

Guia de relações muitos-para-muitos

Este artigo destina-se aos modeladores de dados que trabalham com o Power BI Desktop. Descreve três diferentes cenários de modelos muitos-para-muitos. Também lhe fornece orientações sobre como os estruturar com êxito nos seus modelos.

Nota

A introdução às relações de modelos não é abordada neste artigo. Se não está totalmente familiarizado com as relações, as suas propriedades ou como configurá-las, recomendamos que leia primeiro o artigo [Relações de modelos no Power BI Desktop](#).

Também é importante que compreenda o design do esquema de estrela. Para obter mais informações, veja [Compreender o que é um esquema de estrela e qual a importância para o Power BI](#).

Na realidade, existem três cenários “muitos-para-muitos”. Podem ocorrer quando precisar de:

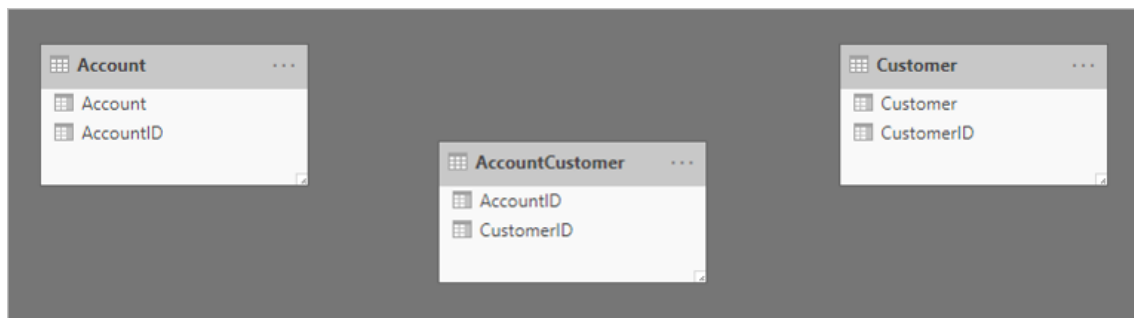
- [Relacionar duas tabelas de dimensão](#)
- [Relacionar tabelas de factos](#)
- [Relacionar tabelas de factos de granulação superior](#), quando a tabela de factos armazena linhas a uma granulação superior do que as linhas da tabela de dimensão

Relacionar dimensões muitos-para-muitos

Vamos considerar o primeiro tipo de cenário muitos-para-muitos com um exemplo. O cenário clássico relaciona duas entidades: clientes de bancos e contas bancárias. Considere que os clientes podem ter múltiplas contas e as contas podem ter múltiplos clientes. Quando as contas têm múltiplos clientes, denominam-se normalmente de *titulares de conta conjunta*.

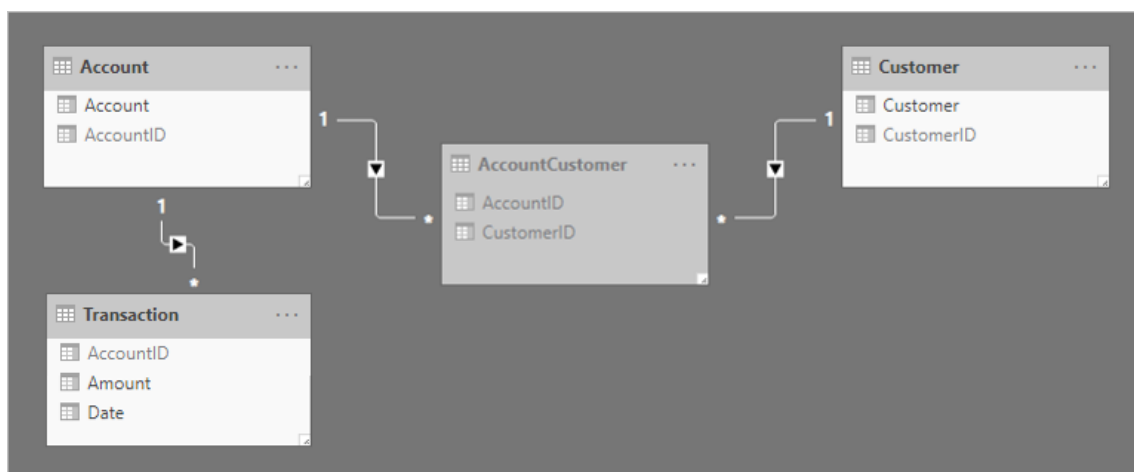
Modelar estas entidades é uma tarefa simples. Uma tabela de tipo de dimensão armazena contas e outra tabela de tipo de dimensão armazena clientes. Como é característico das tabelas de tipo de dimensão, existe uma coluna de ID em cada tabela. Para modelar a relação entre duas tabelas, é necessária uma terceira tabela. Esta tabela é normalmente conhecida como uma *tabela de bridging*. Neste exemplo, a sua finalidade consiste em armazenar uma linha para cada associação de conta de cliente. Curiosamente, quando esta tabela contém apenas as colunas de ID, chama-se uma [tabela de factos sem factos](#).

Segue-se um diagrama de modelo simplista das três tabelas.



A primeira conta chama-se **Account** e contém duas colunas: **AccountID** e **Account**. A primeira conta chama-se **AccountCustomer** e contém duas colunas: **AccountID** e **CustomerID**. A terceira tabela chama-se **Customer** e contém duas colunas: **CustomerID** e **Customer**. As relações não existem entre nenhuma das tabelas.

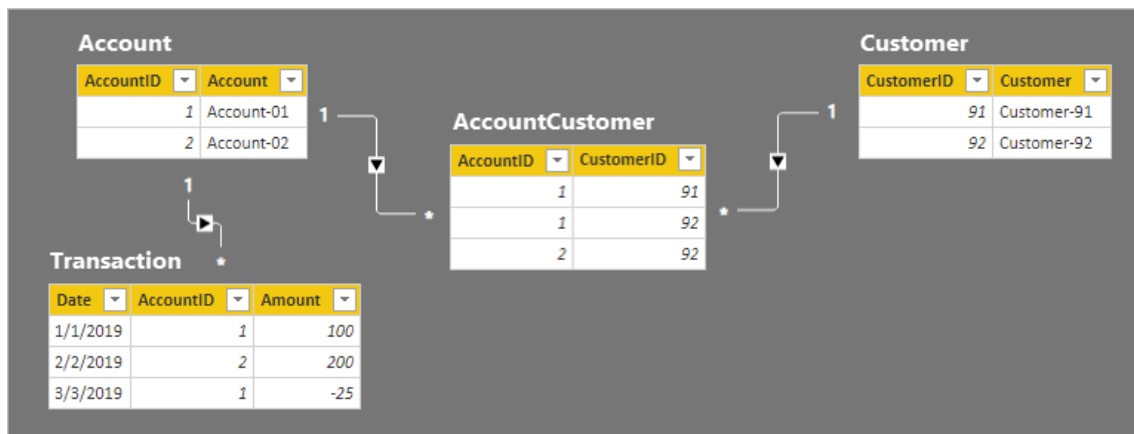
São adicionadas duas relações um-para-muitos para relacionar as tabelas. Segue-se um diagrama de modelo atualizado das tabelas relacionadas. Foi adicionada uma tabela de factos chamada **Transaction**. Regista todas as transações de conta. A tabela de bridging e todas as colunas de ID foram ocultadas.



Para ajudar a descrever como funciona a propagação do filtro de relacionamento, o diagrama de modelo foi modificado para revelar as linhas da tabela.

Nota

Não é possível apresentar linhas de tabela no diagrama de modelo do Power BI Desktop. Isto é feito neste artigo para o demonstrar com exemplos claros.



Os detalhes da linha para as quatro tabelas estão descritos na seguinte lista com marcas:

- A tabela **Account** tem duas linhas:
 - **AccountID** 1 é para Account-01
 - **AccountID** 2 é para Account-02
- A tabela **Customer** tem duas linhas:
 - **CustomerID** 91 é para Customer-91
 - **CustomerID** 92 é para Customer-92
- A tabela **AccountCustomer** tem três linhas:
 - **AccountID** 1 está associada a **CustomerID** 91
 - **AccountID** 1 está associada a **CustomerID** 92
 - **AccountID** 2 está associado ao **CustomerID** 92
- A tabela **Transaction** tem três linhas:
 - **Date** 1 de janeiro de 2019, **AccountID** 1, **Amount** 100
 - **Date** 2 de fevereiro de 2019, **AccountID** 2, **Amount** 200
 - **Date** 3 de março de 2019, **AccountID** 1, **Amount** -25

Vejamos o que acontece quando o modelo é consultado.

Abaixo estão dois elementos visuais que resumem a coluna **Amount** da tabela **Transaction**. O primeiro elemento visual agrupa por conta e, como tal, a soma das colunas **Amount** representa o *saldo da conta*. O segundo elemento visual agrupa por cliente e, como tal, a soma das colunas **Amount** representa o *saldo do cliente*.

Account Balance		Customer Balance	
Account	Amount	Customer	Amount
Account-01	75	Customer-91	275
Account-02	200	Customer-92	275
Total	275	Total	275

O primeiro elemento visual intitula-se **Account Balance** e tem duas colunas: **Account** e **Amount**. Apresenta o seguinte resultado:

- O valor do saldo de Account-01 é 75
- O valor do saldo de Account-02 é 200
- O total é 275

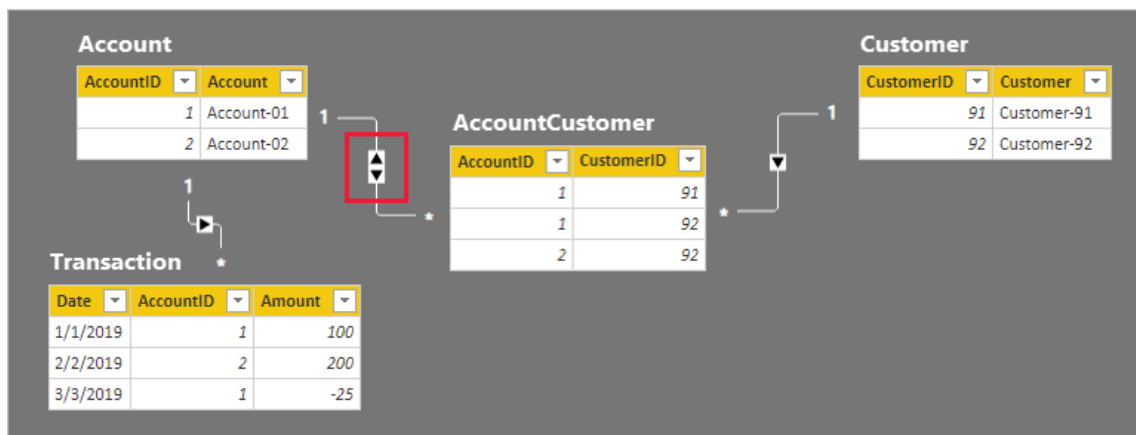
O segundo elemento visual intitula-se **Customer Balance** e tem duas colunas: **Customer** e **Amount**. Apresenta o seguinte resultado:

- O valor do saldo de Customer-91 é 275
- O valor do saldo de Customer-92 é 275
- O total é 275

Uma análise rápida das linhas da tabela e o elemento visual **Account Balance** revela que o resultado está correto, para cada conta e o montante total. Tal deve-se ao facto de cada agrupamento de conta resultar numa propagação do filtro para a tabela **Transaction** para essa conta.

No entanto, algo parece não estar correto no elemento visual **Customer Balance**. Cada cliente no elemento visual **Customer Balance** tem o mesmo saldo que o saldo total. Este resultado só poderia estar correto se todos os clientes fossem titulares de conta conjunta de todas as contas. Não é o caso neste exemplo. O problema está relacionado com a propagação do filtro. Não está a fluir totalmente até à tabela **Transaction**.

Siga as direções do filtro de relação da tabela **Customer** até à tabela **Transaction**. Deverá ser aparente que a relação entre a tabela **Account** e **AccountCustomer** se está a propagar na direção errada. A direção do filtro desta relação deve ser definida para **Both**.



Account Balance		Customer Balance	
Account	Amount	Customer	Amount
Account-01	75	Customer-91	75
Account-02	200	Customer-92	275
Total	275	Total	275

Como esperado, não houve alteração no elemento visual **Account Balance**.

No entanto, o elemento visual **Customer Balance** apresenta agora o seguinte resultado:

- O valor do saldo de Customer-91 é 75
- O valor do saldo de Customer-92 é 275
- O total é 275

O elemento visual **Customer Balance** apresenta agora um resultado correto. Siga as direções do filtro e veja como os saldos do cliente foram calculados. Além disso, compreenda que o total do elemento visual significa *todos os clientes*.

Alguém que não esteja familiarizado com as relações do modelo poderá concluir que o resultado está incorreto. Essa pessoa poderá perguntar: *Porque é que o saldo total para Customer-91 e Customer-92 equivale a 350 (75 + 275)?*

A resposta a esta pergunta está na compreensão da relação muitos-para-muitos. Cada saldo de cliente pode representar a soma de múltiplos saldos de conta e, portanto, os saldos de conta são *não aditivos*.

Orientação para relacionar dimensões muitos-para-muitos

Quando tem uma relação muitos-para-muitos entre tabelas de tipo de dimensão, fornecemos a seguinte orientação:

- Adicione cada entidade relacionada muitos-para-muitos como uma tabela de modelo, garantindo que tem uma coluna de identificador exclusivo (ID)
- Adicione uma tabela de bridging para armazenar entidades associadas
- Crie relações um-para-muitos entre as três tabelas
- Configure **uma** relação bidirecional para permitir que a propagação do filtro continue para as tabelas de factos
- Quando não for apropriado ter valores de ID em falta, defina a propriedade **Is Nullable** da colunas de ID como FALSE. A atualização de dados falhará se os valores em falta estiverem atribuídos
- Oculte a tabela de bridging (a menos que contenha colunas ou medidas adicionais necessárias para a criação de relatórios)
- Oculte todas as colunas de ID que não forem adequadas para a criação de relatórios (por exemplo, quando os IDs são chaves de substituição)
- Se fizer sentido deixar uma coluna de ID visível, certifique-se de que a mesma está no lado "um" da relação. Oculte sempre a coluna do lado "muitos". Isto resulta no melhor desempenho de filtro.
- Para evitar confusão ou uma interpretação errada, comunique as explicações aos utilizadores do seu relatório. Pode adicionar descrições com caixas de texto ou [descrições do cabeçalho do elemento visual](#)

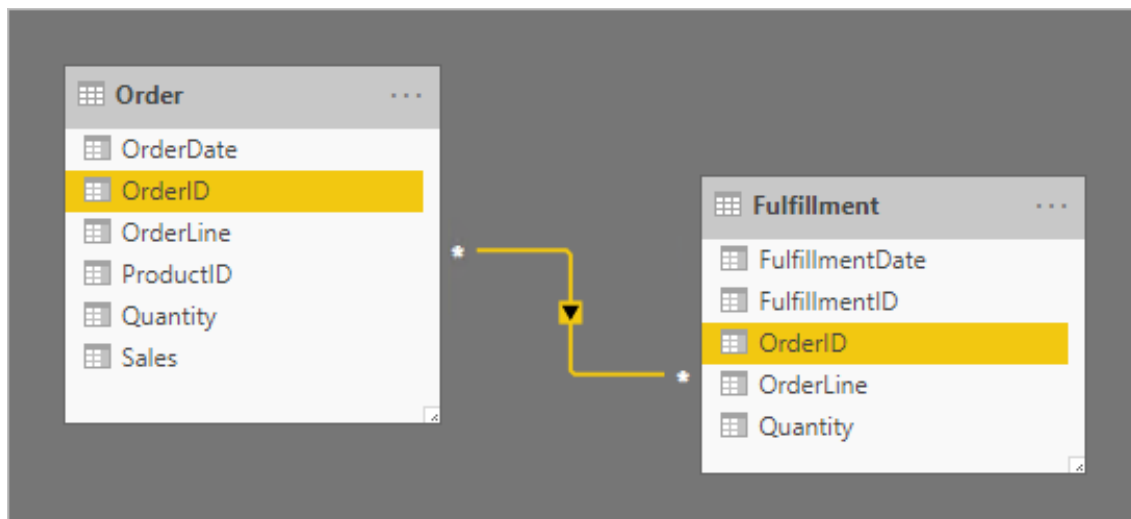
Não recomendamos que relacione tabelas de dimensão muitos-para-muitos diretamente. Esta abordagem de design requer a configuração de uma relação com uma cardinalidade muitos-para-muitos. A nível conceptual, isto pode ser alcançado. No entanto, implica que as colunas relacionadas possam incluir valores duplicados. Trata-se de uma prática de design bem aceite, no entanto, que as tabelas de tipo de dimensão tenham uma coluna de

ID. As tabelas de tipo de dimensão devem utilizar sempre a coluna de ID como o lado "um" da relação.

Relacionar factos muitos-para-muitos

O segundo tipo de cenário muitos-para-muitos envolve relacionar duas tabelas de factos. Duas tabelas de factos podem ser relacionadas diretamente. Esta técnica de design pode ser útil para uma exploração de dados rápida e simples. No entanto, e para esclarecer, normalmente não recomendamos esta abordagem de design. Iremos explicar esta situação mais à frente nesta secção.

Vamos considerar um exemplo que envolve duas tabelas de factos: **Order** e **Fulfillment**. A tabela **Order** contém uma linha por linha de encomenda e a tabela **Fulfillment** pode conter zero ou mais linhas por linha de encomenda. As linhas na tabela **Order** representam as encomendas de vendas. As linhas na tabela **Fulfillment** representam itens de encomenda que foram enviados. Uma relação muitos-para-muitos relaciona as duas colunas **OrderID**, com propagação de filtro apenas a partir da tabela **Order** (**Order** filtra **Fulfillment**).



A cardinalidade da relação é definida como muitos-para-muitos para permitir o armazenamento de valores duplicados **OrderID** em ambas as tabelas. Na tabela **Order**, os valores duplicados **OrderID** podem existir porque uma encomenda pode ter múltiplas linhas. Na tabela **Fulfillment**, os valores duplicados **OrderID** podem existir porque as encomendas podem ter múltiplas linhas e as linhas de encomendas podem ser cumpridas por vários envios.

Vamos agora analisar as linhas da tabela. Na tabela **Fulfillment**, observe que as linhas de encomenda podem ser cumpridas por múltiplos envios. (A ausência de uma linha de encomenda significa que a encomenda ainda não foi entregue.)

Order					
OrderDate	OrderID	OrderLine	ProductID	OrderQuantity	Sales
01/01/2019	1	1	Prod-A	5	50
01/01/2019	1	2	Prod-B	10	80
02/02/2019	2	1	Prod-B	5	40
02/02/2019	2	2	Prod-C	1	20
03/03/2019	3	1	Prod-C	5	100

Fulfillment				
FulfillmentDate	FulfillmentID	OrderID	OrderLine	FulfillmentQuantity
01/01/2019	50	1	1	2
02/02/2019	51	2	1	5
02/02/2019	52	1	1	3
02/02/2019	53	1	2	10

Os detalhes da linha para as duas tabelas estão descritos na seguinte lista com marcas:

- A tabela **Order** tem cinco linhas:
 - **OrderDate** 1 de janeiro de 2019, **OrderID** 1, **OrderLine** 1, **ProductID** Prod-A, **OrderQuantity** 5, **Sales** 50
 - **OrderDate** 1 de janeiro de 2019, **OrderID** 1, **OrderLine** 2, **ProductID** Prod-B, **OrderQuantity** 10, **Sales** 80
 - **OrderDate** 2 de fevereiro de 2019, **OrderID** 2, **OrderLine** 1, **ProductID** Prod-B, **OrderQuantity** 5, **Sales** 40
 - **OrderDate** 2 de fevereiro de 2019, **OrderID** 2, **OrderLine** 2, **ProductID** Prod-C, **OrderQuantity** 1, **Sales** 20
 - **OrderDate** 3 de março de 2019, **OrderID** 3, **OrderLine** 1, **ProductID** Prod-C, **OrderQuantity** 5, **Sales** 100
- A tabela **Fulfillment** tem quatro linhas:
 - **FulfillmentDate** 1 de janeiro de 2019, **FulfillmentID** 50, **OrderID** 1, **OrderLine** 1, **FulfillmentQuantity** 2
 - **FulfillmentDate** 2 de fevereiro de 2019, **FulfillmentID** 51, **OrderID** 2, **OrderLine** 1, **FulfillmentQuantity** 5
 - **FulfillmentDate** 2 de fevereiro de 2019, **FulfillmentID** 52, **OrderID** 1, **OrderLine** 1, **FulfillmentQuantity** 3
 - **FulfillmentDate** 1 de janeiro de 2019, **FulfillmentID** 53, **OrderID** 1, **OrderLine** 2, **FulfillmentQuantity** 10

Vejamos o que acontece quando o modelo é consultado. Segue-se um elemento visual de tabela que compara as quantidades de encomenda e cumprimento pela tabela **Order** coluna **OrderID**.

Order and Fulfillment Quantities		
OrderID	OrderQuantity	FulfillmentQuantity
1	15	15
2	6	5
3	5	
Total	26	20

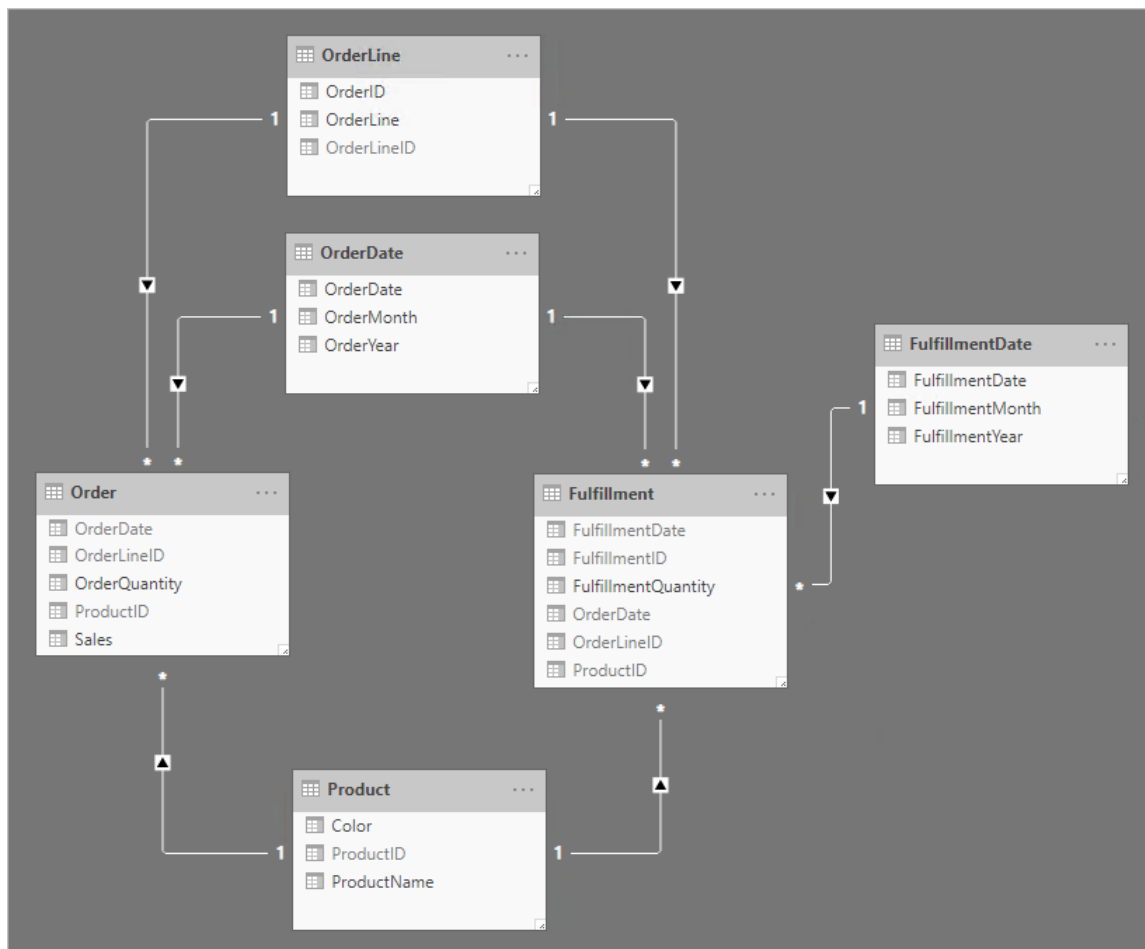
O elemento visual apresenta um resultado preciso. No entanto, a utilidade do modelo é limitada – só pode filtrar ou agrupar pela tabela **Order** coluna **OrderID**.

Orientação para relacionar factos muitos-para-muitos

Normalmente, não recomendamos relacionar duas tabelas de factos diretamente utilizando a cardinalidade muitos-para-muitos. O motivo principal é porque o modelo não irá fornecer a flexibilidade na forma como relata o filtro ou agrupamento de elementos visuais. No exemplo fornecido, só é possível filtrar ou agrupar elementos visuais pela coluna **OrderID** da tabela **Order**. Um motivo adicional está relacionado com a qualidade dos seus dados. Se os dados tiverem problemas de integridade, algumas linhas poderão ser omitidas durante a consulta devido à natureza da *relação limitada*. Para obter mais informações, veja [Relações de modelos no Power BI Desktop \(Avaliação de relações\)](#).

Em vez de relacionar tabelas de factos diretamente, recomendamos que adote os princípios do design [Esquema de Estrela](#). Pode fazê-lo ao adicionar tabelas de dimensão. As tabelas de dimensão relacionam-se então com as tabelas de factos através de relações um-para-muitos. Esta abordagem de design é robusta, pois fornece opções de relatórios flexíveis. Permite-lhe filtrar ou agrupar com qualquer uma das colunas de dimensão e resumir qualquer tabela de factos.

Vamos considerar uma solução melhor.



Observe as seguintes alterações de design:

- O modelo tem agora quatro tabelas adicionais: **OrderLine**, **OrderDate**, **Product** e **FulfillmentDate**
- As quatro tabelas adicionais são todas tabelas de dimensão e as relações um-para-muitos relacionam estas tabelas com as tabelas de factos
- A tabela **OrderLine** contém uma coluna **OrderLineID**, que representa o valor **OrderID** multiplicado por 100, além do valor **OrderLine** — um identificador exclusivo para cada linha de encomenda
- As tabelas **Order** e **Fulfillment** contêm agora uma coluna **OrderLineID** e já não contêm as colunas **OrderID** e **OrderLine**
- A tabela **Fulfillment** contém agora as colunas **OrderDate** e **ProductID**
- A tabela **FulfillmentDate** relaciona-se apenas à tabela **Fulfillment**
- Todas as colunas do identificador exclusivo estão ocultas

Dedicar tempo para aplicar os princípios de design do esquema de estrela resulta nas seguintes vantagens:

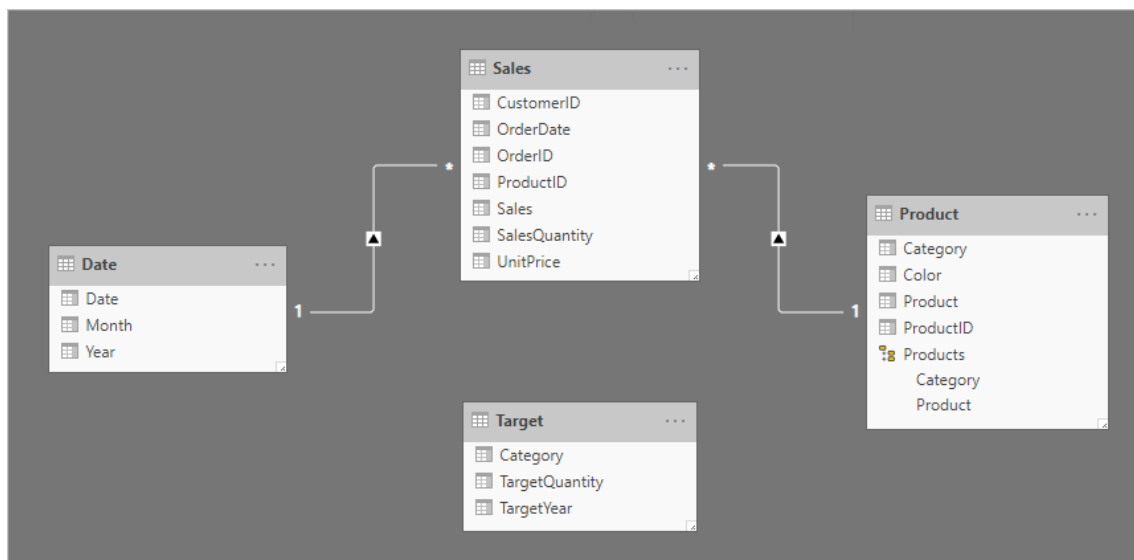
- Os elementos visuais do seu relatório podem *filtrar ou agrupar* por qualquer coluna visível a partir das tabelas de tipo de dimensão
- Os elementos visuais do seu relatório podem *resumir* qualquer coluna visível a partir das tabelas de factos

- Os filtros aplicados às tabelas **OrderLine**, **OrderDate** ou **Product** irão propagar para ambas as tabelas de tipos de factos
- Todas as relações são um-para-muitos e cada relação é uma *relação regular*. Os problemas de integridade de dados não serão dissimulados. Para obter mais informações, veja [Relações de modelos no Power BI Desktop \(Avaliação de relações\)](#).

Relacionar factos de agregação superior

Este cenário muitos-para-muitos é muito diferente dos outros dois já descritos neste artigo.

Vamos considerar um exemplo que envolva quatro tabelas: **Date**, **Sales**, **Product** e **Target**. As tabelas **Date** e **Product** são tabelas de dimensão e as relações um-para-muitos relacionam-se com a tabela de factos **Sales**. Até agora, representa um bom design de esquema de estrela. No entanto, a tabela **Target** ainda não foi relacionada com as outras tabelas.



A tabela **Target** contém três colunas: **Category**, **TargetQuantity** e **TargetYear**. As linhas da tabela revelam a granularidade do ano e categoria do produto. Dito de outra forma, os objetivos (utilizados para medir o desempenho de vendas) são definidos anualmente para cada categoria de produto.

TargetYear	Category	TargetQuantity
01/01/2019	Clothing	10
01/01/2019	Accessories	20
01/01/2019	Clothing	30
01/01/2020	Accessories	40
01/01/2020	Food	100
01/01/2020	Food	200

Como a tabela **Target** armazena dados a um nível superior das tabelas de tipo de dimensão, não é possível criar uma relação um-para-muitos. Bem, tal é verdade para

apenas uma das relações. Vamos explorar como a tabela **Target** pode ser relacionada com as tabelas de tipo de dimensão.

Relacionar períodos de tempo de agregação superior

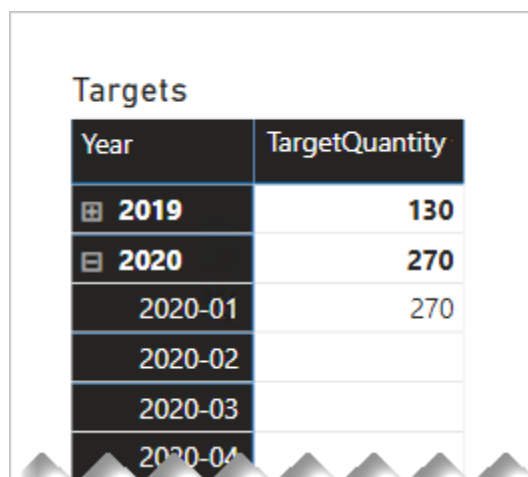
Uma relação entre as tabelas **Date** e **Target** deve ser uma relação um-para-muitos. Tal deve-se ao facto de os valores da coluna **TargetYear** serem datas. Neste exemplo, cada valor da coluna **TargetYear** é a primeira data do ano do objetivo.

Dica

Ao armazenar factos numa granularidade de tempo maior do que o dia, defina o tipo de dados da coluna para **Date** (ou **Whole number** se estiver a utilizar chaves de datas). Na coluna, armazene um valor que represente o primeiro dia do período de tempo. Por exemplo, um período de ano é registado como 1 de janeiro do ano e um período de mês é registado como o primeiro dia desse mês.

No entanto, há que ter cuidado para garantir que esses filtros de nível de mês ou data produzem um resultado significativo. Sem qualquer lógica especial de cálculo, os elementos visuais de relatórios podem relatar que as datas de destino são, literalmente, o primeiro dia de cada ano. Todos os restantes dias (e todos os meses, exceto janeiro) irão resumir a quantidade de destino como BLANK.

O seguinte elemento visual de matriz mostra o que acontece quando o utilizador do relatório analisa um ano e os seus respetivos meses. O elemento visual está a resumir a coluna **TargetQuantity**. (A opção [Mostrar itens sem dados](#) foi ativada para as linhas de matriz.)



Targets	
Year	TargetQuantity
2019	130
2020	270
2020-01	270
2020-02	
2020-03	
2020-04	

Para evitar este comportamento, recomendamos que controle o resumo dos seus dados de factos através de medidas. Uma forma de controlar o resumo consiste em devolver BLANK quando os períodos de tempo de nível inferior forem consultados. Outra forma, definida com um DAX sofisticado, consiste em distribuir valores entre períodos de tempo de nível inferior.

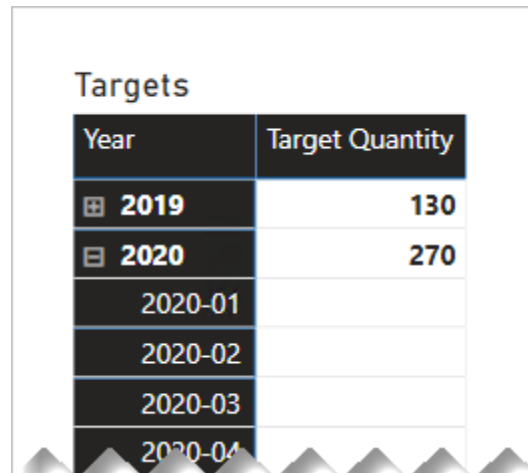
Considere a seguinte definição de medida que utiliza a função DAX [ISFILTERED](#). Devolve apenas um valor quando as colunas **Date** ou **Month** não estão filtradas.

DAX

Target Quantity =

```
IF(
    NOT ISFILTERED('Date'[Date])
    && NOT ISFILTERED('Date'[Month]),
    SUM(Target[TargetQuantity])
)
```

O seguinte elemento visual de matriz utiliza agora a medida **Target Quantity**. Mostra que todas as quantidades de destino mensais apresentam BLANK.

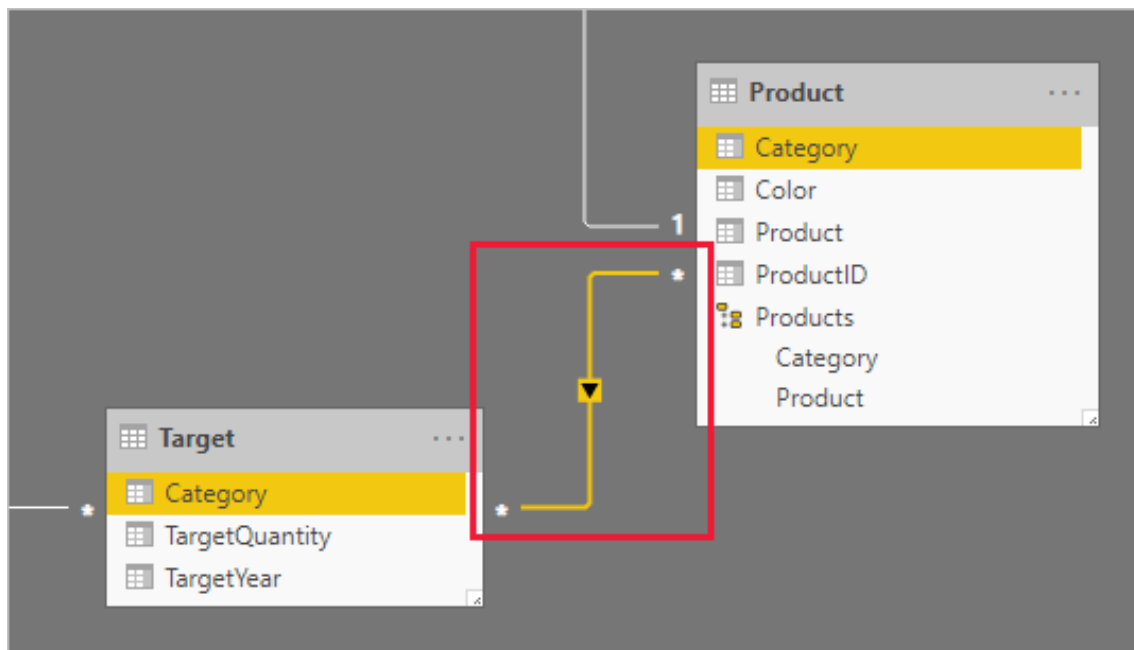


Year	Target Quantity
2019	130
2020	270
2020-01	
2020-02	
2020-03	
2020-04	

Relacionar agregação superior (que não seja uma data)

É necessária uma abordagem diferente ao relacionar uma coluna que não seja uma data a partir de uma tabela de dimensão para uma tabela de factos (e tem uma agregação superior à da tabela de dimensão).

As colunas **Category** (ambas das tabelas **Product** e **Target**) contêm valores duplicados. Portanto, não existe "um" para uma relação um-para-muitos. Neste caso, terá de criar uma relação muitos-para-muitos. A relação deve propagar filtros numa única direção, da tabela de dimensão para a tabela de factos.



Vamos agora analisar as linhas da tabela.

The diagram shows the data rows for the **Target** and **Product** tables. The **Target** table (bottom left) has the following data:

TargetYear	Category	TargetQuantity
01/01/2019	Clothing	10
01/01/2019	Accessories	20
01/01/2020	Clothing	30
01/01/2020	Accessories	40

The **Product** table (top right) has the following data:

ProductID	Product	Category	Color
1	T-shirt	Clothing	Green
2	Jeans	Clothing	Blue
3	Hat	Accessories	Blue

A line connects the **Target** table to the **Product** table, with a '1' near the **Product** table and a '*' near the **Target** table, indicating a one-to-many relationship.

Na tabela **Target**, existem quatro linhas: duas linhas para cada ano de destino (2019 e 2020) e duas categorias (Clothing e Accessories). Na tabela **Product**, existem três produtos. Dois pertencem à categoria de vestuário e um pertence à categoria de acessórios. Uma das cores do vestuário é verde e as restantes duas são azuis.

Um agrupamento do elemento visual da tabela pela coluna **Category** da tabela **Product** produz o seguinte resultado.

Category	TargetQuantity
Accessories	60
Clothing	40
Total	100

O elemento visual produz o resultado correto. Vamos agora considerar o que acontece quando a coluna **Color** da tabela **Product** é utilizada para agrupar uma quantidade de destino.

Color	TargetQuantity
Blue	100
Green	40
Total	100

O elemento visual produz uma deturpação dos dados. O que está a acontecer aqui?

Um filtro na coluna **Color** da tabela **Product** resulta em duas linhas. Uma das linhas é para a categoria Clothing e a outra é para a categoria Accessories. Estes dois valores de categoria são propagados como filtros para a tabela **Target**. Dito de outra forma, como a cor azul é utilizada pelos produtos de duas categorias, *essas categorias* são utilizadas para filtrar os destinos.

Para evitar este comportamento, conforme descrito anteriormente, recomendamos que controle o resumo dos seus dados de factos ao utilizar medidas.

Considere a seguinte definição de medida. Observe que todas as colunas da tabela **Product** que estão abaixo do nível da categoria são testadas para filtros.

DAX

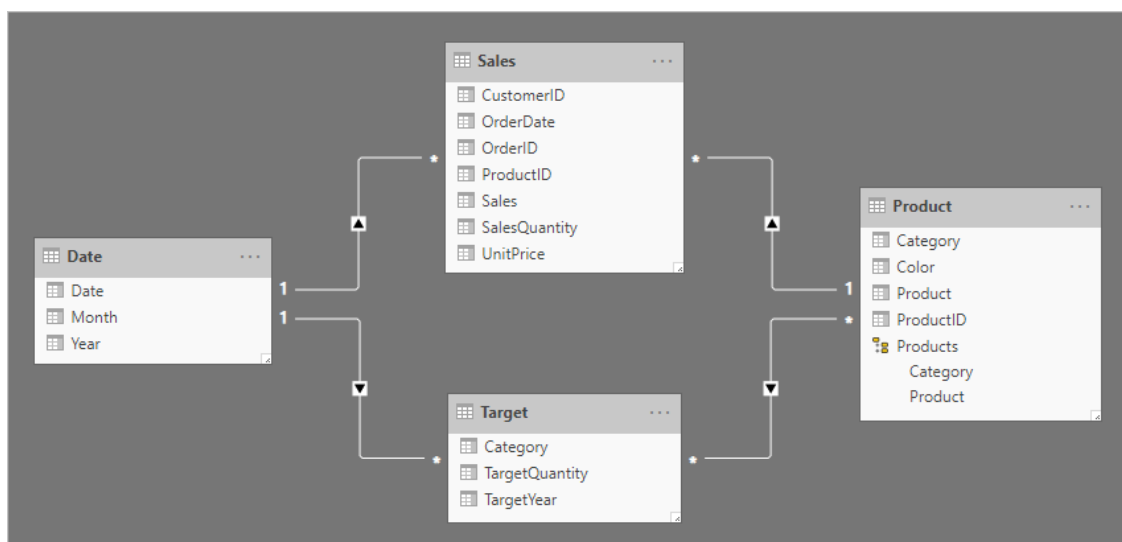
Target Quantity =

```
IF (
    NOT ISFILTERED('Product'[ProductID])
    && NOT ISFILTERED('Product'[Product])
    && NOT ISFILTERED('Product'[Color]),
    SUM(Target[TargetQuantity])
)
```

O seguinte elemento visual de tabela utiliza agora a medida **Target Quantity**. Mostra que todas as quantidades de destino de cores apresentam BLANK.

Color Targets?	
Color	Target Quantity
Blue	
Green	
Total	100

O design do modelo final é semelhante ao seguinte.



Orientação para relacionar factos de agregação superior

Quando precisar de relacionar uma tabela de dimensão a uma tabela de factos e esta última armazenar linhas a uma agregação superior que as linhas da tabela de dimensão, fornecemos a seguinte orientação:

- Para datas de factos de agregação superior:
 - Na tabela de factos, armazene a primeira data do período de tempo
 - Crie uma relação um-para-muitos entre a tabela de data e a tabela de factos
- Para outros factos de agregação superior:
 - Criar uma relação muitos-para-muitos entre a tabela de dimensão e a tabela de factos
- Para ambos os tipos:
 - Controlar o resumo com uma lógica de medida – devolver BLANK quando as colunas de tipo de dimensão de nível inferior forem utilizadas para filtrar ou agrupar
 - Ocultar colunas de tabela de factos resumíveis – deste modo, apenas as medidas podem ser utilizadas para resumir a tabela de factos

