DUAL
QUALIFICAÇÃO PROFISSIONAL

INSTITUTO DO EMPREGO
E FORMAÇÃO PROFISSIONAL

Emprego + Digital

Transformação de Dados

Augusta Martins

www.linkedin.com/in/augustamartins

@maemartins yuoutube

UFCD:10800 - 25 H

UM SERVIÇO AHK Deutsch-Portugiesische Industrie- und Handelskammer Câmara de Comércio e Indústria Luso-Alemã

PRR Plano de Recuperação e Resiliência

REPÚBLICA PORTUGUESA

Financiado pela União Europeia NextGenerationEU

40 ANOS 1983 - 2023 HISTÓRIAS DE QUALIFICAÇÃO

Conteúdo

1. Introdução à transformação de dados 1

2. Transformação de dados (Transform Data) 4

2.1 Extração (Extraction) 4

2.2 Transformação (Transformation) 4

2.3 Carga (Loading) 12

2.4 Função Calculate 12

2.5 Relações eficazmente 13

2.5.1 Cardinalidade (Cardinality) 13

2.5.2 Filtro cruzado (Cross Filter Direction) 15

2.6 Medidas semi-aditivas 17

3. Introdução às medidas DAX no Power BI Desktop 19

3.1 Cálculos 19

3.1.1 Medidas 19

3.1.2 Colunas calculadas 19

3.1.3 Tabelas calculadas 20

3.2 Funções DAX 20

3.3 Funções de Agregação (Simples) 21

3.3.1 SUM 21

3.3.2 AVERAGE 24

3.3.3 MAX 26

3.3.4 MIN 27

3.3.5 COUNT 28

3.3.6 COUNTA 29

3.3.7 COUNTROWS 30

3.3.8 DISTINCTCOUNT 31

3.3.9 DISTINCT 32

3.4 Colunas Calculadas 33

3.5 Medidas rápidas 35

3.6 DAX Avançado 36

3.6.1 CALCULATE 38
3.6.2 FILTER 40
3.6.3 ALL 42
3.6.4 ALLSELECTED 44
3.6.5 ALLEXCEPT 45
3.6.6 REMOVEFILTERS 46
3.6.7 KEEPFILTERS 47
3.7 Funções Iterantes 48
3.7.1 SUMX 48
3.7.2 AVERAGEX 49
3.7.3 MAXX 50
3.7.4 MINX 51
3.7.5 RANKX 52

4. Inteligência de Tempo e Medidas em DAX 54

4.1 SAMEPERIODLASTYEAR 54
4.2 DATEADD 55
4.3 PREVIOUSYEAR 56
4.4 PREVIOUSMONTH 57
4.5 DATESYTD 58
4.6 TOTALYTD 59

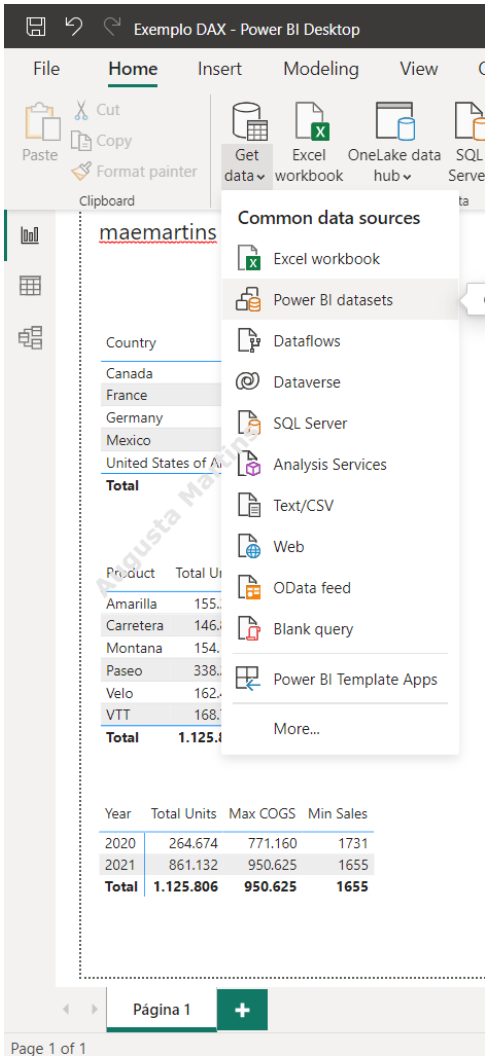
5. Introdução à otimização de desempenho 61

5.1 Desempenho de medidas, relacionamentos e visuais 62
5.1.1 Técnicas de redução de dados para modelos de importação 63
5.2 Variáveis para melhorar o desempenho e a resolução de problemas 64
5.3 Redução de cardinalidade 66
5.4 Otimização dos modelos DirectQuery com armazenamento ao nível da tabela 67
5.5 Criação de gestão de agregações 68

1. Introdução à transformação de dados

A transformação de dados é um processo crucial na análise de dados. Envolve a manipulação e a preparação dos dados brutos para que possam ser utilizados de forma eficaz. As etapas de transformação podem incluir limpeza de dados, correção de erros, filtragem, formatação e combinação de diferentes conjuntos de dados. Estas transformações são realizadas com o objetivo de obter dados estruturados e consistentes, prontos para análise e visualização.

1. Para Transformar os dados clique no separador Base/Home.
2. Obter Dados / Get Data
3. Selecione um conetor



Como podemos observar na imagem, existem vários tipos de conectores dos quais se destacam as seguintes fontes de dados para importar e transformar os dados para análise:

1. **Base de Dados Relacionais:**
   - SQL Server
   - Oracle Database
   - MySQL
   - PostgreSQL
   - IBM DB2
2. **Base de Dados na Nuvem:**
   - Azure SQL Database
   - Amazon Redshift
   - Google BigQuery
3. **Ficheiros:**
   - CSV (Comma Separated Values)
   - Excel
   - XML
   - JSON
4. **Armazenamentos na Nuvem:**
   - Azure Blob Storage
   - SharePoint Online
   - OneDrive
5. **Ferramentas de Análise:**
   - Google Analytics
   - *Sales*force
   - Dynamics 365
   - Adobe Analytics
6. **Redes Sociais e Serviços Web:**
   - Facebook
   - Twitter
   - Web (HTML)
7. **Outros:**
   - OData Feed

- Hadoop File (HDFS)
- SharePoint List

Estes são apenas alguns exemplos dos tipos de conectores disponíveis no Power BI. Cada tipo de conector possui sua própria configuração e requisitos específicos para se ligar às fontes de dados correspondentes. Ao selecionar um conector, é necessário fornecer as informações de conexão adequadas, como endereço do servidor, nome de utilizador, senha, caminho do ficheiro, entre outros, conforme necessário.

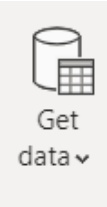
O Power BI oferece uma ampla gama de conectores para facilitar a integração e a importação de dados de diversas fontes, permitindo que os utilizadores acededam e analisem informações de forma eficiente nos seus relatórios e painéis (*dashboards*).

2. Transformação de dados (Transform Data)

A transformação de dados refere-se ao processo de alterar a estrutura ou o formato dos dados. Isto pode envolver a renomeação de colunas, a remoção de colunas desnecessárias, a adição de novas colunas calculadas, a substituição de valores nulos ou ausentes e a agregação de dados em níveis diferentes de granularidade. A transformação de dados é frequentemente realizada usando ferramentas de ETL (Extract, Transform e Load) ou recursos incorporados em ferramentas de análise de dados, como o Power BI.

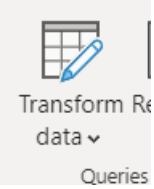
O processo de ETL é uma etapa fundamental no ciclo de vida de um projeto de dados. Ele envolve a coleta, transformação e carregamento de dados de diversas fontes num formato adequado para análise e armazenamento. A seguir, destacam-se as etapas principais do ETL:

2.1 Extração (Extraction)



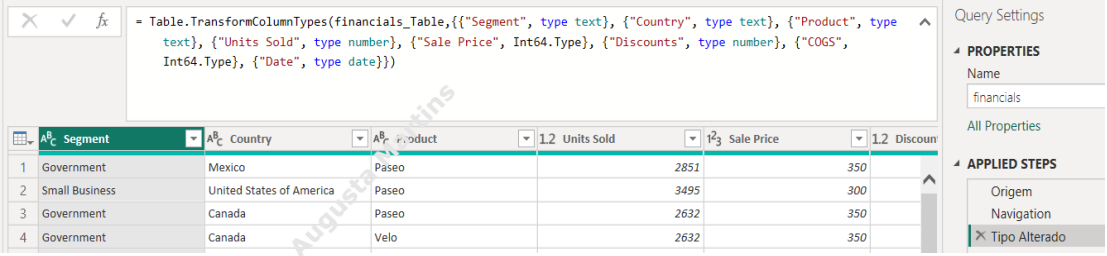
- **Identificação das fontes de dados** - Determine quais as fontes de dados que serão utilizadas no processo de ETL, como base de dados, ficheiros, APIs, entre outros.
- **Definição do âmbito da extração** - Especifique quais os dados que serão extraídos de cada fonte, levando em consideração os requisitos de negócios e análise.
- **Criação de conexões** - Estabeleça as conexões necessárias para aceder às fontes de dados e obter as informações desejadas.

2.2 Transformação (Transformation)



- **Limpeza de dados** - Realize a limpeza dos dados, removendo duplicados, valores inválidos, espaços em branco e outros ruídos.

- **Padronização e formatação** - Normalize os dados para garantir consistência, como padronizar nomes, formatos de data, unidades de medida, etc.



Vá executando os passos/steps, á medida que vai aplicando cada ação o Power Query escreve em linguagem M.

- **Filtragem e seleção -** Aplique filtros para selecionar apenas os dados relevantes para o processo de análise.

Durante o processo de transformação dos dados, é possível aplicar diferentes tipos de filtros para selecionar apenas os dados relevantes para o processo de análise. Alguns dos tipos de filtros comumente usados incluem:

a) Filtro de Valores:

- **Filtragem por valor igual** - Seleciona registos que correspondem a um valor específico numa coluna.
- **Filtragem por valor diferente** - Seleciona registos que não correspondem a um valor específico numa coluna.
- **Filtragem por valor maior ou menor** - Seleciona registos com valores maiores ou menores que um valor especificado numa coluna.
- **Filtragem por intervalo** - Seleciona registos com valores dentro de um intervalo especificado numa coluna.

b) Filtro de Texto:

- **Filtragem por texto contendo** - Seleciona registos em que uma coluna contém um texto específico.
- **Filtragem por texto inicia... ou termina com...** - Seleciona registos em que uma coluna inicia ou termina com um texto específico.
- **Filtragem por texto vazio ou não vazio** - Seleciona registos em que uma coluna está vazia ou contém algum valor.

c) **Filtro de Data e Hora:**

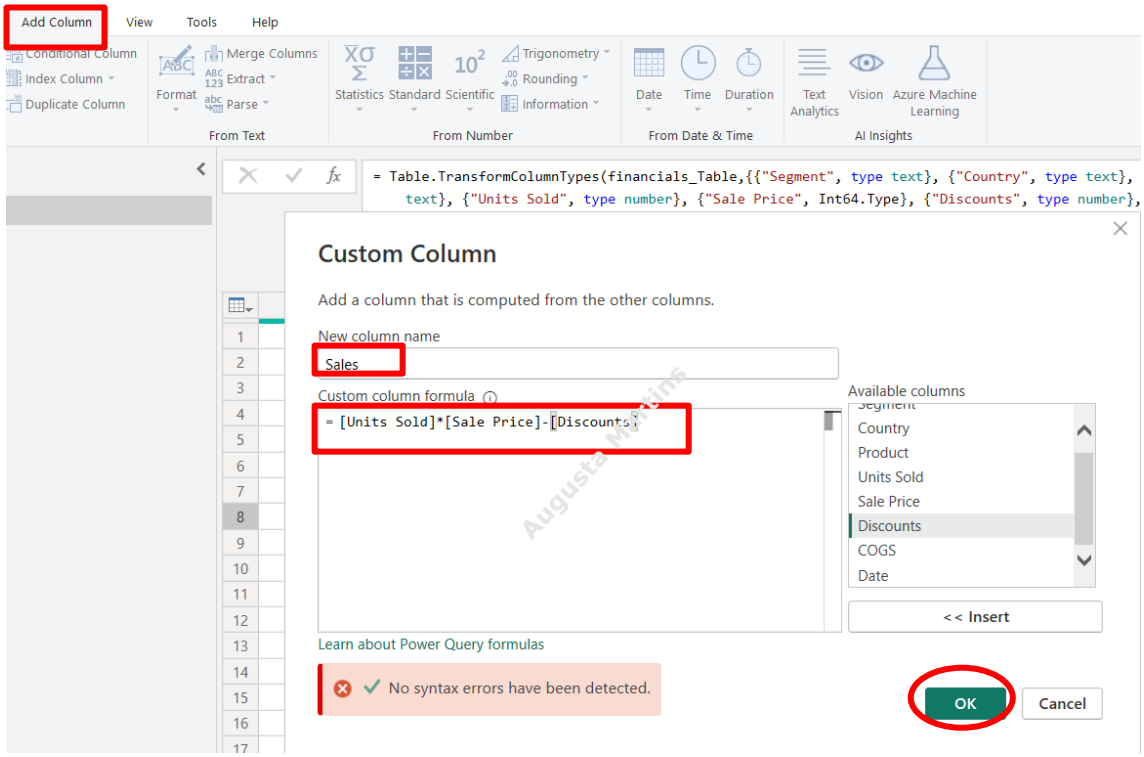
- **Filtragem por data** - Seleciona registos com uma data específica, antes de uma data ou após uma data numa coluna.
- **Filtragem por intervalo de datas** - Seleciona registos com datas dentro de um intervalo especificado numa coluna.
- **Filtragem por parte da data -** (dia, mês, ano) - Seleciona registos com base em partes específicas da data, como o mês ou o ano, numa coluna.

d) **Filtro de Coluna:**

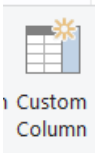
- **Filtragem por coluna calculada** - Seleciona registos com base numa expressão ou fórmula numa coluna calculada.
- **Filtragem por condição lógica** - Seleciona registos com base numa condição lógica, como uma combinação de múltiplos filtros.

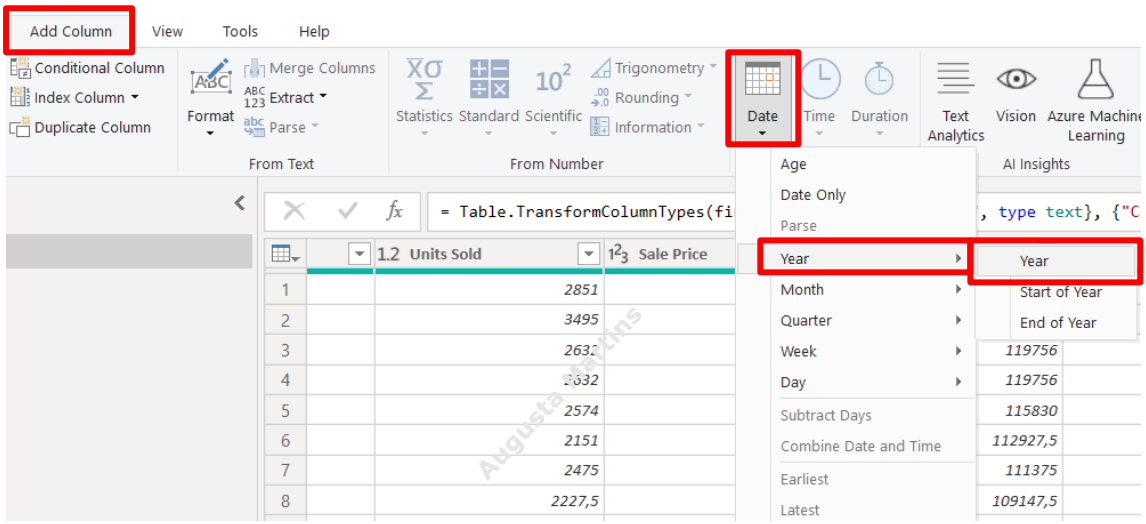
Estes são apenas alguns exemplos dos tipos de filtros que podem ser aplicados durante o processo de transformação dos dados. O Power BI oferece uma ampla gama de opções de filtragem para ajudar a refinar e selecionar os dados necessários para a análise, permitindo que se foque nos aspetos relevantes dos seus dados.

- **Criação de colunas calculadas**: Adicione colunas calculadas com base em fórmulas ou regras específicas para enriquecer os dados.



- Adicione a coluna *Sales*, clique em adicionar coluna/add column
- Clique em Coluna Personalizada/ Custom Column
- Digite o nome da nova coluna "*Sales*"
- Selecione as colunas, digite os operadores e clique em OK

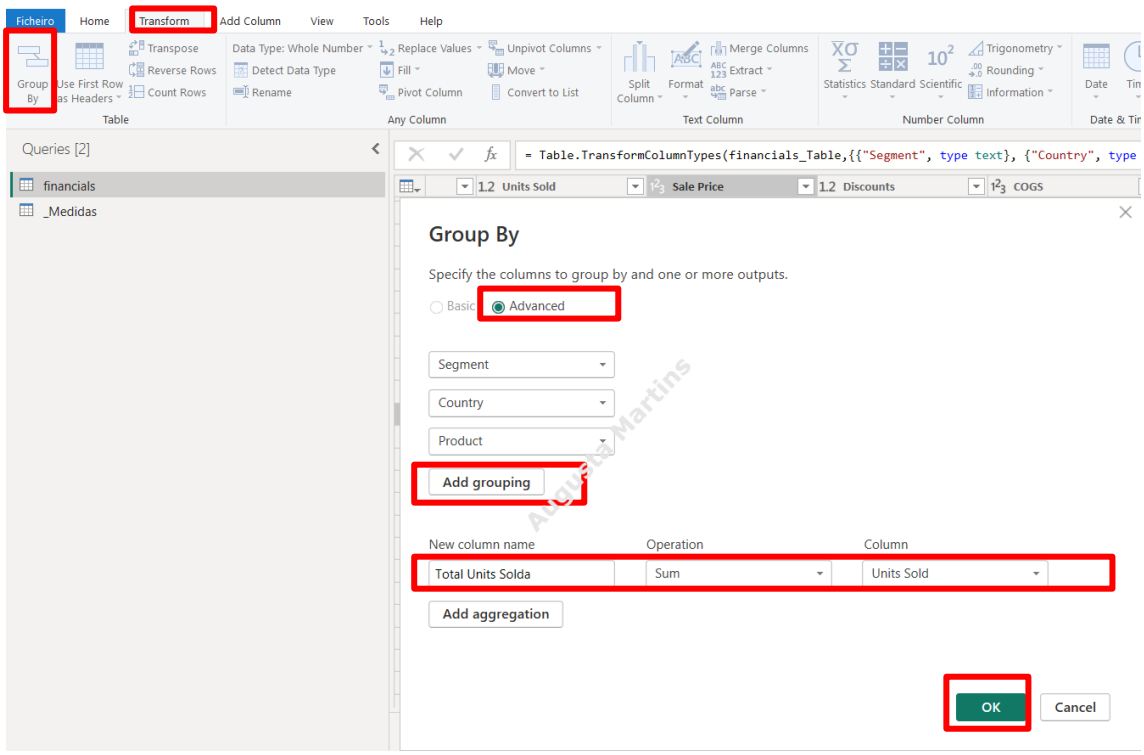




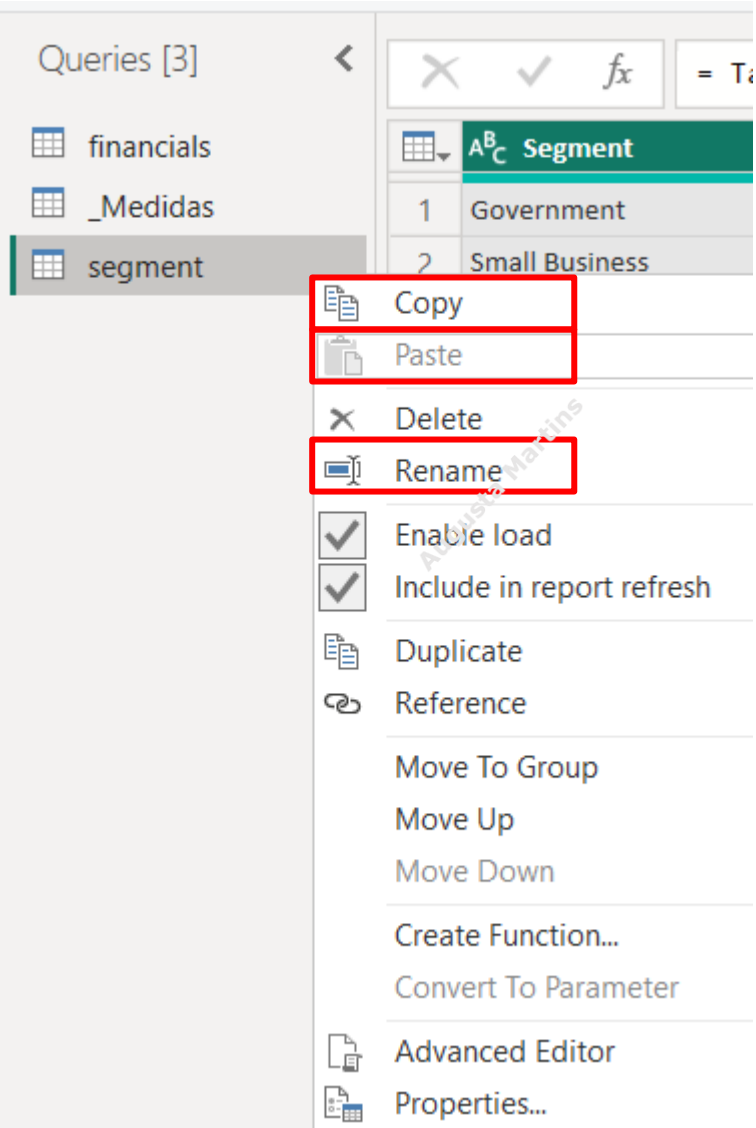
- Adicione a coluna Year, clique em adicionar coluna/Add Column
- Clique em Data/date

- Selecione Year e novamente Year.

- **Agregação e resumo** - Realize operações de agregação para consolidar os dados em níveis mais altos de granularidade, como soma, média, máximo, mínimo, etc.

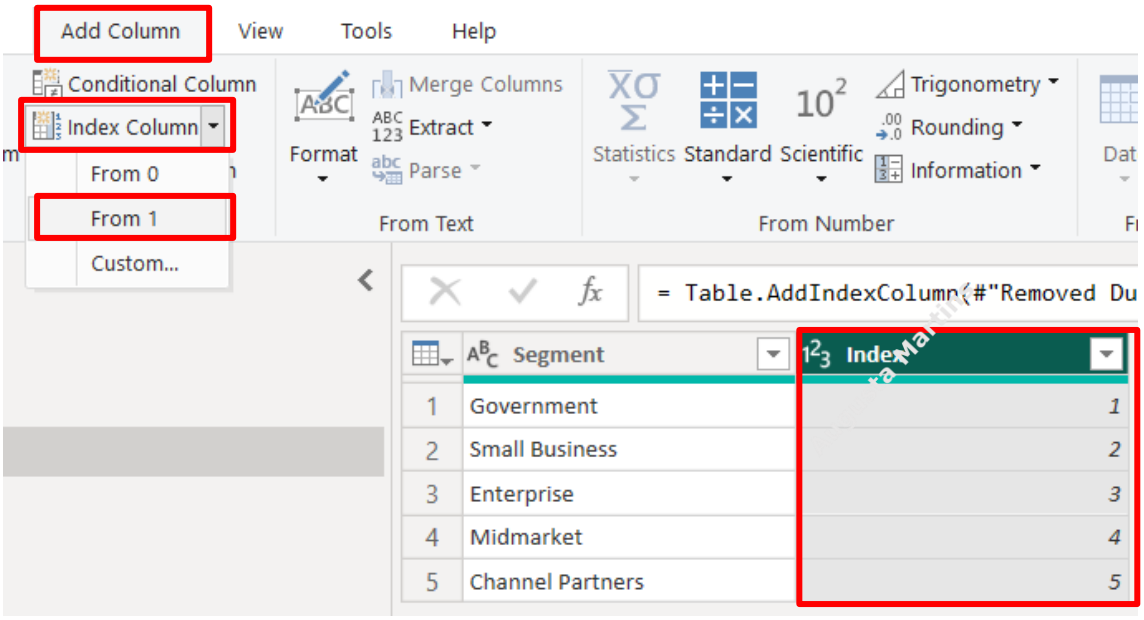


- **União e junção de tabelas -** Combine diferentes conjuntos de dados por meio de uniões e junções para obter uma visão mais completa e integrada das informações.



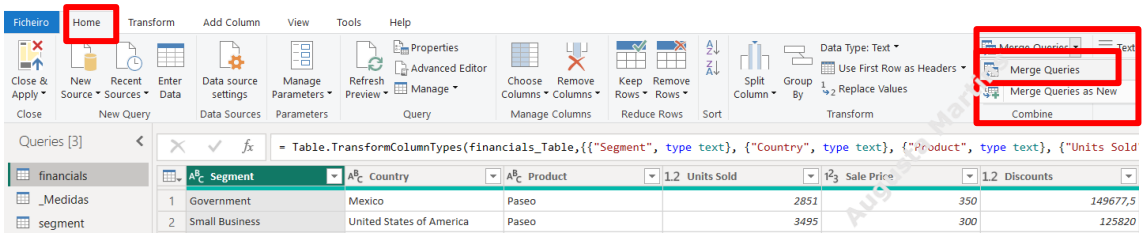
- Clique com o botão direito do rato na consulta *financials*,
- clique em copiar/copy,
- clique novamente com o botão direito do rato em colar / Paste.
- Botão direito do rato, mudar nome para *Segment*.
- Selecione a coluna / Separador Base/ Home

- Clique em adicionar colunas / Add Column
- Selecione Index Column / From 1



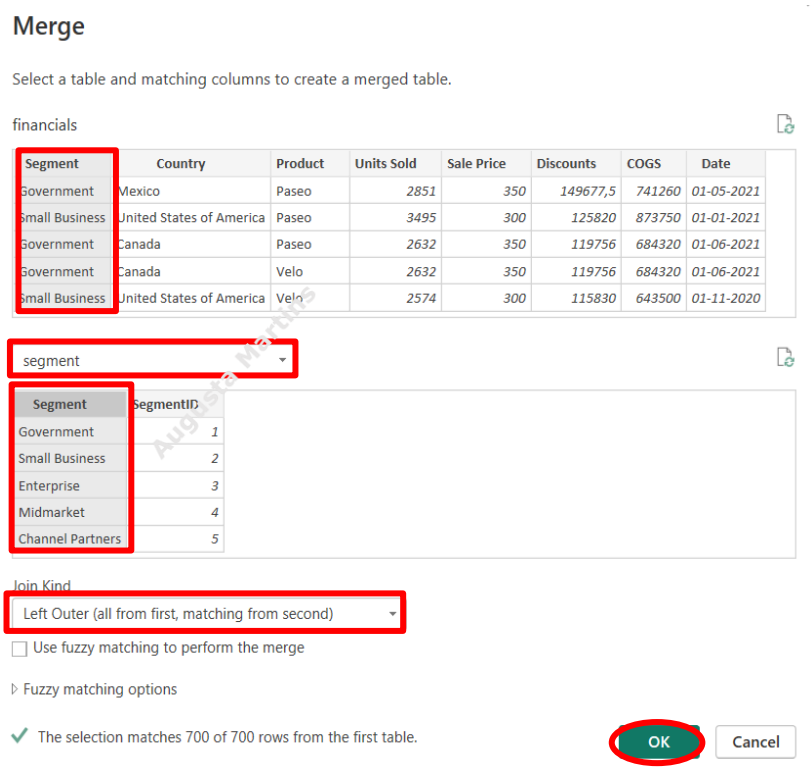
De seguida, duplo clique no nome e renomeie para *Segment*ID.

Execute o combinar intercalar index da tabela *Segment* para a tabela *financials*.

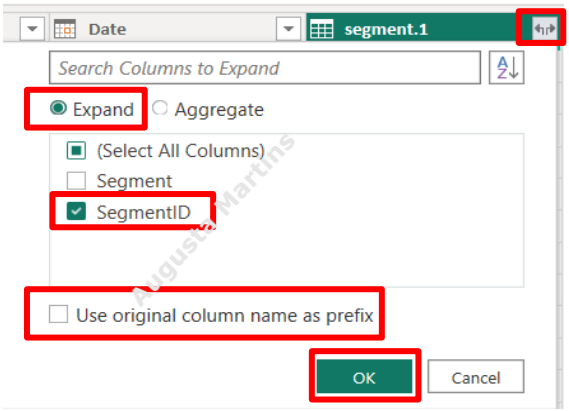


- Selecione Separador Base/Home
- Grupo Combinar/Combine
- Selecione intercalar/Merge Queries

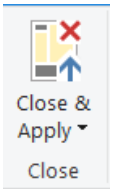
- Clique na coluna *Segment* da tabela *financials*, para ficar cinza
- Clique na seta e escolha a tabela *Segment*
- Clique na coluna *Segment* da tabela *Segment*, para ficar cinza
- Selecione Left Outer
- Clique em OK



De seguida clique nas setas para expandir, e trazer a coluna *Segment*ID. De seguida remova a coluna *Segment*



2.3 Carga (Loading)



- **Criação do esquema de dados** - Defina a estrutura e o esquema da tabela ou do modelo de dados que receberá os dados transformados.
- **Mapeamento e correspondência** - Mapeie as colunas dos dados transformados para as colunas correspondentes na tabela de destino.
- **Carregamento dos dados** - Insira os dados transformados na tabela de destino ou no armazenamento de dados, seja um Base de Dados, um data warehouse ou uma ferramenta de análise.
- **Atualização incremental** - Em algumas situações, é necessário realizar atualizações periódicas ou incrementais dos dados, adicionando apenas registos novos ou atualizados desde a última carga.

É importante ressaltar que o processo de ETL pode ser complexo e envolver várias etapas adicionais, dependendo da natureza e complexidade dos dados. Também é recomendado realizar testes e validações ao longo de cada etapa para garantir a qualidade e a integridade dos dados.

2.4 Função Calculate

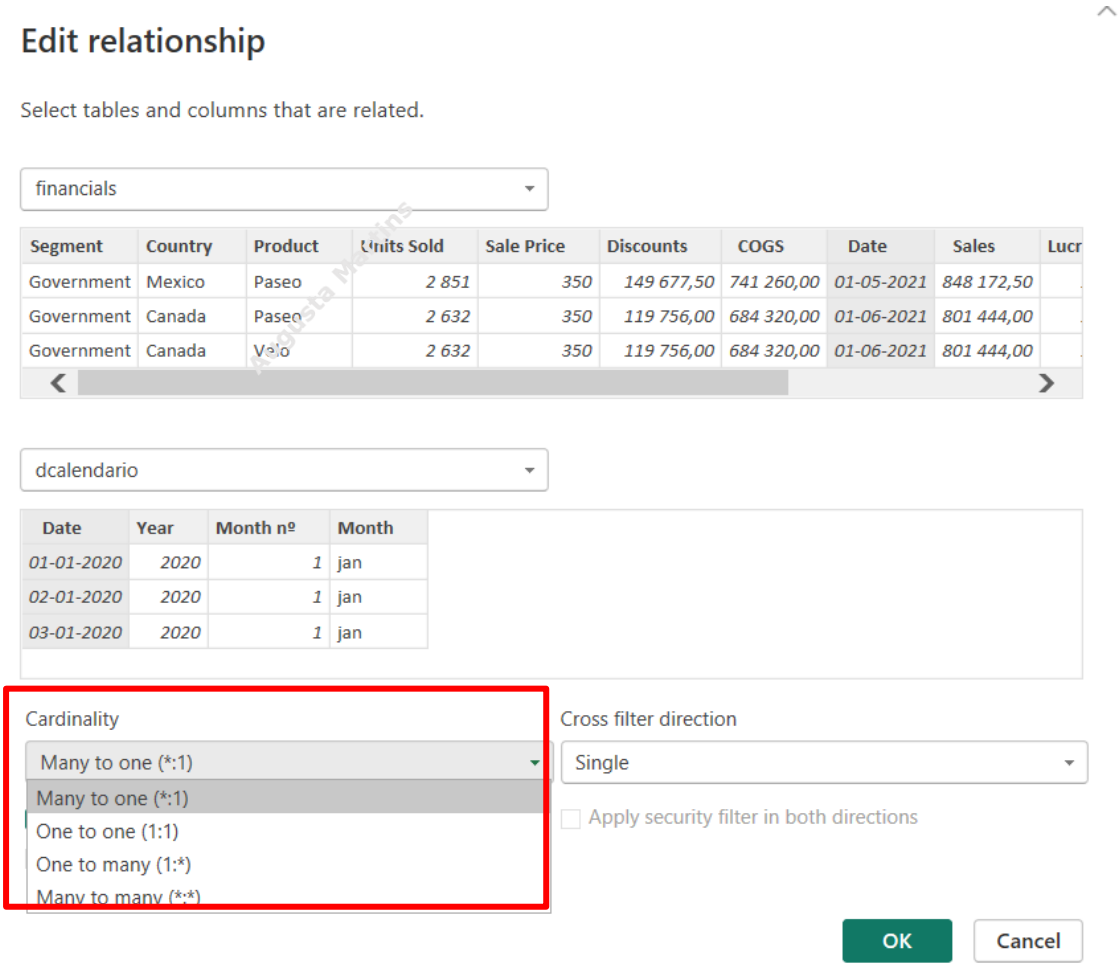
A função CALCULATE é uma função poderosa no DAX (Data Analysis Expressions) utilizada para modificar o contexto de cálculo em fórmulas. Permite que crie medidas personalizadas que respondam a diferentes filtros e condições específicas. A função CALCULATE pode ser usada para filtrar dados, adicionar ou remover filtros, modificar o contexto de linha ou coluna e realizar cálculos condicionais. É uma função essencial para a criação de medidas avançadas e complexas no Power BI.

2.5 Relações eficazmente

As relações eficazes são fundamentais para a análise de dados numa ferramenta como o Power BI. Estabelecem ligações lógicas entre diferentes tabelas de dados, permitindo a combinação e a análise conjunta desses dados. Para criar relações eficazes, é importante entender a estrutura dos dados, identificar as colunas-chave nas tabelas relacionadas e definir a cardinalidade correta (um-para-um, um-para-muitos ou muitos-para-muitos) entre as tabelas. Relações bem projetadas facilitam a criação de visualizações e análises significativas.

2.5.1 Cardinalidade (Cardinality)

No contexto das relações de dados, a cardinalidade refere-se à natureza da relação entre duas tabelas e indica quantos valores únicos de uma tabela estão associados a um único valor da outra tabela.



Existem três tipos principais de cardinalidades no Power BI:

1. **Relação Um-para-Um:**
   - Cada valor numa coluna de uma tabela está associado a um único valor correspondente em outra coluna de outra tabela.
   - É indicada quando as colunas relacionadas têm uma correspondência única e direta entre si.
   - É representada por uma seta simples no diagrama de relacionamento.

2. **Relação Um-para-Muitos:**
   - Nesta relação, cada valor numa coluna de uma tabela pode estar associado a vários valores correspondentes em outra coluna de outra tabela.
   - É o tipo de relação mais comum e amplamente usado no Power BI.
   - É indicada quando a tabela de "um" contém valores exclusivos e a tabela do "muitos" contém valores relacionados a esses valores exclusivos.
   - É representada por uma seta de cruzamento no diagrama de relacionamento.

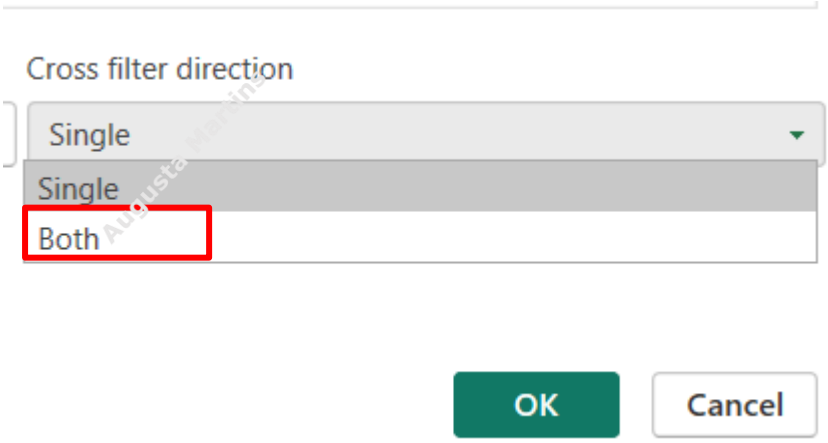
3. **Relação Muitos-para-Muitos:**
   - Nesta relação, vários valores numa coluna de uma tabela podem estar associados a vários valores correspondentes em outra coluna de outra tabela.
   - É indicada quando as tabelas relacionadas têm uma correspondência complexa e múltipla entre si.
   - Geralmente, requer a criação de uma tabela intermedia, conhecida como tabela de junção ou tabela ponte, para conectar as duas tabelas principais.
   - É representada por duas setas de cruzamento no diagrama de relacionamento.

A escolha do tipo de relação adequado depende da estrutura dos dados e da lógica de negócios subjacente. É essencial compreender a natureza dos dados e a relação entre as tabelas para definir as relações corretas. Uma vez estabelecidas as relações eficazes, é possível combinar, filtrar e analisar dados de várias tabelas, criando visualizações e análises poderosas no Power BI.

2.5.2 Filtro cruzado (Cross Filter Direction)

A ligação de filtro cruzado, também conhecida como filtro cruzado, é um recurso do Power BI que permite que filtros aplicados numa visualização afetem outras visualizações relacionadas. Quando interage com uma visualização e aplica um filtro específico, as outras visualizações ligadas são automaticamente atualizadas para mostrar apenas os dados relevantes com base no filtro aplicado.

O objetivo da ligação de filtro cruzado é permitir uma exploração interativa dos dados. Ajuda os utilizadores a analisar e compreender melhor os dados, proporcionando uma visão mais detalhada dos diferentes aspetos e relacionamentos entre os elementos de dados. Com a filtragem cruzada, é possível ver instantaneamente como as alterações num filtro afetam os resultados em outras visualizações, facilitando a análise comparativa, a identificação de tendências e a tomada de decisões informadas.



Existem dois tipos de filtros cruzados no Power BI:

1. **Filtro Cruzado Bidirecional (Both):**
   - Este tipo de filtro cruzado permite que a filtragem ocorra tanto da visualização de origem para a visualização de destino quanto da visualização de destino para a visualização de origem.
   - Isto significa que, ao interagir com uma visualização, os filtros são aplicados em ambas as direções, afetando as visualizações relacionadas em ambas as direções. **Tenha cuidado em aplicar, utilize apenas quando tem a certeza do que pretende. Pode levar a ambiguidades com outras tabelas.** Pode mais tarde utilizar funções.

2. **Filtro Cruzado Unidirecional (Single):**
   - Este tipo de filtro cruzado permite que a filtragem ocorra apenas da visualização de origem para a visualização de destino, e não o contrário.
   - Ao interagir com uma visualização, os filtros são aplicados apenas nas visualizações relacionadas que estão abaixo ou à direita da visualização de origem.
   - A escolha do tipo de filtro cruzado depende das necessidades e da lógica de análise dos dados. O Power BI oferece flexibilidade para definir as ligações de filtro cruzado e ajustar as configurações de acordo com os requisitos específicos de análise.

2.6 Medidas semi-aditivas

As medidas semi-aditivas são medidas que podem ser agregadas em algumas dimensões, mas não em outras. O objetivo das medidas semi-aditivas é calcular valores agregados ou resumidos de forma adequada em determinadas dimensões, levando em consideração a natureza dos dados e as necessidades de análise. Em situações em que certas medidas não podem ser simplesmente somadas ou médias em todas as dimensões, as medidas semi-aditivas oferecem uma solução para calcular valores precisos.

Nestes casos, é necessário utilizar funções DAX específicas, como a função SUMX, juntamente com lógica condicional, para calcular corretamente as medidas semi-aditivas.

Com base nas tabelas fornecidas, podemos dar exemplos de medidas semi-aditivas usando a medida "*Units Sold*" da tabela "*financials*". Suponha que deseja calcular a soma das unidades vendidas por mês, mas não faz sentido somar esses valores num nível superior, como o ano.

Para criar uma medida semi-aditiva, podemos usar a função SUMX juntamente com a função FILTER para aplicar a lógica condicional necessária. A seguir está um exemplo de medida semi-aditiva para calcular as unidades vendidas por mês:

```
Units Sold (Semi-Aditiva) = SUMX(
                                FILTER('financials',
                                        ISFILTERED('dcalendario'[Month])),
                                                'financials'[Units Sold] )
```

Neste exemplo, a função FILTER é usada para verificar se a dimensão 'dcalendario'[Month] está a ser filtrada. Se sim, a função SUMX calcula a soma das unidades vendidas apenas nos meses em que há filtragem. Isto garante que a medida seja corretamente agregada no nível de mês, mas não seja agregada em níveis superiores, como o ano.

```
Units Sold (Semi-Aditiva) =
SUMX(
    FILTER('financials', ISFILTERED('dcalendario'[Month])),
    'financials'[Units Sold]
)
```

Year	Total Units	Units Sold (Semi-Aditiva)
2020	**264.674**	
dez	52.970	52.970
nov	65.481	65.481
out	95.622	95.622
set	50.601	50.601
2021	**861.132**	
Total	**1.125.806**	

Esta é apenas uma ilustração básica de uma medida semi-aditiva. Dependendo dos requisitos específicos, pode ser necessário adicionar mais lógica condicional ou usar outras funções DAX para calcular adequadamente as medidas semi-aditivas, levando em consideração as dimensões relevantes e a agregação desejada.

3. Introdução às medidas DAX no Power BI Desktop

O DAX (Data Analysis Expressions) é uma linguagem de fórmula usada no Power BI Desktop para criar cálculos personalizados e medidas. Permite a criação de fórmulas para realizar cálculos complexos, criar medidas agregadas, filtrar dados e muito mais.

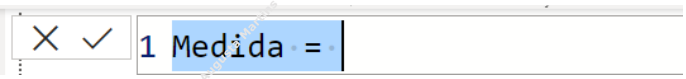
Foi desenvolvido pela Microsoft e é projetado para ser usado em conjunto com o modelo de dados do Power BI.

3.1 Cálculos

As fórmulas DAX são usadas em **medidas**, **colunas calculadas**, **tabelas calculadas**.

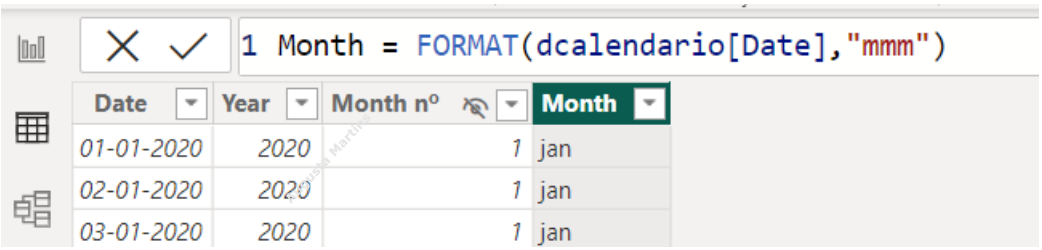
3.1.1 Medidas

Medidas são fórmulas de cálculo dinâmicas onde os resultados mudam dependendo do contexto. As medidas são usadas em relatórios que dão suporte à combinação e filtragem de dados de modelo usando vários atributos, como um relatório do Power BI ou Tabela Dinâmica ou Gráfico Dinâmico do Excel. As medidas são criadas usando a barra de fórmulas DAX.

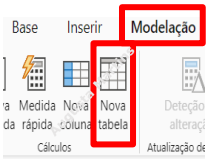


3.1.2 Colunas calculadas

É uma coluna que adiciona a uma tabela existente e, em seguida, cria uma fórmula DAX que define os valores da coluna. Quando uma coluna calculada contém uma fórmula DAX válida, os valores são calculados para cada linha assim que a fórmula é inserida. **Os valores são então armazenados no modelo de dados.**



3.1.3 Tabelas calculadas



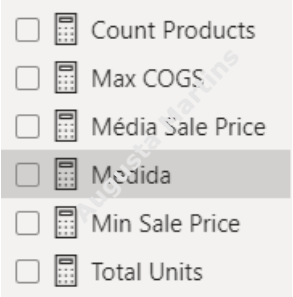
É um objeto calculado, com base numa expressão de fórmula, derivado de todas ou parte de outras tabelas no mesmo modelo. Em vez de consultar e carregar valores nas colunas da nova tabela a partir de uma fonte de dados, uma fórmula DAX define os valores da tabela.

```
dcalendario = CALENDARAUTO()
```

As tabelas calculadas também são úteis ao configurar um conjunto de linhas filtrado ou um subconjunto ou superconjunto de colunas de outras tabelas existentes. Isto permite que mantenha a tabela original intacta enquanto cria variações dessa tabela para oferecer suporte a cenários específicos.

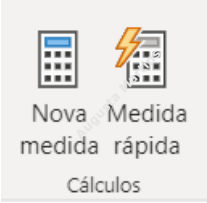
As tabelas calculadas oferecem suporte a relacionamentos com outras tabelas. As colunas na tabela calculada têm tipos de dados, formatação e podem pertencer a uma categoria de dados. As tabelas calculadas podem ser nomeadas e mostradas ou ocultas como qualquer outra tabela. As tabelas calculadas são recalculadas se qualquer uma das tabelas das quais extrai dados for atualizada.

3.2 Funções DAX



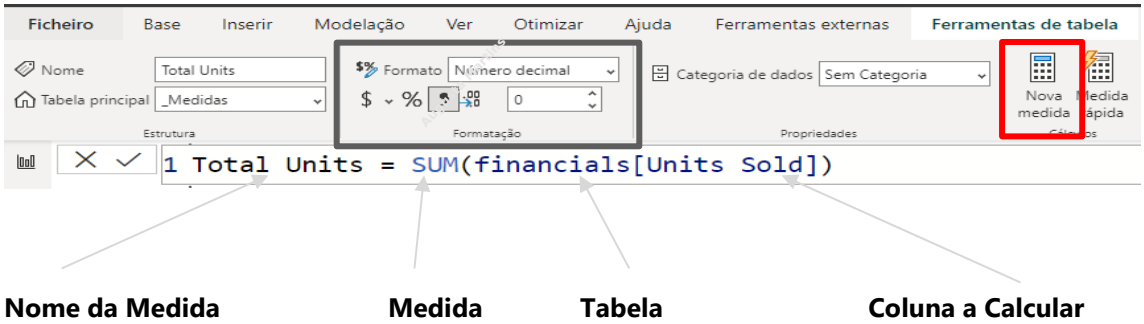
Ao criar um modelo de dados no Power BI Desktop, pode usar o DAX para criar medidas que realizam cálculos específicos. Por exemplo, pode criar uma medida que calcule a soma de Units Sold, a média de *Sale Price* ou o

máximo de *COGS*. Estas medidas podem ser usadas para criar visualizações interativas e dashboards de controlo.



3.3 Funções de Agregação (Simples)

Para utilizar o DAX no Power BI Desktop, clique numa nova medida, pode aceder a barra de fórmula na parte superior do ecrã e começar a digitar as expressões DAX. A medida ou a coluna calculada será atualizada em tempo real à medida que digita a fórmula, facilitando a criação e a depuração das expressões.



Nome da Medida **Medida** **Tabela** **Coluna a Calcular**

No grupo formatação pode optar por escolher o formato.

3.3.1 SUM

A função **SUM** no Power BI é utilizada para calcular a soma dos valores numa coluna numérica ou numa expressão numérica. O seu objetivo principal é obter o total dos valores numa coluna específica, permitindo realizar cálculos de agregação.

A função SUM é especialmente útil para realizar análises que envolvem a soma de valores, como total de vendas, receitas, despesas, entre outros. Permite consolidar os dados e obter a soma total de uma métrica numa coluna.

A sintaxe é a seguinte:

```
SUM(table[column])
```

Exemplo de aplicação:

```
Total Units = SUM(financials[Units Sold])
```

Esta fórmula calcula a soma da coluna "*Units Sold*" na tabela "*financials*", resultando na quantidade total de unidades vendidas.

Na imagem seguinte poderá analisar a medida *"Total Units"*, nos diferentes contextos de análise: Total Units por *Country*, Total Units por *Product* e Total Units por year.

Country	Total Units
Canada	790.544
France	632.172
Germany	711.480
Mexico	594.195
United States of America	636.053
Total	**3.364.443**

Product	Total Units
Amarilla	409.181
Carretera	480.494
Montana	480.060
Paseo	1.043.518
Velo	474.659
VTT	476.533
Total	**3.364.443**

Year	Total Units
2020	839.165
2021	881.150
2022	959.240
2023	684.888
Total	**3.364.443**

Exercício de aplicação:

Calcule a Soma da coluna *COGS*. (Total dos custos dos produtos vendidos)

A função SUM também pode ser usada em combinação com outras funções e operações para realizar cálculos mais complexos e análises. É uma das funções fundamentais para calcular totais e realizar agregações no Power BI.

É importante notar que a função SUM é aplicada apenas a colunas numéricas ou expressões que resultem em valores numéricos. Se aplicada a colunas que contenham valores não numéricos, a função retornará um resultado inválido ou nulo.

3.3.2 AVERAGE

A função **AVERAGE** tem como objetivo calcular a média aritmética de um conjunto de valores. A função é amplamente utilizada para obter uma medida de tendência central, representando um valor médio num conjunto de dados.

Fornecer uma estimativa do valor médio de uma determinada coluna ou expressão numa tabela. Isto pode ser útil para entender a distribuição dos dados, comparar valores em relação à média e realizar análises estatísticas básicas.

É especialmente útil quando se trabalha com dados numéricos, como vendas, quantidades, preços, pontuações, entre outros. Ao calcular a média desses valores, é possível obter uma noção de um ponto de referência que representa o valor médio do conjunto de dados.

A sintaxe é a seguinte:

```
AVERAGE(table[column])
```

Exemplo de aplicação:

```
Média Sale Price = AVERAGE(financials[Sale Price])
```

Esta fórmula calcula a média da coluna "*Sale Price*" na tabela "*financials*", resultando no preço médio de venda por *Country*.

Country	Total Units	Média Sale Price
Canada	790.544	116,11
France	632.172	97,51
Germany	711.480	102,86
Mexico	594.195	108,81
United States of America	636.053	105,29
Total	**3.364.443**	**106,42**

Augusta Martins

Exercício de aplicação:

- Calcule a média da coluna *COGS* da tabela *financials*. (média dos custos dos produtos vendidos)
- Calcule a média da coluna Units Sold da tabela *financials*.

No Power BI, a função AVERAGE pode ser usada de diferentes formas, como em colunas calculadas, medidas ou em fórmulas DAX mais complexas que envolvam agregações. É uma função simples, porém fundamental, para realizar cálculos estatísticos e analisar a tendência central dos dados num conjunto de informações.

3.3.3 MAX

A função **MAX** é usada para obter o valor máximo de uma coluna ou expressão numa tabela. É útil para identificar o maior valor presente num conjunto de dados.

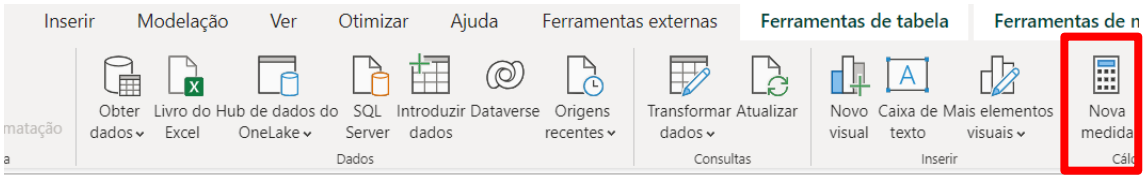
A sintaxe é a seguinte:

```
MAX(table[column])
```

Exemplo de aplicação:

```
Max COGS = MAX(financials[COGS])
```

Esta fórmula encontra o valor máximo na coluna "*COGS*" na tabela "*financials*", resultando no custo máximo.




```
Max COGS = MAX(financials[COGS])
```

Country	Total Units	Max COGS
Canada	790.544	950.625
France	632.172	747.760
Germany	711.480	771.160
Mexico	594.195	771.160
United States of America	636.053	897.000
Total	**3.364.443**	**950.625**

Exercício de aplicação:

- Calcule o Maior número de unidades vendidas (Units Sold)
- Calcule o Preço Máximo (*Sale Price*)

3.3.4 MIN

A função **MIN** é usada para obter o valor mínimo de uma coluna ou expressão numa tabela. É útil para identificar o menor valor presente num conjunto de dados. A sintaxe é a seguinte:

```
MIN(table[column])
```

Exemplo de aplicação:

```
Min Sales = MIN(financials[Sales])
```

Esta fórmula encontra o valor mínimo na coluna "*Sales*" na tabela "*financials*", resultando na venda mínima.

Country	Total Units	Min Sales
Canada	790.544	1655
France	632.172	1764
Germany	711.480	1686
Mexico	594.195	2509
United States of America	636.053	1731
Total	**3.364.443**	**1655**

Exercício de aplicação:

- Calcule a menor quantidade de unidades vendidas (Units Sold)
- Calcule o custo mínimo (*COGS*)

As funções MAX e MIN são úteis para análise e compreensão dos extremos (máximo e mínimo) dos valores numa determinada coluna. Permitem identificar os valores máximos e mínimos, o que pode ser útil para avaliar a variabilidade dos dados, detetar valores atípicos ou definir referências para análises comparativas.

É importante ressaltar que estas funções podem ser usadas em combinação com outras funções ou com o uso de filtros para obter resultados mais específicos ou condicionais, adaptando-as às necessidades analíticas do seu projeto no Power BI.

3.3.5 COUNT

A função **COUNT** é usada para contar o número de valores numéricos numa coluna ou expressão. Ela exclui valores em branco (NULL) e não considera valores não numéricos. O objetivo da função COUNT é fornecer uma contagem dos valores válidos numa determinada coluna. A sintaxe é a seguinte:

```
COUNT(table[column])
```

Exemplo de aplicação:

```
Count Products = COUNT(financials[Product])
```

Esta fórmula conta o número de produtos na coluna "*Product*" na tabela "*financials*".

Country	Total Units	Count Products
Canada	790.544	489
France	632.172	392
Germany	711.480	508
Mexico	594.195	436
United States of America	636.053	420
Total	**3.364.443**	**2245**

Exercício de aplicação:

- Contar o número de vendas registadas na coluna (Units Sold)

3.3.6 COUNTA

A função **COUNTA** é usada para contar o número de valores não nulos numa coluna ou expressão. Diferentemente da função COUNT, ela considera valores não numéricos, como texto e datas. O objetivo da função COUNTA é fornecer uma contagem de todos os valores, incluindo nulos. A sintaxe é a seguinte:

```
COUNTA(table[column])
```

Exemplo de aplicação:

```
Linhas com desconto = COUNTA(financials[Discounts])
```

Esta fórmula conta o número de valores não vazios na coluna "*Discounts*" na tabela "*financials*".

Product	Total Units	Linhas com desconto
Amarilla	409.181	293
Carretera	480.494	332
Montana	480.060	311
Paseo	1.043.518	607
Velo	474.659	355
VTT	476.533	347
Total	**3.364.443**	**2245**

Em resumo, o objetivo da função COUNT é contar o número de valores numéricos numa coluna, enquanto a função COUNTA conta o número total de valores, incluindo valores não numéricos. Estas funções são úteis para determinar a quantidade de registos ou valores presentes numa coluna específica e podem auxiliar em análises estatísticas e na validação dos dados num conjunto de informações.

Exercício de aplicação:

- Contar o número de registos na coluna (*Product*)

3.3.7 COUNTROWS

O objetivo da função **COUNTROWS** no Power BI é contar o número de linhas numa tabela ou numa expressão de tabela. Esta função é útil para determinar a quantidade de registos ou linhas presentes numa tabela específica.

A sintaxe é a seguinte:

```
COUNTROWS(table)
```

Exemplo de aplicação:

```
Count Rows = COUNTROWS(financials)
```

Esta fórmula conta o número total de linhas na tabela "*financials*".

Count Rows
2245

A função COUNTROWS pode ser usada em conjunto com outras funções e operações de filtragem para obter resultados mais específicos. Ela fornece uma maneira simples e direta de contar o número de linhas numa tabela ou numa expressão de tabela, o que pode ser útil em várias análises e cálculos no Power BI. (ver 3.6.2)

É importante destacar que a função COUNTROWS conta apenas o número de linhas e não leva em consideração os valores das colunas individuais. Se deseja contar o número de valores numa coluna específica, é mais adequado utilizar as funções COUNT e COUNTA, conforme explicado anteriormente.

Exercício de aplicação:

- Contar o número de linhas (datas) da dcalendario

3.3.8 DISTINCTCOUNT

O objetivo da função **DISTINCTCOUNT** no Power BI é contar o número de valores distintos numa coluna ou expressão. Esta função é útil quando se deseja determinar a quantidade de valores únicos numa coluna, excluindo duplicados. A sintaxe é a seguinte:

```
DISTINCTCOUNT(table[column])
```

Exemplo de aplicação:

```
Distinctcount Countries = DISTINCTCOUNT(financials[Country])
```

Esta fórmula conta o número de países distintos na coluna "*Country*" na tabela "*financials*". Ela contará apenas os países únicos e ignorará duplicados.

Distinctcount Countries
5

Exercício de aplicação:

- Contar o número de produtos distintos na coluna (*Product*).
- Contar o número de produtos distintos por *Segmento* (*Segment*).

A função DISTINCTCOUNT é especialmente útil quando se deseja obter a contagem de valores únicos numa coluna, seja para fins de análise, *Segmentação* ou agrupamento. Ela permite identificar a diversidade dos valores presentes numa coluna, sem considerar duplicados.

É importante ressaltar que a função DISTINCTCOUNT deve ser usada em colunas ou expressões que contenham valores distintos. Se aplicada em colunas que contenham apenas valores nulos ou duplicados, a função retornará o número de registos distintos, incluindo os nulos.

3.3.9 DISTINCT

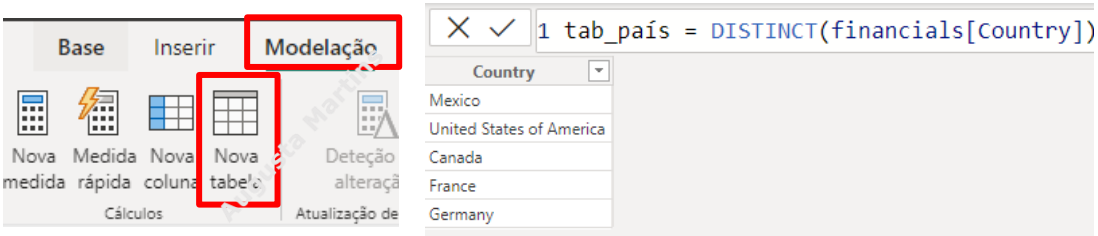
No Power BI, a função **DISTINCT** é usada para devolver uma tabela contendo os valores distintos de uma coluna ou expressão. O objetivo desta função é identificar os valores únicos presentes numa coluna ou expressão específica.

Exemplo de aplicação da função DISTINCT com base na tabela "*financials*":

Sintaxe:

```
DISTINCT(<coluna ou expressão>)
```

Exemplo de aplicação:



Esta fórmula devolverá uma tabela contendo os países distintos presentes na coluna "*Country*" da tabela "*financials*". **Elimina qualquer duplicado e apresenta apenas os valores únicos.**

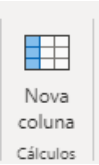
Exercício de aplicação:

- Devolve os produtos distintos na coluna (*Product*), da tabela *financials*

Esta fórmula devolverá uma tabela contendo os produtos distintos presentes na coluna "*Product*" da tabela "*financials*". **Remove qualquer duplicado e apresenta apenas os valores únicos.**

A função DISTINCT é útil quando se deseja obter uma lista de valores distintos para análise, agrupamento ou outras operações. Permite identificar os valores únicos presentes numa coluna ou expressão, ajudando a entender a diversidade dos dados e a realizar cálculos com base nesses valores.

3.4 Colunas Calculadas



As colunas calculadas são colunas adicionadas a uma tabela no Power BI que são calculadas com base em fórmulas ou expressões definidas pelo utilizador. Ao contrário das colunas físicas que armazenam valores diretamente na tabela, as colunas calculadas são calculadas dinamicamente durante a execução das consultas ou interações com os dados.

O objetivo das colunas calculadas no Power BI é permitir a criação de novos campos ou informações derivadas a partir dos dados existentes na tabela. Estas colunas podem ser criadas para realizar cálculos, aplicar lógica condicional, realizar formatação de dados, combinar informações de diferentes colunas, entre outras tarefas.

Uma coluna é calculada com base numa expressão ou fórmula DAX que utiliza os valores de outras colunas na tabela. As colunas calculadas são úteis quando precisa de adicionar uma nova informação derivada dos dados existentes.

Exemplo de algumas colunas calculadas:

- ***Sales*** - Uma coluna calculada que multiplica as colunas "Units Sold" (Unidades Vendidas), "*Sale Price*" (Preço de Venda) e subtrai Discounts para obter o valor total de vendas.

```
1 Sales = financials[Units Sold]*financials[Sale Price]-financials[Discounts]
```

Country	Product	Units Sold	Sale Price	Discounts	COGS	Date	Sales

- **Mês** - Uma coluna calculada que extrai o mês da coluna "Date", permitindo a análise e agrupamento dos dados por mês.

```
1 Month nº = month(dcalendario[Date])
```

```
1 Month = FORMAT(dcalendario[Date],"mmm")
```

Augusta Martins

- **Desc %** - Uma coluna calculada que calcula o percentual de desconto com base na coluna "Discounts" e "*Sales*".

```
1 Desc % = DIVIDE(financials[Discounts],financials[Sales])
```

Country	Product	Units Sold	Sale Price	Discounts	COGS	Date	Sales	Lucro Bruto	Mês	Desc %
Mexico	Paseo	2 851	350	149 677,50	741 260,00	01-05-2021	848 172,50	106 912,50	5	17,65%

- **Calcular o Lucro Bruto -** Suponha que queira adicionar uma coluna que calcule o lucro bruto para cada venda. Clique em nova coluna e construa uma coluna calculada que subtrai o custo dos produtos vendidos (*COGS*) da coluna "*Sales*" para obter o lucro bruto. A fórmula será

```
1 Lucro Bruto = financials[Sales] - financials[COGS]
```

Country	Product	Units Sold	Sale Price	Discounts	COGS	Date	Sales	Lucro Bruto

Esta fórmula subtrai o valor da coluna "*COGS*" do valor da coluna "*Sales*" para obter o lucro bruto.

- **Calcular a Venda Líquida** - Suponha que pretende adicionar uma coluna que calcule o preço de venda líquido após aplicar descontos. A fórmula será:

```
1 Venda liquida = financials[Sales] - financials[Discounts]
```

Country	Product	Units Sold	Sale Price	Discounts	COGS	Date	Sales	Lucro Bruto	Desc %	Venda liquida
Mexico	Paseo	2 851	350	149 677,50	741 260,00	01-05-2021	848 172,50	106 912,50	17,65%	698 495,00

Esta fórmula subtrai o valor da coluna "Discounts" do valor da coluna "*Sales*" para obter a venda líquido.

- **Criar uma Coluna de Ano -** Suponha que pretende adicionar uma coluna que extraia o ano na tabela dcalendario da coluna "Date". A fórmula será:

```
Year = year(dcalendario[Date])
```

Esta fórmula usa a função YEAR para extrair o ano da coluna "Date" e adiciona Esta informação numa nova coluna chamada "Year".

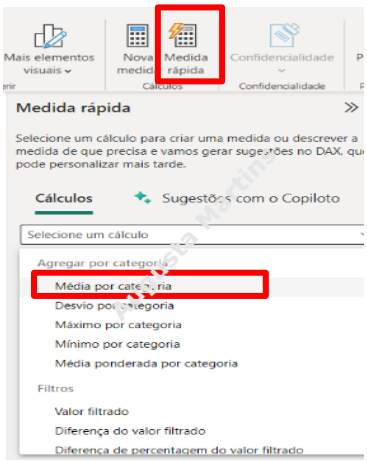
Estes são apenas alguns exemplos de colunas calculadas que pode criar com base na tabela "*financials*". As colunas calculadas permitem que agregue informações adicionais e realize cálculos personalizados com base nos dados existentes.

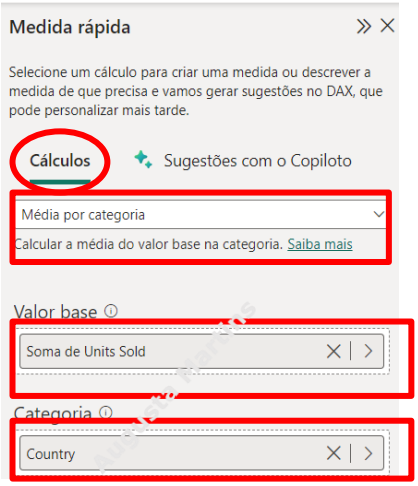
3.5 Medidas rápidas

No contexto do DAX no Power BI, as "medidas rápidas" (*quick measures*) são um recurso que permite aos utilizadores criar cálculos comuns de forma rápida e fácil, sem precisar escrever manualmente as expressões DAX. Elas são pré-definidas e abrangem várias operações comuns, como somas, médias, máximos, mínimos e muito mais.

As medidas rápidas podem ser encontradas na guia "*Modeling*" (Modelagem) no friso de opções do Power BI Desktop. Ao selecionar a opção "*New Quick Measure*" (Nova medida rápida), uma janela será aberta com várias opções pré-configuradas para diferentes tipos de cálculos.

Selecione a opção medida rápida, escolha o cálculo, adicione os campos a analisar, exemplo a média das *units sold* por *Country*.





Clique em adicionar

Resultado:

```
Units Sold média por Country =
AVERAGEX(
    KEEPFILTERS(VALUES('financials'[Country])),
    CALCULATE(SUM('financials'[Units Sold]))
)
```

Nota: Mais á frente no DAX irá compreender melhor esta medida.

Augusta Martins

3.6 DAX Avançado

O DAX (*Data Analysis Expressions*) é uma linguagem de fórmula usada no *Power BI* e em outras ferramentas da Microsoft, como o *Power Pivot, Power Query e Analysis Services*. O DAX é projetado especificamente para a criação de fórmulas e expressões que permitem a análise de dados e a criação de cálculos personalizados.

No contexto do Power BI, o DAX é usado para criar medidas (*measures*), colunas calculadas (*calculated columns*), tabelas calculadas (*calculated tables*) e expressões que manipulam e transformam os dados. Ele oferece uma ampla gama de funções e operadores que permitem a realização de cálculos avançados, agregações, filtragens, classificações e muito mais.

O DAX é uma linguagem baseada em fórmulas, similar ao Excel, mas com recursos adicionais e adaptada para trabalhar com conjuntos de dados maiores e mais complexos. Ele permite a criação de cálculos dinâmicos e interativos, aproveitando as relações entre as tabelas e a estrutura do modelo de dados no Power BI.

Algumas características e capacidades do DAX incluem:

- **Cálculos contextuais** - O DAX é capaz de aplicar cálculos de acordo com o contexto do relatório ou visualização, levando em consideração filtros, *Segment*ações e hierarquias.
- **Funções agregadas** - O DAX oferece uma variedade de funções agregadas, como SUM, AVERAGE, MAX, MIN, entre outras, que permitem calcular totais, médias, valores máximos e mínimos.
- **Funções de tempo** - O DAX possui funções específicas para trabalhar com datas e realizar cálculos relacionados a tempo, como cálculos de períodos, comparações entre datas e análise de tendências.

- **Expressões condicionais** - O DAX suporta expressões condicionais, permitindo a criação de cálculos com base em condições lógicas, como IF, SWITCH, e outras funções condicionais.
- **Referências cruzadas** - O DAX permite realizar cálculos que envolvem a referência cruzada de diferentes tabelas, permitindo a criação de fórmulas que combinam informações de várias fontes de dados.

No geral, o DAX é uma linguagem poderosa e flexível que permite a manipulação e análise de dados de forma avançada. Com a sua sintaxe e funções específicas, os utilizadores podem criar medidas personalizadas, colunas calculadas e expressões que atendam às necessidades específicas de análise e relatórios dentro do ambiente do Power BI.

O DAX permite que realize cálculos complexos e avançados (como iremos ver mais à frente), aplicando lógica personalizada aos seus dados. Com esta linguagem, pode criar medidas, colunas calculadas e tabelas calculadas, bem como filtrar, agregar e manipular dados.

3.6.1 CALCULATE

A função **CALCULATE** é uma das funções mais importantes do DAX. Permite modificar o contexto de avaliação das fórmulas, aplicando filtros, alterando relações e adicionando condições específicas para cálculos personalizados.

O objetivo principal da função CALCULATE é criar cálculos condicionais e manipular o contexto de avaliação das expressões no Power BI. Isto significa que permite ajustar os resultados das medidas com base em determinados critérios ou condições específicas.

A sintaxe básica é a seguinte:

```
CALCULATE(<expression>, <filter1>, <filter2>, ...)
```

A expressão é a fórmula ou medida que deseja calcular, e os filtros são as condições ou restrições aplicadas ao contexto de avaliação. Os filtros podem ser expressões lógicas, colunas de uma tabela ou o resultado de outras funções DAX.

Exemplo de aplicação:

```
Total Sales = CALCULATE(SUM(financials[Sales]))
```

Esta fórmula calcula a soma da coluna "*Sales*" na tabela "*financials*" considerando qualquer filtro aplicado.

A função CALCULATE pode ser usada para realizar várias tarefas, incluindo:

1. **Aplicar filtros condicionais** - pode utilizar a função CALCULATE para aplicar filtros específicos aos cálculos. Por exemplo, calcular a soma das vendas apenas para um determinado país ou período.
2. **Alterar o contexto de avaliação** - A função CALCULATE permite modificar o contexto de avaliação das medidas, ignorando ou

substituindo as relações existentes entre tabelas. Isto é útil quando precisa fazer cálculos com base num contexto diferente do padrão.

3. **Criar medidas condicionais** - Com a função CALCULATE, pode criar medidas que sejam calculadas somente quando determinadas condições forem atendidas. Por exemplo, calcular a média das vendas apenas para produtos com desconto.
4. **Combinação de funções** - A função CALCULATE pode ser combinada com outras funções DAX, como FILTER, ALL, ALLEXCEPT, entre outras, para realizar cálculos mais avançados e complexos.

A função CALCULATE é extremamente versátil e permite uma ampla gama de manipulações de contexto e cálculos condicionais. Desempenha um papel fundamental na criação de análises personalizadas e na obtenção de insights mais profundos dos dados no Power BI.

Exercício de aplicação:

- **Calcular o Total de Vendas com Desconto maior que 0**

```
Total Sales com Desconto = CALCULATE(
SUM(financials[Sales]), financials[Discounts] > 0)
```

- **Calcular a média de vendas por país, ignorando qualquer filtro aplicado à coluna "*Country*".**

```
Média vendas por país = CALCULATE(AVERAGE(financials[Sales]), ALLEXCEPT(Financials,
Financials[Country]))
```

- **Obter o valor máximo de vendas por *Segmento* e produto, removendo todos os outros filtros, exceto as colunas "*Segment*" e "*Product*".**

```
Max_Sales_by_Segment_Product = CALCULATE(MAX(financials[Sales]),
ALLEXCEPT(Financials, Financials[Segment], Financials[Product]))
```

- **Contar o número de todos os produtos distintos vendidos**

```
Distinct Product Count = CALCULATE(DISTINCTCOUNT(Financials[Product]),
ALL(Financials))
```

3.6.2 FILTER

A função **FILTER** no Power BI é utilizada para filtrar uma tabela ou uma expressão de tabela, devolvendo apenas as linhas que atendem a determinadas condições especificadas. O objetivo principal da função FILTER é restringir o conjunto de dados a ser utilizado num cálculo ou medida, com base em critérios específicos.

A sintaxe é a seguinte:

```
FILTER(<table>, <condition>)
```

Exemplo de aplicação:

```
Filtered Sales = CALCULATE(SUM(financials[Sales]),
                           FILTER(financials, financials[Country] = "Mexico"))
```

Esta fórmula calcula a soma da coluna "*Sales*" na tabela "*financials*", considerando apenas as linhas em que o país é "Mexico".

```
Filtered Rows = COUNTROWS(FILTER(financials, [Country] = "Mexico"))
```

Esta fórmula utiliza a função FILTER para aplicar um filtro na coluna "*Country*" e contar o número de linhas resultantes. Neste exemplo, estamos a contar o número de linhas em que o país é "Mexico".

A tabela é a tabela da qual deseja filtrar os dados, e a condição é a expressão lógica que define os critérios de filtragem.

O objetivo da função FILTER pode ser resumido da seguinte forma:

- **Filtrar dados com base em condições** - A função FILTER permite selecionar apenas as linhas que consideram uma ou mais condições específicas. Por exemplo, pode filtrar apenas as vendas de um determinado produto ou apenas as vendas realizadas num determinado país.

- **Restringir o âmbito de cálculos e medidas** - Ao aplicar a função FILTER num cálculo ou medida, pode restringir o conjunto de dados sobre o qual o cálculo é realizado. Isto é útil quando precisa de fazer cálculos específicos com base num subconjunto de dados.
- **Criar cálculos condicionais** - A função FILTER pode ser combinada com outras funções, como SUM, AVERAGE, MAX, MIN, entre outras, para criar cálculos condicionais. Por exemplo, pode calcular a soma das vendas apenas para os produtos com desconto.
- **Manipular e transformar dados** - Ao filtrar uma tabela usando a função FILTER, pode criar uma tabela ou expressão de tabela contendo apenas as linhas desejadas. Isto permite a criação de visualizações personalizadas ou a realização de análises específicas num subconjunto de dados.

A função FILTER é uma ferramenta poderosa para manipulação e filtragem de dados no Power BI. Ela ajuda a controlar o âmbito dos cálculos e medidas, permitindo uma análise mais precisa e granular dos dados disponíveis.

Exercício de aplicação:

- Calcula a soma da medida Total*Sales* na tabela "*financials*", considerando apenas as linhas em que o país é "France".
- Calcula a soma da medida Total*Sales* na tabela "*financials*", considerando apenas as linhas em que o país é "Canada".

3.6.3 ALL

A função **ALL** no Power BI é utilizada para remover ou **ignorar filtros aplicados a uma ou mais colunas numa expressão de tabela ou medida**. O objetivo principal da função ALL é alterar o contexto de avaliação dos dados, permitindo que os cálculos sejam realizados num contexto mais amplo.

A função ALL pode ser usada de diferentes maneiras e com diferentes parâmetros, conforme necessário.

A sintaxe é a seguinte:

```
ALL(<table>)
```

Exemplo de aplicação:

- **Remover todos os filtros de uma tabela**

```
TotalSales_All = CALCULATE(SUM(financials[Sales]), ALL(Financials))
```

A função ALL é usada com a tabela inteira "*Financials*". Isto remove todos os filtros aplicados à tabela, permitindo que o cálculo seja realizado considerando todas as linhas da tabela, independentemente dos filtros aplicados em outras partes do relatório.

- **Remover filtros da coluna *Country***

```
Total Sales (All Countries) = CALCULATE(SUM(financials[Sales]), ALL(financials[Country]))
```

Esta fórmula calcula a soma da coluna "*Sales*" na tabela "*financials*" removendo todos os filtros da coluna "*Country*".

- **Remover filtros de várias colunas**

```
Total Sales 2 colunas = CALCULATE(SUM(financials[Sales]),
                        ALL(Financials[Country], Financials[Segment]))
```

Augusto Martins

Neste caso, a função ALL é aplicada a várias colunas, "*Country*" e "*Segment*". Isto remove os filtros aplicados a ambas as colunas, permitindo que o cálculo seja feito considerando todas as combinações possíveis de país e *Segmento*.

Em resumo, o objetivo da função ALL é alterar o contexto de avaliação dos dados, removendo os filtros aplicados a uma ou mais colunas. Isto permite que os cálculos sejam realizados num contexto mais amplo, levando em consideração todos os valores disponíveis numa tabela ou coluna específica. A função ALL é útil quando precisa de realizar cálculos com base em todos os dados, independentemente dos filtros aplicados, fornecendo uma análise mais abrangente e precisa.

Exercício de aplicação:

- Calcula a soma da medida "Total*Sales*" na tabela "*financials*" removendo todos os filtros da coluna "*Product*".

3.6.4 ALLSELECTED

A função **ALLSELECTED** remove todos os filtros de uma tabela, **exceto aqueles selecionados pelo utilizador**. A sintaxe é a seguinte:

```
ALLSELECTED(<table>)
```

Exemplo de aplicação:

```
Total Sales (All Selected Countries) = CALCULATE(SUM(financials[Sales]),
                                        ALLSELECTED(financials[Country]))
```

Esta fórmula calcula a soma da coluna "*Sales*" na tabela "*financials*", removendo todos os filtros, exceto os selecionados pelo utilizador na coluna "*Country*".

Country	Total Sales	TotalSales_All
Canada	75.209.092	331.734.048,74
France	52.817.778	331.734.048,74
Germany	77.085.191	331.734.048,74
Mexico	60.443.242	331.734.048,74
United States of America	66.178.745	331.734.048,74
Total	**331.734.049**	**331.734.048,74**

Product	Total Sales	Total Sales (All Selected Countries)
Amarilla	56.090.749	56.090.748,89
Carretera	33.622.757	33.622.757,18
Montana	57.552.113	57.552.113,25
Paseo	92.664.608	92.664.608,21
Velo	48.980.692	48.980.692,28
VTT	42.823.129	42.823.128,93
Total	**331.734.049**	**331.734.048,74**

Year	Total Sales	Total Sales (All Selected Countries)
2020	78.154.102	78.154.102,38
2021	96.859.940	96.859.939,88
2022	95.761.037	95.761.037,15
2023	60.958.969	60.958.969,33
Total	**331.734.049**	**331.734.048,74**

Exercício de aplicação:

- Calcula a soma da medida "Total*Sales*" na tabela "*financials*" removendo apenas os filtros do contexto da coluna "*Product*".

3.6.5 ALLEXCEPT

A função **ALLEXCEPT** remove todos os filtros de uma tabela, exceto aqueles especificados nas colunas fornecidas. A sintaxe é a seguinte:

```
ALLEXCEPT(<table>, <column1>, <column2>, ...)
```

Exemplo de aplicação:

```
TotalSales(AllExceptCountry) = CALCULATE(SUM(financials[Sales]),
                                         ALLEXCEPT(financials, financials[Country]))
```

Esta fórmula calcula a soma da coluna "*Sales*" na tabela "*financials*", removendo todos os filtros, exceto o filtro aplicado na coluna "*Country*".

Country	Total Sales	TotalSales_All	Total Sales (All Except Country)
Canada	75.209.092	331.734.048,74	75.209.091,56
France	52.817.778	331.734.048,74	52.817.778,31
Germany	77.085.191	331.734.048,74	77.085.191,23
Mexico	60.443.242	331.734.048,74	60.443.242,39
United States of America	66.178.745	331.734.048,74	66.178.745,25
Total	**331.734.049**	**331.734.048,74**	**331.734.048,74**

Product	Total Sales	Total Sales (All Selected Countries)	Total Sales (All Except Country)
Amarilla	56.090.749	56.090.748,89	331.734.048,74
Carretera	33.622.757	33.622.757,18	331.734.048,74
Montana	57.552.113	57.552.113,25	331.734.048,74
Paseo	92.664.608	92.664.608,21	331.734.048,74
Velo	48.980.692	48.980.692,28	331.734.048,74
VTT	42.823.129	42.823.128,93	331.734.048,74
Total	**331.734.049**	**331.734.048,74**	**331.734.048,74**

Year	Total Sales	Total Sales (All Selected Countries)	Total Sales (All Except Country)
2020	78.154.102	78.154.102,38	331.734.048,74
2021	96.859.940	96.859.939,88	331.734.048,74
2022	95.761.037	95.761.037,15	331.734.048,74
2023	60.958.969	60.958.969,33	331.734.048,74
Total	**331.734.049**	**331.734.048,74**	**331.734.048,74**

Exercício de aplicação:

- Calcula o "Total*Sales*", removendo todos os filtros, exceto na tabela *financials*, o filtro aplicado na coluna "*Product*".

3.6.6 REMOVEFILTERS

A função **REMOVEFILTERS** remove todos os filtros de todas as tabelas na consulta atual. A sintaxe é a seguinte:

```
REMOVEFILTERS()
```

Exemplo de aplicação:

```
TotalUnits (No Filters) = CALCULATE([Total Units], REMOVEFILTERS())
```

```
TotalUnits (No Filters country) = CALCULATE([Total Units], REMOVEFILTERS(financials[Country]))
```

Augusta Martins

Esta fórmula calcula a soma da medida "Total Units" na tabela "*financials*, removendo todos os filtros.

Country	TotalSales_All	Total Units (No Filters)
Canada	331.734.048,74	3.364.443,00
France	331.734.048,74	3.364.443,00
Germany	331.734.048,74	3.364.443,00
Mexico	331.734.048,74	3.364.443,00
United States of America	331.734.048,74	3.364.443,00
Total	**331.734.048,74**	**3.364.443,00**

Product	Total Sales	Total Units (No Filters)
Amarilla	56.090.749	3.364.443,00
Carretera	33.622.757	3.364.443,00
Montana	57.552.113	3.364.443,00
Paseo	92.664.608	3.364.443,00
Velo	48.980.692	3.364.443,00
VTT	42.823.129	3.364.443,00
Total	**331.734.049**	**3.364.443,00**

Year	Total Sales	Total Units (No Filters)
2020	78.154.102	3.364.443,00
2021	96.859.940	3.364.443,00
2022	95.761.037	3.364.443,00
2023	60.958.969	3.364.443,00
Total	**331.734.049**	**3.364.443,00**

Como se pode ver no exemplo, nos diferentes contextos, REMOVEFILTERS, ignora todos os filtros, enquanto que `REMOVEFILTERS(financials[Country])`, apenas remove os filtros do país, não afeta os outros contextos.

Exercício de aplicação:

- Calcula o "Total*Sales*", removendo todos os filtros da tabela *financials*, na coluna "*Product*".

3.6.7 KEEPFILTERS

A função KEEPFILTERS é uma função de manipulação de contexto de filtro que permite preservar os filtros existentes numa expressão DAX enquanto realiza operações de cálculo. Ajuda a garantir que os filtros que já foram aplicados às colunas de uma tabela sejam mantidos durante o cálculo, evitando que sejam ignorados ou substituídos por novos filtros.

A sintaxe básica da função KEEPFILTERS é a seguinte:

```
KEEPFILTERS(<expression>)
```

Onde <expression> é a expressão DAX na qual deseja manter os filtros existentes.

Exemplo de aplicação: Somar as vendas apenas para um país específico mantendo os filtros de segmento:

```
SalesGemany Keepfiltes = CALCULATE( SUM(Financials[Sales]),
KEEPFILTERS(financials[País] = "Germany") )
```

Neste exemplo, a função KEEPFILTERS preserva quaisquer filtros existentes na coluna "Segment" enquanto somamos as vendas para o país " Germany ".

A função KEEPFILTERS é especialmente útil quando deseja realizar cálculos contextuais em DAX sem afetar os filtros que já foram aplicados em outras colunas ou tabelas. Ajuda a evitar resultados inesperados ou imprecisos que podem ocorrer quando os filtros são ignorados durante os cálculos.

Exercício de aplicação:

- Calcula o "*Total Sales*", mantendo todos os filtros da tabela financials, na coluna Product

3.7 Funções Iterantes

As funções iterantes no DAX, como SUMX, AVERAGEX, MAXX e MINX, são usadas para iterar sobre uma tabela ou expressão de tabela e realizar cálculos linha a linha.

3.7.1 SUMX

A função **SUMX** é usada para iterar sobre uma tabela e devolver a soma de uma expressão para cada linha. A sintaxe é a seguinte:

```
SUMX(table, expression)
```

Exemplo de aplicação:

```
TotalSales = SUMX(financials, financials[Units Sold]*financials[Sale Price]-financials[Discounts])
```

Esta fórmula itera sobre a tabela "*financials*" e devolve o produto da soma da coluna "*Units Sold*" pelo produto da soma da coluna "*Sale Price*" para cada linha, subtraídas do valor da coluna "*discounts*", resultando na soma total das unidades vendidas. Como se pode analisar nos três diferentes contextos, do exemplo seguinte:

Country	TotalSales
Canada	270.706.573,56
France	103.194.800,30
Germany	243.945.597,23
Mexico	236.329.922,39
United States of America	175.303.787,26
Total	**1.029.480.680,74**

Product	TotalSales
Amarilla	183.306.628,69
Carretera	70.623.717,18
Montana	264.795.509,25
Paseo	308.643.264,20
Velo	126.991.758,29
VTT	75.119.802,93
Total	**1.029.480.680,74**

Year	TotalSales
2020	514.788.388,38
2021	96.859.939,88
2022	100.132.291,15
2023	317.700.061,34
Total	**1.029.480.680,74**

Exercício de aplicação:

- Calcula o "*Sales*" por linha.

3.7.2 AVERAGEX

A função **AVERAGEX** é usada para iterar sobre uma tabela e devolver a média de uma expressão para cada linha. A sintaxe é a seguinte:

```
AVERAGEX(table, expression)
```

Exemplo de aplicação:

```
AveragexSales = AVERAGEX(financials, financials[Units Sold]*financials[Sale Price]-financials[Discounts])
```

```
AveragexTotalSales = AVERAGEX(financials,[TotalSales])
```

Esta fórmula itera sobre a tabela "*financials*" e devolve a média da coluna "*Units Sold*" pelo produto da soma da coluna "*Sale Price*" para cada linha, subtraídas do valor da coluna "*discounts*", resultando no preço médio de venda.

Como pode ver no exemplo também pode utilizar a medida explícita.

Country	TotalSales	AverageSales
Canada	270.706.573,56	553.592,17
France	103.194.800,30	263.252,04
Germany	243.945.597,23	480.207,87
Mexico	236.329.922,39	542.041,11
United States of America	175.303.787,26	417.389,97
Total	**1.029.480.680,74**	**458.566,00**

Exercício de aplicação:

- Calcula a média do "*Sales*" por linha.

3.7.3 MAXX

A função **MAXX** é usada para iterar sobre uma tabela e devolver o valor máximo de uma expressão para cada linha. A sintaxe é a seguinte:

```
MAXX(table, expression)
```

Exemplo de aplicação:

```
Maxx COGS = MAXX(financials, financials[COGS])
```

Esta fórmula itera sobre a tabela "*financials*" e devolve o valor máximo da coluna "*COGS*" para cada linha, resultando no maior custo dos produtos vendidos.

Product	TotalSales	Maxx COGS
Amarilla	183.306.628,89	771.160
Carretera	70.623.717,18	741.520
Montana	264.795.509,25	950.625
Paseo	308.643.264,20	897.000
Velo	126.991.758,29	948.375
VTT	75.119.802,93	748.020
Total	**1.029.480.680,74**	**950.625**

Exercício de aplicação:

- Calcula o Máximo de "*Sales*" por linha.

3.7.4 MINX

A função **MINX** é usada para iterar sobre uma tabela e devolver o valor mínimo de uma expressão para cada linha. A sintaxe é a seguinte:

```
MINX(table, expression)
```

Exemplo de aplicação:

```
MinSalePrice = MINX(financials, financials[Sale Price])
```

Esta fórmula itera sobre a tabela "*financials*" e devolve o valor mínimo da coluna "*Sale Price*" para cada linha, resultando no menor preço de venda.

Year	TotalSales	Minx SalePrice
2020	514.788.388,38	7
2021	96.859.939,88	7
2022	100.132.291,15	7
2023	317.700.061,34	7
Total	**1.029.480.680,74**	**7**

Estas funções iterativas são úteis quando precisa de calcular valores agregados com base em cada linha da tabela. Elas permitem realizar cálculos personalizados usando expressões condicionais ou lógica mais complexa em cada iteração.

Exercício de aplicação:

- Calcula o Mínimo de "*Sales*" por linha.

3.7.5 RANKX

A função **RANKX** no Power BI é utilizada para atribuir um ranking a uma coluna específica numa tabela, com base num critério de classificação. O objetivo principal da função RANKX é fornecer uma classificação numérica para os valores numa coluna, permitindo a análise e comparação de dados com base na sua posição no ranking.

A sintaxe básica da função RANKX é a seguinte:

```
RANKX(<tabela>, <expressão>, [valor], [ordem], [empate])
```

- A tabela é a tabela na qual deseja realizar a classificação,
- a expressão é a coluna que será usada para determinar a classificação,
- o valor é o valor específico que deseja classificar,
- a ordem (opcional) especifica se o ranking é ascendente ou descendente,
- e o empate (opcional) especifica como tratar empates nos valores classificados.

Exemplo de aplicação da função RANKX:

Suponha que queira atribuir um ranking às unidades vendidas por país na tabela "*financials*". Pode usar a função RANKX da seguinte forma:

```
RankxTotalSales = RANKX(all(financials[Country]),[TotalSales],,DESC)
```

Neste exemplo, a função RANKX calcula o ranking do país, com base na medida "*TotalSales*" da tabela "*financials*". O parâmetro "DESC" especifica que o ranking deve ser atribuído em ordem decrescente.

Observe na seguinte imagem que o Ranking é apenas calculado no exemplo que tem o contexto país.

Country	TotalSales	RankxTotalSales
Canada	270.706.573,56	1
France	103.194.800,30	5
Germany	243.945.597,23	2
Mexico	236.329.922,39	3
United States of America	175.303.787,26	4
Total	**1.029.480.680,74**	**1**

Product	TotalSales	RankxTotalSales
Amarilla	183.306.628,89	1
Carretera	70.623.717,18	1
Montana	264.795.509,25	1
Paseo	308.643.264,20	1
Velo	126.991.758,29	1
VTT	75.119.802,93	1
Total	**1.029.480.680,74**	**1**

Isto resulta numa coluna calculada que mostrará o ranking de cada país com base nas unidades vendidas. Por exemplo, Canada tem o maior número valor de vendas, ele será atribuído o ranking 1, a Germany é atribuído o segundo maior valor de vendas, ele será atribuído o ranking 2, e assim por diante.

A função RANKX é útil para análises comparativas, identificação de melhores e piores desempenhos e classificação de valores com base numa métrica específica. Ela permite visualizar a posição relativa de cada valor em relação aos demais, fornecendo insights sobre o desempenho e a classificação dos dados.

Exercício de aplicação:

- Criar um ranking para o "*TotalSales*" dos produtos, ordenando DESC.

4. Inteligência de Tempo e Medidas em DAX

As funções de inteligência de tempo no DAX são úteis para realizar cálculos e análises com base em informações temporais, como datas, anos, trimestres, meses, etc. Elas permitem que faça comparações temporais, cálculos acumulados, análises por períodos e muito mais.

Algumas das funções mais importantes:

4.1 SAMEPERIODLASTYEAR

A função **SAMEPERIODLASTYEAR** devolve um conjunto de datas correspondentes no mesmo período do ano anterior. É útil para fazer comparações de dados entre o ano atual e o ano anterior.

A sintaxe da função SAMEPERIODLASTYEAR é a seguinte:

```
SAMEPERIODLASTYEAR(<date>, [<dates column>])
```

- <date>: É a data, para a qual se deseja obter o período correspondente no ano anterior.
- <dates column> (opcional): É a coluna de datas da tabela na qual se deseja encontrar o período correspondente. Se não for especificada, a função usará a coluna de datas padrão da tabela atual.

Exemplo de aplicação da função:

```
TotalSalesLastYear = CALCULATE([TotalSales], SAMEPERIODLASTYEAR(dcalendario[Date]))
```

Year	TotalSales	TotalSalesLastYear
2020	514.788.388,38	
2021	96.859.939,88	514.788.388,38
2022	100.132.291,15	96.859.939,88
2023	317.700.061,34	100.132.291,15
Total	**1.029.480.680,74**	**711.780.619,41**

Exercício de aplicação:

- Crie uma medida com o nome ano-1 que calcule total units do ano-1

4.2 DATEADD

A função **DATEADD** permite adicionar ou subtrair uma quantidade específica de tempo a uma data. Ela permite que adicione ou subtraia um número específico de unidades de tempo (como dias, meses, anos, horas, minutos, etc.) a uma data.

A sintaxe da função DATEADD é a seguinte:

DATEADD(<tabela>[<coluna de data>], <número de unidades>, <unidade de tempo>).

- <tabela>: A tabela que contém a coluna de data.
- <coluna de data>: A coluna de data à qual o intervalo será adicionado ou subtraído.
- <número de unidades>: O número de unidades de tempo que será adicionado ou subtraído. No exemplo, é -1 para voltar um mês.
- <unidade de tempo>: A unidade de tempo que será adicionada ou subtraída. No exemplo, é MONTH.

É útil para realizar cálculos com datas deslocadas, como calcular as vendas por exemplo do último Mês.

Exemplo:

```
SalesLast Month = CALCULATE([Total Sales], DATEADD(dcalendario[Date], -1, MONTH))
```

Year	TotalSales	TotalSalesLastYear	SalesLast Month
2020	514.788.388,38		71.682.154,88
jan	7.526.138,20		
fev	24.810.463,58		3.127.398,20
mar	20.228.459,24		6.068.881,58
abr	34.566.723,07		5.411.433,24
mai	9.888.912,04		7.712.417,07
jun	6.210.013,65		3.652.110,04
jul	24.383.879,45		2.444.185,65
ago	19.926.713,95		4.205.327,45
set	103.697.711,30		4.716.779,95
out	105.781.503,00		9.194.651,30
nov	110.455.781,40		9.747.623,00
dez	47.312.009,50		15.401.347,40
2021	96.859.939,88	514.788.388,38	90.369.567,66
jan	1.967.579,60	7.526.138,20	6.471.947,50
fev	2.970.544,90	24.810.463,58	1.967.579,60

Exercício de aplicação:

- Crie uma medida com o nome mês-2 que calcule total units do mês-2

4.3 PREVIOUSYEAR

A função **PREVIOUSYEAR** devolve um conjunto de datas correspondentes ao ano anterior ao período atual. É útil para fazer comparações de dados entre o ano atual e o ano anterior.

A função PREVIOUSYEAR tem a seguinte sintaxe:

```
PREVIOUSYEAR(<date_column>)
```

No exemplo seguinte, a medida Total*Sales*PreviousYear calcula o total de vendas do ano anterior.

```
TotalSalesPreviousYear = CALCULATE([TotalSales], PREVIOUSYEAR(dcalendario[Date]))
```

Esta medida pode ser utilizada num visual ou relatório para mostrar o total de vendas do ano anterior, permitindo comparar o desempenho de vendas entre diferentes anos.

Year	TotalSales	TotalSalesPreviousYear
2020	514.788.388,38	
2021	96.859.939,88	514.788.388
2022	100.132.291,15	96.859.940
2023	317.700.061,34	100.132.291
Total	**1.029.480.680,74**	

A medida Total*Sales*PreviousYear usa a função CALCULATE para calcular o total de vendas ([*TotalSales*]) considerando a filtragem do ano anterior. A função PREVIOUSYEAR é aplicada à coluna de data "dcalendario[Date]", devolvendo um conjunto de dados correspondente ao ano anterior. Em seguida, a função CALCULATE avalia a medida [Total*Sales*] apenas para esse conjunto de dados.

Exercício de aplicação:

- Crie uma medida com o nome LY que calcule total units do ano -1

4.4 PREVIOUSMONTH

A função **PREVIOUSMONTH** é usada para obter um período anterior ao mês atual em uma coluna de data. É uma função de inteligência de tempo que permite realizar cálculos comparativos entre períodos, como comparar vendas deste mês com as vendas do mês anterior.

A função PREVIOUSMONTH tem a seguinte sintaxe:

```
PREVIOUSMONTH(<date_column>)
```

No exemplo seguinte, a medida Total*Sales*PreviousMonth calcula o total de vendas do mês anterior.

```
TotalSalesPreviousMonth = CALCULATE([TotalSales], PREVIOUSMONTH(dcalendario[Date]))
```

Esta medida pode ser utilizada num visual ou relatório para mostrar o total de vendas do mês anterior, permitindo comparar o desempenho de vendas entre diferentes meses.

Year	TotalSales	TotalSalesPreviousYear	TotalSalesPreviousMonth
2020	**514.788.388,38**		
jan	7.526.138,20		
fev	24.810.463,58		7.526.138,20
mar	20.228.459,24		24.810.463,58
abr	34.566.723,07		20.228.459,24
mai	9.888.912,04		34.566.723,07
jun	6.210.013,65		9.888.912,04
jul	24.383.879,45		6.210.013,65
ago	19.926.713,95		24.383.879,45
set	103.697.711,30		19.926.713,95
out	105.781.583,00		103.697.711,30
nov	110.455.781,40		105.781.583,00
dez	47.312.009,50		110.455.781,40
2021	**96.859.939,88**	**514.788.388,38**	**47.312.009,50**
jan	1.967.579,60	514.788.388,38	47.312.009,50
fev	2.970.544,90	514.788.388,38	1.967.579,60
mar	7.832.900,43	514.788.388,38	2.970.544,90
abr	9.546.142,38	514.788.388,38	7.832.900,43
mai	7.425.168,85	514.788.388,38	9.546.142,38

Exercício de aplicação:

- Crie uma medida com o nome LM que calcule *total units* do mês -1

4.5 DATESYTD

A função **DATESYTD** devolve um conjunto de datas que representa o período do início do ano até a data especificada. É útil para calcular totais acumulados de vendas no ano atual.

A função DATESYTD tem a seguinte sintaxe:

```
DATESYTD(<tabela>[<coluna de data>])
```

Exemplo: Deseja calcular as vendas acumuladas do ano atual até a data atual. A fórmula será a seguinte:

```
SalesYTD = CALCULATE([TOTALSales], DATESYTD(dcalendario[Date]))
```

Neste exemplo, a medida "*Sales*YTD" está a usar função CALCULATE, que permite aplicar um filtro ou modificar o contexto de cálculo. O argumento da função DATESYTD é usado como filtro na função CALCULATE para devolver as vendas totais acumuladas do ano atual até a data atual em relação à coluna de data da tabela "dcalendario".

Year	TotalSales	SalesYTD
2020	**514.788.388,38**	**514.788.388,38**
jan	7.526.138,20	7.526.138,20
fev	24.810.463,58	32.336.601,78
mar	20.228.459,24	52.565.061,02
abr	34.566.723,07	87.131.784,09
mai	9.888.912,04	97.020.696,13
jun	6.210.013,65	103.230.709,78
jul	24.383.879,45	127.614.589,23
ago	19.926.713,95	147.541.303,18
set	103.697.711,30	251.239.014,48
out	105.781.583,00	357.020.597,48
nov	110.455.781,40	467.476.378,88
dez	47.312.009,50	514.788.388,38
2021	**96.859.939,88**	**96.859.939,88**
jan	1.967.579,60	1.967.579,60
fev	2.970.544,90	4.938.124,50

Esta medida pode ser usada em visualizações e relatórios para acompanhar o desempenho das vendas ao longo do ano e comparar as vendas acumuladas com as metas estabelecidas. Ela fornece insights sobre o desempenho geral das vendas em relação ao objetivo anual.

Exercício de aplicação:

- Crie uma medida com o nome acumulado que calcule total units.

4.6 TOTALYTD

A função **TOTALYTD** devolve um valor acumulado para uma medida desde o início do ano até a data especificada. É útil para calcular totais acumulados de vendas no ano atual. Analisar o desempenho de uma medida em relação ao tempo, fornecendo uma visão acumulada das métricas durante todo o ano.

```
TOTALYTD(<expressão>, <tabela>[<coluna de data>])
```

Exemplo: Deseja calcular o total acumulado de vendas até a data atual

```
TotalSalesYTD = TOTALYTD( [TotalSales], dcalendario[Date])
```

Neste exemplo, a medida "Total*Sales*YTD" é calculada usando a função TOTALYTD. Ela recebe a medida "Total*Sales*" como a expressão a ser acumulada e usa a coluna de data "Date" da tabela "dcalendario" como referência para o período de acumulação.

Esta medida pode ser usada em visualizações e relatórios para mostrar o total acumulado de vendas até a data atual, permitindo uma análise do desempenho das vendas ao longo do ano.

Year	TotalSales	SalesYTD	TotalSalesYTD
2020	**514.788.388,38**	**514.788.388,38**	**514.788.388,38**
jan	7.526.138,20	7.526.138,20	7.526.138,20
fev	24.810.463,58	32.336.601,78	32.336.601,78
mar	20.228.459,24	52.565.061,02	52.565.061,02
abr	34.566.723,07	87.131.784,09	87.131.784,09
mai	9.888.912,04	97.020.696,13	97.020.696,13
jun	6.210.013,65	103.230.709,78	103.230.709,78
jul	24.383.879,45	127.614.589,23	127.614.589,23
ago	19.926.713,95	147.541.303,18	147.541.303,18
set	103.697.711,30	251.239.014,48	251.239.014,48
out	105.781.583,00	357.020.597,48	357.020.597,48
nov	110.455.781,40	467.476.378,88	467.476.378,88
dez	47.312.009,50	514.788.388,38	514.788.388,38
2021	**96.859.939,88**	**96.859.939,88**	**96.859.939,88**
jan	1.967.579,60	1.967.579,60	1.967.579,60
fev	2.970.544,90	4.938.124,50	4.938.124,50
mar	7.832.900,43	12.771.024,93	12.771.024,93
abr	9.546.142,38	22.317.167,31	22.317.167,31
mai	7.425.168,85	29.742.336,16	29.742.336,16
jun	12.571.599,75	42.313.935,91	42.313.935,91
Total	**1.029.480.680,74**	**317.700.061,34**	**317.700.061,34**

Exercício de aplicação:

- Crie uma medida com o nome TotalAcumulado que calcule total units.

É importante ressaltar que, embora o DAX seja uma linguagem poderosa, ele também requer um bom entendimento do modelo de dados e dos conceitos de análise. É necessário compreender as relações entre as tabelas, as colunas-chave e as medidas que deseja calcular para obter resultados precisos e relevantes.

Em resumo, o DAX é uma linguagem essencial para realizar cálculos personalizados e criar medidas avançadas no Power BI Desktop. Com o seu conhecimento e aplicação adequada, pode explorar e visualizar os seus dados de forma significativa, obtendo insights valiosos para a tomada de decisões informadas.

5. Introdução à otimização de desempenho

A otimização de desempenho no Power BI refere-se à aplicação de técnicas e práticas para melhorar a velocidade e eficiência das consultas, cálculos e visualizações num relatório ou Dashboard. O objetivo é garantir que as análises sejam executadas de forma rápida e responsiva, mesmo ao lidar com grandes volumes de dados.

Existem três modos de conjuntos de dados:

- Importação,
- DirectQuery e
- Composto.

A otimização de desempenho no Power BI é importante para oferecer aos utilizadores uma experiência fluida e interativa, garantindo que eles possam explorar e obter insights dos dados de maneira eficiente. Algumas das técnicas comuns de otimização de desempenho incluem:

- **Modelagem eficiente** - Isto envolve a criação de um modelo de dados bem projetado, com relacionamentos apropriados entre as tabelas, o uso adequado de tipos de dados e formatação correta. Uma modelagem adequada contribui para consultas mais rápidas e eficientes.
- **Filtragem seletiva** - Ao aplicar filtros aos dados, é importante usar filtros seletivos para reduzir o volume de dados processados. Isto pode incluir a utilização de filtros diretos (aplicados diretamente na fonte de dados) ou a criação de medidas calculadas que realizam pré-filtragem nos dados antes da análise.
- **Uso de agregações** - Para consultas que envolvem cálculos agregados, é benéfico usar agregações pré-calculadas ou resumos para evitar cálculos repetitivos e melhorar o desempenho das consultas.
- **Otimização de fórmulas DAX** - Ao escrever fórmulas DAX, é importante considerar a eficiência das expressões, evitando cálculos desnecessários e

usando funções otimizadas, como as funções de agregação SUMX, AVERAGEX, MAXX e MINX.

- **Indexação de fontes de dados** - Se a fonte de dados subjacente suportar a criação de índices, é recomendável criar índices relevantes para melhorar o desempenho das consultas.

Estas são apenas algumas das estratégias e técnicas comuns utilizadas na otimização de desempenho no Power BI. O objetivo final é garantir que os relatórios e painéis sejam rápidos, responsivos e eficientes, proporcionando aos utilizadores uma experiência de análise de dados eficaz.

5.1 Desempenho de medidas, relacionamentos e visuais

O desempenho de medidas, relacionamentos e visuais no Power BI refere-se à eficiência e velocidade com que as medidas são calculadas, os relacionamentos são processados e os visuais são renderizados num relatório ou painel.

Desempenho de medidas - As medidas são fórmulas DAX que realizam cálculos e agregações nos dados. O desempenho das medidas é crucial para garantir que os resultados sejam obtidos rapidamente ao realizar análises e consultas. O objetivo é escrever medidas eficientes que sejam calculadas de forma rápida e não sobrecarreguem o modelo de dados.

Desempenho de relacionamentos - Os relacionamentos no Power BI são usados para ligar tabelas de dados e permitir a combinação e análise conjunta desses dados. O desempenho dos relacionamentos é importante para garantir que as consultas e cálculos que envolvem várias tabelas sejam executados de forma eficiente. Relacionamentos bem projetados e otimizados podem melhorar significativamente o desempenho das consultas e reduzir o tempo de resposta.

Desempenho de visuais - Os visuais no Power BI são as representações gráficas dos dados, como gráficos, tabelas e mapas. O desempenho dos visuais está relacionado à

velocidade com que os dados são processados e mostrados nos visuais. Isto inclui a renderização rápida dos visuais, a capacidade de lidar com grandes volumes de dados e a interatividade responsiva para filtragem e navegação.

Para melhorar o desempenho de medidas, relacionamentos e visuais no Power BI, é recomendável seguir boas práticas, como:

- Utilizar medidas otimizadas, evitando cálculos complexos e excessivos.
- Projetar relacionamentos adequados, escolhendo a cardinalidade correta e evitando relacionamentos desnecessários.
- Utilizar técnicas de otimização, como agregações, para melhorar o desempenho de medidas agregadas.
- Limitar a quantidade de dados mostrados em visuais, usando filtros e *Segment*ação de dados adequados.

O objetivo final é garantir que as medidas sejam calculadas rapidamente, os relacionamentos sejam eficientes e os visuais sejam responsivos, proporcionando aos utilizadores uma experiência de análise de dados ágil e eficaz. Isto permite explorar os dados de forma interativa e obter insights de maneira eficiente.

5.1.1 Técnicas de redução de dados para modelos de importação

Os modelos de importação são carregados com dados que são comprimidos, otimizados e depois armazenados para um disco pelo motor de armazenamento VertiPaq. Quando os dados de origem são carregados para a memória, é possível observar uma compressão de 10x e é razoável esperar que 10 GB de dados de origem possam ser comprimidos para cerca de 1 GB. Além disso, quando os dados persistem num disco, é possível alcançar uma redução adicional de 20%.

Técnicas de redução de dados diferentes. Estas técnicas incluem:

- Remover colunas desnecessárias
- Remover linhas desnecessárias

- Agrupar por e resumir
- Otimizar os tipos de dados de colunas
- Preferir as colunas personalizadas
- Desativar o carregamento de consultas do Power Query
- Desativar data/hora automáticas
- Mudar para o Modo misto

5.2 Variáveis para melhorar o desempenho e a resolução de problemas

As variáveis no Power BI são usadas para armazenar valores temporários ou intermédios durante a execução de fórmulas DAX. Elas desempenham um papel importante na melhoria do desempenho e na resolução de problemas ao permitir a reutilização de cálculos e a simplificação de fórmulas complexas.

O objetivo principal das variáveis é melhorar a legibilidade, a manutenção e o desempenho das fórmulas DAX. Aqui estão algumas das principais utilidades das variáveis no contexto da tabela "*financials*":

Armazenar valores repetidos ou calculados: Em vez de repetir um cálculo complexo várias vezes numa fórmula, pode atribuir o resultado a uma variável e usá-la posteriormente. Isto evita repetições desnecessárias e melhora a clareza e a eficiência da fórmula.

Exemplo:

```
TotalRevenue =
    VAR SalePrice = SUM(financials[Sale Price])
    VAR UnitsSold = SUM(financials[Units Sold])
    RETURN SalePrice * UnitsSold
```

Simplificar fórmulas complexas - Quando uma fórmula DAX se torna complexa e difícil de entender, o uso de variáveis pode simplificar o código, dividindo-o em partes menores e mais legíveis. Isto facilita a resolução de problemas e a depuração de erros.

Exemplo:

```
GrossProfit =
        VAR TotalCOGS = SUM(financials[COGS])
        VAR TotalRevenue = SUM(financials[Units Sold] * financials[Sale Price])
        RETURN
        TotalRevenue - TotalCOGS
```

Otimizar o desempenho - Ao armazenar valores intermédios em variáveis, pode reduzir a quantidade de cálculos repetitivos e melhorar o desempenho geral das fórmulas. As variáveis permitem que realize cálculos complexos uma vez e use o resultado armazenado em várias partes da fórmula.

Exemplo:

```
ProfitMargin=
        VAR GrossProfit = SUM(financials[Units Sold] * financials[Sale Price]) -
        SUM(financials[COGS])
        VAR TotalRevenue = SUM(financials[Units Sold] * financials[Sale Price])
        RETURN
        DIVIDE(GrossProfit, TotalRevenue)
```

Em resumo, o uso de variáveis no Power BI permite simplificar fórmulas complexas, melhorar a legibilidade, reutilizar cálculos e otimizar o desempenho. Isto facilita a resolução de problemas e contribui para um desenvolvimento mais eficiente de modelos e relatórios.

5.3 Redução de cardinalidade

A redução de cardinalidade refere-se a um processo de transformação de dados que visa diminuir o número de valores distintos numa coluna ou dimensão. O objetivo principal é reduzir o volume de dados e melhorar o desempenho das consultas e das operações de análise.

No contexto da tabela "*financials*", aqui estão alguns exemplos de como a redução de cardinalidade pode ser aplicada:

- **Agregação de dados** - Uma abordagem comum para reduzir a cardinalidade é realizar agregações nos dados. Por exemplo, em vez de manter as unidades vendidas em nível individual, podemos agregá-las por mês ou trimestre. Isto reduz o número de linhas na tabela e simplifica as consultas, especialmente quando são necessários cálculos ou análises em nível mais alto.
- **Agrupamento de categorias** - Outra técnica é agrupar categorias ou valores em grupos mais amplos. Por exemplo, em vez de ter países individuais, podemos agrupá-los por regiões geográficas. Isto reduz o número de valores únicos na coluna "*Country*" e simplifica as análises que envolvem agrupamentos maiores.
- **Criação de hierarquias** - A criação de hierarquias é outra forma de reduzir a cardinalidade e tornar a análise mais intuitiva. Por exemplo, podemos criar uma hierarquia de tempo com níveis como ano, trimestre, mês e dia. Isto permite que os utilizadores naveguem pelos diferentes níveis de detalhe sem a necessidade de incluir todas as datas individuais nos cálculos.

A redução de cardinalidade é benéfica porque ajuda a melhorar o desempenho do modelo de dados e das consultas, reduz a complexidade das análises e facilita a visualização e a compreensão dos dados. No entanto, é importante encontrar um equilíbrio entre a redução de cardinalidade e a necessidade de detalhe e granularidade nas análises, garantindo que as informações essenciais não sejam perdidas durante o processo de redução.

5.4 Otimização dos modelos DirectQuery com armazenamento ao nível da tabela

A otimização dos modelos DirectQuery com armazenamento ao nível da tabela refere-se a uma técnica de otimização utilizada no Power BI quando se trabalha com ligações DirectQuery.

Quando se utiliza o DirectQuery, os dados são consultados em tempo real diretamente da fonte de dados subjacente, em vez de serem importados e armazenados no Power BI. Isto é útil quando se lida com grandes volumes de dados ou quando a fonte de dados está em constante atualização.

A otimização dos modelos DirectQuery com armazenamento ao nível da tabela envolve o uso de tabelas auxiliares armazenadas localmente no Power BI para melhorar o desempenho das consultas. Estas tabelas são criadas a partir de consultas otimizadas que recuperam apenas os dados necessários para as análises, filtrando e resumindo-os conforme necessário. Estas tabelas auxiliares são então usadas como fonte para os visuais e análises, reduzindo a carga de trabalho na base de dados de origem e melhorando a velocidade de resposta das consultas.

Com base na tabela "*financials*", um exemplo de otimização dos modelos DirectQuery com armazenamento ao nível da tabela seria criar uma tabela auxiliar que contenha apenas as colunas necessárias para uma determinada análise ou visualização, filtrando os dados relevantes com base nas necessidades do utilizador. Isto pode ajudar a reduzir a quantidade de dados recuperados da fonte de dados subjacente, melhorando o desempenho das consultas.

Esta técnica é útil para melhorar a velocidade de resposta das consultas e otimizar o desempenho geral ao trabalhar com conexões DirectQuery no Power BI. No entanto, é importante considerar o tamanho das tabelas auxiliares e a frequência de atualização dos dados para garantir que as consultas continuem a fornecer resultados precisos e atualizados.

5.5 Criação de gestão de agregações

A criação de gestão de agregações refere-se a uma técnica usada no Power BI para otimizar o desempenho de consultas ao trabalhar com grandes volumes de dados. Esta técnica envolve a pré-calculação e o armazenamento de valores agregados em tabelas específicas, chamadas de tabelas de agregação.

O objetivo da criação de gestão de agregações é melhorar a velocidade de resposta das consultas, reduzindo a quantidade de cálculos e processamento necessários para obter resultados agregados. Em vez de calcular agregações em tempo real a partir de um grande conjunto de dados, as tabelas de agregação contêm valores pré-calculados que podem ser acedidos de forma mais rápida.

Com base na tabela "*financials*", um exemplo de criação de gestão de agregações seria criar uma tabela de agregação que armazena valores agregados, como a soma das unidades vendidas por país e produto. Esta tabela conteria os valores agregados pré-calculados para cada combinação de país e produto, eliminando a necessidade de realizar cálculos repetidos toda vez que uma consulta é executada.

Ao utilizar as tabelas de agregação, as consultas podem ser direcionadas para Estas tabelas em vez de consultar diretamente a tabela original. Isto resulta em consultas mais rápidas e eficientes, especialmente quando são necessários resultados agregados.

No exemplo da tabela "*financials*", alguns exemplos úteis de agregação podem ser:

- **Total de unidades vendidas por país** - pode criar uma agregação para calcular a soma das unidades vendidas agrupadas por país. Isso permitirá que obtenha rapidamente o total de unidades vendidas para cada país, facilitando a análise da distribuição geográfica das vendas.
- **Receita total por produto** - Uma agregação pode ser criada para calcular a receita total de vendas para cada produto. Isso permitirá que obtenha rapidamente a receita gerada por cada produto, facilitando a identificação dos produtos mais lucrativos.

- **Descontos totais por *Segmento*** - Uma agregação pode ser criada para calcular a soma dos descontos aplicados agrupados por *Segmento*. Isso permitirá que obtenha rapidamente o total de descontos concedidos para cada *Segmento*, facilitando a análise do impacto dos descontos nas vendas.
- **Custo total de mercadorias vendidas por mês** - Uma agregação pode ser criada para calcular o custo total de mercadorias vendidas agrupado por mês. Isso permitirá que obtenha rapidamente o custo das mercadorias vendidas em cada mês, facilitando a análise de tendências sazonais ou variações mensais nos custos.

No entanto, é importante ressaltar que a criação de gestão de agregações requer um planeamento cuidadoso. É necessário identificar as agregações relevantes com base nas análises e consultas mais comuns, determinar a granularidade adequada para as tabelas de agregação e atualizá-las regularmente para manter os dados atualizados.

Em resumo, a criação de gestão de agregações no Power BI visa melhorar o desempenho das consultas pré-calculando e armazenando valores agregados em tabelas específicas. Isto resulta em consultas mais rápidas e eficientes ao trabalhar com grandes volumes de dados.

Conclusão

Este manual explorou de forma abrangente e detalhada as diferentes etapas da transformação de dados, a utilização de medidas DAX e a otimização de desempenho no ambiente do Power BI Desktop. Desde a extração e transformação dos dados até a implementação de medidas complexas utilizando funções DAX, este material proporcionou uma visão abrangente das capacidades do software para análise e visualização de dados. A compreensão das relações eficazes, medidas semi-aditivas e funções iterantes, aliada à exploração de medidas temporais, demonstrou como o Power BI pode ser uma ferramenta poderosa para análises sofisticadas.

Além disso, a secção dedicada à otimização de desempenho ressaltou a importância de garantir que modelos e visuais permaneçam eficientes mesmo quando lidando com grandes volumes de dados. As técnicas apresentadas, como a redução de cardinalidade, o uso de variáveis e a criação de gestão de agregações, oferecem abordagens práticas para melhorar a velocidade e a responsividade das visualizações, promovendo uma experiência mais fluida para os usuários.

Em suma, a combinação de conhecimentos sobre transformação de dados, medidas DAX e otimização de desempenho fornece aos profissionais a capacidade de criar modelos robustos, análises detalhadas e painéis interativos que atendem às demandas complexas do mundo dos dados. O Power BI Desktop se revela não apenas como uma ferramenta poderosa, mas também como um recurso fundamental para impulsionar a tomada de decisões informadas e estratégicas em diversos contextos empresariais.

Bibliografia / Infografia

1. Documentação do Power BI - Power BI | Microsoft Learn
2. Referência do DAX (Data Analysis Expressions) - DAX | Microsoft Learn
3. Alberto Ferrari e Marco Russo "*Analyzing Data with Power BI and Power Pivot for Excel*"
4. Brett Powell, "*Mastering Microsoft Power BI*"
5. Mr. Grant Gamble , "*Power BI Step-by-Step*"