

# Sumário

Alavancando no DAX.....	1
Construção de colunas calculadas .....	2
Funções de string - Mês, Ano.....	3
Função de formatação - Mês Ano.....	4
Cálculo de idade.....	4
SWITCH() - divisão por idade .....	5
SWITCH() – divisão de idade .....	6
Funções de navegação – RELATED.....	7
Medidas – Básico .....	8
Medidas – Agregação básica.....	9
Total de Vendas .....	9
Total de Custos.....	10
Lucro.....	10
Margem de Lucro .....	11
Parâmetros opcionais .....	11
Contexto de Filtro .....	11
CALCULATE.....	12
Cálculo de percentual do total.....	13
Inteligência de Tempo.....	13
Vendas do Ano até a Data (Year to Date Sales) .....	13
Vendas do Ano até a Data (Calendário Fiscal) .....	14
Vendas do Ano Anterior.....	14
Resumo .....	15

## Alavancando no DAX

(Expressões de Análise de Dados) é uma linguagem de fórmula que fez sua estreia em 2010 com o lançamento do Power Pivot no Excel. Grande parte do DAX é semelhante às funções do Excel, e, portanto, aprender DAX é uma transição fácil para usuários do Excel e usuários avançados. Na verdade, o DAX é tão semelhante ao Excel que já vi novos alunos ficarem confortáveis com a linguagem e começarem a escrever DAX em questão de minutos.

O objetivo deste capítulo é apresentar você ao DAX e dar a confiança necessária para começar a explorar essa linguagem por conta própria. Devido à brevidade deste capítulo, não haverá discussões sobre conceitos e teorias avançadas de DAX. Existem, é claro, muitos outros livros dedicados exclusivamente a isso.

Agora, vamos dar uma olhada no que é abordado neste capítulo:

- Construção de colunas calculadas.
- Medidas calculadas - o básico.
- Medidas calculadas - contexto de filtro.

- Medidas calculadas - inteligência de tempo.

## Construção de colunas calculadas

Abra o arquivo pbix Chapter 4 - Leveraging DAX dos arquivos do livro. Colunas calculadas são armazenadas na tabela em que são atribuídas, e os valores são estáticos até que os dados sejam atualizados. Você aprenderá mais sobre a atualização de dados em um capítulo posterior. Existem muitos casos de uso para colunas calculadas, mas os dois mais comuns são:

- Atributos descritivos.
- Colunas de chave concatenadas.

Agora você vai criar sua primeira coluna calculada. Antes de começar, no entanto, é preciso saber que o Power BI Desktop possui o IntelliSense. O IntelliSense ajudará bastante ao escrever código, como você descobrirá em breve. Essa funcionalidade integrada autocompletará seu código conforme você digita e também o ajudará a explorar e descobrir novas funções na linguagem DAX. Para aproveitar o IntelliSense, basta começar a digitar na barra de fórmulas. Agora você está pronto para começar a escrever DAX!

Clique em Data View, localizado no lado esquerdo da tela do Power BI Desktop. Em seguida, clique na tabela de clientes na lista de Campos. Depois de selecionar a tabela de clientes, clique em Nova Coluna, encontrado na faixa de modelagem, conforme mostrado na captura de tela a seguir:

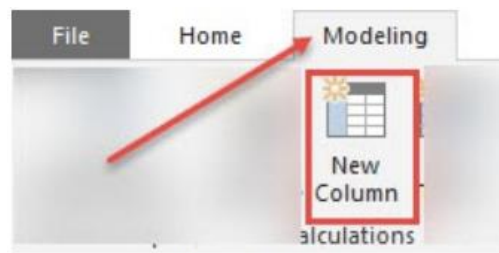


Figure 1- New column

Agora você verá o texto "Column =" na barra de fórmulas. Primeiro, nomeie a nova coluna substituindo o texto padrão de "Column" por "Full Name". Em seguida, mova o cursor para depois do sinal de igual e digite um caractere de apóstrofo. Imediatamente após digitar o caractere de apóstrofo, uma lista de opções de autocompletar aparecerá antes da barra de fórmulas. Isso é o IntelliSense em ação. A primeira opção nesta lista é o nome da tabela que você tem atualmente selecionada - Customer. Clique na tecla Tab e o nome da tabela será adicionado automaticamente à barra de fórmulas, conforme mostrado na captura de tela a seguir:

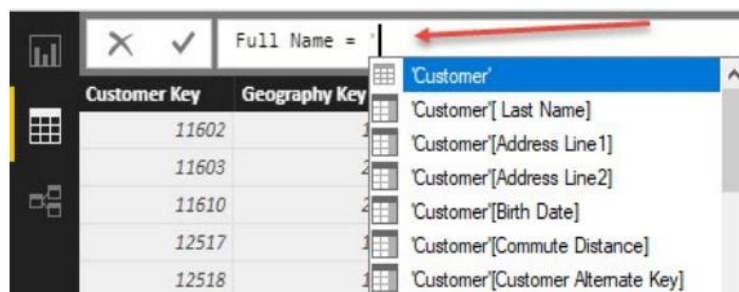


Figure 2-Adding name of the table

*Em algum momento, você descobrirá inevitavelmente que pode fazer referência apenas ao nome da coluna. Como prática recomendada, sempre recomendamos fazer referência tanto ao nome da tabela quanto ao nome da coluna sempre que você usar uma coluna em seu código DAX.*

Em seguida, digite um colchete aberto na barra de fórmulas, seguido por uma letra maiúscula F, fazendo [F. Mais uma vez, você será imediatamente apresentado com opções de autocompletar. A lista de opções foi limitada apenas

às colunas que contêm a letra f, e a primeira opção disponível no menu suspenso é First Name. Clique em Tab para autocompletar. A barra de fórmulas agora deve conter a seguinte fórmula:

```
Full Name = 'Customer'[First Name]
```

O próximo passo é adicionar um espaço, seguido pelo sobrenome. Existem duas opções no DAX para combinar valores de string. A primeira opção é a função concatenate. Infelizmente, a concatenate aceita apenas dois parâmetros; portanto, se você tiver mais de dois parâmetros, seu código exigirá chamadas múltiplas da função concatenate. Por outro lado, você também tem a opção de usar o sinal de e comercial (&) para combinar strings. O sinal de e comercial primeiro levará ambos os parâmetros de entrada e os converterá em strings. Após esta etapa de conversão de dados, as duas strings são então combinadas em uma. Vamos continuar com o restante da expressão. Lembre-se de usar a funcionalidade de autocompletar incorporada para ajudar a escrever o código.

Em seguida, adicione um espaço e a coluna de sobrenome. Para adicionar um espaço - ou qualquer valor literal de string, você usará aspas em ambos os lados da string. Por exemplo, " " insere um espaço entre os nomes e sobrenomes. A fórmula DAX concluída será parecida com o seguinte:

```
Full Name = 'Customer'[First Name] & " " & 'Customer'[Last Name]
```

## Funções de string - Mês, Ano

Agora que você concluiu sua primeira coluna calculada, vamos construir uma coluna calculada que armazena o valor mês-ano. O objetivo é retornar uma coluna mês-ano com o mês de dois dígitos e o ano de quatro dígitos separados por um traço, formando "MM-YYYY". Vamos construir esse cálculo incrementalmente.

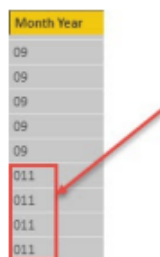
Selecione a tabela Date (Order) e clique em New Column na faixa de modelagem. Escreva o seguinte código na barra de fórmulas e pressione Enter:

```
Month Year = 'Date (Order)'[Month Number of Year]
```

Ao começar a validar o código, você perceberá que isso retorna apenas o mês de um dígito sem zero à frente. Sua próxima tentativa pode se parecer com o seguinte:

```
Month Year = "0" & 'Date (Order)'[Month Number of Year]
```

Isso funcionará para meses de um dígito; no entanto, meses de dois dígitos agora retornarão três dígitos. Veja a captura de tela a seguir:



Month Year
09
09
09
09
09
011
011
011
011
011

Figure 3-Displaying Month Year

Para aprimorar isso e retornar apenas o mês com dois dígitos, você pode usar a função RIGHT. A função RIGHT retorna um número específico de caracteres do lado direito de uma string. Modifique sua fórmula DAX existente para ficar assim:

```
Month Year = RIGHT("0" & 'Date (Order)'[Month Number of Year], 2)
```

Para obter uma lista completa das funções de texto no DAX, visite o seguinte link: <https://tinyurl.com/pbigs-text>

O restante desta fórmula pode ser concluído facilmente. Primeiro, para adicionar um traço, o seguinte código DAX pode ser usado:

```
Month Year = RIGHT("0" & 'Date (Order)'[Month Number of Year], 2) & "-"
```

Complete a fórmula Month Year combinando a string atual com a coluna do ano do calendário:

```
RIGHT("0" & 'Date (Order)'[Month Number of Year], 2) & "-" & 'Date (Order)'[Year]
```

*Você pode ter notado que a coluna Year tem um tipo de dados de número inteiro, e você pode ter esperado que esse valor numérico precisasse ser convertido em uma string antes da operação de combinação. No entanto, lembre-se de que o operador de e comercial (&) converterá automaticamente ambas as entradas em uma string antes de realizar a operação de combinação!*

## Função de formatação - Mês Ano

Assim como em qualquer outro idioma, você perceberá que geralmente existem várias maneiras de fazer algo. Agora você vai aprender como realizar o cálculo que vimos na seção anterior usando a função FORMAT. A função FORMAT permite que você pegue uma coluna de número ou data e a personalize de várias maneiras. Um efeito colateral da função FORMAT é que o tipo de dados resultante será texto. Vamos realizar o cálculo anterior novamente, mas desta vez usando a função FORMAT.

Certifique-se de ter a tabela Date (Order) selecionada e clique em Criar uma Nova Coluna Calculada, selecionando Nova Coluna na faixa de modelagem. Na barra de fórmulas, escreva a seguinte expressão:

```
Month Year Format = FORMAT('Date (Order)'[Date], "MM-YYYY")
```

*Se você quiser dar uma olhada completa em todas as opções de formatação personalizada disponíveis usando a função FORMAT, por favor, dê uma olhada em <https://tinyurl.com/pbigs-format>.*

## Cálculo de idade

A seguir, você vai determinar a idade de cada cliente. A tabela Customer atualmente contém uma coluna com a data de nascimento de cada cliente. Esta coluna, juntamente com a função TODAY e algum DAX, permitirá que você determine a idade de cada cliente. Sua primeira tentativa neste cálculo pode ser usar a função DATEDIFF em um cálculo que se parece com o seguinte:

```
Customer Age = DATEDIFF('Customer'[Birth Date], TODAY(), YEAR)
```

A função TODAY retorna a data e hora atuais. A função DATEDIFF retorna a contagem do intervalo especificado entre duas datas; no entanto, ela não leva em consideração o dia e o mês e, portanto, nem sempre retorna a idade correta para cada cliente.

Vamos reescrever a fórmula DAX anterior de uma maneira diferente. Neste exemplo, você vai aprender a usar lógica condicional e a função FORMAT para retornar a idade correta do cliente. Por favor, tenha em mente que existem muitas maneiras de realizar este cálculo.

Selecione a coluna Customer Age da etapa anterior e reescreva a fórmula para se parecer com o seguinte:

```
Customer Age =  
IF(  
    FORMAT('Customer'[Birth Date], "MMDD") <= FORMAT(TODAY(), "MMDD"), //Logical Test  
    DATEDIFF('Customer'[Birth Date], TODAY(), YEAR), //Result If True  
    DATEDIFF('Customer'[Birth Date], TODAY(), YEAR) -1) //Result If False
```

Figure 4-Select Customer age and rewrite the formula

A formatação do código é muito importante para a legibilidade e manutenção do código. O Power BI Desktop possui uma funcionalidade integrada para auxiliar na formatação do código. Quando você digita Shift + Enter para navegar até a próxima linha na barra de fórmulas, seu código será automaticamente indentado onde aplicável.

Quando concluído, o código anterior retorna a idade correta para cada cliente. A função FORMAT é utilizada para retornar o mês com dois dígitos e o dia com dois dígitos para cada data (a data de nascimento e a data de hoje). Após a parte do teste lógico da instrução IF, existem duas expressões. A primeira expressão é acionada se o teste lógico for avaliado como verdadeiro, e a segunda expressão é acionada se o resultado do teste for falso. Portanto, se a combinação de mês e dia do cliente for menor ou igual ao mês e dia de hoje, então o aniversário já ocorreu este ano, e o teste lógico será avaliado como verdadeiro, acionando a primeira expressão. Se o aniversário do cliente ainda não ocorreu este ano, então a segunda expressão será executada.

*Na fórmula DAX anterior, adicionei comentários usando duas barras inclinadas para frente no código. Comentários são descritivos e não são executados junto com o restante da fórmula DAX. Comentar o código é sempre incentivado e tornará seu código mais legível e fácil de manter.*

## SWITCH() - divisão por idade

Agora que você tem a idade do cliente, é hora de colocar cada cliente em um intervalo de idade. Para este exemplo, haverá quatro intervalos de idade separados:

18-34  
35-44  
45-54  
55 +

A função SWITCH é preferível à função IF ao realizar múltiplos testes lógicos em uma única fórmula DAX. Isso ocorre porque a função SWITCH é mais fácil de ler e facilita a depuração do código.

Com a tabela Customer selecionada, clique em Nova Coluna na faixa de modelagem. Digite a fórmula DAX concluída para o exemplo a seguir:

```
Age Breakdown =  
SWITCH(TRUE(),  
    'Customer'[Customer Age] >= 55, "55 +", //If 55 or older then 55 +  
    'Customer'[Customer Age] >= 45, "45-54", //If 45-54 then 45-54  
    'Customer'[Customer Age] >= 35, "35-44", //If 35-44 then 35-44  
    "18-34") //ELSE, 18-34
```

Figure 5-Completed DAX formula

A formatação do código é muito importante para a legibilidade e manutenção do código. O Power BI Desktop possui uma funcionalidade incorporada para auxiliar na formatação do código. Quando você pressiona Shift + Enter para navegar para a próxima linha na barra de fórmulas, seu código será automaticamente indentado quando aplicável.

Quando concluído, o código anterior retorna a idade correta para cada cliente. A função FORMAT é utilizada para retornar o mês com dois dígitos e o dia com dois dígitos para cada data (a data de nascimento e a data atual). Após a parte de teste lógico da instrução IF, existem duas expressões. A primeira expressão é acionada se o teste lógico for avaliado como verdadeiro, e a segunda expressão é acionada se o resultado do teste for falso. Portanto, se a combinação de mês e dia do cliente for menor ou igual ao mês e dia de hoje, então o aniversário deles já ocorreu este ano, e o teste lógico será avaliado como verdadeiro, acionando a primeira expressão. Se o aniversário do cliente ainda não ocorreu este ano, a segunda expressão será executada.

*Na fórmula DAX anterior, adicionei comentários usando duas barras diagonais no código. Comentários são descritivos e não são executados com o restante da fórmula DAX. Comentar o código é sempre incentivado e tornará seu código mais legível e fácil de manter.*

## SWITCH() – divisão de idade

Agora que você tem a idade do cliente, é hora de colocar cada cliente em uma faixa etária. Para este exemplo, haverá quatro faixas etárias separadas:

- 18-34
- 35-44
- 45-54
- 55 +

A função SWITCH é preferível à função IF ao realizar vários testes lógicos em uma única fórmula DAX. Isso ocorre porque a função SWITCH é mais fácil de ler e facilita a depuração do código.

Com a tabela Customer selecionada, clique em Nova Coluna na faixa de modelagem. Digite a fórmula DAX concluída para o exemplo a seguir:

```
Age Breakdown =  
SWITCH(TRUE(),  
    'Customer'[Customer Age] >= 55, "55 +", //If 55 or older then 55 +  
    'Customer'[Customer Age] >= 45, "45-54", //If 45-54 then 45-54  
    'Customer'[Customer Age] >= 35, "35-44", //If 35-44 then 35-44  
    "18-34") //ELSE, 18-34
```

Figure 5-Completed DAX formula

A fórmula anterior é muito legível e compreensível. Existem três testes lógicos, e se a idade do cliente não for avaliada como verdadeira em nenhum desses testes lógicos, então esse cliente é automaticamente colocado na faixa etária de 18-34 anos.

O leitor perspicaz pode ter notado que o segundo e o terceiro teste lógico não têm uma faixa superior atribuída. Por exemplo, o segundo teste simplesmente verifica se a idade do cliente é 45 ou mais. Naturalmente, você pode presumir que um cliente com idade de 75 anos seria incorretamente atribuído à faixa etária de 45-54 anos.

No entanto, uma vez que uma linha é avaliada como verdadeira, ela não está mais disponível para testes lógicos subsequentes. Alguém com 75 anos teria sido avaliado como verdadeiro no primeiro teste lógico (55 +) e não estaria mais disponível para qualquer teste adicional.

*Se você quiser ter uma compreensão melhor do uso da instrução SWITCH em vez de aninhar várias instruções IF, pode conferir um post no blog de Rob Collie em <https://tinyurl.com/pbiqs-switch>.*

## Funções de navegação – RELATED

Finalmente, é hora de criar uma relação entre a tabela de temperatura e a tabela de vendas pela internet. A chave na tabela de temperatura é uma combinação do nome da região e do número do mês do ano. Essa combinação de colunas torna uma única linha única nesta tabela, como mostrado na captura de tela a seguir

Region	Month	MonthNumber	Key	Avg Temp	Temperature Range
Northeast	Jan	1	Northeast1	26.3	Cold
Northeast	Feb	2	Northeast2	25.4	Cold
Northeast	Mar	3	Northeast3	31.4	Cold
Northeast	Apr	4	Northeast4	48.1	Cool

Figure 6-Column combination that makes a single row unique

Infelizmente, atualmente nenhuma dessas duas colunas existe na tabela de Vendas pela Internet. No entanto, a tabela de Vendas pela Internet possui uma relação com a tabela de Território de Vendas, e a tabela de Território de Vendas possui a região. Portanto, você pode determinar a região para cada venda fazendo uma operação de pesquisa simples. Bem, deveria ser tão simples, mas não é exatamente assim. Vamos ver por quê.

As colunas calculadas não utilizam automaticamente as relações existentes no modelo de dados. Isso é uma característica única das colunas calculadas; medidas calculadas automaticamente enxergam e interagem com todas as relações no modelo de dados. Agora, vamos ver por que isso é importante.

Na captura de tela a seguir, criei uma nova coluna na tabela de Vendas pela Internet e estou tentando retornar o nome da região da tabela de Território de Vendas. Dê uma olhada na captura de tela a seguir:

```
Temperature Key =  
'Sales T
```

Figure 7-Sales Territory table

Observe que não há IntelliSense e que a funcionalidade de autocompletar não está disponível enquanto digito "Sales Territory". A razão para isso é porque a coluna calculada não consegue enxergar as relações existentes no modelo de dados e, portanto, não retorna automaticamente a coluna desejada de outra tabela.

Existe uma explicação muito mais complicada por trás disso, mas por enquanto, basta dizer que as funções de navegação (RELATED e RELATEDTABLE) permitem que colunas calculadas interajam e usem as relações existentes.

Se eu reescrever a seguinte fórmula DAX com a função RELATED, você notará que o IntelliSense retornou, juntamente com a funcionalidade de autocompletar que foi discutida anteriormente





Figure 8-Temperature key column

Agora é hora de criar uma coluna de chave de temperatura (Temperature Key) na tabela de Vendas pela Internet. Crie uma nova coluna na tabela de Vendas pela Internet e, em seguida, digite a seguinte fórmula DAX:

```
Temperature Key =  
RELATED('Sales Territory'[Sales Territory Region]) & //Return the region from Sales Territory table  
RELATED('Date (Order)'[Month Number Of Year]) //Return the Month number of year from the Date table
```

Figure 9-Temperature Key column on the Internet Sales table

Agora que a chave de temperatura foi criada na tabela de Vendas pela Internet, vamos criar a relação. Clique em "Gerenciar Relacionamentos" na faixa de opções principal e, em seguida, clique em "Novo..." para abrir a janela de criação de relacionamento. Em seguida, siga as etapas abaixo para criar um novo relacionamento. Os campos relevantes e entradas para cada etapa estão destacados na captura de tela a seguir:

- Selecione "Vendas pela Internet" no primeiro menu suspenso.
- Selecione "Chave de Temperatura" na lista de colunas.
- Selecione "Temperatura" no segundo menu suspenso (role para a direita).
- Selecione "Chave" na lista de colunas.
- Clique em "OK" para salvar o novo relacionamento.

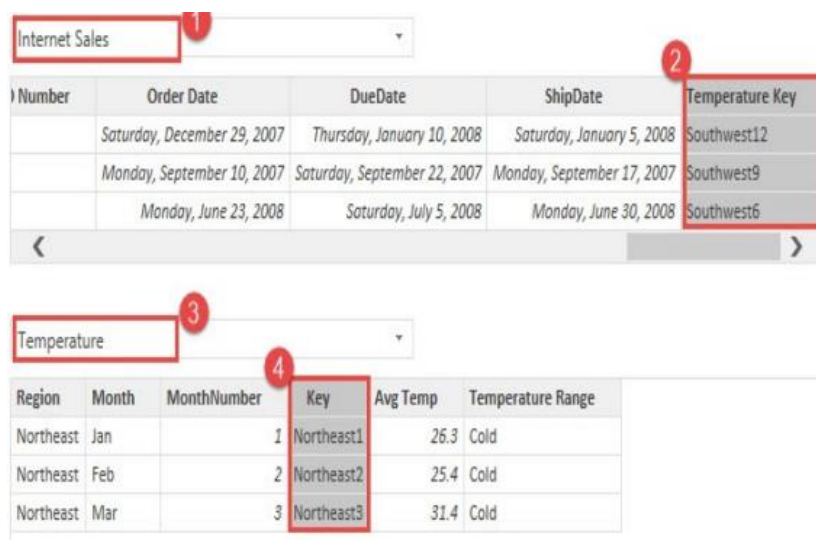


Figure 10-Creating new relationship

## Medidas – Básico

Medidas calculadas são muito diferentes das colunas calculadas. Medidas calculadas não são estáticas e operam dentro do contexto de filtro atual de um relatório; portanto, medidas calculadas são dinâmicas e estão em constante mudança conforme o contexto de filtro se altera. Você foi apresentado ao contexto de filtro no capítulo anterior. O conceito do contexto de filtro será levemente expandido mais adiante neste capítulo. Medidas calculadas são



ferramentas analíticas poderosas e, devido à forma automática como as medidas funcionam com contextos de filtro, são surpreendentemente simples de serem criadas.

Antes de começar a aprender sobre a criação de medidas, vamos primeiro discutir a diferença entre medidas implícitas e explícitas. Agregações implícitas ocorrem automaticamente em colunas com tipos de dados numéricos. Você viu isso no capítulo anterior quando a coluna de ano foi agregada incorretamente após ser adicionada a um relatório. Existem algumas vantagens neste comportamento padrão, por exemplo, se você simplesmente arrastar a coluna de Valor de Vendas para um relatório, o valor será automaticamente agregado e você não precisará gastar tempo criando uma medida. Como discutido na próxima seção, geralmente é considerado uma boa prática criar medidas explícitas em vez de medidas implícitas.

Uma medida explícita permite que um usuário crie uma medida calculada, e há várias vantagens em usar medidas explícitas:

- Medidas podem ser construídas umas sobre as outras.
- Elas encapsulam código, tornando as mudanças lógicas menos demoradas.
- Elas definem centralmente a formatação de números, criando consistência.
- Medidas calculadas podem fazer o seguinte:
- Podem ser atribuídas a qualquer tabela.

## Medidas – Agregação básica

Interagem automaticamente com todas as relações no modelo de dados, ao contrário das colunas calculadas. Não são materializadas em uma coluna e, portanto, não podem ser validadas na visualização de dados.

Nesta seção, você vai criar quatro medidas calculadas simples:

- Total de Vendas
- Total de Custos
- Lucro
- Margem de Lucro

## Total de Vendas

Para criar sua primeira medida, selecione a tabela Vendas pela Internet e clique em Nova Medida... na faixa de modelagem. Na barra de fórmulas, digite o seguinte código e pressione Enter:

Total de Vendas = SUM('Vendas pela Internet'[Valor de Vendas])

Um dos benefícios de criar medidas explícitas é a capacidade de centralizar a formatação. Uma vez que a medida foi criada, navegue até a faixa de modelagem e altere a formatação para \$ Inglês (Estados Unidos), como mostrado na captura de tela a seguir:



Figure 11-Change formatting to \$ English(United States)

## Total de Custos

Agora vamos criar a medida Total de Custos. Mais uma vez, esta é uma operação simples de SOMA. Clique em Nova Medida... na faixa de modelagem e digite a seguinte fórmula DAX:

Total de Custos = SUM('Vendas pela Internet'[Custo Total do Produto])

Lembre-se de aplicar a formatação a esta nova medida; é fácil perder este passo ao aprender a criar medidas. A formatação deve ser \$ Inglês (Estados Unidos).

## Lucro

Lucro é a próxima medida que você irá criar. Você pode tentar escrever algo como o seguinte:

Lucro = SUM('Vendas pela Internet'[Valor de Vendas]) - SUM('Vendas pela Internet'[Custo Total do Produto])

Este cálculo seria tecnicamente correto; no entanto, não é a maneira mais eficiente de escrever o código. Na verdade, outro benefício de construir medidas explícitas é que elas podem ser construídas uma sobre a outra.

Reutilizar medidas calculadas existentes tornará o código mais legível e facilitará as mudanças no código e menos demoradas. Imagine por um momento que você descobriu que o cálculo de Total de Vendas não está correto. Se você encapsulou toda essa lógica em uma única medida e reutilizou essa medida em suas outras medidas, então você só precisaria mudar a medida original, e qualquer atualização seria aplicada a todas as outras medidas.

Agora é hora de criar a medida Lucro. Selecione sua tabela de Vendas pela Internet e clique em Nova Medida... na faixa de modelagem. Digite o seguinte na barra de fórmulas - lembre-se de formatá-lo:

Lucro = [Total de Vendas] - [Total de Custos]

Este cálculo retorna os mesmos resultados da tentativa original. A diferença é que agora você está reutilizando medidas que já foram criadas no modelo de dados. Você pode ter notado que eu referenciei o nome da medida sem o nome da tabela. Ao referenciar medidas explícitas em seu código, é considerada uma boa prática excluir o nome da tabela.

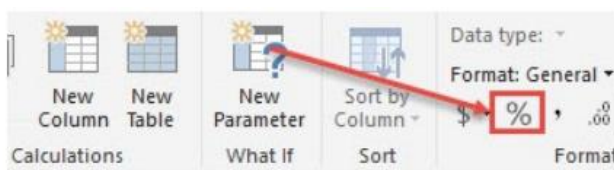
## Margem de Lucro

Agora é hora de criar o cálculo da Margem de Lucro (a margem de lucro é simplesmente o lucro dividido pelas vendas). Para esta medida, você vai usar a função DIVIDE. A função DIVIDE é recomendada em vez do operador de divisão (/) porque a função DIVIDE lida automaticamente com ocorrências de divisão por zero. No caso de ocorrências de divisão por zero, a função DIVIDE retorna vazio.

Crie uma nova medida na tabela Vendas pela Internet usando o seguinte código:

Margem de Lucro = DIVIDE([Lucro], [Total de Vendas])

Em seguida, defina a formatação como porcentagem. Na faixa de modelagem, clique no ícone %, como mostrado na captura de tela a seguir:



```
Profit Margin =  
DIVIDE([Profit], [Total Sales])
```

Figure 12-Setting formatting as a percentage

## Parâmetros opcionais

Você pode ter notado que a função DIVIDE aceitou três parâmetros e você forneceu apenas dois. O terceiro parâmetro permite que você defina um resultado alternativo para ocorrências de divisão por zero. Este resultado alternativo é opcional. Parâmetros opcionais são indicados por colchetes em ambos os lados do parâmetro. Esses parâmetros opcionais são comuns em muitas funções DAX. Dê uma olhada na captura de tela a seguir:

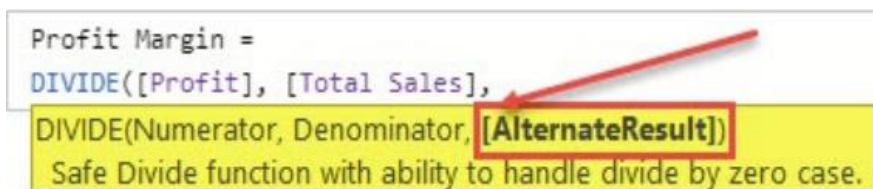


Figure 13-Optional parameters in DAX functions

## Contexto de Filtro

A filtragem automática que ocorre no Power BI é uma característica realmente incrível e é uma das razões pelas quais tantas empresas estão migrando para essa ferramenta. A filtragem automática está diretamente ligada ao conceito de contexto de filtro. Você foi apresentado ao contexto de filtro no capítulo anterior. Quero expandir brevemente sobre o capítulo anterior aqui antes de discutir a função CALCULATE.

Uma definição simples do contexto de filtro seria que é simplesmente tudo em seu relatório que está filtrando uma medida. Existem muitos elementos que compõem o contexto de filtro. Vamos dar uma olhada em alguns deles:

- Quaisquer atributos nas linhas; isso inclui os diferentes eixos em gráficos.
- Quaisquer atributos nas colunas.
- Quaisquer filtros aplicados por filtros visuais (slicers); os slicers são discutidos no próximo capítulo.
- Quaisquer filtros aplicados explicitamente através do painel de Filtros.
- Quaisquer filtros adicionados explicitamente à medida calculada.

## CALCULATE

A função CALCULATE é uma ferramenta extremamente poderosa no arsenal de qualquer autor DAX. Isso porque a função CALCULATE pode ser usada para ignorar, sobrescrever ou alterar o contexto de filtro existente.

Você pode estar se perguntando por quê - por que alguém gostaria de ignorar o comportamento padrão do Power BI? Vamos dar uma olhada em um exemplo.

Vamos supor que você queira retornar as vendas totais de cada país como porcentagem do total de todos os países. Esta é uma calculação muito básica de percentual do total: Vendas Totais por país divididas por Vendas Totais para todos os países. No entanto, como você obtém as vendas totais de todos os países para poder realizar esse cálculo? É aqui que a função CALCULATE entra em cena. Dê uma olhada na captura de tela a seguir:

Sales Territory Country	Total Sales	Total Sales all Countries
Australia	9,061,000.58	29,358,677
Canada	1,977,844.86	29,358,677
France	2,644,017.71	29,358,677
Germany	2,894,312.34	29,358,677
United Kingdom	3,391,712.21	29,358,677
United States	9,389,789.51	29,358,677
<b>Total</b>	<b>29,358,677.22</b>	

Figure14-Calculating total sales of all the countries

Para fazer o cálculo de percentual do total, você precisa obter as Vendas Totais de todos os países na mesma linha que as Vendas Totais. Isso significa que você precisa criar uma nova medida calculada que ignore quaisquer filtros provenientes do atributo de país. Crie uma nova medida calculada em sua tabela de Vendas pela Internet usando a seguinte fórmula DAX:

```
Total Sales all Countries =  
CALCULATE(  
    [Total Sales],  
    ALL(  
        'Sales Territory'[Sales Territory Country]))
```

Figure 15-Create a new calculated measure on Internet sales table using DAX formula

O cálculo anterior retornará todas as vendas para todos os países, ignorando explicitamente quaisquer filtros provenientes da coluna País. Vamos discutir brevemente por que isso funciona.

O primeiro parâmetro da função CALCULATE é uma expressão, e você pode pensar nisso como uma agregação de algum tipo. Neste exemplo, a agregação é simplesmente Vendas Totais. O segundo parâmetro é um filtro que permite que o contexto de filtro atual seja modificado de alguma forma. No exemplo anterior, o contexto de filtro é modificado ignorando quaisquer filtros provenientes do atributo de país. Vamos dar uma olhada na definição da função ALL usada no segundo parâmetro da função CALCULATE:

ALL: Retorna todas as linhas em uma tabela ou todos os valores em uma coluna, ignorando quaisquer filtros que possam ter sido aplicados.

## Cálculo de percentual do total

Agora, crie outra medida calculada na tabela de Vendas pela Internet usando o seguinte código. Certifique-se de formatar a medida como porcentagem:

% de Todos os Países = DIVIDE([Total de Vendas], [Total de Vendas todos os Países])

Na captura de tela a seguir, você pode ver o exemplo completo com ambas as novas medidas criadas nesta seção. Sem um entendimento básico da função CALCULATE, esse tipo de cálculo de percentual do total seria quase impossível:

Sales Territory Country	Total Sales	Total Sales all Countries	% of All Countries
Australia	9,061,000.58	1 \$29,358,677.22	2 30.86%
Canada	1,977,844.86	\$29,358,677.22	6.74%
France	2,644,017.71	\$29,358,677.22	9.01%
Germany	2,894,312.34	\$29,358,677.22	9.86%
NA		\$29,358,677.22	
United Kingdom	3,391,712.21	\$29,358,677.22	11.55%
United States	9,389,789.51	\$29,358,677.22	31.98%
<b>Total</b>	<b>29,358,677.22</b>	<b>\$29,358,677.22</b>	<b>100.00%</b>

Figure 16- Completed example with both of the new measures

## Inteligência de Tempo

Outra vantagem do Power BI é como é fácil adicionar inteligência de tempo ao seu modelo de dados. Dentro das expressões de análise de dados (DAX), você tem uma lista abrangente de funções de inteligência de tempo integradas para facilitar muito esse processo. Nesta seção, você vai usar essas funções integradas para criar as seguintes medidas:

- Vendas do Ano até a Data (Year to Date Sales)
- Vendas do Ano até a Data (Calendário Fiscal)
- Vendas do Ano Anterior (Prior Year Sales)

Cálculos de inteligência de tempo integrados não funcionam se você estiver usando uma conexão de consulta direta com sua fonte de dados em vez de importar dados. Dê uma olhada nos métodos alternativos para calcular inteligência de tempo na folha de dicas DAX em <https://tinyurl.com/pbigqs-daxcheatsheet>.

## Vendas do Ano até a Data (Year to Date Sales)

Crie uma nova medida calculada em sua tabela de Vendas pela Internet usando a seguinte fórmula DAX. Lembre-se de formatar a medida como \$ Inglês (Estados Unidos):

Vendas YTD = TOTALYTD([Total de Vendas], 'Data (Pedido)'[Data])

## Vendas do Ano até a Data (Calendário Fiscal)

Talvez sua necessidade seja um pouco mais complexa, e você precise ver as vendas do ano até a data com base no final do seu ano fiscal em vez da data de término do ano calendário. A função TOTALYTD possui um parâmetro opcional que permite alterar a data de término do ano padrão de "31/12" para uma data diferente. Crie uma nova medida calculada em sua tabela de Vendas pela Internet usando a seguinte fórmula DAX:

Vendas YTD Fiscal = TOTALYTD([Total de Vendas], 'Data (Pedido)'[Data], "31/03")

Agora, vamos dar uma olhada em ambas essas novas medidas em uma tabela no Power BI:

Year	English Month Name	Total Sales	YTD Sales	Fiscal YTD Sales
2005	July	473,388.16	\$473,388	\$473,388
2005	August	506,191.69	\$979,580	\$979,580
2005	September	473,943.03	\$1,453,523	\$1,453,523
2005	October	513,329.47	\$1,966,852	\$1,966,852
2005	November	543,993.41	\$2,510,846	\$2,510,846
2005	December	755,527.89	\$3,266,374	\$3,266,374
2006	January	596,746.56	\$596,747	\$3,863,120
2006	February	550,816.69	\$1,147,563	\$4,413,937
2006	March	644,135.20	\$1,791,698	\$5,058,072
2006	April	663,692.29	\$2,455,391	\$663,692
2006	May	673,556.20	\$3,128,947	\$1,337,248
Total		29,358,677.22		

Figure 17-Both the new measure in a table

As novas medidas criadas, Vendas YTD e Vendas YTD Fiscal, foram ambas adicionadas à tabela anterior. Vamos examinar mais de perto como essas duas medidas são diferentes; as seções relevantes na tabela estão anotadas com os números de um a quatro, correspondendo às seguintes observações:

- O valor exibido para dezembro de 2005 é \$3,266,374. Este é o total acumulado de todas as vendas de 1º de janeiro de 2005 a dezembro de 2005.
- Como esperado, o total acumulado reinicia quando o ano muda de 2005 para 2006; portanto, o valor de Vendas YTD para janeiro de 2006 é \$596,747.
- Na coluna Vendas YTD Fiscal, o total acumulado funciona um pouco diferente. O valor exibido de \$5,058,072 é o total acumulado de todas as vendas de 1º de abril de 2005 a 31 de março de 2006.
- Ao contrário da medida Vendas YTD, a medida Vendas YTD Fiscal não reinicia até 1º de abril. O valor exibido para abril de 2006, de \$663,692, é o total acumulado para abril. Esse número aumentará a cada mês até 31 de maio, momento em que o número será redefinido novamente.

## Vendas do Ano Anterior

Muitas análises de séries temporais consistem em comparar métricas atuais com o mês anterior ou o ano anterior. Existem muitas funções no DAX que funcionam em conjunto com a função CALCULATE para facilitar esse tipo de cálculo. Você vai criar uma nova medida para retornar as vendas totais do ano anterior.

Crie uma nova medida calculada em sua tabela de Vendas pela Internet usando a seguinte fórmula DAX:

```
Prior Year Sales =  
CALCULATE(  
    [Total Sales],           // SUM('Internet Sales'[Sales Amount])  
    SAMEPERIODLASTYEAR(     // Change the filter context to go back one year  
        'Date (Order)'[Date])  
)
```

Figure 18-Create a new calculated measure on your Internet sales



CALCULATE permite que você ignore ou até mesmo altere o contexto de filtro atual. Na fórmula anterior, o CALCULATE é usado para pegar o contexto de filtro atual e alterá-lo para um ano atrás. Essa medida calculada também funciona nos níveis de dia, mês, trimestre e ano da hierarquia. Por exemplo, se você estiver analisando as vendas para 15 de junho de 2018, a medida Vendas do Ano Anterior retornaria as vendas para 15 de junho de 2017. No entanto, se você estiver apenas analisando suas vendas agregadas no nível mensal para junho de 2018, a medida retornaria as vendas para junho de 2017.

*Para uma lista abrangente de todas as funções integradas de inteligência de tempo, consulte*

*<https://tinyurl.com/pbiqs-timeintelligence>.*

## Resumo

Neste capítulo, você aprendeu que o DAX permite que você aprimore significativamente seu modelo de dados, melhorando as capacidades analíticas com uma quantidade relativamente pequena de código. Você também aprendeu a criar colunas calculadas e medidas e como usar o DAX para realizar análises úteis de séries temporais em seus dados. Este capítulo apenas arranhou a superfície do que é possível com o DAX. Conforme você explora mais a linguagem DAX por conta própria, você rapidamente se tornará um autor proficiente de fórmulas DAX. Como todos que aprendem DAX, você inevitavelmente descobrirá que há uma camada de complexidade no DAX que exigirá mais educação para realmente dominar. Quando você chegar a esse ponto, seria vantajoso procurar aulas ou livros que o ajudem a realmente dominar o DAX!