

Advanced Power BI Modeling Techniques

by Martin Schoombee

About Me

“I help people make sense
of their data”

My blog: martinschoombee.com

My company: 28twelve.consulting

Tweet me: [@sqlmartin](https://twitter.com/sqlmartin)

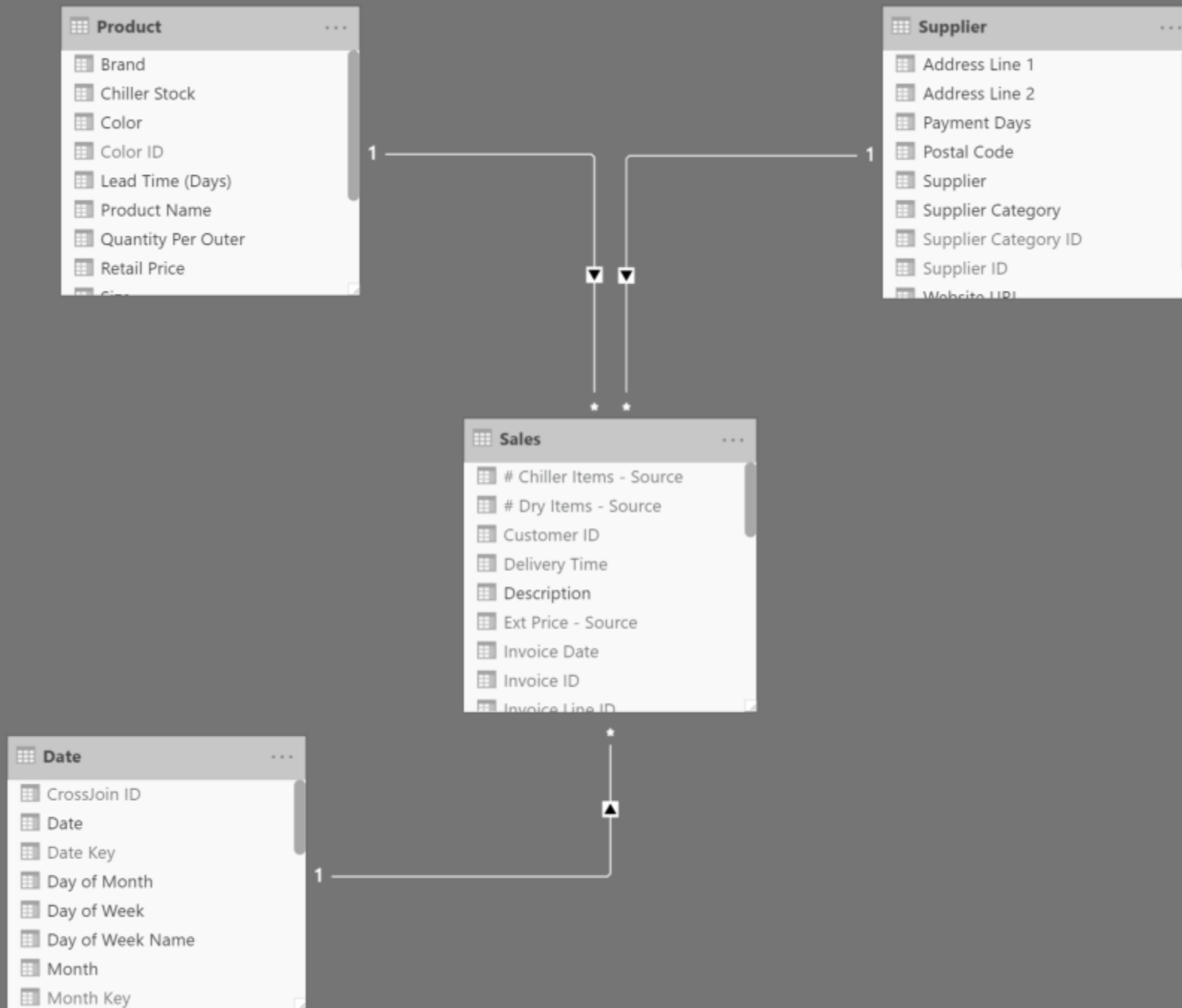


Agenda

- Measure Tables
- Dynamic Currency Conversions
- Role-Playing Dimensions

Measure Table?





Supplier Category	Qty	Ext Price	Profit
Clothing Supplier	2,624,338	\$53,407,430.60	\$21,929,432.50
Novelty Goods Supplier	525,778	\$11,849,641.64	\$6,071,706.15
Packaging Supplier	5,696,347	\$115,763,819.96	\$51,663,616.75
Toy Supplier	104,165	\$17,022,547.25	\$6,064,425.50
Total	8,950,628	\$198,043,439.45	\$85,729,180.90

Measure Name	Clothing Supplier	Novelty Goods Supplier	Packaging Supplier	Toy Supplier	Total
Qty	2,624,338	525,778	5,696,347	104,165	8,950,628
Ext Price	53,407,431	11,849,642	115,763,820	17,022,547	198,043,439
Profit	21,929,433	6,071,706	51,663,617	6,064,426	85,729,181

Clothing Supplier			Novelty Goods Supplier		
Qty	Ext Price	Profit	Qty	Ext Price	Profit
2,624,338	\$53,407,430.60	\$21,929,432.50	525,778	\$11,849,641.64	\$6,071,706.15

Step 1

Create a calculated table

ROW

DAX Function (Table manipulation)

[Syntax](#) | [Return values](#) | [Examples](#) | [Articles](#)

Returns a single row table with new columns specified by the DAX expressions.

Syntax

```
ROW ( <Name>, <Expression> [, <Name>, <Expression> [, ... ] ] )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
Name	Repeatable	Name of the new column.
Expression	Repeatable	The expression for the column.

Return values

TABLE An entire table or a table with one or more columns.

A single row table.

UNION

DAX Function (Table manipulation)

[Syntax](#) | [Return values](#) | [Remarks](#) | [Related](#)

Returns the union of the two tables whose columns match.

Syntax

```
UNION ( <Table> [, <Table> [, ... ] ] )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
Table	Repeatable	A table that will participate in the crossjoin union.

Return values

TABLE

An entire table or a table with one or more columns.

A table that contains all the rows from each of the table expressions.

My Measures =

UNION

```
(  
    ROW("Measure Name", "Qty", "Sort Order", 1)  
,  
    ROW("Measure Name", "Ext Price", "Sort Order", 2)  
,  
    ROW("Measure Name", "Profit", "Sort Order", 3)  
)
```

DATATABLE

DAX Function (Table manipulation)

[Syntax](#) | [Return values](#) | [Remarks](#) | [Examples](#) | [Articles](#)

Returns a table with data defined inline.

Syntax

```
DATATABLE ( <name>, <type> [, <name>, <type> [, ... ] ], <data> )
```

PARAMETER	ATTRIBUTES	DESCRIPTION
name	Repeatable	A column name to be defined.
type	Repeatable	A type name to be associated with the column.
data		The data for the table.

Return values



TABLE

An entire table or a table with one or more columns.

A table declaring an inline set of values.

```
My Measures =  
DATATABLE  
(  
    "Measure Name", STRING  
    , "Sort Order", INTEGER  
    , {  
        {"Qty", 1}  
        , {"Ext Price", 2}  
        , {"Profit", 3}  
    }  
)
```


Table Constructor

12/09/2018 • 2 minutes to read • Contributors  

Returns a table of one or more columns.

Syntax

DAX

 Copy

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ), ... }
```

Parameters

Term	Definition
scalarExprN	Any DAX expression that returns a scalar value.

Return value

A table of one or more columns. When there is only one column, the name of the column is Value. When there are N columns where $N > 1$, the names of the columns from left to right are Value1, Value2, ..., ValueN.

```
My Measures =  
{  
    ("Qty", 1)  
    , ("Ext Price", 2)  
    , ("Profit", 3)  
}
```


Step 2

Define a measure

```
value =  
SWITCH  
(  
    FIRSTNONBLANK('My Measures'[Measure Name])  
    , "Qty", [Qty]  
    , "Ext Price", [Ext Price]  
    , "Profit", [Profit]  
    , BLANK()  
)
```

Supplier Category	Qty	Ext Price	Profit
Clothing Supplier	2,624,338	\$53,407,430.60	\$21,929,432.50
Novelty Goods Supplier	525,778	\$11,849,641.64	\$6,071,706.15
Packaging Supplier	5,696,347	\$115,763,819.96	\$51,663,616.75
Toy Supplier	104,165	\$17,022,547.25	\$6,064,425.50
Total	8,950,628	\$198,043,439.45	\$85,729,180.90

Measure Name	Clothing Supplier	Novelty Goods Supplier	Packaging Supplier	Toy Supplier	Total
Qty	2,624,338	525,778	5,696,347	104,165	8,950,628
Ext Price	53,407,431	11,849,642	115,763,820	17,022,547	198,043,439
Profit	21,929,433	6,071,706	51,663,617	6,064,426	85,729,181

Dynamic Currency Conversion



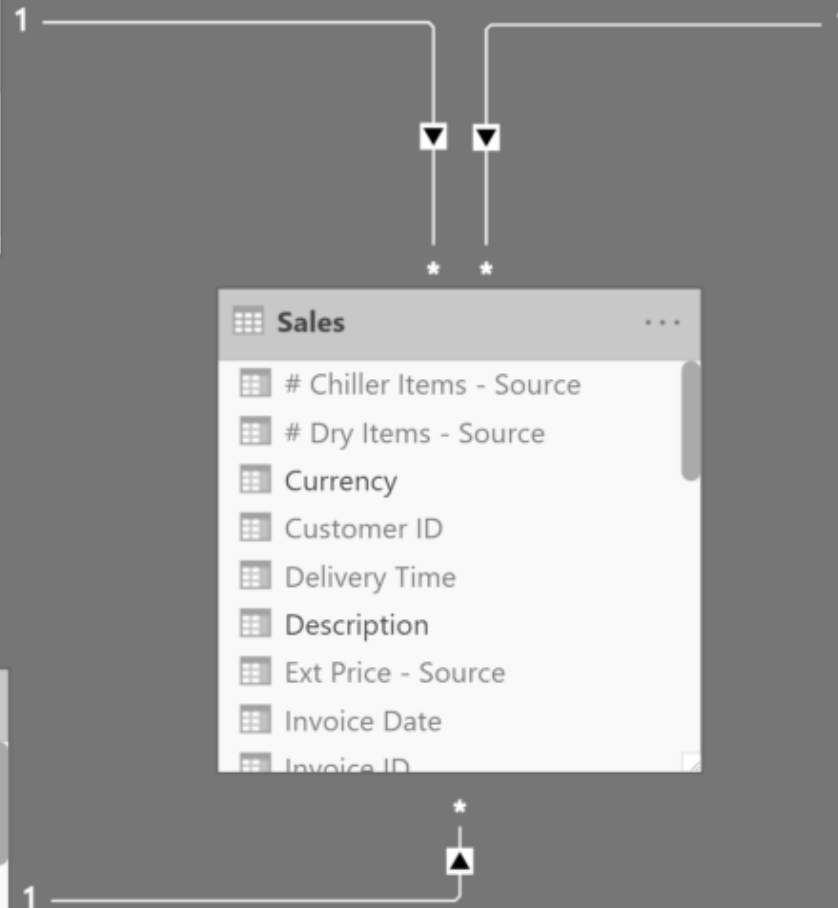
Product
Brand
Chiller Stock
Color
Color ID
Lead Time (Days)
Product Name
Quantity Per Outer
Retail Price
Size

Supplier
Address Line 1
Address Line 2
Payment Days
Postal Code
Supplier
Supplier Category
Supplier Category ID
Supplier ID
Website URL

Sales
Chiller Items - Source
Dry Items - Source
Currency
Customer ID
Delivery Time
Description
Ext Price - Source
Invoice Date
Invoice ID

Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name
Month
Month Key

Currency Exchange Rate
Date
Destination Currency
Exchange Rate
Source Currency

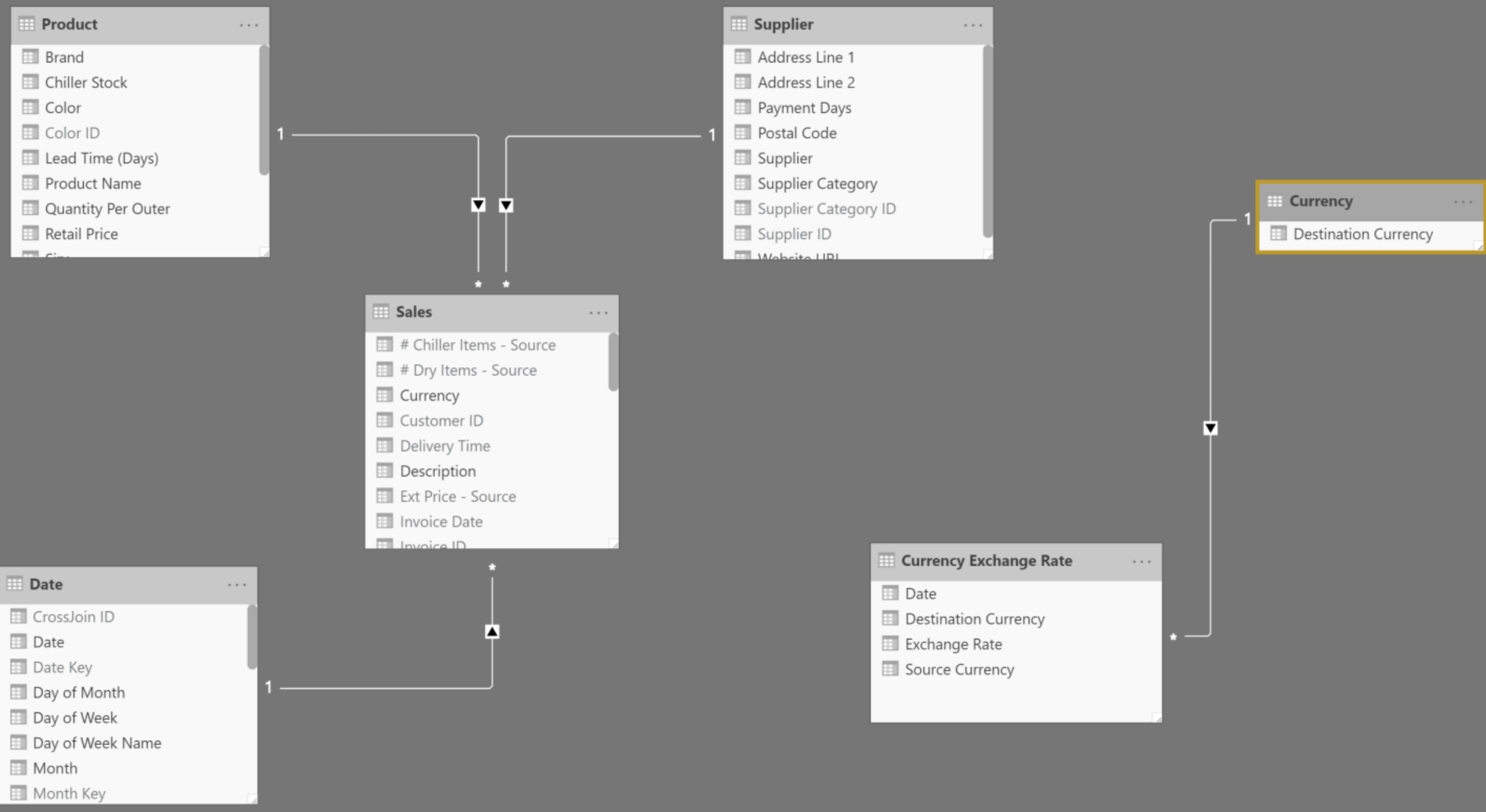


Demo Time



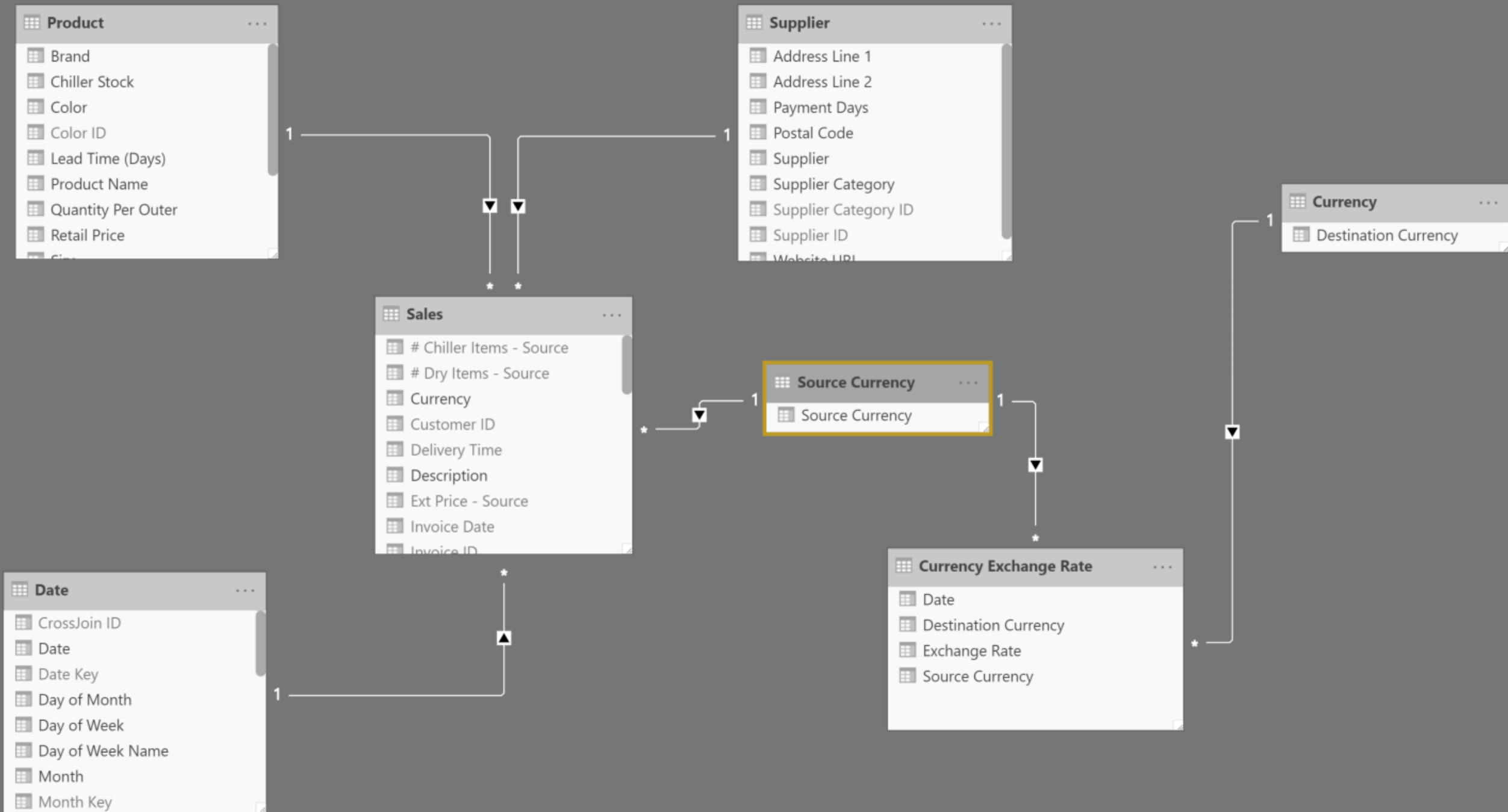
Step 1

Create a filter dimension



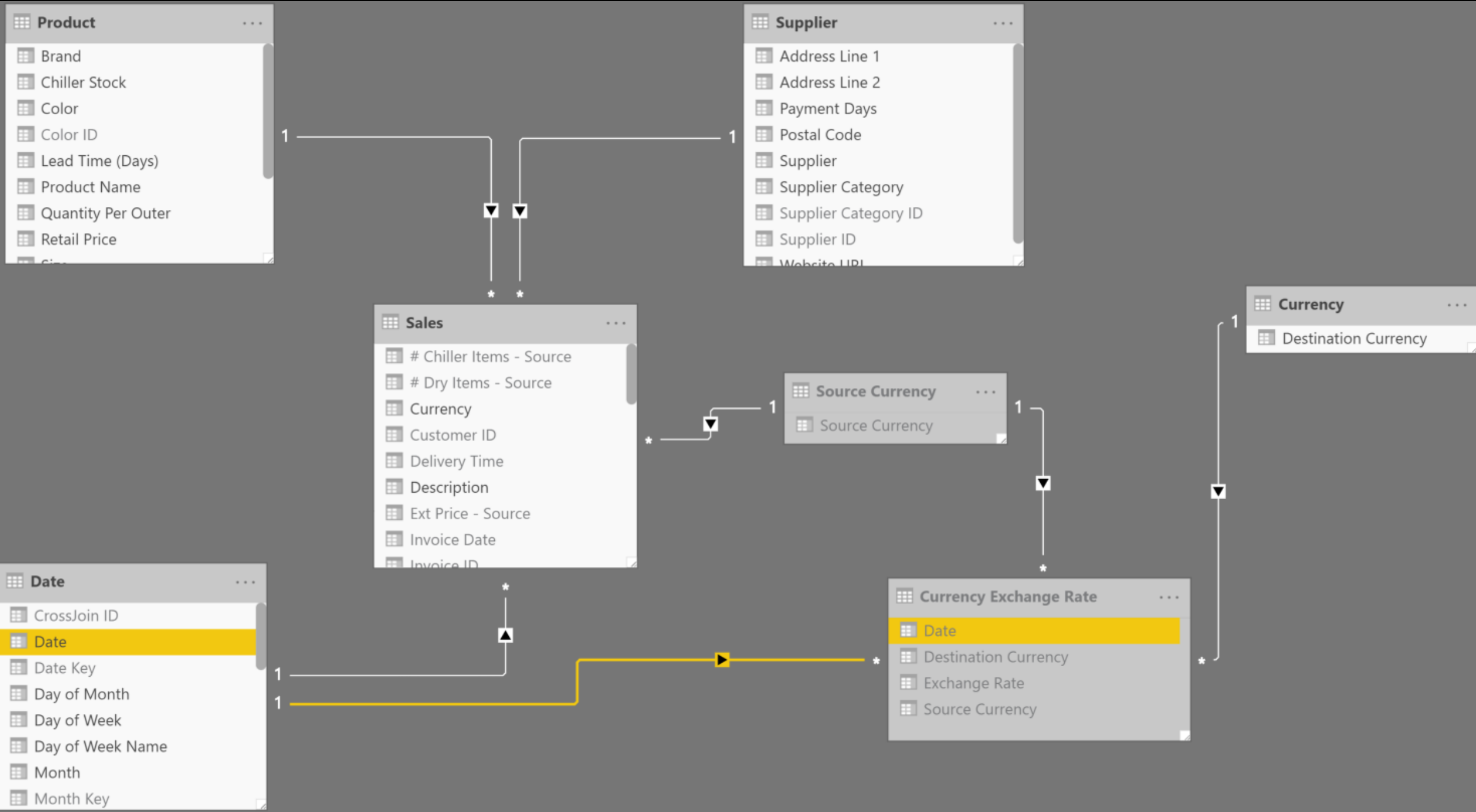
Step 2

Create a bridge table



Step 3

Add a relationship to the
Date entity



Step 4

Create the measures

Exchange Rate = MAX('Currency Exchange Rate'[Exchange Rate])

New Ext Price =

```
IF
(
    HASONEVALUE('Currency'[Destination Currency])
    ,
    SUMX
    (
        Sales
        , [Ext Price] * [Exchange Rate]
    )
    ,
    CALCULATE
    (
        SUMX
        (
            Sales
            , [Ext Price] * [Exchange Rate]
        )
        , 'Currency'[Destination Currency] = "USD"
    )
)
```

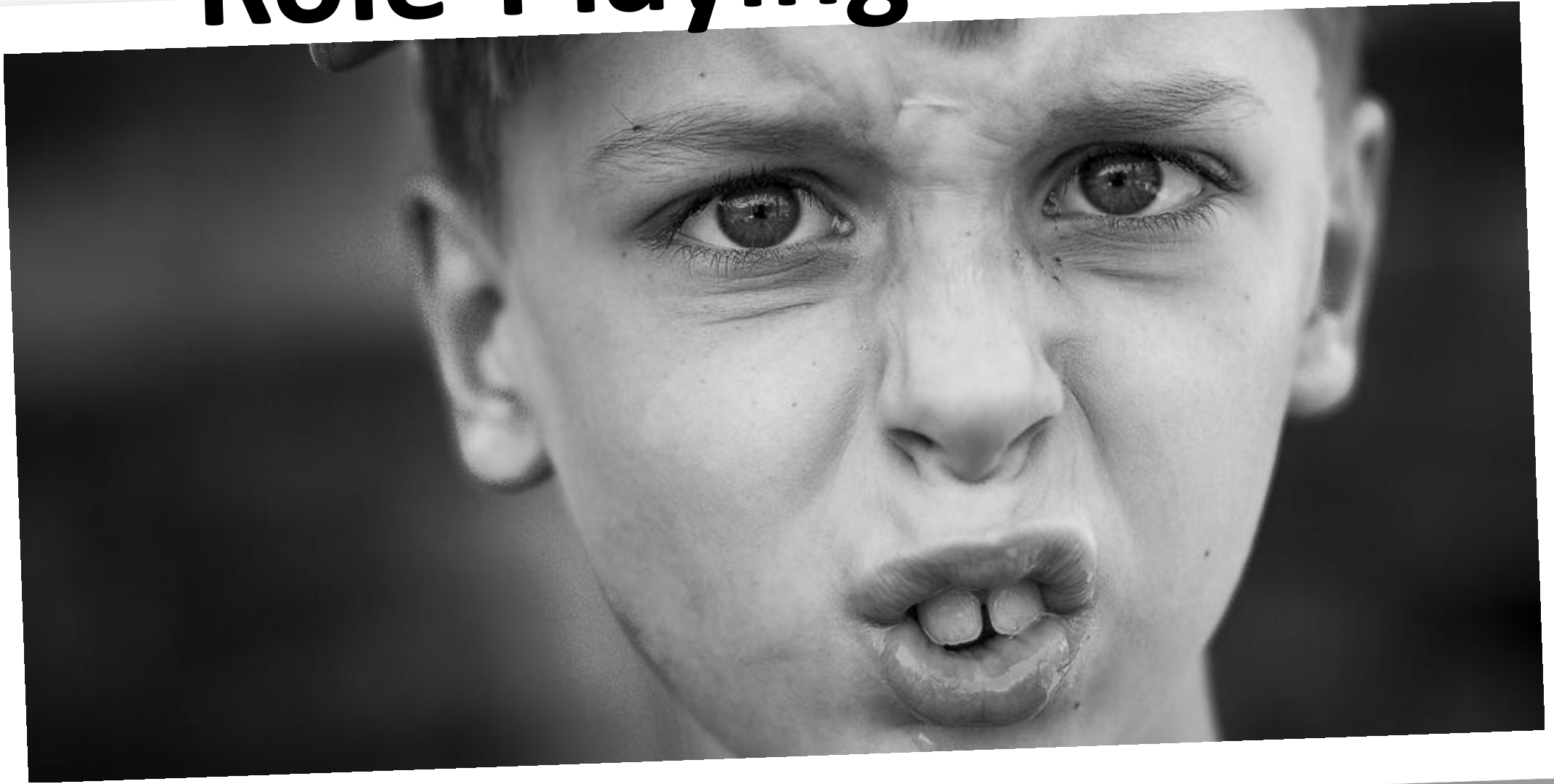
Demo Time



Role-Playing Dimensions



Role-Playing What?

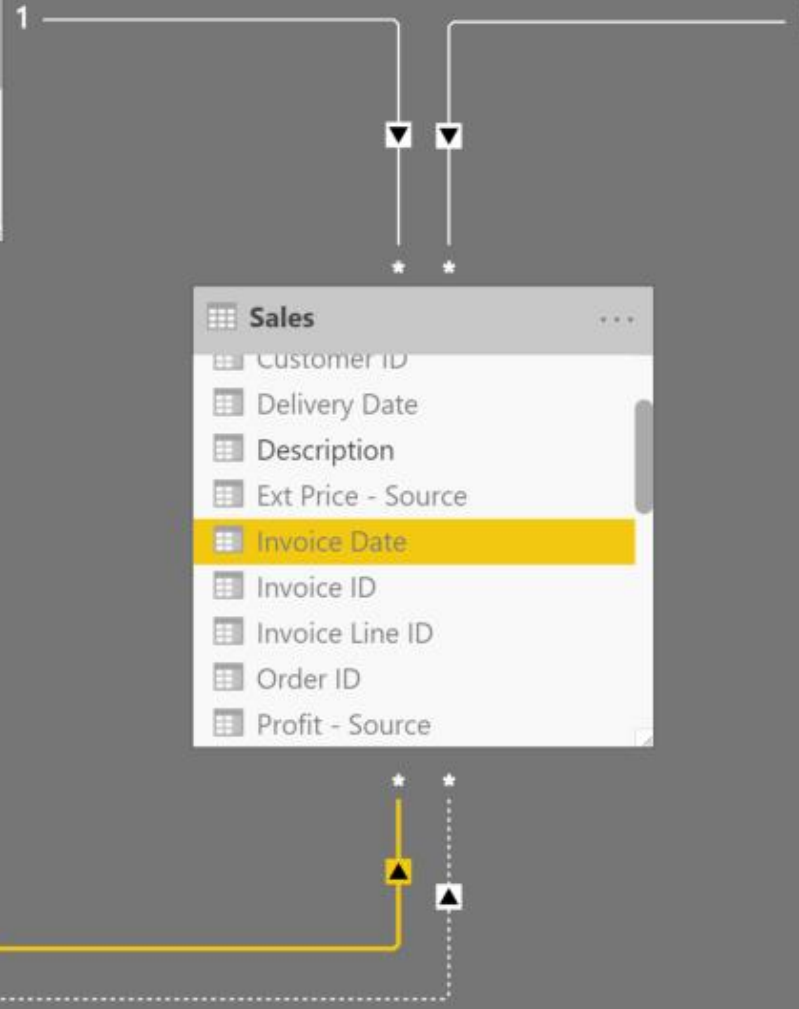


Product
Brand
Chiller Stock
Color
Color ID
Lead Time (Days)
Product Name
Quantity Per Outer
Retail Price
Size

Supplier
Address Line 1
Address Line 2
Payment Days
Postal Code
Supplier
Supplier Category
Supplier Category ID
Supplier ID
Website URL

Sales
Customer ID
Delivery Date
Description
Ext Price - Source
Invoice Date
Invoice ID
Invoice Line ID
Order ID
Profit - Source

Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name
Month
Month Key



Product
Brand
Chiller Stock
Color
Color ID
Lead Time (Days)
Product Name
Quantity Per Outer
Retail Price

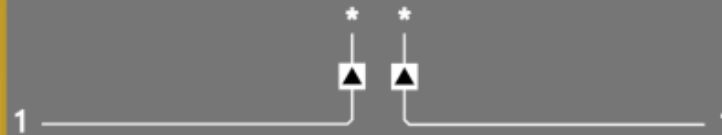
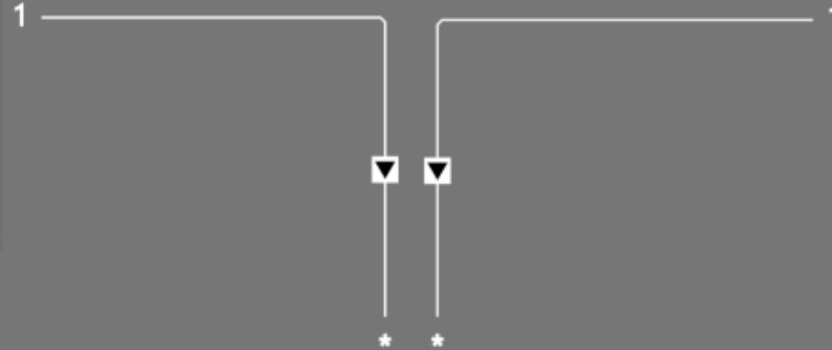
Supplier
Address Line 1
Address Line 2
Payment Days
Postal Code
Supplier
Supplier Category
Supplier Category ID
Supplier ID
Website URL

Sales
Chiller Items - Source
Dry Items - Source
Currency
Customer ID
Delivery Date
Description
Ext Price - Source
Invoice Date
Invoice ID

Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name
Month
Month Key

Delivery Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name

Invoice Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name

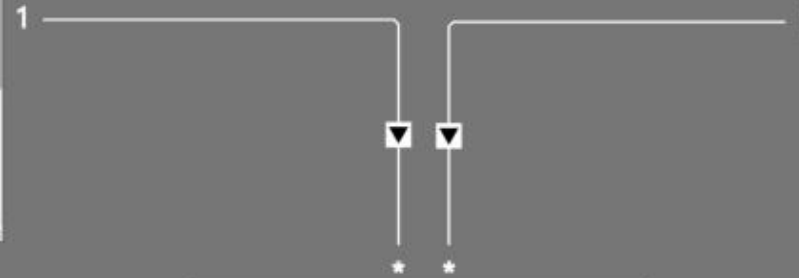


Product
Brand
Chiller Stock
Color
Color ID
Lead Time (Days)
Product Name
Quantity Per Outer
Retail Price
Size

Supplier
Address Line 1
Address Line 2
Payment Days
Postal Code
Supplier
Supplier Category
Supplier Category ID
Supplier ID
Website URL

Sales
Customer ID
Delivery Date
Description
Ext Price - Source
Invoice Date
Invoice ID
Invoice Line ID
Order ID
Profit - Source

Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name
Month
Month Key



```
Ext Price by Delivery Date =  
CALCULATE  
(  
    [Ext Price]  
    , USERRELATIONSHIP(Sales[Delivery Date], 'Date'[Date])  
)
```

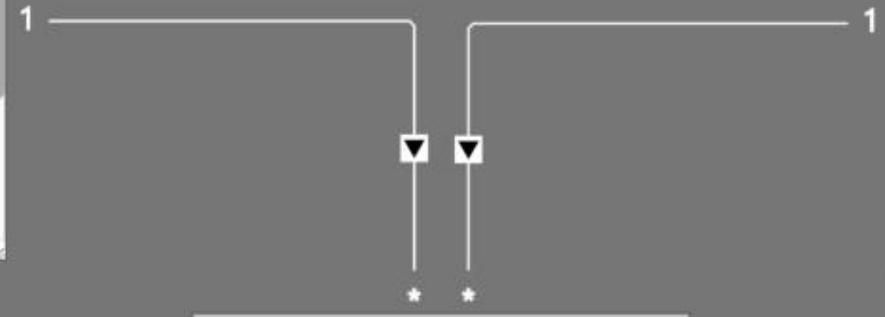
```
Ext Price by Invoice Date =  
CALCULATE  
(  
    [Ext Price]  
    , USERRELATIONSHIP(Sales[Invoice Date], 'Date'[Date])  
)
```

Step 1

Create a filter dimension

Product
Brand
Chiller Stock
Color
Color ID
Lead Time (Days)
Product Name
Quantity Per Outer
Retail Price
Size

Supplier
Address Line 1
Address Line 2
Payment Days
Postal Code
Supplier
Supplier Category
Supplier Category ID
Supplier ID
Website URL



Date Filter
Date To Filter

Sales
Chiller Items - Source
Dry Items - Source
Currency
Customer ID
Delivery Date
Description
Ext Price - Source
Invoice Date
Invoice ID

Date
CrossJoin ID
Date
Date Key
Day of Month
Day of Week
Day of Week Name
Month
Month Key



Step 2

Create a measure that's
“Date aware”


```

Ext Price - Date Aware =
VAR
    SelectedDate = SELECTEDVALUE('Date Filter'[Date To Filter], "Invoice Date")
VAR
    UseInvoiceDate = CALCULATE
        (
            [Ext Price]
            , USERRELATIONSHIP(Sales[Invoice Date], 'Date'[Date])
        )
VAR
    UseDeliveryDate = CALCULATE
        (
            [Ext Price]
            , USERRELATIONSHIP(Sales[Delivery Date], 'Date'[Date])
        )
RETURN
    SWITCH
        (
            SelectedDate
            , "Invoice Date"
            , UseInvoiceDate
            , "Delivery Date"
            , UseDeliveryDate
            , UseInvoiceDate
        )

```

Demo Time



About Me

“I help people make sense
of their data”

My blog: martinschoombee.com

My company: 28twelve.consulting

Tweet me: [@sqlmartin](https://twitter.com/sqlmartin)

