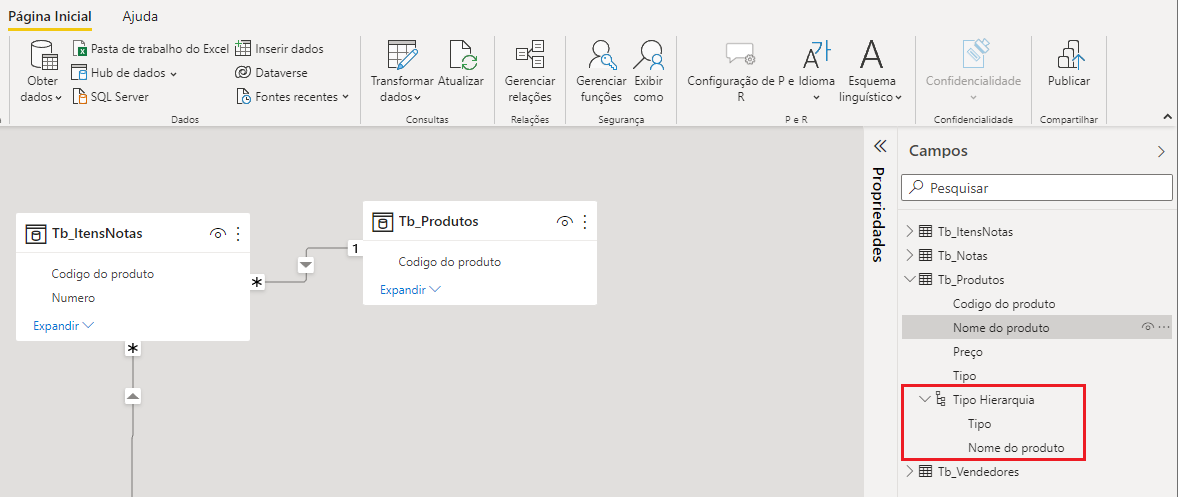
**Aula 01**

**Conectando e relacionando**

Podemos criar uma hierarquia nas colunas de uma tabela. No nosso exemplo temos a tabela “Produtos” e temos a coluna “Tipo” que tem dois valores, se é livro ou e-book, e temos outra coluna chamada “Nome do produto”. Vamos criar uma hierarquia que a coluna “Tipo” vai ser a coluna “Pai” da “Nome do produto”.

Para isso vamos clicar na coluna “Tipo” com botão direito e clicar em “Adicionar hierarquia”, será criado um novo espaço na tabela e podemos arrastar a coluna “Nome produto” para dentro dela, assim ficando mais fácil a visualização da hierarquia.



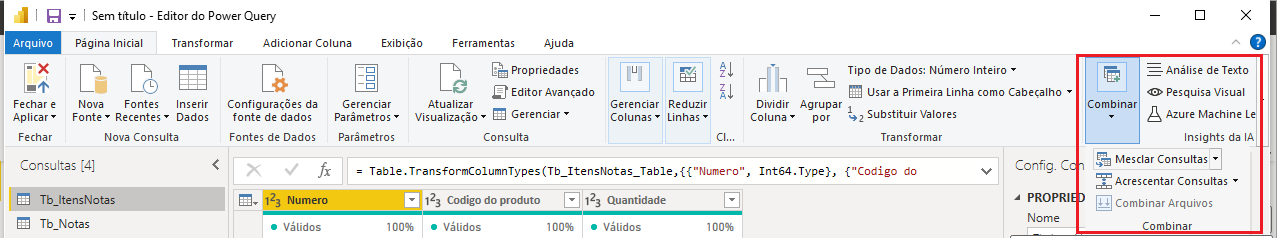
Isso deixa a nossa visualização mais enxuta e fácil de entender.

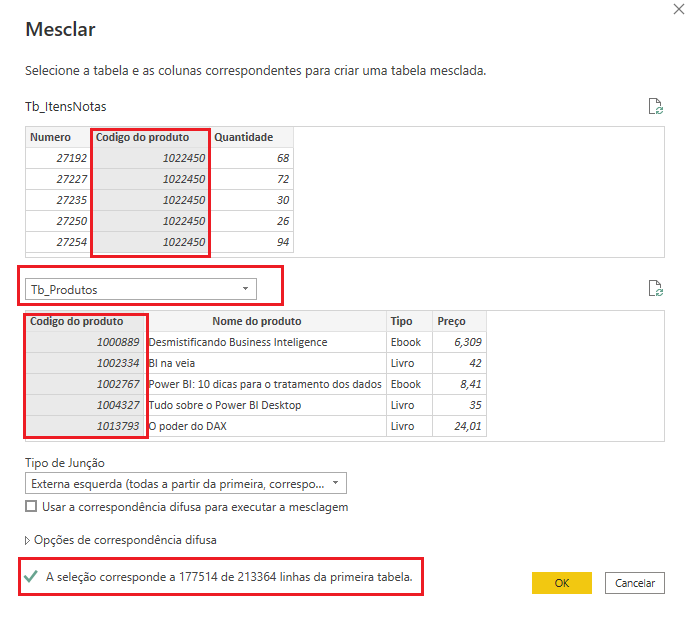
**Aula 02**

**Linguagem DAX**

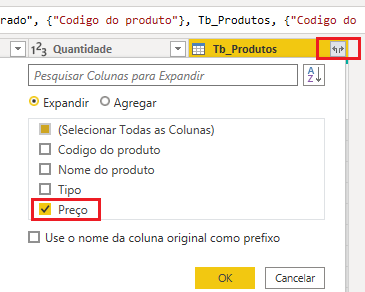
Na tabela de “Itens notas” temos duas colunas interessantes que é a de “Código do produto” e “quantidade”, mas não temos aqui o preço de cada produto e vamos precisar desse preço para poder fazer a conta de quantidade vezes o preço desse produto e depois realizar a soma para criar uma visualização.

Na tabela “Produtos” nós temos a coluna “Preço” de cada produto, o que temos que fazer é criar uma combinação dessas colunas, para poder ter essa coluna “Preço” na tabela de “Itens notas”, mas como vamos fazer isso?

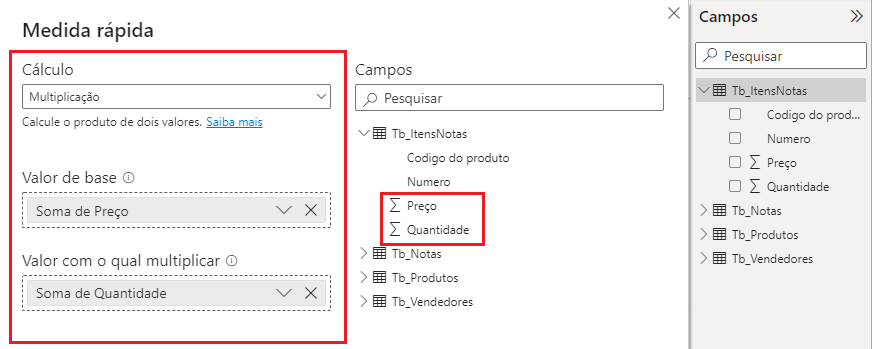
Vamos na pagina inicial e depois vamos clicar em “Transformar dados” assim abrindo o editor do Power Query, lá vamos deixar a tabela “Itens notas” selecionada e clicar no ícone de “Combinar” que fica no canto direito do cabeçalho, e depois em “Mesclar consultas”. 

Logo em seguida vamos selecionar a tabela “Produtos” e clicar nas colunas que se relacionam entre as duas tabelas, que no caso é a de “Código do produto”.

Clicando no OK vamos voltar para o Power Query e será adicionado uma nova coluna na tabela “Itens notas” e clicando no ícone ao lado do nome da coluna podemos escolher qual é a informação que será exibido da tabela de “Produtos” no nosso caso será a coluna “Preço”

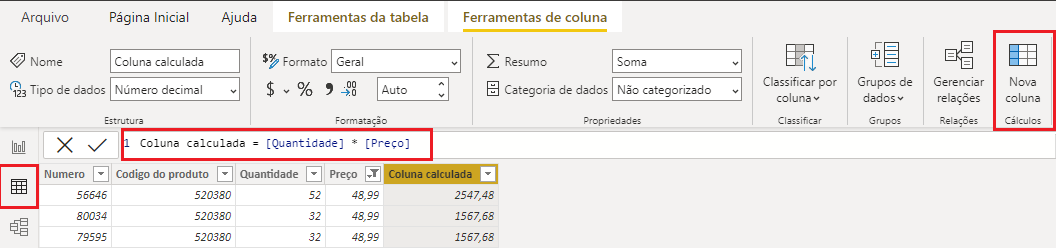


Agora vamos fazer um cálculo já que temos essa nova coluna. Para isso vamos na parte de criação de visualizações, clicar na tabela “Itens notas” com o botão direito e selecionar a opção “Novas medidas rápidas”, nessa opção vamos ter vários tipos de cálculos que podemos fazer, o que vamos escolher é a multiplicação que fica na parte de Operações matemáticas. Com isso vamos ter dois campos para adicionar as colunas que queremos, no nosso caso será a coluna “Preço” e no segundo campo a coluna “Quantidade”. Com isso será gerado uma medida que podemos colocar em uma visualização de cartão, porém o valor não parece muito certo, para isso vamos precisar tratar esse cálculo de outra forma.



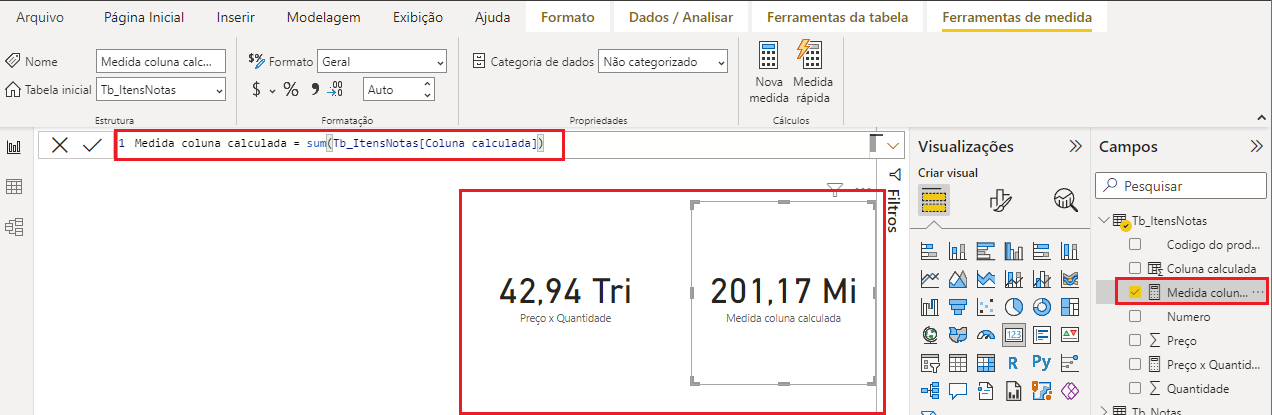
Depois de colocar essa medida em um cartão vimos que o valor está muito fora da realidade, vamos analisar a conta clicando na medida. Vamos ver que essa medida está pegando todos os valores da coluna “Preço” e somando e depois está multiplicando por todas as quantidades somadas, oque dá um valor extremamente alto.

Vamos criar uma nova coluna para poder fazer a conta linha a linha, para isso precisamos clicar no ícone de “Dados” que fica no canto esquerdo do programa, depois vamos clicar em “Nova coluna” e para essa nova coluna vamos nomear ela como “Coluna Calculada” e passar um conta que é a coluna [Quantidade] vezes (\*) a coluna [preço], apertando Enter os valores vão aparecer nas linhas.



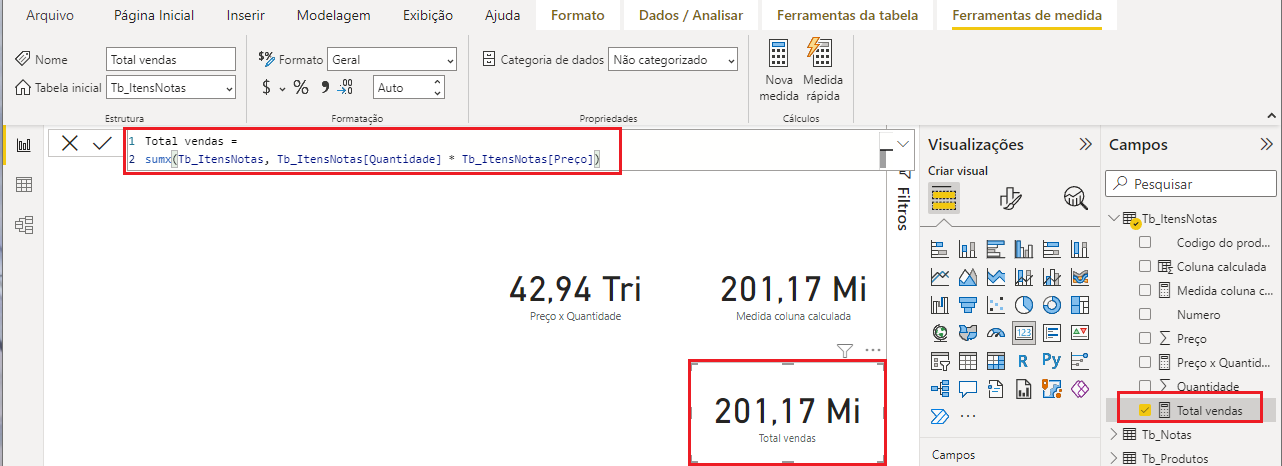
Agora vamos criar uma nova medida somando todos esses valores dessa nova coluna que criamos.

Vamos até a tabela de “Itens notas” clicamos com o botão direito e selecionamos a opção “Nova medida”. Agora vamos dar um nome para essa medida de “Soma coluna calculada” e usar a função de soma que é a SUM () e passar como parâmetro a coluna que criamos, depois basta passar essa medida para um cartão de visualização e vamos ver a diferença entre os números de antes e agora.



Existe outra forma que conseguimos executar uma ação linha a linha e depois executar uma soma sem precisar dessa nova coluna que criamos.

Vamos criar uma nova medida e dar o nome dela de “Total vendas” e essa medida vai ter a função chamada de “SUMX()”, vamos passar o primeiro parâmetro dela que é a tabela e o segundo é a expressão que queremos, no nosso caso é a coluna “Quantidade” vezes a coluna “Preço”. Com isso vamos ter uma nova medida e podemos colocar ela em um cartão e vemos que ela fica igual a anterior, sem precisar criar uma nova coluna para isso.



**Tabela Calendário**

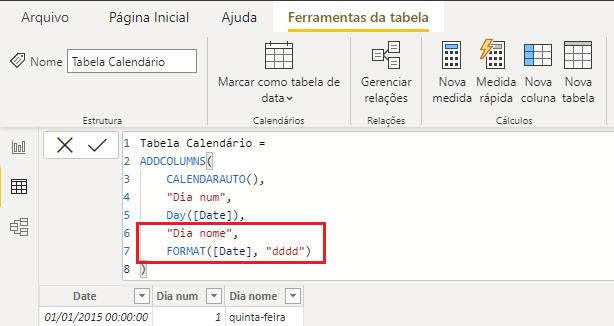
Para criar uma tabela calendário vamos até “Dados” e clicar na opção “Nova tabela”, com isso vai ser adicionado uma nova tabela com uma coluna calculada que precisa de algum cálculo, a função que vamos usar é uma que é capaz de ler todas as datas que tem dentro do nosso projeto e ela vai passar esse intervalo.

Então vamos colocar o nome dessa tabela de “Tabela Calendário” e dentro dela vamos usar uma função chamada “CALENDARAUTO()” e não vamos passar nenhum parâmetro para ela, com isso vamos ver o intervalo de tempo que temos no nosso projeto.

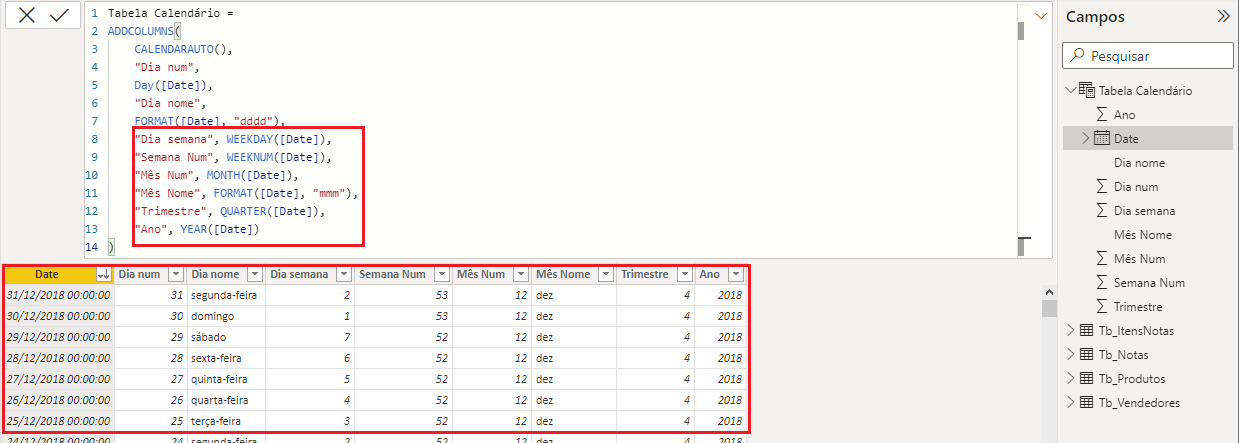
Porém queremos poder separar por dia entre outros períodos, para isso vamos usar a linguagem DAX. Lá no campo onde colocamos os comandos, vamos adicionar o comando “ADDCOLUMNS()”, como parâmetro vamos passar primeiro a tabela que vamos usar, porem como está sendo automático a adição das datas, vamos passar como tabela o comando “CALENDARAUTO()” e indentando esse comando e adicionar a virgula para poder ir para o próximo parâmetro. O próximo parâmetro é nome dessa nova coluna que vamos adicionar, vamos colocar o nome de “Dia num” com essas aspas mesmo. E agora como ultimo parâmetro vamos ter que passar a função que vai extrair o dia da tabela, essa função é a “DAY()” e como parâmetro vamos passar a “[DATA]”. Agora basta fechar o comando e ver como ficou.



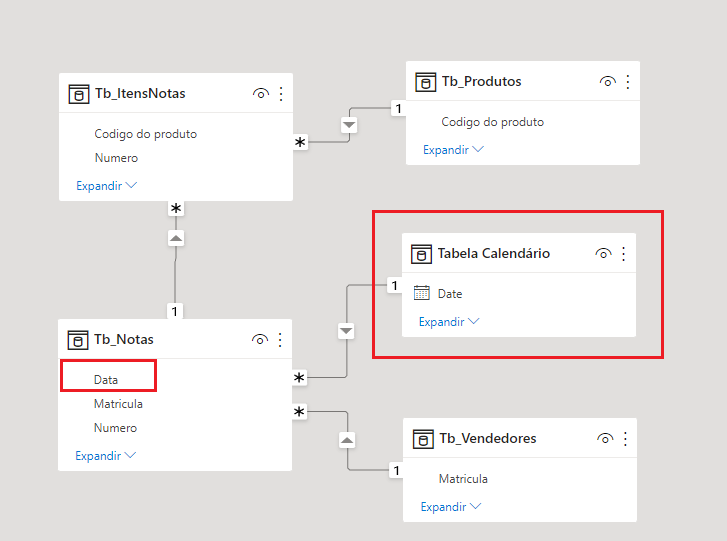
Agora digamos que eu queria adicionar uma coluna que tenha também o dia na semana, basta adicionarmos mais um nome de coluna que vamos chamar de “Dia nome”, e usar uma função chamada “FORMAT()”, essa função recebe dois parâmetros, o primeiro é a tabela que no nosso caso é a “[DATE]” e o segundo parâmetro é como o nome será exibido, no nosso caso vamos adicionar 4 D’s entre aspas para mostra o nome inteiro “dddd”, mas caso queira mostra só a abreviação dos nomes basta deixar apenas 3 D’s.



Vamos passar mais algumas colunas com o dia da semana, numeração da semana, o numero do mês, o nome do mês, o trimestre e o ano.



Agora basta a gente relacionar essa nova “tabela calendário” com a tabela “Notas” pela coluna “Date”.

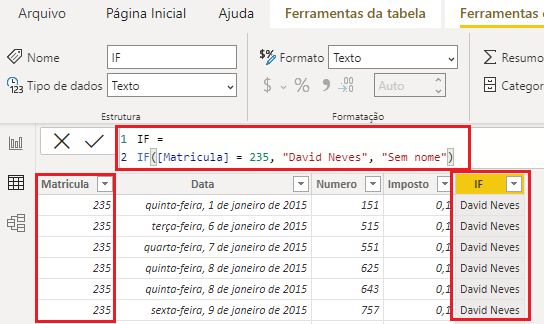


**Aula 03**

**Aprofundando na linguagem**

Na tabela de Notas temos uma coluna referente a matricula de cada vendedor, nesse momento vamos criar uma coluna nessa tabela para usar uma condição de verificação para mostrar o nome do funcionário referente aquela matricula. Para isso vamos usar a função IF (), nela vamos passar o nome da coluna, que no nosso caso é a “Matricula” depois o sinal de igual (=) e o numero de matricula que desejamos verificar, nesse caso vai ser a 235 referente ao funcionário “David Neves”, logo em seguida adicionamos a virgula e o nome que queremos que apareça caso a condição seja verdadeira, nessa caso vamos passar o nome do “David Neves”, adicionamos outra virgula e o valor que irá aparecer caso a condição seja falsa, nesse caso vamos colocar o valor “Sem nome” e fechamos a condição, o comando ficará assim:

Condição = IF ([Matricula] = 235, “David Neves”, “Sem nome”)



Caso queira testar uma condição ou outra podemos adicionar duas barras verticais e a outra condição:

Condição = IF([Matricula] = 235 || [Matricula] = 236, “David Neves”, “Sem nome”)

Caso seja uma condição e outra, vamos substituir as barras verticais por dois E’s comerciais:

Condição = IF([Matricula] = 235 && [Matricula] = 236, “David Neves”, “Sem nome”)

Mas assim é muito trabalhoso e vamos ter mais vendedores para aparecer o nome nessa lista, então vamos criar outra coluna e usar uma outra função que é a SWITCH (), essa função verifica todas as condições baseadas em uma única expressão.

Então vamos passar a função SWITCH e como expressão vamos usar a coluna “Matricula” e nessa coluna vamos verificar se tem o valor 235, se tiver vamos passar a virgula e depois o valor que é o nome do funcionário com aquela matricula, no nosso caso é o “David Neves”, na linha de baixo fazemos a mesma coisa com as outras matriculas.

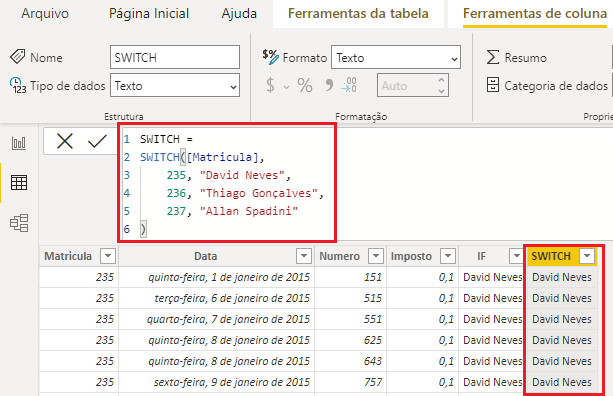
Condição = SWITCH ([Matricula]

235, “David Neves”,

236, “Thiago Gonçalves”,

237, “Allan Spadini”

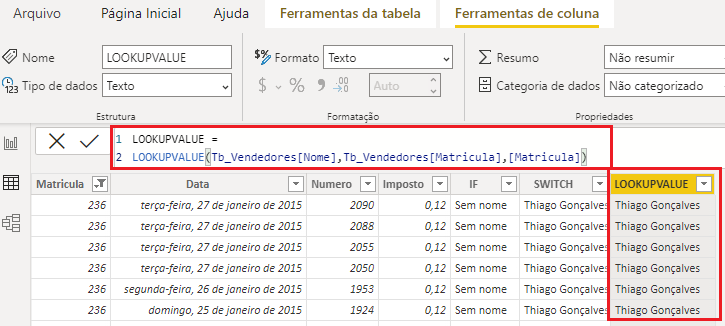
)



Temos mais uma maneira de fazer essa procurar de relações entre nome e matricula através de uma função chamada LOOKUPVALUE () que faz essa busca através das relações entre duas tabelas.

Vamos usar a função LOOKUPVALUE, e vamos passar como primeiro parâmetro a coluna que eu quero como resultado o nosso caso é a “Nome” da tabela de “Vendedores”, agora como segundo parâmetro vamos passar o relacionamento das tabelas, nesse caso vai a coluna “Matricula” da tabela de Vendedores, vamos separar pro virgula e agora a outra coluna que á coluna “Matricula” da tabela de “Notas”.

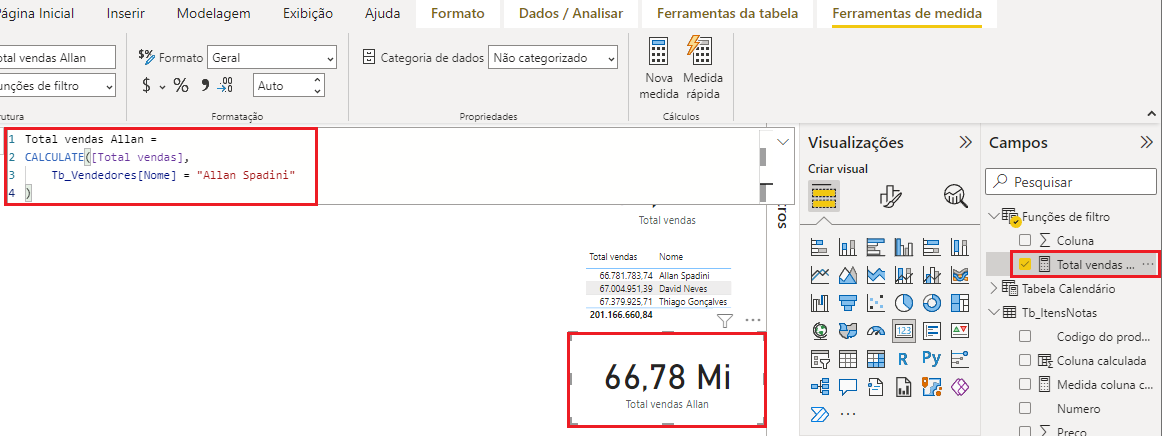
Condição = LOOKUPVALUE( Tb\_Vendedores[Nome], Tb\_Vendedores[Matricula], [Matricula])



Agora vamos criar uma medida onde será filtrado o total de vendas de um determinado vendedor, para isso vamos criar uma nova tabela chamada “Funções de filtro” e criara uma nova medida chamada “Total vendas Allan”, nessa medida vamos usar a função CALCULATE ().

Agora como primeiro parâmetro dessa função vamos assar uma medida que já temos que é a “Total Vendas” vamos adicionar uma virgula e vamos passar para a parte do filtro, lá vamos passar a coluna “Nome” da tabela de Vendedores, depois adicionamos o sinal de igual (=) e o nome do vendedor que queremos, no nosso caso é “Allan Spadini” e fechamos a função.

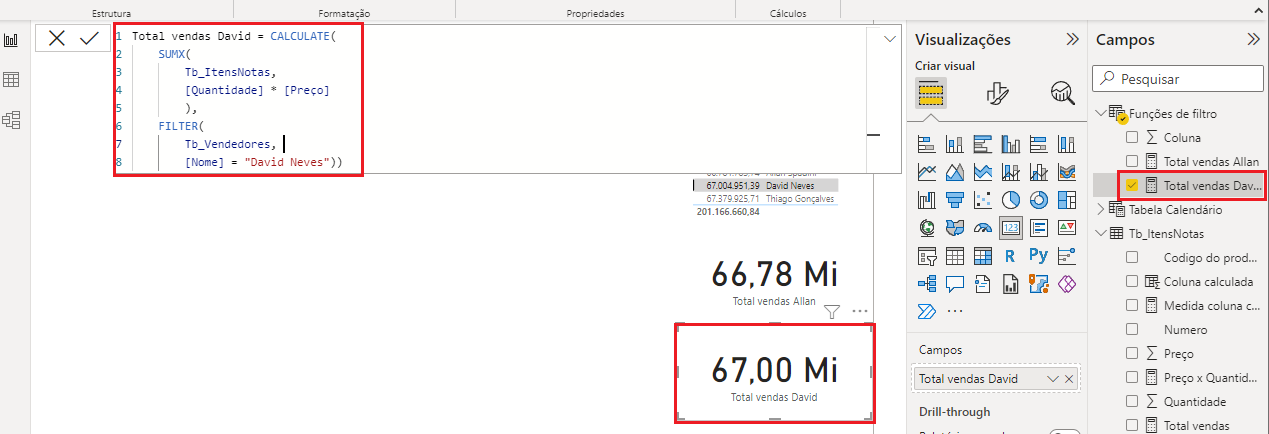
Total vendas Allan = CALCULATE([Total Vendas], Tb\_vendedores[nome] = “Allan Spanidi”)



Podemos também utilizar uma conta dentro do CALCULATE, antes usamos uma medida já pronta, agora vamos usar uma conta.

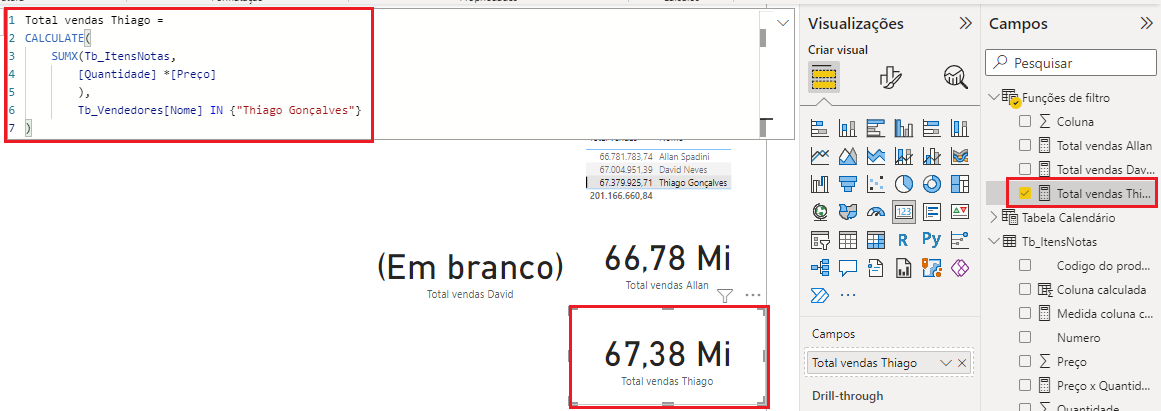
Então vamos na tabela de “Funções de filtro” e criar uma nova medida. Vamos chamar essa medida de “Total vendas David”, vamos usar a função CALCULATE e passar a conta pra ela como expressão que vai ser “SUMX(Tb\_itensNotas, [Quantidade] \* [Preço])” depois vamos adicionar uma virgula para começar a condição do filtro, e nesse caso vamos usar uma outra função chamada “FILTER” que ela retorna uma tabela usando determina critério, vamos passar para ela como primeiro parâmetro a tabela que queremos filtrar que no nosso vaso é a tabela de “Vendedores” e como segundo parâmetro a critério que é na coluna “Nome” depois o sinal de igual (=) tenha o valor “David Neves” e vamos fechar a função.

Total vendas David = CALCULATE( SUMX(Tb\_ItensNotas, [Quantidade] \* [Preço]), FILTER(Tb\_Vendedores, [Nome] = “David Neves”))



Agora vamos criar o cartão do Thiago com uma alteração, vamos criar uma nova medida igual a anterior porem no filtro, não vamos usar a função FILTER, vamos passar a tabela de Vendedores com a coluna “Nome” e usar o comando IN que requer um valor que esteja dentro dessa coluna, e entre chaves {} vamos passar o nome “Thiago Gonçalves” e pronto.

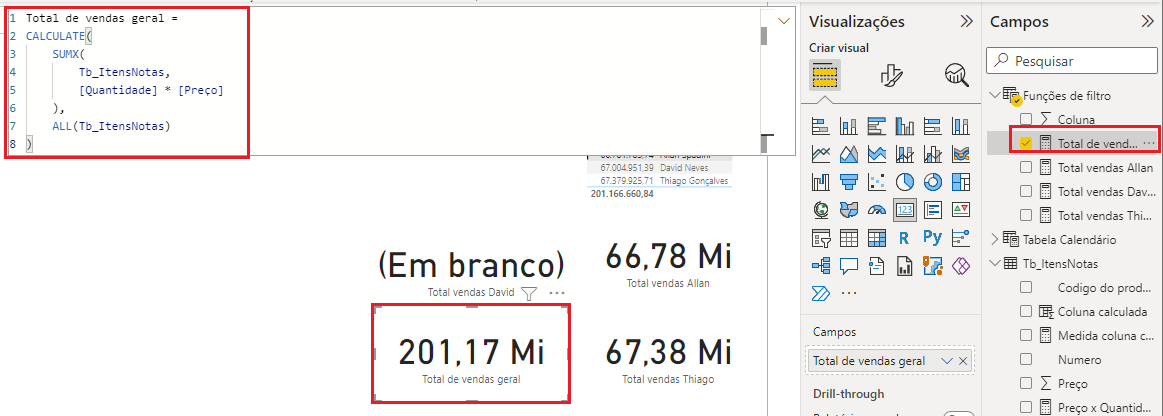
Total vendas David = CALCULATE( SUMX(Tb\_ItensNotas, [Quantidade] \* [Preço]), Tb\_Vendedores[Nome] IN {“Thiago Gonçalves”})



Agora vamos criar uma medida com o total de vendas geral, no qual não vai ser alterada quando clicarmos em qual filtro ou cartão no dashboard.

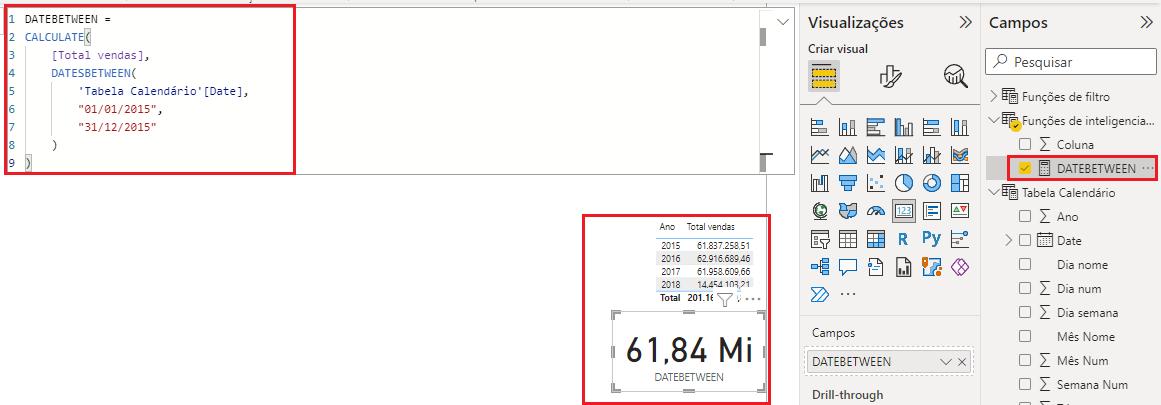
Para isso vamos criar uma nova medida, e vamos chamar ela de “Total de vendas geral” e vamos passar a função CALCULATE () com a função SUMX () anterior, porém na parte de filtro vamos usar uma função chamada de ALL (), essa função ela traz o resultado puro sem nenhum filtro, ela irá ignorar todo e qualquer filtro que seja utilizado no dashboard. Vamos passar para essa função a tabela “Itens Notas” e vamos fechar a função.

Total de vendas Geral = CALCULATE( SUMX(Tb\_ItensNotas, [Quantidade] \* [Preço]), ALL(Tb\_ItensNotas))



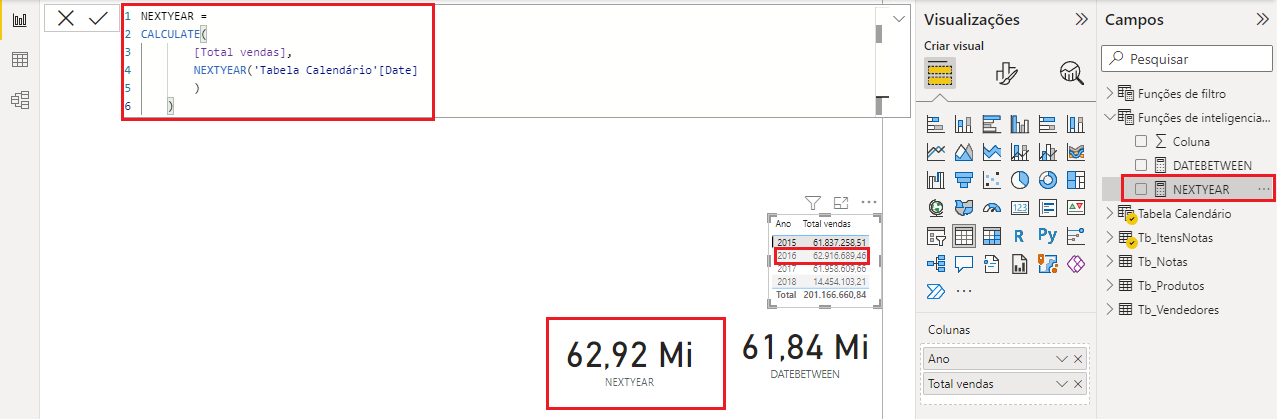
Agora vamos criar uma tabela de visualização e passar o “Ano” da tabela “Calendário” e a medida “Total vendas” da tabela de “Itens Notas”. Tendo isso pronto vamos criar uma tabela chamada “Funções de inteligência de dados” e criar uma medida que vai receber a função CALCULATE () e como parâmetro vai receber a medida “Total vendas”, como filtro vamos usar uma função chamada DATEBETWEEN () que retorna um total entre duas datas, como primeiro parâmetro vamos passar a coluna “DATE” da tabela “Tabela Calendário”, adicionamos a virgula e agora vamos passar a data inicias que vai ser “01/01/2015” adicionamos outra virgula e passamos a data final que vai ser “31/12/2015” depois de fechar a função vamos adicionar essa medida em uma cartão e verificar se está batendo com o valor da tabela.

Condição = CALCULATE([Total vendas], DATEBETWEEN(‘Tabela Calendário’[Date], “01/01/2015”, “31/12/2015”))

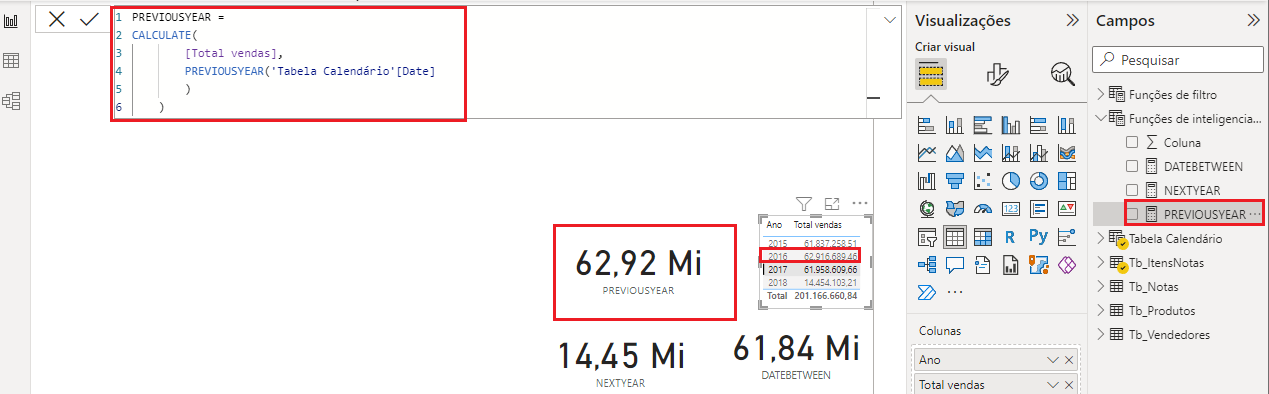


Podemos também criar uma medida que mostre o resultado do próximo ano, assim que selecionarmos um ano da tabela, por exemplo: se selecionarmos o ano de 2015 a medida trará o resultado de 2016 e assim por diante.

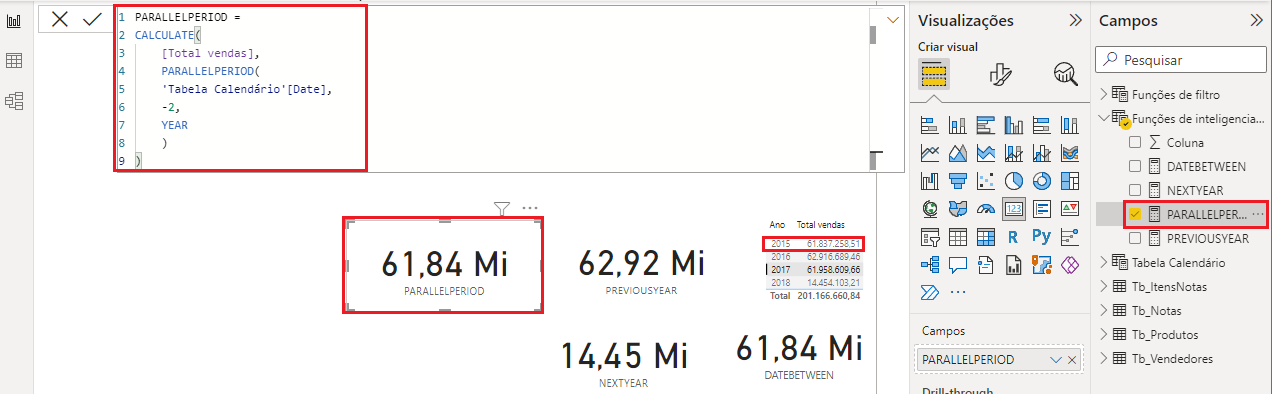
Para isso vamos criar uma nova medida com a função CALCULATE () e passando como primeiro parâmetro a medida “Total Vendas” e depois o parâmetro de filtro vai ser a função NEXTYEAR () e vamos passar para ela como parâmetro a coluna “DATE” da “Tabela Calendário”. Assim quando passarmos essa medida para um cartão e selecionarmos um ano, ela vai retornar o valor do ano seguinte.



Assim como olhamos para o futuro também podemos olhar para o passado usando a função PREVIOUSYEAR (). Vamos usar a mesma estrutura anterior porem vamos alterar a função de filtro.

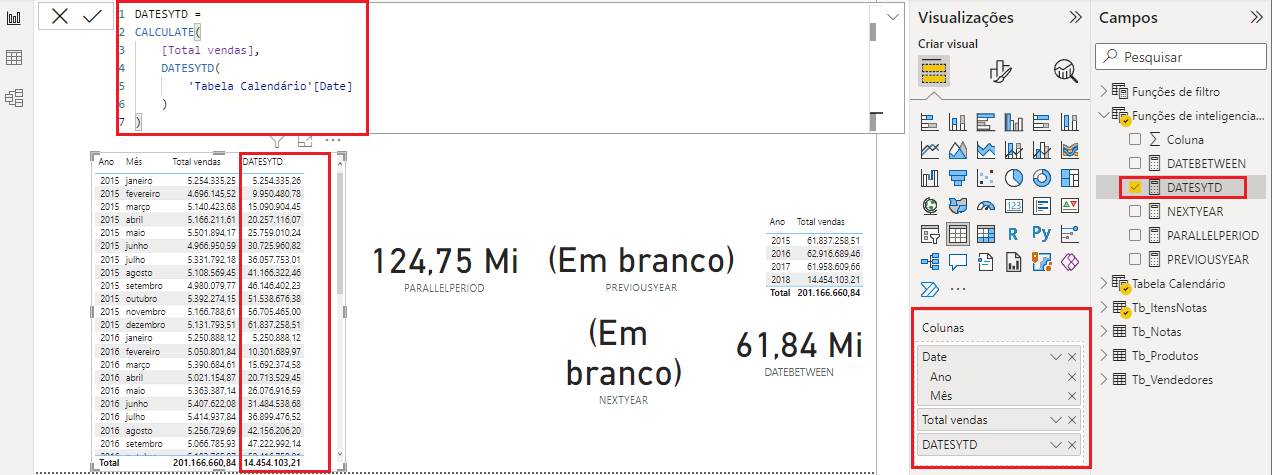


Podemos também ter uma medida que mostra anos a frente. Para isso vamos usar a função PARALLELPERIOD (), nessa função vamos passar a nossa “Tabela calendário” e depois vamos passar a quantidade de anos que queremos visualizar a frente. Caso queiramos ver os anos para trás basta adicionar o sinal de menos antes do número. Como terceiro parâmetro podemos escolher se vamos andar em meses (MONTH), trimestres (QUARTER) ou anos (YEAR).

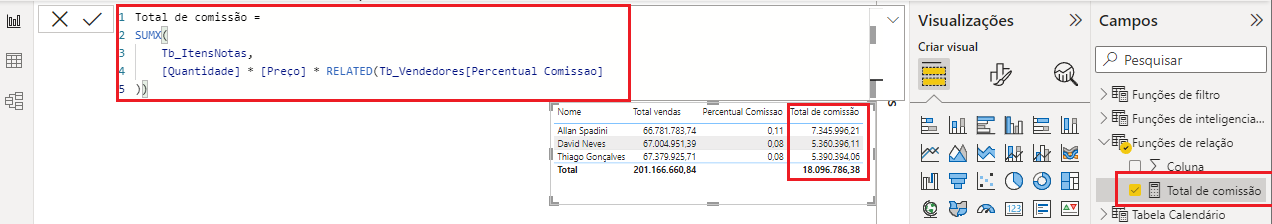


Vamos criar uma tabela de visualização com as colunas “Ano” e “Mês” da “Tabela calendário” e vamos adicionar a medida “Total vendas”. Agora vamos usar uma função para ver os valores de modo acumulativo de cada mês, com o valor de um mês acumulando com o do próximo e assim vai.

Para isso vamos usar a função DATASYTD () e passar como parâmetro a coluna “Date” da “Tabela calendário” com a mesma estrutura das medidas anteriores.



Agora vamos encontrar o total de comissão de cada vendedor. Para isso vamos primeiro criar uma nova tabela com o nome de “Funções de relação” e com ela criada vamos adicionar uma nova medida que vai se chama “Total de comissão”, nela vamos usar a função SUMX () para somar linha a linha, e vamos passar como parâmetro a tabela “Itens notas”, e vamos pegar a coluna “Quantidade” vezes (\*) a coluna “Preço” só que agora precisamos fazer mais uma multiplicação com uma coluna que está em outra tabela, e para isso vamos adicionar o sinal de multiplicação e usar a função RELATED () que traz uma coluna de outra tabela que tenha relação, e a coluna que vamos trazer é a “Percentual Comissão” da tabela de “Vendedores”

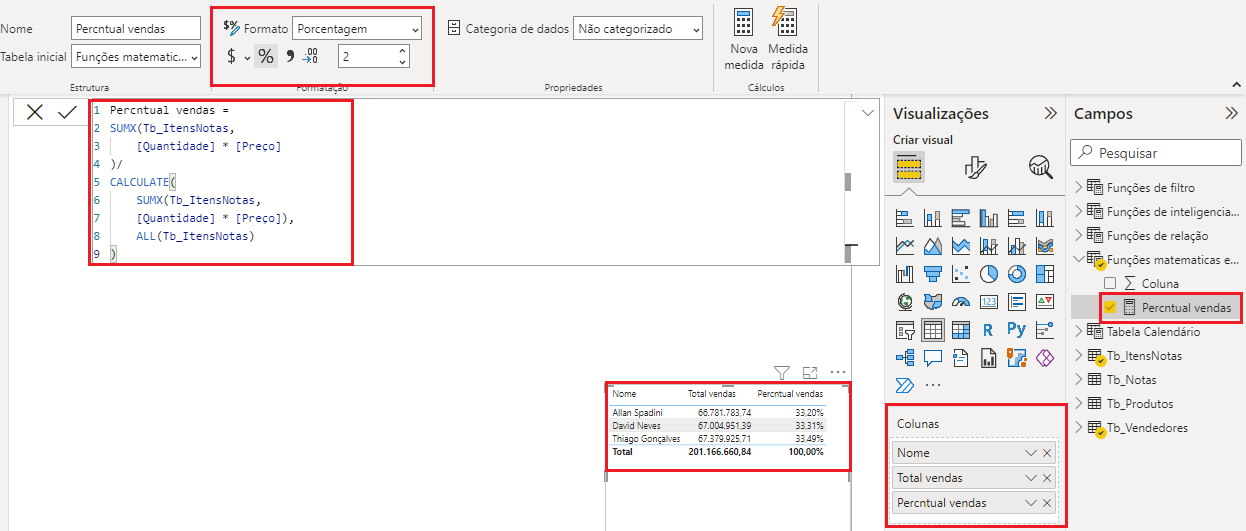


Agora digamos que temos uma segmentação de dados com o nome dos vendedores e um cartão que exibe o nome do vendedor selecionado, quando não estamos selecionando nenhum nome o cartão exibe o primeiro nome na lista, e não queremos isso. Vamos fazer com que o cartão só exiba o nome de algum vendedor quando ele for selecionado na segmentação e quando não tiver nenhum selecionado, o cartão irá exibir a mensagem “Todos os vendedores”.

Para isso vamos criar uma medida na tabela de funções e vamos usar a função IF () para verificar uma condição, dentro dela vamos passar como primeiro parâmetro uma outra função chamada HASONEFILTER () e passar para ela a coluna “Nome” da tabela de “Vendedores”, agora vamos usar a função MAX () para trazer somente o nome da tabela de vendedores, passando como parâmetro as mesmas informações da função anterior. Agora vamos passar uma String caso não tenha nenhum nome selecionado que vai ser “Todos os vendedores” e agora vamos fechar a função IF e passar essa nova medida para o cartão de exibição.

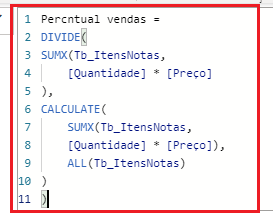
Agora vamos criar uma medida que calcule o percentual de vendas de cada vendedores em relação ao total de vendas. Para isso vamos criar uma nova tabela chamada “Funções matemáticas e estatísticas” nela vamos criar uma nova medida que vai ter o nome de “Percentual Vendas”, nela vamos fazer o calculo do total de vendas, e para isso vamos usar a função SUMX () passar como parâmetro a tabela “Itens notas” e fazer a multiplicação da coluna “Quantidade” e “Preço”, agora vamos fechar essa função e dividir o total dessa conta por outra função que é a CALCULATE () e como parâmetro vamos passar toda a função SUMX anterior e como filtro vamos usar a função ALL () passando como parâmetro a tabela “Itens notas” assim não vai ter nenhum filtro aplicado nela.

Vamos adicionar essa nova medida a tabela de exibição de exibição junto com o nome dos vendedores e o total de vendas. Lembrando que temos que formatar para percentagem essa medida.



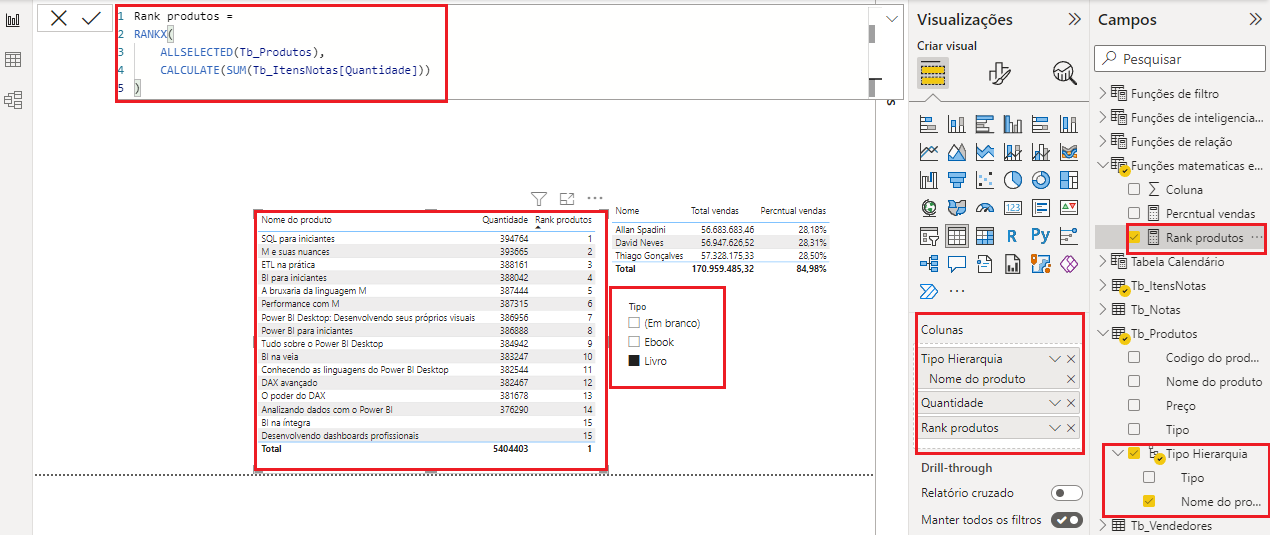
Se usamos uma segmentação de dados por produto o resultado dessa porcentagem será incorreto pois o total tem que ser baseado em cima desse filtro. Para isso vamos apenas fazer uma alteração na medida, ao invés de usar a função ALL () como filtro, vamos altera-la para ALLSELECTED ().

Podemos evitar outro erro como por exemplo o numero zero como denominador, fazendo uma outra alteração na nossa medida. Ao invés de usar o sinal de barra para dividir (/) vamos usar a função DIVIDE (), como primeiro parâmetro vamos passar o numerador que é toda a função SUMX e o segundo parâmetro o denominador que é toda a função CALCULATE, basta fechar a função e pronto, evitamos mais um erro.



Agora vamos criar uma tabela que vai ranquear os nossos produtos. Para isso vamos adicionar uma tabela de visualização e passar para ela a coluna “Nome” da tabela de “Produtos” e a coluna “Quantidade” da tabela de “Itens notas”. Agora vamos adicionar uma nova medida na tabela de “Funções matemáticas” que vai ser chama de “Rank produtos” que vai ter a função RANKX () e passar como primeiro parâmetro a função ALLSELECTED e dentro dela vamos passar a tabela de “Produtos” e o segundo parâmetro vai ser a função CALCULATED e dentro dela precisamos fazer a soma da quantidade desses produtos, vamos usar a função SUM e passar como parâmetro a coluna “Quantidade” da tabela “Itens notas”.

Agora quando clicamos na segmentação de Tipos de produto vamos ter o ranqueamento certinho.



Aula 04

**Ferramentas externas**

Vamos instalar o DAX Studio e quando abrirmos ele pode ver que ele já reconhece o projeto do Power BI que está em uso, caso seja o que queremos basta conectar e ele já trais as tabelas com os dados, mas podemos nos conectar ao DAX Studio de outra forma. No Power BI quando abrirmos o projeto vamos ter uma aba nova na parte de cima que é da “Ferramentas”, lá basta clicar no ícone do DAX Studio e ele já vai nos redirecionar para o programa já com a base dedados carregada.

Na área onde vamos adicionar os códigos para fazer as consultas, vamos sempre usar um comando chamado EVALUATE que quer dizer “avaliar”, a partir desse comando o DAX entende que vai ter que avaliar as condições do comando a seguir. Vamos usar como o exemplo o comando VALUES e entre parênteses vamos usar a tabela de Vendedores, da seguinte forma “EVALUATE VALUES(Tb\_Vendedores)”. Esse comando vai retornar os valores únicos de toda a tabela.

Agora se quisermos criar uma medida com um valor único, vamos ter que criar uma tabela com uma linha e uma coluna só,