especialmente útil quando você quer realizar cálculos específicos sem afetar os filtros já aplicados**.**

* **Preservando Filtros:**

A função **KEEPFILTERS** permite que você execute cálculos em uma coluna, mantendo os filtros ativos em outras colunas. Isso é útil quando você deseja realizar cálculos em um subconjunto específico de dados sem afetar os filtros em outras partes do conjunto de dados.

**Exemplo:**

Suponha que você tenha uma tabela chamada "Vendas" com colunas "Data", "Produto" e "Quantidade". Você deseja calcular a média de quantidades vendidas, mantendo os filtros de data aplicados.

**MédiaQuantidadeVendida = AVERAGE(KEEPFILTERS('Vendas', 'Vendas'[Quantidade]))**

**Filtros Contextuais:**

A função **KEEPFILTERS** também pode ser usada para criar cálculos que mantêm filtros específicos durante a avaliação de expressões. Isso ajuda a garantir que os filtros sejam aplicados em cálculos que dependam do contexto.

**Exemplo:**

Digamos que você queira calcular a porcentagem das vendas de um determinado produto em relação ao total de vendas, mantendo o filtro no produto selecionado.

**PorcentagemVendasProduto = DIVIDE(SUM(KEEPFILTERS('Vendas', 'Vendas'[Quantidade])), SUM('Vendas'[Quantidade]))**

**Lembre-se de que a função KEEPFILTERS pode ser especialmente útil em situações onde você precisa garantir que os cálculos sejam realizados apenas em um subconjunto específico de dados, mantendo os filtros aplicados. Isso ajuda a evitar resultados inesperados ou incorretos em suas análises.**

**FUNÇÃO IN**: Possibilita agrupar dois tipos de expressões a qual deseja visualizar.