



Level-up your Power BI
data modeling skills





Marc Lelijveld

Solution Architect
Macaw Netherlands

 @MarcLelijveld

 linkedin.com/in/MarcLelijveld

 Data-Marc.com



Jeroen ter Heerdt

Senior Program Manager
Microsoft



 @JayPowerBI

 linkedin.com/in/jeroenterheerdt

 DutchDataDude.com

Agenda

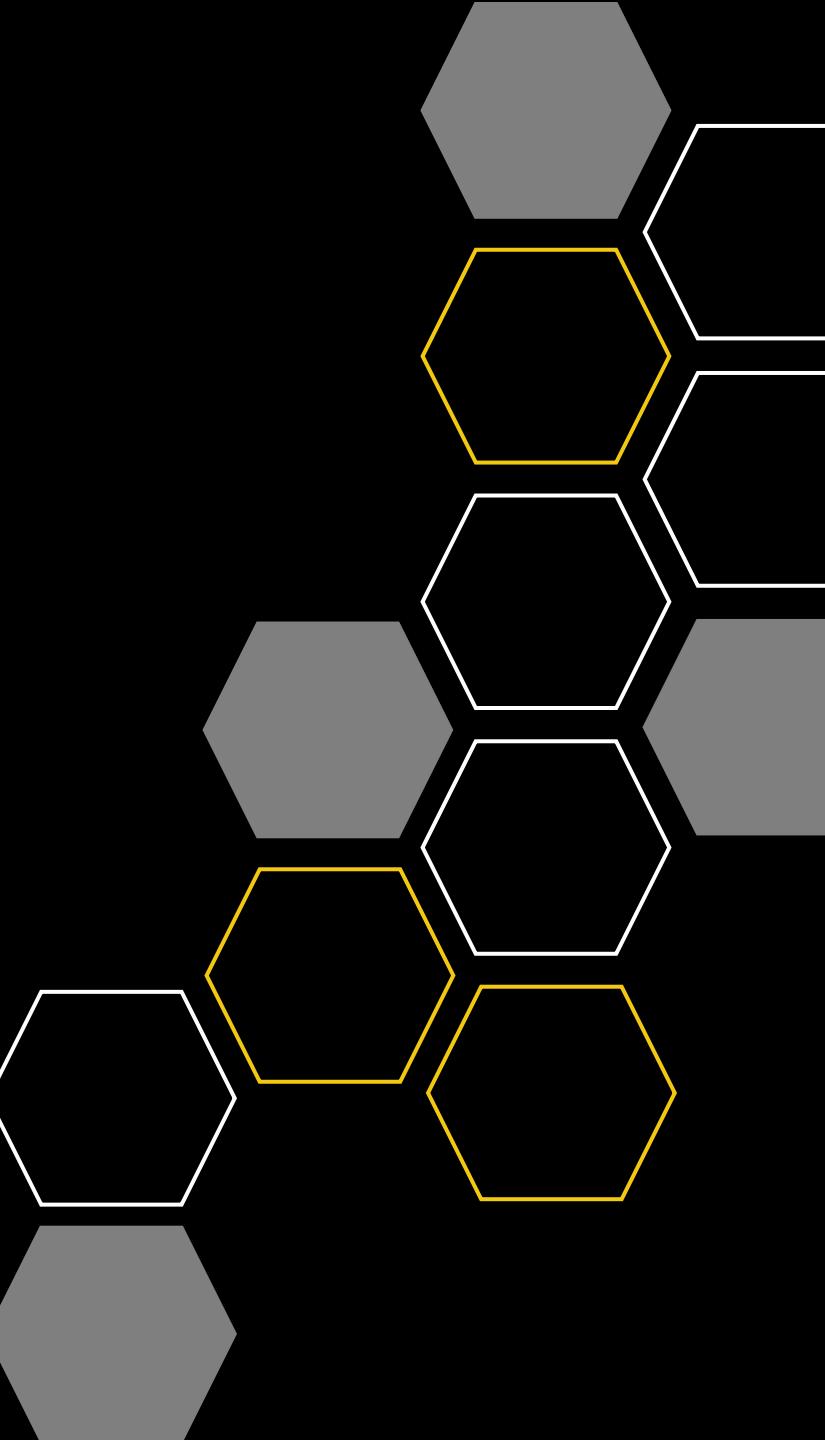
- Data modeling 101
 - What is a good data model
 - Building a good data model
 - Role playing dimensions
 - Type of data models
 - Enriching your data
- Build your own data model (**hands-on**)

 << lunch break >>
- Relationships revisited
- Handling multiple facts
- Fact types
- Extend your own data model (**hands-on**)
- Optional topics

Optional topics

- General best practices
 - Naming conventions
 - Auto date/time
- Slowly changing dimensions
- RLS / OLS
- Aggregations
- Composite models
 - Wholesale / Retail
- Storage Modes
- Regular vs Limited relationship
- Calc groups
- Ambiguity
 - Over filtering
- Data groups & hierarchies

Data modeling 101



After this session you will be able to:

-  Explain basic dimensional modeling concepts
-  Create a basic dimensional model that works and performs well with Power BI
-  Decide when to use Power Query vs the data model and Calculated Columns vs Measures
-  Handle complex situations with inactive relationships and role-playing dimensions



What are we talking about?

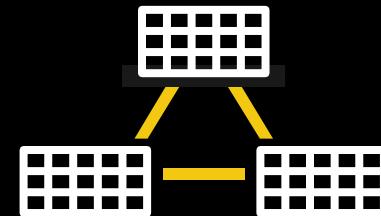
Gather



Clean



Model



Visualize



What is a data model?

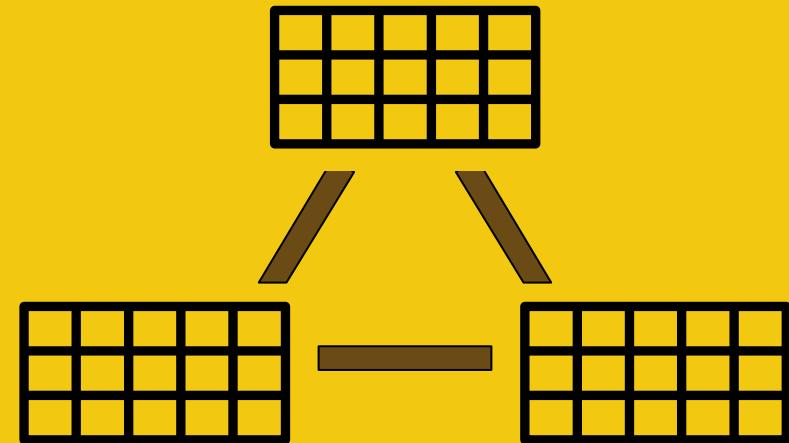
- A **Power BI Data Model** is a collection of **tables with relationships** which enable business users to easily understand and explore their data to get business insights.
- A good data model is important, because:
 - Improves understandability of the data
 - Increases performance of dependent processes and systems
 - Increases resilience to change

Why do you need a data model in Power BI?

- Easily drag-and-drop visuals
- Re-use of components (when expanding the model)
- Optimize end-user rendering performance

What is a **good** data model?

- A good data model for Power BI is a **dimensional data model**.
- A **dimensional data model** consists of:
 - One or more fact tables
 - One or more dimension tables
 - One or more relationships



Building a dimensional data model

Starting point



INVOICE

INVOICE SO74869

ORDER DATE: 6/11/2020

SHIP DATE:

Adventure Works

TO:

AW00019377
Dominic Chandra
Ingolstadt
85049 Germany

QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
1	AWC Logo Cap (CA-1098)	\$8.99	\$8.99
1	Fender Set – Mountain (FE-6654)	\$21.98	\$21.98
1	ML Mountain Tire (TI-M602)	\$29.99	\$29.99
1	Mountain Bottle Cage (BC-M005)	\$9.99	\$9.99
1	Mountain Tire Tube (TT-M928)	\$4.99	\$4.99
1	Sport-100 <u>Helmet</u> , Blue (HL-U509-B)	\$34.99	\$34.99
1	Water Bottle - 30 oz. (WB-H098)	\$4.99	\$4.99
TOTAL DUE			\$115.92

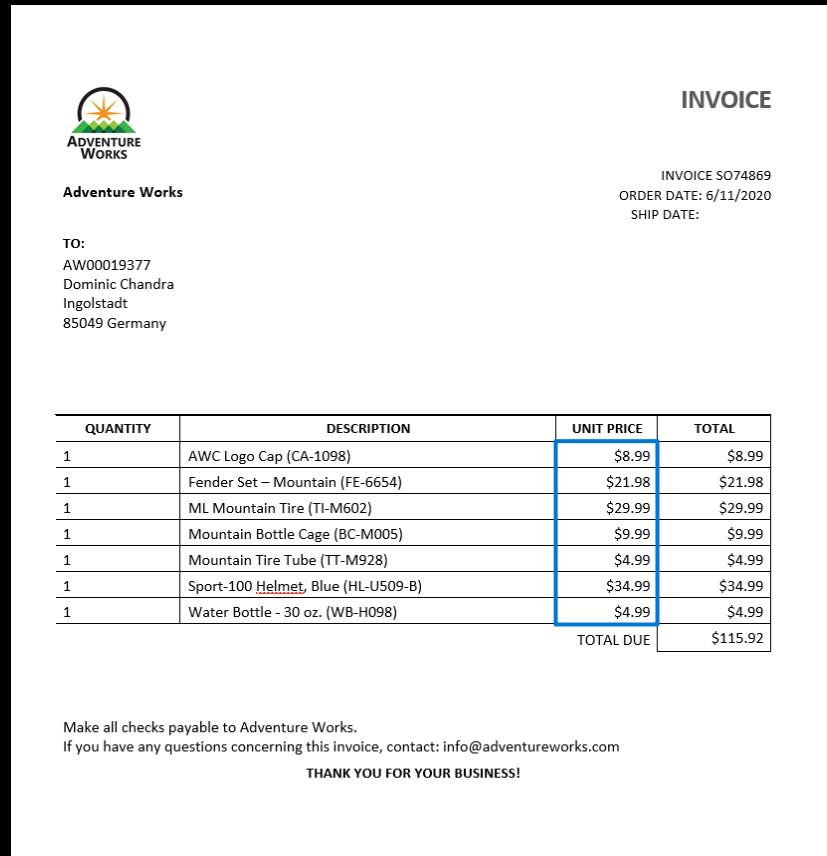
Make all checks payable to Adventure Works.

If you have any questions concerning this invoice, contact: info@adventureworks.com

THANK YOU FOR YOUR BUSINESS!

Building a dimensional data model

Creating Sales Amount fact

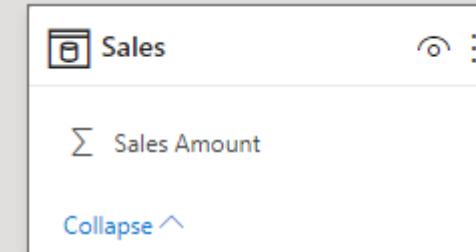


Tables

Sales

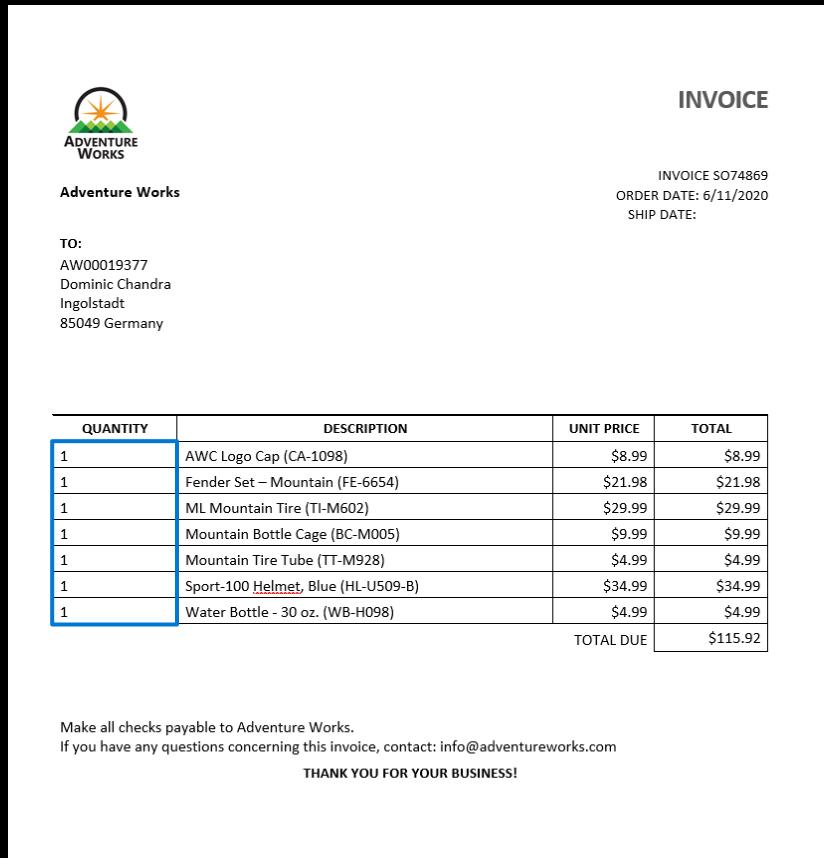
Sales Amount
\$8.99
\$21.98
\$29.99
\$9.99
\$4.99
\$34.99
\$4.99

Diagram



Building a dimensional data model

Adding Order Quantity fact

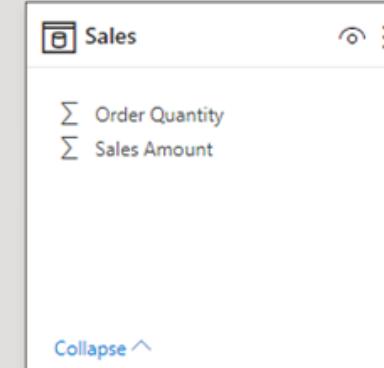


Tables

Sales

Sales Amount	Order Quantity
\$8.99	1
\$21.98	1
\$29.99	1
\$9.99	1
\$4.99	1
\$34.99	1
\$4.99	1

Diagram



Fact table

- Contains **numerical information** about a business process or items to be aggregated
- Examples: Sales Amount, Cost, Units Sold, Transactions
- Aggregations provide totals, averages, etc.
 - Power BI implements these using **Measures**
- Usefulness limited without context
 - Context is provided by **dimensions** that slice the data
- ***Without facts there is no aggregation***



Building a dimensional data model

Adding Product dimension table

INVOICE																																							
Adventure Works																																							
INVOICE SO74869 ORDER DATE: 6/11/2020 SHIP DATE:																																							
TO: AW00019377 Dominic Chandra Ingolstadt 85049 Germany																																							
<table border="1"><thead><tr><th>QUANTITY</th><th>DESCRIPTION</th><th>UNIT PRICE</th><th>TOTAL</th></tr></thead><tbody><tr><td>1</td><td>AWC Logo Cap (CA-1098)</td><td>\$8.99</td><td>\$8.99</td></tr><tr><td>1</td><td>Fender Set – Mountain (FE-6654)</td><td>\$21.98</td><td>\$21.98</td></tr><tr><td>1</td><td>ML Mountain Tire (TI-M602)</td><td>\$29.99</td><td>\$29.99</td></tr><tr><td>1</td><td>Mountain Bottle Cage (BC-M005)</td><td>\$9.99</td><td>\$9.99</td></tr><tr><td>1</td><td>Mountain Tire Tube (TT-M928)</td><td>\$4.99</td><td>\$4.99</td></tr><tr><td>1</td><td>Sport-100 Helmet, Blue (HL-U509-B)</td><td>\$34.99</td><td>\$34.99</td></tr><tr><td>1</td><td>Water Bottle - 30 oz. (WB-H098)</td><td>\$4.99</td><td>\$4.99</td></tr><tr><td colspan="2">TOTAL DUE</td><td></td><td>\$115.92</td></tr></tbody></table>				QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL	1	AWC Logo Cap (CA-1098)	\$8.99	\$8.99	1	Fender Set – Mountain (FE-6654)	\$21.98	\$21.98	1	ML Mountain Tire (TI-M602)	\$29.99	\$29.99	1	Mountain Bottle Cage (BC-M005)	\$9.99	\$9.99	1	Mountain Tire Tube (TT-M928)	\$4.99	\$4.99	1	Sport-100 Helmet, Blue (HL-U509-B)	\$34.99	\$34.99	1	Water Bottle - 30 oz. (WB-H098)	\$4.99	\$4.99	TOTAL DUE			\$115.92
QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL																																				
1	AWC Logo Cap (CA-1098)	\$8.99	\$8.99																																				
1	Fender Set – Mountain (FE-6654)	\$21.98	\$21.98																																				
1	ML Mountain Tire (TI-M602)	\$29.99	\$29.99																																				
1	Mountain Bottle Cage (BC-M005)	\$9.99	\$9.99																																				
1	Mountain Tire Tube (TT-M928)	\$4.99	\$4.99																																				
1	Sport-100 Helmet, Blue (HL-U509-B)	\$34.99	\$34.99																																				
1	Water Bottle - 30 oz. (WB-H098)	\$4.99	\$4.99																																				
TOTAL DUE			\$115.92																																				
Make all checks payable to Adventure Works. If you have any questions concerning this invoice, contact: info@adventureworks.com																																							
THANK YOU FOR YOUR BUSINESS!																																							

Tables

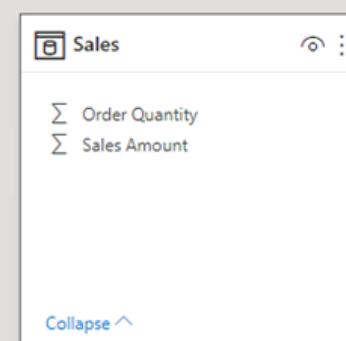
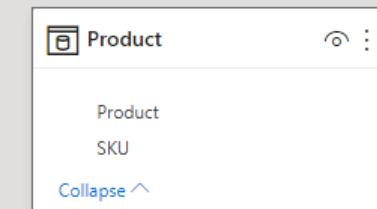
Product

SKU	Product
CA-1098	AWC Logo Cap
FE-6654	Fender Set - Mountain
TI-M602	ML Mountain Tire
BC-M005	Mountain Bottle Cage
TT-M928	Mountain Tire Tube
HL-U509-B	Sport-100, Helmet, Blue
WB-H098	Water Bottle - 30 oz.

Sales

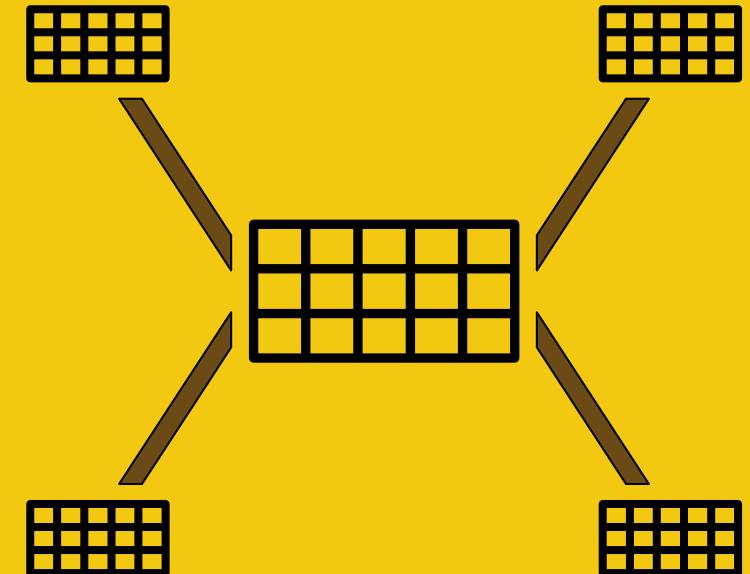
Sales Amount	Order Quantity
\$8.99	1
\$21.98	1
\$29.99	1
\$9.99	1
\$4.99	1
\$34.99	1
\$4.99	1

Diagram



Dimension table

- Contains **descriptive information** that define how a fact should roll up.
- Examples: Date, Month, Customer, Geography, Product, Payment type.
- Without dimensions there is no context.
- Also called: **Lookup table on steroids**.



Building a dimensional data model

Adding Product attribute to fact table

 ADVENTURE WORKS

Adventure Works

TO:
AW00019377
Dominic Chandra
Ingolstadt
85049 Germany

INVOICE

INVOICE SO74869
ORDER DATE: 6/11/2020
SHIP DATE:

QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
1	AWC Logo Cap (CA-1098)	\$8.99	\$8.99
1	Fender Set – Mountain (FE-6654)	\$21.98	\$21.98
1	ML Mountain Tire (TI-M602)	\$29.99	\$29.99
1	Mountain Bottle Cage (BC-M005)	\$9.99	\$9.99
1	Mountain Tire Tube (TT-M928)	\$4.99	\$4.99
1	Sport-100 Helmet, Blue (HL-U509-B)	\$34.99	\$34.99
1	Water Bottle - 30 oz. (WB-H098)	\$4.99	\$4.99
	TOTAL DUE		\$115.92

Make all checks payable to Adventure Works.
If you have any questions concerning this invoice, contact: info@adventureworks.com

THANK YOU FOR YOUR BUSINESS!

Tables

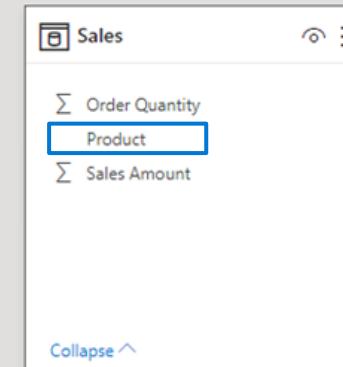
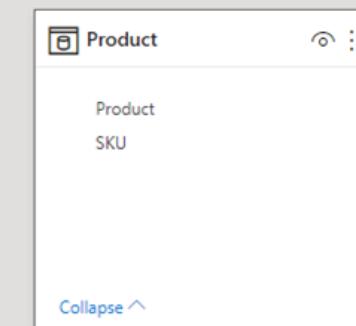
Product

SKU	Product
CA-1098	AWC Logo Cap
FE-6654	Fender Set - Mountain
TI-M602	ML Mountain Tire
BC-M005	Mountain Bottle Cage
TT-M928	Mountain Tire Tube
HL-U509-B	Sport-100, Helmet, Blue
WB-H098	Water Bottle - 30 oz.

Sales

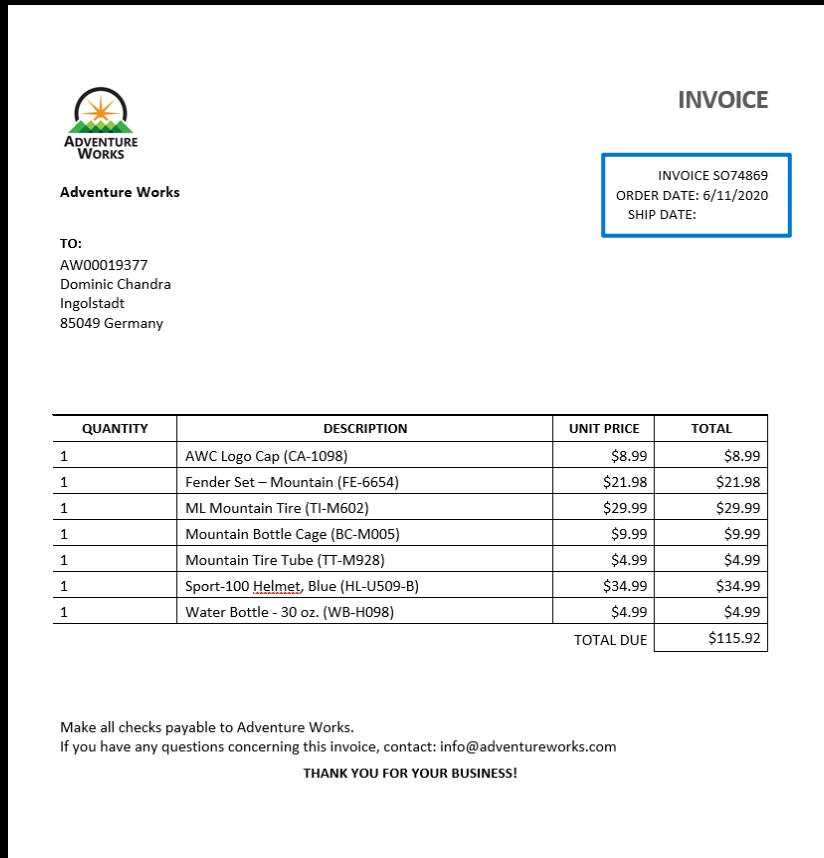
Sales Amount	Order Quantity	Product
\$8.99	1	CA-1098
\$21.98	1	FE-6654
\$29.99	1	TI-M602
\$9.99	1	BC-M005
\$4.99	1	TT-M928
\$34.99	1	HL-U509-B
\$4.99	1	WB-H098

Diagram



Building a dimensional data model

Adding Sales Order and Order Date attribute to fact



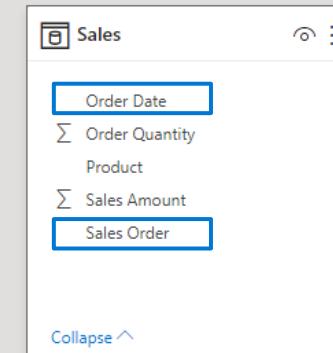
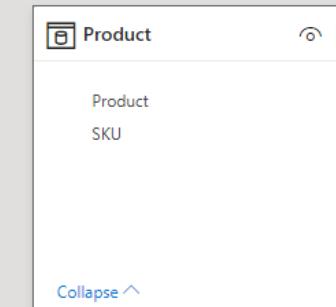
Tables

Sales

Sales Amount	Order Quantity	Product	Sales Order	Order Date
\$8.99	1	CA-1098	SO74869	6/11/2020
\$21.98	1	FE-6654	SO74869	6/11/2020
\$29.99	1	TI-M602	SO74869	6/11/2020
\$9.99	1	BC-M005	SO74869	6/11/2020
\$4.99	1	TT-M928	SO74869	6/11/2020
\$34.99	1	HL-U509-B	SO74869	6/11/2020
\$4.99	1	WB-H098	SO74869	6/11/2020
\$8.99	2	CA-1098	SO74870	7/11/2020

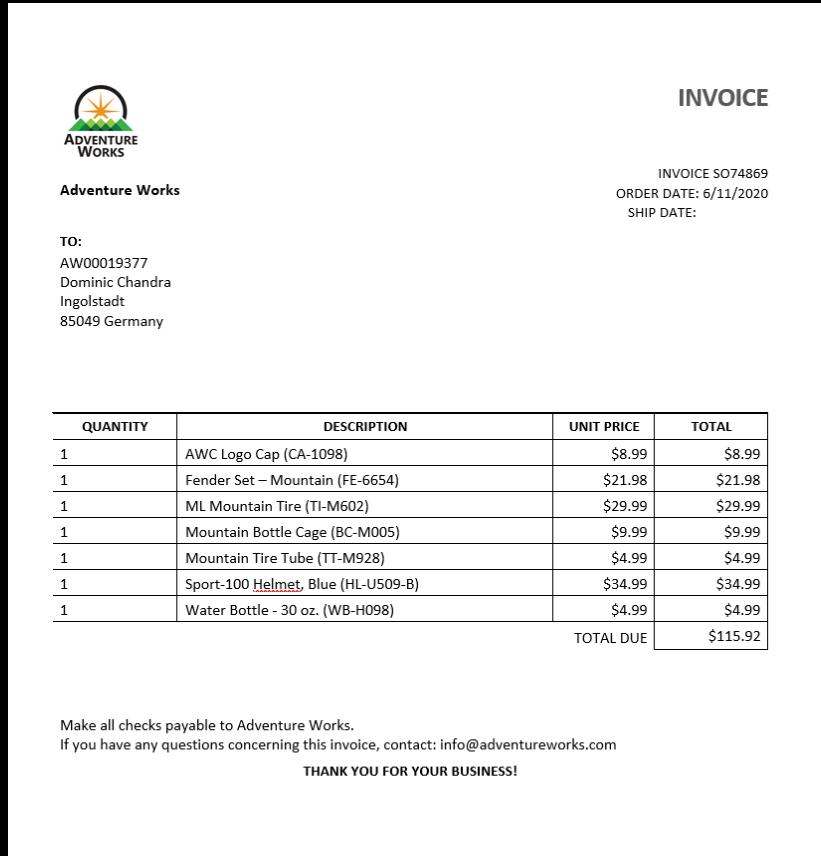
Product

Diagram



Building a dimensional data model

Adding relationship between Product and Sales



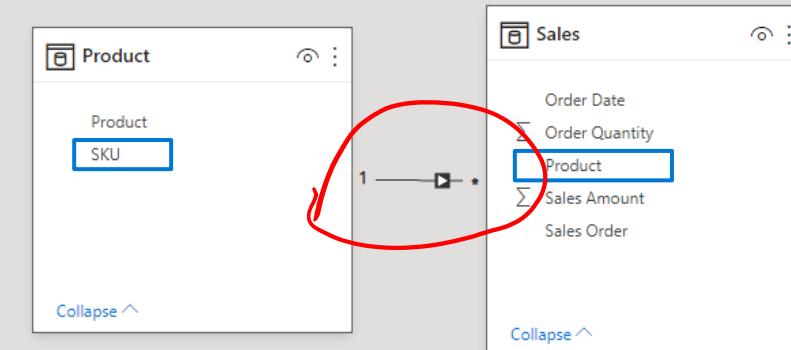
Tables

Sales

Sales Amount	Order Quantity	Product	Sales Order	Order Date
\$8.99	1	CA-1098	SO74869	6/11/2020
\$21.98	1	FE-6654	SO74869	6/11/2020
\$29.99	1	TI-M602	SO74869	6/11/2020
\$9.99	1	BC-M005	SO74869	6/11/2020
\$4.99	1	TT-M928	SO74869	6/11/2020
\$34.99	1	HL-U509-B	SO74869	6/11/2020
\$4.99	1	WB-H098	SO74869	6/11/2020
\$8.99	2	CA-1098	SO74870	7/11/2020

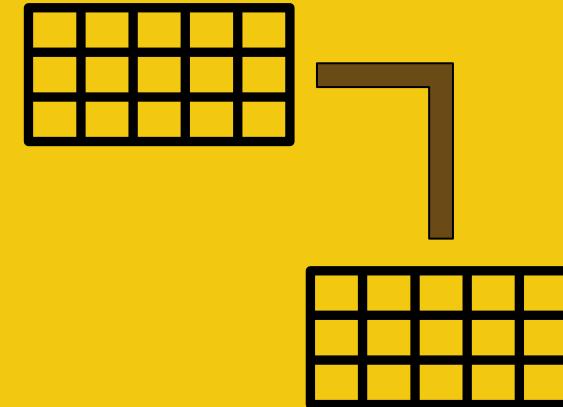
Product

Diagram



Relationship

- Connection between two tables using columns from each.
- Usually between fact and dimension or dimension and dimension.
- *Without relationships there is no slicing.*
- Three kinds:
 - One-to-many (1:M)
 - One-to-one (1:1)
 - Many-to-many (M:N)



Relation types

There are **three types of relationships** in data modeling

1 to 1

Every **individual record** in dataset A is mapped to one **individual record** in dataset B

Dataset A Customer information		Dataset B Customer information	
Customer ID	Country	Customer ID	Date of Birth
AW000111024	United States	AW000111024	9 April 1990
AW00019377	Germany	AW00019377	9 April 1983

1 to many

One record of dataset A is mapped to **multiple records** in dataset B

Dataset A Customer information		Dataset B Sales Information		
Customer ID	Country	Customer ID	Product	Order Quantity
AW000111024	United States	AW000111024	CA-1098	2
AW00019377	Germany	AW00019377	BC-M005	1
		AW00019377	CA-1098	1
		AW00019377	FE-6654	1
		AW00019377	HL-U509-B	1
		AW00019377	TI-M602	1
		AW00019377	TT-M928	1
		AW00019377	WB-H098	1

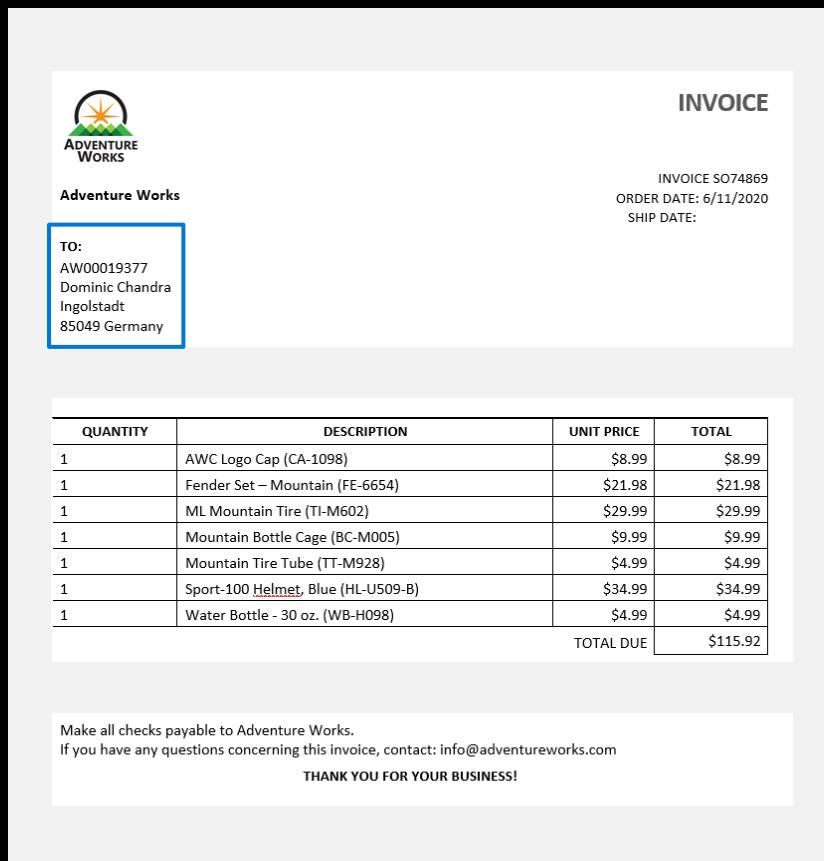
Many to many

Multiple records of dataset A are mapped to **multiple records** in dataset B

Dataset A Customer information		Dataset B Sales Information		
Customer ID	Store ID	Store Type	Customer ID	Product
AW000111024	ON-1	Online	AW000111024	CA-1098
AW000111024	ST-1	Regional City Store	AW000111024	TT-M928
AW000111024	ST-2	Regional City Store	AW00019377	BC-M005
AW00019377	ON-1	Online	AW00019377	CA-1098
AW00019377	ST-1	Regional City Store	AW00019377	FE-6654
AW00019377	ST-2	Regional City Store	AW00019377	HL-U509-B
			AW00019377	TI-M602
			AW00019377	TT-M928
			AW00019377	WB-H098

Building a dimensional data model

Adding Customer dimension



Tables

Customer

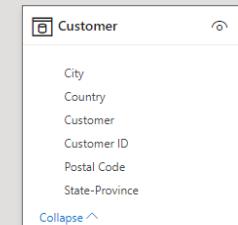
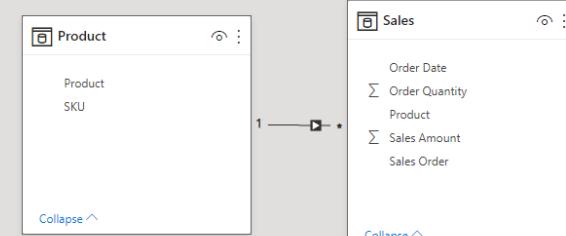
Customer ID	Customer	City	Postal Code	Country	State-Province
AW00019377	Dominic Chandra	Ingolstadt	85049	Germany	Bayern

Sales

Sales Amount	Order Quantity	Product	Sales Order	Order Date
\$8.99	1	CA-1098	SO74869	6/11/2020
\$21.98	1	FE-6654	SO74869	6/11/2020
\$29.99	1	TI-M602	SO74869	6/11/2020
\$9.99	1	BC-M005	SO74869	6/11/2020
\$4.99	1	TT-M928	SO74869	6/11/2020
\$34.99	1	HL-U509-B	SO74869	6/11/2020
\$4.99	1	WB-H098	SO74869	6/11/2020
\$8.99	2	CA-1098	SO74870	7/11/2020

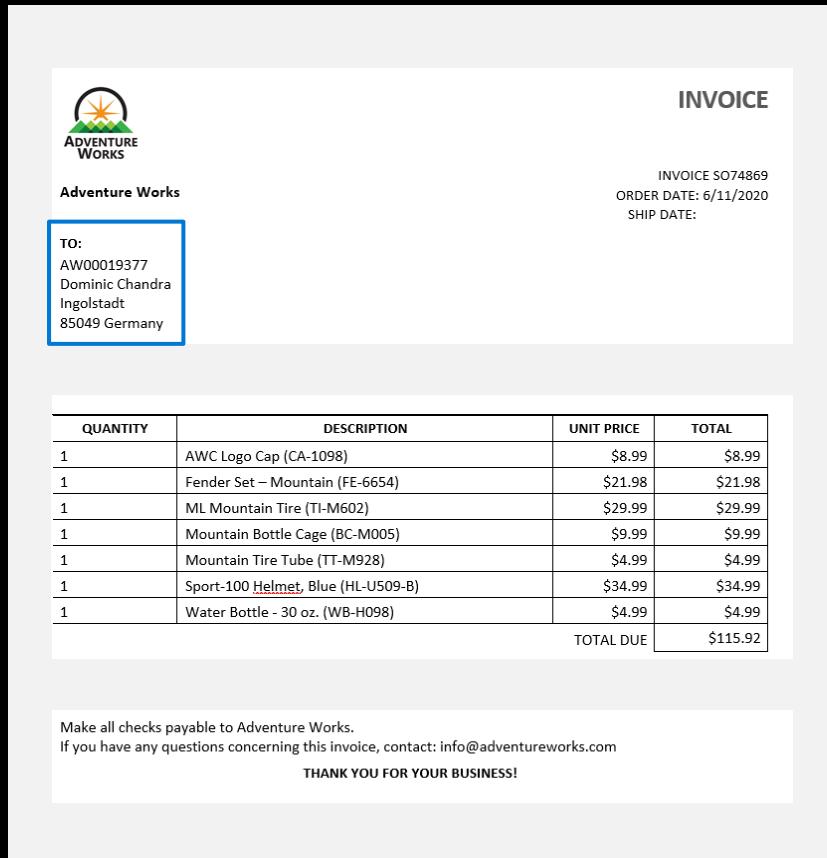
Product

Diagram



Building a dimensional data model

Adding Customer attribute on fact and relationship



Tables

Customer

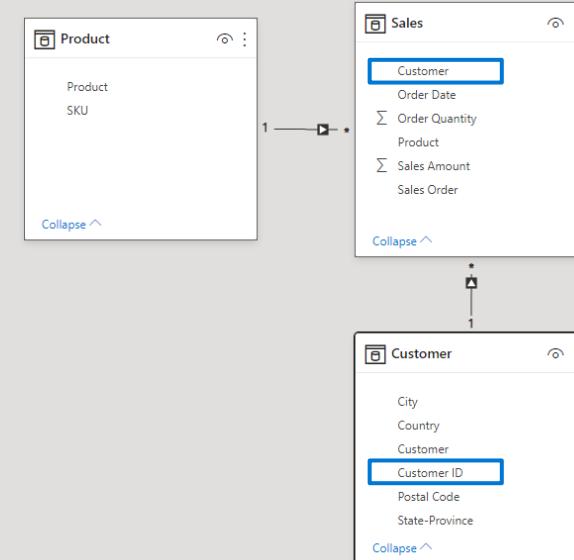
Customer ID	Customer	City	Postal Code	Country	State-Province
AW00019377	Dominic Chandra	Ingolstadt	85049	Germany	Bayern

Sales

Sales Amount	Order Quantity	Product	Sales Order	Order Date	Customer
\$8.99	1	CA-1098	S074869	6/11/2020	AW00019377
\$21.98	1	FE-6654	S074869	6/11/2020	AW00019377
\$29.99	1	TI-M602	S074869	6/11/2020	AW00019377
\$9.99	1	BC-M005	S074869	6/11/2020	AW00019377
\$4.99	1	TT-M928	S074869	6/11/2020	AW00019377
\$34.99	1	HL-U509-B	S074869	6/11/2020	AW00019377
\$4.99	1	WB-H098	S074869	6/11/2020	AW00019377

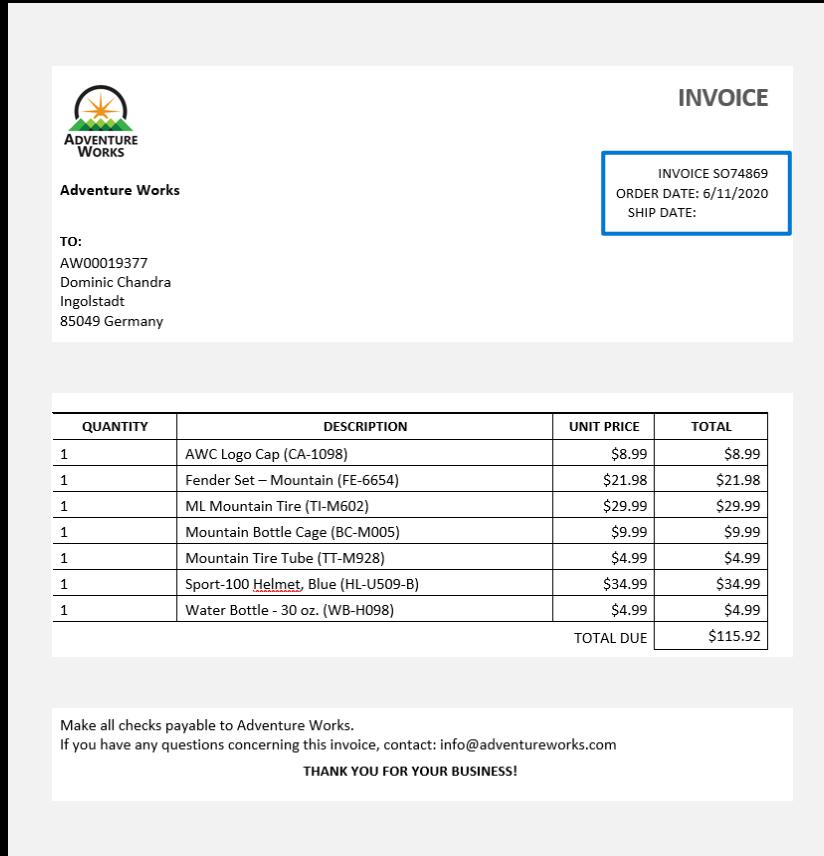
Product

Diagram



Building a dimensional data model

Adding Date dimension and relationship



Tables

Date

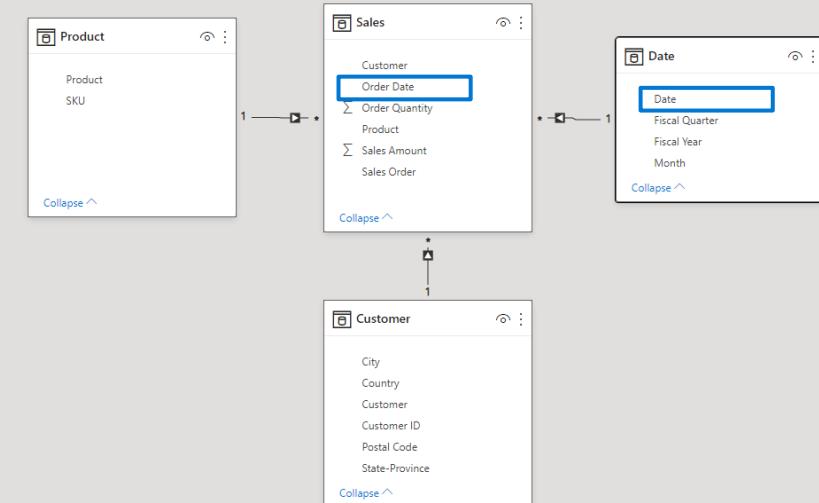
Date	Fiscal Quarter	Fiscal Year	Month
6/11/2020	FY2020 Q4	FY2020	2020 Jun
7/11/2020	FY2021 Q1	FY2021	2021 July

Product

Customer

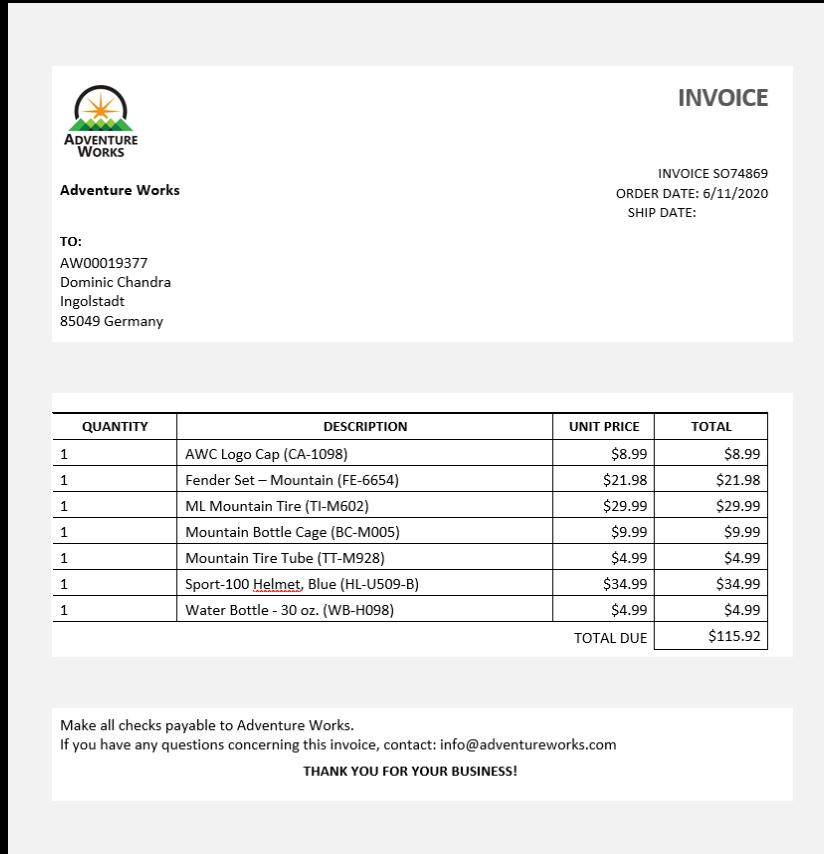
Sales

Diagram



Building a dimensional data model

Adding Ship Date attribute and relationship



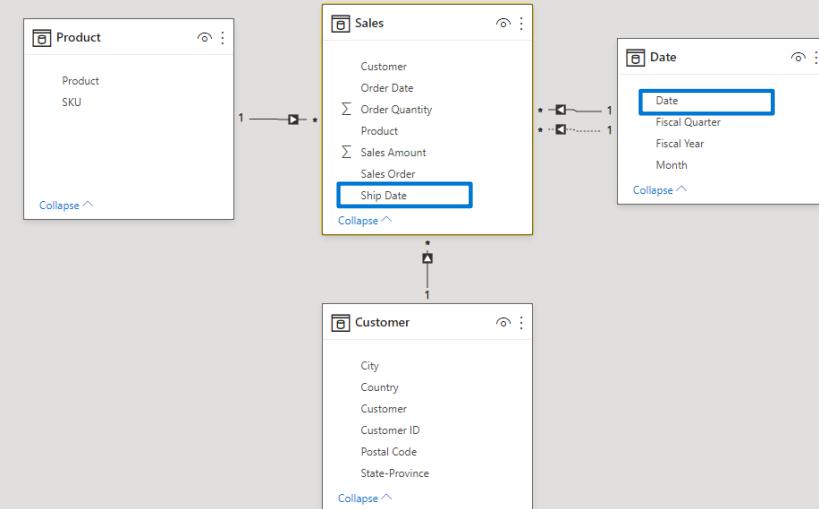
Tables

Sales

Sales Amount	Order Quantity	Product	Sales Order	Order Date	Customer	Ship Date
\$8.99	1	CA-1098	SO74869	6/11/2020	AW00019377	
\$21.98	1	FE-6654	SO74869	6/11/2020	AW00019377	
\$29.99	1	TI-M602	SO74869	6/11/2020	AW00019377	
\$9.99	1	BC-M005	SO74869	6/11/2020	AW00019377	
\$4.99	1	TT-M928	SO74869	6/11/2020	AW00019377	
\$34.99	1	HL-U509-B	SO74869	6/11/2020	AW00019377	
\$4.99	1	WB-H098	SO74869	6/11/2020	AW00019377	

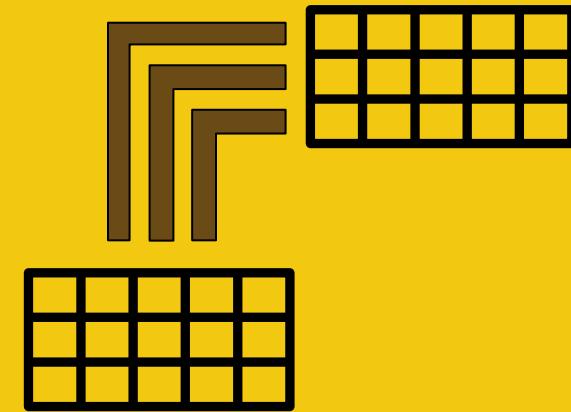
Product Customer Sales

Diagram



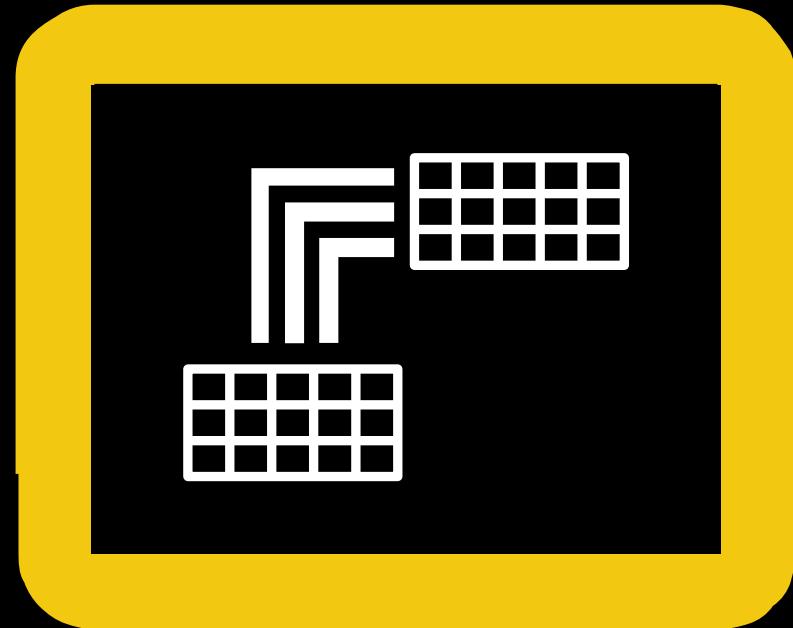
Role playing dimensions

- Role playing dimension: same dimension being used multiple times
- Example: Date dimension (OrderDate, ShipDate)
- In Power BI: one active relationship between two tables
 - Any other will be inactive
 - USERELATIONSHIP()
- Auto date/time
 - Auto date/time creates a date table for each date that is not used in a relationship
 - Might be worth it to introduce your own date dimension



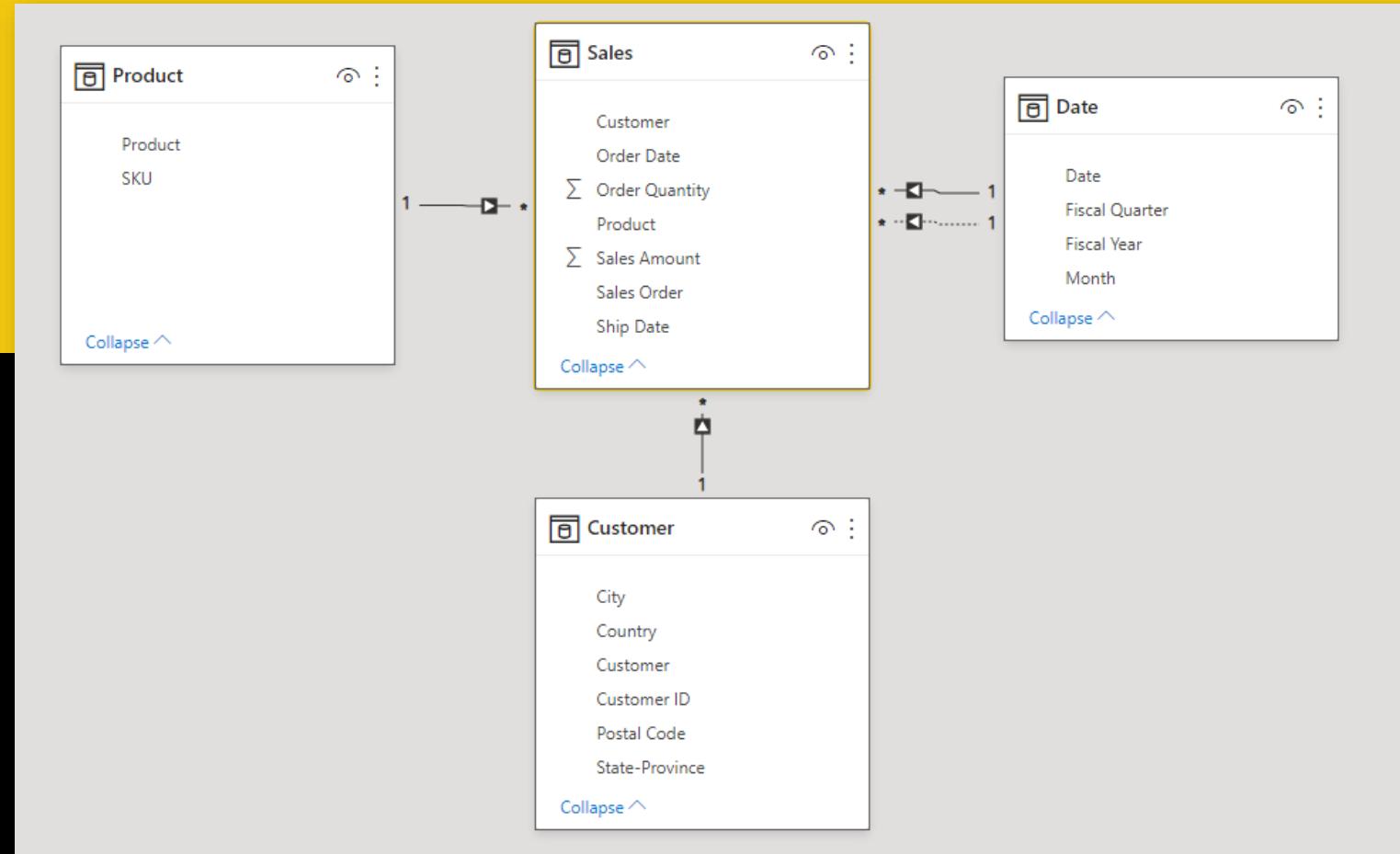
Role-playing dimensions

Demo



Building a dimensional data model

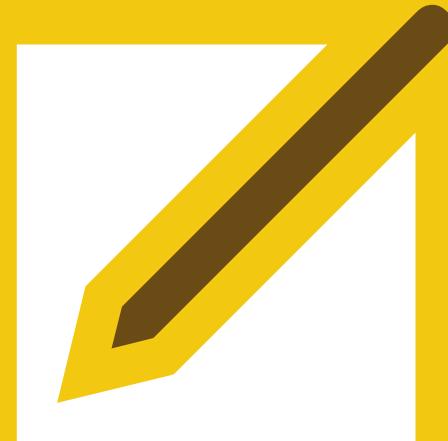
Final model



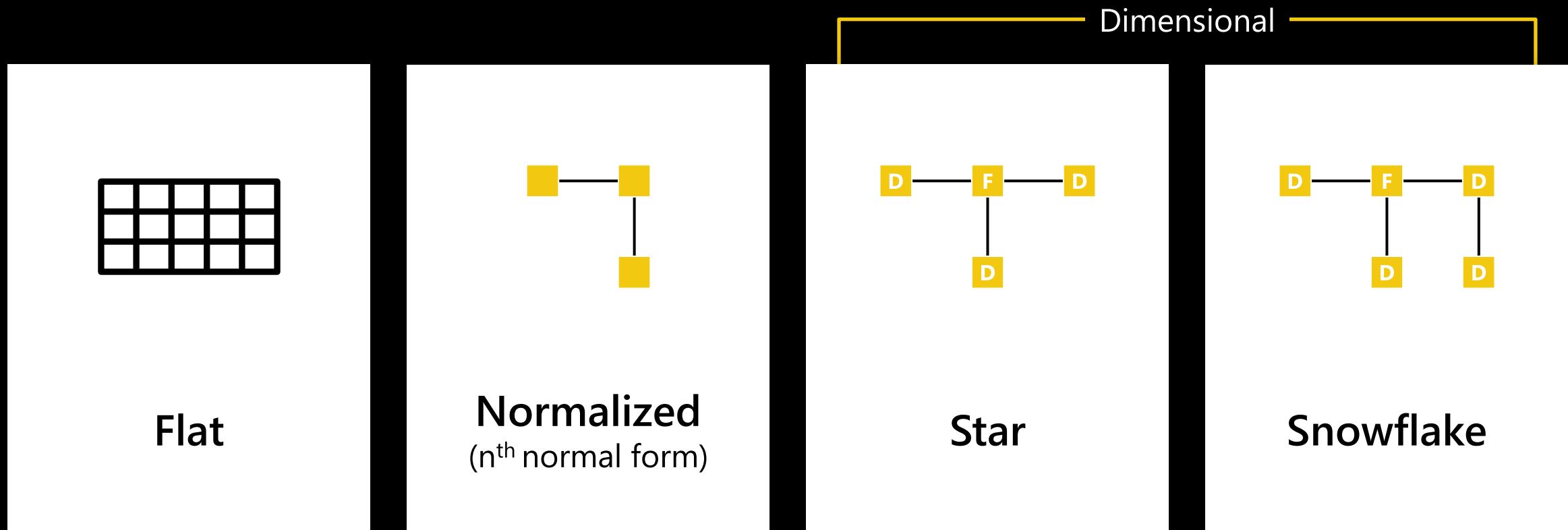
Note

→ This is a simplified, imperfect data model:

- Keys should be numerical instead of strings (surrogate keys)
- Sales Order should be separate dimension since the fact is on Sales Order Line, not Sales Order.



Major types of data models



Different data model types

	Flat table	Normalized	Star schema	Snowflake
 Performance for analytics	Low	Medium	High	High
 Development effort	Low	High	High	High
 Query volume and complexity	Low volume Low complexity	High volume Low complexity	Low volume High complexity	Low volume High complexity
 Intended for	No database	CRM / ERP / Applications	Analytical systems / data warehouses	Analytical systems / data warehouses
 Compression	Row	Row	Column	Column

Considerations for data modeling in Power BI

- Do data cleansing activity in one place.
- Enrich your data with measures and calculated columns.



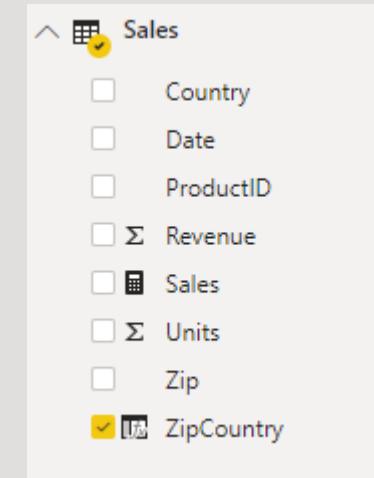
Calculated Column

→ To create a calculated column using a DAX formula, select the New column button from the Modeling tab.

→ A calculated column:

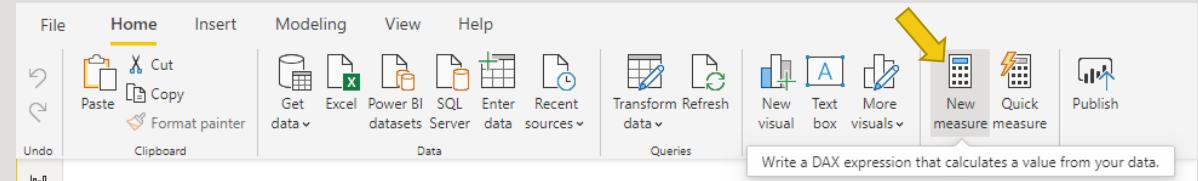
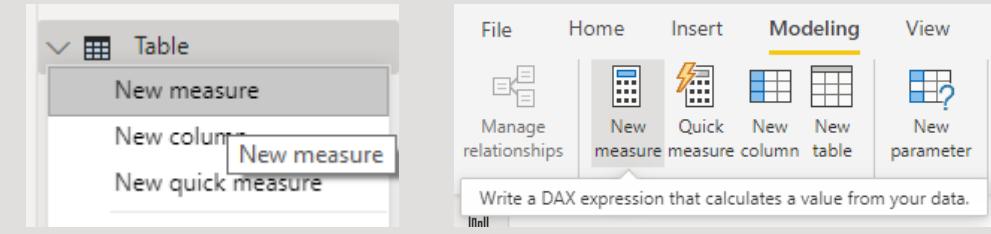
- Is useful for slicing or filtering a value, or for doing a calculation on every row in the table
- Saves its result in the data model
- Evaluates while reloading the data model, not while executing the query
- Consumes **memory!** (increases model size)
- Evaluates per row

```
DiscountCategory =  
IF(  
    Sales[DiscountPct] = 0; "FULL PRICE";  
    IF(Sales[DiscountPct] <= 0,05; "LOW";  
        IF(Sales[DiscountPct] <= 0,1; "MEDIUM"; "HIGH")))
```



Measure

- To create a measure using a DAX formula, select the New measure button from the Modeling tab.
- A measure:
 - Aggregates a column
 - Evaluates a total, not row-per-row
 - Evaluates while executing the query
 - Costs **CPU-power** instead of memory
 - Always has one answer



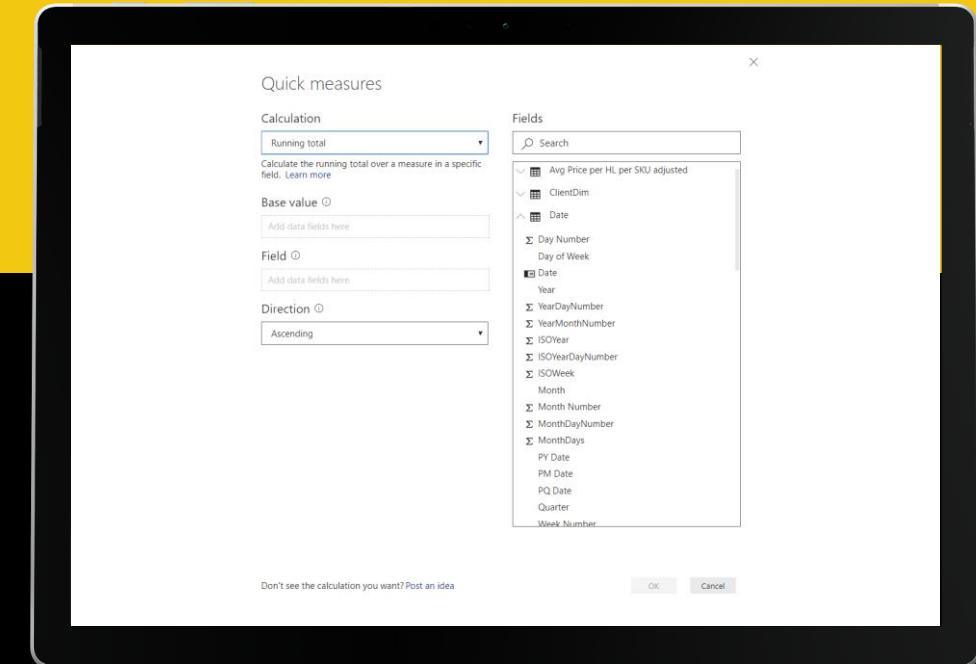
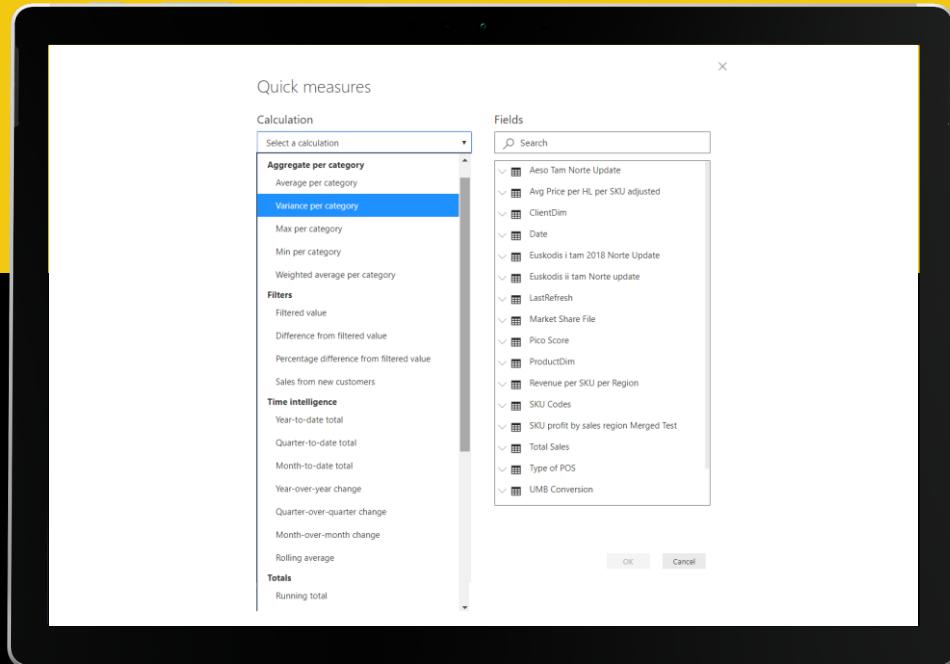
Calculated Column vs Measure

	Calculated Column	Measure
Persistence	Definition and result saved in data model	Definition saved in data model, result is calculated when a visual is created
Usage	Can be used on slicers, filters, rows, columns, etc.	Can only be used as a value in a visual and visual level filters
Language	DAX / M (Power Query)	DAX
Scope	Single row	Set of rows / table
Calculation happens	On data refresh	On demand when a visual is created
Effect	Blows up data model	Increases report performance (compared to calculated column)

ProductID	Date	CustomerID	CampaignID	Units	Sales Amount
449	7/29/14	128304		1	\$102.37
449	7/29/14	89917		1	\$102.37
449	7/29/14	128811		1	\$102.37
449	7/29/14	59550		1	\$102.37
449	7/29/14	207690		1	\$102.37
449	7/29/14	121043		1	\$102.37
449	7/29/14	148675		1	\$102.37
449	7/29/14	65048		1	\$102.37

Quick Measures

You can make many difficult DAX calculations without code, **just drag and drop.**



Measures and Calculated Columns

Demo



Rules to live by



DAX complexity down,
Performance up



Simple DAX is a sign of
a good data model



Microsoft
Power BI

Key take-aways

- Do data transformation as close to the source as possible. In Power BI: use the “Transform Data” options.
- Use a star schema or snowflake data model to get the best out of Power BI.
- Consider auto date/time vs creating a Date dimension
- Use a role-playing dimension instead of adding the same dimension multiple times.
 - Most common example: Date



Build your own data model

aka.ms/datamodelingworkshop



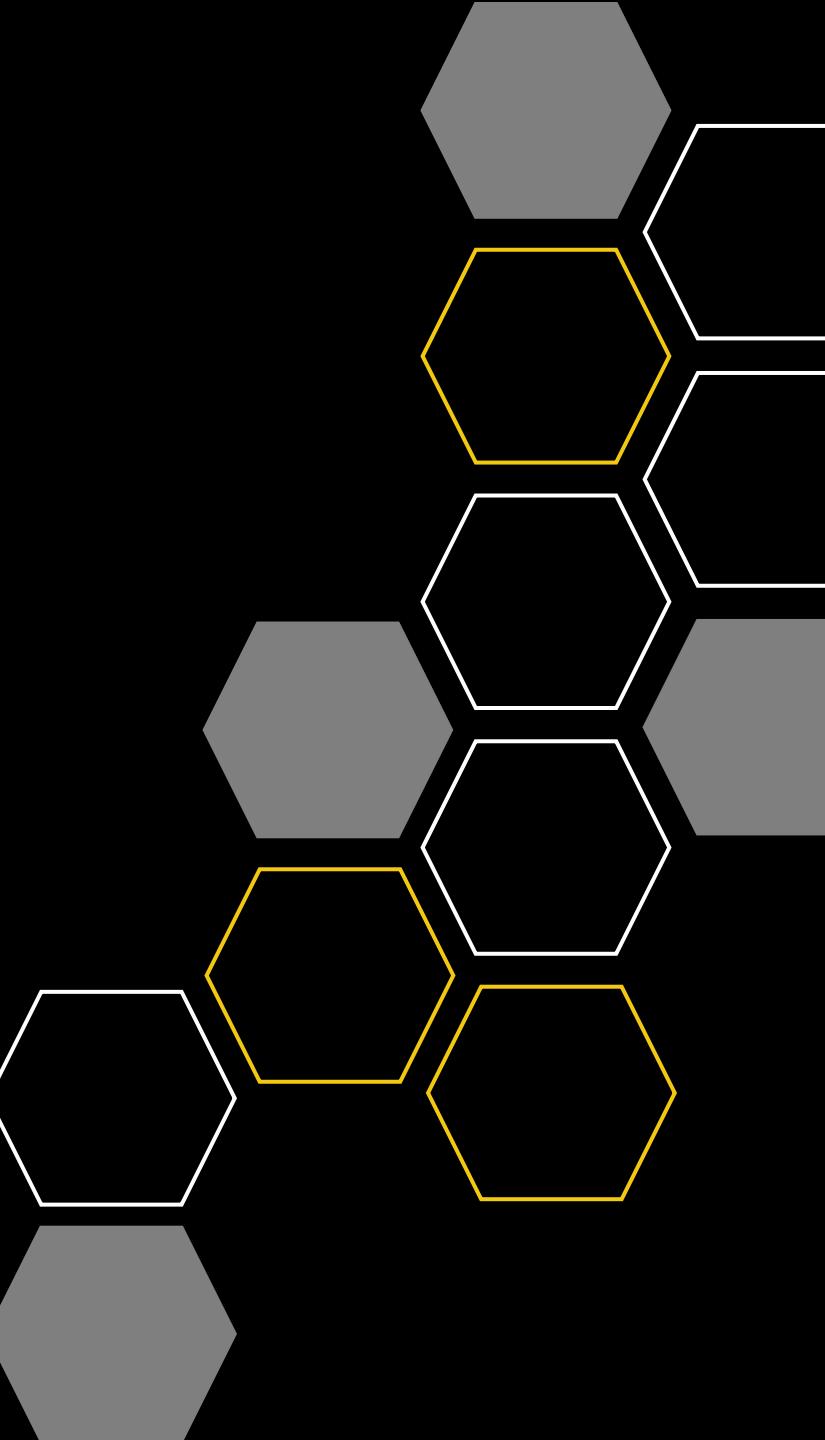
From now until lunch:

- Team name
- Discuss and design your data model
 - Pen & paper, flipovers available
- Be prepared to present your findings

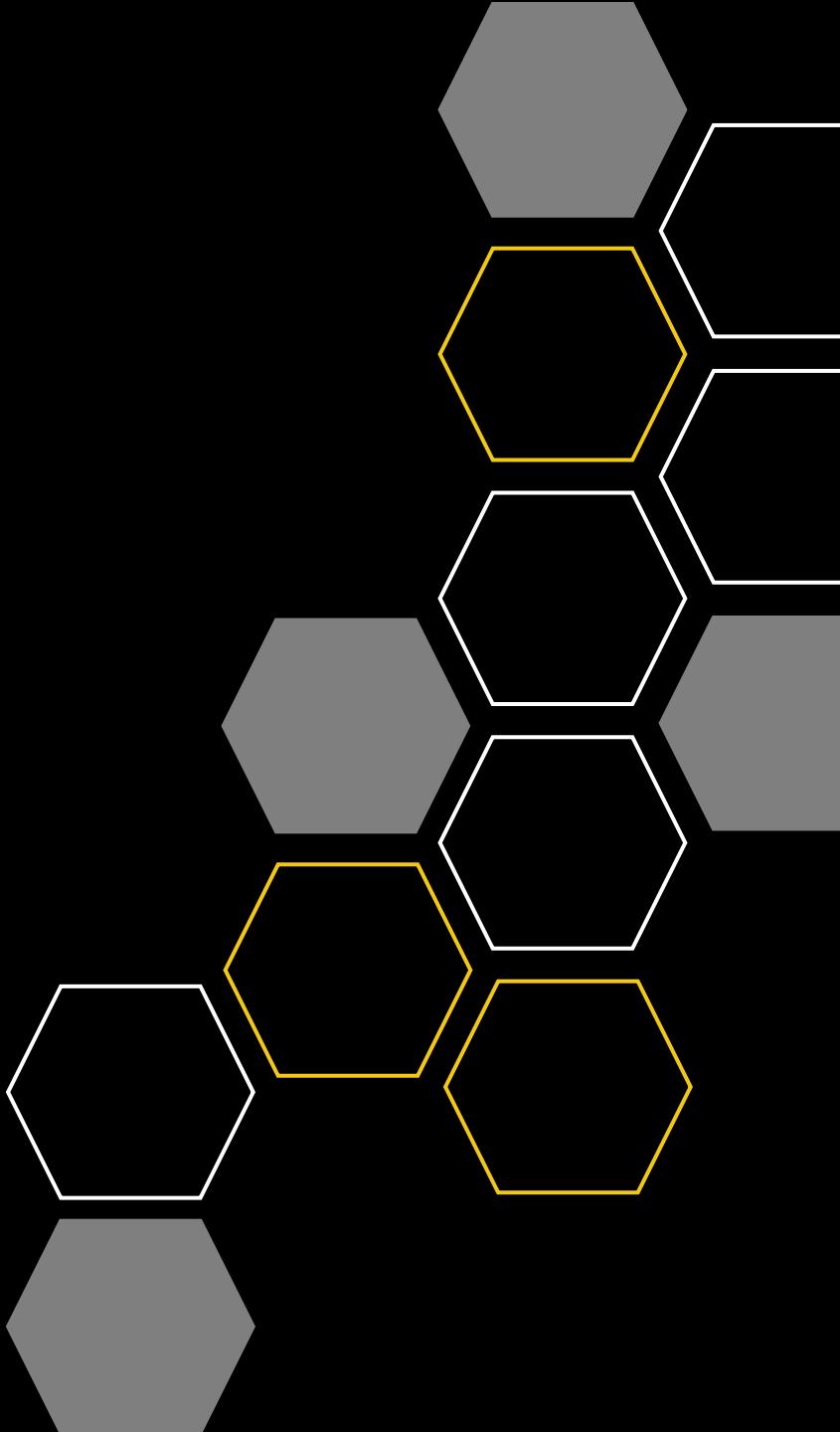
Lunch!



**Present your findings
(max 5 minutes)**



Relationships revisited



Relation types

There are **three types of relationships** in data modeling

1 to 1

Every **individual record** in dataset A is mapped to one **individual record** in dataset B

Dataset A Customer information		Dataset B Customer information	
Customer ID	Country	Customer ID	Date of Birth
AW000111024	United States	AW000111024	9 April 1990
AW00019377	Germany	AW00019377	9 April 1983

1 to many

One record of dataset A is mapped to **multiple records** in dataset B

Dataset A Customer information		Dataset B Sales Information		
Customer ID	Country	Customer ID	Product	Order Quantity
AW000111024	United States	AW000111024	CA-1098	2
AW00019377	Germany	AW00019377	BC-M005	1
		AW00019377	CA-1098	1
		AW00019377	FE-6654	1
		AW00019377	HL-U509-B	1
		AW00019377	TI-M602	1
		AW00019377	TT-M928	1
		AW00019377	WB-H098	1

Many to many

Multiple records of dataset A are mapped to **multiple records** in dataset B

Dataset A Customer information		Dataset B Sales Information		
Customer ID	Store ID	Store Type	Customer ID	Product
AW000111024	ON-1	Online	AW000111024	CA-1098
AW000111024	ST-1	Regional City Store	AW000111024	TT-M928
AW000111024	ST-2	Regional City Store	AW00019377	BC-M005
AW00019377	ON-1	Online	AW00019377	CA-1098
AW00019377	ST-1	Regional City Store	AW00019377	FE-6654
AW00019377	ST-2	Regional City Store	AW00019377	HL-U509-B
			AW00019377	TI-M602
			AW00019377	TT-M928
			AW00019377	WB-H098

Relationship direction

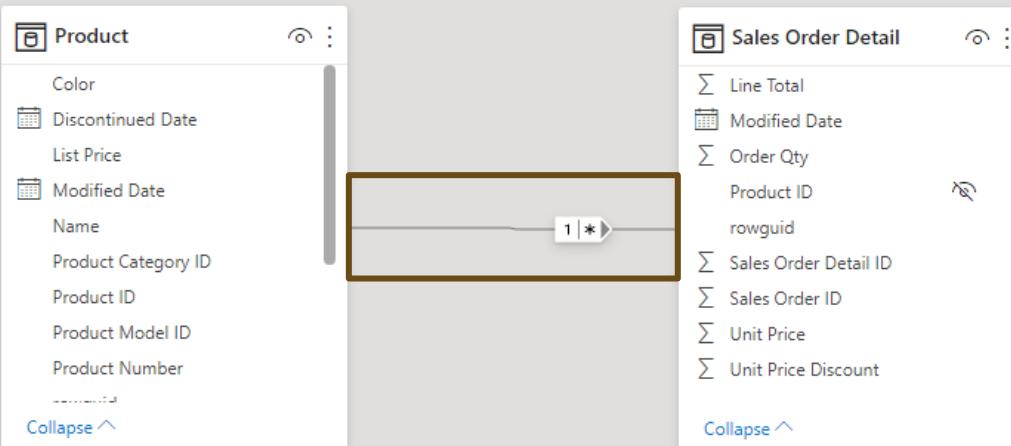
We explained **three types of relationships**, but skipped over the relationship direction



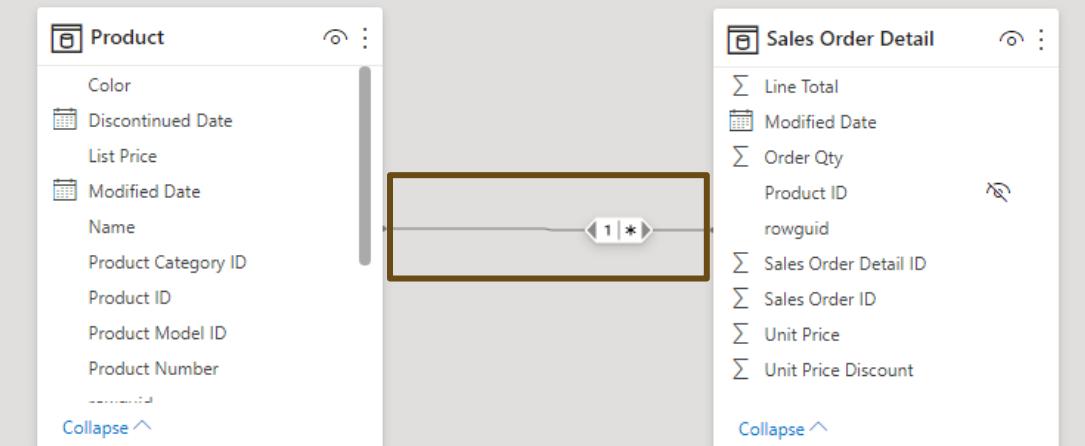
Relationship direction

There are **two types** of relationship directions

Singular



Bi-directional



Relationship direction

Bi-directional relationships can result in surprising results, especially when working with multiple fact tables

- It can have performance impacts and 'overfilter'
- Can lead to ambiguity

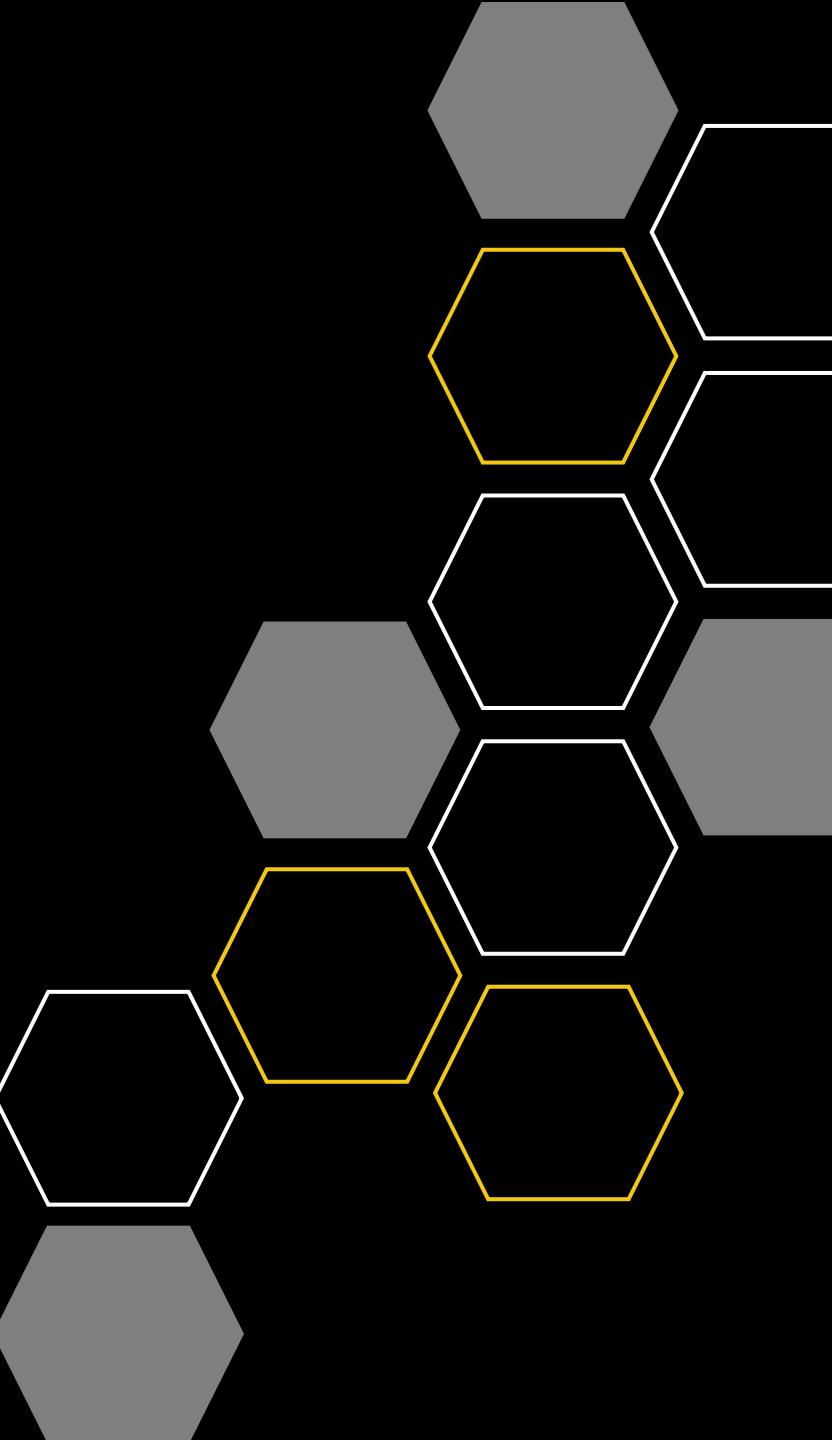
Try to avoid bi-directional relationships

You can influence the direction of a relationship for the context of a calculation by CROSSFILTER in DAX:

- CROSSFILTER(column1, column2, direction)

Relationship direction

Marc Lelijveld



Working with multiple fact tables



Multiple fact tables

Multiple fact tables are common in real-life scenarios. Common scenarios:

Fact tables that you can append into one because they share common dimensions

Facts on different levels of granularity, different topics or that do not share common dimensions

Not sure what scenario you have? Use a mapping table.

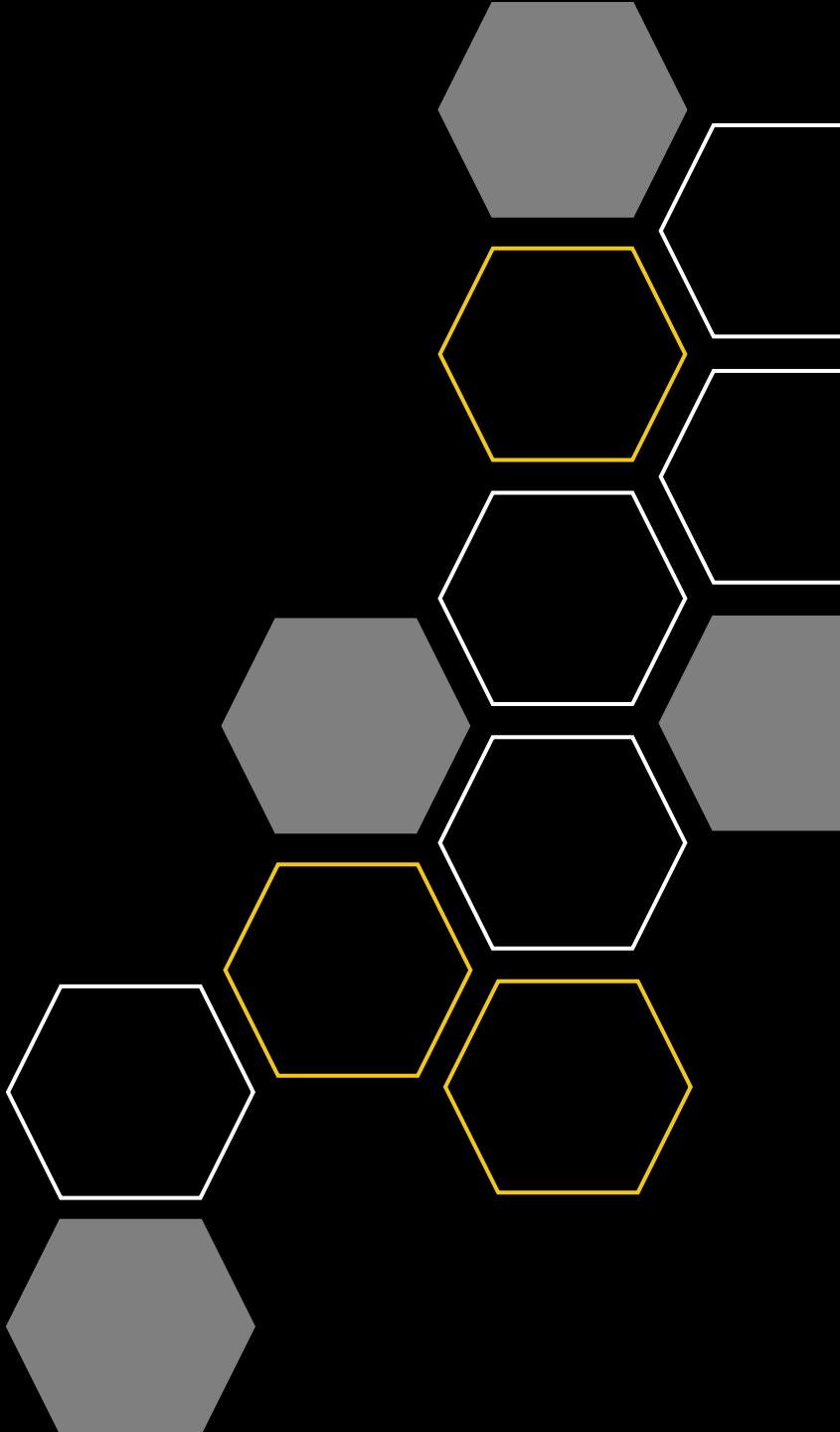
Mapping table sales

	Currency	Customer	Due date	Employee	Order date	Product	Promotion	Reseller	Sales territory	Ship date
Internet sales	X	X	X		X	X	X		X	X
Reseller sales	X		X	X	X	X	X	X	X	X

If you decide to append these, you
will get (blanks) when you use one of
the unmatched columns

Strategies for working with multiple fact tables

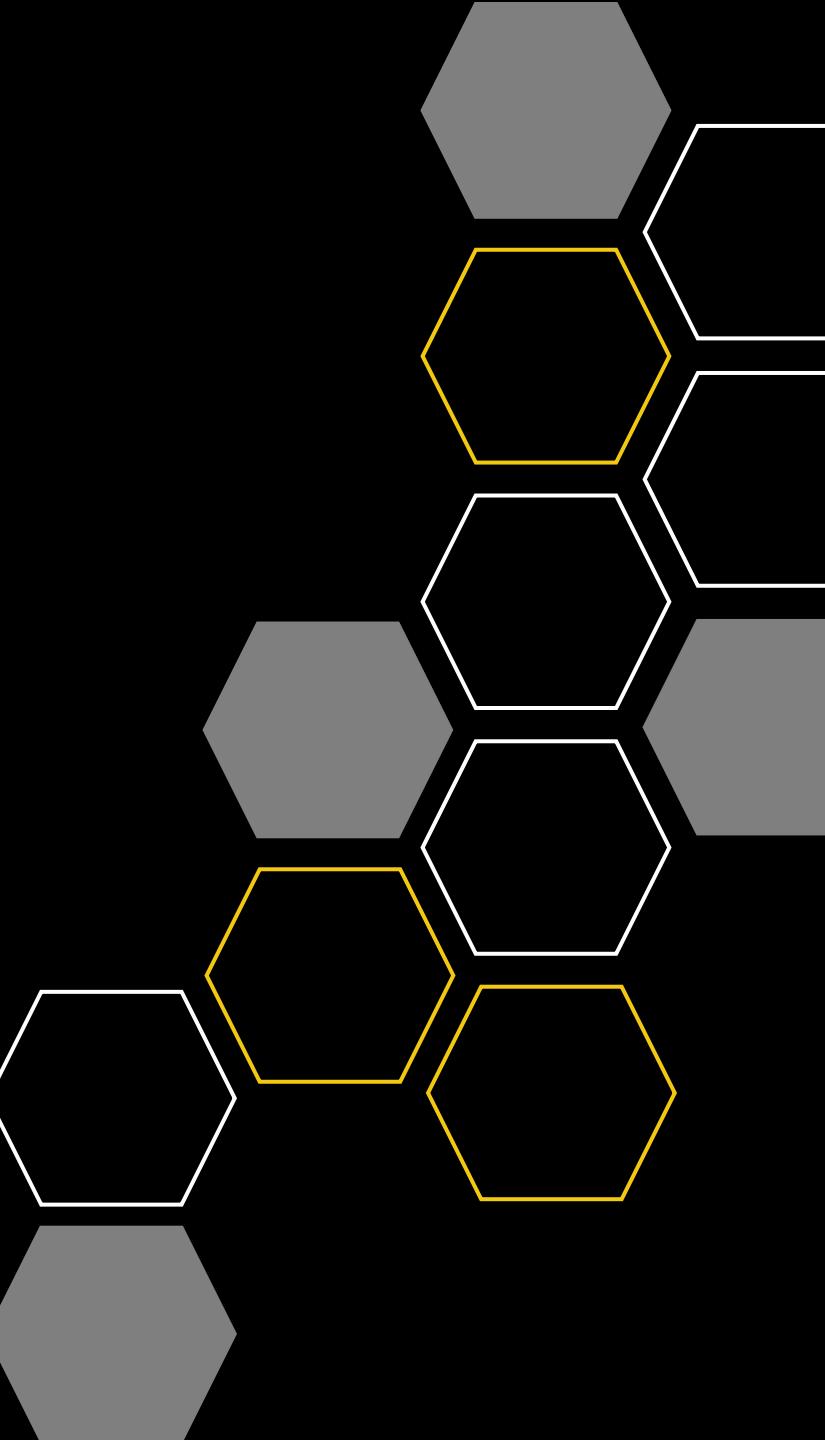
Jeroen ter Heerdt



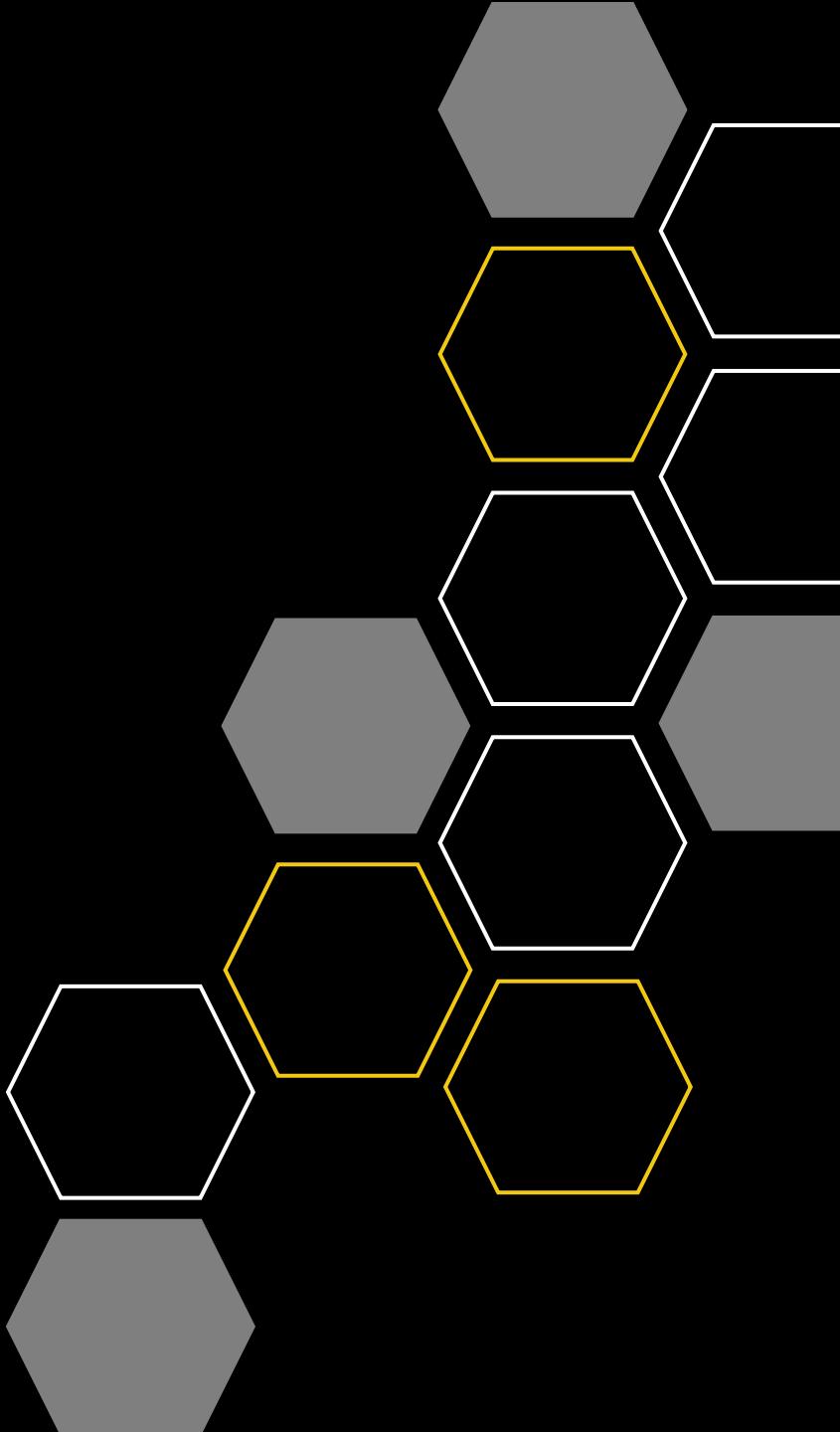
Extend
your own
data model



**Present your findings
(max 5 minutes)**



Different types of fact tables

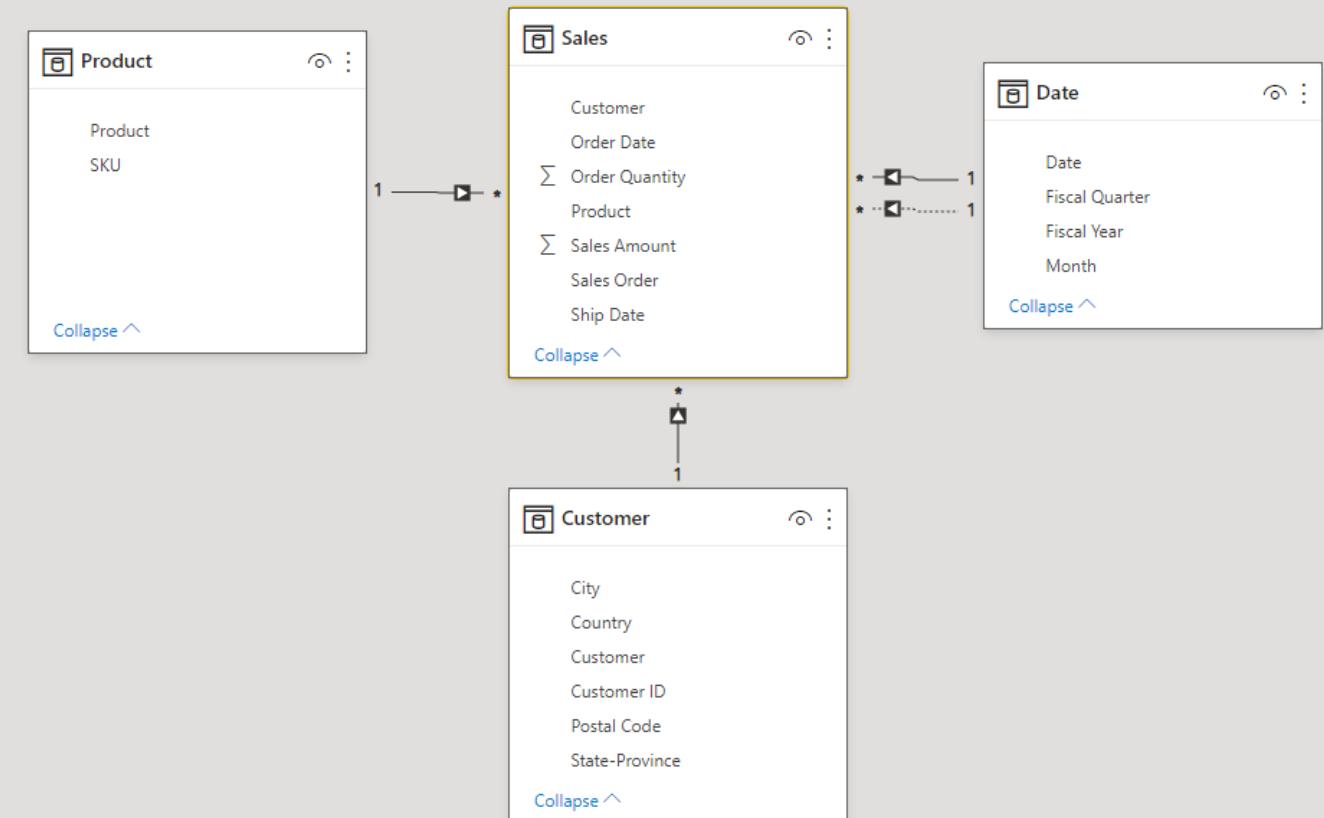


Revisiting different types of data models

Dimensional Schemas:

→ **Dimensions:** define how a fact should roll up

→ **Facts:** numerical information and measurements



Different types of fact tables



Additive facts

Can be summed across any of the dimensions associated with the fact table.

- Most common fact
- Measurements can be made in the fact table related to all dimensions

Example:

- We **can** say that the total sales amount for product 528 = \$ 9,98
- We **can** say that product 480 has been sold 3 times

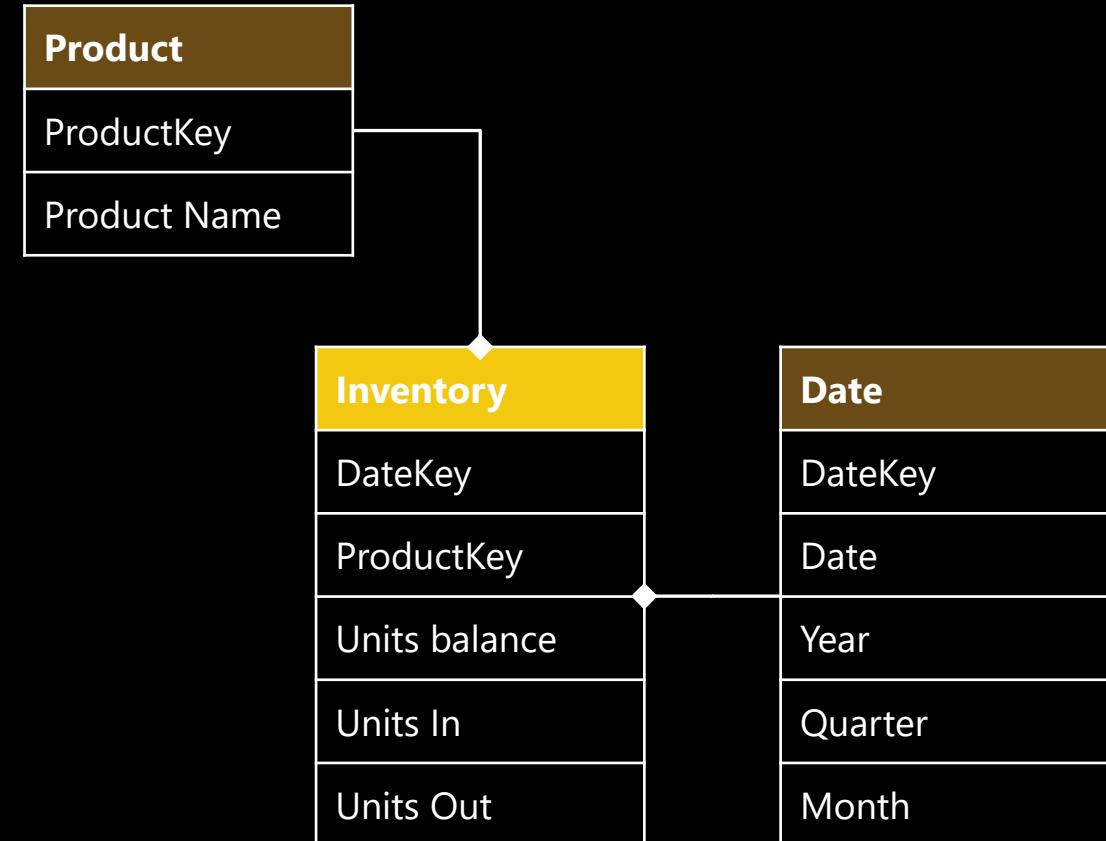
ProductKey	OrderDateKey	Sales Amount
528	20190701	\$ 4,99
528	20190702	\$ 4,99
480	20200213	\$ 2,29
480	20200202	\$ 2,29
480	20200127	\$ 2,29

Semi-additive facts

Measurements can be summed in the fact table related to **some** dimensions, but not all.

Example:

- Inventory, I can sum the Units balance on inventory per day, but it does not make sense to sum by month / year
- Units In / Out = Additive
- Units Balance = Semi-additive



Non-additive facts

- Fact values that cannot be summed
- In some cases, it can still be aggregated

Example:

- We **cannot** say that the total margin = 100%
- We **can** state that the average margin = 50%

Other example: Age can be averaged, but it does not make sense to take a sum of age.

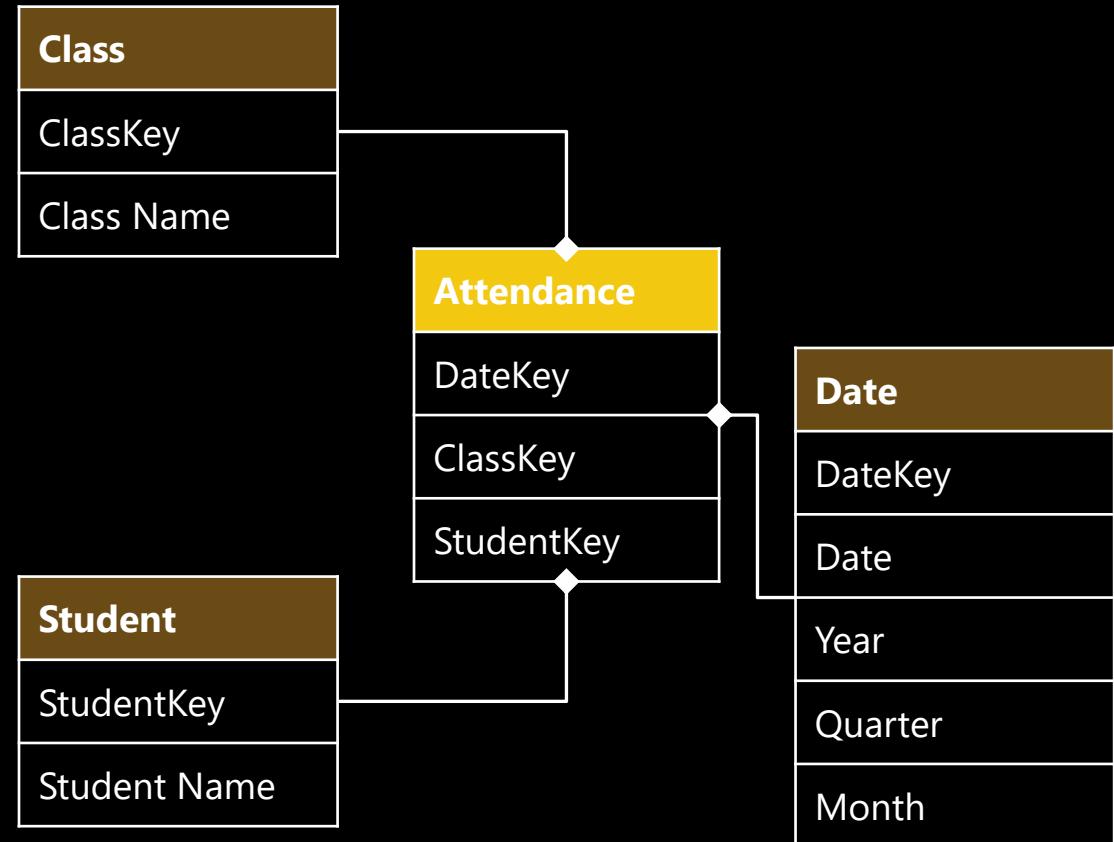
Employee	Margin percentage
Account manager A	55%
Account manager B	45%

Factless facts

- Do not include any measurements
- Only includes foreign keys to related dimensions that define the transaction

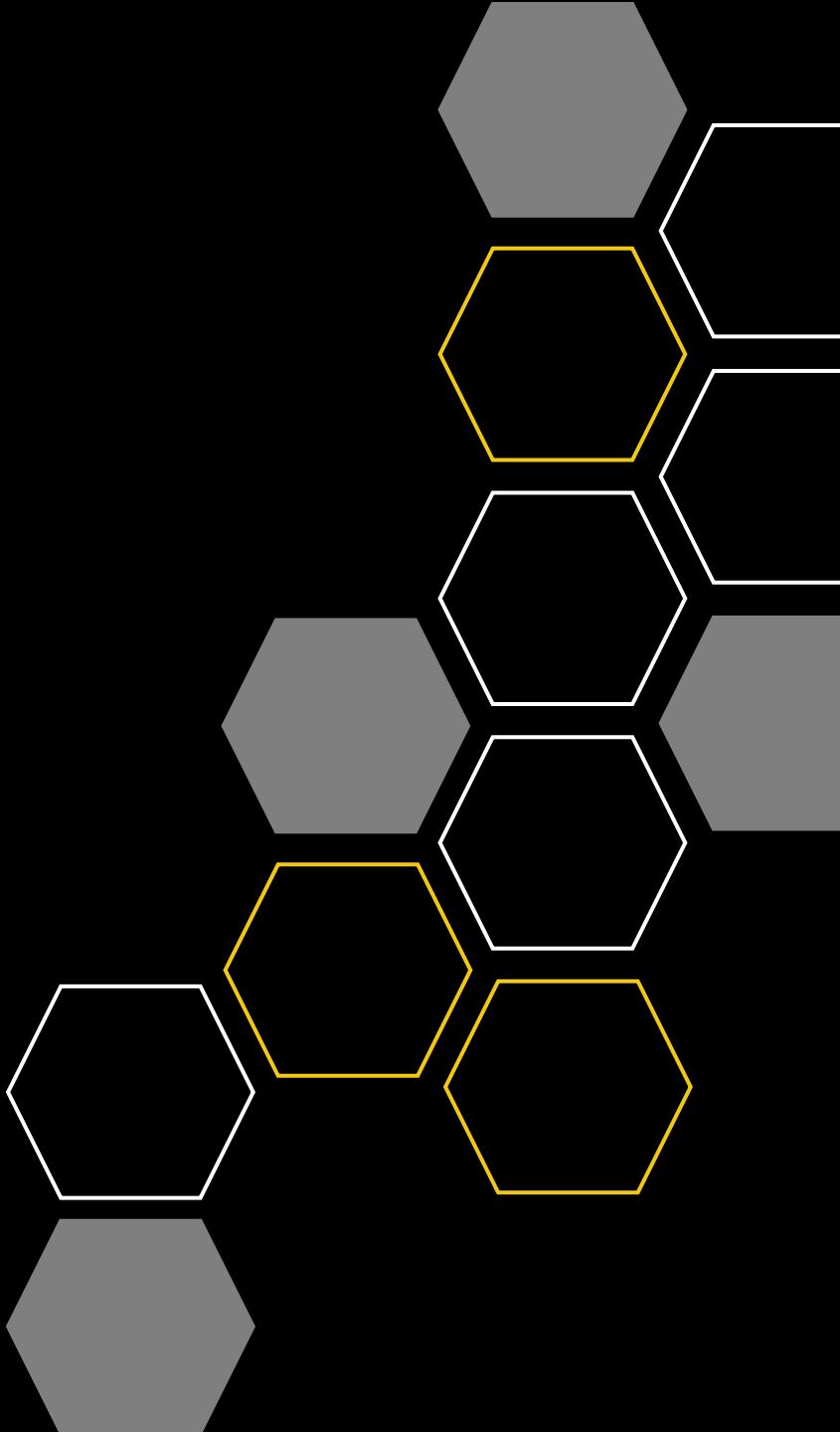
Example:

- Student attendance to a class
- If there is a record in the attendance table, the student was there, otherwise there will not be a record

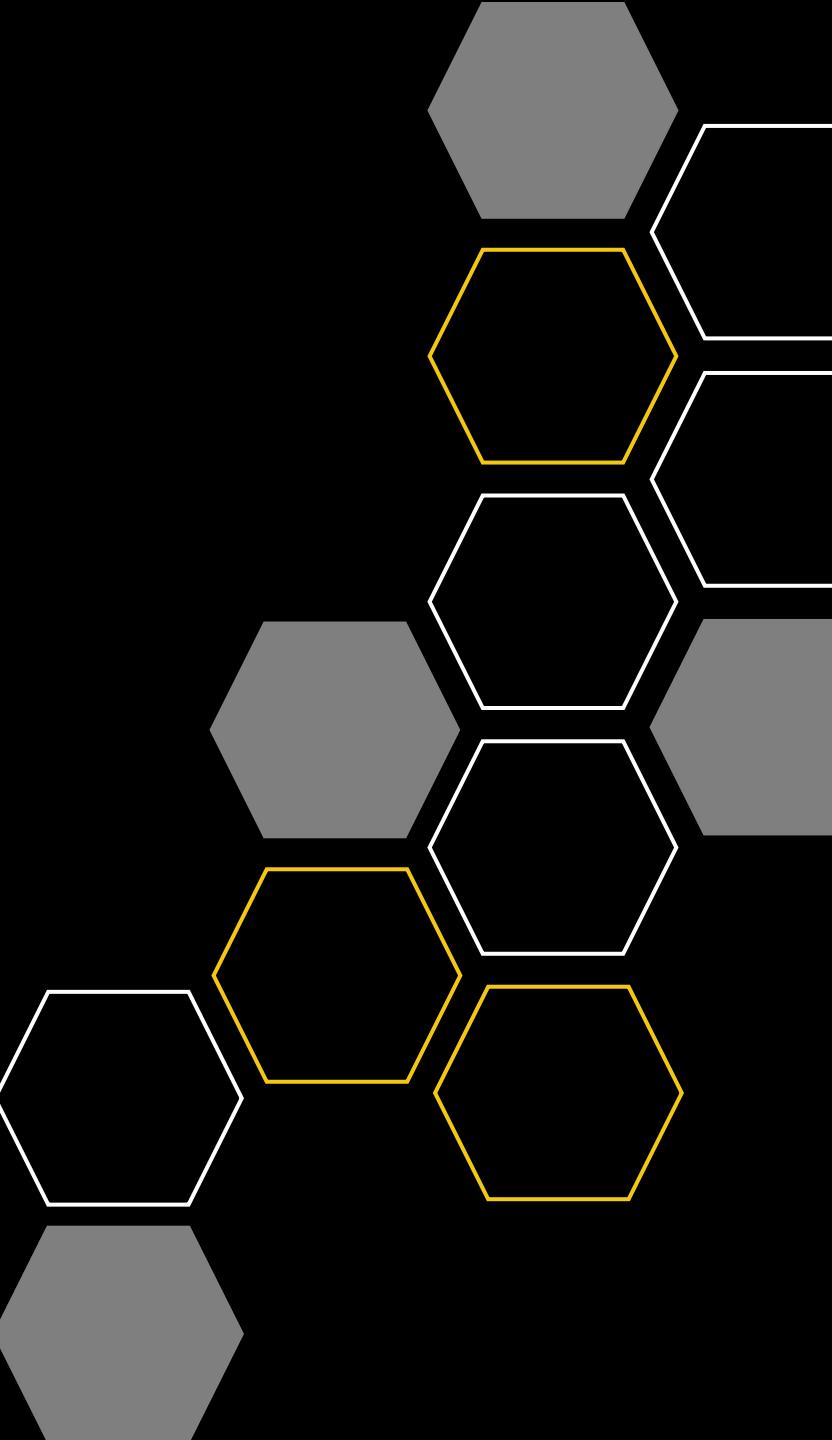


Demo: additive, semi-additive and non-additive facts

Marc Lelijveld



Data modeling best practices



Naming of tables

Many sources use technical names

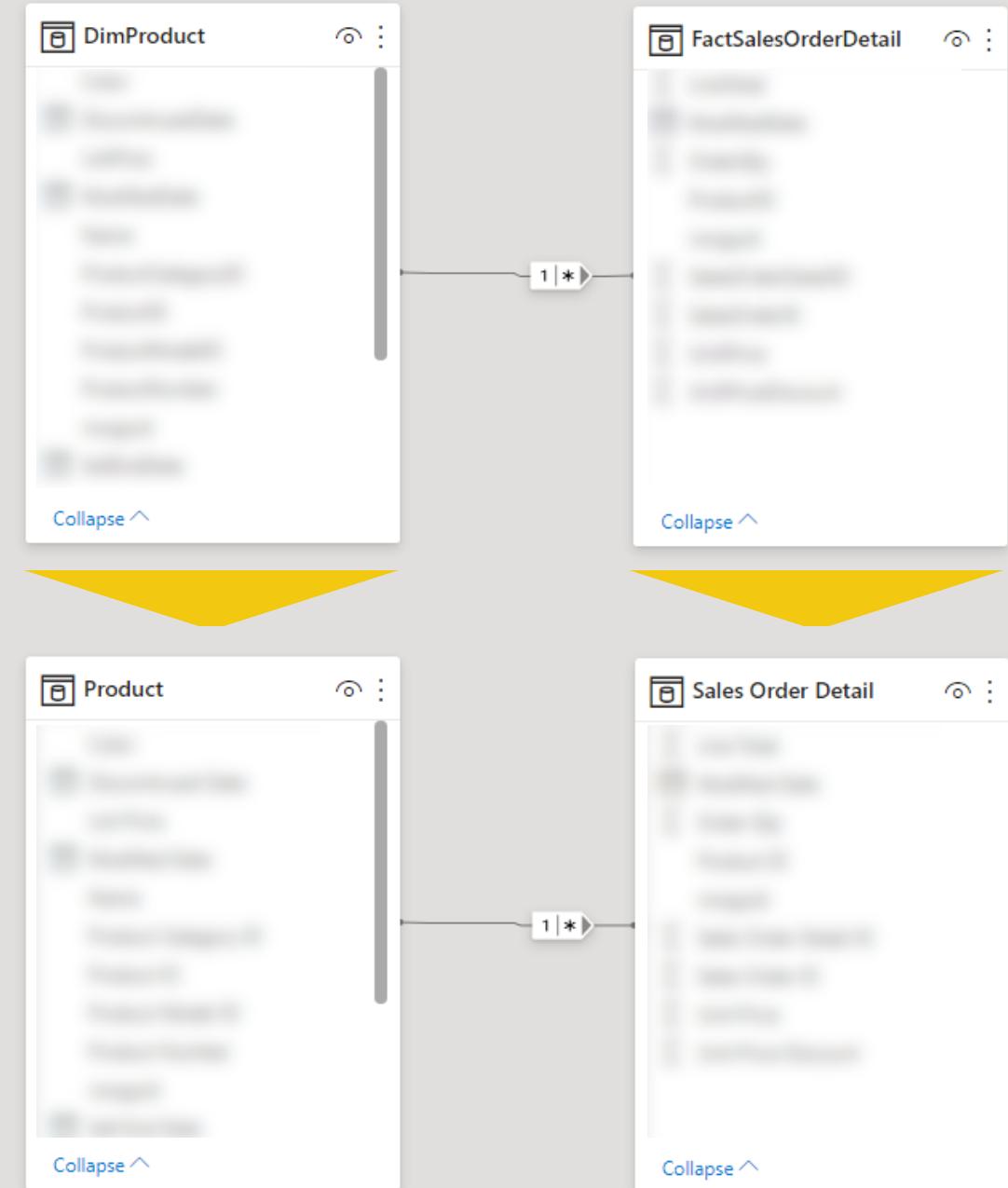
- FactSalesOrderDetail
- CustomerID

Remove Dim/Fact moniker and add spaces

- Easier to use
- Better searchability
- Q&A / AI support

FactSalesOrderDetail → Sales Order Details

DimProduct → Product



Naming of columns

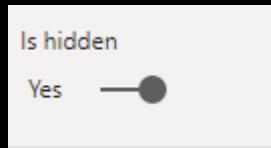
Consider renaming columns as well to be consistent.

	1 ² 3 SalesOrderID	1 ² 3 SalesOrderDetailID	1.2 OrderQty	1 ² 3 ProductID	\$ UnitPrice	\$ UnitPriceDiscount	1.2 LineTotal
1							
2							
3							
4							
5							
6							
7							
8							
9							

	1 ² 3 Sales Order ID	1 ² 3 Sales Order Detail ID	1.2 Order Qty	1 ² 3 Product ID	\$ Unit Price	\$ Unit Price Discount	1.2 Line Total
1							
2							
3							
4							
5							
6							
7							
8							
9							

Hide what doesn't matter

- Hide all keys in the tables to avoid confusion
- Bulk edit in the Properties pane of the Model View



The screenshot shows the Power BI Model View for the 'Sales' table. The table contains the following columns:

Column	Type	Visibility
CustomerKey		Visible
Σ DueDateKey		Visible
Σ Extended Amount		Visible
Σ Order Quantity		Visible
OrderDateKey		Visible
Σ Product Standard Cost		Visible
ProductKey		Visible
ResellerKey		Visible
Σ Sales Amount		Visible
SalesOrderLineKey		Visible
SalesTerritoryKey		Visible
Σ ShipDateKey		Visible
Σ Total Product Cost		Visible
Σ Unit Price		Visible
Σ Unit Price Discount Pct		Visible
Sales Amount by Due Date	Calculated Column	Visible

A vertical column of visibility icons is on the right, showing a sequence of 'Visible' (green checkmark) and 'Hidden' (red crossed-out checkmark) symbols. The last four icons are highlighted with a brown rectangle, corresponding to the last four visible columns listed above.

Column Settings

Hide columns

- If measure has been created
- Hide the original column

Consider Summarization:

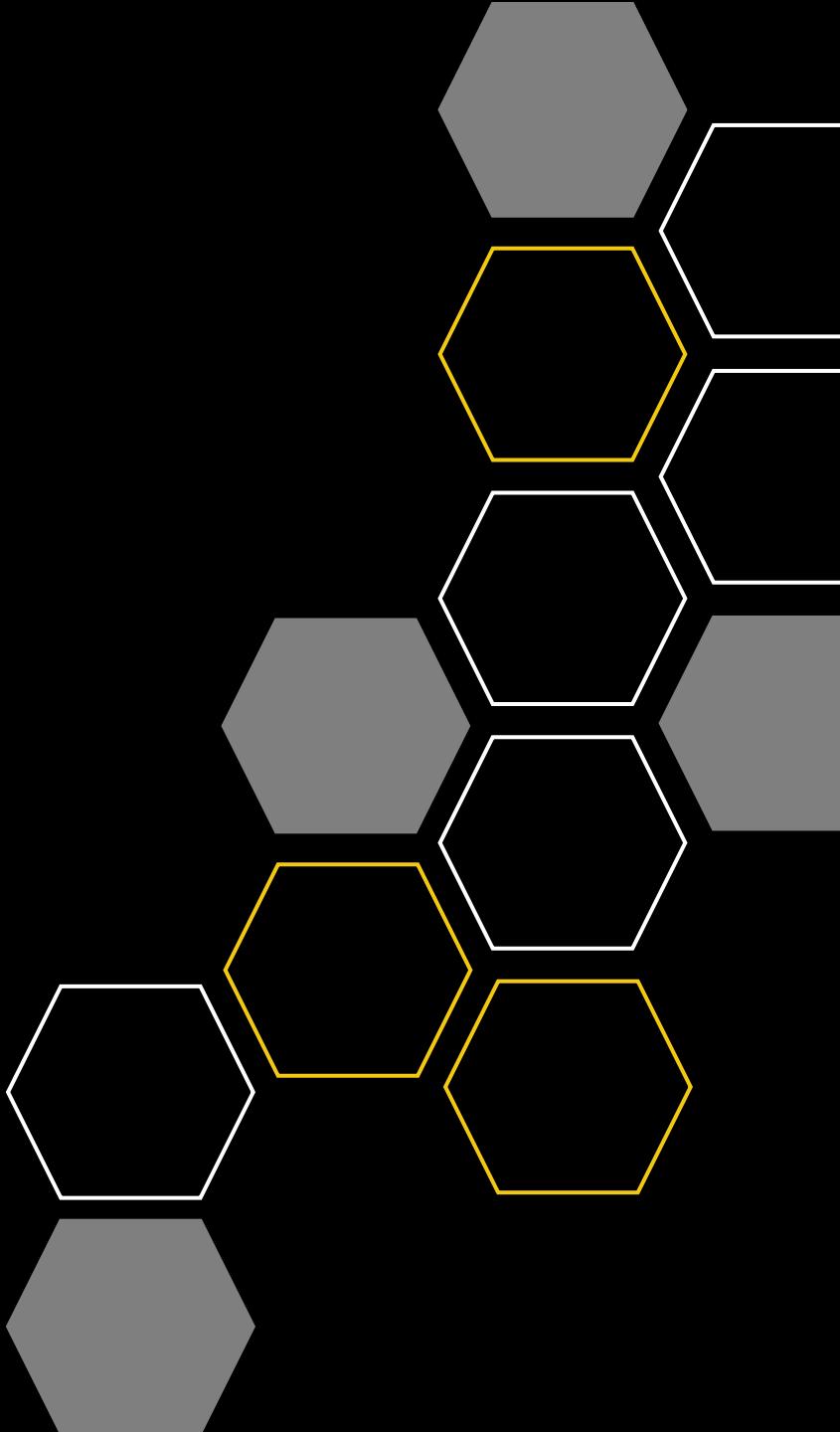
- Disable for key columns (if not hidden)
- Potentially change for others
 - Example: Age should not be set to Sum, but could be set to Average, Min, Max, depending on use case

The screenshot shows the 'Sales' table settings in Power BI. The table contains 15 columns, each with a 'Settings' icon (gear) to the right. The columns are listed below:

- CustomerKey
- DueDateKey
- Extended Amount
- Order Quantity
- OrderDateKey
- Product Standard Cost
- ProductKey
- ResellerKey
- Sales Amount
- SalesOrderLineKey
- SalesTerritoryKey
- ShipDateKey
- Total Product Cost
- Unit Price
- Unit Price Discount Pct
- Total Sales

The 'Sales Amount' and 'Total Sales' columns are highlighted with a brown border, indicating they are selected or being focused on.

Date and time dimensions

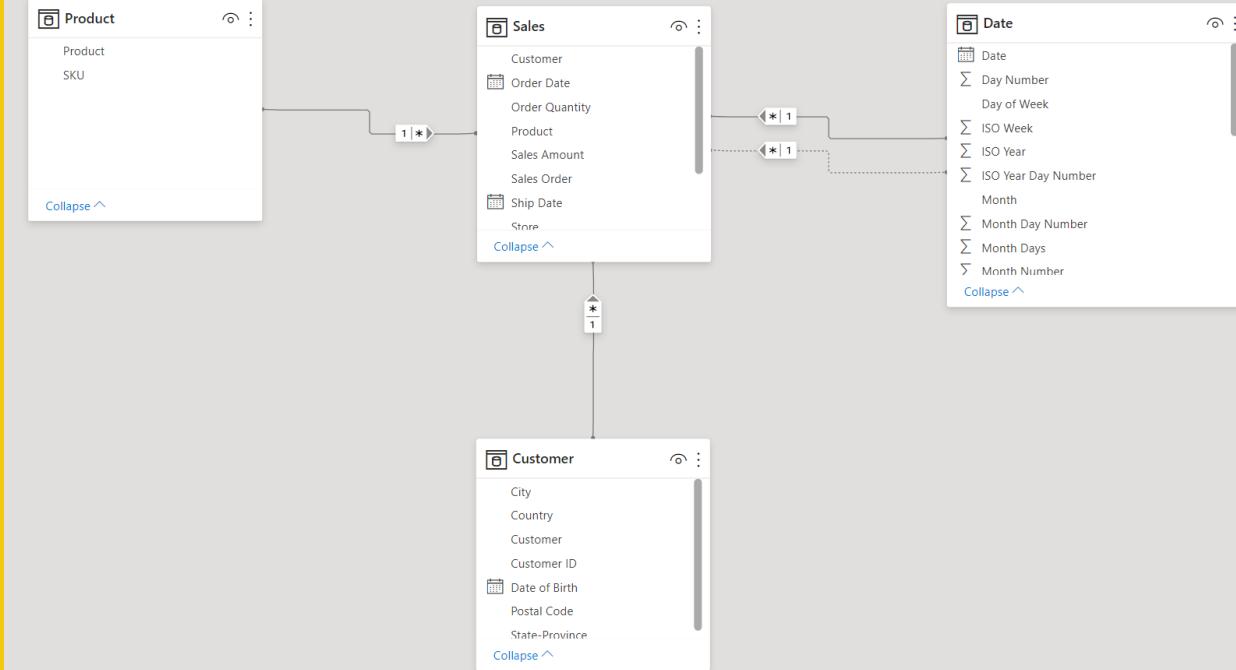


Auto date/time

Auto date/time creates a date table for each date that is not used in a relationship

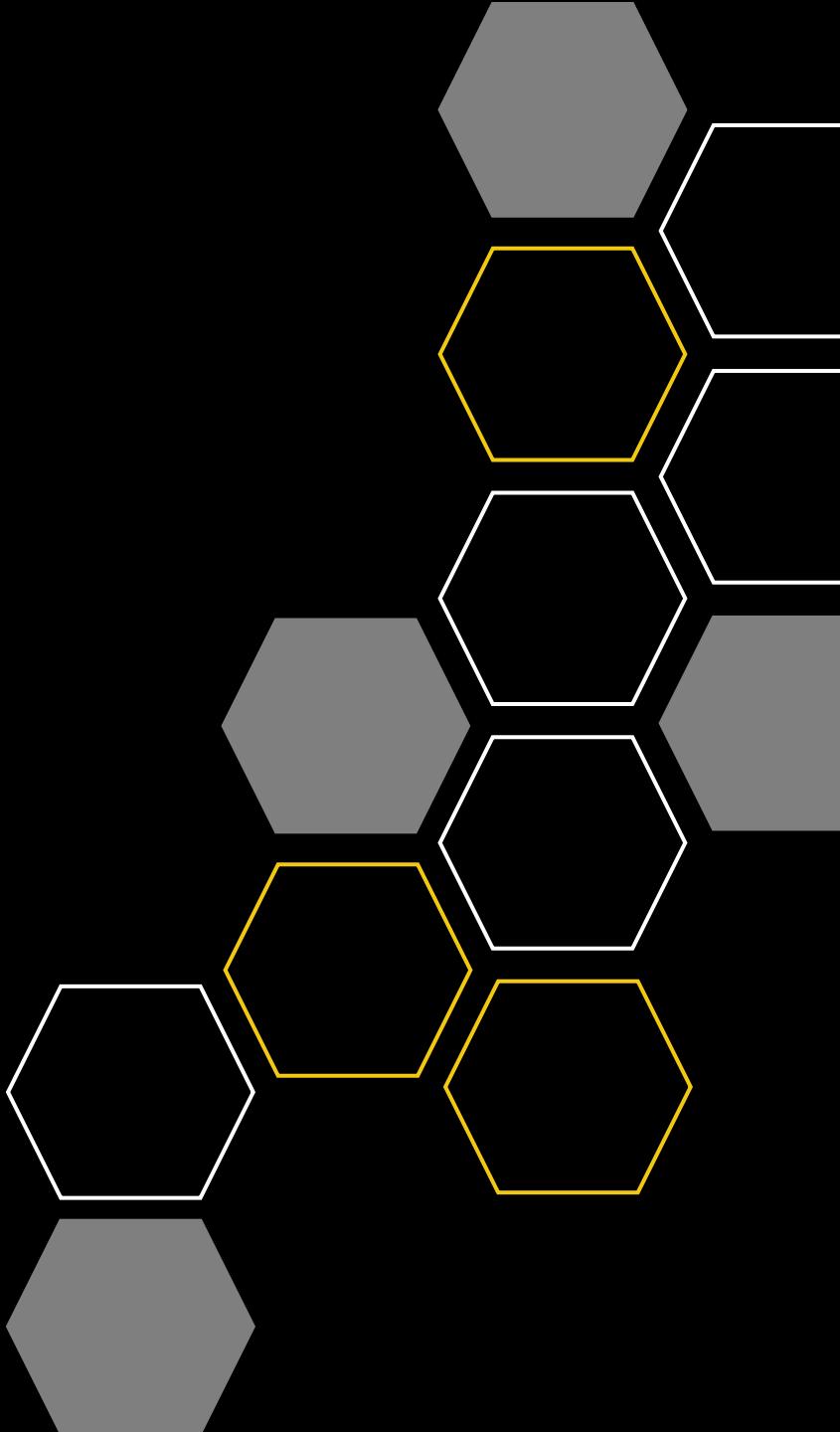
→ Bloats model

Might be worth it to introduce your own dimension for date, especially if you have multiple date columns

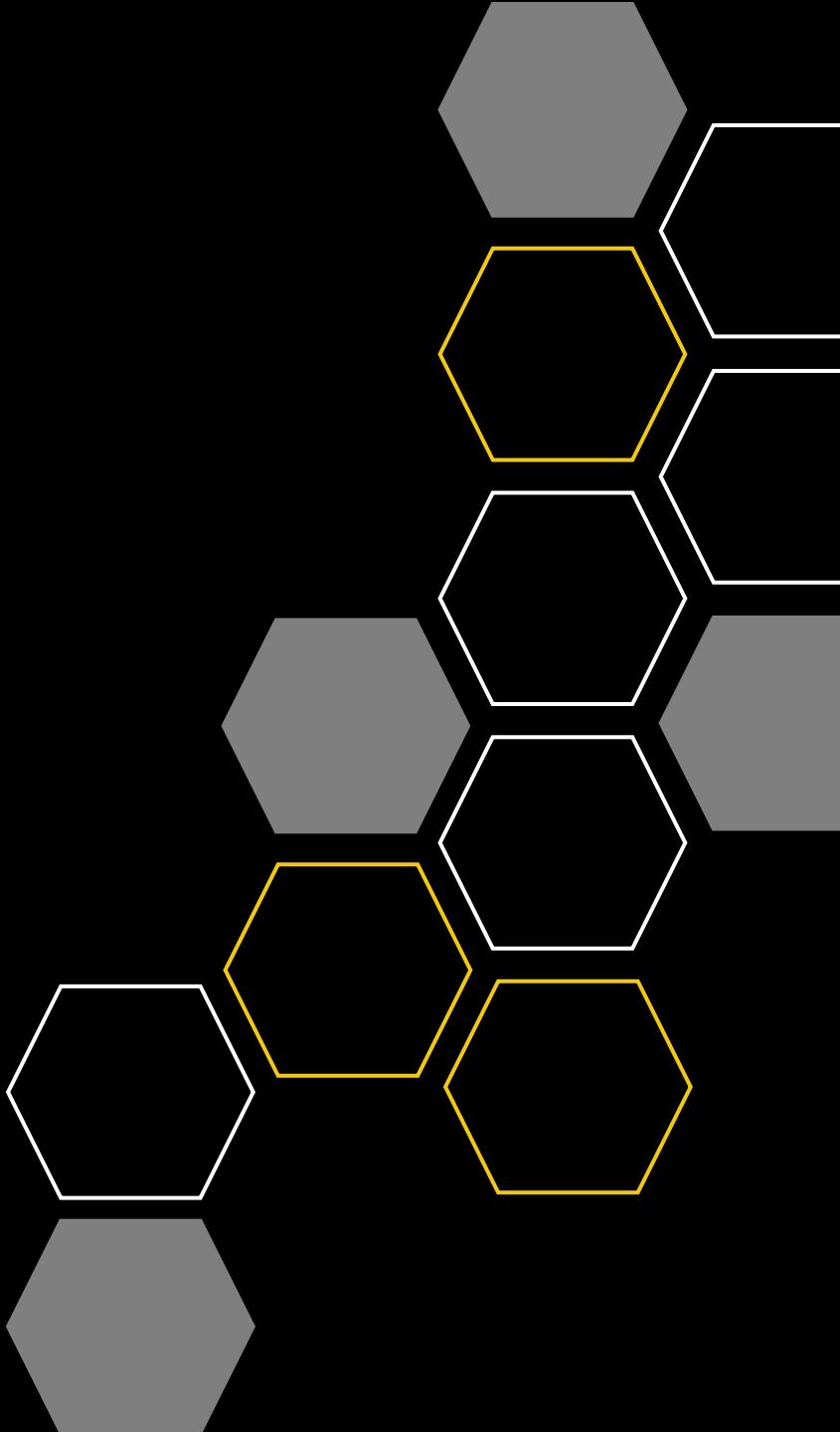


Auto date/time

Marc Lelijveld



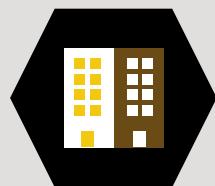
Slowly changing dimensions



Slowly changing dimensions

Although dimension data does not change often (hence 'slowly'), it might change.

If it does, it's important to consider historical reporting:

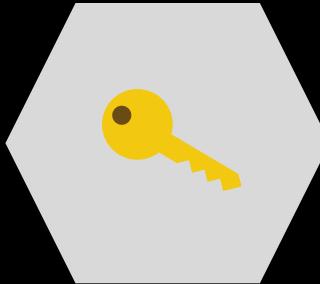


- A manager switches stores – let's say they move from A to B. Are all the sales in store A now counted against the manager until they switched? Or against the new manager? What about for the sales in store B?



- A salesperson switching regions – let's say they move from South to North. Are all their sales in South now counted as if they happened in the North region, including the sales that happened before the Salesperson changed Regions?

Let's talk about keys



Surrogate key (SK)

- Key generated in the data warehouse
- Unique identifier
- Typically auto-generated (identity)



Alternate key (AK)

- Primary key from the source system that loads the data warehouse

Types of attributes on slowly changing dimensions

Type 0: Retain. No changes allowed. Once written, attribute does not change.

Type 1: Overwrite. Does not track historical data.

Customer SK	Customer AK	Name	State
1	C_1	A	WA



Customer SK	Customer AK	Name	State
1	C_1	A	NY

Type 2: Add new row. Tracks historical data by creating multiple records for the same alternate key (**most common**).

Customer SK	Customer AK	Name	State
1	C_1	A	WA



Customer SK	Customer AK	Name	State	Start Date	End Date	Current
1	C_1	A	WA	2000-01-01	2020-12-31	0
2	C_1	A	NY	2021-01-01	Null	1

optional

Type 3: Limited historical data.

Customer SK	Customer AK	Name	State
1	C_1	A	WA



Customer SK	Customer AK	Name	Original State	Current State	Effective Date
1	C_1	A	WA	NY	2021-01-01

Working with slowly changing dimensions in Power BI

Go back to your requirements

If the Salesperson moves – are historical sales still reported in the original Region?

Yes?

Then you're in luck as Power BI already handles this when using type 2 slowly changing dimension.

No?

You need to do something:

- Are you sure? This can lead to confusing situations and unhappy sales managers: you might "take away" sales that happened in their region but suddenly start showing as if it always happened in another region.
- Either use type 1 slowly changing dimension (no history).
- Or handle make some changes in the source (or Data Transformation step) and use USERELATIONSHIP.
- Alternatively, you can do this with a calculation group.

Calculation groups

Address issue in complex models where there is a proliferation of redundant measures using the same calculation.

Provides a way to “change the type” of calculation without adding another measure.

Sales														
DateKey														
ProductKey														
Sales Amount														
<table><tr><td></td><td>Sales month-to-date</td></tr><tr><td></td><td>Sales quarter-to-date</td></tr><tr><td></td><td>Sales year-to-date</td></tr><tr><td></td><td>Cost month-to-date</td></tr><tr><td></td><td>Cost quarter-to-date</td></tr><tr><td></td><td>Cost year-to-date</td></tr><tr><td></td><td>....</td></tr></table>		Sales month-to-date		Sales quarter-to-date		Sales year-to-date		Cost month-to-date		Cost quarter-to-date		Cost year-to-date	
	Sales month-to-date													
	Sales quarter-to-date													
	Sales year-to-date													
	Cost month-to-date													
	Cost quarter-to-date													
	Cost year-to-date													
													

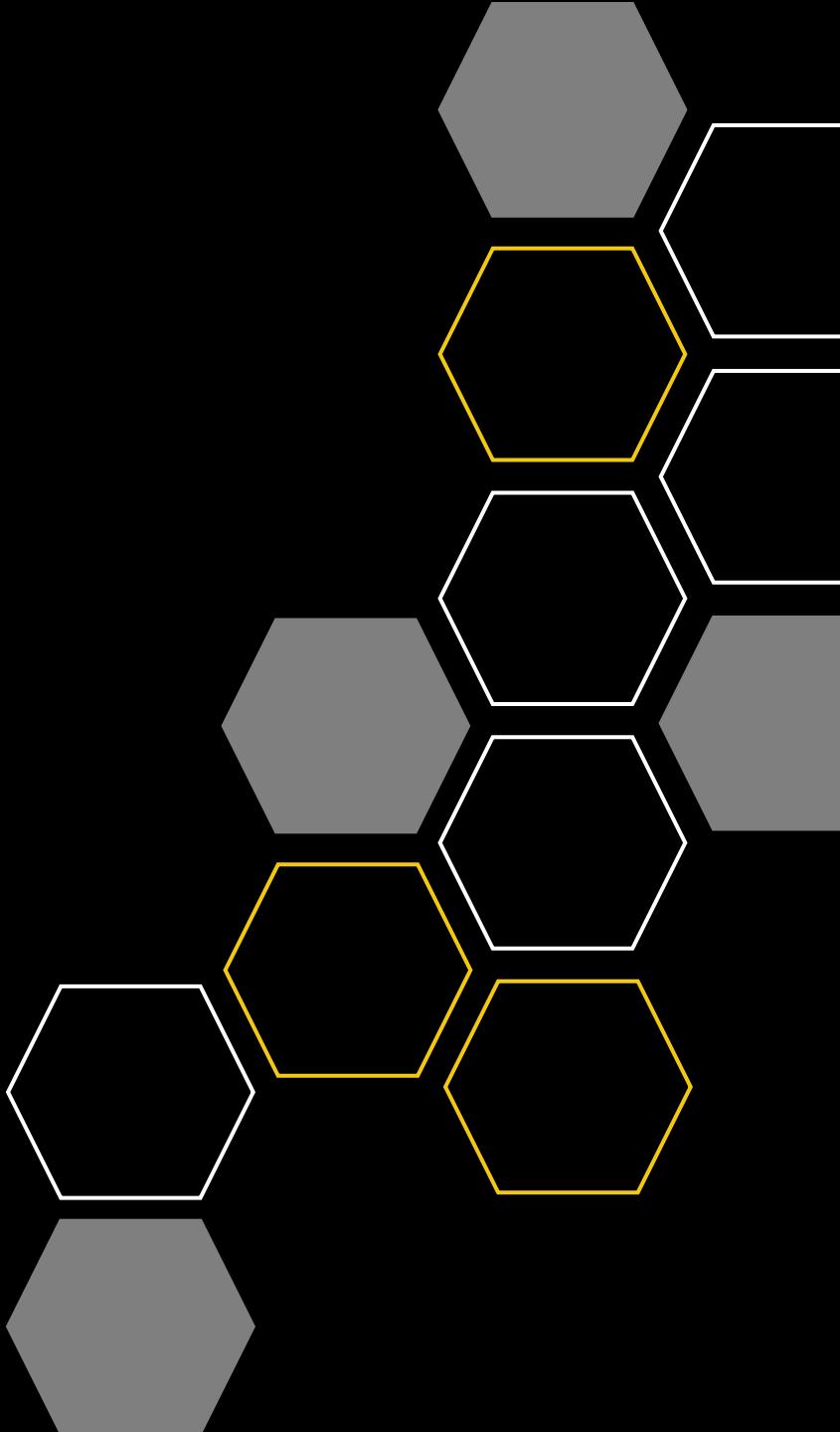


Sales
DateKey
ProductKey
Sales Amount

Time calculation
None
Month-to-date
Quarter-to-date
Year-to-date

Demo: slowly changing dimensions

Jeroen ter Heerdt

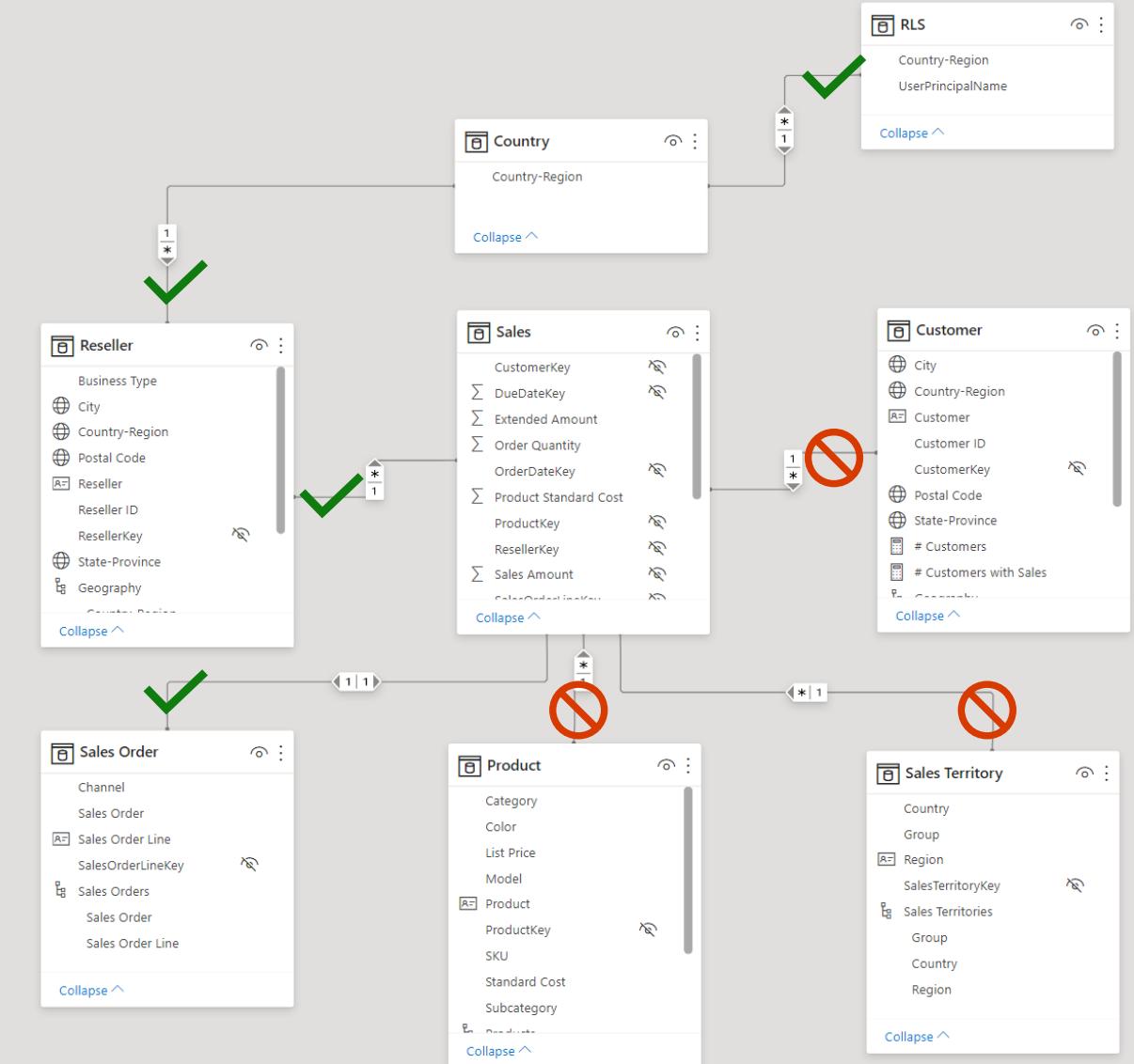


Row level security

Filters mainly the Fact table

Does not filter dimensional tables

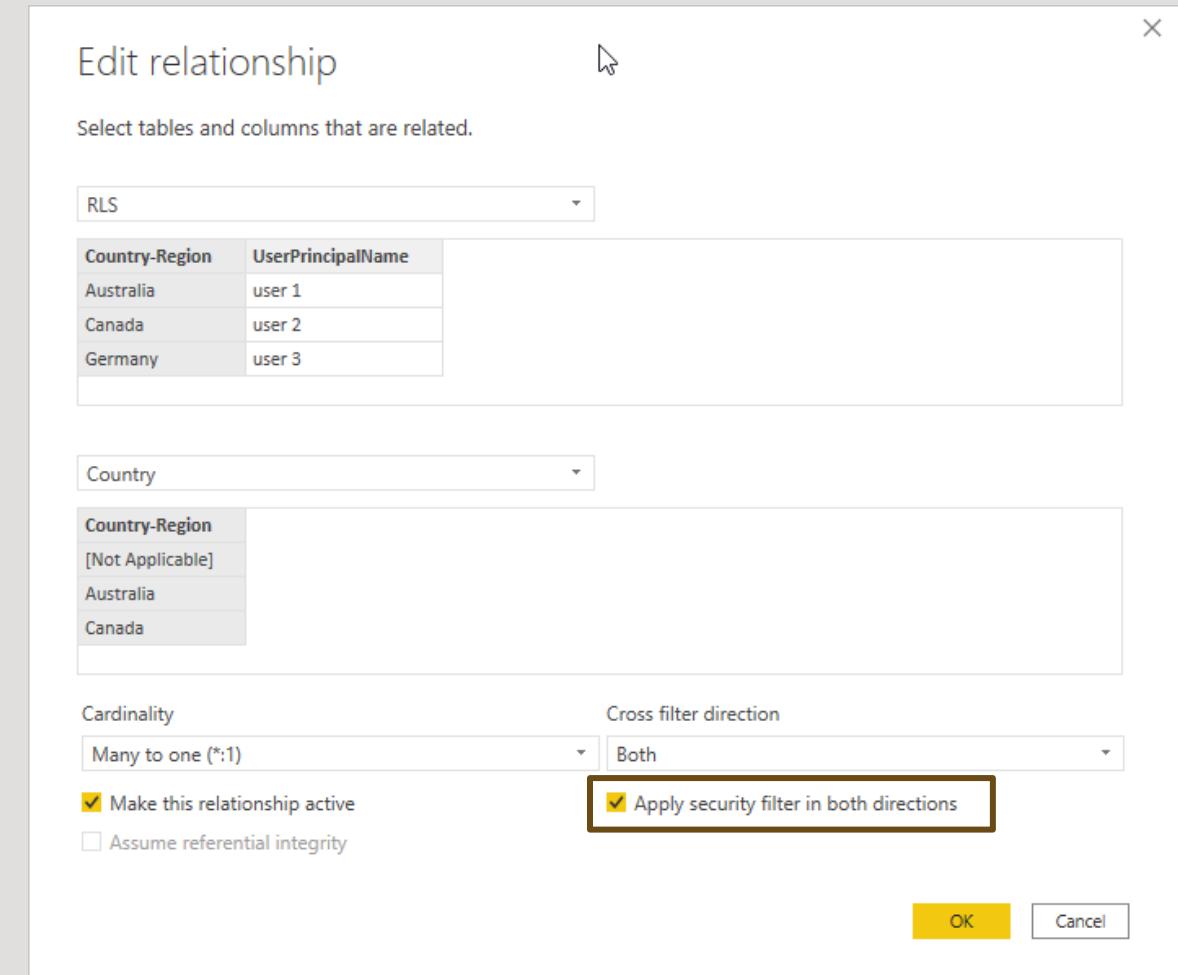
- Except over 1:1 bi-directional relationships
- Possible when adding a RLS filter to each table individually
- Requires a specific setup regarding relationships



Row level security

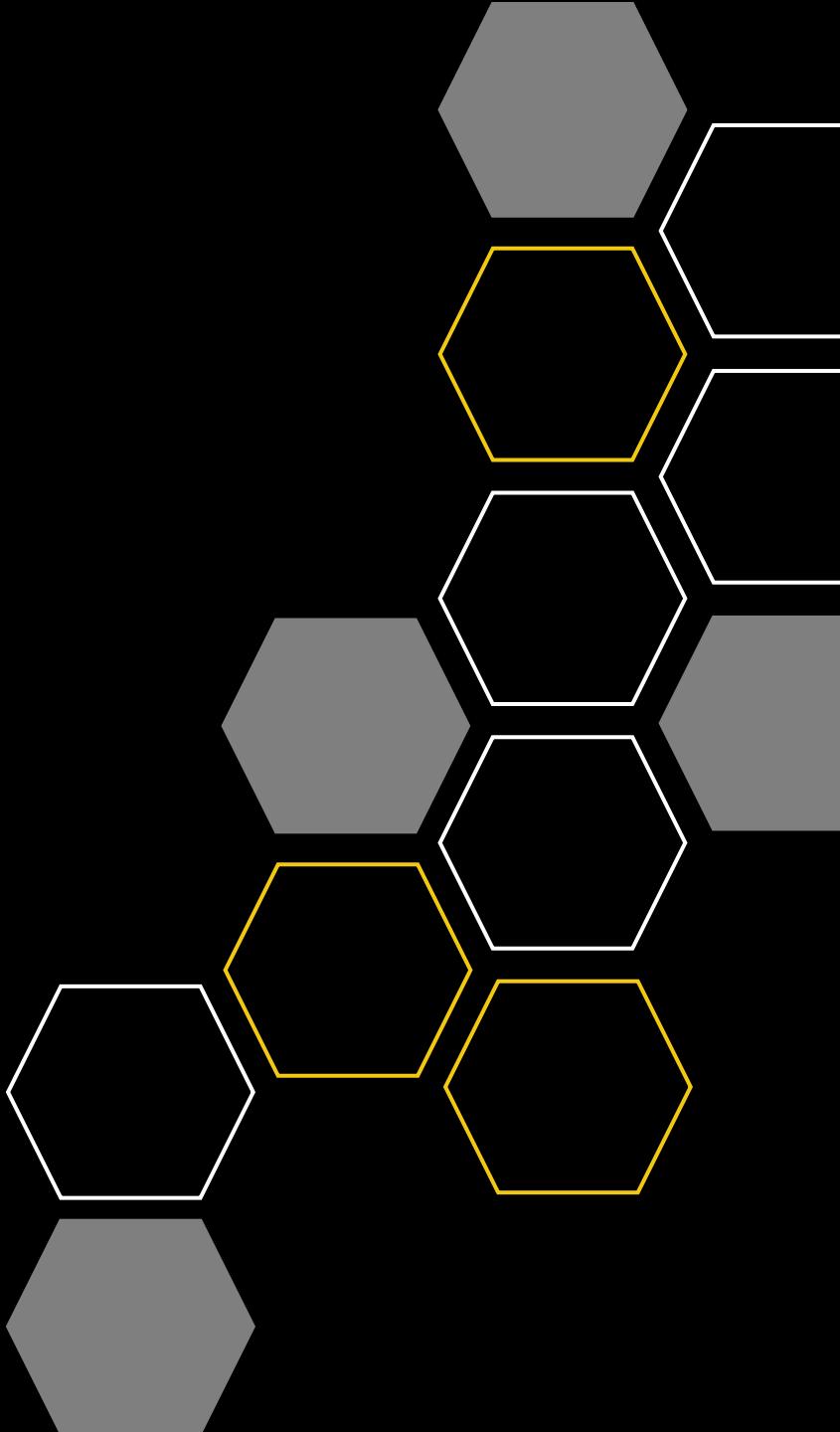
Uses single directional filters by default

- you can set 'Apply the Filter Direction when using Row Level security' on a bi-directional relationship to force RLS to use bi-directional filtering
- Useful for dynamic row level security



Row level security

Marc Lelijveld



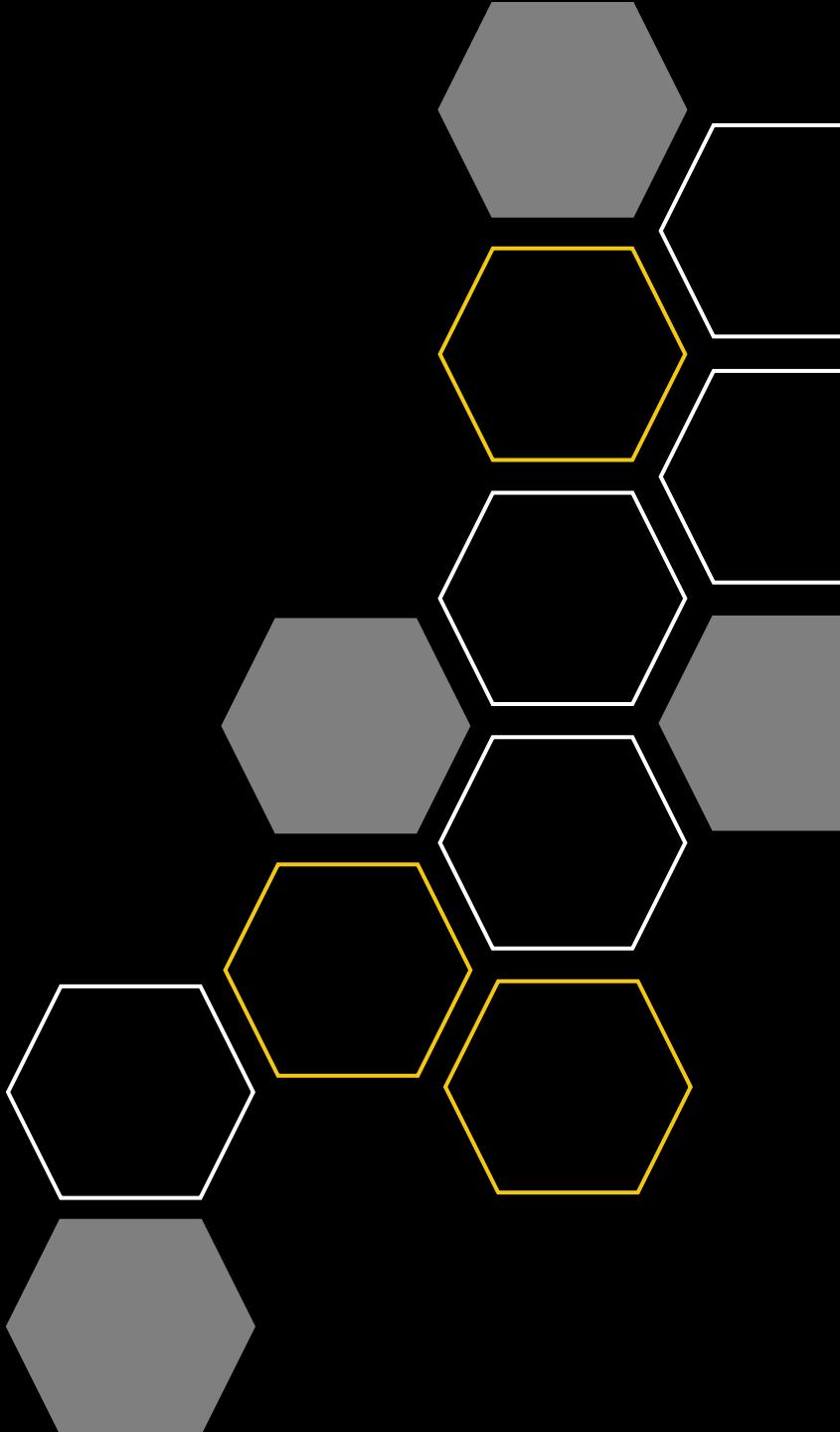
Extend your own data model

A large part of your users should only be able to see internet sales.

bit.ly/pbigws3



Implementing aggregations



Aggregations

Store data at a higher level of granularity than the original table

The following aggregations are available:
count, groupby, max, min, sum, and count table rows

- Aggregated data is stored in-memory (imported), details are accessed through DirectQuery
- You can create the aggregated table in the Data Transformations (Power Query) or in your source (preferred)



Aggregations and storage mode

Aggregation, that hit based on relationships, require *regular* relationships.

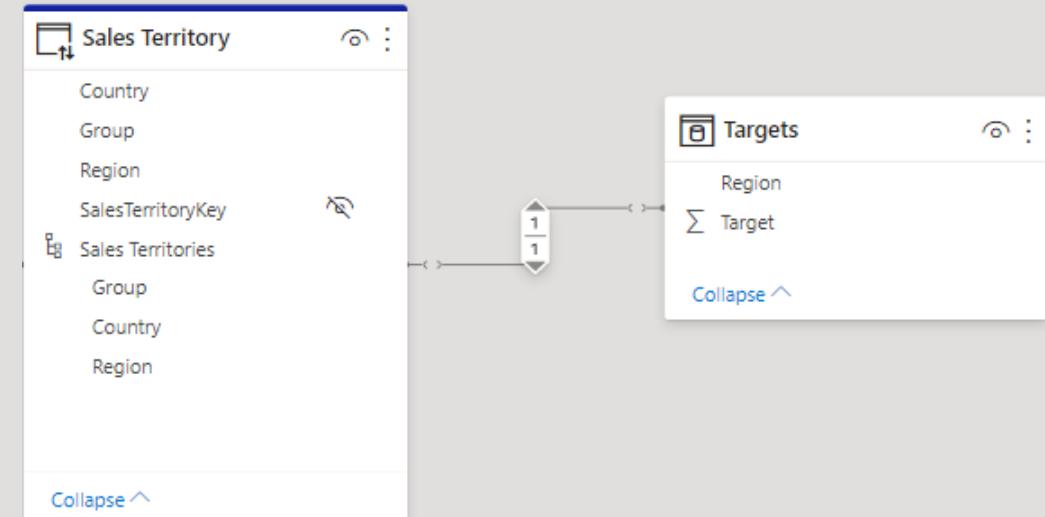
Regular relationships include the following storage mode combinations, where both tables are from a single source group:

Table on the many side	Table on the 1 side
Dual	Dual
Import	Import or Dual
DirectQuery	DirectQuery or Dual

Relationships: Regular vs limited

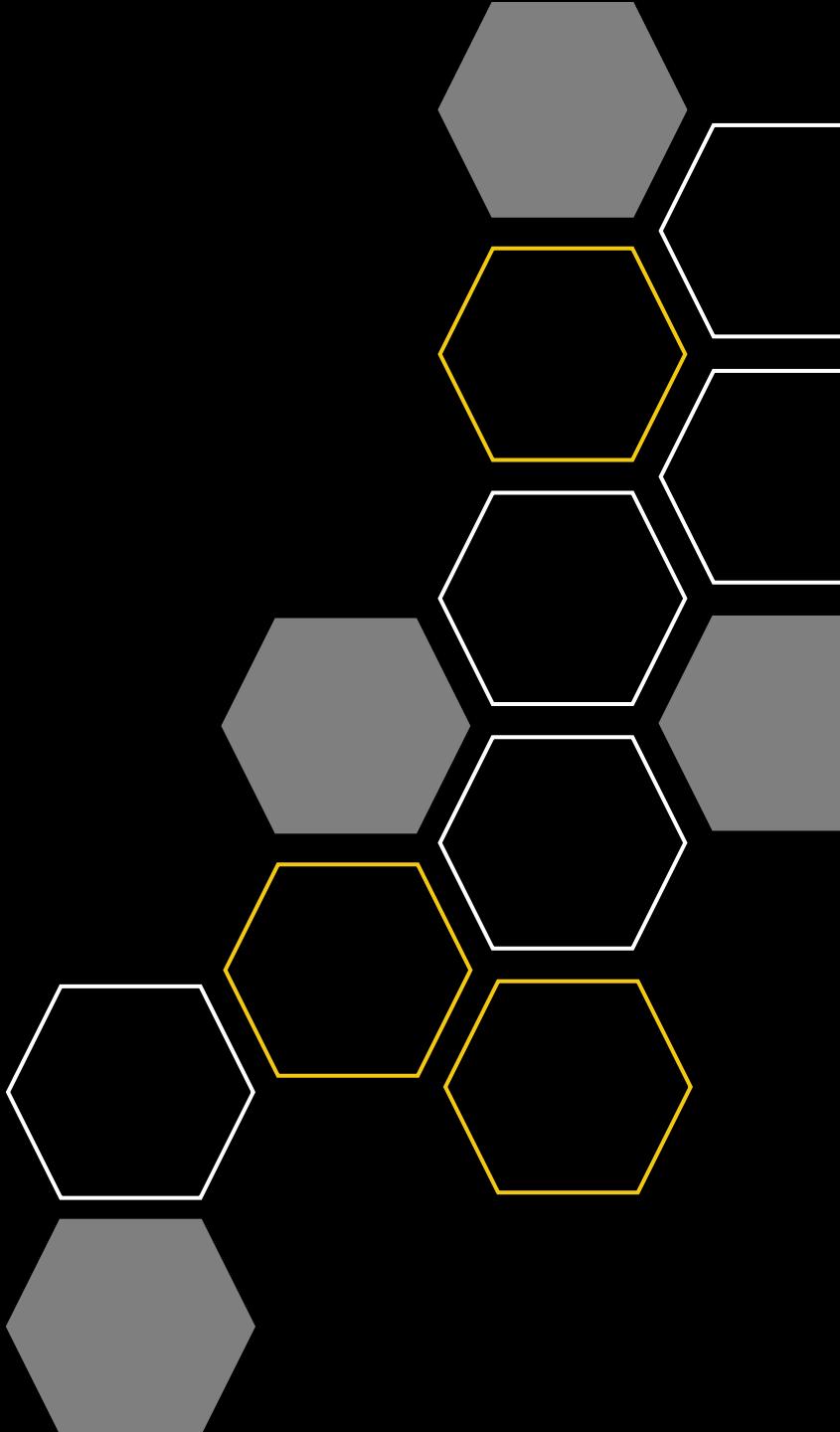
A model relationship is *limited* when there's no guaranteed "one" side. It can be the case for three reasons:

- The relationship uses a Many-to-many cardinality type (even if one or both columns contain unique values)
- The storage mode combination is Import and DirectQuery
- The relationship is cross source group



Defining and using aggregations

Jeroen ter Heerdt



Composite models

Introducing composite models

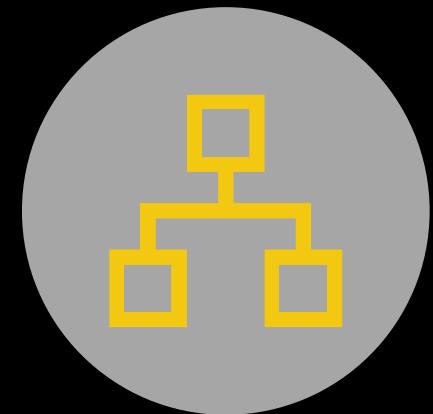
Composite models do either or both of the following actions

- Combine data from one or more DirectQuery sources
- Combines data from DirectQuery and import data

Three related features in Power BI

- Composite models, to allow different connections and storage models in a single data model
- Many-to-many relationships
- Storage modes

Limited relationships might occur, when relationships cross source group



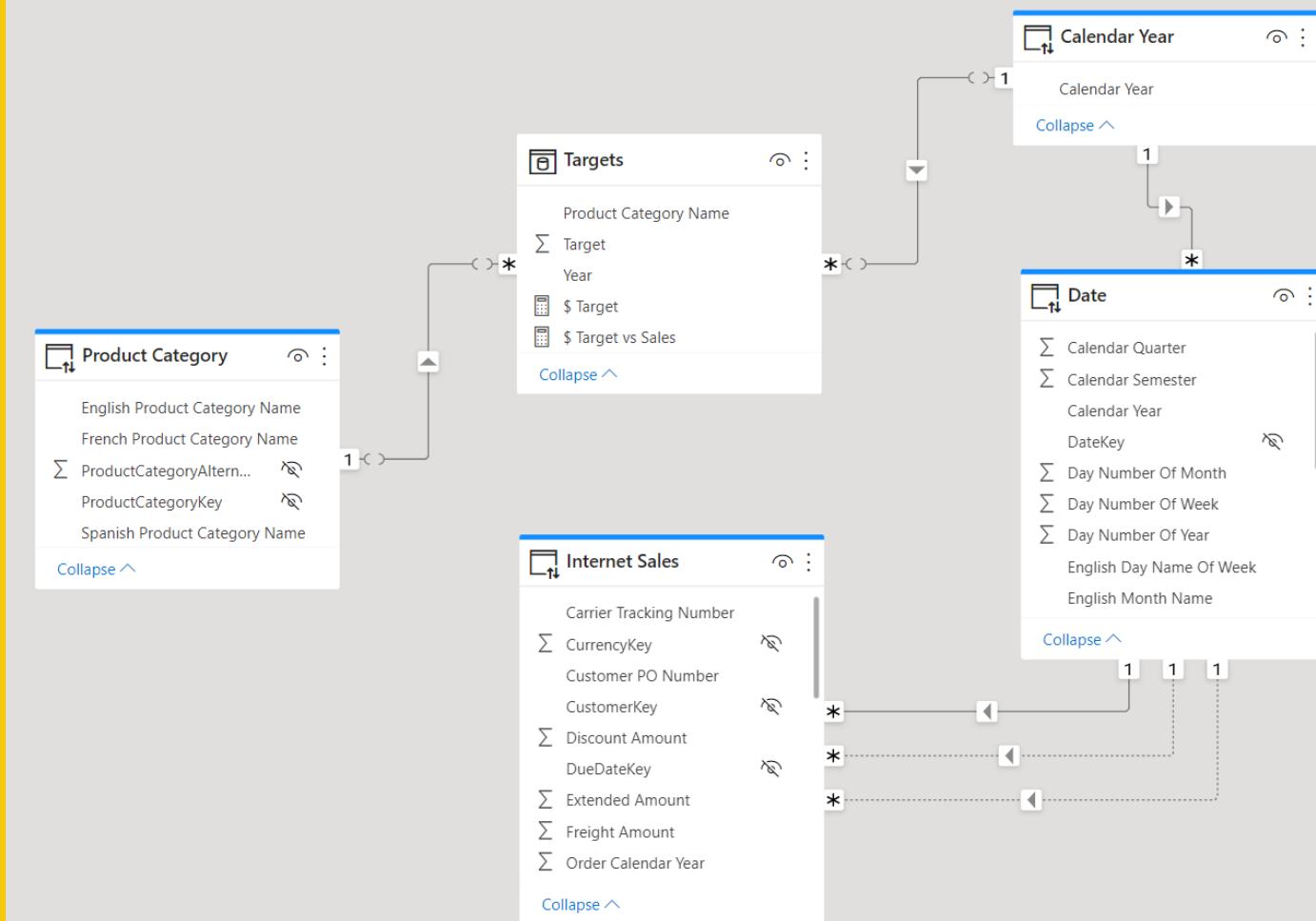
Example, sales model with local target sheet

Example use case for composite models in Power BI

- Centrally maintained data model, containing all sales
- Region has their own sheet with targets in Excel, which they can blend with the central data model

Source group – everything loaded from one DirectQuery source

All import tables and calculated tables are in one source group

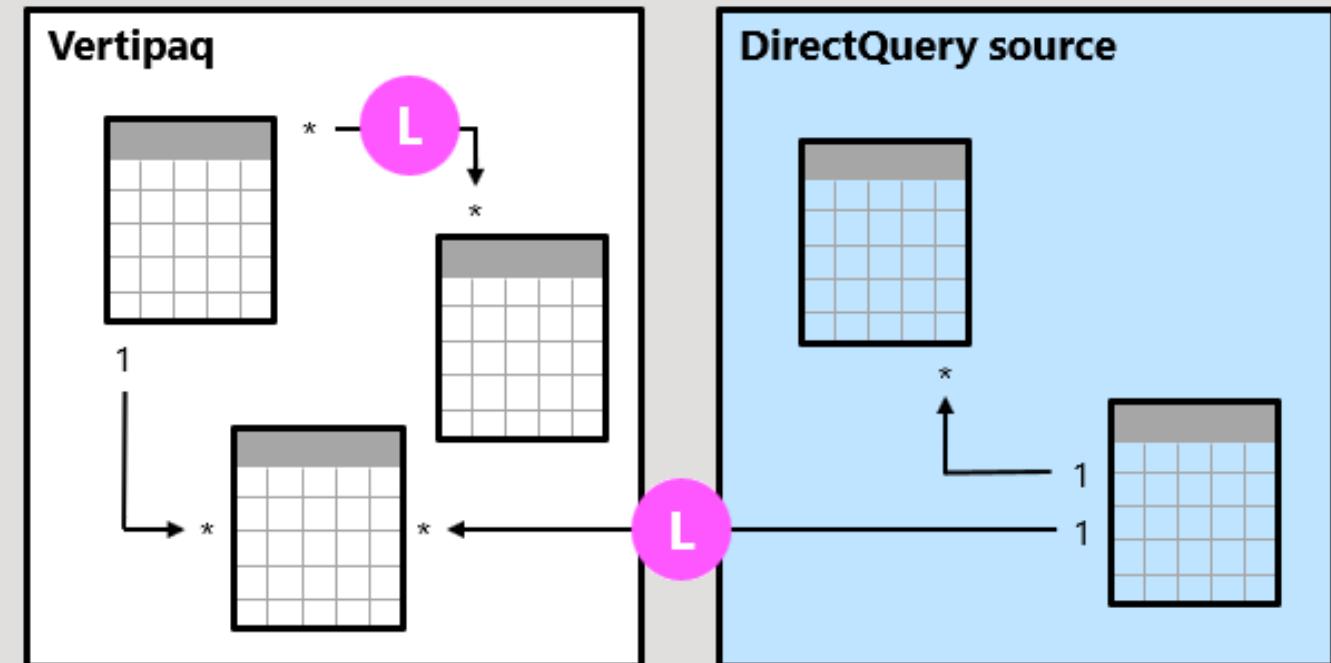


Limited relationships revisited

A model relationship is *limited* when there's no guaranteed "one" side

It can be the case for two reasons:

- The relationship uses a Many-to-many cardinality type (even if one or both columns contain unique values)
- The relationship is cross source group (which can only ever be the case for Composite models)
- It can be a cross-storage mode but does not have to be



Composite models

Demo

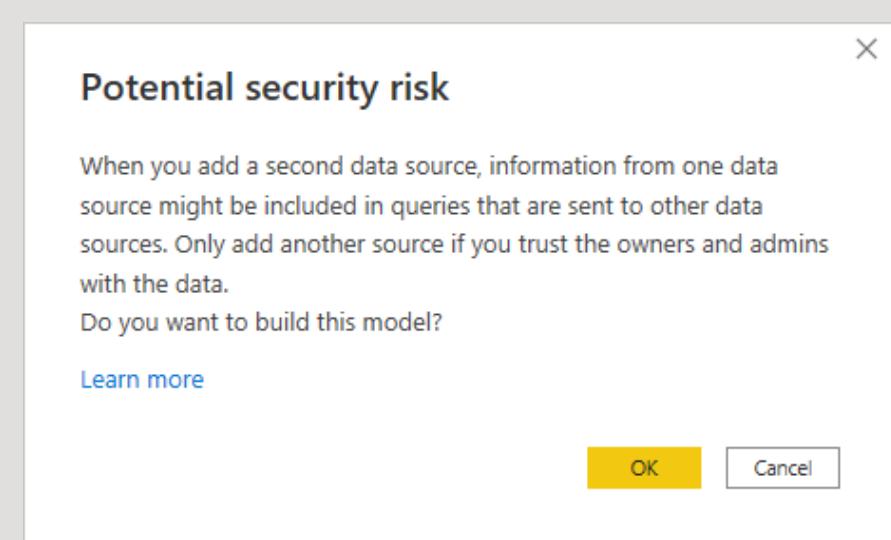
Security implications

Composite models do either or both of the following actions

- A query sent to one data source can include data values that have been retrieved from another data source

Potentially

- Any administrator of the data sources, who can view the traces or audit logs could view this information, without permissions to the original data source
- The encryption setting for each source should be considered, to avoid breaching encryption while the data is included in a query sent to another unencrypted source



Performance implications

Composite models might have slower performance

- **General limitations of DirectQuery** apply, make sure the back-end data source has sufficient resources to provide best performance as possible
- Query execution potentially needs to **cross sources**, where values from one source needs to be evaluated in another source (rich set of values in the WHERE clause)
- **Difference in granularity** of data across sources, where on-demand aggregation is required before displaying data in a visual
- **Multiple queries** must be executed to retrieve results (typically with DistinctCount)



Composite models using Power BI datasets and Azure Analysis Services

With the ability to connect to Power BI datasets and Azure Analysis Services models using DirectQuery (in preview since December 2020) you can now build a composite model using these sources.

Note the following design goals / non-goals for this feature:

Goals:

- **Extend:** Add data to enterprise semantic model
- **Enrich:** Add measure, calculation, formatting to enrich enterprise semantic model without adding any data
- **Overview:** disjoined datasets to give business overview ('dashboard')

Non goals:

- Combine two or more sources by joining large shared dimension tables
- Combine large dimension table and large fact table from different sources

Composite models 2

Demo

Wholesale and non-wholesale queries

Remote vs local model

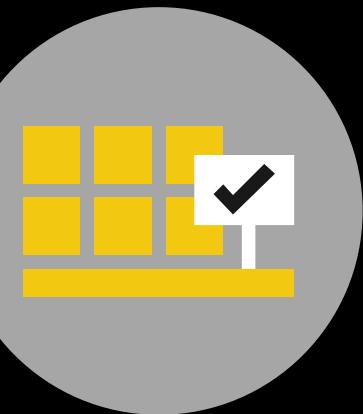
Remote model – a model that serves as a data source for the current (local) model

- Remote / vs local table – a remote table is a table that is sourced from a remote model.
Any table that is not from a remote model is considered local
- Remote measures are treated as a black box to the composite model: a calc groups might be applied, or not, depends on wholesale vs non-wholesale



Wholesale vs non-wholesale

- **Wholesale:** depends on objects in single remote source group
 - Calculated columns defined on a table from a remote source group must be wholesale
- **Non-wholesale:** depends on objects in more than one source group



Calculation groups and measures in a composite model

Depending on the configuration, calculation groups and measures have interesting interactions

Measure location	Is measure wholesaleable	Remote calc group applied
Remote	Yes	Yes
Local	Yes	Yes
Local	No (either because it depends on more than one source group or just the local source group)	No – but can be indirectly

Remember: measures might be impacted by remote calculation groups, when applied

Calculation groups and wholesale / non-wholesale measures

Demo

Storage modes

Different types of storage modes

Three storage modes

- **Import** – data cached in the model
- **DirectQuery** – queries are submitted to the back-end data source
- **Dual** – can act in both above storage modes, depending on query context

Configuring storage modes

- Storage modes are set on table level
- Setting storage mode to Import is an irreversible operation
- Data in DirectQuery mode cannot be displayed in the data tab

The screenshot shows the Power BI Desktop interface with the 'Data' tab selected. Three tables are visible in the data model: 'Product Subcategory', 'Product', and 'Internet Sales - Agg'. The 'Internet Sales - Agg' table is currently selected, as indicated by the orange border around its preview and the orange selection bar at the bottom. The 'Properties' pane on the right is open, showing various configuration options for the selected table. A red box highlights the 'Storage mode' dropdown in the 'Advanced' section of the properties pane, which is set to 'DirectQuery'. Other visible properties include 'Name' (set to 'Internet Sales'), 'Description' (set to 'Enter a description'), 'Synonyms' (set to 'Enter a comma-separated list of synonyms for Q&A'), 'Row label' (set to 'Select a row label'), 'Key column' (set to 'Select a column with unique values'), 'Is hidden' (set to 'No'), and 'Is featured table' (set to 'No').

Understanding Dual mode

Example

- Internet Sales → DirectQuery
- Internet Sales - Agg → Import
- Product Category → Dual

Screenshot of Power BI Desktop showing the Data model view with tables and relationships. A red box highlights the 'Tables' icon in the ribbon.

The Data model view displays four tables:

- Product Subcategory**:
 - English Product Subcategory Name
 - French Product Subcategory Name
 - ProductCategoryKey
 - ProductSubcategoryAltKey
 - ProductSubcategoryKey
 - Spanish Product Subcategory Name
- Product**:
 - Arabic Description
 - Chinese Description
 - Class
 - Color
 - Days To Manufacture
 - Dealer Price
- Internet Sales - Agg**:
 - Count
 - Order Calendar Year
 - ProductSubcategoryKey
 - Sales Amount
- Internet Sales**:
 - Carrier Tracking Number
 - CurrencyKey
 - Customer PO Number
 - CustomerKey
 - Discount Amount
 - DueDateKey
 - Extended Amount
 - Freight Amount
 - Order Calendar Year

A relationship exists between Product Subcategory and Product. Another relationship exists between Internet Sales - Agg and Internet Sales.

The Properties pane on the right shows the following settings for the Internet Sales table:

- Name: Internet Sales
- Description: Enter a description
- Synonyms: Enter a comma-separated list of synonyms for Q&A
- Row label: Select a row label
- Key column: Select a column with unique values
- Is hidden: No
- Is featured table: No
- Storage mode: DirectQuery (highlighted with a red box)

Benefits

Benefits to choose one or the other

- Improve query performance – cache data for faster end-user performance
- Data refresh optimization – no need to refresh for non-cached data
- Near-real time requirements – reduce query latency when in DirectQuery mode
- Large datasets – choose to not import certain data



Caches and DirectQuery

Risks of mixing storage modes

- Avoid mismatch in results when query bits DirectQuery compared to Import
- Data cached (import) could be behind compared to DirectQuery data
- Make sure cached data is kept **in sync** – regularly refresh!



Behavior of Dual mode

Query context defines storage mode used

- Sales per product subcategory, both tables will use Import mode
- Sales per product, both tables will hit DirectQuery
- Extended Amount per product subcategory, both tables will hit DirectQuery

Screenshot of Power BI Desktop showing the Data model view.

The Data model view displays four tables:

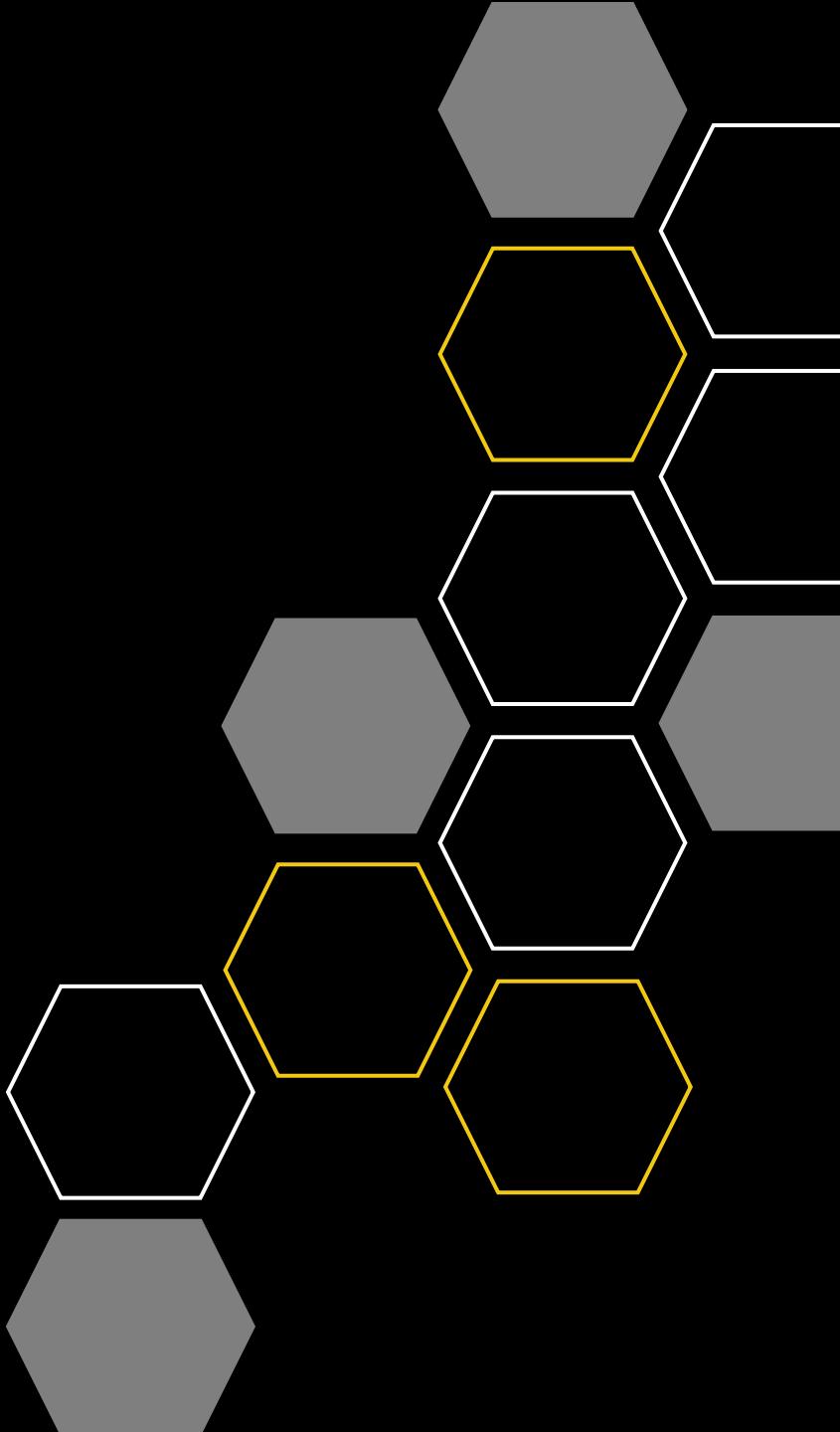
- Product Subcategory**:
 - English Product Subcategory Name
 - French Product Subcategory Name
 - ProductCategoryKey
 - ProductSubcategoryAltKey
 - ProductSubcategoryKey
 - Spanish Product Subcategory Name
- Product**:
 - Arabic Description
 - Chinese Description
 - Class
 - Color
 - Days To Manufacture
 - Dealer Price
- Internet Sales - Agg**:
 - Count
 - Order Calendar Year
 - ProductSubcategoryKey
 - Sales Amount
- Internet Sales**:
 - Carrier Tracking Number
 - CurrencyKey
 - Customer PO Number
 - CustomerKey
 - Discount Amount
 - DueDateKey
 - Extended Amount
 - Freight Amount
 - Order Calendar Year

The **Internet Sales** table is highlighted with a yellow border, indicating it is the active table in the query context.

The Properties pane on the right shows the following settings for the **Internet Sales** table:

- Name: Internet Sales
- Description: Enter a description
- Synonyms: Enter a comma-separated list of synonyms for Q&A
- Row label: Select a row label
- Key column: Select a column with unique values
- Is hidden: No
- Is featured table: No
- Storage mode: DirectQuery

Limited relationships

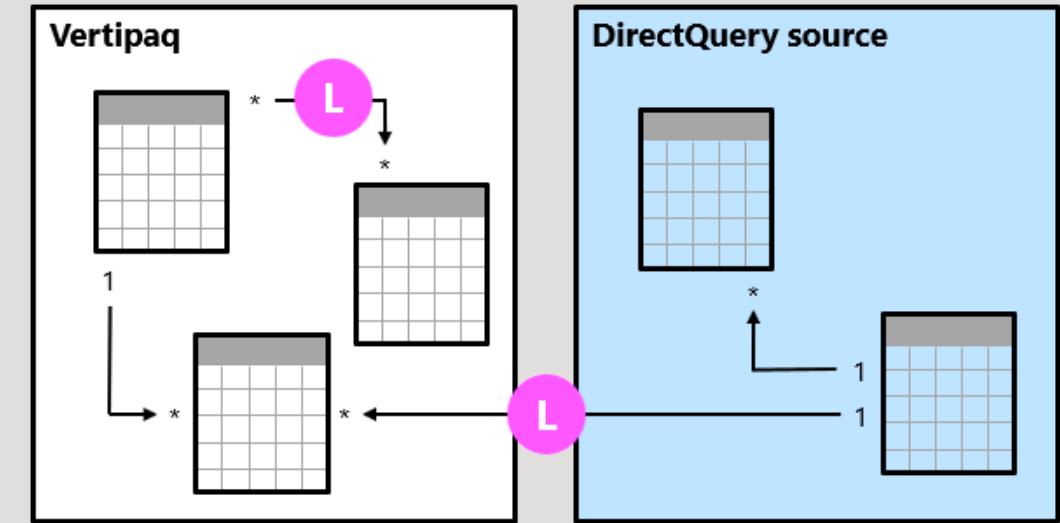


Regular vs limited relationships

A model relationship is *limited* when there's no guaranteed "one" side.

It can be the case for two reasons:

- The relationship uses a Many-to-many cardinality type (even if one or both columns contain unique values)
- The relationship is cross source group (which can only ever be the case for Composite models)



Impact of limited relationships

Cross source group relationships have performance implications.

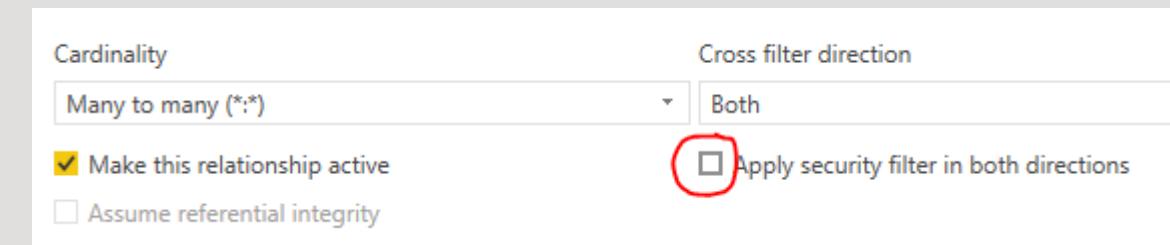
Limited optimization: joins are done on-demand for DirectQuery.

No blank rows: table joins are achieved by using INNER JOIN.

- Blank rows are not added for referential integrity violations

Additional restrictions:

- RELATED DAX function cannot be used to retrieve the 'one' side of the relationship
- Enforcing RLS requires you to check the following checkbox ►

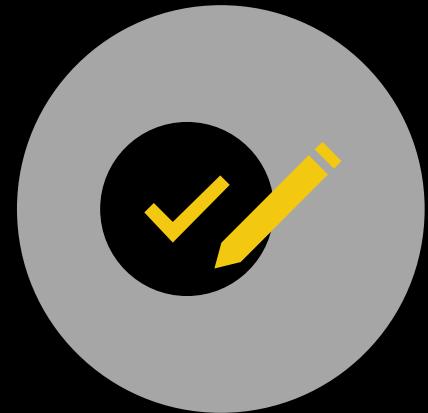


Calculation groups

Introducing calculation groups

Benefits

- Reduce the number of redundant measures and grouping common measure expressions as calculation items
- Avoids duplicating logic in different measures
- Typical use cases are
 - Time-intelligence calculations (YTD / QTD / MTD / ...)
 - Format string change, like currency conversions



Limitations

- Can only be created from external tools in Power BI (Any tool using the XMLA endpoint such as Tabular Editor)
- Object level security on Calculation group items is not supported
- Smart narrative visuals in Power BI are not supported with Calculation Groups
- Be aware of potential unexpected behavior in combination with Composite Models

How calculation groups work

Specific DAX expressions for Calculation Groups

- SELECTEDMEASURE()
- SELECTEDMEASURENAME()
- ISSELECTEDMEASURE()
- SELECTEDMEASUREFORMATSTRING()

Classic measure:

```
MTD =  
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    DATESMTD ( DimDate[Date] )  
)
```

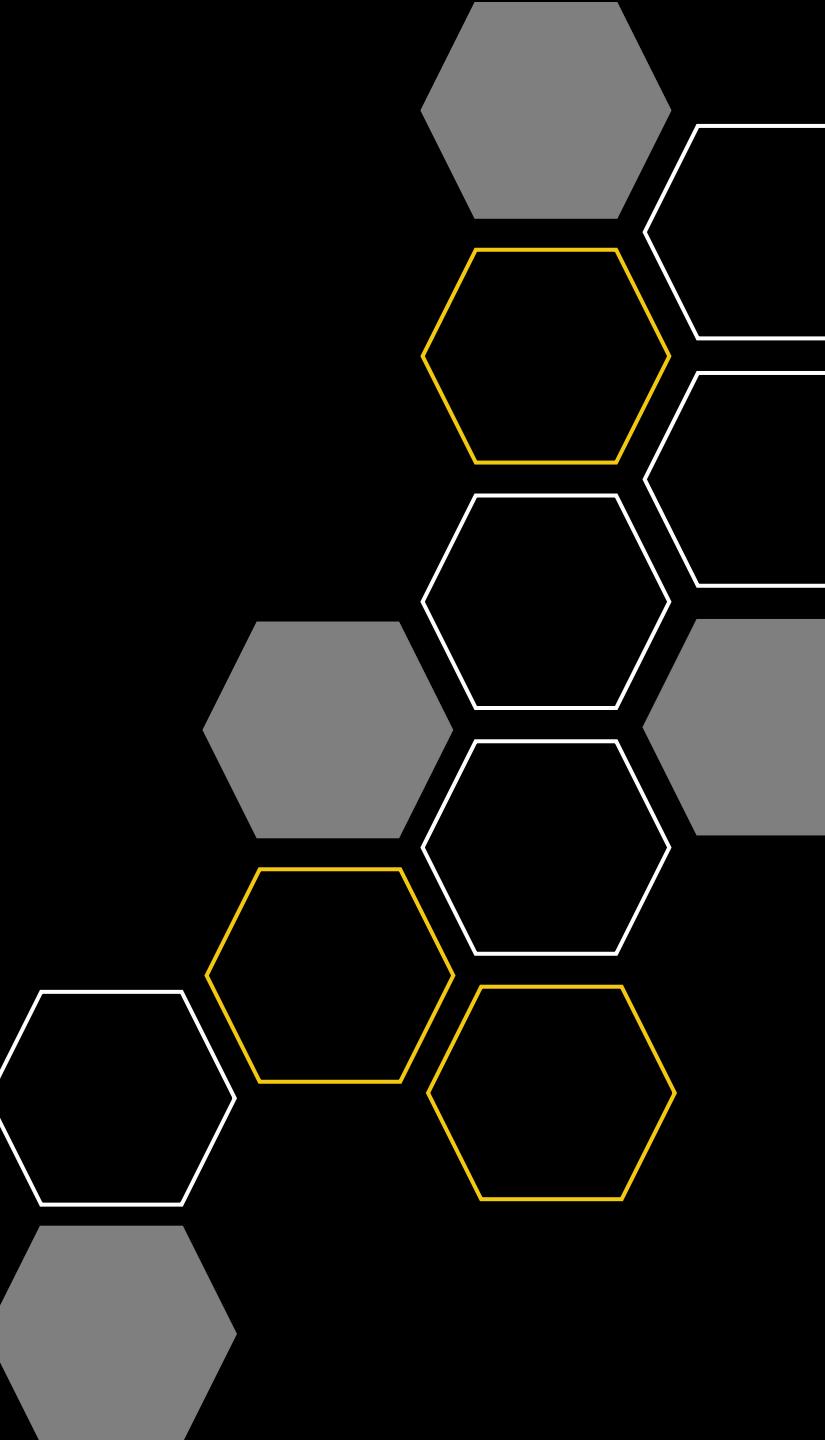
Dynamic measure context MTD with Calculation Group:

```
MTD =  
CALCULATE (  
    SELECTEDMEASURE (),  
    DATESMTD ( DimDate[Date] )  
)
```

Create and use calculation groups

Demo

Ambiguous data models



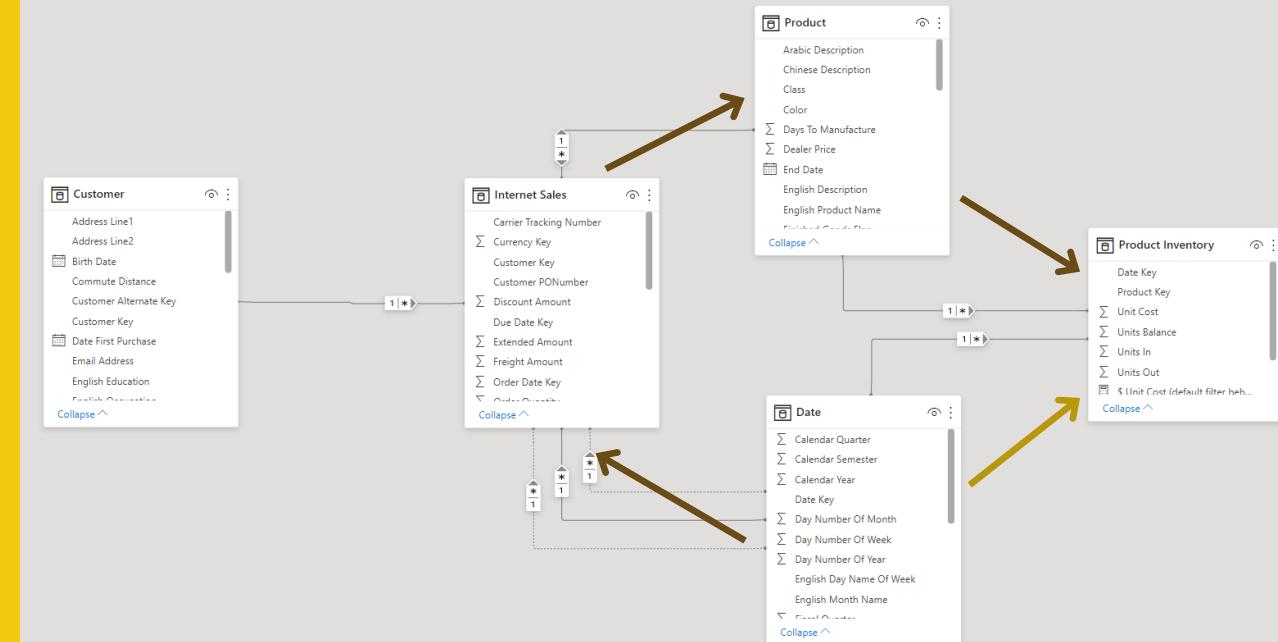
Ambiguous data models

Two filter paths to the same table

→ Leads to unexpected results

Can happen with bi-directional relationships

→ Avoid bi-directional relationships as much as possible

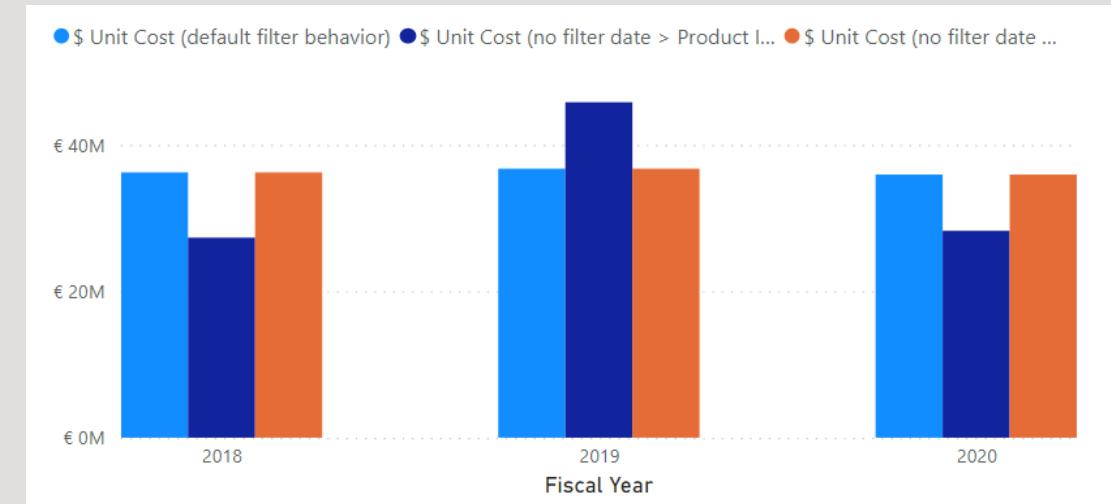


Ambiguous data models 1/2

Can lead to unpredictable results

Each of these measures calculates the same, but removes one of the relationships

By using CROSSFILTER you can change relationships in a measure context

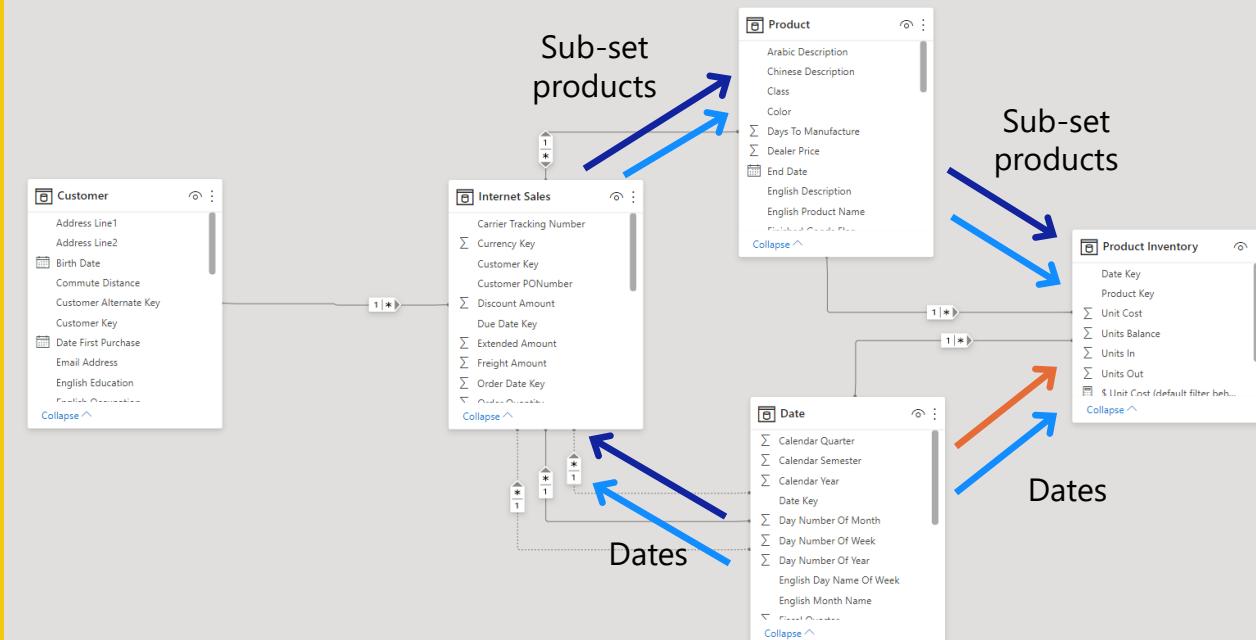
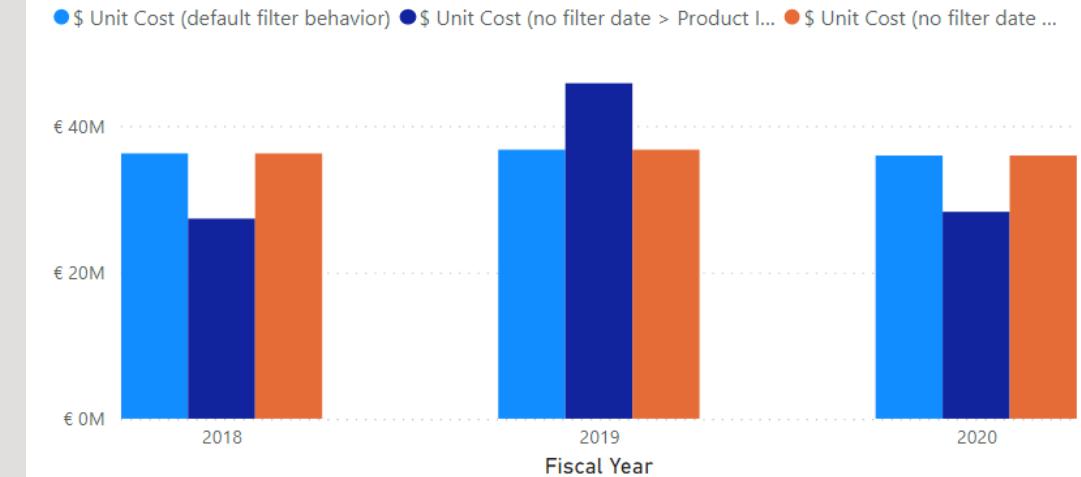


Fiscal Year	\$ Unit Cost (default filter behavior)	\$ Unit Cost (no filter date > Product Inventory)	\$ Unit Cost (no filter date > Internet Sales)
2018	€ 36.244.707,62	€ 27.337.603,34	€ 36.244.707,62
2019	€ 36.758.094,11	€ 45.834.231,75	€ 36.758.094,11
2020	€ 35.954.533,19	€ 28.267.611,68	€ 35.954.533,19
Total	€ 108.957.334,92	€ 108.957.334,92	€ 108.957.334,92

Ambiguous data models 2/2

Unexpected filter behavior

- Result will be filtered by a subset of products as part of the Internet Sales (dark blue filter path)
- Result will be filtered by a subset of dates (orange filter path)



Data groups and hierarchies

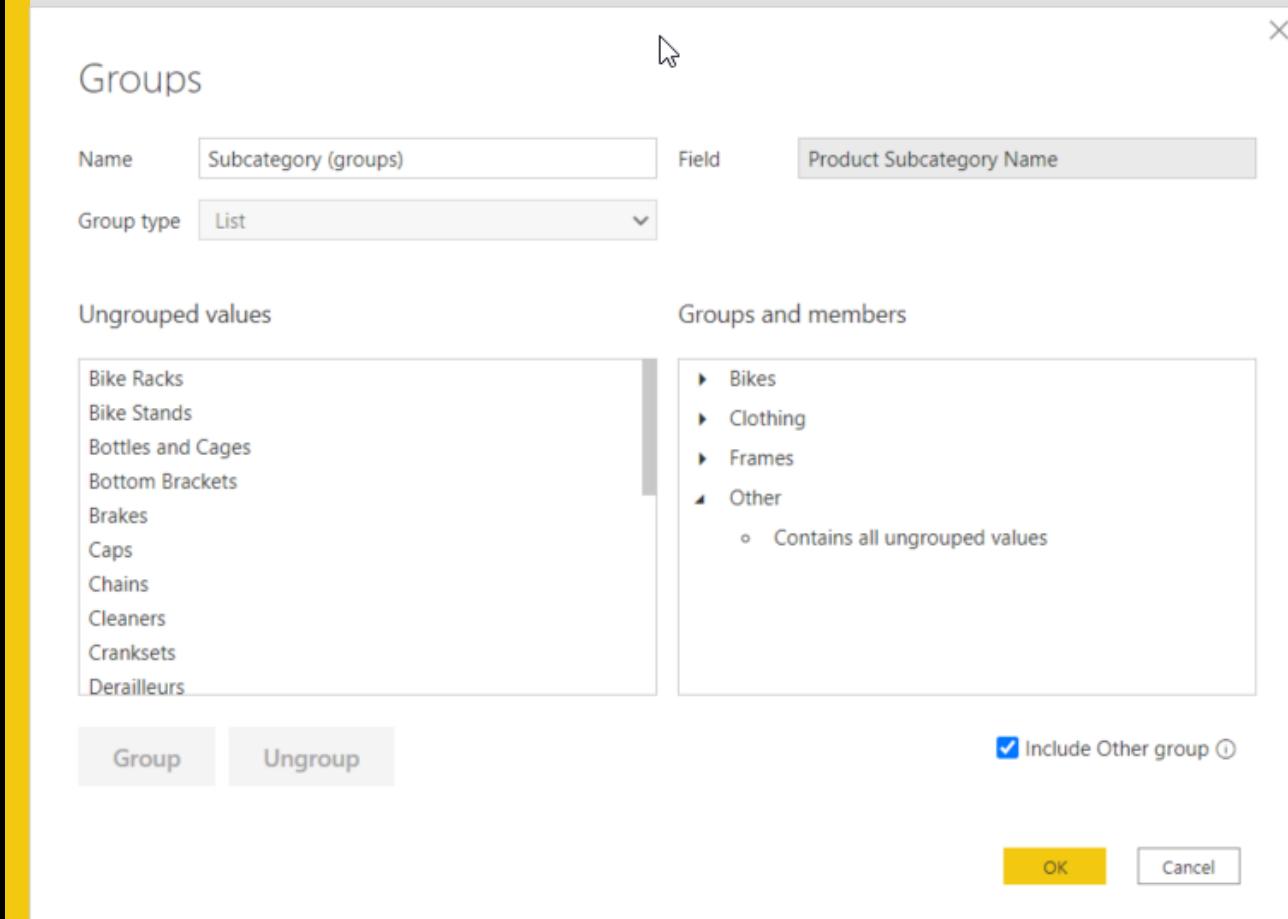


Data groups

- Creates a new model object
- Helps to limit number of dimensional values in a visual presentation

Common use cases:

- Top 3 [groups] & others
- Show % revenue for core business [groups] compared to others

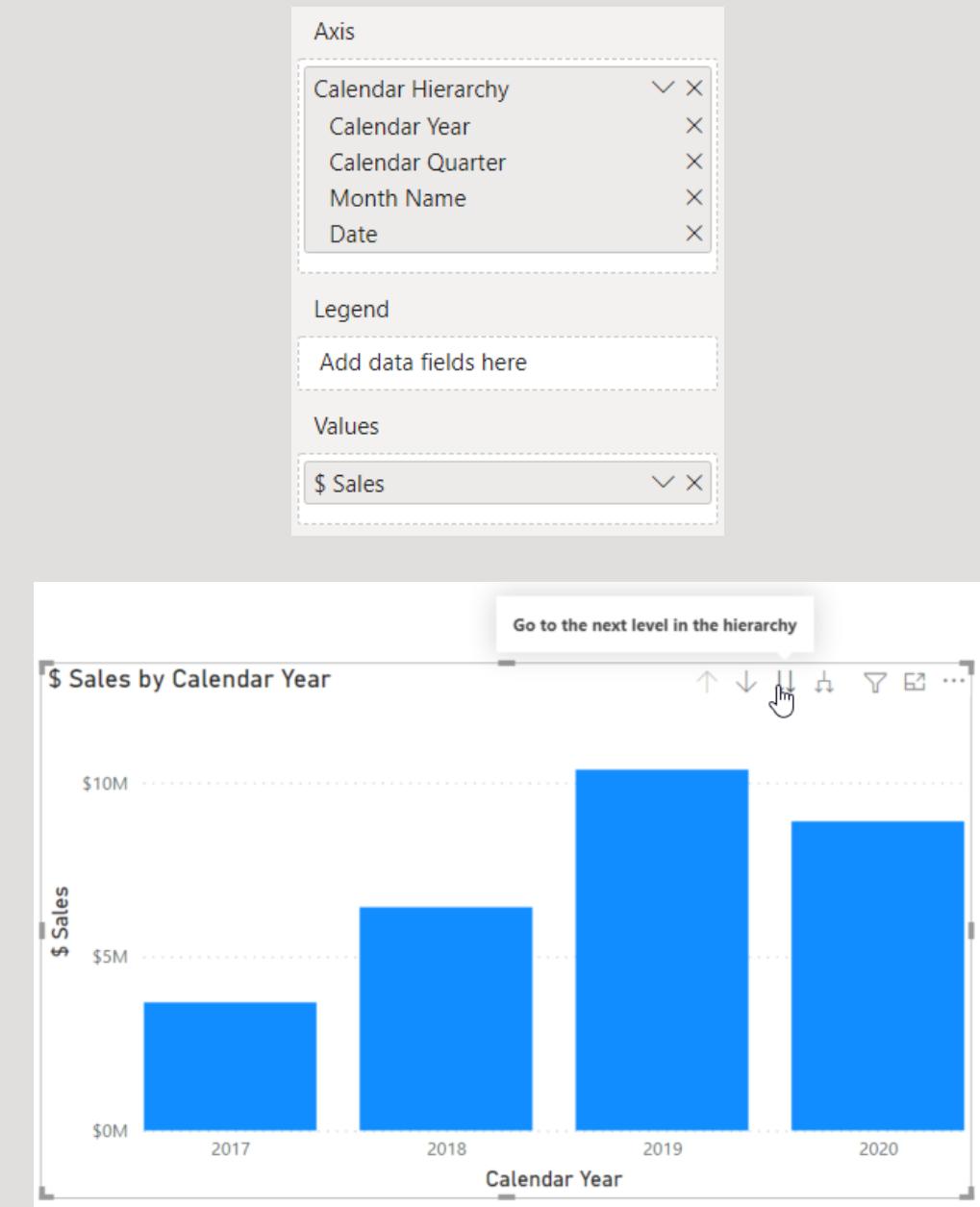


Hierarchies

- Benefit in visual drill up/down
- Consistency across visuals & pages
- With persistent hierarchies, the engine potentially can pre-calculate results

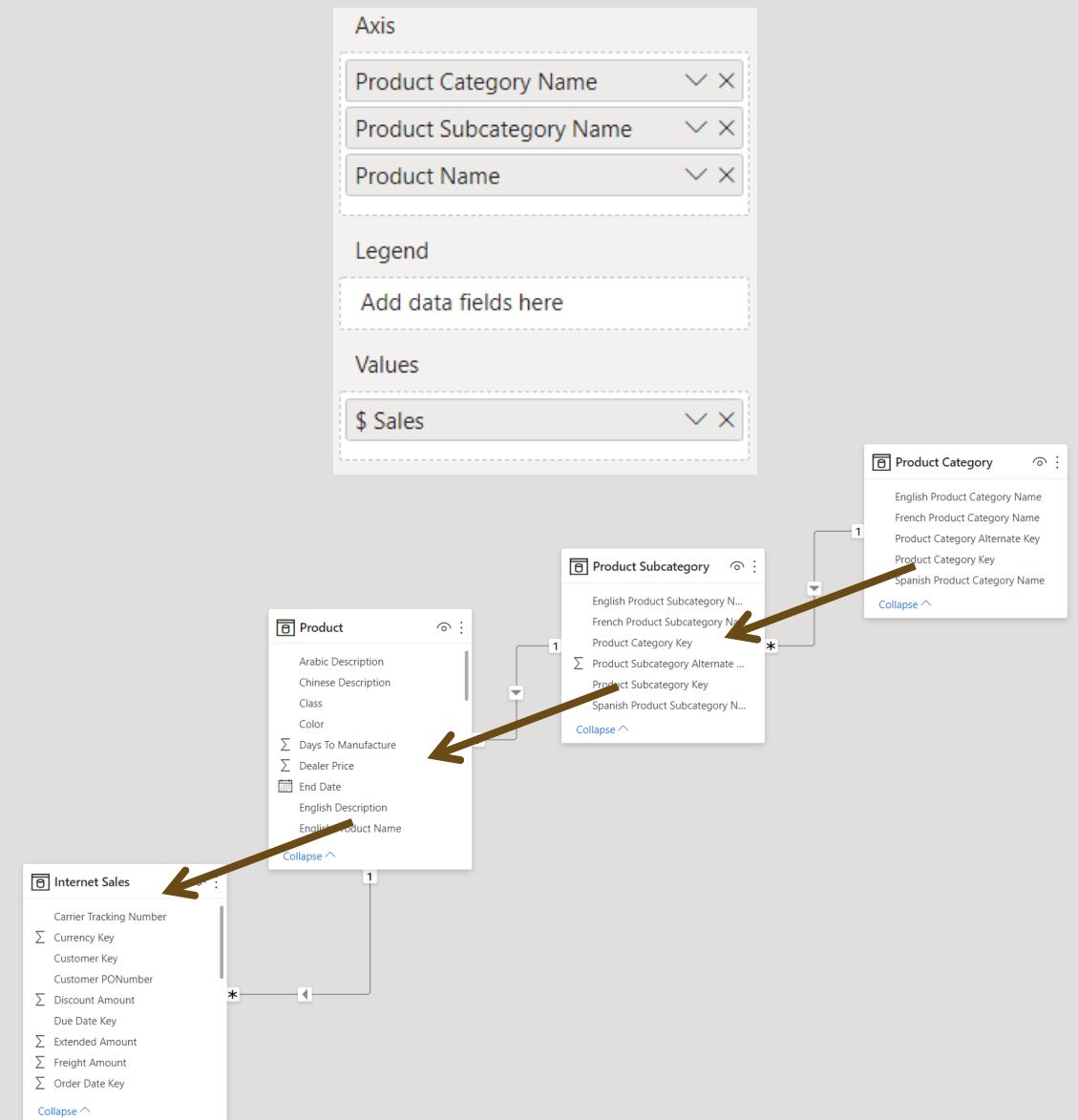
Typical examples:

- Year > Quarter > Month > Day
- Category > Subcategory > Product



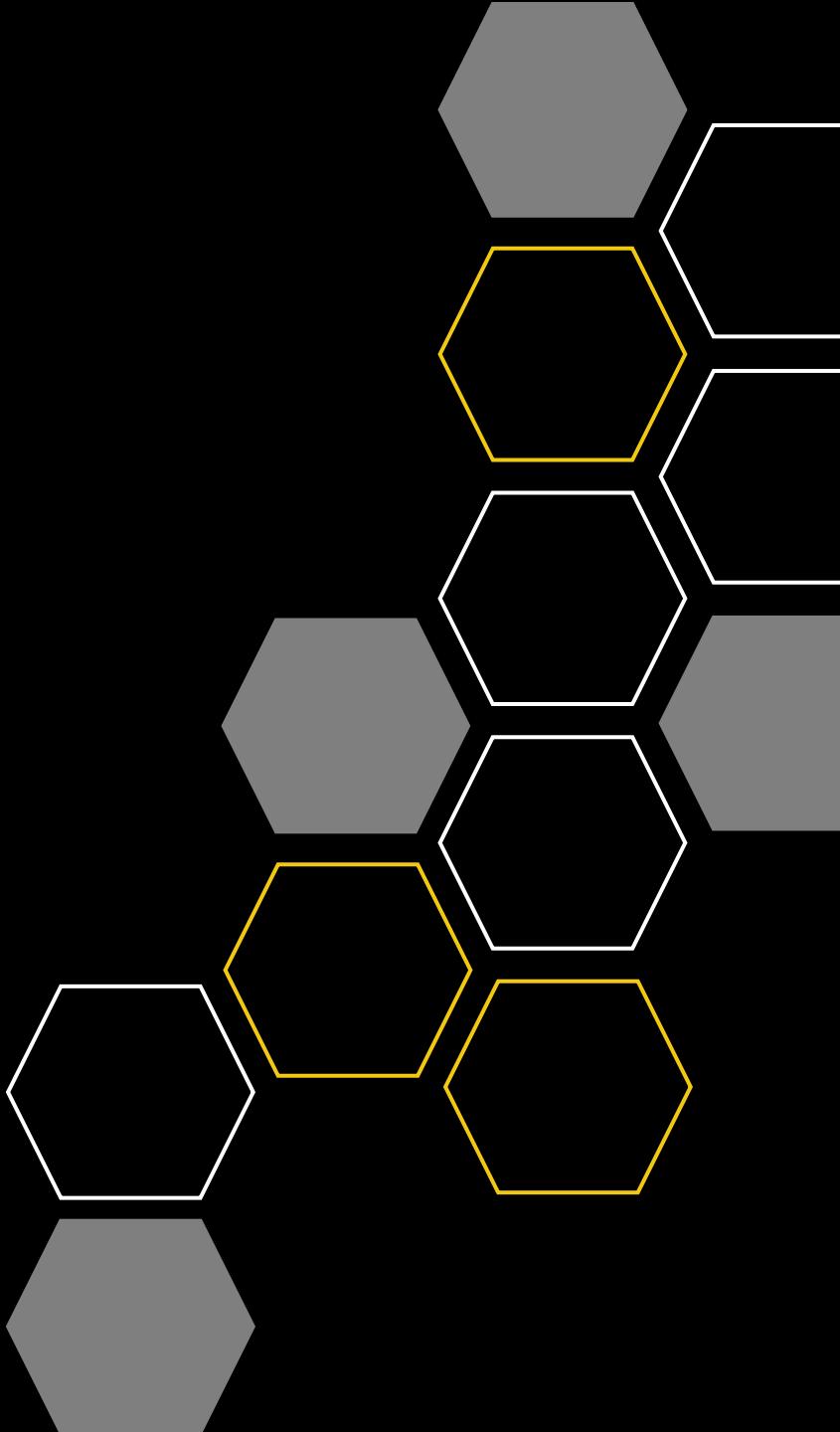
Hierarchies across multiple tables

- Hierarchies cannot be made across different tables
- Snowflake schema + multiple fields in the visualization



Demo: data groups and hierarchies

Marc Lelijveld



Resources

- [Guidance: auto date/time in Power BI Desktop](#)
- [Quick measures gallery](#)
- [DAX Function: USERELATIONSHIP](#)



Resources

- Learning path: Model data in Power BI:
<https://aka.ms/DataModelingLearningPath>
- Learning path: Use DAX in Power BI Desktop:
<https://aka.ms/DAXLearningPath>
- Aggregations in Power BI Desktop:
<https://aka.ms/PowerBIDesktopAggregations>
- DAX function: CROSSFILTER: <https://aka.ms/DAXCrossfilter>
- Model relationships in Power BI Desktop:
<https://aka.ms/ModelRelationships>
- Whitepaper: Bidirectional cross-filtering
for Power BI Desktop:
<https://aka.ms/BidirectionalCrossFilteringWhitepaper>
- AdventureWorks 2020 demo dataset: <https://aka.ms/AW2020Dataset>
- Easy bulk-edit of column names:
<https://aka.ms/PQSplitTextByCharacter>



Resources

Types of facts

aka.ms/TypesOfFacts

Grouping and binning

aka.ms/PBIDesktopGroupingAndBinning

Slowly changing dimensions

aka.ms/PBISlowlyChangingDimensions

Calculation groups

aka.ms/PowerBICalculationGroups

Different storage modes in Power BI

aka.ms/PowerBIStorageModes

Using Direct Query in Power BI

aka.ms/DirectQueryPBI

Guidance for Composite Models

aka.ms/CompositeModelsPBI

Relationship evaluation

aka.ms/DirectQueryRealationshipEvalPBI

DirectQuery for Power BI datasets and Azure Analysis Services (preview)

aka.ms/PBIDirectQueryDatasets

Model relationships in Power BI

aka.ms/ModelRelationshipsPBI

Want to review all this content?

Webinar series: Mastering data modeling with Power BI

Episode 1 - Data modeling 101: Increasing the impact of Power BI

by Jeroen ter Heerdt, Microsoft and Marc Lelijveld, Macaw Netherlands

[Watch now ↗](#)

Webinar series: Mastering data modeling with Power BI

Episode 2 - Learn advanced data modeling with Power BI

by Jeroen ter Heerdt, Microsoft and Marc Lelijveld, Macaw Netherlands

[Watch now ↗](#)

Webinar series: Mastering data modeling with Power BI

Episode 3 - Data modeling for experts with Power BI

by Jeroen ter Heerdt, Microsoft and Marc Lelijveld, Macaw Netherlands

[Watch now ↗](#)

Webinar series: Data modeling with Power BI

Episode 4 - Calculation groups and composite models

By Jeroen ter Heerdt, Microsoft and Marc Lelijveld, Macaw Netherlands

[Watch now ↗](#)

- **Find them all here:**

docs.microsoft.com/en-us/power-bi/fundamentals/webinars



Thank you