

Aula 3 - Criando Relacionamentos

Docupedia Export

Author:Goncalves Donathan (SO/OPM-TS21-BR)
Date:25-Jul-2024 13:38

Table of Contents

1	3.1 - Introdução e conceitos básicos	4
1.1	Introdução aos Relacionamentos	4
1.1.1	Por que criar Relacionamentos	4
1.1.1.1	Criando Relacionamentos x Mesclando Colunas	5
1.2	Conceitos Básicos	5
1.2.1	Tabela Fato x Tabela Característica	5
1.2.1.1	Tabela Fato	5
1.2.1.2	Tabela Características	6
1.2.1.3	Identificando as Tabelas Fato e Tabelas Características	9
1.2.2	Chave Primária x Chave Estrangeira	12
1.2.2.1	Chave Primária	12
1.2.2.2	Chave Estrangeira	13
1.2.2.3	Identificando as Chaves Primárias e Chaves Estrangeiras	14
1.2.3	Cardinalidade dos Relacionamentos	17
1.2.3.1	Cardinalidade um para um (1:1)	17
1.2.3.2	Cardinalidade um para muitos (1:M)	19
1.2.3.3	Cardinalidade muitos para muitos (N:M)	19
2	3.2 - Criando e Gerenciando Relacionamentos	20
2.1	Criando Relacionamentos	31
2.2	Gerenciando e Editando Relacionamentos	50
3	3.3 - Relações Ativas e Inativas, Tabelas Fato e Direção do Filtro	59
3.1	Relações Ativas e Inativas	59
3.2	Trabalhando com várias Tabelas Fato	75
3.3	Direção do Filtro	80
3.3.1	Entendendo a Direção do Filtro	80
3.3.2	Filtro Bidirecional	89

3.3.2.1 Cuidados com o filtro bidirecional

96

4 3.4 - Ocultando Campos do Relatório e Boas Práticas

98

4.1 Aula 3 - Boas Práticas

107

4.1.1 Crie um modelo de dados organizado e normalizado

107

4.1.2 Organize Tabelas Características acima das Tabelas Fato

107

4.1.3 Evite usar filtro bidirecional a não ser que seja necessário

107

4.1.4 Oculte no relatório campos que não precisam ser usados

107

1 3.1 - Introdução e conceitos básicos

1.1 Introdução aos Relacionamentos

Nessas aulas veremos o que são Relacionamentos, alguns conceitos básicos, sua importância e a criar/gerenciar essa ferramenta que é o coração do Power BI. Nessa aula vamos trabalhar principalmente na guia Modelo.

O que são Relacionamentos

Os relacionamentos são a base para a criação de um modelo de dados eficiente e coerente. Eles permitem conectar tabelas de dados diferentes com base em colunas-chave correspondentes - colunas que possuem valores que relacionam entre si e são usadas para combinar os dados de diferentes tabelas de forma inteligente.

Ao criar relacionamentos entre tabelas, estabelecemos uma conexão lógica entre elas. Isso significa que poderemos criar consultas, visualizações e análises que envolvem várias tabelas, permitindo a exploração das informações de maneira integrada - muitas vezes temos dados em tabelas diferentes que se complementam, e essa é a forma de conseguir ligar essas informações.

Eles funcionam de maneira semelhante a relacionamentos em um banco de dados relacional, garantindo a consistência e integridade dos dados - evitando, assim, duplicações e reduzindo a redundância.

1.1.1 Por que criar Relacionamentos

A criação dos relacionamentos traz uma série de benefícios e possibilita uma análise de dados mais poderosa e abrangente. Aqui estão alguns motivos para tal:

- **Integração de dados:** ao criar relacionamentos entre tabelas, podemos unir informações de diferentes fontes de dados em único modelo coeso. Isso permite uma visão abrangente de todos os dados, independentemente de onde eles estejam armazenados;
- **Análises combinadas:** com relacionamentos adequados, podemos criar visualizações e análises que envolvam dados de várias tabelas; isso significa que é possível explorar conexões e padrões entre diferentes conjuntos de dados, obtendo *insights* mais profundos e significativos;
- **Atualizações automáticas:** quando configurados corretamente, o Power BI atualiza automaticamente as visualizações e análises sempre que os dados de origem são alterados. Isso garante que nossos relatórios estejam sempre atualizados e reflitam as informações mais recentes;
- **Evitar a duplicação de dados:** eles permitem a normalização dos dados, eliminando a necessidade de repetir as mesmas informações em várias tabelas. Isso reduz a redundância e melhora a eficiência do modelo de dados;
- **Filtros cruzados:** com os relacionamentos, é possível aplicar filtros em uma tabela e ver os efeitos nos dados relacionados em outras tabelas. Isso facilita a exploração e a compreensão do impacto de diferentes dimensões nos dados.

Em resumo, a conexão entre esses dados e essas tabelas é fundamental para explorar e analisar os dados de forma integrada. Isso permite criar relatórios robustos, atualizados e repletos de *insights* maravilhosos, fornecendo uma visão poderosa das informações.

1.1.1.1 Criando Relacionamentos x Mesclando Colunas

Reforçando: A grande diferença entre o uso das duas opções é que o Mesclar Colunas é mais usado quando queremos combinar os dados de tabelas pequenas, com poucas informações, onde não vale a pena deixar essas informações em tabelas separadas, e os Relacionamentos entram em ação quando estamos falando de tabelas com milhares de informações que estão relacionadas com características presentes em outras tabelas - não seria prático juntar as informações em uma única tabela pois a deixaríamos muito mais pesada e com informações repetitivas.

1.2 Conceitos Básicos

1.2.1 Tabela Fato x Tabela Característica

Existem dois tipos de tabelas do Power BI: **Tabelas Fato** e **Tabelas Características**.

1.2.1.1 Tabela Fato

Essas tabelas representam informações sobre fatos que aconteceram - venda, produção, transporte, devolução etc., onde cada linha apresenta as informações sobre esse acontecimento. Em geral possuem centenas ou milhares de linhas.

Um exemplo de uma tabela **fato** são as nossas Base de Vendas:

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente
TE10021	5	Recife	01/01/2020	576
TE10021	2	Brasília	01/01/2020	342
TE10019	1	Cuiabá	01/01/2020	437
TE10023	1	Florianópolis	01/01/2020	809
TE10016	5	Belo Horizonte	01/01/2020	810
TE10022	4	Campinas	01/01/2020	91
TE10001	2	Cuiabá	01/01/2020	704
TE10005	3	Fortaleza	01/01/2020	567
TE10011	2	Porto Alegre	01/01/2020	41
TE10015	4	Porto Alegre	01/01/2020	626
TE10006	4	Salvador	01/01/2020	186
TE10004	5	Porto Alegre	01/01/2020	766
TE10006	5	Brasília	01/01/2020	222
TE10018	5	Salvador	01/01/2020	258
TE10003	4	Belo Horizonte	01/01/2020	696
TE10011	4	Cuiabá	01/01/2020	458
TE10009	3	Cuiabá	01/01/2020	588
TE10016	5	Brasília	01/01/2020	401
TE10016	3	Fortaleza	01/01/2020	114

Ela funciona como um controle das vendas, com informações da loja onde a compra foi efetuada, qual produto foi vendido, a quantidade e etc. Em um mesmo ano podemos ter diversos produtos sendo vendidos em diferentes lojas, assim como diferentes registros de vendas para a mesma loja. Com base nisso, podemos identificar uma **particularidade** da **Tabela Fato**: nela vários dados (linhas) se repetem.

1.2.1.2 Tabela Características

São tabelas menores que representam características ou atributos sobre uma informação contida na tabela fato (nome dos clientes, nome do produto, entre outros). Exemplos de tabelas **características** são as tabelas de Cadastro de Clientes e Cadastro de Produtos:

Código Client	Primeiro Nome	Sobrenome	Sexo	Nº de Filhos	Data de Nascimento	E-mail	CPF
1	NATÁLIA	BARBOSA	F	1	3/16/1983	natalia78@icloud.com	98854964654
2	ROBERTO	FIGUEIREDO	M	0	4/9/1951	roberto46@gmail.com	33264169658
3	STELA	FERREIRA	F	1	9/12/1992	stela60@gmail.com	52259617998
4	RAFAEL	ARAÚJO	M	1	1/28/1958	rafael53@live.com	00771652825
5	BRENO	RIBEIRO	M	0	6/19/1982	breno49@icloud.com	95867983463
6	LUCAS	MESQUITA	M	0	5/10/2000	lucas23@hotmail.com	41957227222
7	DEBORA	RAMOS	F	0	11/27/1979	debora66@outlook.com	14655489345
8	ALEXANDRE	GUEDES	M	0	5/14/1972	alexandre42@yahoo.com	00779784787
9	JULIANA	MONTANHOLI	F	1	2/5/1977	juliana15@outlook.com	71458882949
10	GABRIELA	FONTOURA	F	1	2/15/1982	gabriela57@gmail.com	29394547534
11	HENRIQUE	CORREA	M	2	2/24/1984	henrique66@gmail.com	34474247454
12	JULIANA	FERREIRA	F	0	10/14/1975	juliana44@yahoo.com	86633893224
13	EDUARDO	SONE	M	0	8/5/1956	eduardo22@hotmail.com	41565439828
14	CAROLINA	HEIMLICH	F	0	3/3/1958	carolina92@yahoo.com	38438633324
15	FABIO	MARINS	M	1	1/11/1985	fabio22@terra.com.br	15173328848
16	BRUNO	MESQUITA	M	1	5/26/1951	bruno59@gmail.com	01955674715
17	BERNARDO	MELLO	M	2	8/11/1950	bernardo80@icloud.com	77652594391
18	MARIA	KOHN	F	1	3/3/1967	maria93@hotmail.com	87512228696
19	DÉBORA	SOARES	F	2	7/11/1968	debora82@outlook.com	21275846529

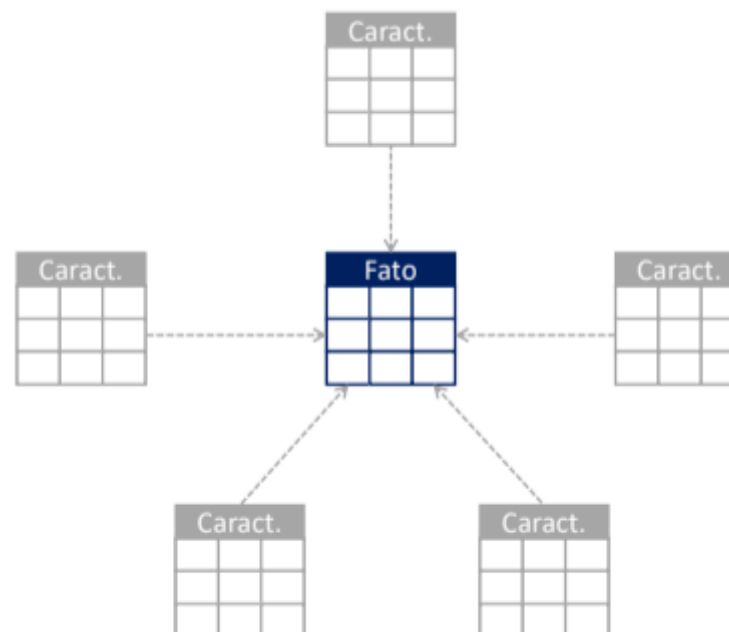
SKU	Produto	Marca	Tipo do Produto	Preço Unitário	Custo Unitário	Observação
TE10001	Lenovo Gamer Gaming I3 - Preto	Lenovo	Notebook	4200.00	3500.00	
TE10002	Asus MD15 - Prata	Asus	Notebook	3100.00	2400.00	
TE10003	Notebook Gamer Acer Nitro 5 - Vermelho e preto	Acer	Notebook	5100.00	4100.00	
TE10004	Notebook Gamer Dell G15 - Preto	Dell	Notebook	5900.00	5000.00	Conferir estoque
TE10005	Galaxy S23 Ultra - Prata	Samsung	Celular	6200.00	5000.00	
TE10006	iPhone 12 Pro Max - Branco	Apple	Celular	4800.00	4000.00	
TE10007	Zenfone 8 - Preto	Asus	Celular	2880.00	2100.00	
TE10008	Edge 30 Ultra - Cinza	Motorola	Celular	3750.00	3000.00	
TE10009	Watch Series 6 - Rose	Apple	Smart Watch	3500.00	2700.00	
TE10010	Galaxy Waych 4 - Preto	Samsung	Smart Watch	1150.00	800.00	
TE10011	Galaxy Tab S7 FE - Branco	Samsung	Tablet	3400.00	2600.00	
TE10012	iPad 10ª Geração - Preto	Apple	Tablet	4400.00	3300.00	
TE10013	Headset Gamer Cloud Stinger - Preto	HyperX	Headset	250.00	175.00	
TE10014	Headset Gamer G332 - Vermelho	Logitech	Headset	210.00	150.00	
TE10015	Headset Gamer Lamia 2 - Branco	Redragon	Headset	300.00	250.00	
TE10016	Monitor Gamer Hero - Vermelho e preto	AOC	Monitor	1200.00	820.00	
TE10017	Monitor Gamer Snow - Preto	Husky	Monitor	1100.00	750.00	
TE10018	Monitor Gamer LG 34 Ultra Wide - Preto	LG	Monitor	2400.00	1800.00	
TE10019	Controle p/ XBOX sem fio - Branco	Microsoft	Video Game	450.00	350.00	

Como dito acima, a tabela característica vai descrever as características de uma tabela fato.

Por exemplo, na base de cadastro dos produtos, um SKU refere-se apenas a um único produto e suas informações, como nome, cor, marca e etc. Quando olhamos para a tabela de vendas, a única informação que temos sobre o produto vendido é o seu código; para achar as informações referentes a esse produto, pegamos o código e vamos procurá-lo na tabela de produtos, sabendo que quando acharmos o código temos as informações sobre esse produto na mesma linha.

Chegamos aqui a uma **importante conclusão**: em uma **Tabela Característica**, as linhas **nunca se repetem**!! - não teremos um SKU igual para dois produtos, pois não faz sentido, assim como não teremos um mesmo código para dois clientes e assim sucessivamente.

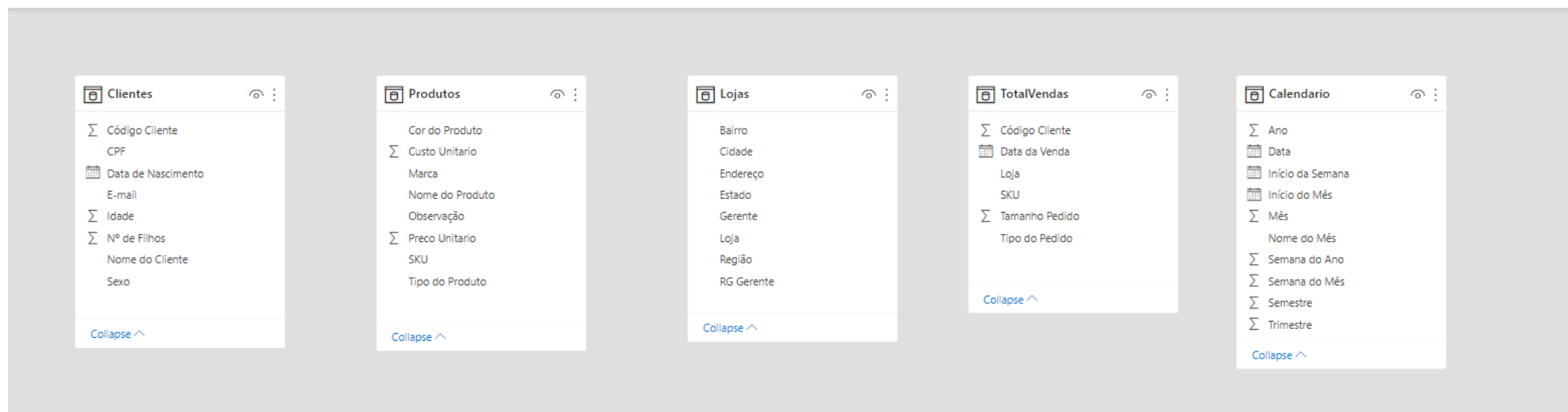
De forma genérica, podemos resumir a relação entre as tabelas fato e tabelas características da seguinte maneira: a Tabela Fato vai se associar com diferentes Tabelas Características para completar suas informações.



Voltando ao assunto de saber quando mesclar colunas e quando criar relacionamentos: de uma forma bem simples, quando temos **Tabelas Características**, se necessário vamos mesclar as colunas, pois temos poucas informações e não é necessário mantê-las em tabelas separadas. Já quando estamos falando de **Tabelas Fato**, que possuem suas milhares de linhas, faz mais sentido criar relacionamentos com as outras tabelas, pois relacionamos suas colunas sem a necessidade de repetir os dados várias e várias vezes.

1.2.1.3 Identificando as Tabelas Fato e Tabelas Características

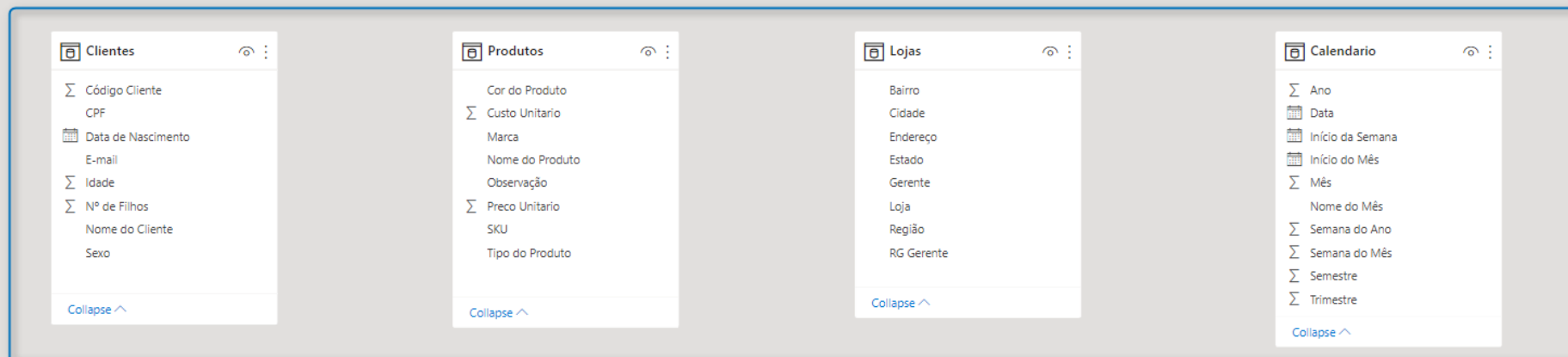
Para entendermos bem quais são essas tabelas, vamos analisar as tabelas da nossa loja:



Por padrão, para manter nosso projeto organizado, vamos seguir o seguinte formato: as **tabelas características** ficam na parte de cima e as **tabelas fato** ficam na parte de baixo.

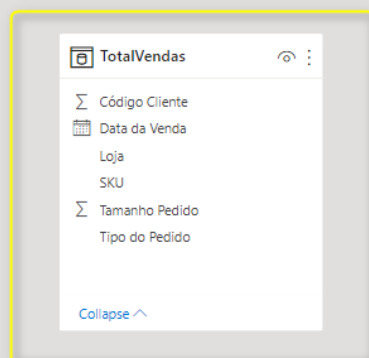
Dessa maneira, separe as tabelas fato e as tabelas características que aparecem acima. Para move-las basta clicar em cima e arrastar; busque sempre manter um espaçamento razoável entre elas para não ficar muito apertado.

Resposta



Clientes	Produtos	Lojas	Calendario
Σ Código Cliente	Cor do Produto	Bairro	Σ Ano
CPF	Σ Custo Unitário	Cidade	Data
Data de Nascimento	Marca	Endereço	Início da Semana
E-mail	Nome do Produto	Estado	Início do Mês
Σ Idade	Observação	Gerente	Σ Mês
Σ Nº de Filhos	Σ Preço Unitário	Loja	Nome do Mês
Nome do Cliente	SKU	Região	Σ Semana do Ano
Sexo	Tipo do Produto	RG Gerente	Σ Semana do Mês
			Σ Semestre
			Σ Trimestre

Tabelas Características



TotalVendas
Σ Código Cliente
Data da Venda
Loja
SKU
Σ Tamanho Pedido
Tipo do Pedido

Tabela Fato

A nossa única tabela **fato** é a base de vendas, que possui todas as informações sobre cada venda da nossa loja - o SKU do produto, o tamanho do pedido, loja e data da venda e o código do cliente que realizou a compra. As outras tabelas possuem as características referentes as informações da nossa tabela de vendas, por isso são as tabelas **características**. Elas trazem informações sobre os clientes, produtos, lojas e sobre as datas das vendas.

A principal tabela do nosso projeto é a tabela de vendas, justamente por ela trazer todos esses dados. As outras também são importantes, mas como complemento da nossa tabela principal.

1.2.2 Chave Primária x Chave Estrangeira

Esse é um conceito fundamental que precisamos entender antes de trabalhar com relacionamentos - ou qualquer outro software de banco de dados. Essas chaves, basicamente, se referem a determinadas colunas das tabelas que as identificam - ou seja, alguma informação presente nas linhas que seja única e que possibilite a busca dessa linha em uma determinada tabela.

Vamos dar uma olhada melhor:

1.2.2.1 Chave Primária

A coluna de chave primária é uma coluna com valores que não se repetem. Essas colunas estão associadas as tabelas características, onde cada linha representa uma informação única - sobre um único produto ou um único cliente -, e podem ser usadas como referência em outras tabelas.

Um dos exemplos existentes no nosso projeto é a coluna Código Cliente presente na tabela Clientes, onde cada código está associado com uma pessoa, e pode ser usado justamente para acessar as informações sobre essa pessoa.

Código Cliente	Primeiro Nome	Sobrenome	Sexo	Nº de Filhos	Data de Nascimento	E-mail	CPF
1	NATÁLIA	BARBOSA	F	1	3/16/1983	natalia78@icloud.com	98854964654
2	ROBERTO	FIGUEIREDO	M	0	4/9/1951	roberto46@gmail.com	33264169658
3	STELA	FERREIRA	F	1	9/12/1992	stela60@gmail.com	52259617998
4	RAFAEL	ARAUJO	M	1	1/28/1958	rafael53@live.com	00771652825
5	BRENO	RIBEIRO	M	0	6/19/1982	breno49@icloud.com	95867983463
6	LUCAS	MESQUITA	M	0	5/10/2000	lucas23@hotmail.com	41957227222
7	DEBORA	RAMOS	F	0	11/27/1979	debora66@outlook.com	14655489345
8	ALEXANDRE	GUEDES	M	0	5/14/1972	alexandre42@yahoo.com	00779784787
9	JULIANA	MONTANHOLI	F	1	2/5/1977	juliana15@outlook.com	71458882949
10	GABRIELA	FONTOURA	F	1	2/15/1982	gabriela57@gmail.com	29394547534
11	HENRIQUE	CORREA	M	2	2/24/1984	henrique66@gmail.com	34474247454
12	JULIANA	FERREIRA	F	0	10/14/1975	juliana44@yahoo.com	86633893224
13	EDUARDO	SONE	M	0	8/5/1956	eduardo22@hotmail.com	41565439828
14	CAROLINA	HEIMLICH	F	0	3/3/1958	carolina92@yahoo.com	38438633324
15	FABIO	MARINS	M	1	1/11/1985	fabio22@terra.com.br	15173328848
16	BRUNO	MESQUITA	M	1	5/26/1951	bruno59@gmail.com	01955674715
17	BERNARDO	MELLO	M	2	8/11/1950	bernardo80@icloud.com	77652594391
18	MARIA	KOHN	F	1	3/3/1967	maria93@hotmail.com	87512228696
19	DÉBORA	SOARES	F	2	7/11/1968	debora82@outlook.com	21275846529
20	BEATRIZ	SILVA	F	0	6/15/1979	beatriz79@outlook.com	04685496764

1.2.2.2 Chave Estrangeira

Enquanto a Chave Primária é utilizada como referência de uma informação em uma tabela, a Chave Estrangeira é justamente a coluna, em outra tabela, que possui a referência que vamos buscar na tabela com as informações.

Por exemplo, na nossa tabela de vendas temos a coluna Código Cliente, que nos informa qual cliente realizou a compra; entretanto, não temos as outras informações sobre esse cliente - nome, idade, etc. Então, usamos esses códigos para ir na tabela Clientes e acessar essas informações.

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente
TE10021	5	Recife	01/01/2020	576
TE10021	2	Brasília	01/01/2020	342
TE10019	1	Cuiabá	01/01/2020	437
TE10023	1	Florianópolis	01/01/2020	809
TE10016	5	Belo Horizonte	01/01/2020	810
TE10022	4	Campinas	01/01/2020	91
TE10001	2	Cuiabá	01/01/2020	704
TE10005	3	Fortaleza	01/01/2020	567
TE10011	2	Porto Alegre	01/01/2020	41
TE10015	4	Porto Alegre	01/01/2020	626
TE10006	4	Salvador	01/01/2020	186
TE10004	5	Porto Alegre	01/01/2020	766
TE10006	5	Brasília	01/01/2020	222
TE10018	5	Salvador	01/01/2020	258
TE10003	4	Belo Horizonte	01/01/2020	696
TE10011	4	Cuiabá	01/01/2020	458
TE10009	3	Cuiabá	01/01/2020	588
TE10016	5	Brasília	01/01/2020	401
TE10016	3	Fortaleza	01/01/2020	114
TE10015	4	Porto Alegre	01/01/2020	346

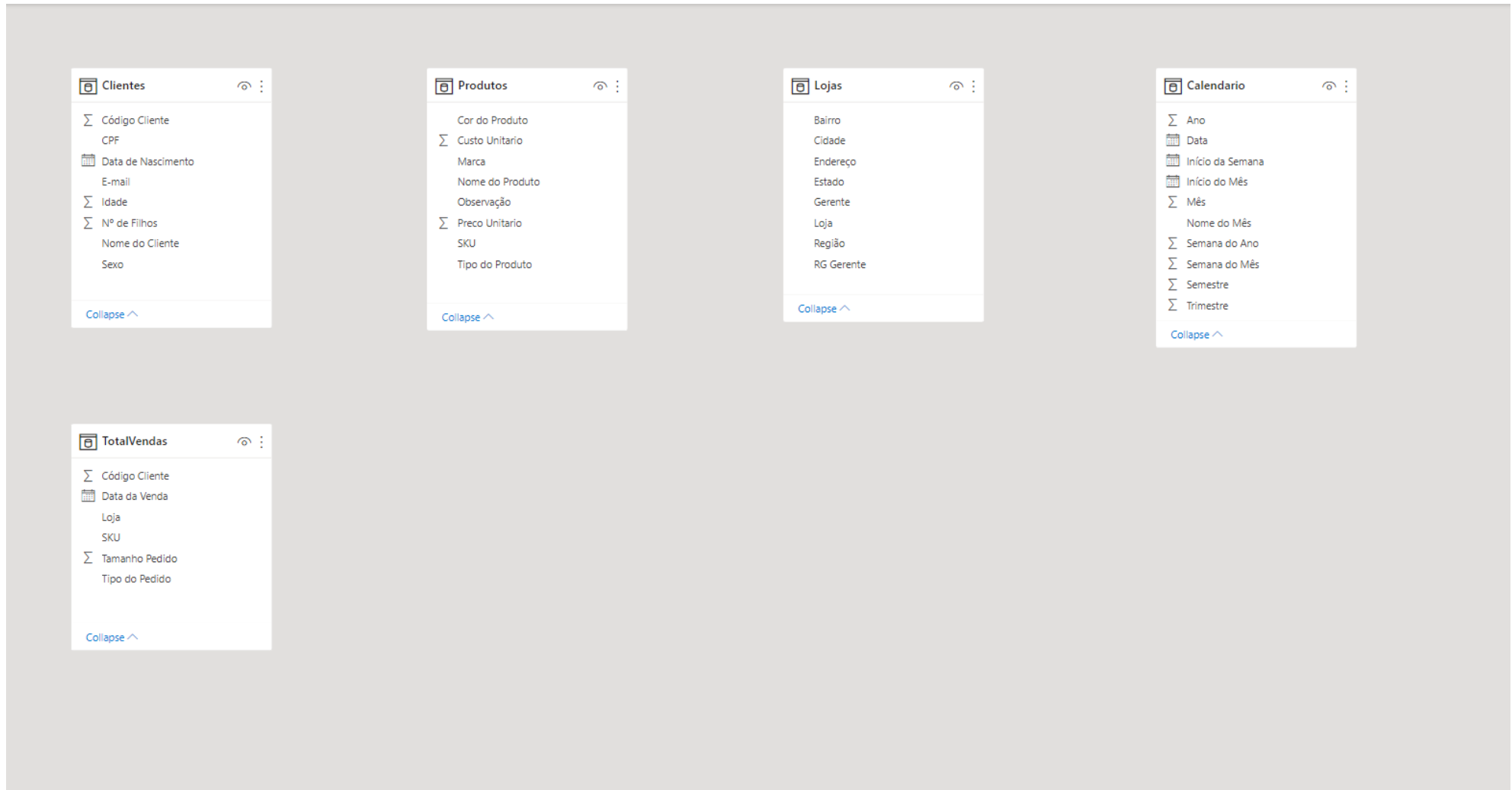
Importante: para existir uma Chave Estrangeira, **precisa** existir uma Chave Primária correspondente em algum lugar.

Isso está associado com o outro conceito que acabamos de ver, de Tabelas Fato e Tabelas Características:

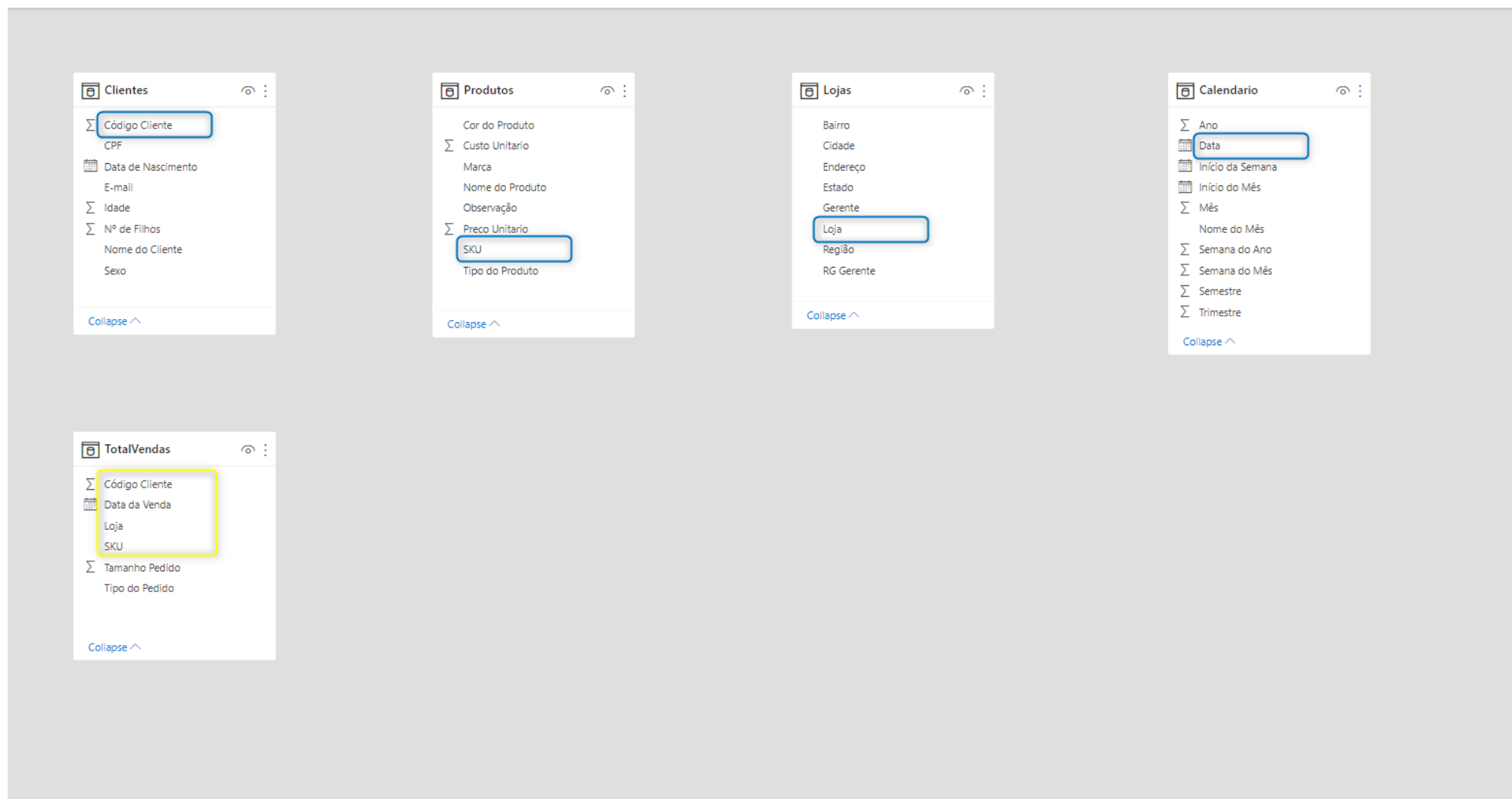
- Nas Tabelas Fato temos as informações sobre fatos ocorridos, o que nos leva a dados repetidos nas colunas; dessa forma, essas colunas possuem as **Chaves Estrangeiras**, que serão usadas para buscar as informações complementares nas respectivas tabelas;
- Já nas Tabelas Características, que são essas tabelas onde vamos procurar as informações complementares, temos as **Chaves Primárias** - que não se repetem e são usadas para acessar os detalhes de algum elemento.

1.2.2.3 Identificando as Chaves Primárias e Chaves Estrangeiras

Para fixar bem o conceito, vamos voltar as tabelas do nosso projeto e identificar as chaves primárias e as estrangeiras:



Resposta



Como vimos, as **Chaves Primárias** estão relacionadas com as **Tabelas Características**, que estão acima no nosso modelo, e as **Chaves Estrangeiras** com as **Tabelas Fato**, que é nossa tabela na parte de baixo.

Obs. 1: nem toda coluna das tabelas características pode ser usada como chave primária. Por exemplo, na tabela dos clientes a coluna Sexo possui apenas "Masculino" e "Feminino"; é impossível identificar alguém precisamente apenas com essa informação.

Obs. 2: precisamos analisar também como relacionaremos essas colunas, já que é justamente essa a intenção de identificar as chaves das tabelas. Poderíamos usar a coluna CPF da tabela Clientes como chave primária, já que o CPF é único de cada indivíduo; porém, essa informação não está presente nas outras tabelas, o que torna impossível esse relacionamento. Por isso usamos a coluna com o código dos clientes, que é uma informação comum em todas as tabelas.

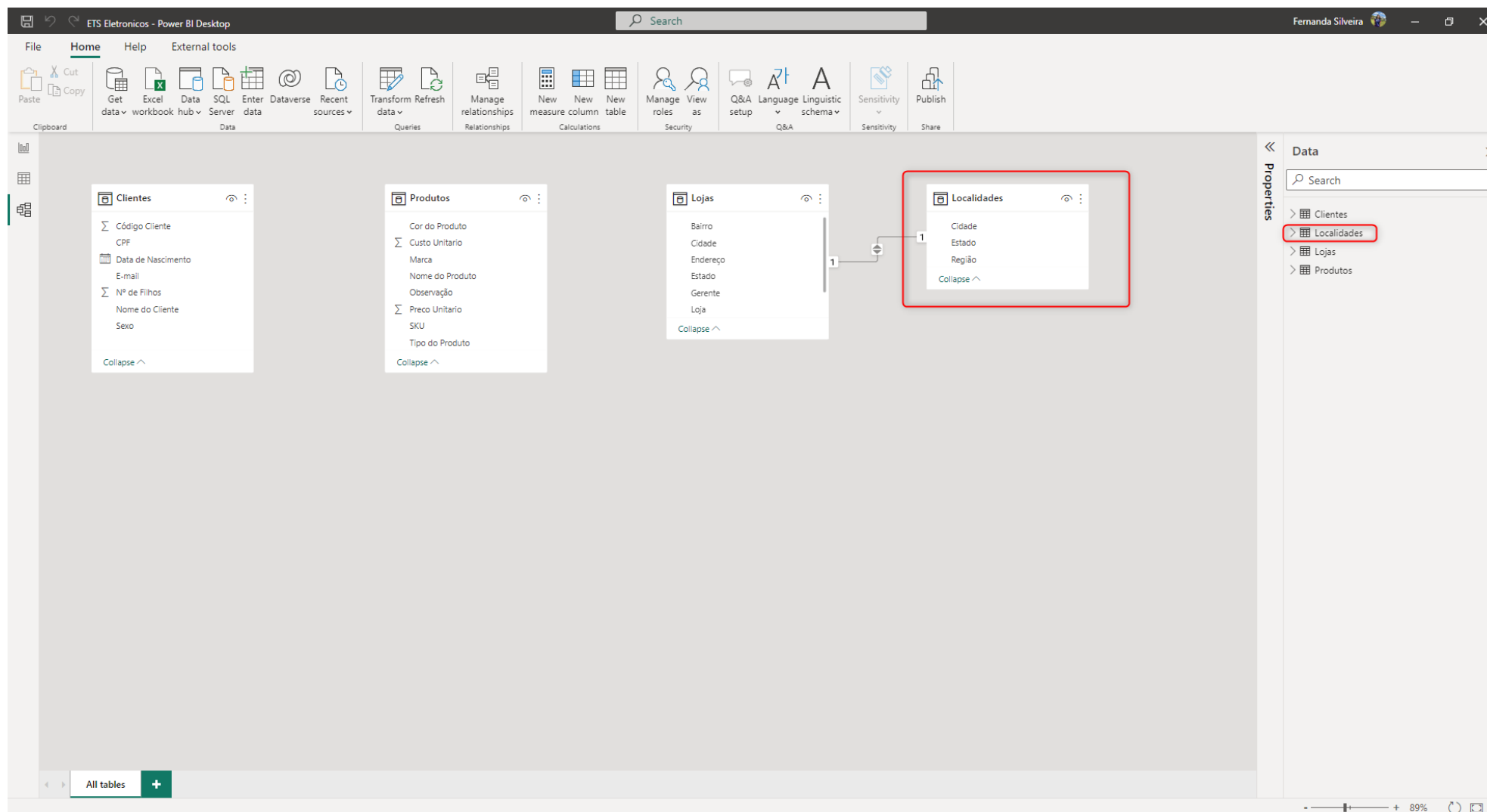
Obs. 3: a coluna Loja foi usada, porque apesar de, nesse caso, a loja ser o próprio nome da cidade, caso fosse um nome diferente faria mais sentido relacionar essas colunas do que a coluna com a cidade.

1.2.3 Cardinalidade dos Relacionamentos

A cardinalidade de um relacionamento se refere a quantidade de informações que podem existir entre as duas tabelas relacionadas. Existem três tipos principais de cardinalidade: **um para um (1:1)**, **um para muitos (1:M)** e **muitos para muitos (N:M)**.

1.2.3.1 Cardinalidade um para um (1:1)

Neste tipo de relacionamento, uma informação de uma tabela está relacionada no máximo a uma informação de uma outra tabela. Quando vimos a ferramenta de mesclar colunas, entre as duas tabelas foi criada uma relação de 1 para 1:



Veremos isso melhor mais a frente, mas a linha entre as tabelas Lojas e Localidades representa esse relacionamento.

Como vimos no exemplo, relacionamentos de 1:1 não fazem muito sentido para nós no Power BI, porque quando uma informação de uma tabela complementa uma única informação de uma outra tabela faz mais sentido juntarmos essas informações em apenas uma tabela.

1.2.3.2 Cardinalidade um para muitos (1:M)

Neste tipo de relacionamento, uma informação da tabela **X** está associada a várias informações da tabela **Y**, mas uma informação da tabela **Y** está associada a apenas uma informação da tabela **X**.

Por exemplo, podemos analisar nossa tabela de Vendas e a tabela com os Produtos: o código de um produto está associado a várias linhas na base de vendas - porque um mesmo produto foi vendido diversas vezes -, mas quando pegamos uma venda específica temos apenas um produto correspondente na tabela de vendas. É o tipo de relacionamento que vamos usar sempre no Power BI.

1.2.3.3 Cardinalidade muitos para muitos (N:M)

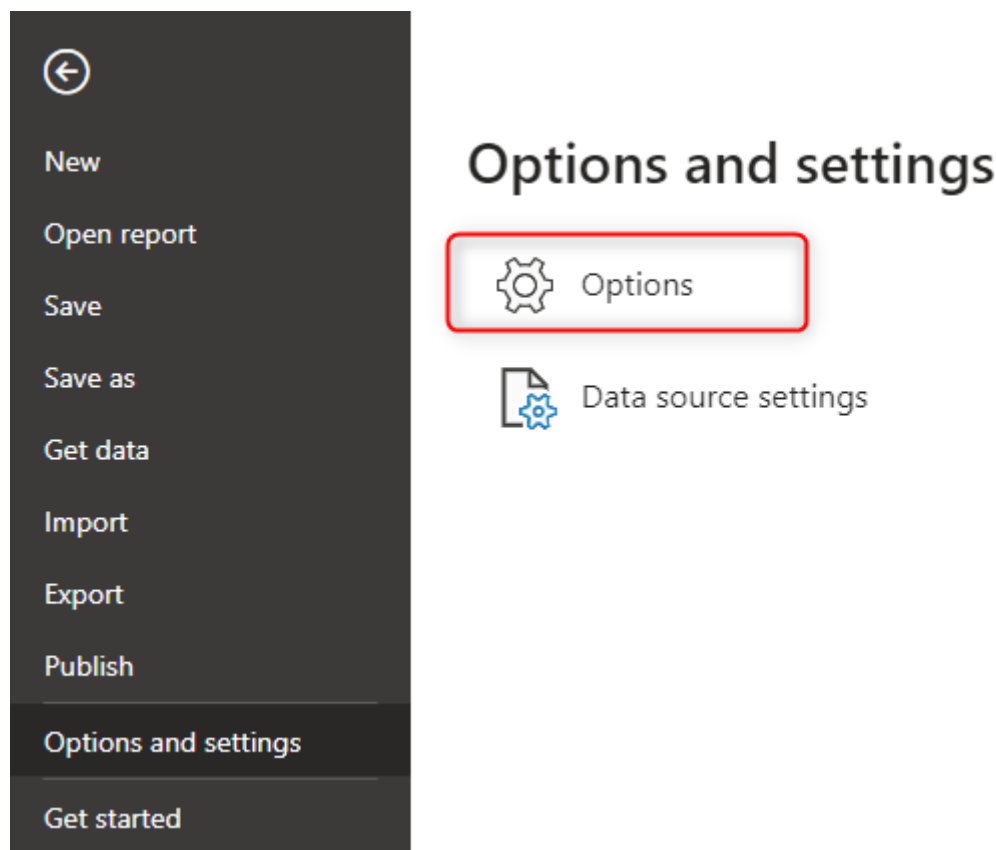
É como se relacionássemos duas tabelas de vendas entre si, ou a base de vendas com a base de devoluções, por exemplo. Esse tipo de relacionamento não funciona na prática, pois ao procurarmos a informação de uma tabela na outra vamos encontrar diversas correspondências, o que não faz sentido.

Para que esse relacionamento funcione, é necessária a criação de uma tabela intermediária, e um relacionamento de muitos para muitos vira dois relacionamentos de 1 para muitos. Não veremos isso nesse curso, pois isso está mais relacionado a área de banco de dados. Dentro do Power BI, se acharmos algo desse estilo, precisamos procurar por uma tabela característica para usá-la - ele nem nos permite criar um relacionamento N:M.

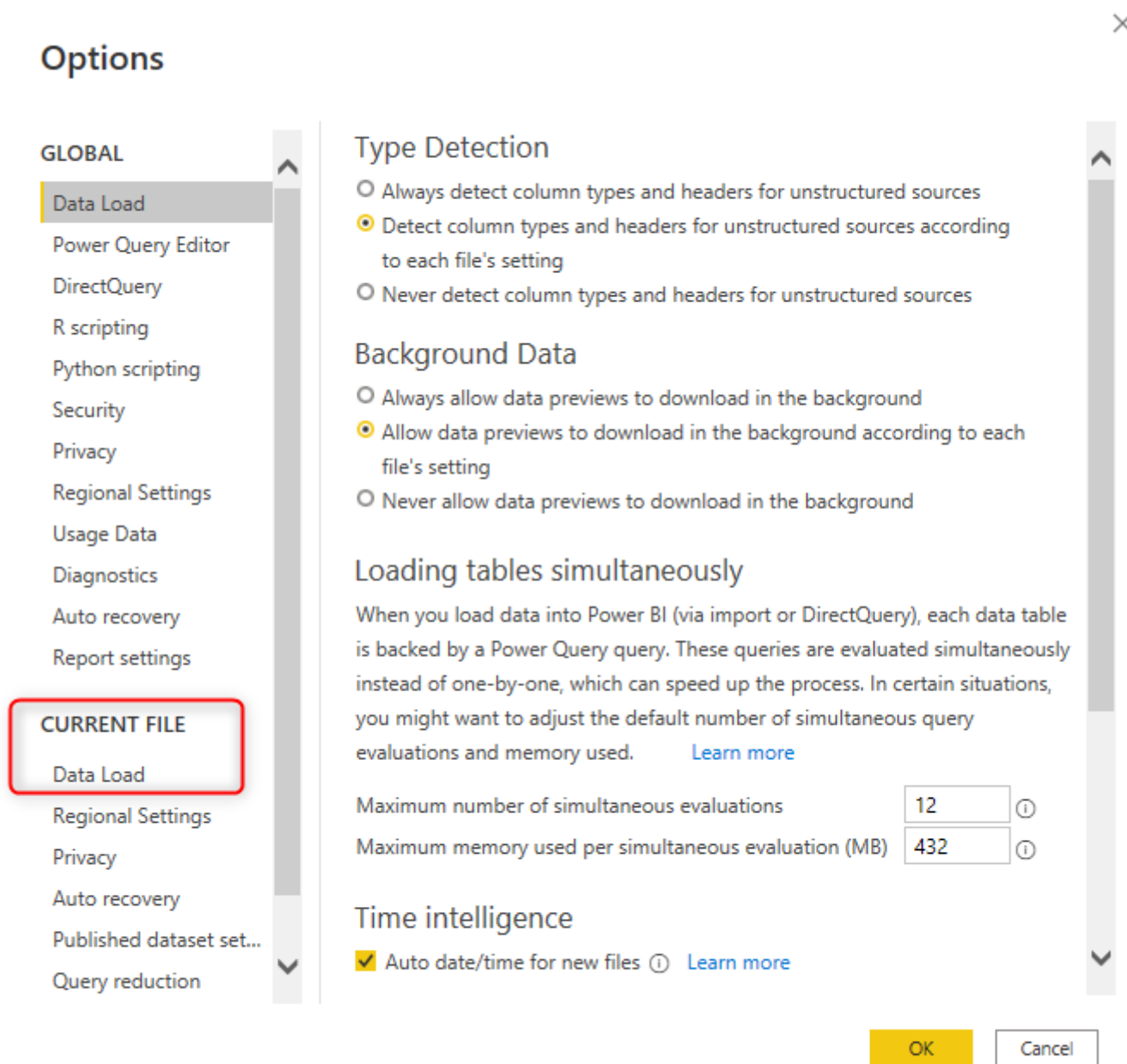
2 3.2 - Criando e Gerenciando Relacionamentos

Vamos mostrar agora na prática a diferença que os relacionamentos causam no nosso BI e como criamos/editamos esses relacionamentos.

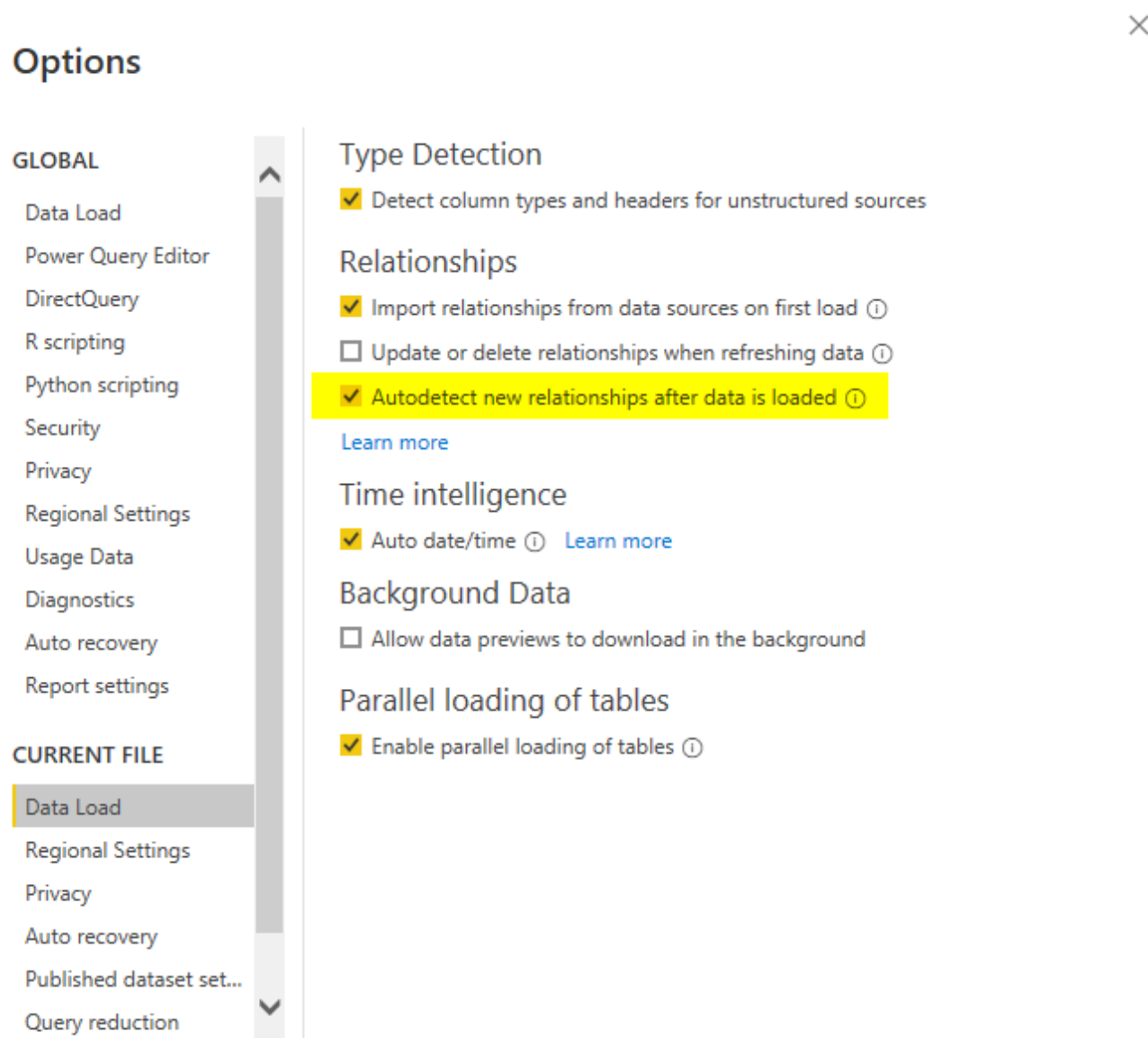
Primeiramente, antes de começarmos a criar os relacionamentos precisamos desabilitar a detecção automática, para que possamos construí-los do zero. Para isso, vamos na guia **Arquivo** (*File*), na opção **Opções e configurações** (*Options and settings*) e escolhemos **Opções** (*Options*):



Na janela de opções que é aberta, vamos até a parte de opções do nosso arquivo atual:



Na nova janela, **desmarcamos** a opção que habilita a detecção automática de relacionamentos assim que os dados são carregados:

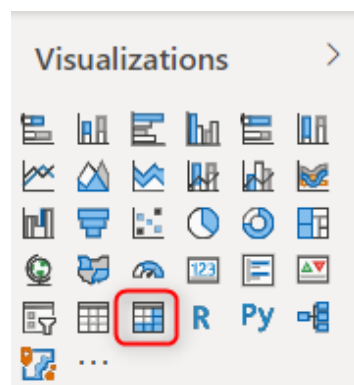


OK

Cancel

Agora sim! Tudo certo para irmos para a aula!

Para começar, vamos criar uma matriz simples para vermos a relação do total de pedidos por loja. Para selecionar essa visualização, na guia **Relatório** (Report) vamos no grupo **Visualizações** (Visualizations) na **Página Inicial** (Home) e selecionamos seu símbolo:



Não precisamos nos preocupar em como personalizá-la por agora; vamos nos atentar as suas informações. Usando as informações da tabela TotalVendas, no campo **Linhas** (Rows) colocamos nossas Lojas - já que queremos saber quanto foi vendido por loja -, e no campo **Valores** (Values) o a nossa coluna Tamanho Pedido, que é a soma dos valores da nossa coluna:

The image shows a data visualization tool interface. On the left is a table with two columns: 'Loja' and 'Tamanho Pedido'. The table lists 14 cities and their corresponding order sizes, with a total of 71442. On the right is a configuration panel with three sections: 'Rows', 'Columns', and 'Values'. The 'Rows' section has 'Loja' selected. The 'Columns' section has a placeholder 'Add data fields here'. The 'Values' section has 'Tamanho Pedido' selected. Below these sections is a 'Drill through' section with a 'Cross-report' toggle set to 'Off' and a 'Keep all filters' toggle set to 'On'. There is also a placeholder 'Add drill-through fields here'.

Loja	Tamanho Pedido
Belo Horizonte	5070
Brasília	4948
Campinas	4950
Cuiabá	9895
Curitiba	5080
Florianópolis	5273
Fortaleza	5267
Porto Alegre	5159
Recife	5149
Rio de Janeiro	5250
Salvador	5242
São Paulo	4957
Vila Velha	5202
Total	71442

Rows

Loja

Columns

Add data fields here

Values

Tamanho Pedido

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Aqui ele vai pegar essa soma que ele fez e dividir de acordo com os campos, que nesse caso são das lojas. Podemos ver que ele calculou corretamente o tamanho dos pedidos, separando por loja conforme nós queremos. Agora, vamos analisar outro caso, que é a relação entre o tamanho do pedido e os produtos:

The image shows a data visualization tool interface. On the left is a table with two columns: 'Nome do Produto' and 'Tamanho Pedido'. The table lists various electronic products and their corresponding order sizes, with a total at the bottom. On the right is a configuration panel with sections for 'Rows', 'Columns', 'Values', and 'Drill through'. The 'Rows' section has 'Nome do Produto' selected. The 'Values' section has 'Tamanho Pedido' selected. The 'Drill through' section has 'Cross-report' set to 'Off' and 'Keep all filters' set to 'On'.

Nome do Produto	Tamanho Pedido
Zenfone 8	71442
Watch Series 6	71442
Notebook Gamer Dell G15	71442
Notebook Gamer Acer Nitro 5	71442
Monitor Gamer Snow	71442
Monitor Gamer LG 34 Ultra Wide	71442
Monitor Gamer Hero	71442
Lenovo Gamer Gaming I3	71442
iPhone 12 Pro Max	71442
iPad 10ª Geração	71442
Headset Gamer Lamia 2	71442
Headset Gamer G332	71442
Headset Gamer Cloud Stinger	71442
Galaxy Waych 4	71442
Galaxy Tab S7 FE	71442
Galaxy S23 Ultra	71442
Edge 30 Ultra	71442
EarPods	71442
Controle p/ XBOX sem fio	71442
Controle Dualshock 4 PS4	71442
Controle DualSense PS5	71442
Buds 4 Pro	71442
Asus MD15	71442
AirPods Pro (3ª geração)	71442
Total	71442

Rows
Nome do Produto

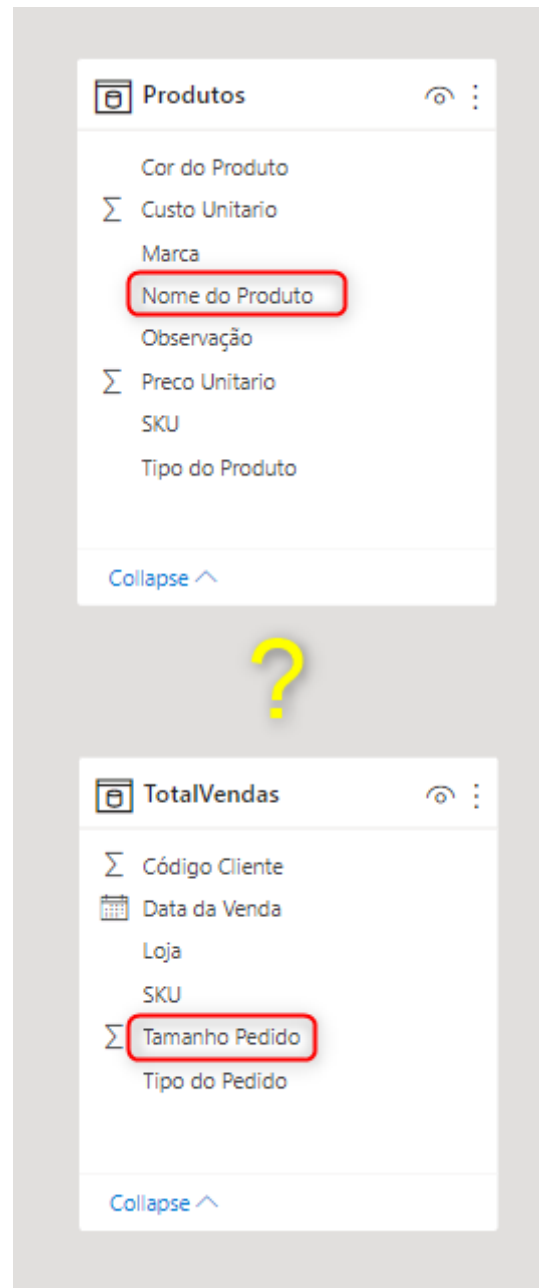
Columns
Add data fields here

Values
Tamanho Pedido

Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here

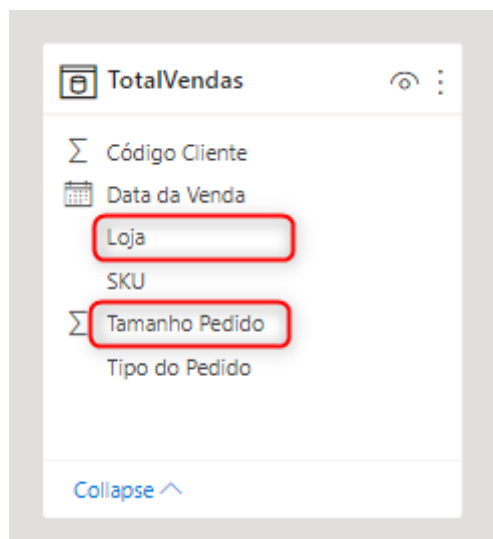
Ele repetiu o 71442 para todos os produtos, o que obviamente não está correto já que esse número representa o total de vendas que tivemos. Porque o exemplo anterior deu certo, mas esse não?

Quando olhamos para as nossas tabelas e suas colunas conseguimos entender isso:

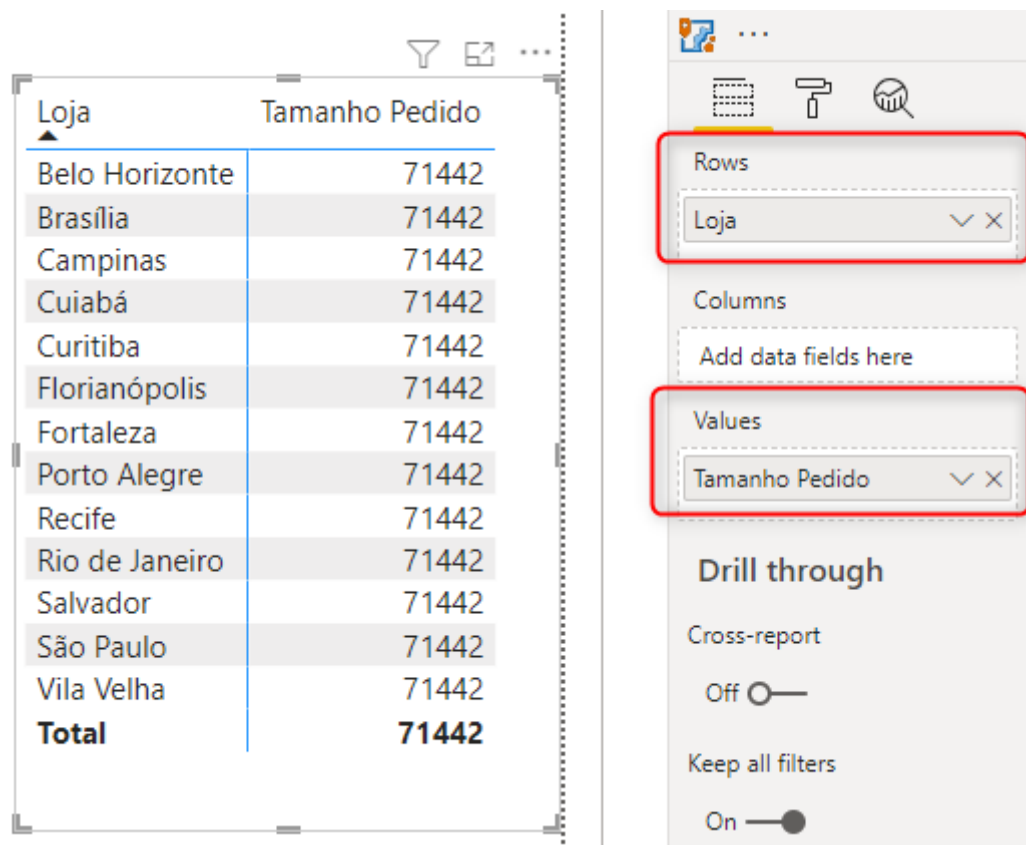


Não temos nenhuma relação entre essas tabelas; portanto, o Power BI não entende que existe uma relação entre a coluna nome dos produtos que usamos com o tamanho dos pedidos. Mas então porque a primeira matriz que fizemos funcionou?

Reparando na nossa tabela TotalVendas, temos a informação sobre a loja nela. Na hora de montar a matriz, usamos as seguintes colunas:



Como ambas as colunas estão na mesma tabela, o Power BI conseguiu identificar os valores e fazer os cálculos. Quando usamos a coluna Loja da tabela Lojas, nosso resultado é diferente:



The image shows a Power BI interface. On the left is a table with two columns: 'Loja' and 'Tamanho Pedido'. The table lists 14 cities and a total row, all with the value 71442. On the right is a visual configuration pane. The 'Rows' section has 'Loja' selected. The 'Columns' section is empty with the text 'Add data fields here'. The 'Values' section has 'Tamanho Pedido' selected. Below these are 'Drill through' and 'Keep all filters' options, both currently turned off.

Loja	Tamanho Pedido
Belo Horizonte	71442
Brasília	71442
Campinas	71442
Cuiabá	71442
Curitiba	71442
Florianópolis	71442
Fortaleza	71442
Porto Alegre	71442
Recife	71442
Rio de Janeiro	71442
Salvador	71442
São Paulo	71442
Vila Velha	71442
Total	71442

Apesar de ambas colunas terem as mesmas informações, elas não estão relacionadas; logo, o Power BI não entende essa relação e não calcula corretamente para nós esses valores.

Agora vocês podem estar se perguntando: porque usar a coluna da tabela Lojas, se usando a coluna da tabela TotalVendas eu não tenho esse problema?

Voltando a nossa matriz do total de vendas por produto, nós temos um problema para montar essa matriz: a única informação que temos na base de vendas sobre o produto é o SKU dele. Podemos montar uma matriz usando essa informação, mas não faz sentido para quem for visualizar esses dados:

The image shows a data visualization tool interface. On the left is a table with two columns: 'SKU' and 'Tamanho Pedido'. The table contains 24 rows of data, with the last row being a 'Total' row. On the right is a configuration panel with three sections: 'Rows', 'Columns', and 'Values'. The 'Rows' section has a dropdown menu with 'SKU' selected. The 'Columns' section has a placeholder 'Add data fields here'. The 'Values' section has a dropdown menu with 'Tamanho Pedido' selected. Below these sections is a 'Drill through' section with a 'Cross-report' toggle set to 'Off', a 'Keep all filters' toggle set to 'On', and a placeholder 'Add drill-through fields here'.

SKU	Tamanho Pedido
TE10001	3030
TE10002	3081
TE10003	3001
TE10004	2864
TE10005	2955
TE10006	3024
TE10007	3048
TE10008	2958
TE10009	2852
TE10010	2903
TE10011	2969
TE10012	2961
TE10013	3143
TE10014	2875
TE10015	2961
TE10016	2977
TE10017	2821
TE10018	2914
TE10019	3061
TE10020	3063
TE10021	3028
TE10022	3075
TE10023	3024
TE10024	2854
Total	71442

Rows

SKU

Columns

Add data fields here

Values

Tamanho Pedido

Drill through

Cross-report

Off

Keep all filters

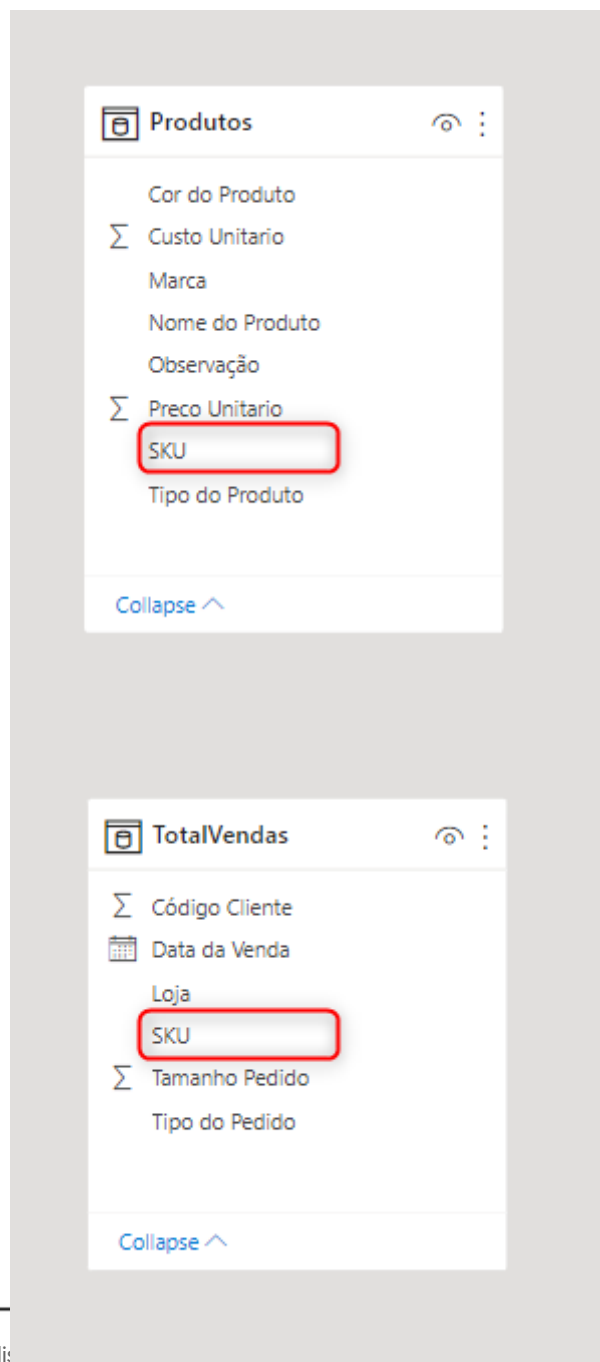
On

Add drill-through fields here

Agora ele calculou da maneira correta, porque usamos os dados da mesma tabela. Mas o que é TE10001? E TE10002? Para conseguirmos ter a informação completa, com o nome dos produtos e suas vendas, assim como as outras informações da tabela, precisamos relacionar suas colunas!

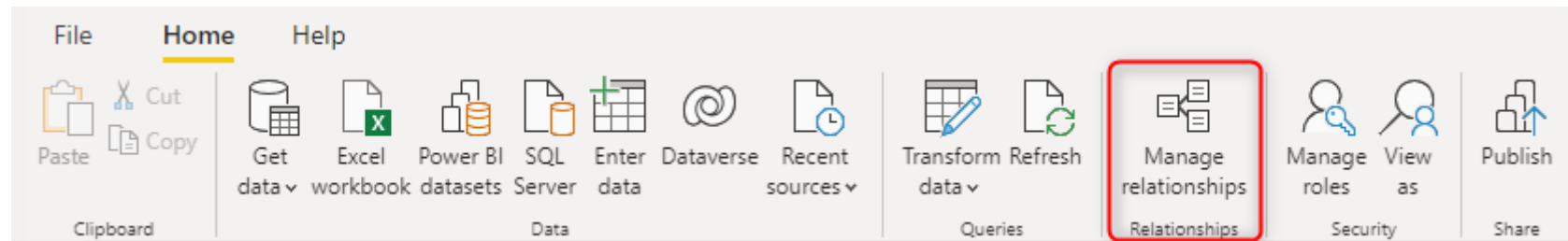
2.1 Criando Relacionamentos

Como vimos, precisamos relacionar a tabela com as vendas com a tabela que tem as informações sobre os produtos, para que ele consiga calcular os valores corretamente para nós. O primeiro passo para criar esse relacionamento é olhar para as tabelas e encontrar qual coluna elas tem em comum - aquela relação entre chave primária e chave estrangeira que vimos lá no começo. Essa coluna é o SKU, pois nas duas tabelas é ela que contém as informações correspondentes (os códigos dos produtos):



Sabendo disso, podemos criar esse relacionamento de duas maneiras:

- **Primeira maneira:** na guia **Modelo** (*Model*), na **Página Inicial** (*Home*) temos um grupo chamado **Relações** (*Relationships*), onde temos a ferramenta **Gerenciar Relações** (*Manage Relationships*):



Na nova janela, vamos em **Nova** (*New*) para criar uma nova relação:

Manage relationships



Active	From: Table (Column)	To: Table (Column)
There are no relationships defined yet.		

New...

Autodetect...

Edit...

Delete

[Close](#)

Ele abre uma nova guia para nós, com todas as informações em branco:

Create relationship



Select tables and columns that are related.

Cardinality

Cross filter direction

☐ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Basta selecionar as tabelas que ele já configura a relação para nós:



Create relationship

Select tables and columns that are related.

Produtos

SKU	Nome do Produto	Cor do Produto	Marca	Tipo do Produto	Preço Unitário	Custo Unitário
TE10001	Lenovo Gamer Gaming I3	Preto	Lenovo	Notebook	4200	3
TE10002	Asus MD15	Prata	Asus	Notebook	3100	2
TE10003	Notebook Gamer Acer Nitro 5	Vermelho e preto	Acer	Notebook	5100	4

TotalVendas

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido
TE10009	3	Cuiabá	quarta-feira, 1 de janeiro de 2020	588	Pedido Múltiplo
TE10005	3	Cuiabá	sábado, 4 de janeiro de 2020	48	Pedido Múltiplo
TE10015	3	Cuiabá	terça-feira, 7 de janeiro de 2020	340	Pedido Múltiplo

Cardinality

One to many (1:*)

Cross filter direction

Single

☒ Make this relationship active


☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Clicando em OK ele cria a relação para nós. Temos ali algumas opções como Direção do Filtro e Relação Ativa, veremos isso mais a frente; temos também a cardinalidade que nós já vimos, que aqui é de **1:*** (*one to many*):

A screenshot of a software interface showing a dropdown menu for 'Cardinality'. The dropdown is open, displaying the selected option 'One to many (1:*)'. The text 'Cardinality' is visible above the dropdown box.

Essa forma é um pouco mais trabalhosa, usamos mais para editar as relações, suas opções e afins. Em geral, vamos para a segunda maneira. Para excluir a relação que criamos, basta clicar com o botão direito e ir em **Excluir** (*Delete*), ou na mesma janela que a gente estava também temos a opção **Excluir** (*Delete*):



Manage relationships

Active	From: Table (Column)	To: Table (Column)
<input checked="" type="checkbox"/>	TotalVendas (SKU)	Produtos (SKU)

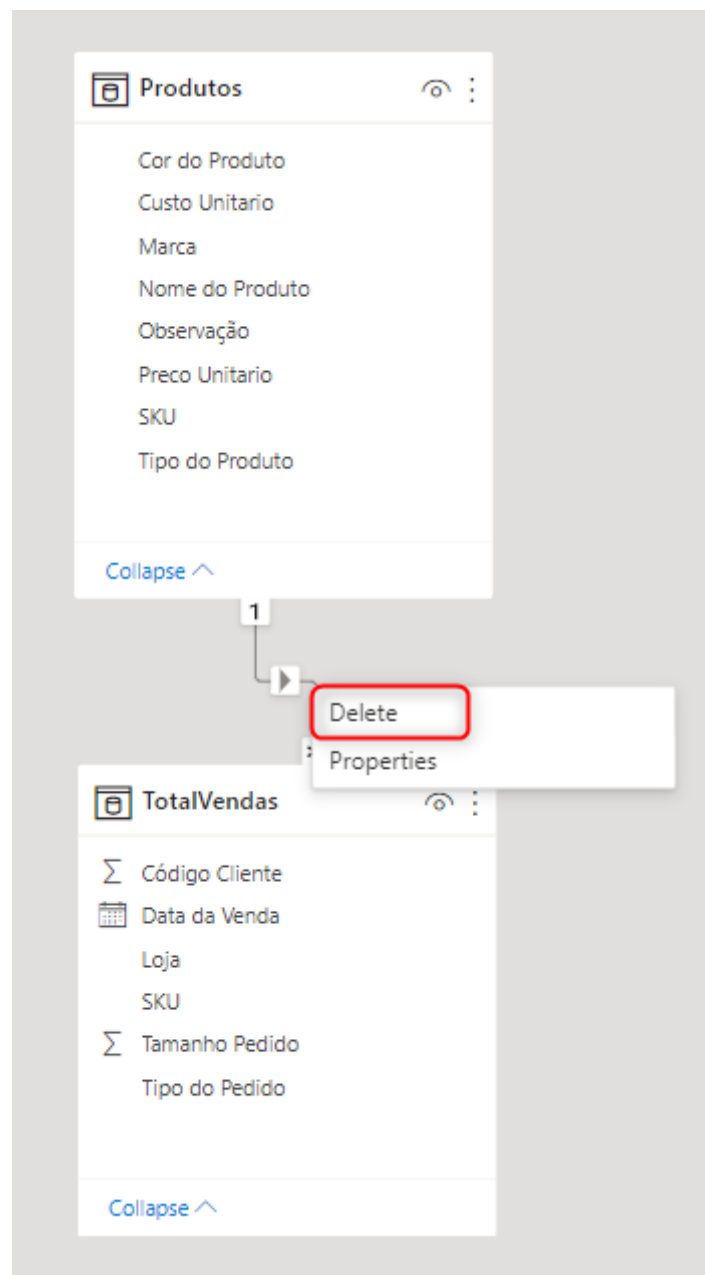
New...

Autodetect...

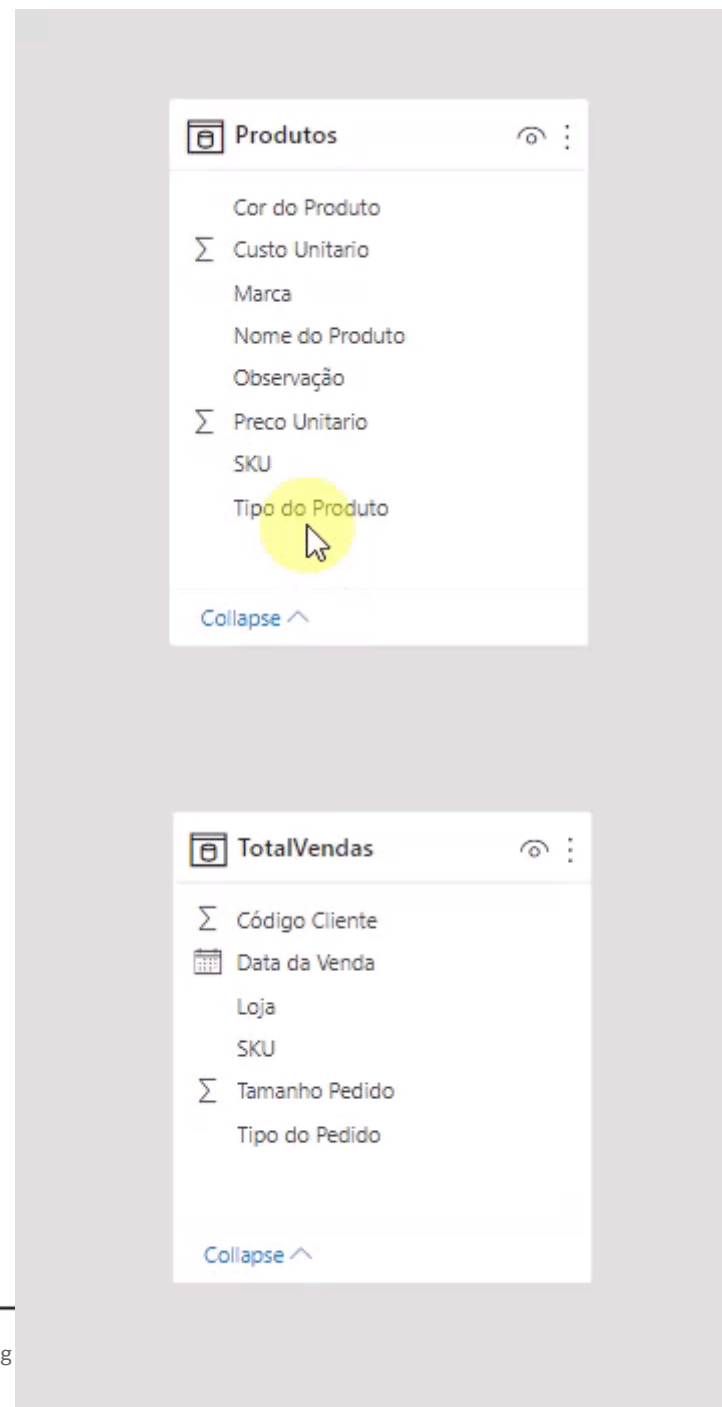
Edit...

Delete

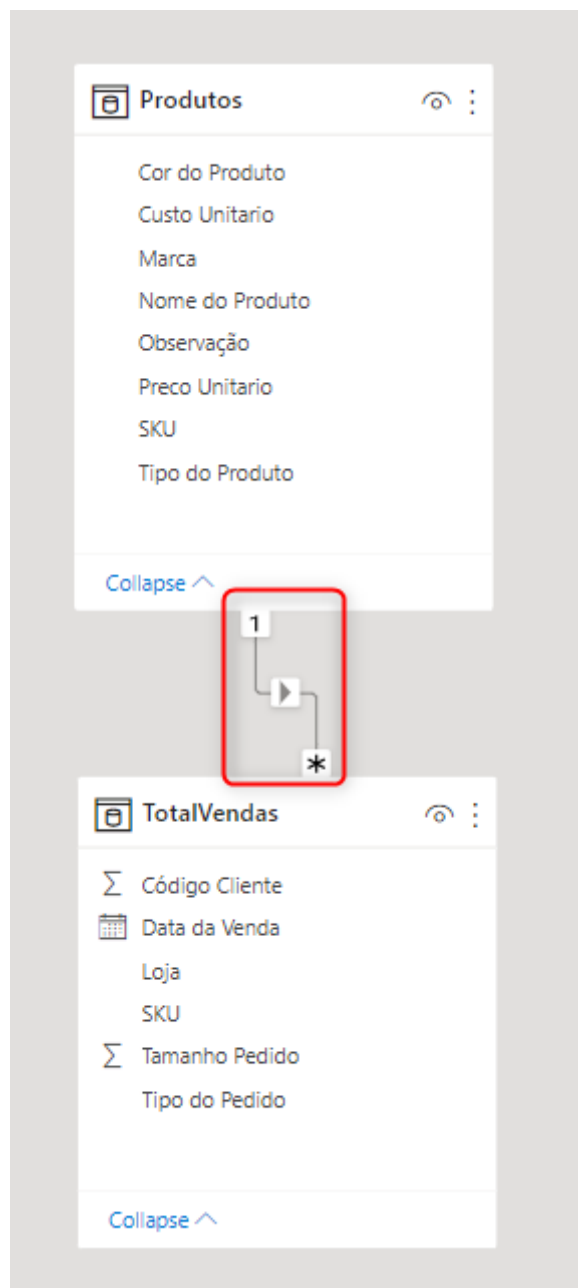
Close



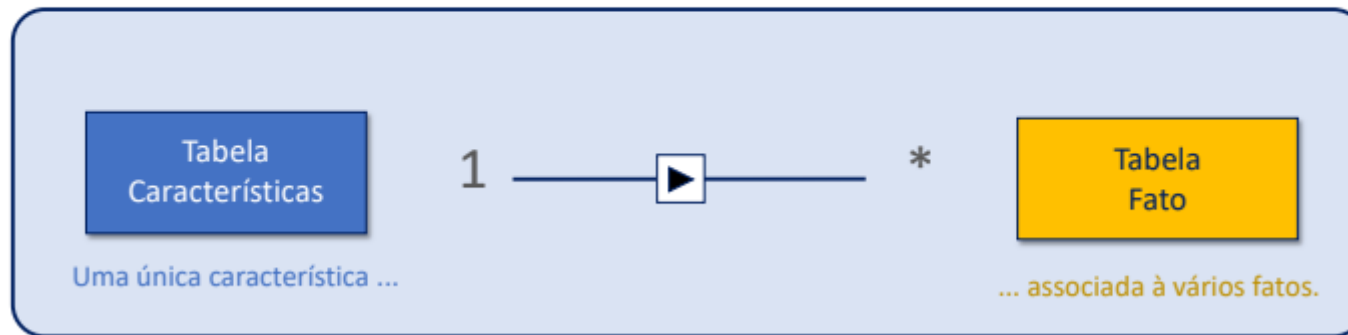
- **Segunda maneira:** basta clicar em cima de uma das colunas e arrastar até a outra, independentemente de ir de cima para baixo ou de baixo para cima, que a relação é criada:



O relacionamento criado é representado por uma linha, que mostra também a cardinalidade do relacionamento - uma linha na tabela Produtos corresponde a várias linhas na tabela TotalVendas:



Podemos então estabelecer a seguinte relação entre Cardinalidade e Tabelas Fato/Características:



Voltando ao nosso relatório com o nome dos produtos e a quantidade vendida:

Nome do Produto	Tamanho Pedido
AirPods Pro (3ª geração)	3075
Asus MD15	3081
Buds 4 Pro	3024
Controle DualSense PS5	3028
Controle Dualshock 4 PS4	3063
Controle p/ XBOX sem fio	3061
EarPods	2854
Edge 30 Ultra	2958
Galaxy S23 Ultra	2955
Galaxy Tab S7 FE	2969
Galaxy Waych 4	2903
Headset Gamer Cloud Stinger	3143
Headset Gamer G332	2875
Headset Gamer Lamia 2	2961
iPad 10ª Geração	2961
iPhone 12 Pro Max	3024
Lenovo Gamer Gaming I3	3030
Monitor Gamer Hero	2977
Monitor Gamer LG 34 Ultra Wide	2914
Monitor Gamer Snow	2821
Notebook Gamer Acer Nitro 5	3001
Notebook Gamer Dell G15	2864
Watch Series 6	2852
Zenfone 8	3048
Total	71442

Rows

Nome do Produto

Columns

Add data fields here

Values

Tamanho Pedido

Drill through

Cross-report

Off

Keep all filters

On

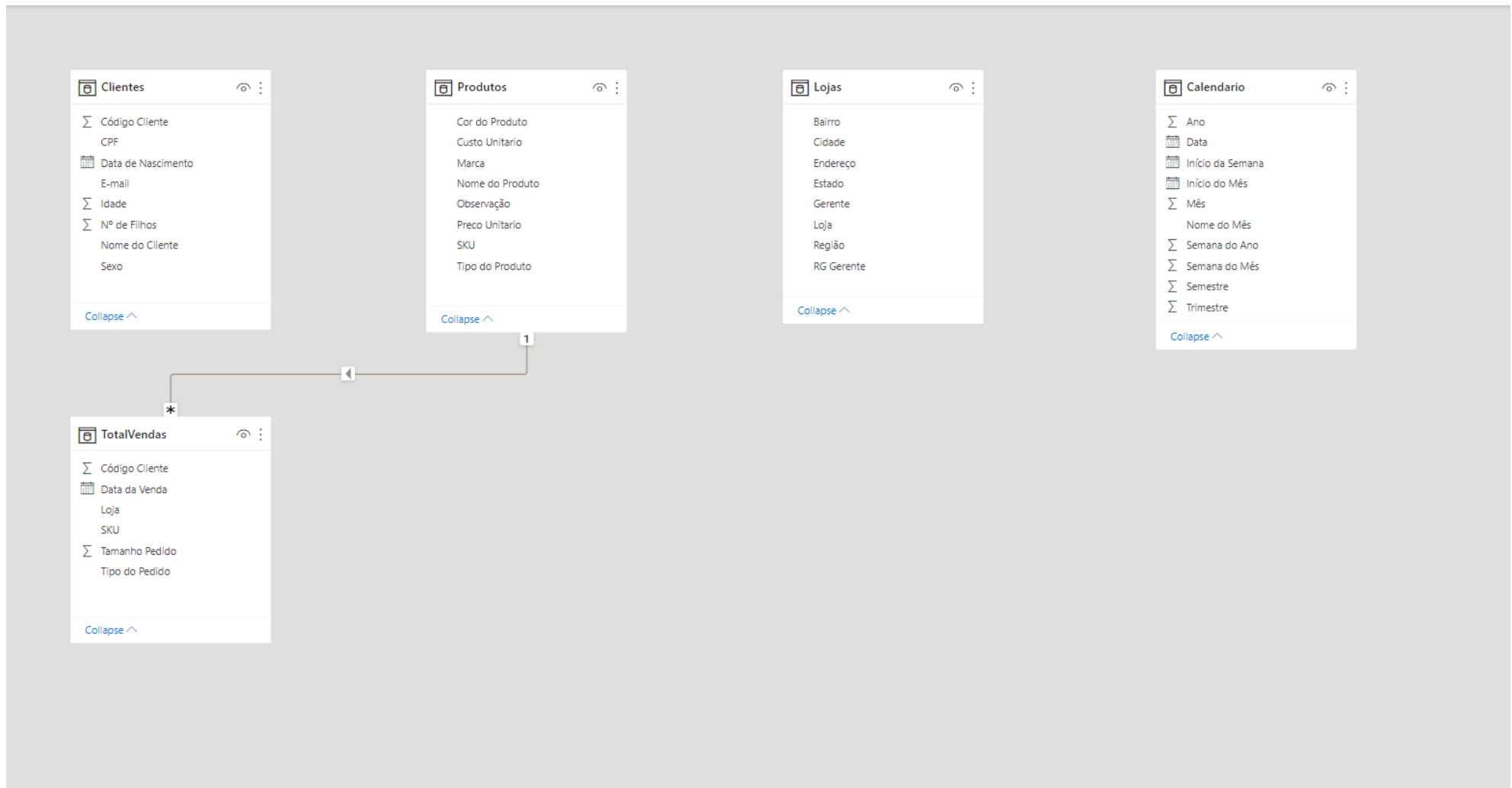
Add drill-through fields here

Com o relacionamento criado, ele calculou corretamente para nós quantos produtos foram vendidos por nome do produto!!

Observações:

- o nome das colunas é indiferente para nós, tanto faz se as colunas tiverem o mesmo nome ou nomes diferentes. O importante é que elas tenham as mesmas informações, e que essas informações sejam comuns entre si;
- quando tentarmos fazer uma análise e o resultado for diferente do esperado, como todas as linhas contendo o mesmo valor, significa que está faltando um relacionamento entre as tabelas.

Agora que já sabemos como criar relacionamentos, vamos relacionar a nossa tabela fato - TotalVendas - com as tabelas características. Lembrando de manter as tabelas características na parte de cima e as tabelas fato na parte de baixo, vamos criar todos os relacionamentos:



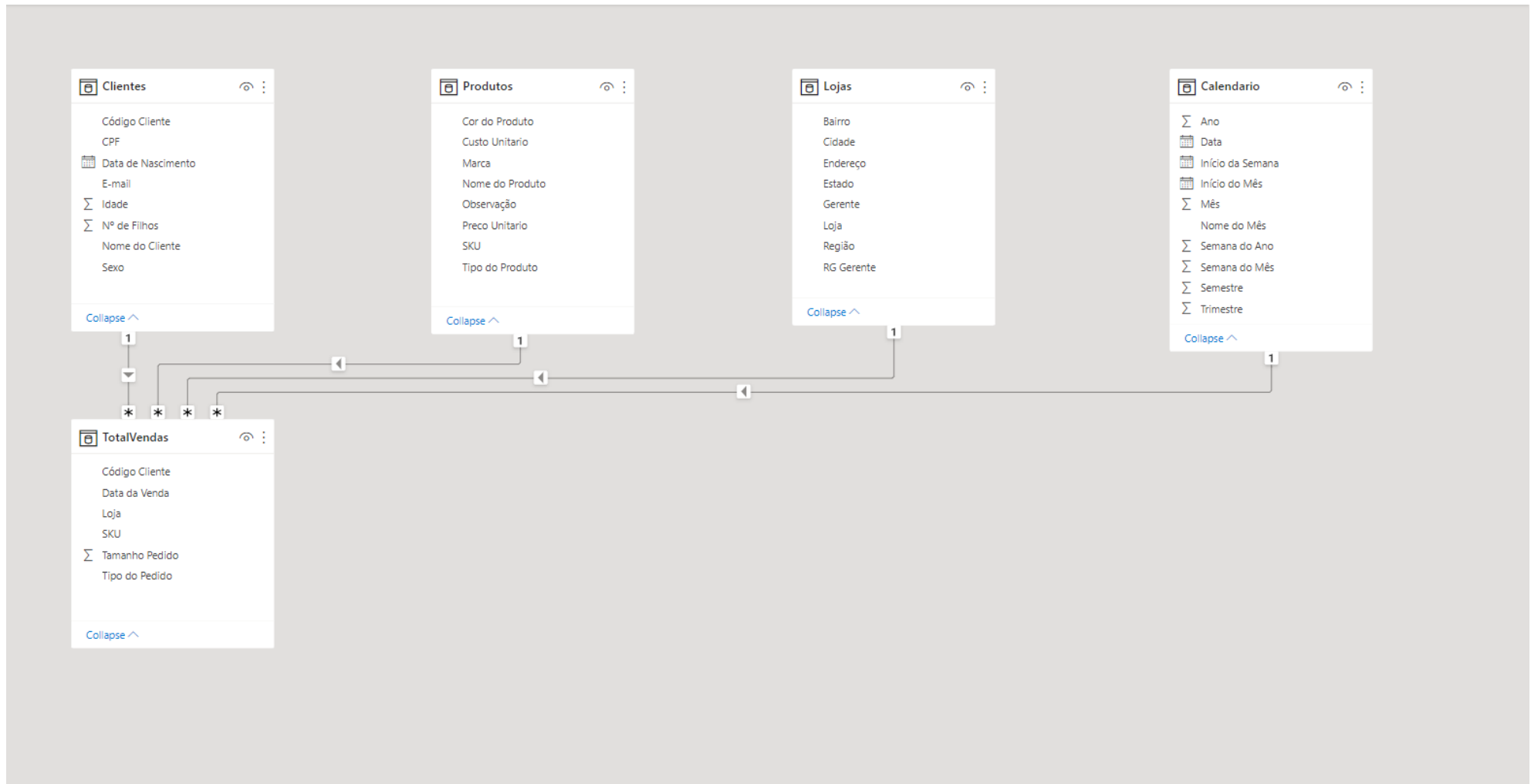
Resposta

As relações a serem criadas são:

- **Tabelas Clientes e TotalVendas:** colunas Código Cliente;
- **Tabelas Lojas e TotalVendas:** colunas Loja;

- **Tabelas Calendario e TotalVendas:** colunas Data.

Todas as relações criadas são 1:M.



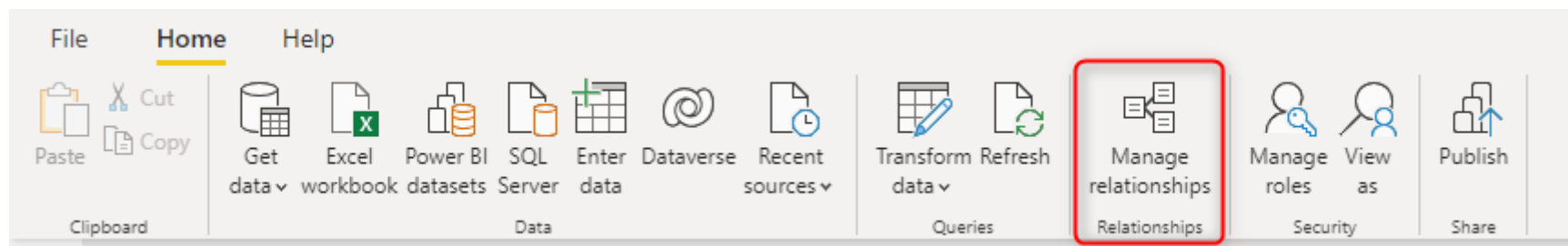
2.2 Gerenciando e Editando Relacionamentos

Toda vez que criamos alguma coisa, naturalmente pensamos em como editar o que foi criado. Então, como faríamos para editar esses relacionamentos que acabamos de criar?

Vamos imaginar que agora queremos ver apenas as vendas de um único ano, e não de todos os anos como estávamos fazendo. Para isso, vamos usar a base de vendas de 2020.

Tempos 2 maneiras possíveis de editar os relacionamentos já criados:

- **Maneira 1:** na guia **Modelo** (*Model*), na aba **Página Inicial** (*Home*) e clicando na opção **Gerenciar Relações** (*Manage Relationships*):



Isso vai nos levar àquela janela, onde podemos escolher a relação que queremos editar:

Manage relationships



Selecionamos uma relação

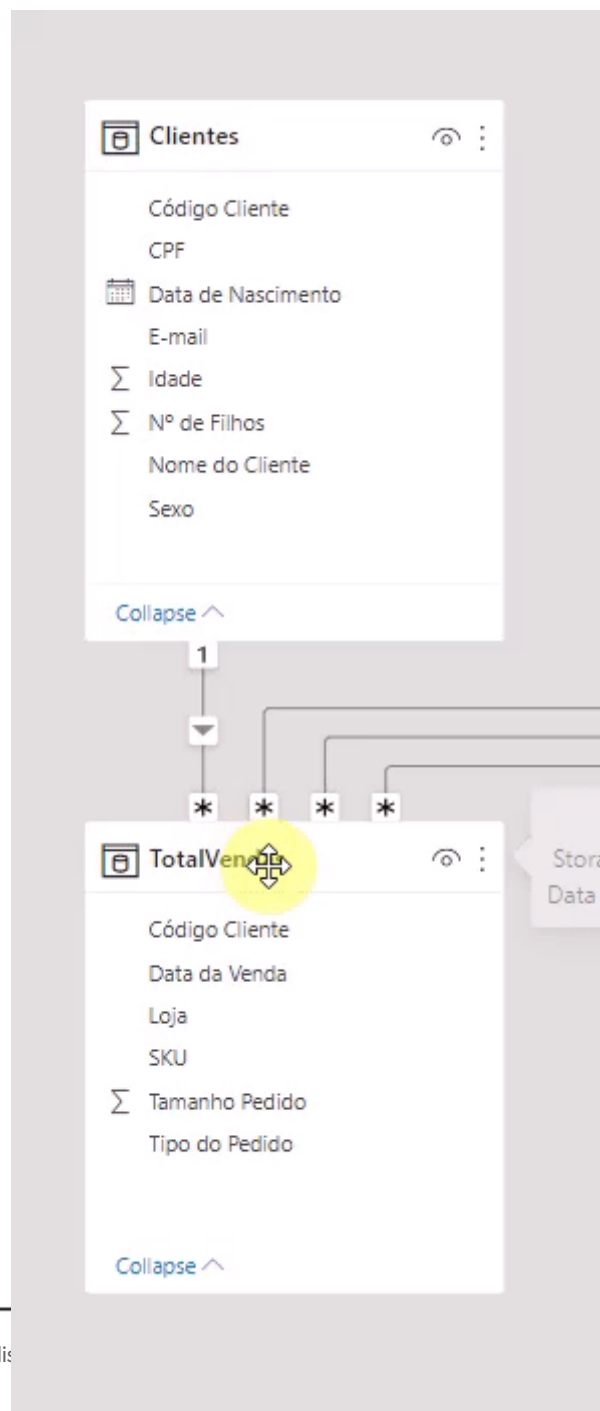
Active	From: Table (Column)	To: Table (Column)
<input checked="" type="checkbox"/>	TotalVendas (Código Cliente)	Clientes (Código Cliente)
<input checked="" type="checkbox"/>	TotalVendas (Data da Venda)	Calendario (Data)
<input checked="" type="checkbox"/>	TotalVendas (Loja)	Lojas (Loja)
<input checked="" type="checkbox"/>	TotalVendas (SKU)	Produtos (SKU)

New... Autodetect... **Edit...** Delete

E selecionamos a opção de Editar

Close

- **Maneira 2:** ou então damos um duplo clique diretamente na relação que queremos mudar:



Seguindo um dos passos, a tela de edição abre novamente para nós:



Edit relationship

Select tables and columns that are related.

TotalVendas

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido
TE10009	3	Cuiabá	quarta-feira, 1 de janeiro de 2020	588	Pedido Múltiplo
TE10005	3	Cuiabá	sábado, 4 de janeiro de 2020	48	Pedido Múltiplo
TE10015	3	Cuiabá	terça-feira, 7 de janeiro de 2020	340	Pedido Múltiplo

Clientes

Código Cliente	Sexo	Nº de Filhos	Data de Nascimento	E-mail	CPF	Nome
2	Masculino	0	segunda-feira, 9 de abril de 1951	roberto46@gmail.com	33264169658	Rober
5	Masculino	0	sábado, 19 de junho de 1982	breno49@icloud.com	95867983463	Breno
6	Masculino	0	quarta-feira, 10 de maio de 2000	lucas23@hotmail.com	41957227222	Lucas

Cardinality

Cross filter direction

Many to one (*:1)

Single

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Se eu quisesse alterar essa relação, e ao invés de relacionar a tabela com o total de vendas e a tabela clientes eu relacionasse as vendas de 2020 com a tabela clientes, basta alterar a tabela selecionada:



Edit relationship

Select tables and columns that are related.

Vendas2020

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente
TE10011	4	Cuiabá	quarta-feira, 1 de janeiro de 2020	458
TE10023	4	Cuiabá	quarta-feira, 1 de janeiro de 2020	398
TE10010	4	Cuiabá	quinta-feira, 2 de janeiro de 2020	246

Clientes

Código Cliente	Sexo	Nº de Filhos	Data de Nascimento	E-mail	CPF	Nome
2	Masculino	0	segunda-feira, 9 de abril de 1951	roberto46@gmail.com	33264169658	Rober
5	Masculino	0	sábado, 19 de junho de 1982	breno49@icloud.com	95867983463	Breno
6	Masculino	0	quarta-feira, 10 de maio de 2000	lucas23@hotmail.com	41957227222	Lucas

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

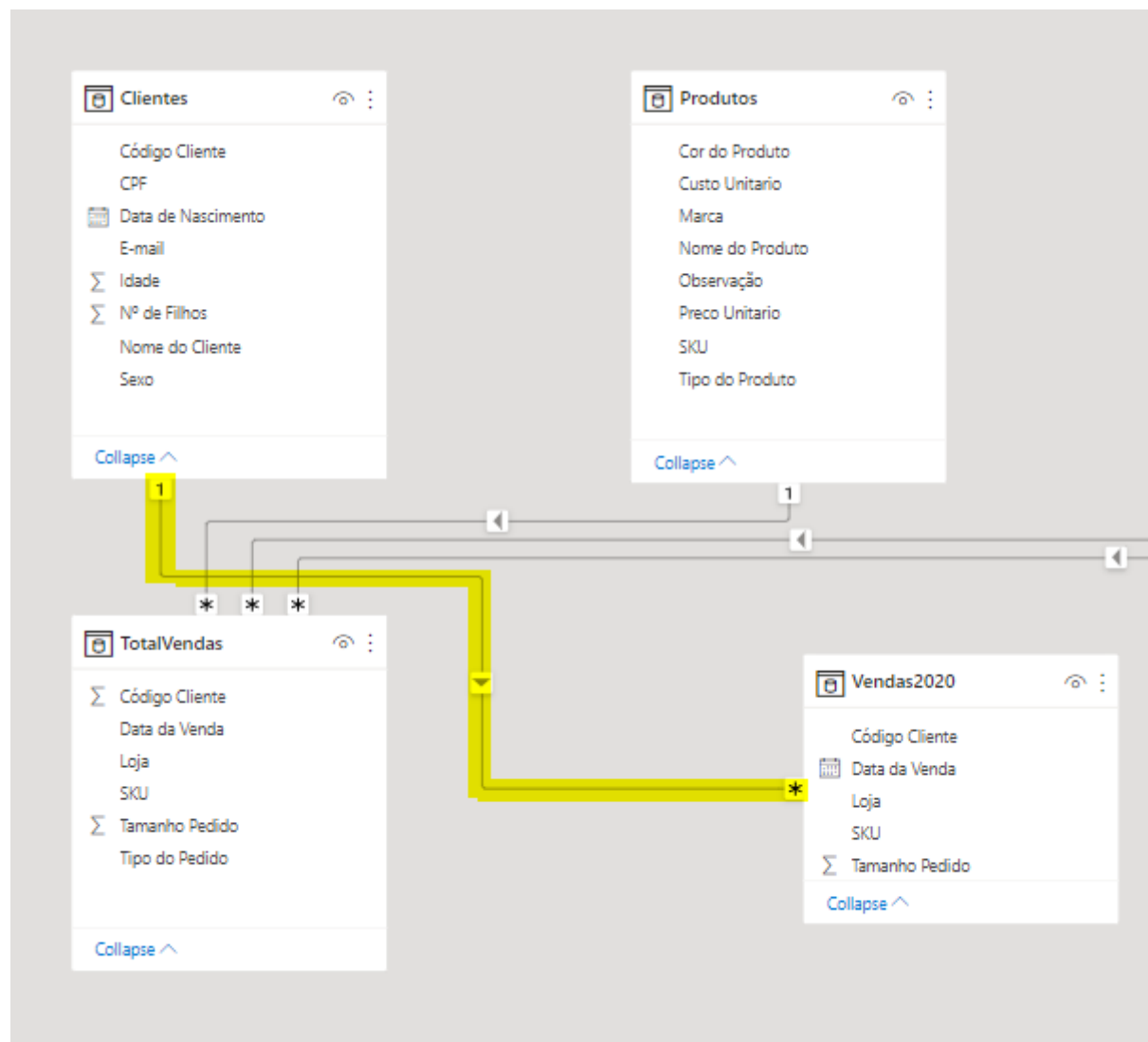
☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Dando um OK, podemos ver a relação alterada:



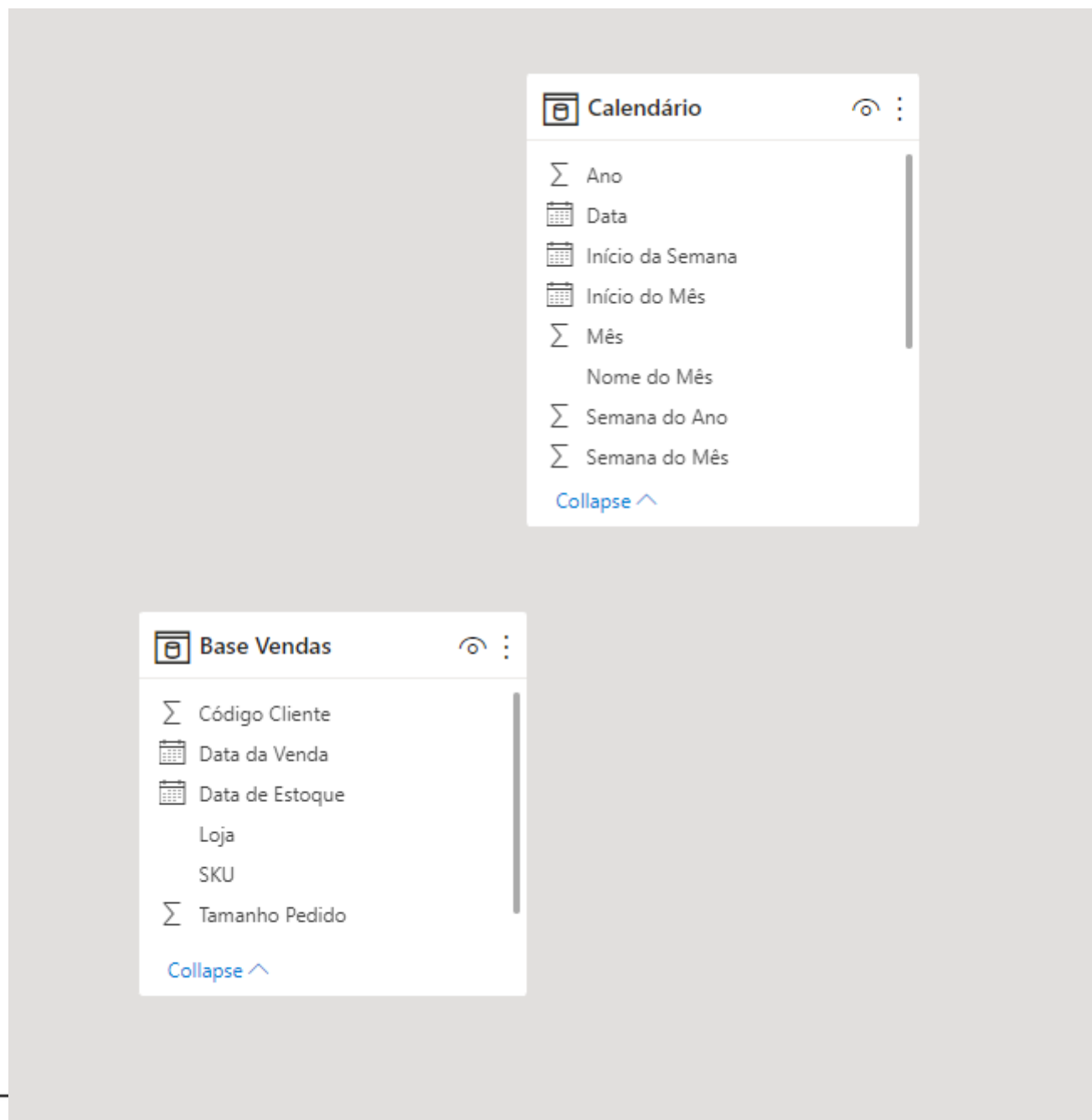
Dessa maneira não precisamos ficar excluindo relacionamentos quando quisermos alterá-los, basta apenas editar suas relações. Como essa última alteração foi apenas para demonstrar, voltamos ao que estava antes, com as tabelas TotalVendas e Clientes relacionadas!!

3 3.3 - Relações Ativas e Inativas, Tabelas Fato e Direção do Filtro

Vamos entender agora como trabalhar com mais de uma tabela fato no nosso projeto, além de ver duas configurações dos relacionamentos que ainda não estudamos: a **diferença entre relações ativas e relações inativas** e a **direção do filtro**.

3.1 Relações Ativas e Inativas

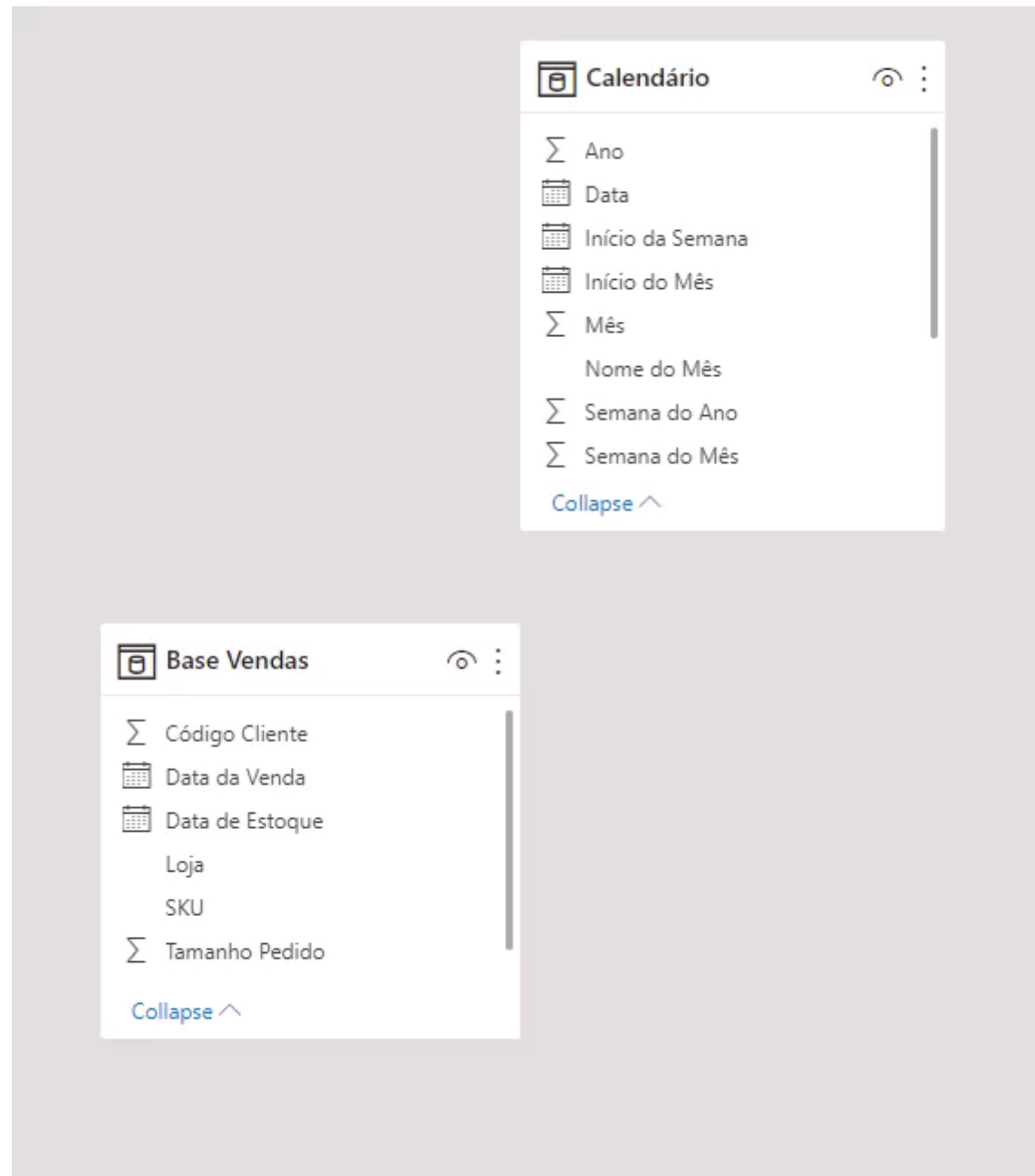
Para conseguirmos entender essas relações na prática, vamos usar um arquivo exemplo para a explicação:



Nesse exemplo, a tabela de vendas tem uma outra coluna que se refere à quando o produto entrou no estoque:

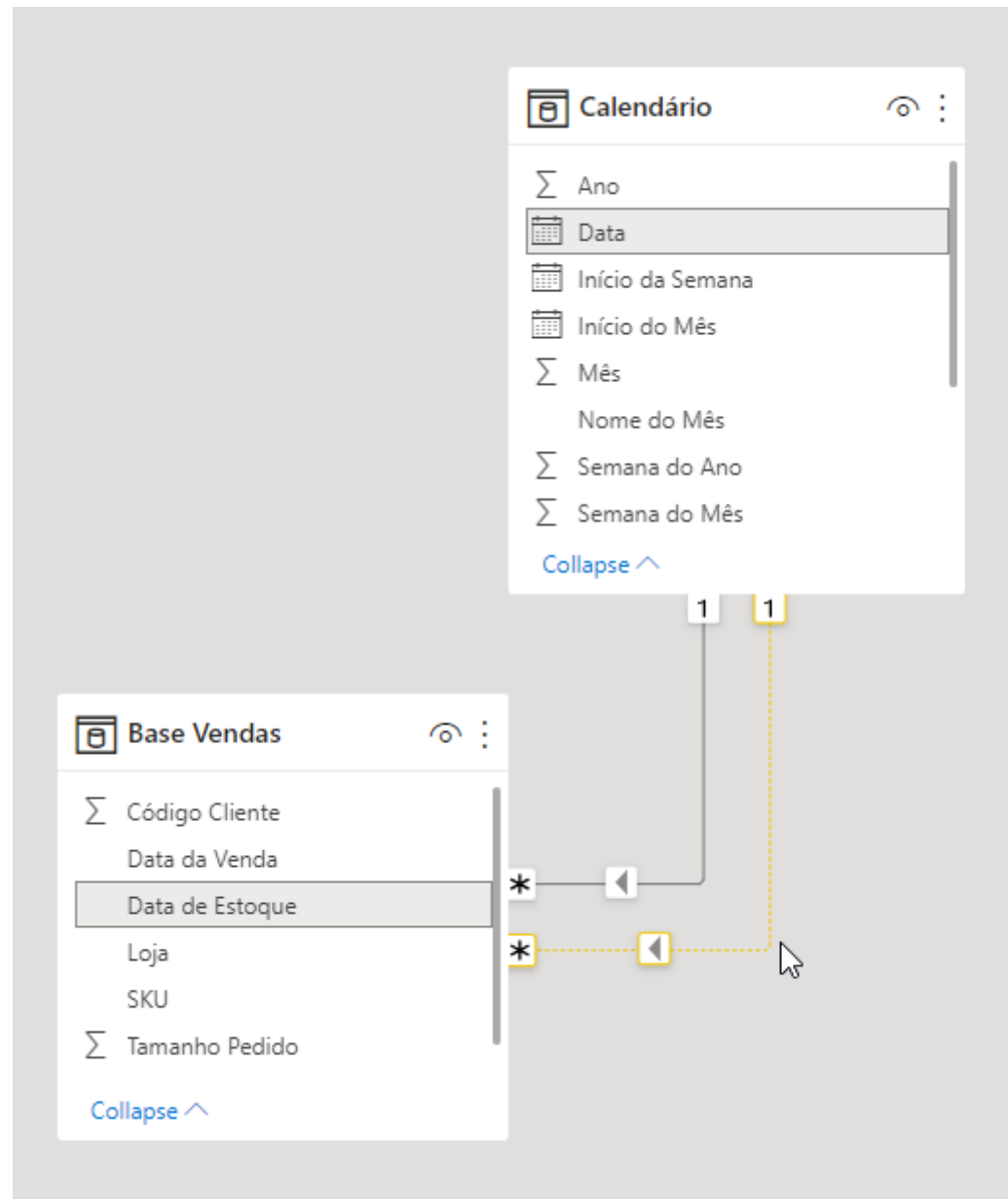
SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido	Data de Estoque
HL1024	2	Guarulhos	segunda-feira, 1 de janeiro de 2018	318	Pedido Múltiplo	domingo, 5 de novembro de 2017
HL1014	2	Guarulhos	quarta-feira, 3 de janeiro de 2018	187	Pedido Múltiplo	quinta-feira, 14 de dezembro de 2017
HL1006	2	Guarulhos	sexta-feira, 5 de janeiro de 2018	403	Pedido Múltiplo	quarta-feira, 6 de dezembro de 2017
HL1023	2	Guarulhos	sexta-feira, 5 de janeiro de 2018	334	Pedido Múltiplo	segunda-feira, 18 de dezembro de 2017
HL1004	2	Guarulhos	domingo, 7 de janeiro de 2018	75	Pedido Múltiplo	segunda-feira, 11 de dezembro de 2017
HL1011	2	Guarulhos	terça-feira, 9 de janeiro de 2018	137	Pedido Múltiplo	domingo, 26 de novembro de 2017
HL1024	2	Guarulhos	terça-feira, 9 de janeiro de 2018	630	Pedido Múltiplo	terça-feira, 21 de novembro de 2017
HL1015	2	Guarulhos	terça-feira, 9 de janeiro de 2018	479	Pedido Múltiplo	terça-feira, 12 de dezembro de 2017
HL1005	2	Guarulhos	quarta-feira, 10 de janeiro de 2018	327	Pedido Múltiplo	sábado, 18 de novembro de 2017
HL1010	2	Guarulhos	sábado, 13 de janeiro de 2018	600	Pedido Múltiplo	quinta-feira, 23 de novembro de 2017
HL1019	2	Guarulhos	domingo, 14 de janeiro de 2018	465	Pedido Múltiplo	sexta-feira, 17 de novembro de 2017
HL1015	2	Guarulhos	domingo, 14 de janeiro de 2018	56	Pedido Múltiplo	terça-feira, 5 de dezembro de 2017
HL1012	2	Guarulhos	terça-feira, 16 de janeiro de 2018	152	Pedido Múltiplo	domingo, 24 de dezembro de 2017
HL1015	2	Guarulhos	terça-feira, 16 de janeiro de 2018	302	Pedido Múltiplo	segunda-feira, 27 de novembro de 2017
HL1008	2	Guarulhos	quarta-feira, 17 de janeiro de 2018	47	Pedido Múltiplo	domingo, 24 de dezembro de 2017
HL1003	2	Guarulhos	quarta-feira, 17 de janeiro de 2018	185	Pedido Múltiplo	sábado, 18 de novembro de 2017
HL1003	2	Guarulhos	quinta-feira, 18 de janeiro de 2018	74	Pedido Múltiplo	sábado, 16 de dezembro de 2017
HL1013	2	Guarulhos	domingo, 21 de janeiro de 2018	98	Pedido Múltiplo	segunda-feira, 1 de janeiro de 2018
HL1002	2	Guarulhos	domingo, 21 de janeiro de 2018	417	Pedido Múltiplo	sábado, 2 de dezembro de 2017
HL1009	2	Guarulhos	segunda-feira, 22 de janeiro de 2018	41	Pedido Múltiplo	domingo, 31 de dezembro de 2017

Primeiramente, vamos analisar a quantidade de vendas por mês da nossa empresa. Então, relacionamos as tabelas e criamos uma matriz para ver esses valores:



Nome do Mês	Tamanho Pedido
abril	2565
agosto	1842
dezembro	3526
fevereiro	1596
janeiro	3340
julho	3043
junho	1545
maio	2466
março	2009
novembro	3536
outubro	2114
setembro	2420
Total	30002

Feito isso, vamos imaginar que agora queremos ver quantos produtos entraram no estoque por mês. Para isso criamos outro relacionamento com a data do estoque.



Aqui vemos uma diferença entre os dois relacionamentos: o primeiro possui uma **linha contínua**, enquanto o segundo uma **linha tracejada**. Mas o que isso quer dizer?

No Power BI não é possível criar dois relacionamentos entre as mesmas tabelas simultaneamente. Quando existem duas relações, apenas uma pode estar ativa, representada pela linha contínua, e as outras permanecem inativas - representadas pela linha tracejada. A relação que permaneceu ativa no nosso exemplo foi a relação criada primeiro.

Nas opções de edição do relacionamento conseguimos ativar e desativar as relações. Dando um duplo clique no relacionamento, a opção se encontra embaixo na esquerda:



Edit relationship

Select tables and columns that are related.

Base Vendas

	Loja	Data da Venda	Código Cliente	Tipo do Pedido	Data de Estoque
2	Guarulhos	segunda-feira, 1 de janeiro de 2018	318	Pedido Múltiplo	domingo, 5 de novembro de 2017
2	Guarulhos	quarta-feira, 3 de janeiro de 2018	187	Pedido Múltiplo	quinta-feira, 14 de dezembro de 2017
2	Guarulhos	sexta-feira, 5 de janeiro de 2018	403	Pedido Múltiplo	quarta-feira, 6 de dezembro de 2017

Calendário

Data	Ano	Mês	Nome do Mês	Início do Mês	Trimestre	Semana
domingo, 3 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
segunda-feira, 4 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
terça-feira, 5 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	

Cardinality

Many to one (*:1)

Cross filter direction

Single

☐ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Ativar esse relacionamento

Essa é a nossa segunda relação. Como a primeira está ativa, ele não nos permite ativar a segunda:



Edit relationship

Select tables and columns that are related.

Base Vendas

	Loja	Data da Venda	Código Cliente	Tipo do Pedido	Data de Estoque
2	Guarulhos	segunda-feira, 1 de janeiro de 2018	318	Pedido Múltiplo	domingo, 5 de novembro de 2017
2	Guarulhos	quarta-feira, 3 de janeiro de 2018	187	Pedido Múltiplo	quinta-feira, 14 de dezembro de 2017
2	Guarulhos	sexta-feira, 5 de janeiro de 2018	403	Pedido Múltiplo	quarta-feira, 6 de dezembro de 2017

Calendário

Data	Ano	Mês	Nome do Mês	Início do Mês	Trimestre	Semana
domingo, 3 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
segunda-feira, 4 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
terça-feira, 5 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

! You can't create a direct active relationship between Base Vendas and Calendário because an active set of indirect relationships already exists between these tables. To make this relationship active, set the Cross filter direction to "single", delete, or deactivate any of the indirect relationships first.

OK

Cancel

Você não pode criar uma relação ativa direta entre Base Vendas e Calendário porque um conjunto ativo de relações indiretas já existe entre essas duas tabelas. Para ativar essa relação, configure a direção do filtro como "único", exclua ou desative uma das relações indiretas antes.

Então, antes de ativá-lo precisamos desativar o outro relacionamento:



Edit relationship

Select tables and columns that are related.

Base Vendas

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido	
HL1024	2	Guarulhos	segunda-feira, 1 de janeiro de 2018	318	Pedido Múltiplo	domi
HL1014	2	Guarulhos	quarta-feira, 3 de janeiro de 2018	187	Pedido Múltiplo	quinta-fe
HL1006	2	Guarulhos	sexta-feira, 5 de janeiro de 2018	403	Pedido Múltiplo	quarta-f

Calendário

Data	Ano	Mês	Nome do Mês	Início do Mês	Trimestre	Semai
domingo, 3 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
segunda-feira, 4 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	
terça-feira, 5 de janeiro de 2016	2016	1	janeiro	sexta-feira, 1 de janeiro de 2016	1	

Cardinality

Many to one (*:1)

Cross filter direction

Single

☐ Make this relationship active

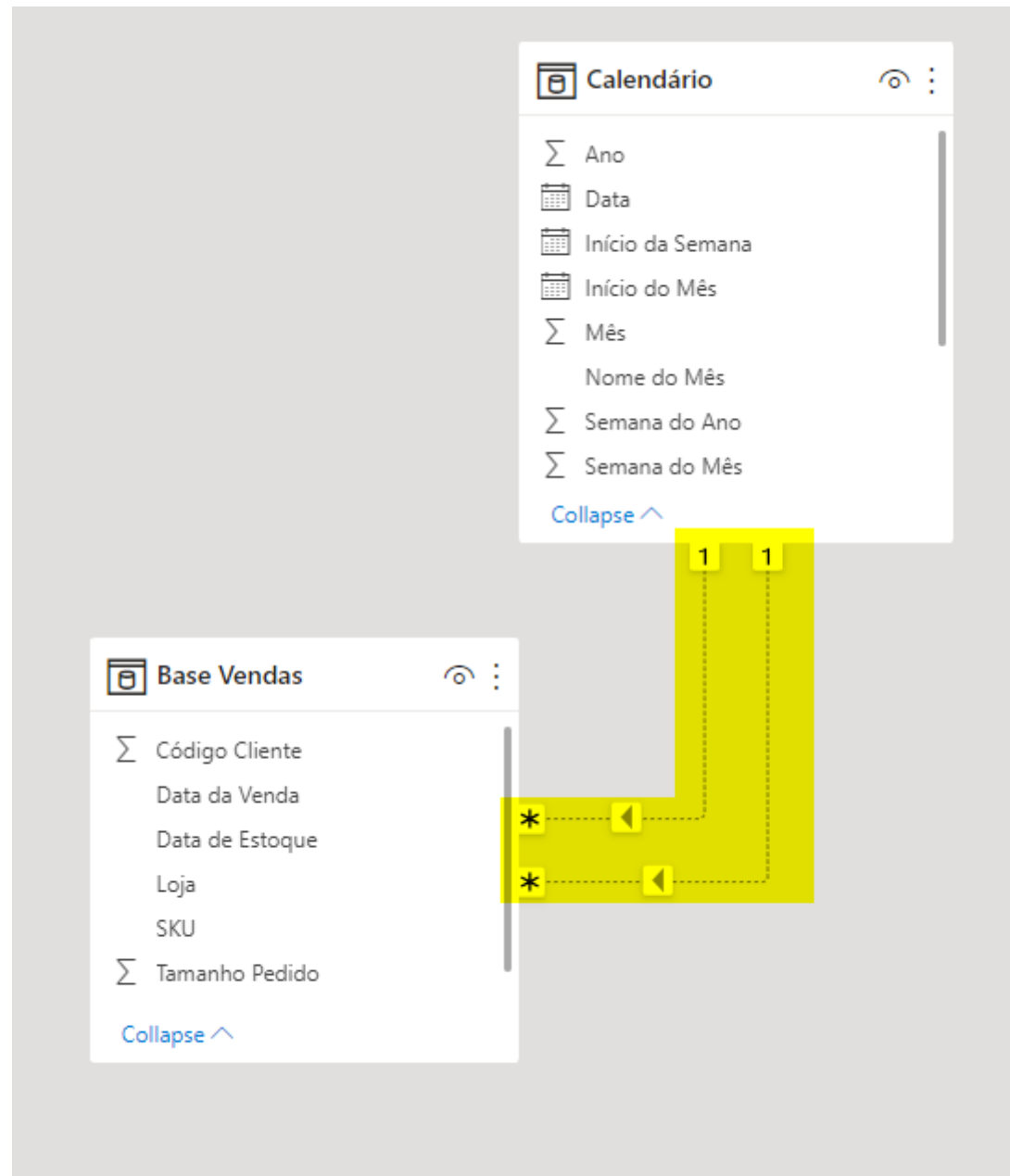
☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

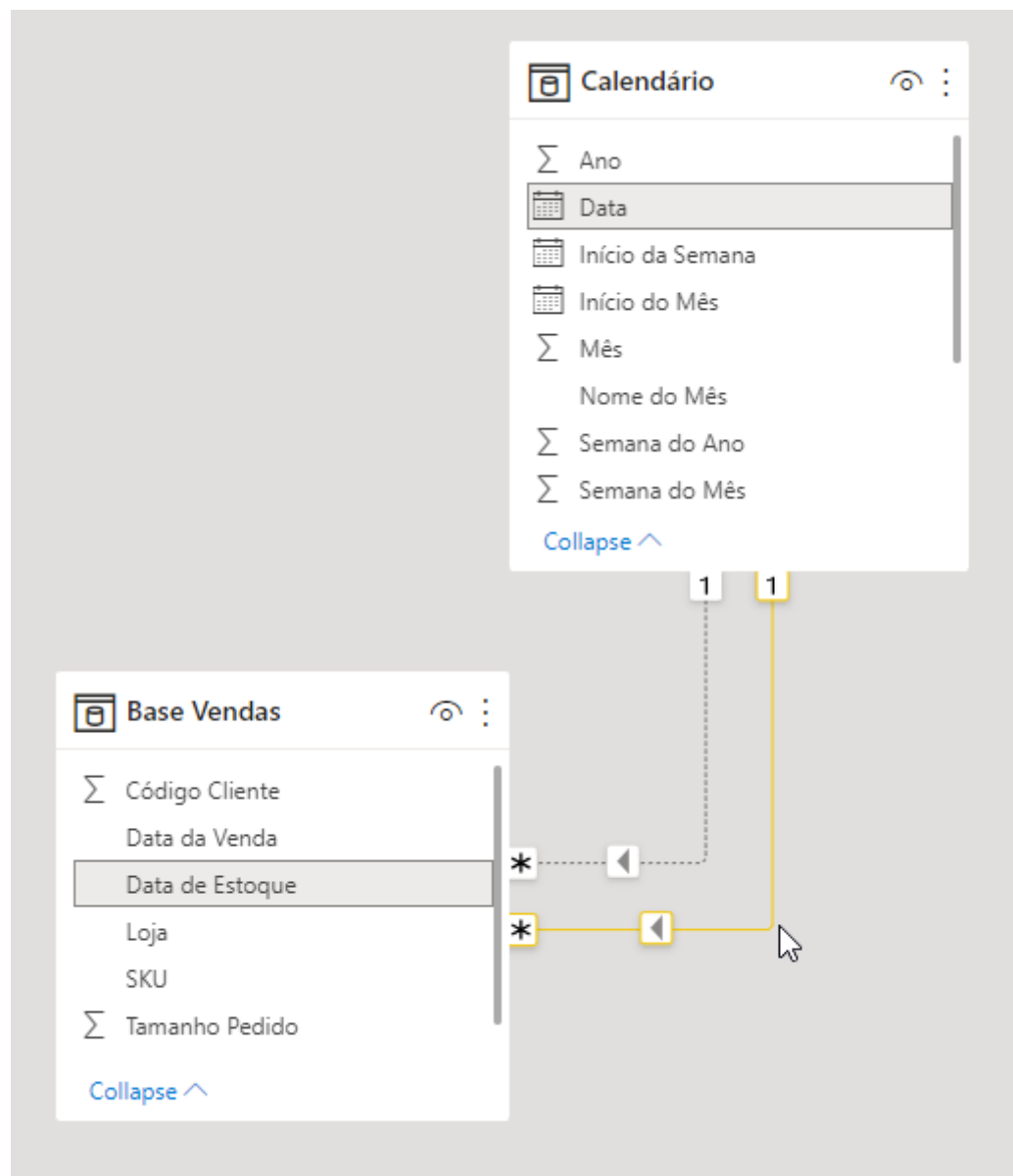
Agora as duas relações estão inativas:



Voltando ao nosso relatório:

Nome do Mês	Tamanho Pedido
abril	30002
agosto	30002
dezembro	30002
fevereiro	30002
janeiro	30002
julho	30002
junho	30002
maio	30002
março	30002
novembro	30002
outubro	30002
setembro	30002
Total	30002

Por fim ativamos a outra relação:



Podemos ver que mesmo usando os mesmos campos na nossa matriz o valor se altera com base na relação estabelecida, que agora é entre a quantidade de produtos e a data de entrada:

Nome do Mês	Tamanho Pedido
abril	2175
agosto	2309
dezembro	2915
fevereiro	1977
janeiro	1919
julho	2082
junho	2557
maio	2194
março	2520
novembro	3363
outubro	3460
setembro	2531
Total	30002

3.2 Trabalhando com várias Tabelas Fato

Não podemos ter duas relações ativas entre duas tabelas iguais simultaneamente; quando falamos em tabelas diferentes, podemos ter várias relações de uma tabela com outras tabelas sem problema algum.

Até agora estávamos trabalhando com 5 bases de dados, sendo 4 tabelas **características** e 1 tabela **fato**. Agora vamos importar uma nova tabela para o Power BI, contendo as informações sobre as Devoluções:

	SKU	Quantidade Devolvida	Loja	Data da Devolução
1	TE10022	3	Campinas	01/01/2020
2	TE10011	3	Campinas	01/01/2020
3	TE10011	1	São Paulo	01/01/2020
4	TE10001	3	Fortaleza	02/01/2020
5	TE10009	1	Campinas	03/01/2020
6	TE10024	1	Curitiba	03/01/2020
7	TE10011	3	Vila Velha	05/01/2020
8	TE10011	3	Fortaleza	05/01/2020
9	TE10015	1	Campinas	06/01/2020
10	TE10015	2	Cuiabá	06/01/2020
11	TE10015	3	Recife	06/01/2020
12	TE10003	1	Salvador	09/01/2020
13	TE10011	2	Curitiba	10/01/2020
14	TE10005	3	Salvador	11/01/2020
15	TE10022	2	Florianópolis	11/01/2020
16	TE10013	3	Fortaleza	12/01/2020
17	TE10014	3	Florianópolis	13/01/2020
18	TE10008	1	Fortaleza	14/01/2020
19	TE10014	1	Porto Alegre	15/01/2020
20	TE10010	3	São Paulo	17/01/2020

PROPERTIES

Name

Devoluções

[All Properties](#)

APPLIED STEPS

Source



Navigation



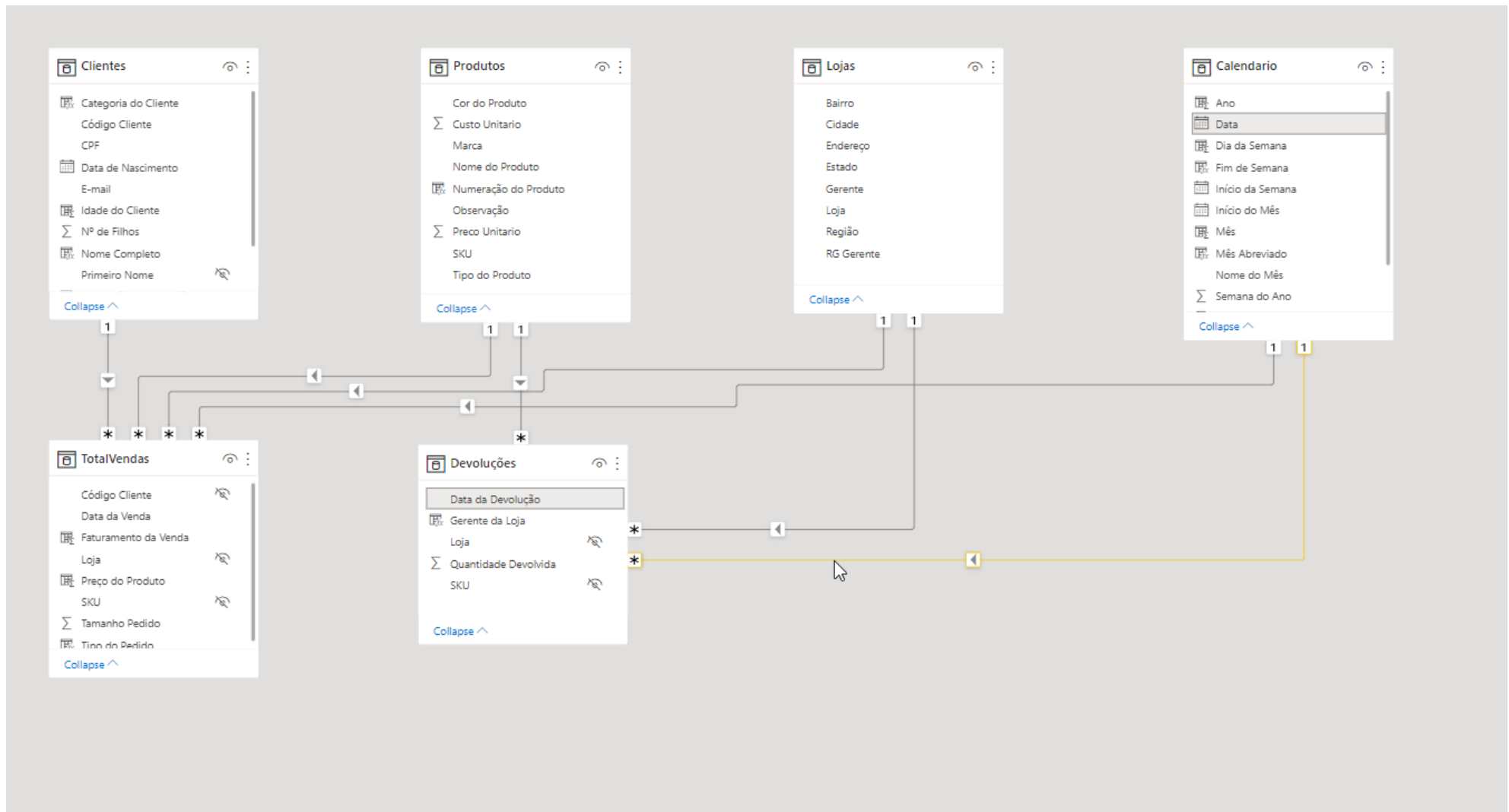
Promoted Headers



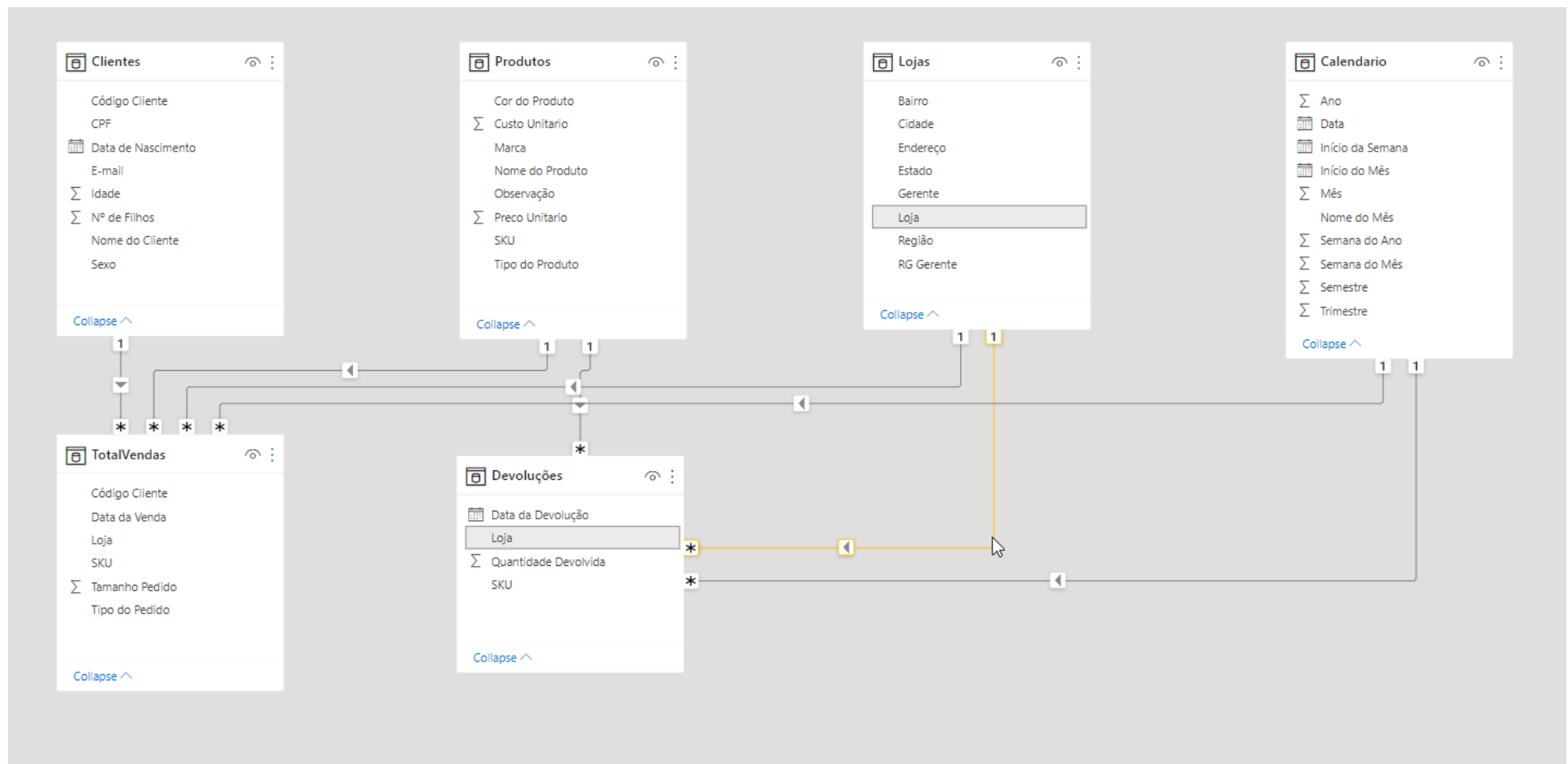
Changed Type

Ela possui três relações:

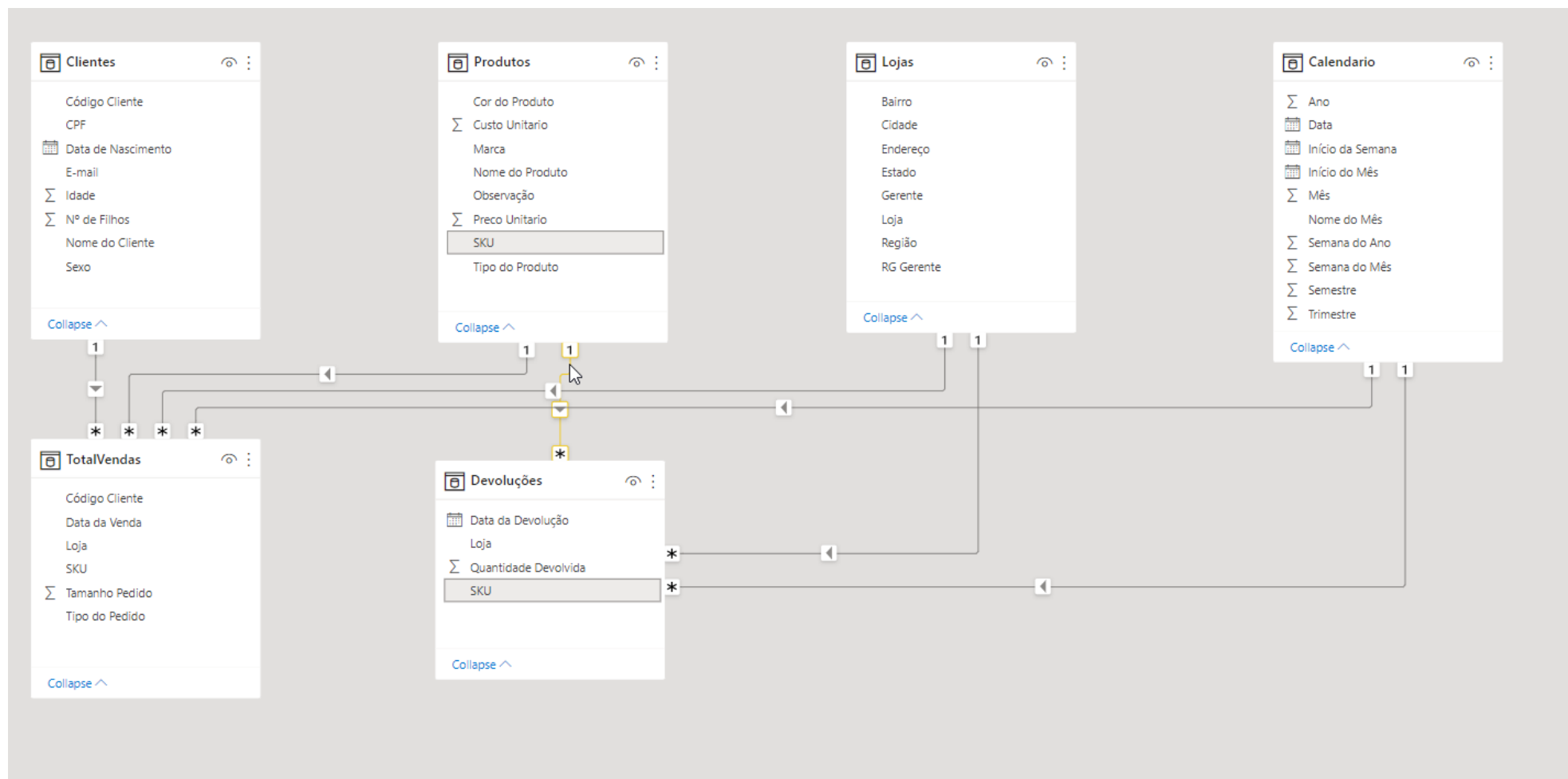
- Com a base **Calendário**:



- Com a base **Lojas**:



- Com a base **Produtos**:

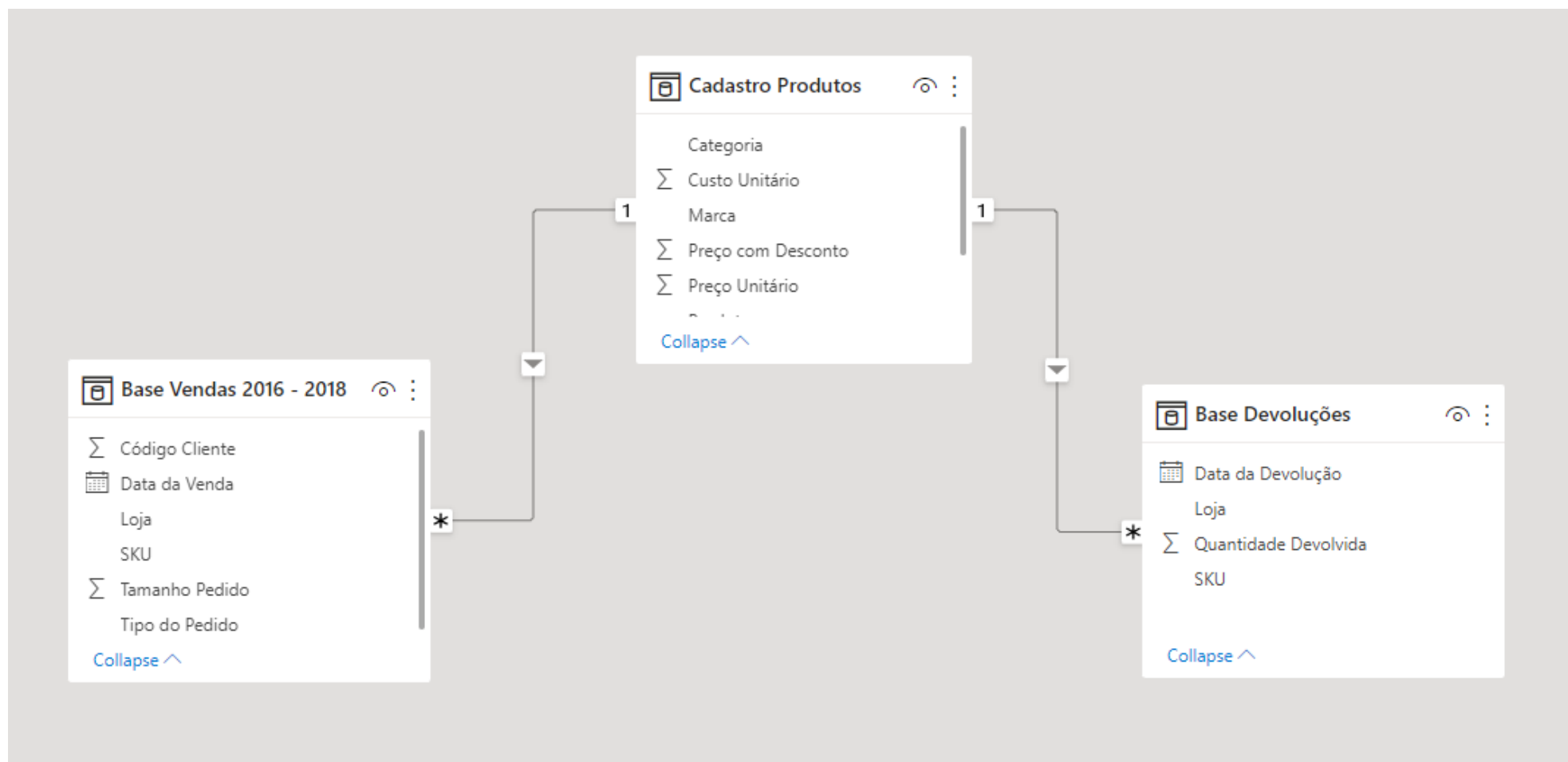


Como não temos nenhuma chave estrangeira na Base de Devoluções que tenha uma chave primária correspondente na base de Clientes, não conseguimos relacionar as duas. Isso é algo que pode acontecer na prática - nem todas as tabelas são relacionáveis, já que nem sempre vamos ter as mesmas informações em diferentes tabelas.

3.3 Direção do Filtro

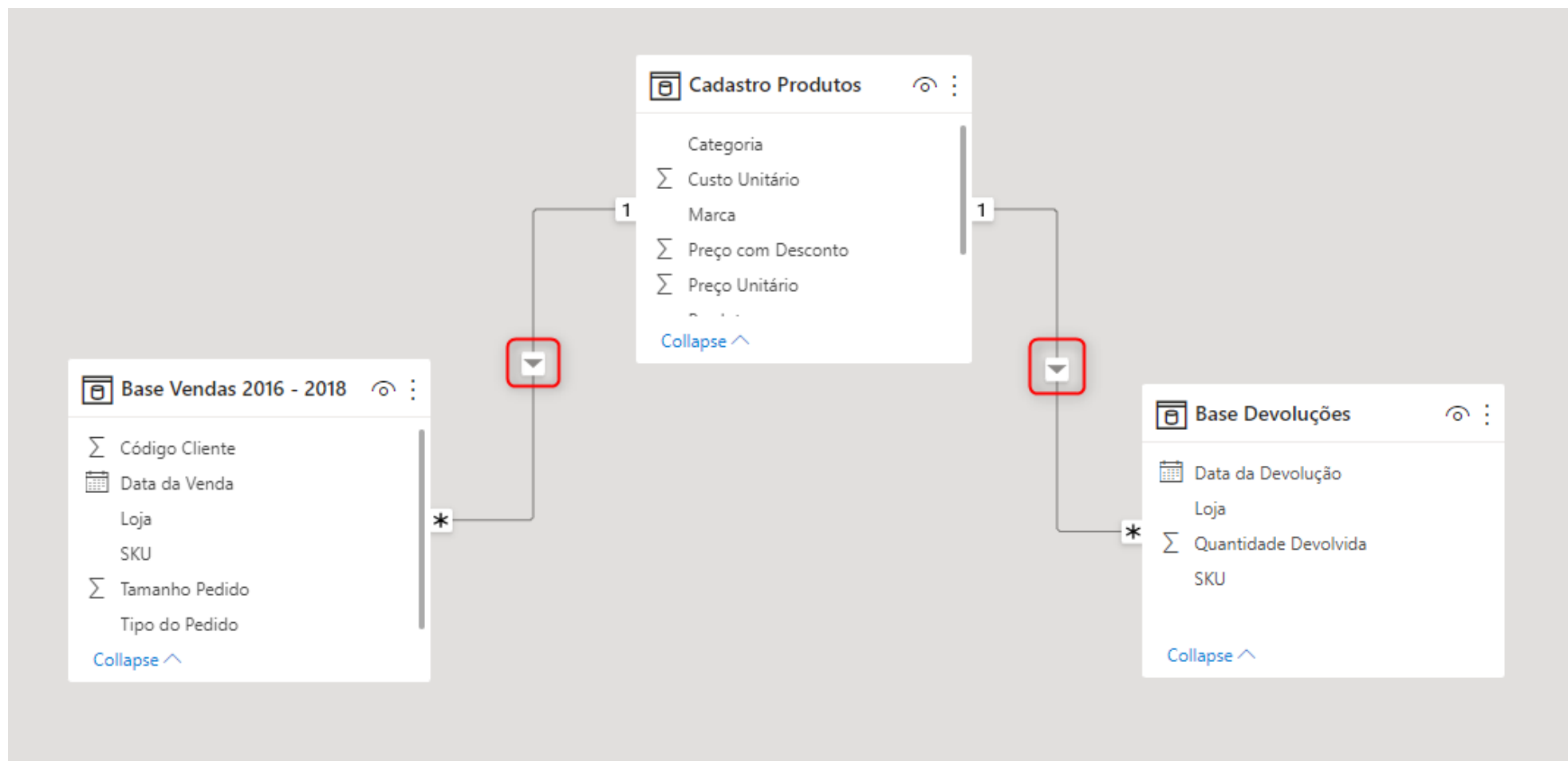
3.3.1 Entendendo a Direção do Filtro

Vamos entender agora o que significa a direção das setas que aparecem nos relacionamentos e qual a sua relevância. Para isso vamos usar um arquivo auxiliar com apenas três tabelas, para deixar a visualização mais simples:



Filtros.pbix

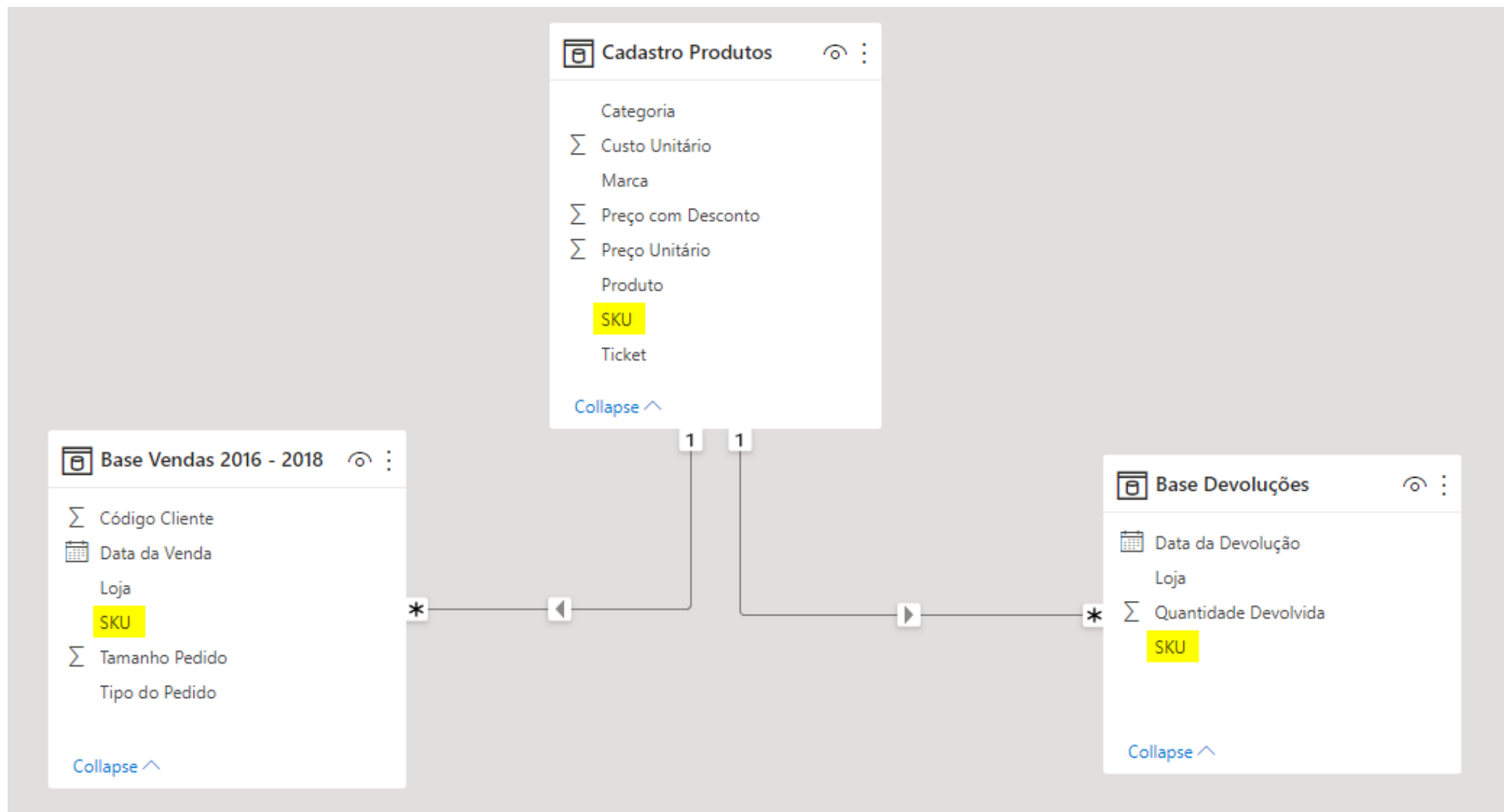
A seta aparece nas linhas que ligam as tabelas:



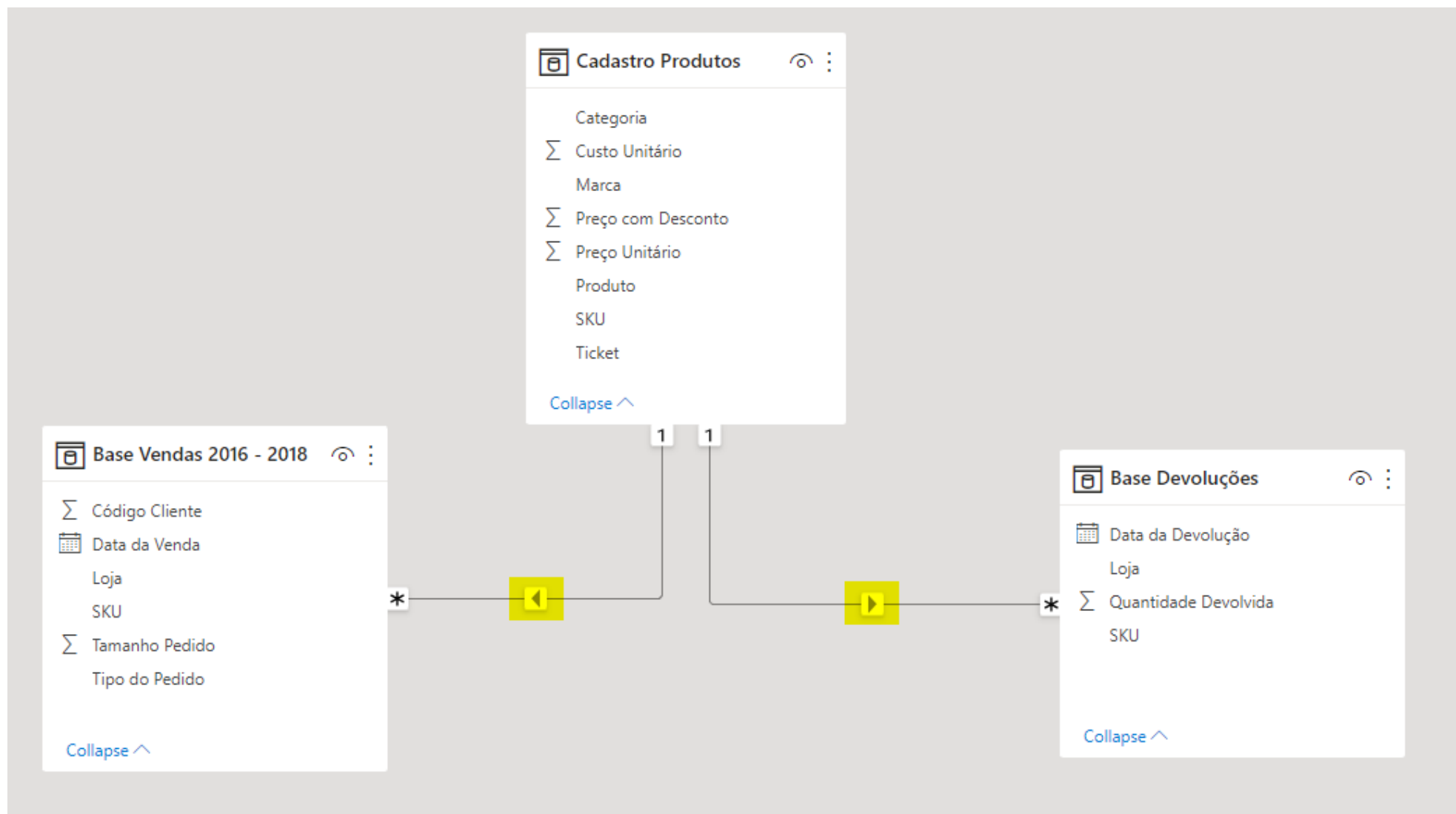
Vamos imaginar que nesse exemplo a gente queira fazer um relatório para saber a quantidade de vendas e a quantidade de devoluções dos produtos - que se trata exatamente das informações de SKU.

Olhando na base dos produtos, podemos ver que temos 24 produtos distintos. Então, o que vamos fazer é listar esses produtos em uma matriz e ver a quantidade de vendas e devoluções de cada um.

Voltando na nossa guia **Modelo (Model)**, vemos que temos as informações de SKU nas três tabelas:



Então nos perguntamos: quando formos montar a matriz, nós vamos puxar os dados de SKU da base de vendas, da base de devoluções ou da base de produtos? Para saber qual dos 3 escolher, vamos olhar exatamente para o direcionamento do filtro - ou seja, para a setinha da nossa relação:



A melhor escolha vai ser a coluna de SKU da base dos Produtos. Por quê?

Basicamente, a direção da seta mostra que podemos usar essa informação tanto na base de devoluções quando para a base de vendas.

Vamos mostrar na prática:

1: Nas linhas o SKU da base de **produtos** e nas colunas a soma dos tamanhos dos pedidos das bases de vendas e devoluções:

Compras e Devoluções por Produto		
SKU	Tamanho Pedido	Quantidade Devolvida
HL1001	3030	101
HL1002	3081	
HL1003	3001	108
HL1004	2864	96
HL1005	2955	129
HL1006	3024	126
HL1007	3048	95
HL1008	2958	101
HL1009	2852	192
HL1010	2903	121
HL1011	2969	219
HL1012	2961	83
HL1013	3143	101
HL1014	2875	148
HL1015	2961	110
HL1016	2977	120
HL1017	2821	86
HL1018	2914	211
HL1019	3061	103
HL1020	3063	89
HL1021	3028	
HL1022	3075	128
HL1023	3024	
HL1024	2854	123
Total	71442	2590

2: Nas linhas o SKU da base de **vendas** e nas colunas a soma dos tamanhos dos pedidos das bases de vendas e devoluções:

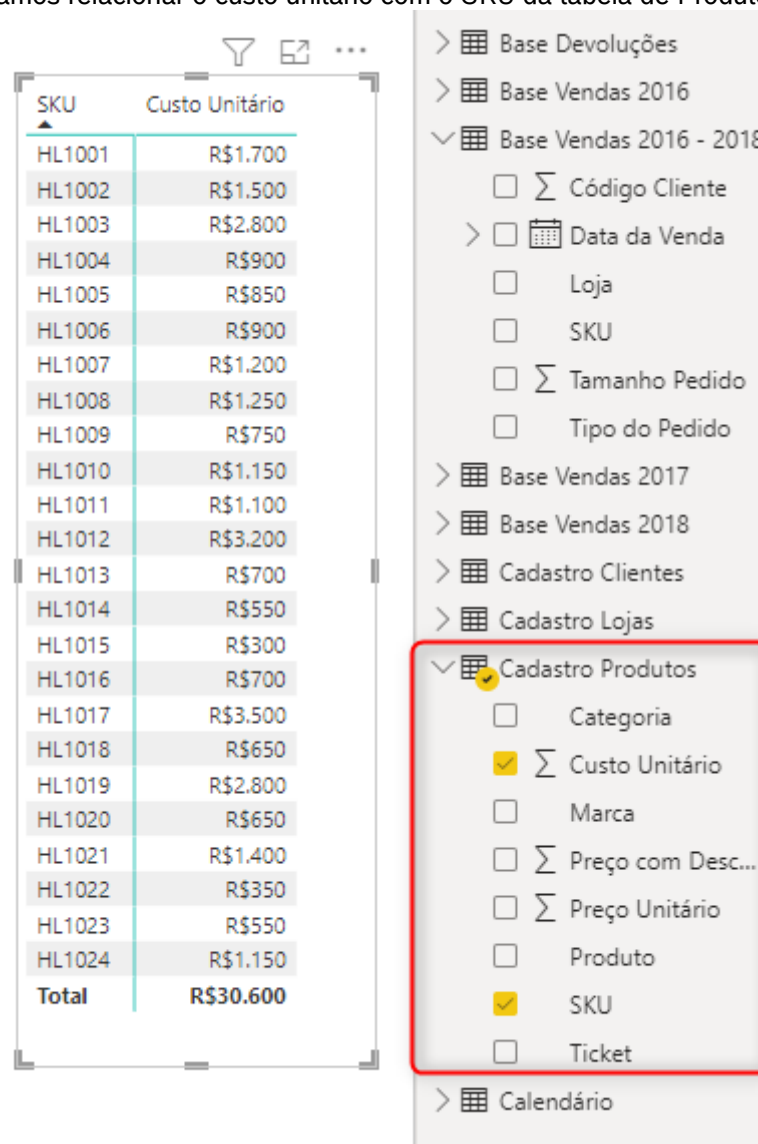
Compras e Devoluções por Produto

SKU	Tamanho Pedido	Quantidade Devolvida
HL1001	3030	2590
HL1002	3081	2590
HL1003	3001	2590
HL1004	2864	2590
HL1005	2955	2590
HL1006	3024	2590
HL1007	3048	2590
HL1008	2958	2590
HL1009	2852	2590
HL1010	2903	2590
HL1011	2969	2590
HL1012	2961	2590
HL1013	3143	2590
HL1014	2875	2590
HL1015	2961	2590
HL1016	2977	2590
HL1017	2821	2590
HL1018	2914	2590
HL1019	3061	2590
HL1020	3063	2590
HL1021	3028	2590
HL1022	3075	2590
HL1023	3024	2590
HL1024	2854	2590
Total	71442	2590

Por que funcionou quando escolhemos a coluna de SKU da base de produtos e não funcionou quando pegamos a coluna SKU da base de vendas? Voltando na imagem lá em cima, podemos ver que o filtro da coluna de SKU se direciona tanto para a tabela de vendas quanto para a tabela de devoluções; entretanto, o SKU da base de vendas não está relacionado ao SKU da base de devoluções. Isso significa que podemos usar essa relação entre a tabela característica e as duas tabelas fato, mas não podemos fazer o caminho inverso.

Um outro exemplo disso: vamos montar uma matriz com o código do produto e seu custo unitário. O processo que o Power BI faz é somar todos os valores do custo, e então dividir na matriz de acordo com seu código.

1: vamos relacionar o custo unitário com o SKU da tabela de Produtos:



SKU	Custo Unitário
HL1001	R\$1.700
HL1002	R\$1.500
HL1003	R\$2.800
HL1004	R\$900
HL1005	R\$850
HL1006	R\$900
HL1007	R\$1.200
HL1008	R\$1.250
HL1009	R\$750
HL1010	R\$1.150
HL1011	R\$1.100
HL1012	R\$3.200
HL1013	R\$700
HL1014	R\$550
HL1015	R\$300
HL1016	R\$700
HL1017	R\$3.500
HL1018	R\$650
HL1019	R\$2.800
HL1020	R\$650
HL1021	R\$1.400
HL1022	R\$350
HL1023	R\$550
HL1024	R\$1.150
Total	R\$30.600

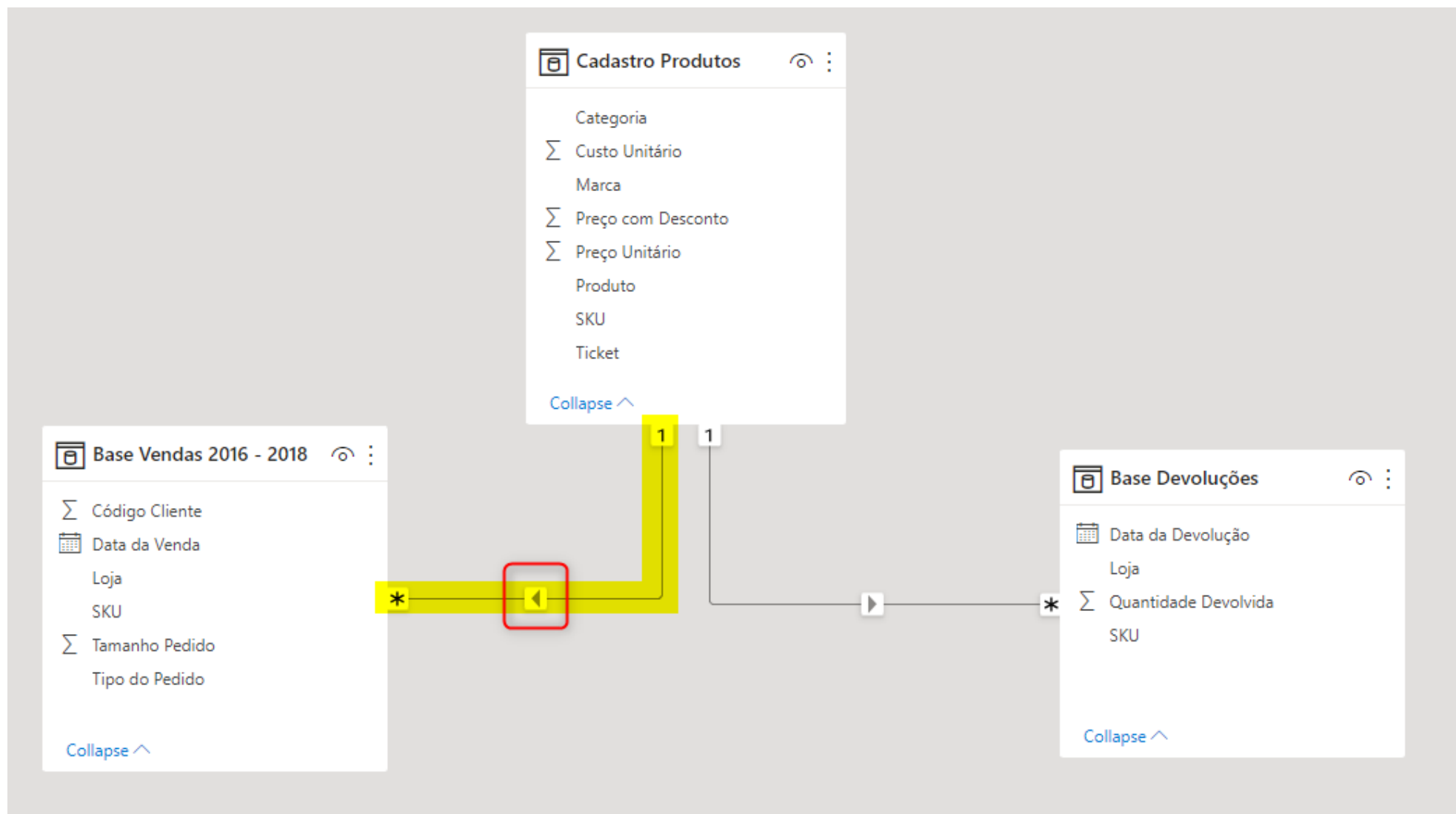
- > Base Devoluções
- > Base Vendas 2016
- ✓ Base Vendas 2016 - 2018
 - ☐ Σ Código Cliente
 - > ☐ Data da Venda
 - ☐ Loja
 - ☐ SKU
 - ☐ Σ Tamanho Pedido
 - ☐ Tipo do Pedido
- > Base Vendas 2017
- > Base Vendas 2018
- > Cadastro Clientes
- > Cadastro Lojas
- ✓ Cadastro Produtos
 - ☐ Categoria
 - ✓ Σ Custo Unitário
 - ☐ Marca
 - ☐ Σ Preço com Desc...
 - ☐ Σ Preço Unitário
 - ☐ Produto
 - ✓ SKU
 - ☐ Ticket
- > Calendário

2: agora vamos relacionar o custo unitário da tabela de produtos com o SKU da tabela das vendas:

SKU	Custo Unitário
HL1001	R\$30.600
HL1002	R\$30.600
HL1003	R\$30.600
HL1004	R\$30.600
HL1005	R\$30.600
HL1006	R\$30.600
HL1007	R\$30.600
HL1008	R\$30.600
HL1009	R\$30.600
HL1010	R\$30.600
HL1011	R\$30.600
HL1012	R\$30.600
HL1013	R\$30.600
HL1014	R\$30.600
HL1015	R\$30.600
HL1016	R\$30.600
HL1017	R\$30.600
HL1018	R\$30.600
HL1019	R\$30.600
HL1020	R\$30.600
HL1021	R\$30.600
HL1022	R\$30.600
HL1023	R\$30.600
HL1024	R\$30.600
Total	R\$30.600

- > Base Devoluções
- > Base Vendas 2016
- ✓ Base Vendas 2016 - 2018
 - ☐ Σ Código Cliente
 - > ☐ Data da Venda
 - ☐ Loja
 - ☒ SKU
 - ☐ Σ Tamanho Pedido
 - ☐ Tipo do Pedido
- > Base Vendas 2017
- > Base Vendas 2018
- > Cadastro Clientes
- > Cadastro Lojas
- ✓ Cadastro Produtos
 - ☐ Categoria
 - ☒ Σ Custo Unitário
 - ☐ Marca
 - ☐ Σ Preço com Desc...
 - ☐ Σ Preço Unitário
 - ☐ Produto
 - ☐ SKU
 - ☐ Ticket
- > Calendário

Mesmo as tabelas estando relacionadas, o filtro segue apenas uma direção - da tabela de produtos para a tabela de vendas:



Isso significa que quando usarmos a coluna do SKU da tabela de produtos como um filtro aplicado na tabela de vendas, esse filtro vai funcionar. Porém, quando fazemos o contrário ele não funciona, já que o filtro segue apenas uma direção.

3.3.2 Filtro Bidirecional

No exemplo com os números totais das vendas e devoluções dos produtos, vimos que a solução foi usar a coluna de SKU da nossa tabela características. Mas e se a gente não tivesse nenhuma alternativa além de optar pela coluna de SKU que está na base de vendas?

Nesse caso, teremos que usar o **Filtro Bidirecional**.

Para isso, vamos na tela de edição do relacionamento, onde acharemos a opção **Direção do Filtro Cruzado** (*Cross Filter Direction*):



Edit relationship

Select tables and columns that are related.

Base Vendas 2016 - 2018

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido
HL1009	3	Guarulhos	01/01/2016	334	Pedido Múltiplo
HL1005	3	Guarulhos	04/01/2016	184	Pedido Múltiplo
HL1015	3	Guarulhos	07/01/2016	184	Pedido Múltiplo

Cadastro Produtos

SKU	Produto	Marca	Categoria	Preço Unitário	Custo Unitário	Preço com Desconto	Ticket
HL1001	Smart TV 50' 4K	LG	Televisão	R\$2.600	R\$1.700	R\$2.210	Ticket Méd
HL1002	iPhone 7	Apple	Celular	R\$2.500	R\$1.500	R\$2.125	Ticket Baix
HL1003	Galaxy S10	Samsung	Celular	R\$4.500	R\$2.800	R\$3.825	Ticket Méd

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Aqui basta trocar de **Único** (*Single*) para **Ambas** (*Both*):



Edit relationship

Select tables and columns that are related.

Base Vendas 2016 - 2018

SKU	Tamanho Pedido	Loja	Data da Venda	Código Cliente	Tipo do Pedido
HL1009	3	Guarulhos	01/01/2016	334	Pedido Múltiplo
HL1005	3	Guarulhos	04/01/2016	184	Pedido Múltiplo
HL1015	3	Guarulhos	07/01/2016	184	Pedido Múltiplo

Cadastro Produtos

SKU	Produto	Marca	Categoria	Preço Unitário	Custo Unitário	Preço com Desconto	Ticket
HL1001	Smart TV 50' 4K	LG	Televisão	R\$2.600	R\$1.700	R\$2.210	Ticket Méd
HL1002	iPhone 7	Apple	Celular	R\$2.500	R\$1.500	R\$2.125	Ticket Baix
HL1003	Galaxy S10	Samsung	Celular	R\$4.500	R\$2.800	R\$3.825	Ticket Méd

Cardinality

Cross filter direction

Many to one (*:1)

Both

☒ Make this relationship active

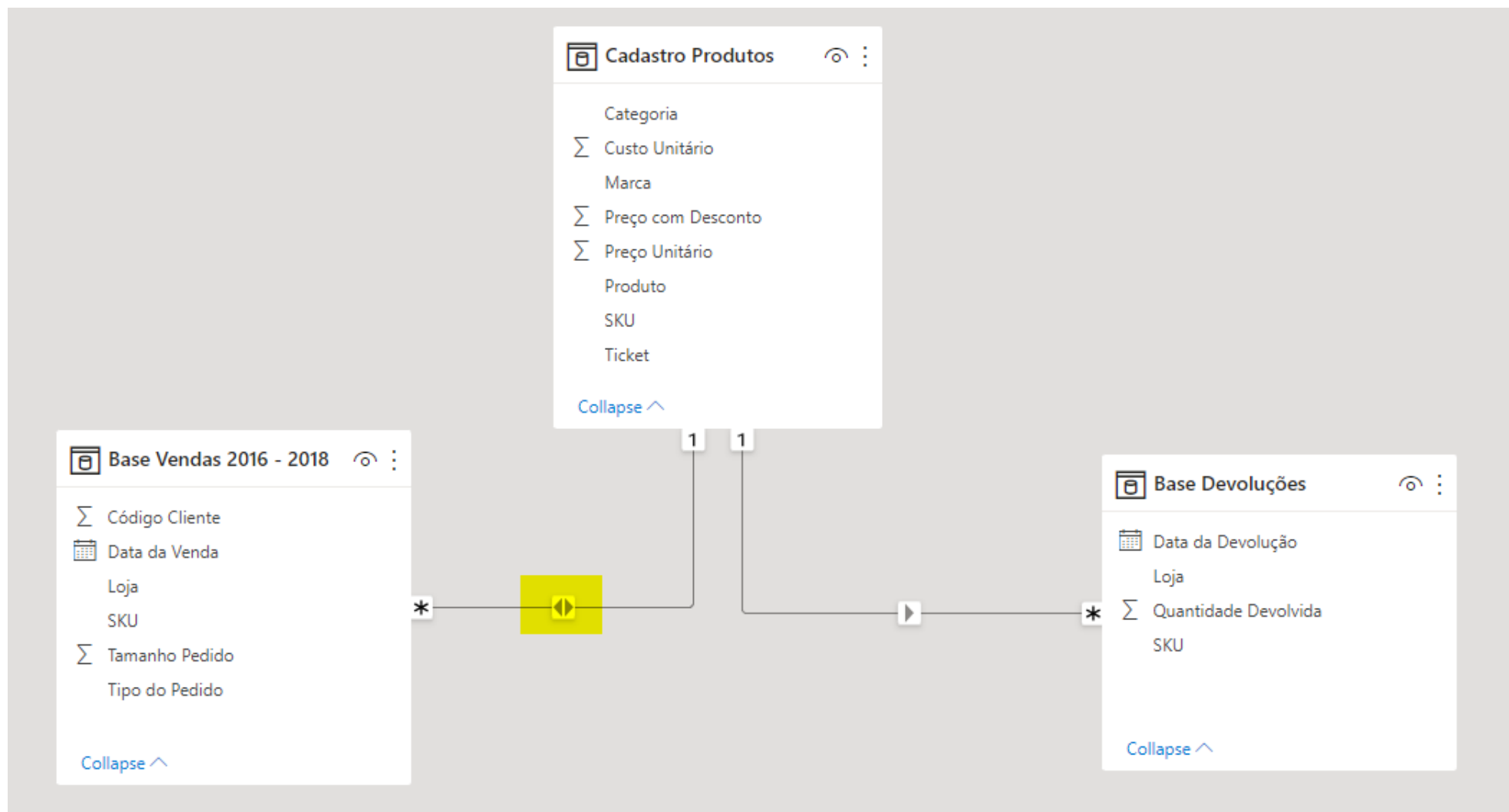
☐ Apply security filter in both directions

☐ Assume referential integrity

OK

Cancel

Agora temos um filtro bidirecional, ou seja, que funciona em ambos os sentidos:



A primeira diferença que vemos é que agora a seta aponta para os dois lados. Olhando nossos relatórios:

Compras e Devoluções por		
SKU	Tamanho Pedido	Quantidade Devolvida
HL1001	3030	101
HL1002	3081	
HL1003	3001	108
HL1004	2864	96
HL1005	2955	129
HL1006	3024	126
HL1007	3048	95
HL1008	2958	101
HL1009	2852	192
HL1010	2903	121
HL1011	2969	219
HL1012	2961	83
HL1013	3143	101
HL1014	2875	148
HL1015	2961	110
HL1016	2977	120
HL1017	2821	86
HL1018	2914	211
HL1019	3061	103
HL1020	3063	89
HL1021	3028	
HL1022	3075	128
HL1023	3024	
HL1024	2854	123
Total	71442	2590

- > Base Devoluções
- > Base Vendas 2016
- > Base Vendas 2016 - 2018
 - ☐ Σ Código Cliente
 - > ☐ Data da Venda
 - ☐ Loja
 - ☒ SKU
 - ☒ Σ Tamanho Pedido
 - ☐ Tipo do Pedido
- > Base Vendas 2017
- > Base Vendas 2018
- > Cadastro Clientes
- > Cadastro Lojas
- > Cadastro Produtos
 - ☐ Categoria
 - ☐ Σ Custo Unitário
 - ☐ Marca
 - ☐ Σ Preço com Desc...
 - ☐ Σ Preço Unitário
 - ☐ Produto
 - ☐ SKU
 - ☐ Ticket
- > Calendário

SKU	Custo Unitário
HL1001	R\$1.700
HL1002	R\$1.500
HL1003	R\$2.800
HL1004	R\$900
HL1005	R\$850
HL1006	R\$900
HL1007	R\$1.200
HL1008	R\$1.250
HL1009	R\$750
HL1010	R\$1.150
HL1011	R\$1.100
HL1012	R\$3.200
HL1013	R\$700
HL1014	R\$550
HL1015	R\$300
HL1016	R\$700
HL1017	R\$3.500
HL1018	R\$650
HL1019	R\$2.800
HL1020	R\$650
HL1021	R\$1.400
HL1022	R\$350
HL1023	R\$550
HL1024	R\$1.150
Total	R\$30.600

- > Base Devoluções
- > Base Vendas 2016
- ✓ Base Vendas 2016 - 2018
 - ☐ Σ Código Cliente
 - > ☐ Data da Venda
 - ☐ Loja
 - ✓ SKU
 - ☐ Σ Tamanho Pedido
 - ☐ Tipo do Pedido
- > Base Vendas 2017
- > Base Vendas 2018
- > Cadastro Clientes
- > Cadastro Lojas
- ✓ Cadastro Produtos
 - ☐ Categoria
 - ✓ Σ Custo Unitário
 - ☐ Marca
 - ☐ Σ Preço com Desc...
 - ☐ Σ Preço Unitário
 - ☐ Produto
 - ☐ SKU
 - ☐ Ticket
- > Calendário

3.3.2.1 Cuidados com o filtro bidirecional

Poderíamos tornar o filtro do relacionamento entre a tabela de produtos e de devoluções bidirecional também; entretanto, temos alguns motivos para não fazê-lo. Se mantermos mais de um filtro com duas direções, o Power BI pode dar erro - ele pode acabar não identificando as informações corretamente. Além disso, vamos analisar a nossa base de devoluções:

Compras e Devoluções por Produto

SKU	Tamanho Pedido	Quantidade Devolvida
HL1001	3030	101
HL1002	3081	
HL1003	3001	108
HL1004	2864	96
HL1005	2955	129
HL1006	3024	126
HL1007	3048	95
HL1008	2958	101
HL1009	2852	192
HL1010	2903	121
HL1011	2969	219
HL1012	2961	83
HL1013	3143	101
HL1014	2875	148
HL1015	2961	110
HL1016	2977	120
HL1017	2821	86
HL1018	2914	211
HL1019	3061	103
HL1020	3063	89
HL1021	3028	
HL1022	3075	128
HL1023	3024	
HL1024	2854	123
Total	71442	2590

Não são todos os produtos que já foram devolvidos; em contrapartida, todos já foram vendidos. Se fossemos usar o SKU da tabela de devoluções, teríamos um problema com essa informação, já que não encontraríamos todos os produtos no nosso filtro. Por isso precisamos ter muita atenção com os filtros bidirecionais, e evitá-los ao máximo - a não ser em casos que eles sejam extremamente necessários.

4 3.4 - Ocultando Campos do Relatório e Boas Práticas

Agora que já entendemos como funcionam os filtros, como criá-los e como gerenciá-los, vamos ver como manter a organização do nosso projeto simplificando-o (ocultando campos que não serão necessários) e algumas boas práticas que devemos seguir sempre.

Ocultando Campos do Relatório

Voltando para o nosso arquivo original...

Vimos na parte de filtros que se a gente escolhesse a coluna de SKU da base de vendas ou da base de devoluções teríamos um problema por não estarmos respeitando a direção do filtro. Então, vimos que **para fazer um filtro é melhor escolher a informação que esteja em uma tabela característica**.

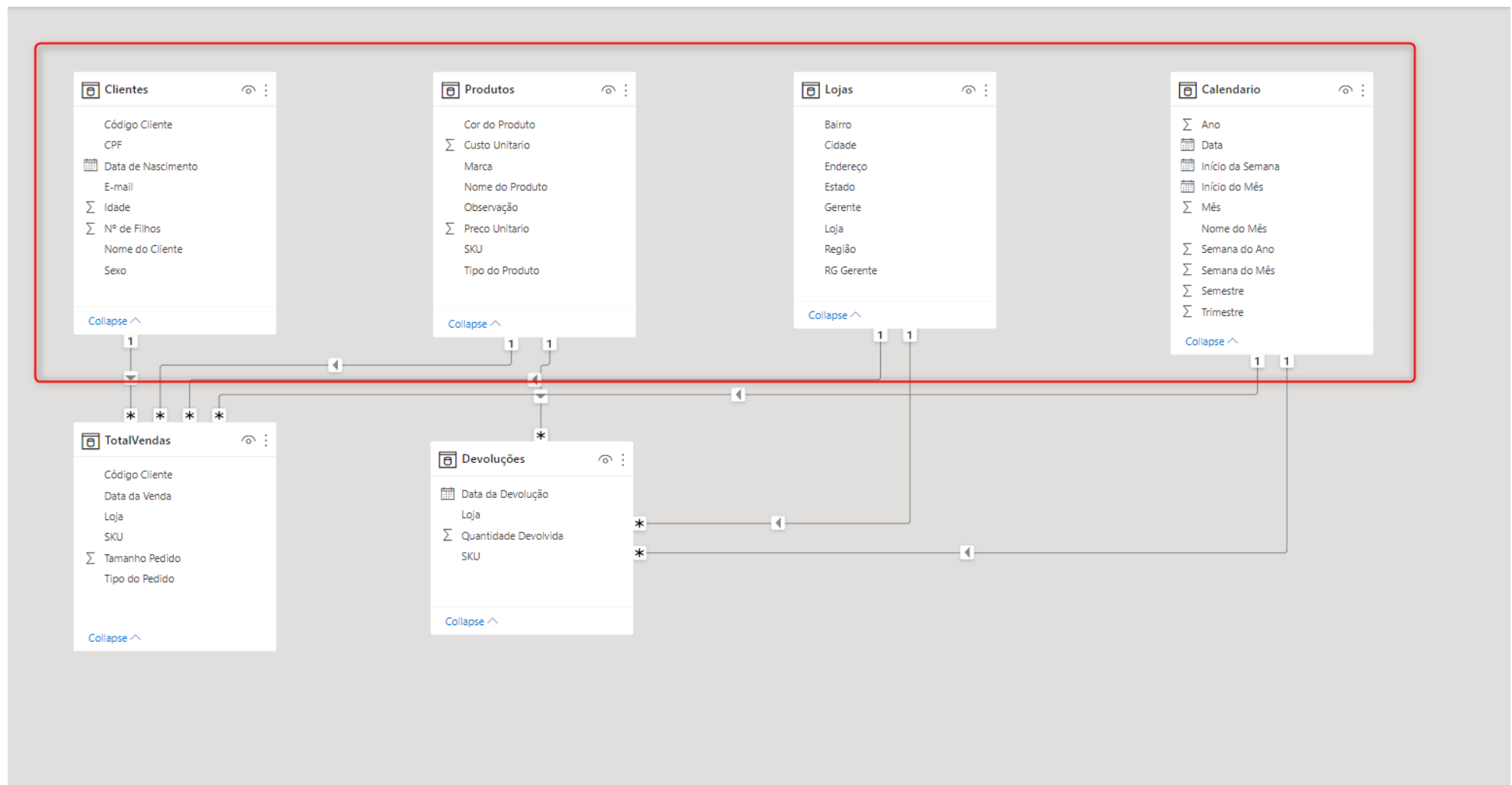
O ponto que levantamos aqui é que, dentro do relatório, podemos acabar nos confundindo com o tanto de informações que aparecem para nós. Nesse momento temos 6 bases diferentes, cada uma com várias colunas dentro.

Como um exemplo, quando fossemos criar uma matriz para analisar o total de produtos (SKU) vendidos, poderíamos acidentalmente selecionar a coluna SKU da base de vendas ao invés da base de produtos.

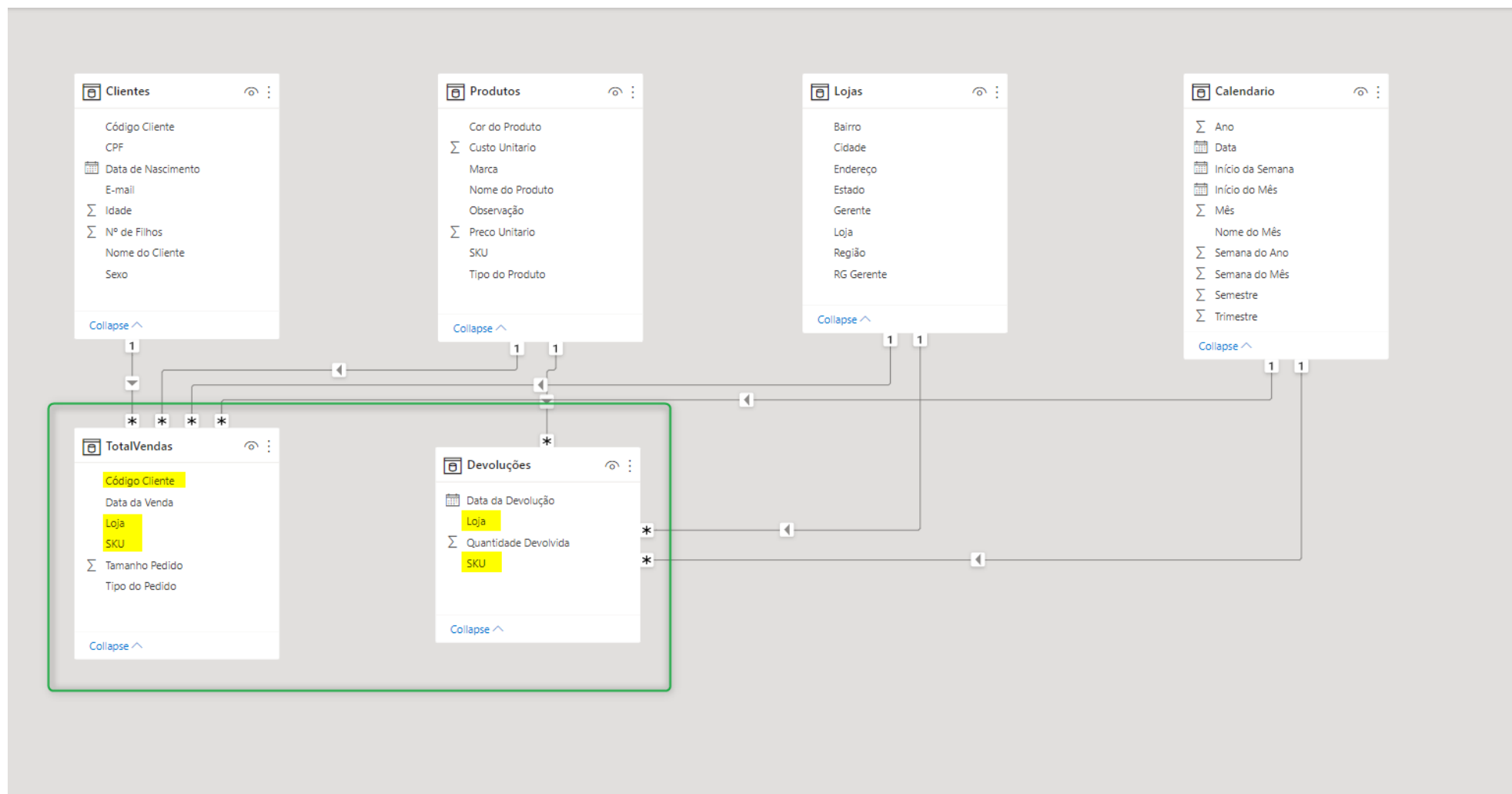
Portanto, o que vamos ver aqui é como "ocultar" essas colunas que não vamos utilizar na criação das matrizes, assim deixamos nosso projeto mais limpo e organizado.

Voltando na guia modelo, vamos analisar quais colunas vamos ocultar para evitar qualquer seleção errada na hora de montar as matrizes e gráficos. **A dica basicamente é identificar as chaves primárias (tabelas características) e estrangeiras (tabelas fato) e ocultar as chaves estrangeiras das tabelas fato.**

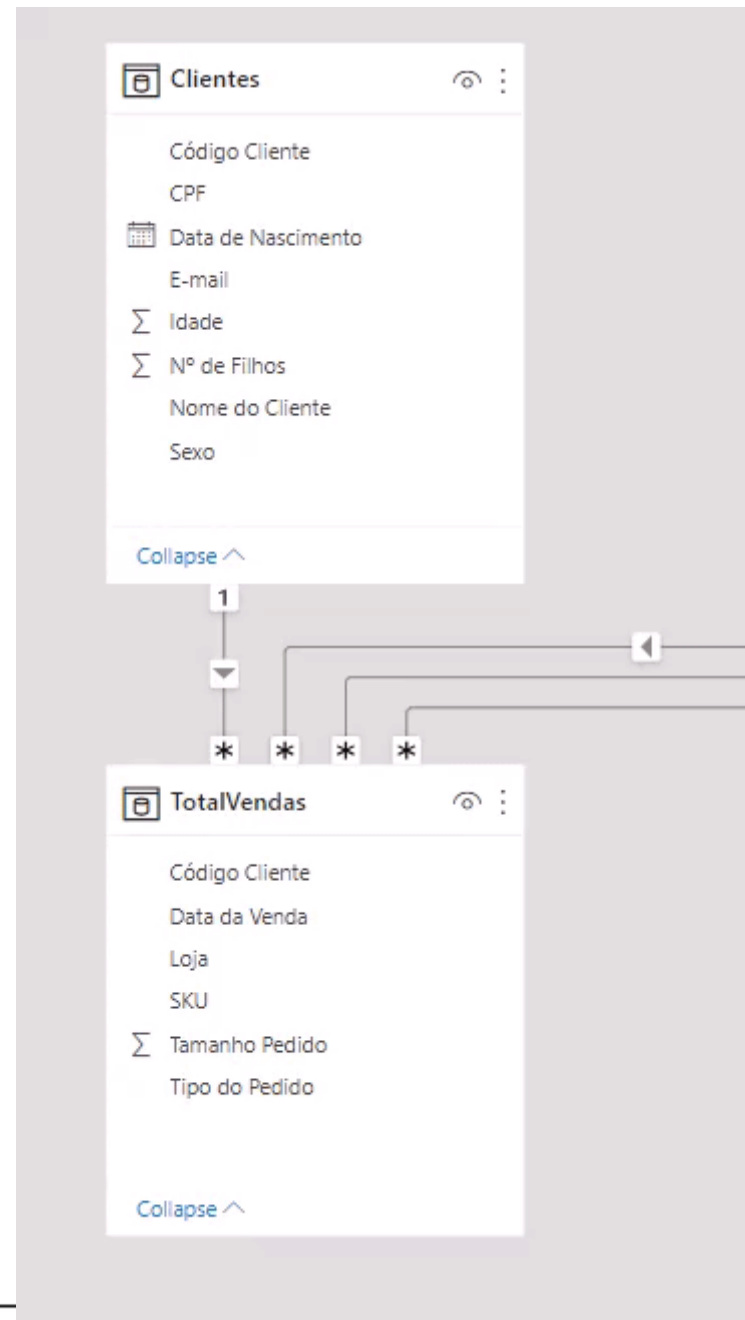
Normalmente, quando criarmos os filtros nas análises dos relatórios, vamos puxar as informações dessas quatro bases características:



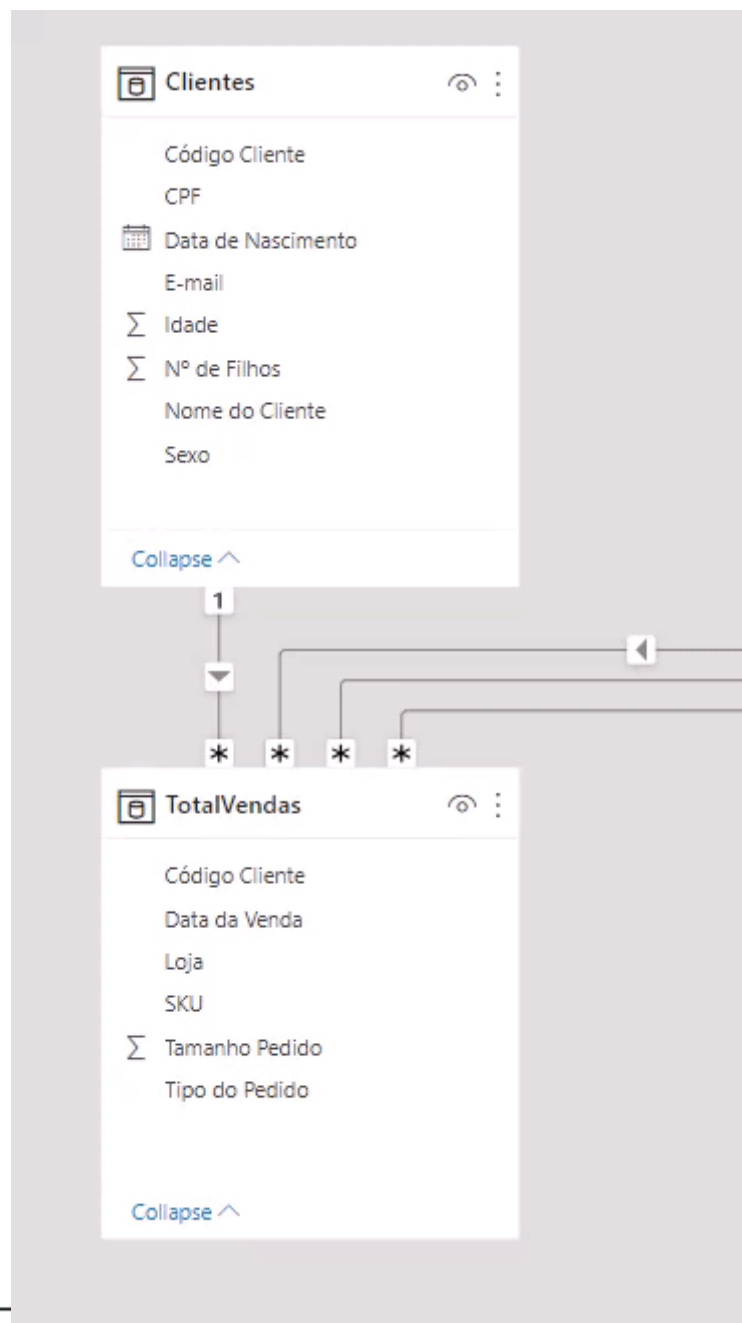
As colunas que foram grifadas já aparecem como chaves primárias, então não é necessário que apareçam como opções nas bases da guia Relatórios; logo, **podemos ocultá-las**:



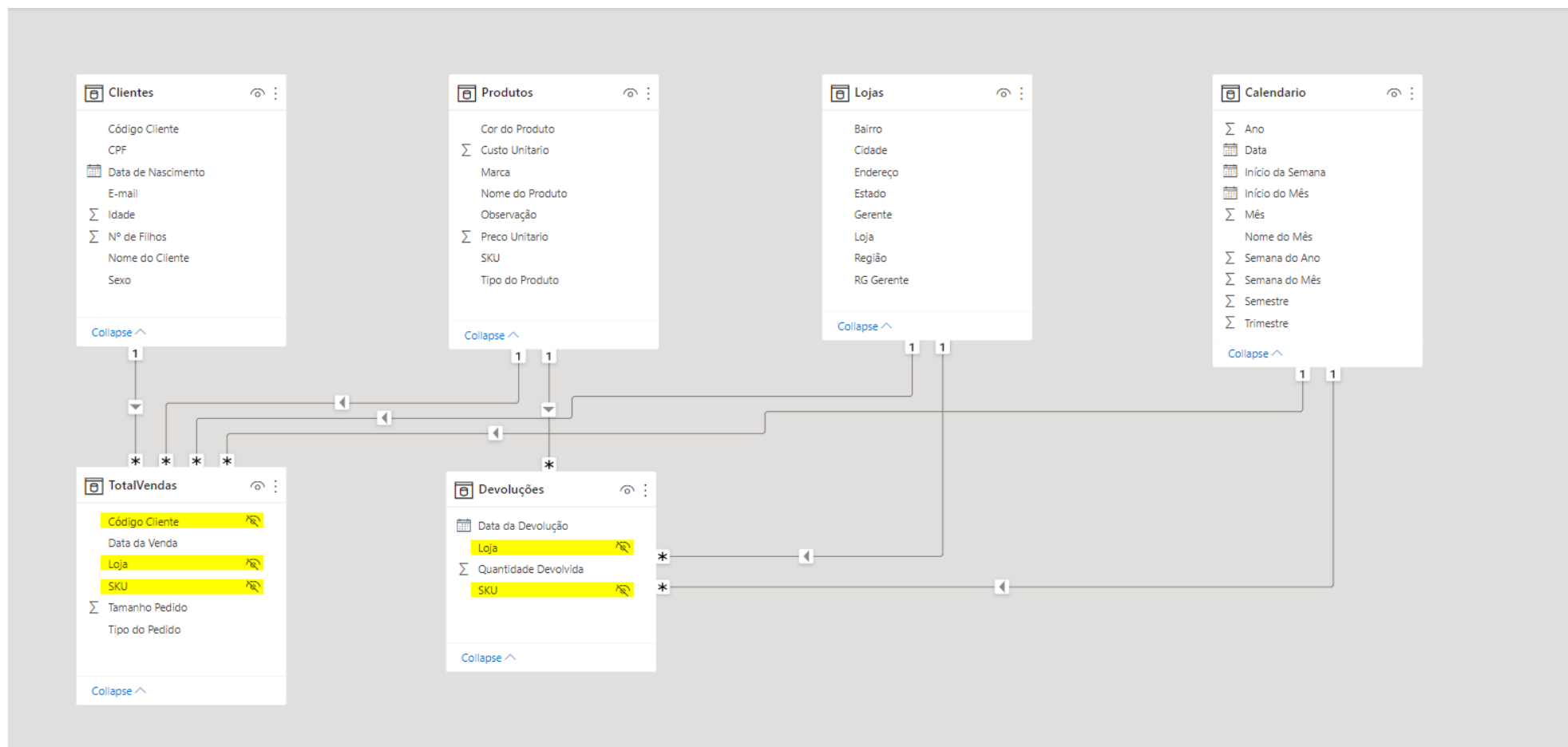
Para ocultar uma coluna em uma base, basta clicar em cima dela com o botão direito e selecionar a opção **Ocultar da Visualização do Relatório** (*Hide in report view*):



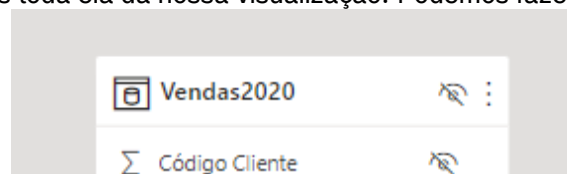
Após fazer isso, vamos ver um símbolo de olho com um risco atravessado - representando que a visualização foi ocultada. De maneira mais simples, podemos apenas passar o mouse por cima do nome da coluna, e ele nos mostra o símbolo do olho; clicando nele conseguimos alterar seu estado, entre visível e oculto:

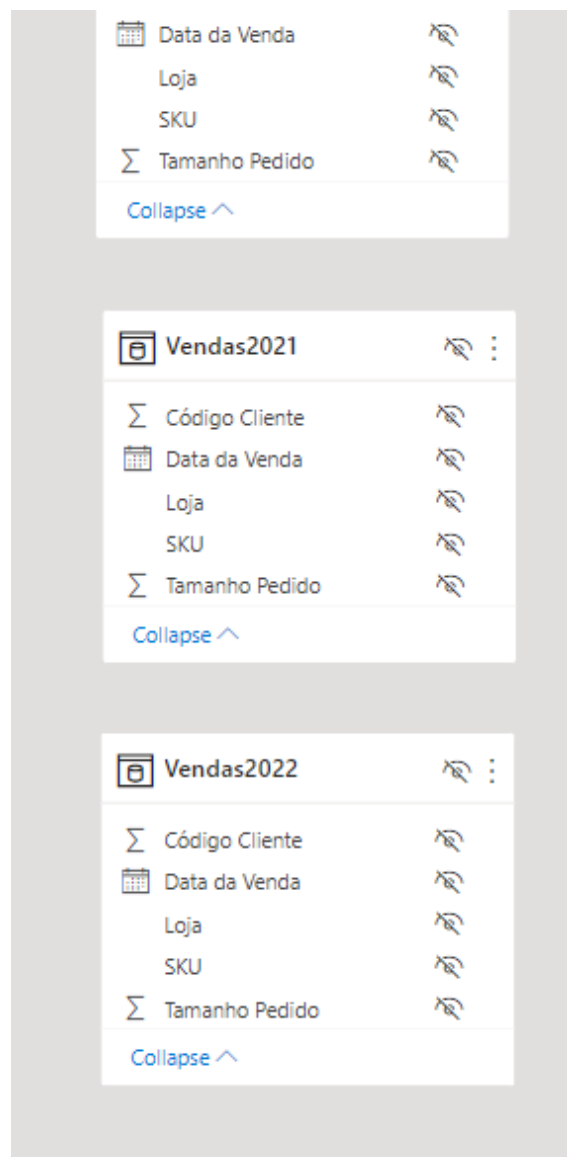


Vamos fazer isso para as 5 colunas: **Código Cliente**, **Loja** e **SKU** da base Vendas e **Loja** e **SKU** da base Devoluções. Feito, o resultado é o seguinte:



Clicando no símbolo ao lado do nome da tabela, ocultamos toda ela da nossa visualização. Podemos fazer isso com as bases de vendas anuais:

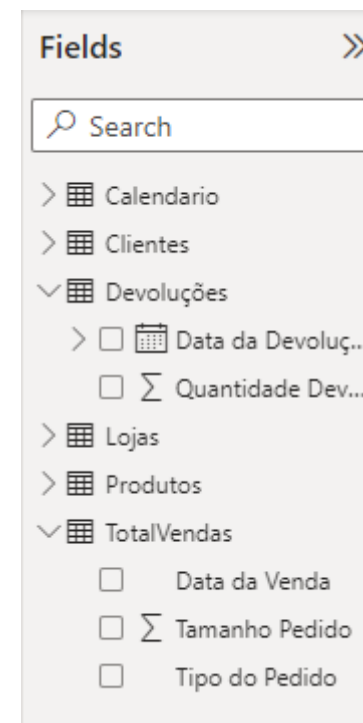
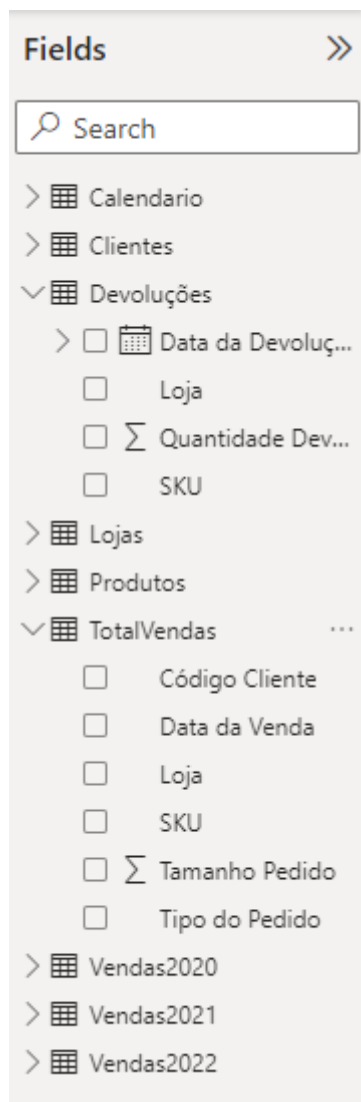




Voltando na guia relatório, podemos fazer um antes e depois com os campos ocultos:

• **Antes dos campos ocultos:**

• **Depois:**



Com a lista de campos mais limpa, facilitamos o nosso trabalho na hora de montar os relatórios e *dashboards*, deixando o nosso projeto mais dinâmico e organizado!!

4.1 Aula 3 - Boas Práticas

4.1.1 Crie um modelo de dados organizado e normalizado

- Cada tabela deve servir para um propósito (Vendas, Devoluções, Clientes, etc.);
- Use relacionamentos no lugar de colunas mescladas - as tabelas ficam mais leves;

4.1.2 Organize Tabelas Características acima das Tabelas Fato

- Isso torna mais visual e intuitivo criar os relacionamentos entre Chaves Primárias e Chaves Estrangeiras;
- Ajuda a lembrar que o filtro só desce de Tabelas Características para as Tabelas Fato;

4.1.3 Evite usar filtro bidirecional a não ser que seja necessário

- Isso pode causar erros no seu relatório, como a falta de informações;

4.1.4 Oculte no relatório campos que não precisam ser usados

- Isso ajuda a evitar erros de filtro no relatório;
- Deixa o relatório muito mais organizado e visual.