



Data Modeling 101:

Increasing the impact of Power BI

Marc Lelijveld & Jeroen ter Heerdt



Jeroen ter Heerdt

He/Him

Senior Program Manager
Microsoft

@JayPowerBI

 DutchDataDude.com



Expert on advanced analytics, big data, machine learning, data visualization and Business Intelligence; Microsoft as well as open source technologies. Speaker, blogger and data enthusiast

Marc Lelijveld

He/Him

Technical Evangelist

Macaw Netherlands



@MarcLelijveld

 Data-Marc.com



Data Platform MVP, FastTrack
Recognized Solution Architect,
Microsoft Certified Trainer, Power BI
enthusiastic, public speaker and
passionate for everything which
transforms data into action! Working at
Macaw in the Netherlands as Technical
Evangelist and Solution Architect.

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

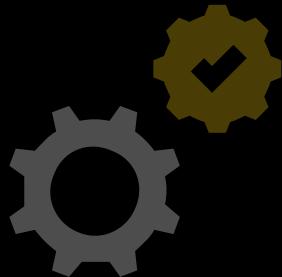


What are we talking about?

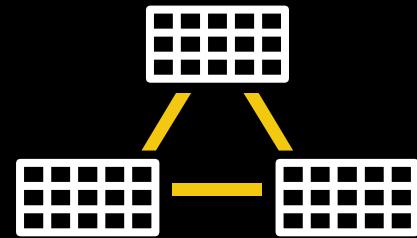
Gather



Clean



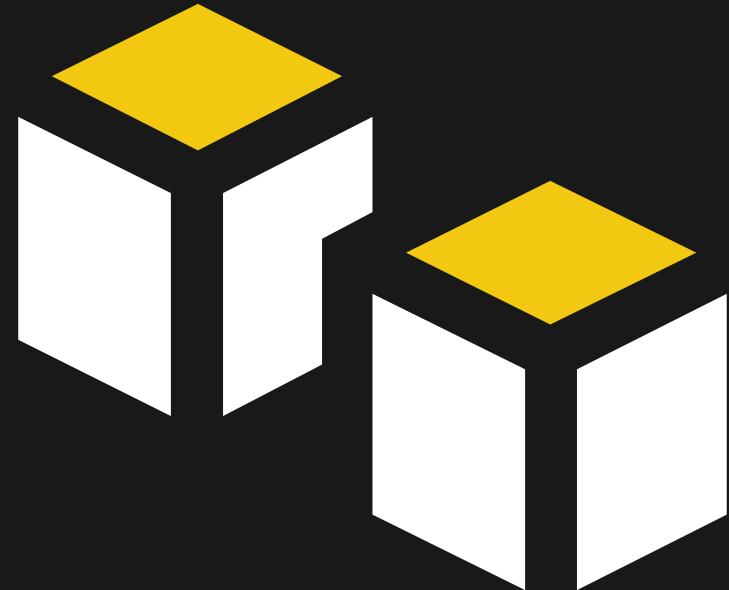
Model



Visualize



Data modeling basics



What is a data model?

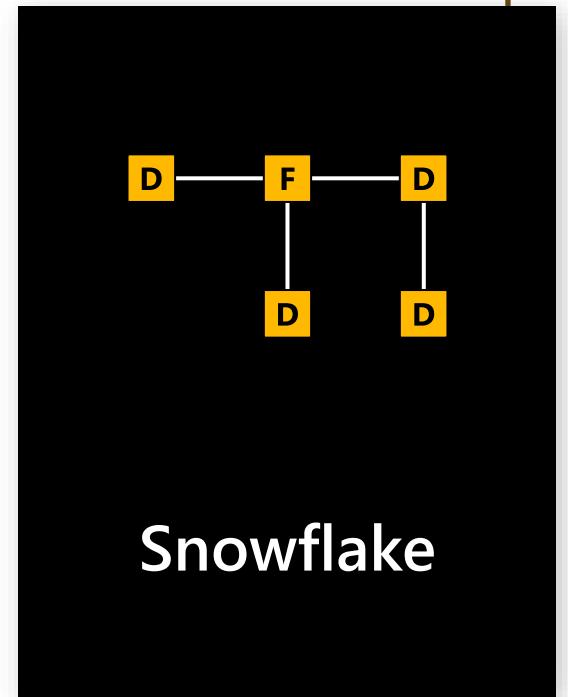
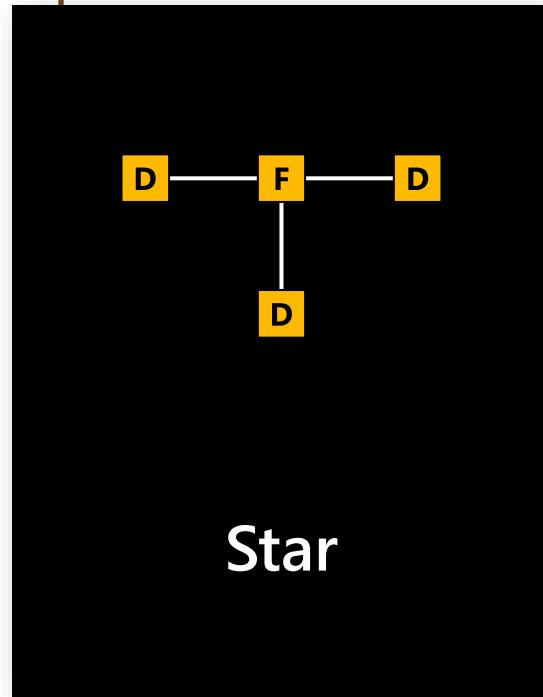
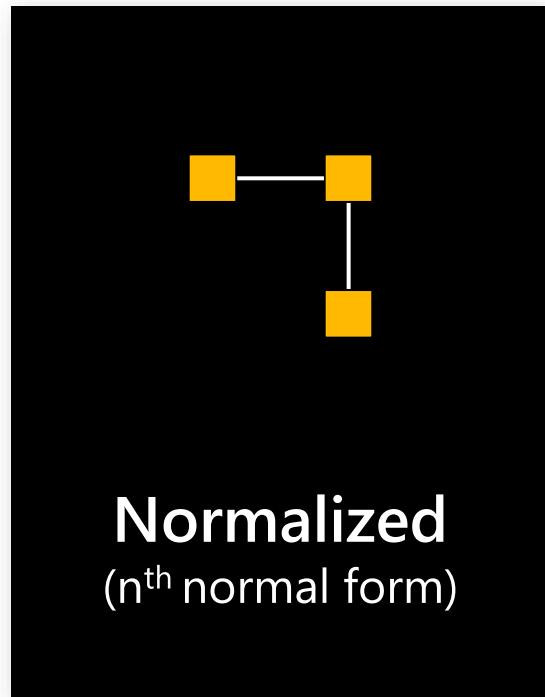
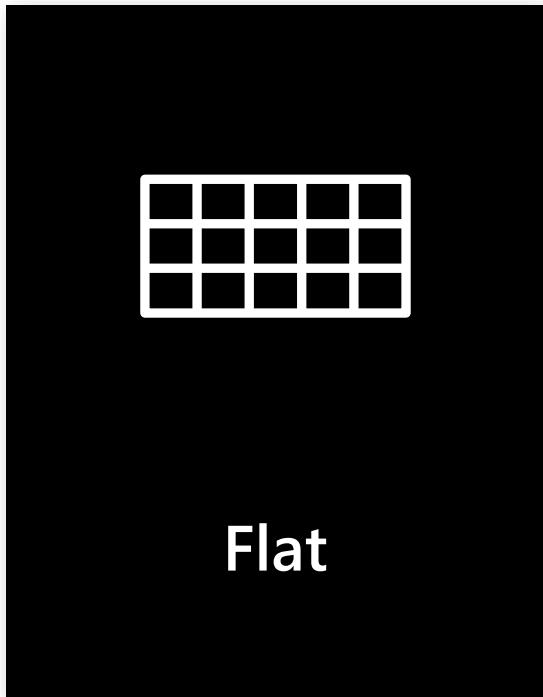
- A Power BI Data Model is a collection of tables with relationships which enable business users to easily understand and explore their data to get business insights.
- A good data model is important, because:
 - Improves understandability of the data
 - Increases performance of dependent processes and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



Major types of data models



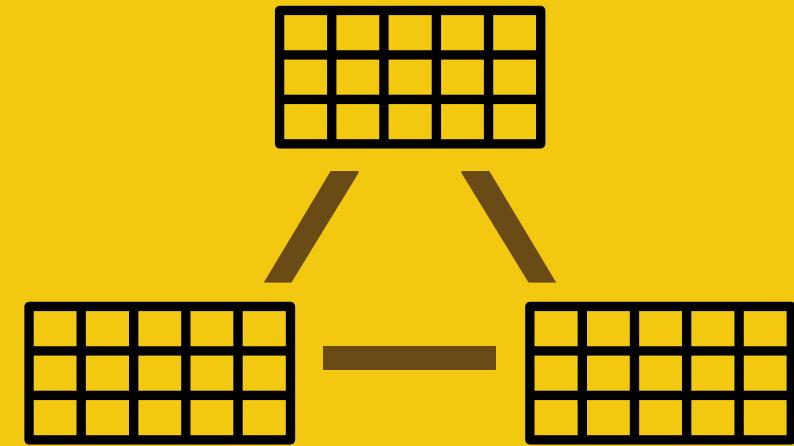
Dimensional

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

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

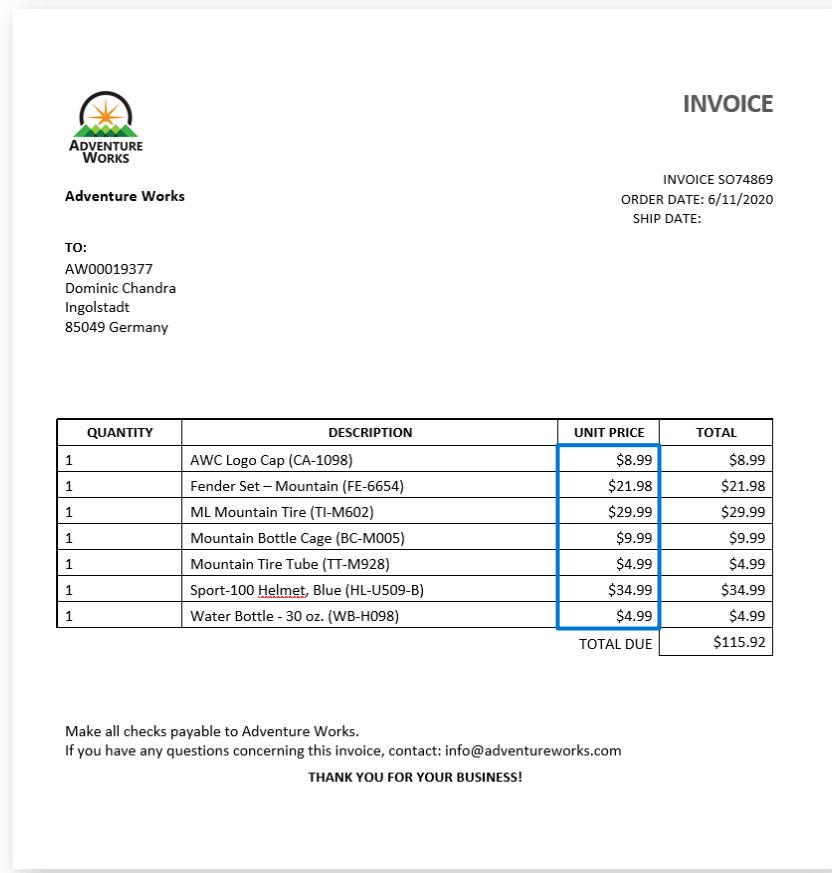
THANK YOU FOR YOUR BUSINESS!



Microsoft
Power BI

Building a dimensional data model

Creating Sales Amount fact

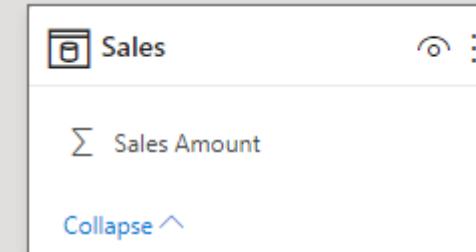


Tables

Sales

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

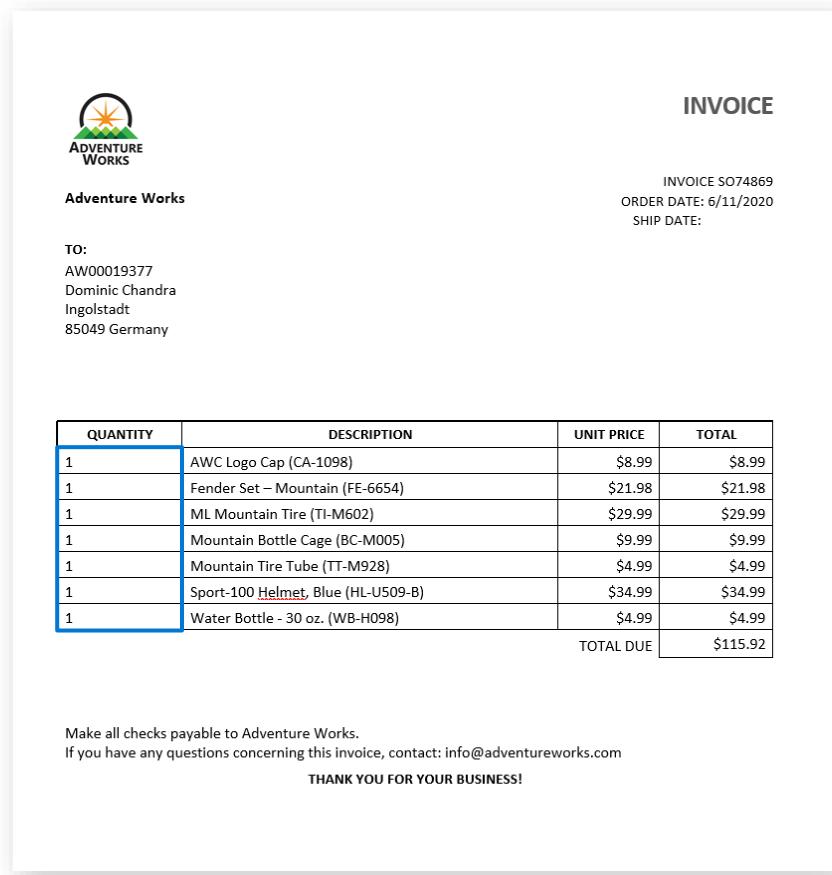
Diagram



Microsoft
Power BI

Building a dimensional data model

Adding Order Quantity fact

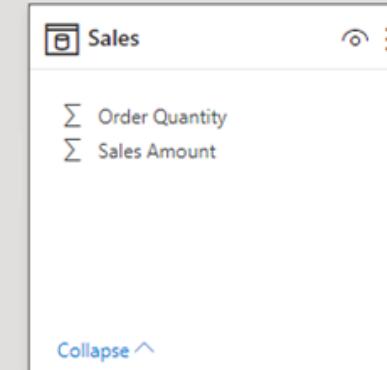


Tables

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



Microsoft
Power BI

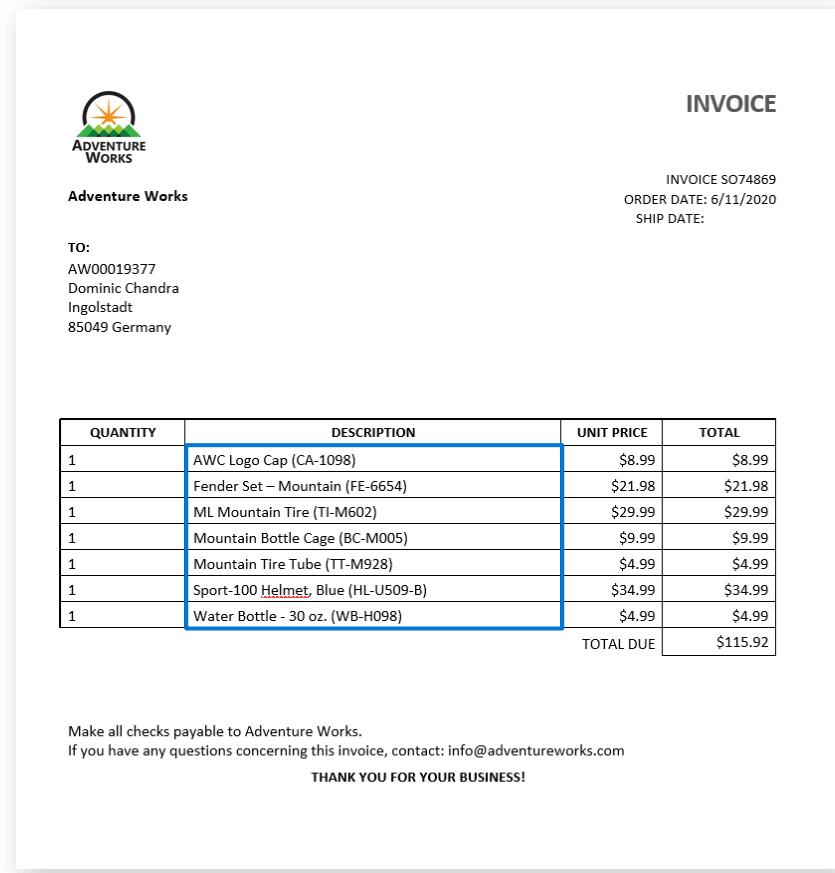
Fact table

- 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



Tables

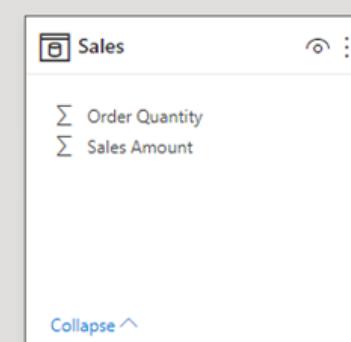
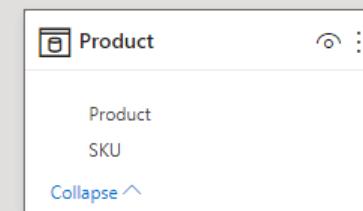
Product

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

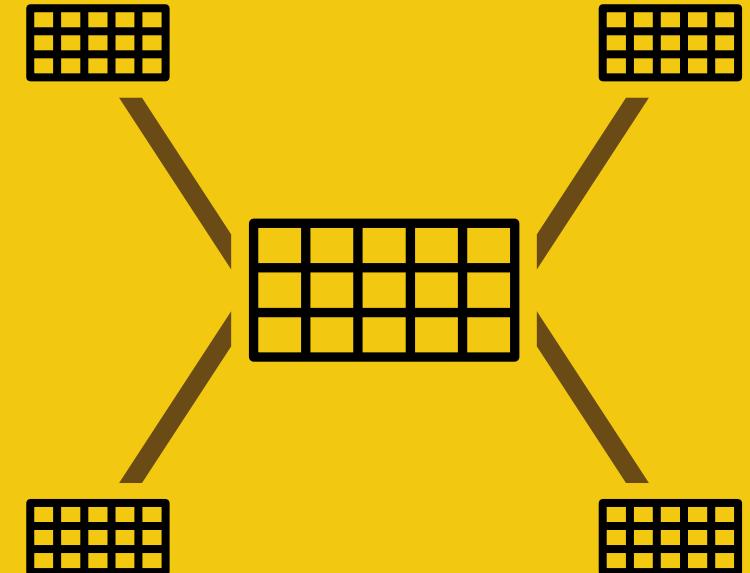
Sales Amount	Order Quantity
\$8.99	1
\$21.98	1
\$29.99	1
\$9.99	1
\$4.99	1
\$34.99	1
\$4.99	1

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

INVOICE

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

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 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

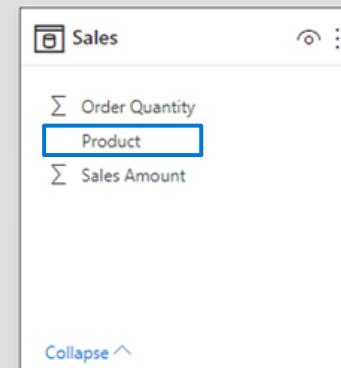
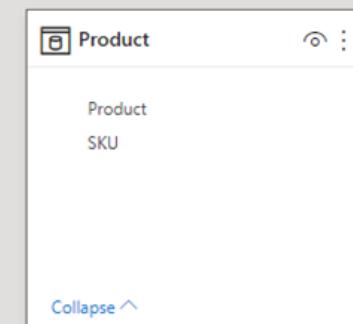
Product

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

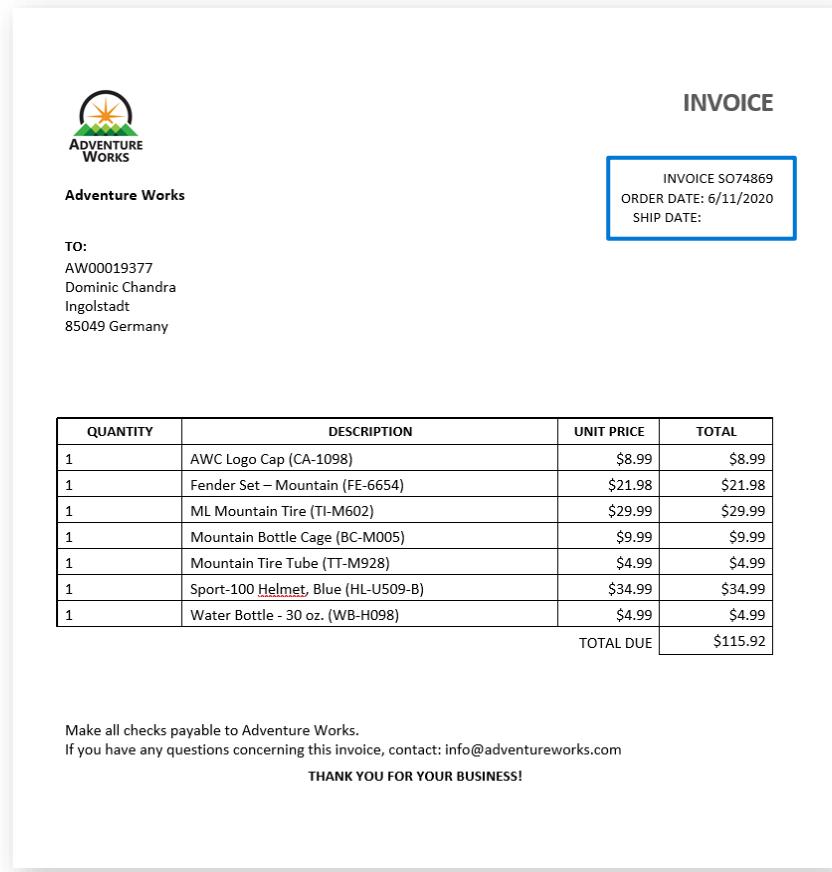
Sales Amount	Order Quantity	Product
\$8.99	1	CA-1098
\$21.98	1	FE-6654
\$29.99	1	TI-M602
\$9.99	1	BC-M005
\$4.99	1	TT-M928
\$34.99	1	HL-U509-B
\$4.99	1	WB-H098

Diagram



Building a dimensional data model

Adding Sales Order and Order Date attribute to fact



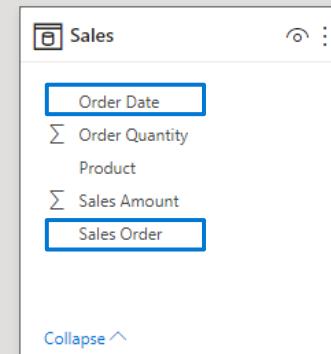
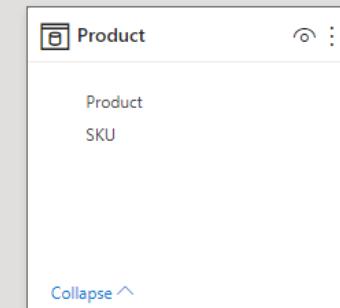
Tables

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

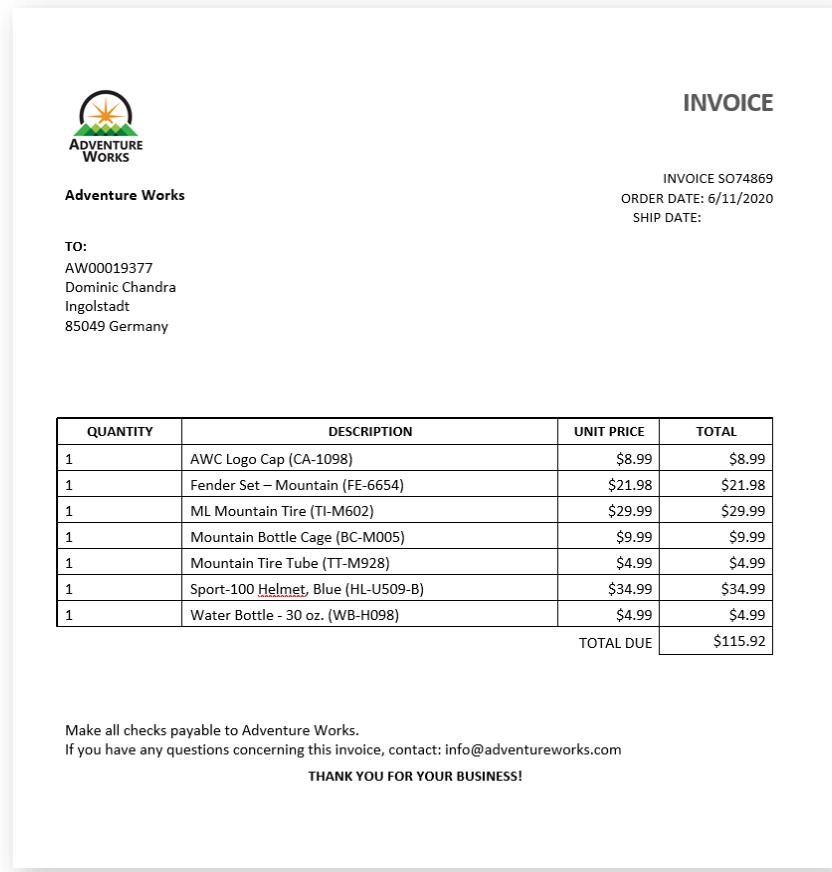
Product

Diagram



Building a dimensional data model

Adding relationship between Product and Sales



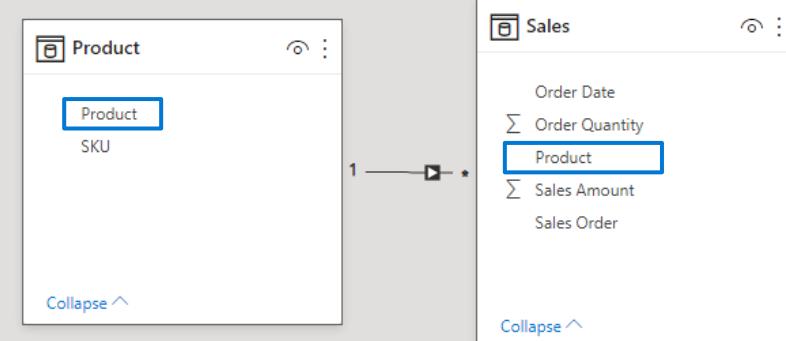
Tables

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

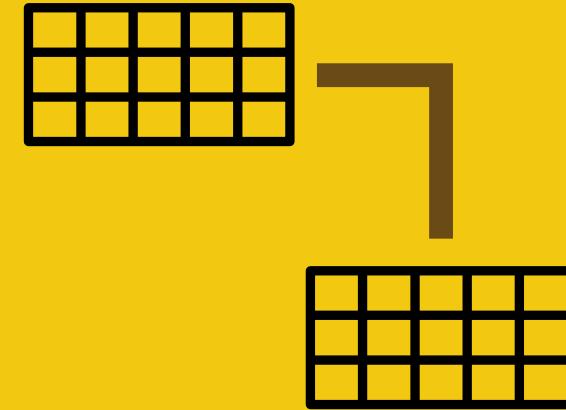
Product

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



Relationship direction

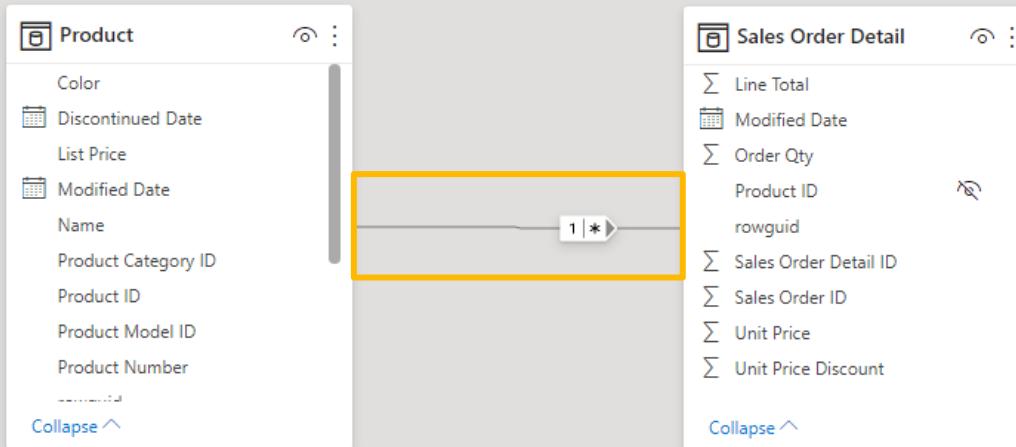
We explained **three types of relationships**, but skipped over the relationship direction



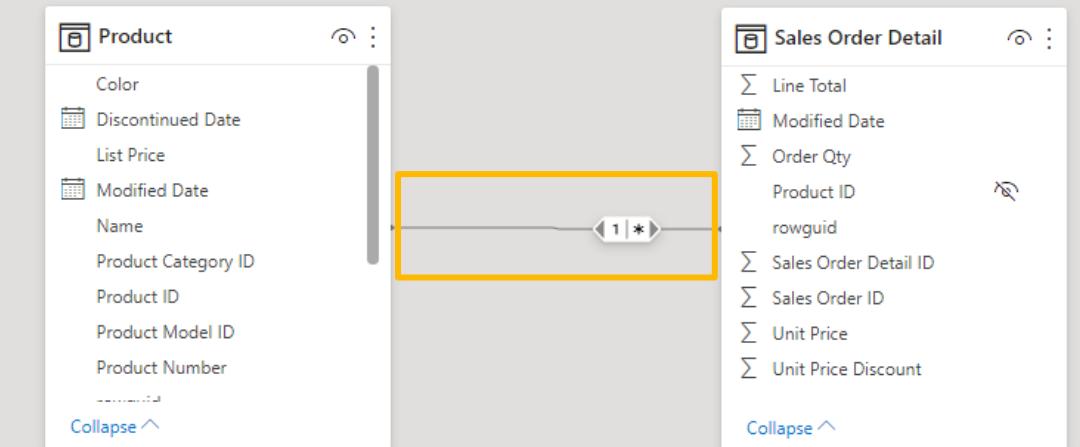
Relationship direction

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)

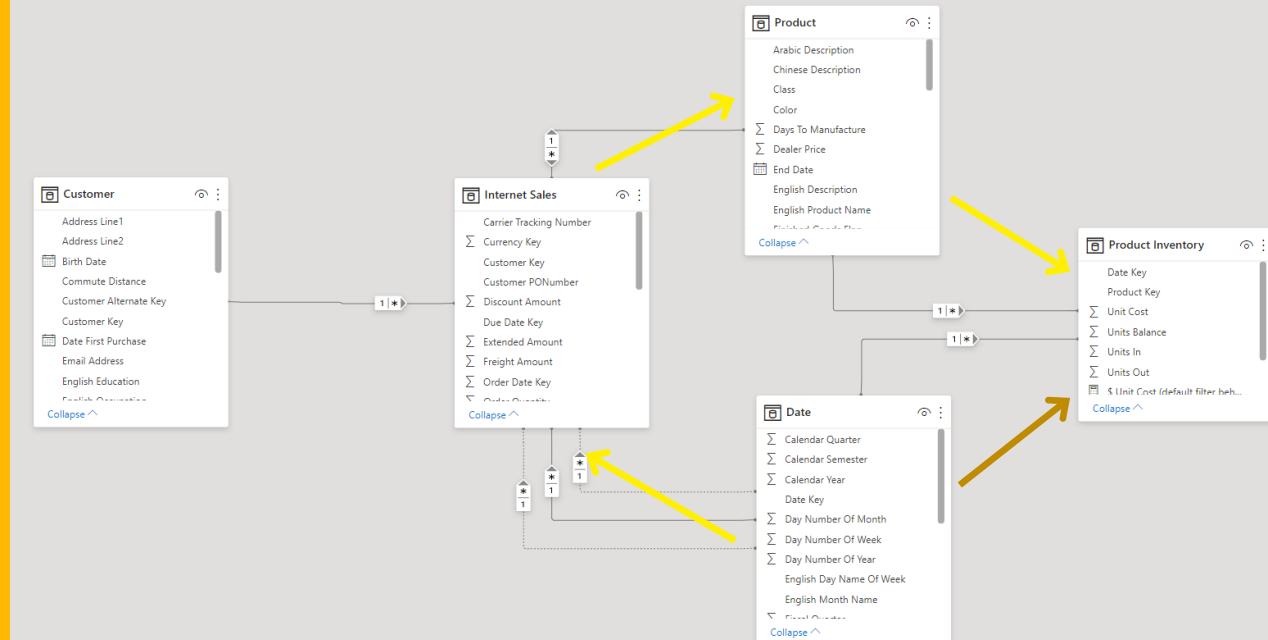
Ambiguous data models

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

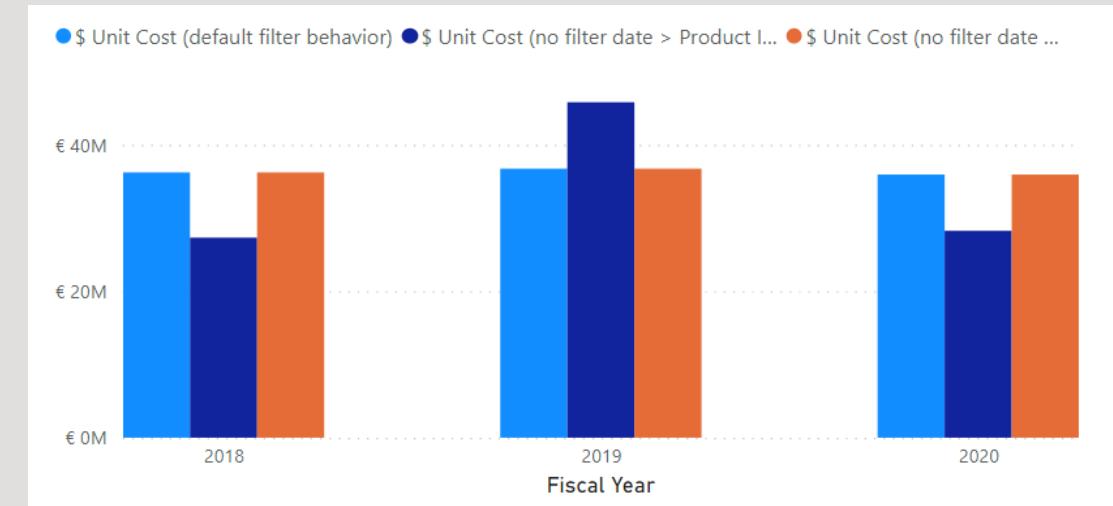


Ambiguous data models 1/2

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

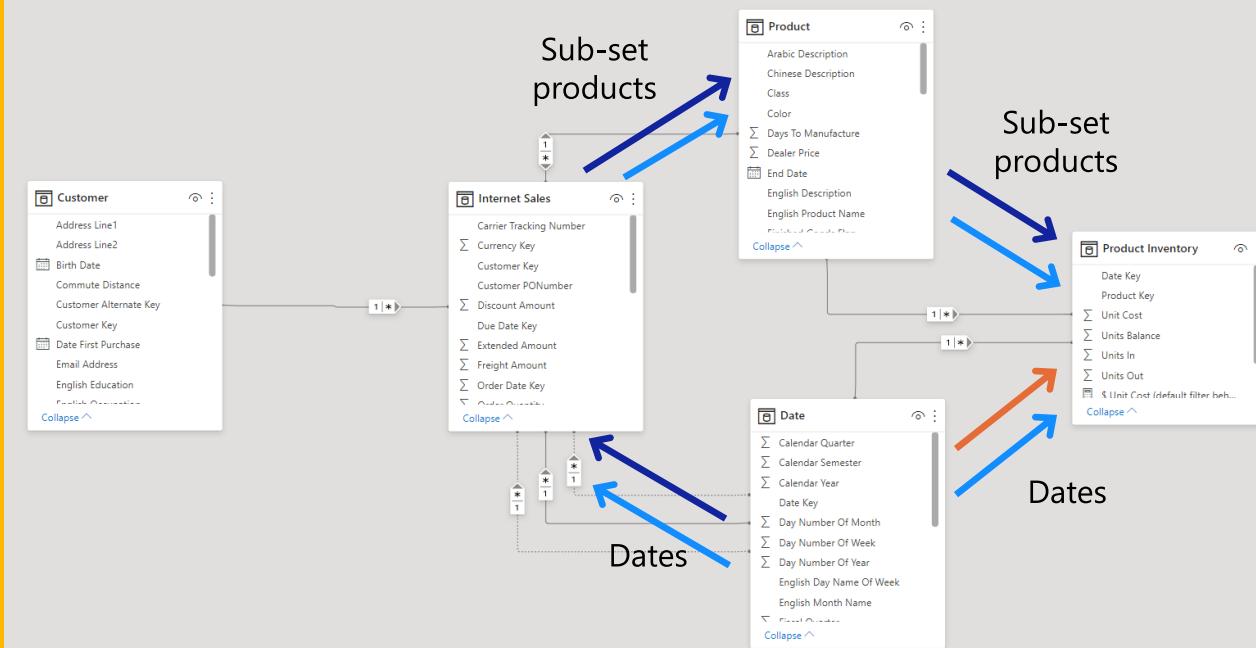
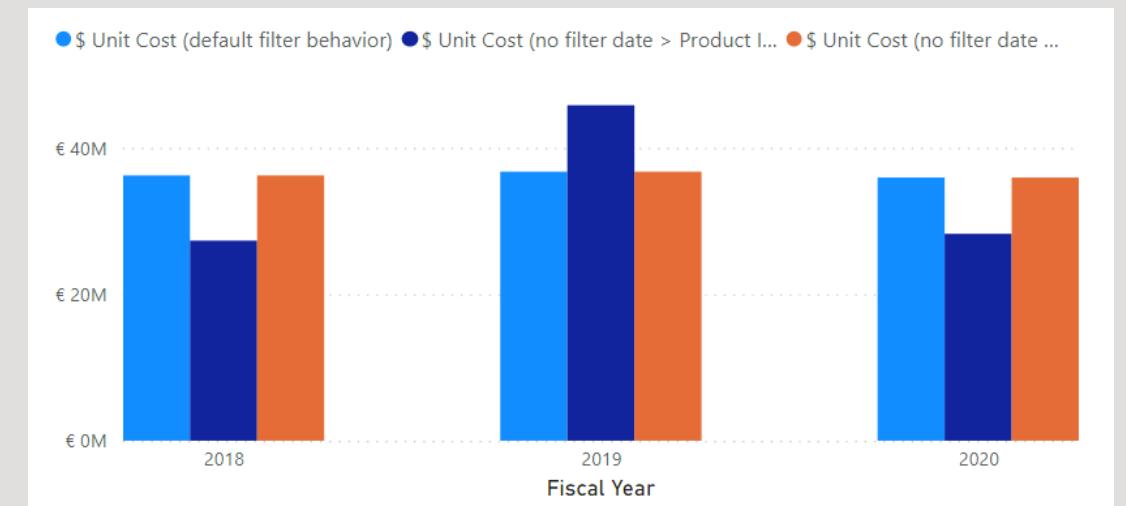


Fiscal Year	\$ Unit Cost (default filter behavior)	\$ Unit Cost (no filter date > Product Inventory)	\$ Unit Cost (no filter date > Internet Sales)
2018	€ 36.244.707,62	€ 27.337.603,34	€ 36.244.707,62
2019	€ 36.758.094,11	€ 45.834.231,75	€ 36.758.094,11
2020	€ 35.954.533,19	€ 28.267.611,68	€ 35.954.533,19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

Ambiguous data models 2/2

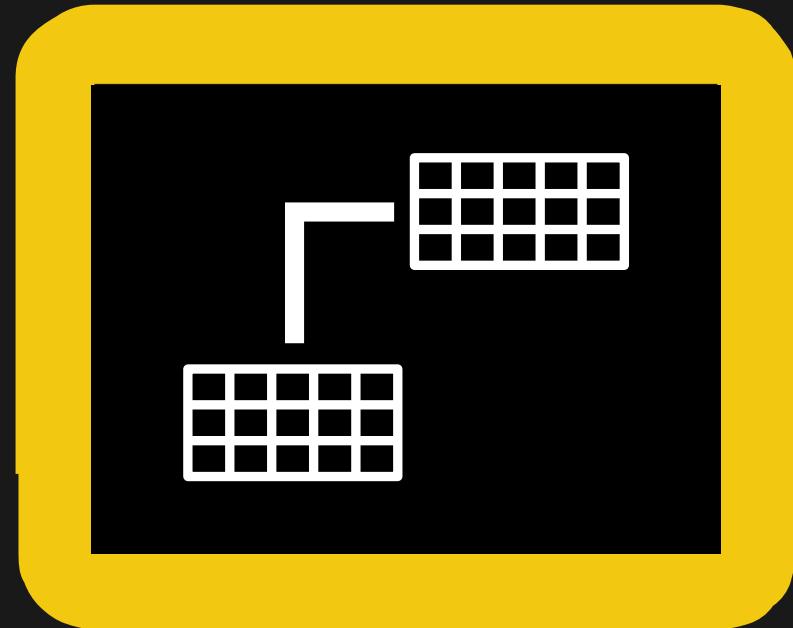
Unexpected filter behavior

- 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)



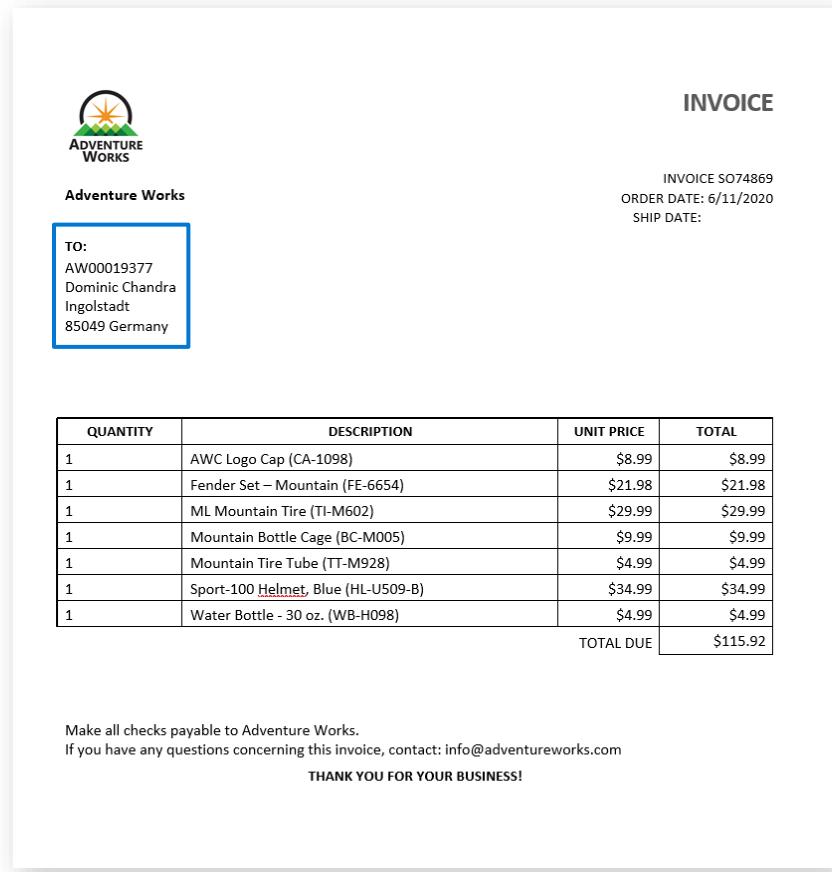
Relationship direction

Demo



Building a dimensional data model

Adding Customer dimension



Tables

Customer

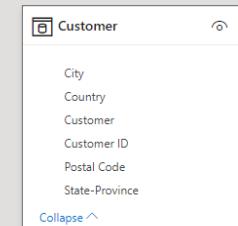
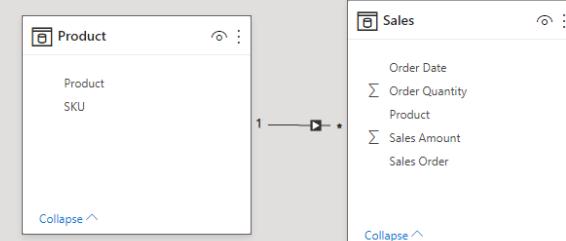
Customer ID	Customer	City	Postal Code	Country	State-Province
AW00019377	Dominic Chandra	Ingolstadt	85049	Germany	Bayern

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

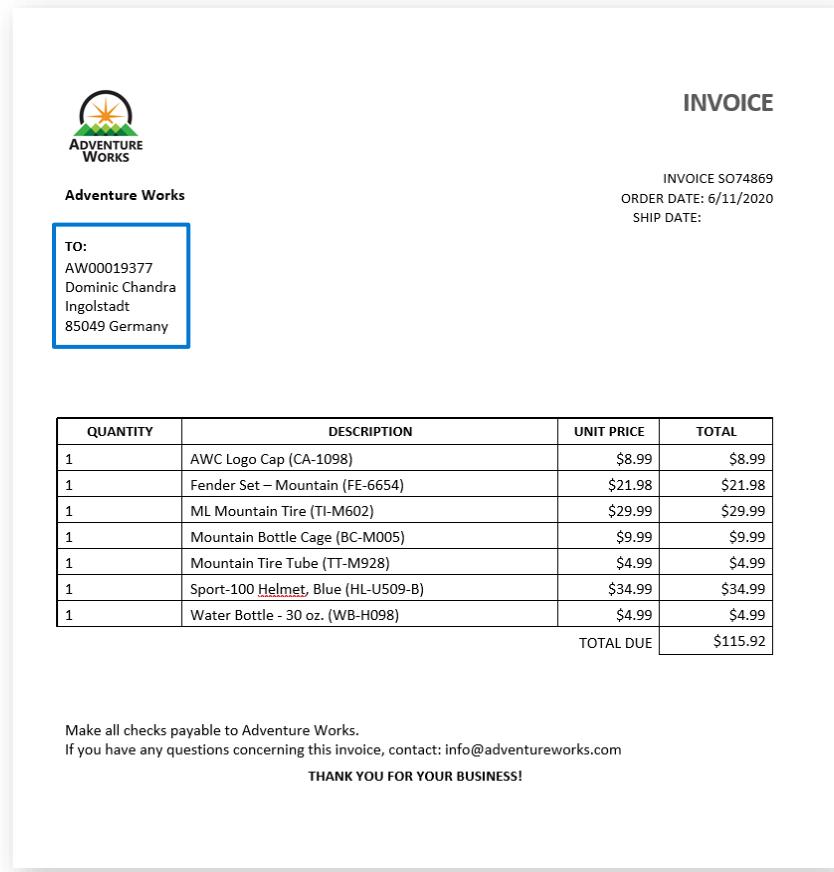
Diagram



Microsoft
Power BI

Building a dimensional data model

Adding Customer attribute on fact and relationship



Tables

Customer

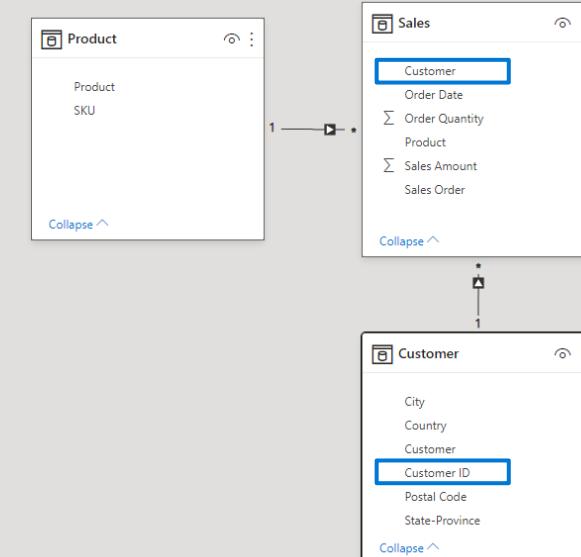
Customer ID	Customer	City	Postal Code	Country	State-Province
AW00019377	Dominic Chandra	Ingolstadt	85049	Germany	Bayern

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

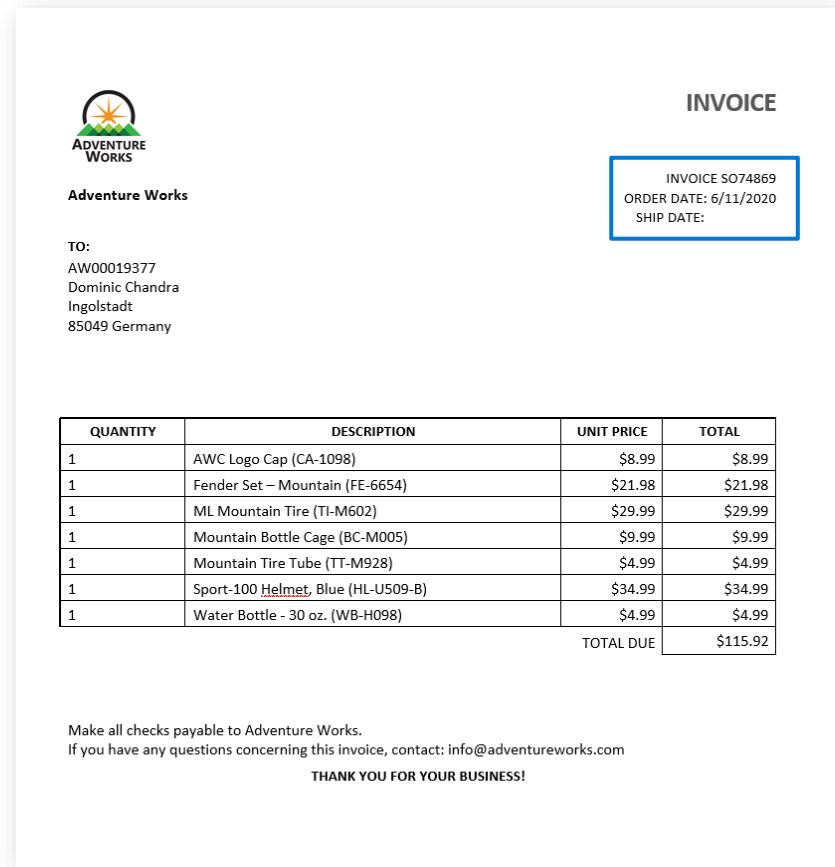
Diagram



Microsoft
Power BI

Building a dimensional data model

Adding Date dimension and relationship



Tables

Date

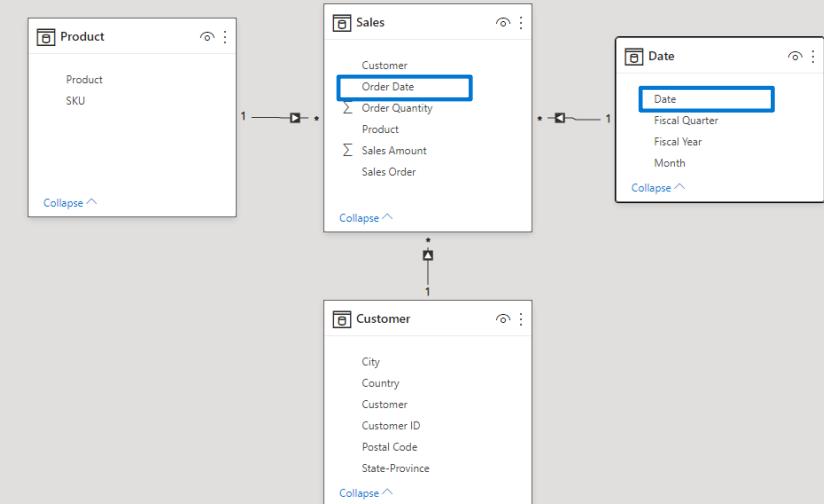
Date	Fiscal Quarter	Fiscal Year	Month
6/11/2020	FY2020 Q4	FY2020	2020 Jun
7/11/2020	FY2021 Q1	FY2021	2021 July

Product

Customer

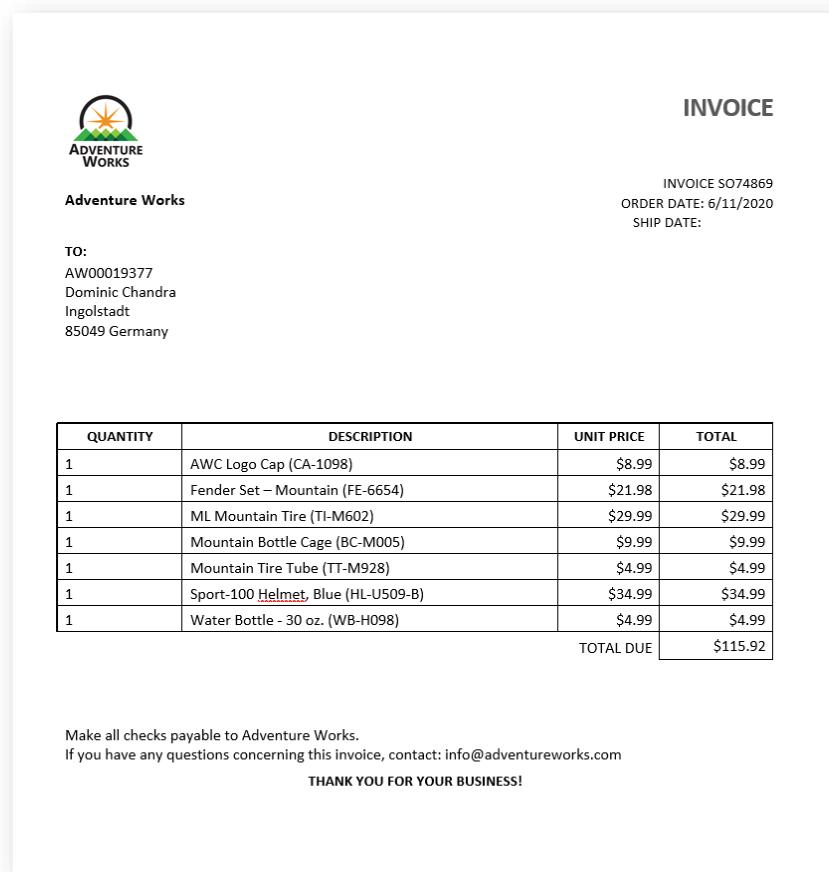
Sales

Diagram



Building a dimensional data model

Adding Ship Date attribute and relationship



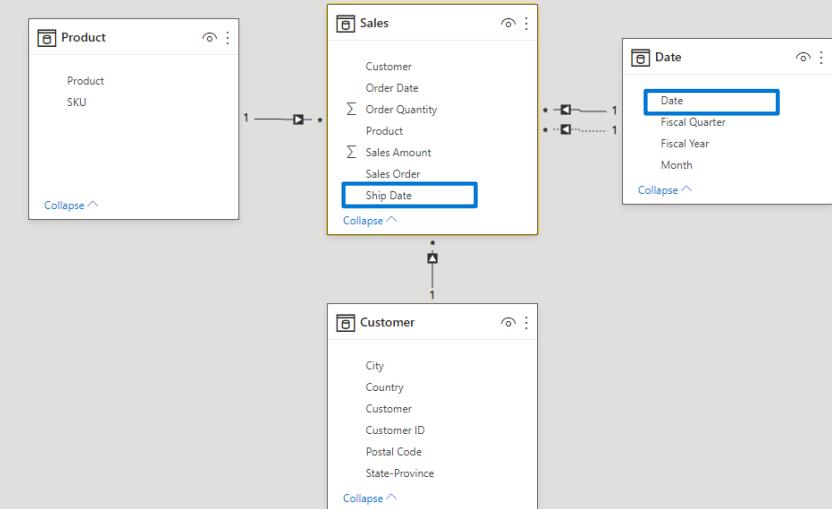
Tables

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	

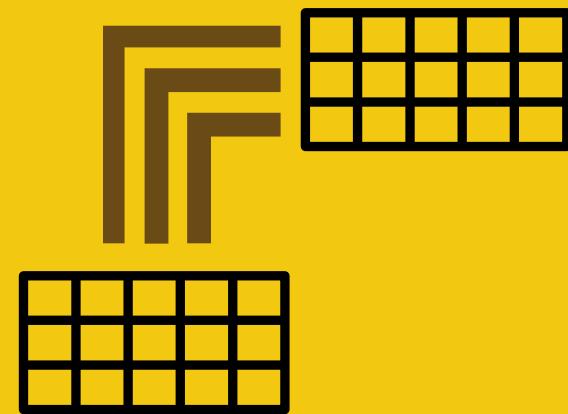
Product Customer Sales

Diagram



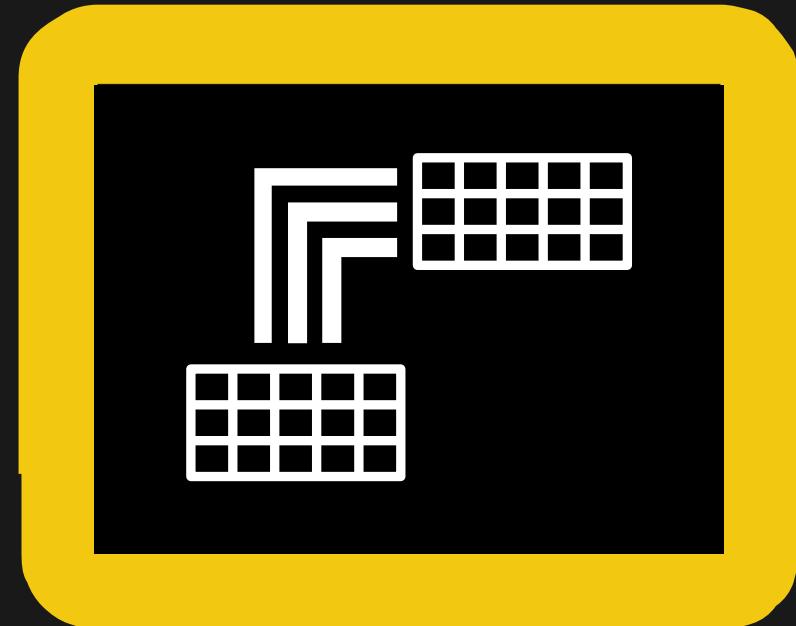
Role playing dimensions

- 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



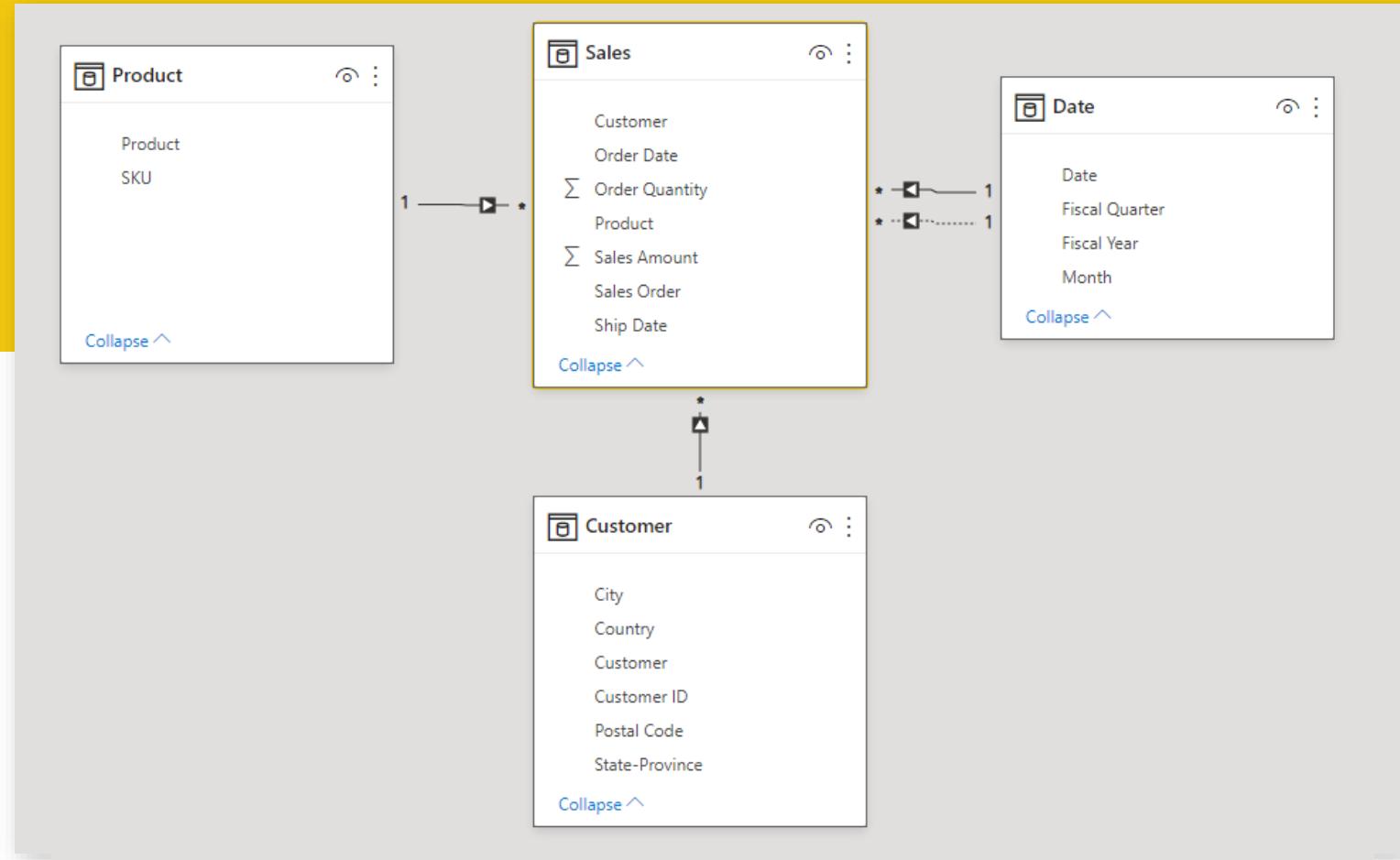
Role-playing dimensions

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.



Considerations for data modeling in Power BI

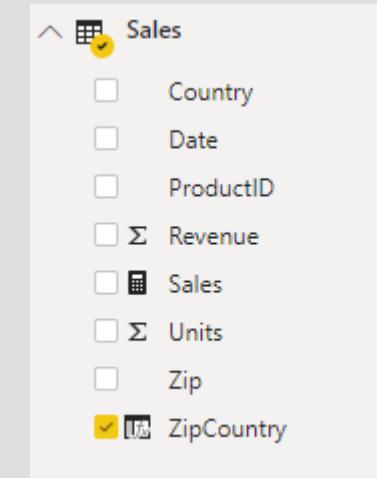
- Do data cleansing activity in one place.
- Enrich your data with measures and calculated columns.



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")))
```

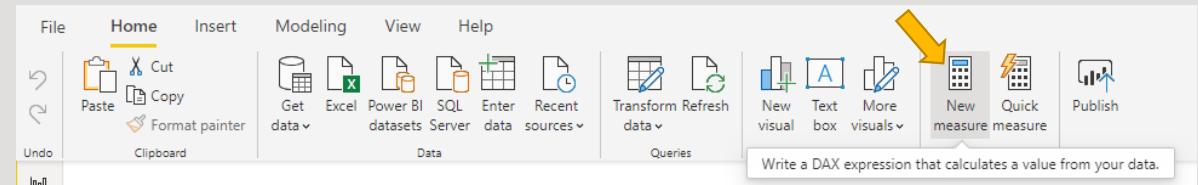
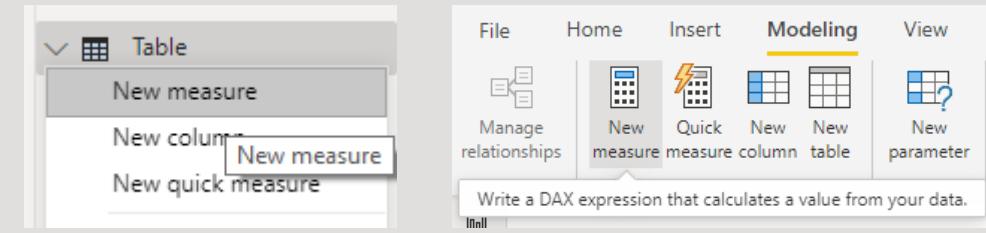


Measure

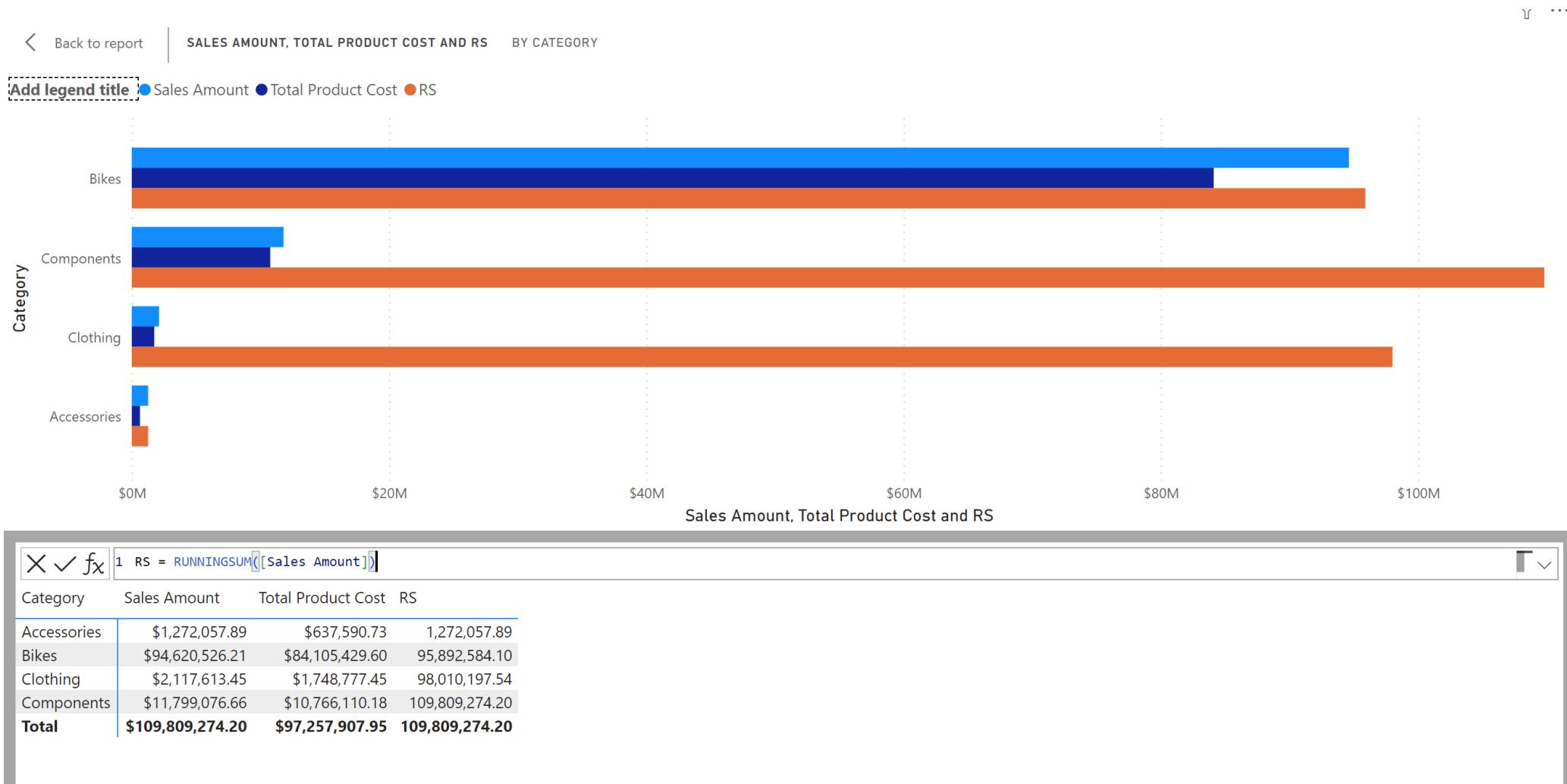
→ To create a measure using a DAX formula, select the New measure button from the Modeling tab.

→ 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



Visual Calculations



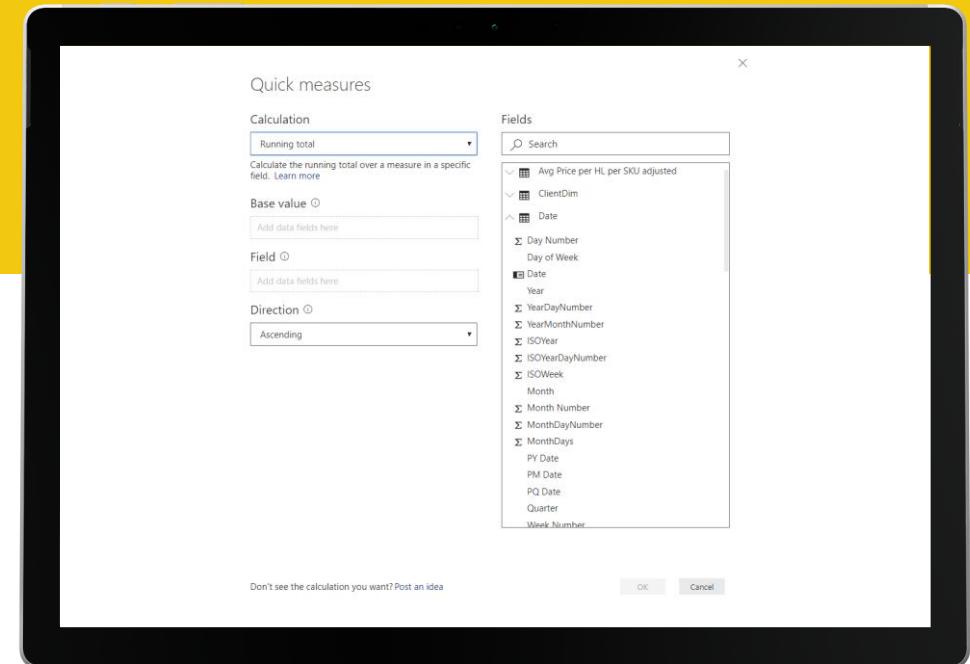
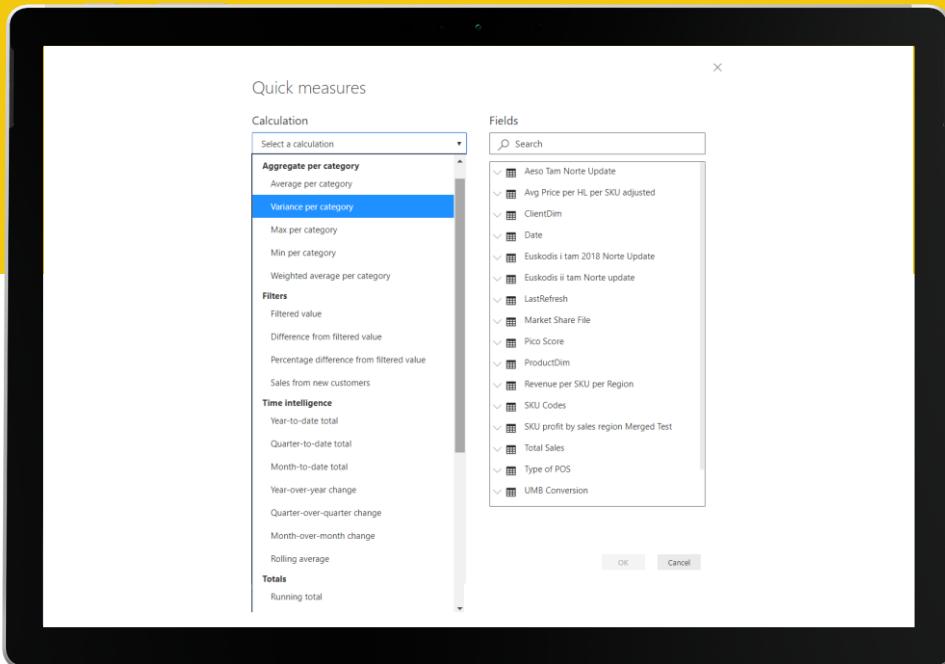
Calculated Column vs Measure vs Visual Calculation

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



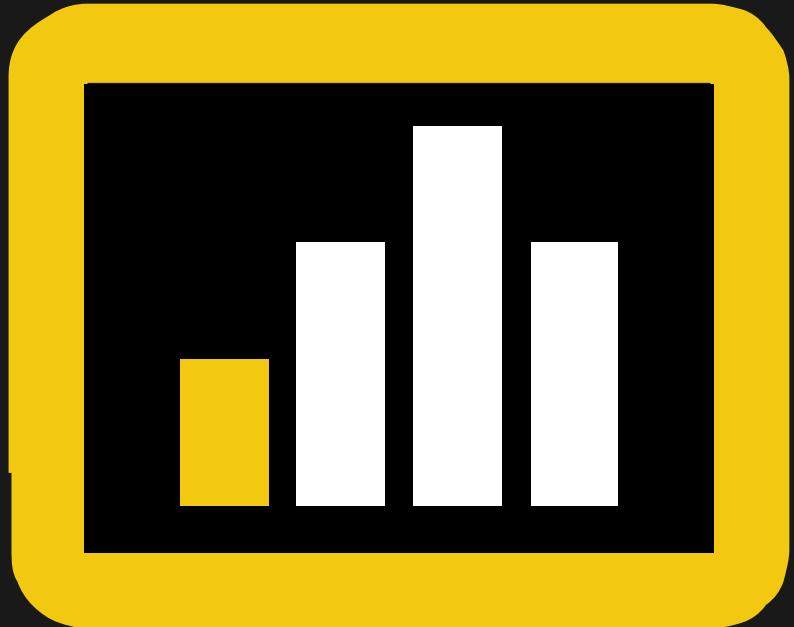
Quick Measures

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



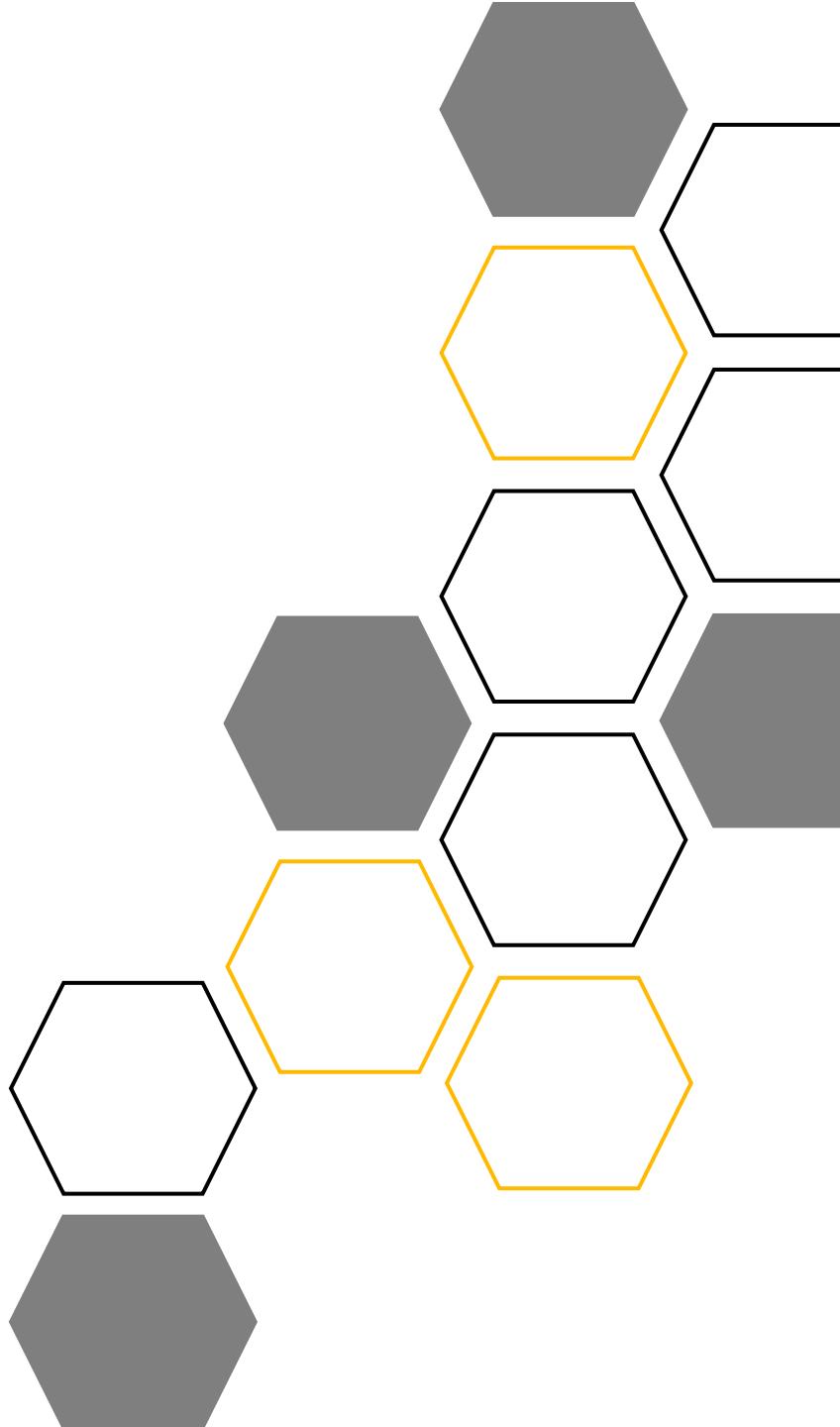
Measures and Calculated Columns

Demo



Microsoft
Power BI

Data modeling best practices



Naming of tables

Many sources use technical names

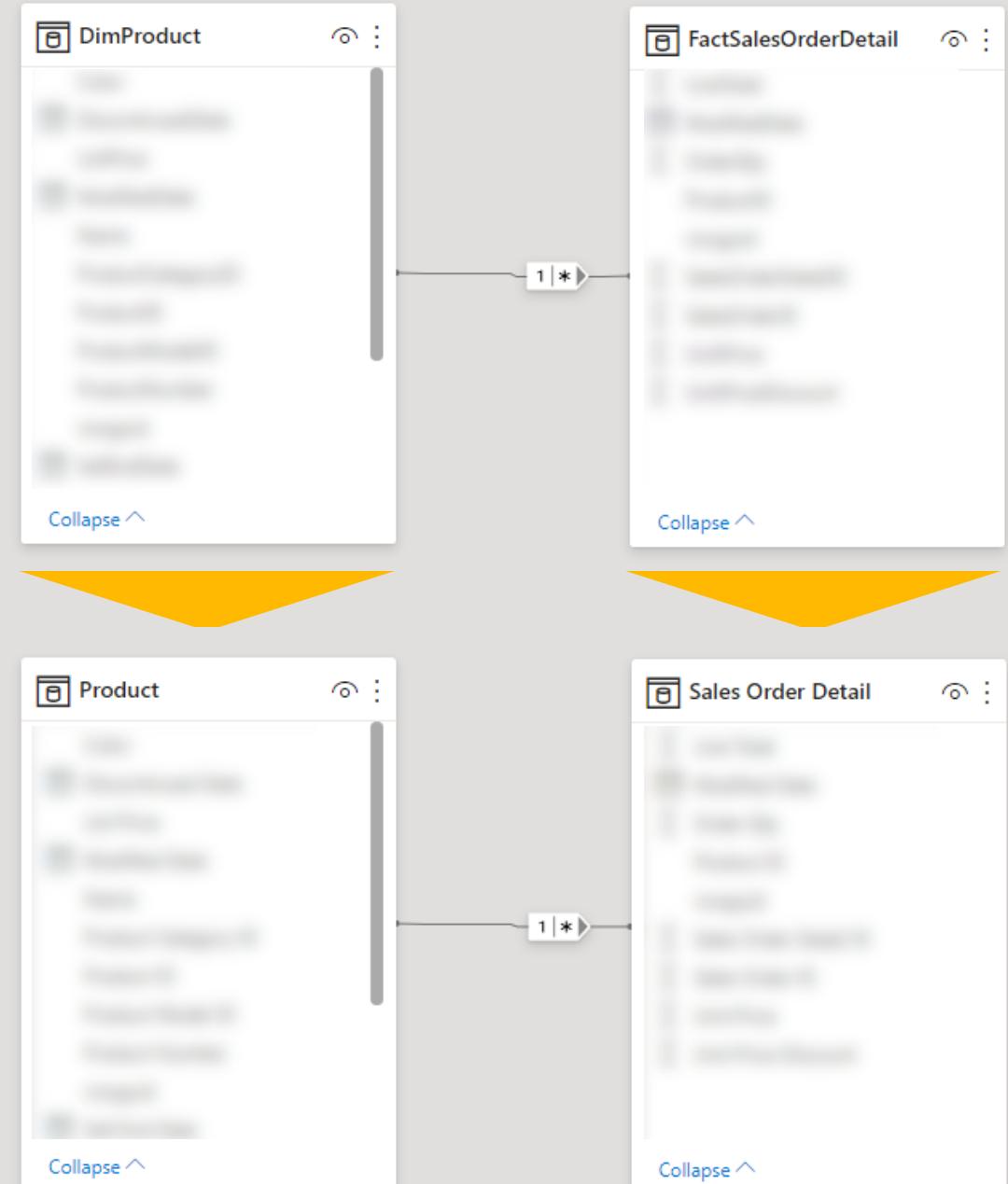
- FactSalesOrderDetail
- CustomerID

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							

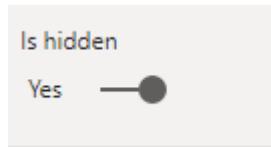


	1 ² 3 Sales Order ID	1 ² 3 Sales Order Detail ID	1.2 Order Qty	1 ² 3 Product ID	\$ Unit Price	\$ Unit Price Discount	1.2 Line Total
1							
2							
3							
4							
5							
6							
7							
8							
9							



Hide what doesn't matter

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



Sales

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

[Collapse ^](#)

A screenshot of the Microsoft Power BI Model View for the 'Sales' table. The view lists various columns with their data types and aggregation functions. A vertical yellow box highlights the first 12 columns: CustomerKey, Σ DueDateKey, Σ Extended Amount, Σ Order Quantity, OrderDateKey, Σ Product Standard Cost, ProductKey, ResellerKey, Σ Sales Amount, SalesOrderLineKey, SalesTerritoryKey, and Σ ShipDateKey. Each highlighted column has a '☒' icon in its last column. At the bottom of the list is a 'Collapse ^' button.

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 'Summarize' icon and a 'Settings' icon. The columns are listed below:

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

The 'Σ Sales Amount' and 'Total Sales' columns are highlighted with a yellow border.

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



Resources

- [Learning path: Model data in Power BI](#)
[learning path](#)
- [Learning path: Use DAX in Power BI Desktop](#)
- [Guidance: auto date/time in Power BI Desktop](#)
- [Quick measures gallery](#)
- [DAX Function: USERELATIONSHIP](#)
- [AdventureWorks 2020 demo dataset](#)



Session evaluation

Your feedback is important to us

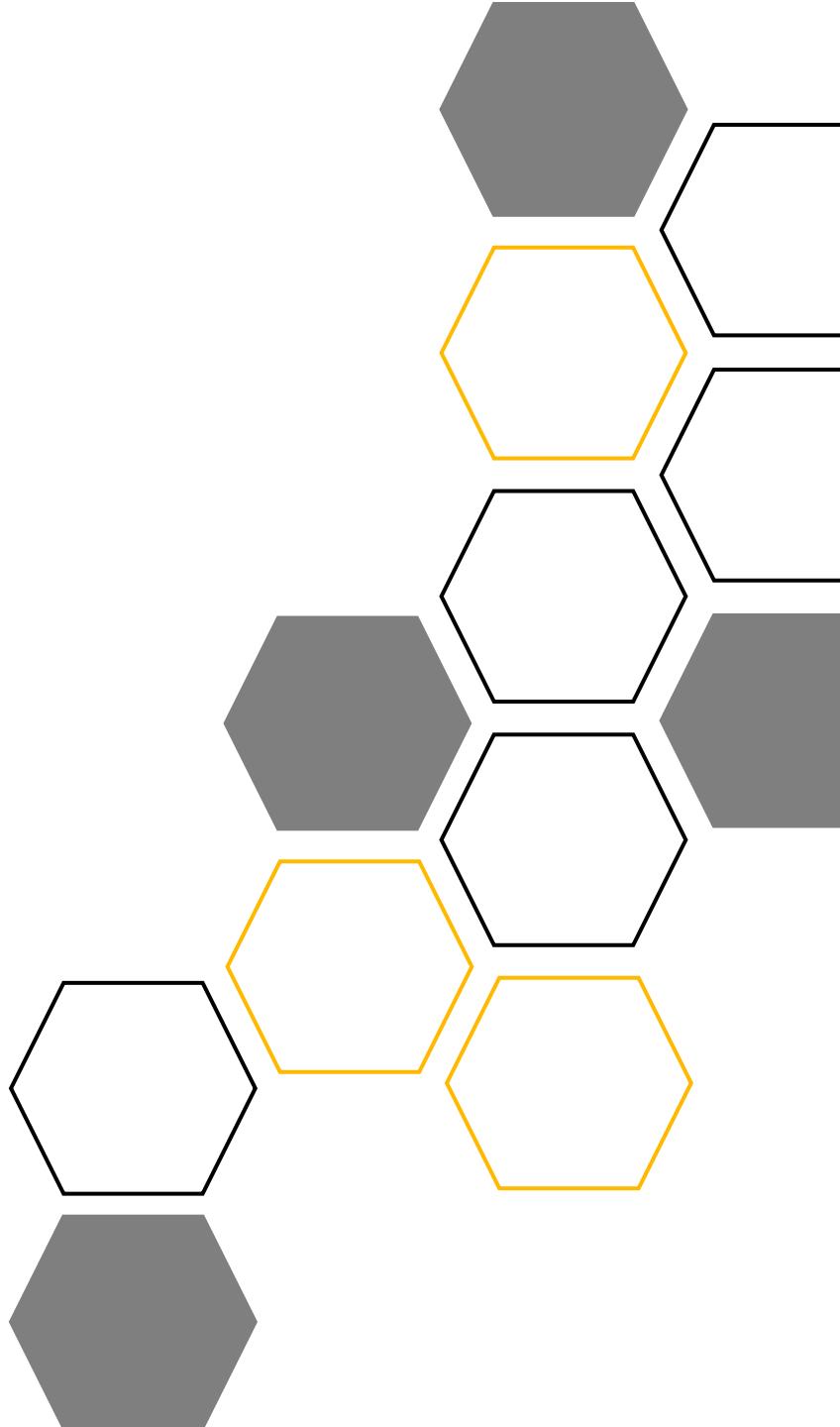


Evaluate this session at:

www.PASSDataCommunitySummit.com/evaluation



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.

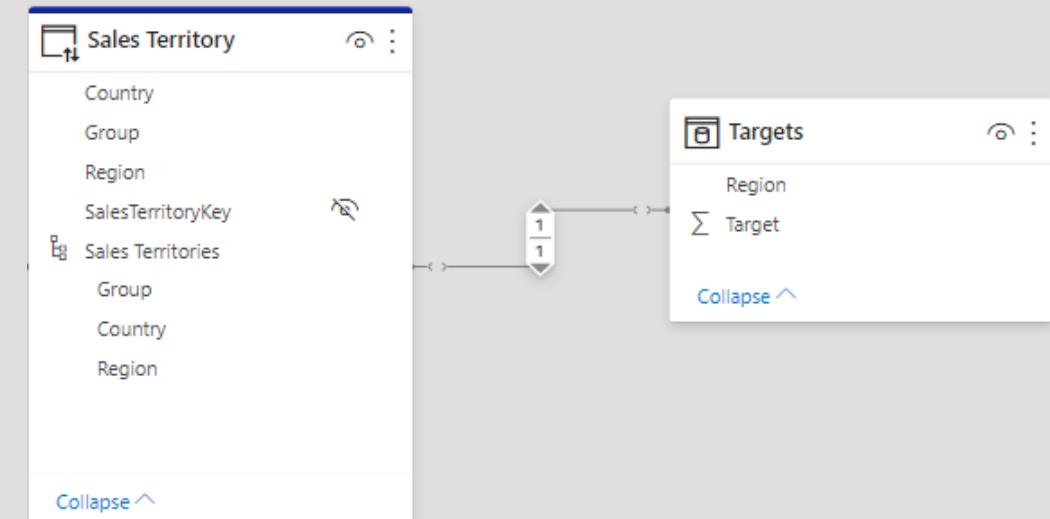
Regular relationships include the following storage mode combinations, where both tables are from a single source group:

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

Relationships: Regular vs limited

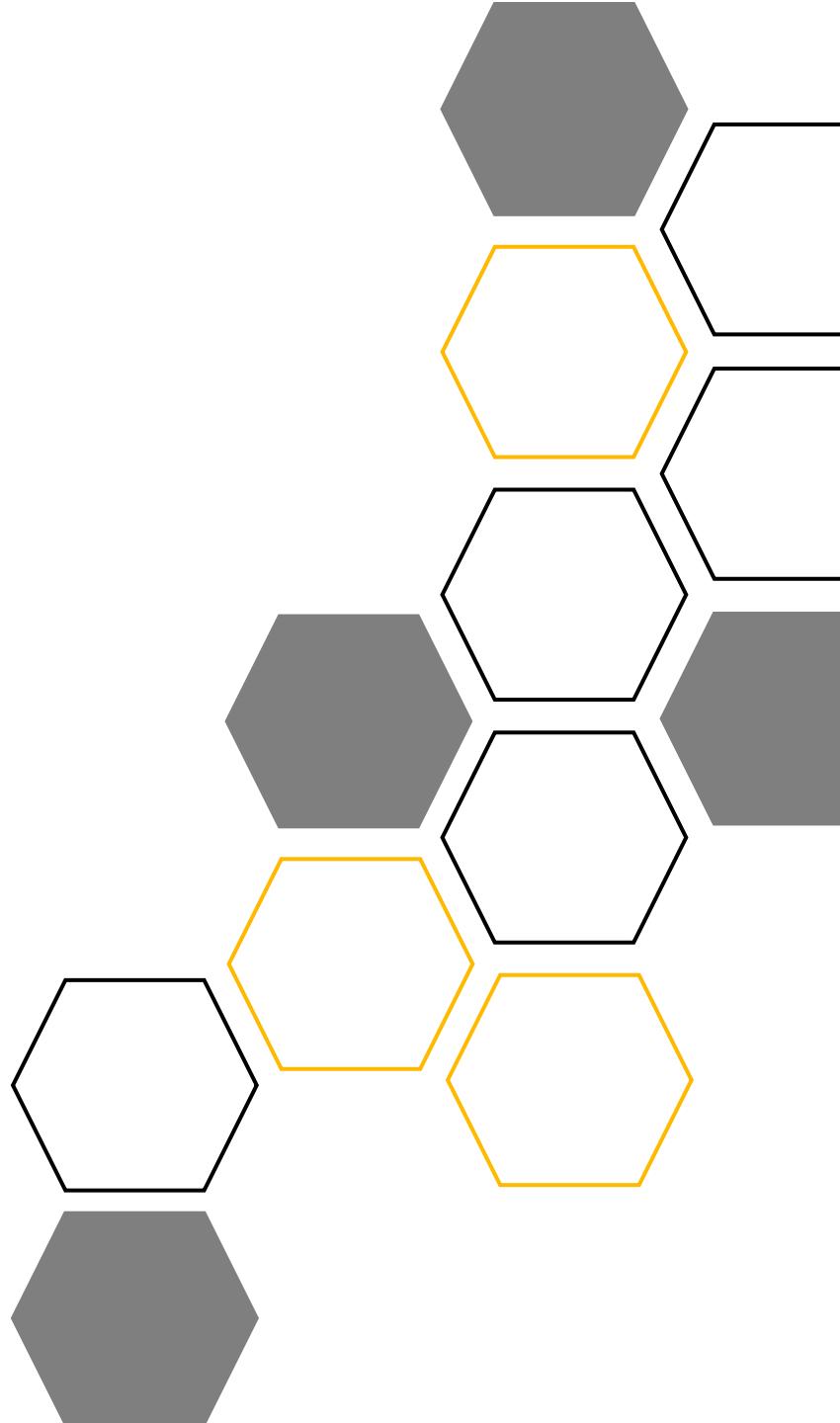
A model relationship is *limited* when there's no guaranteed "one" side. It can be the case for three reasons:

- 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

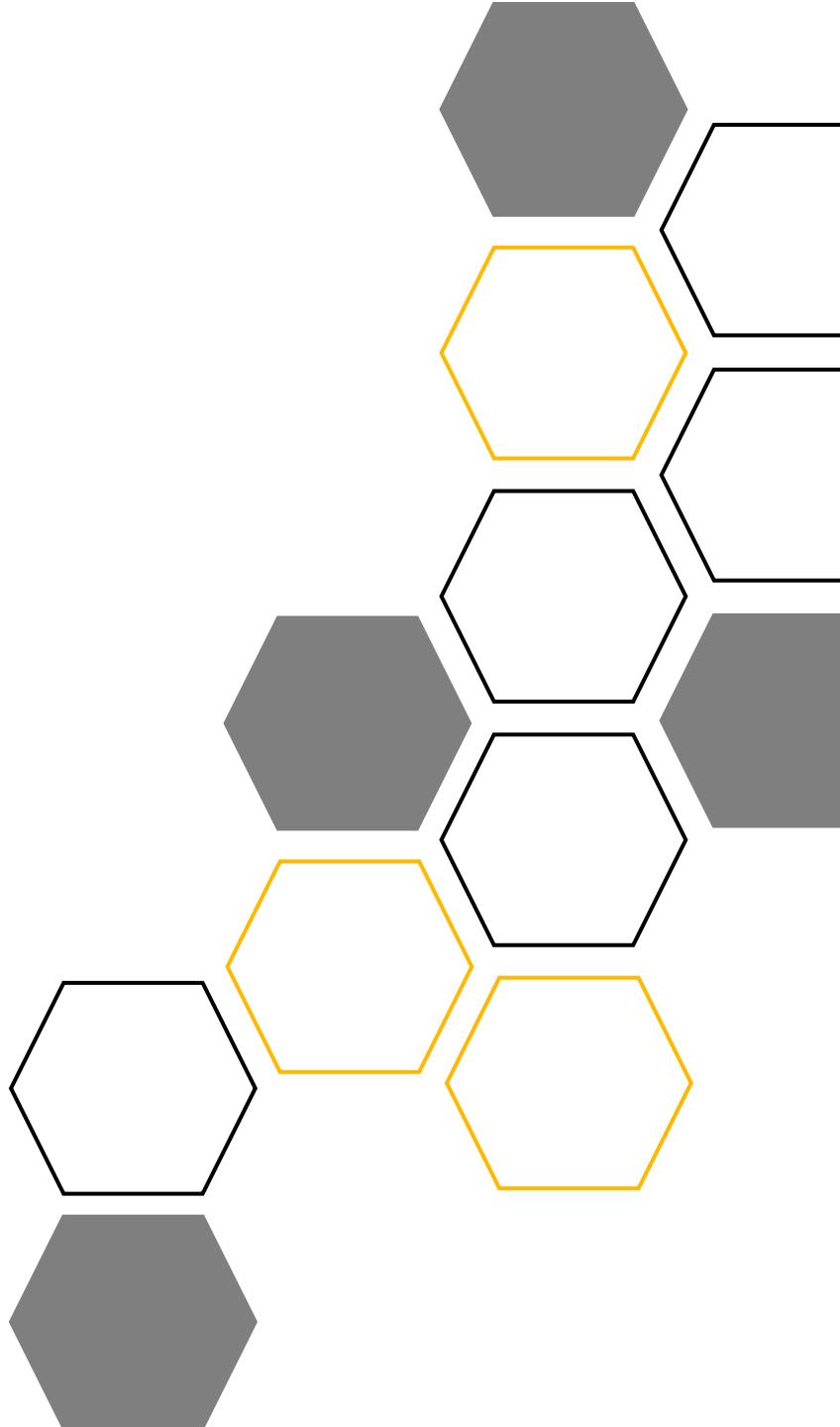


Defining and using aggregations

Jeroen ter Heerdt



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.

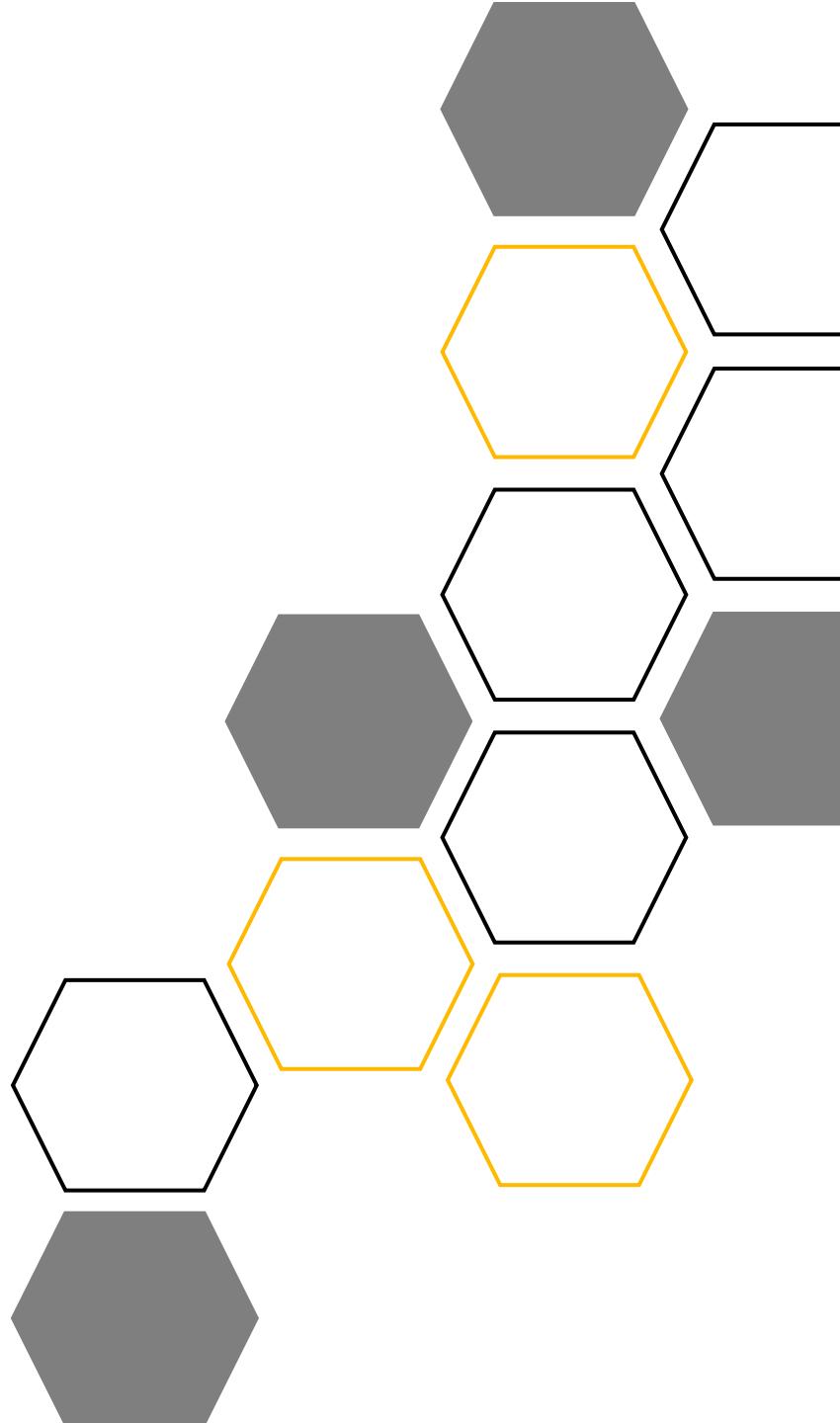
Mapping table sales

	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

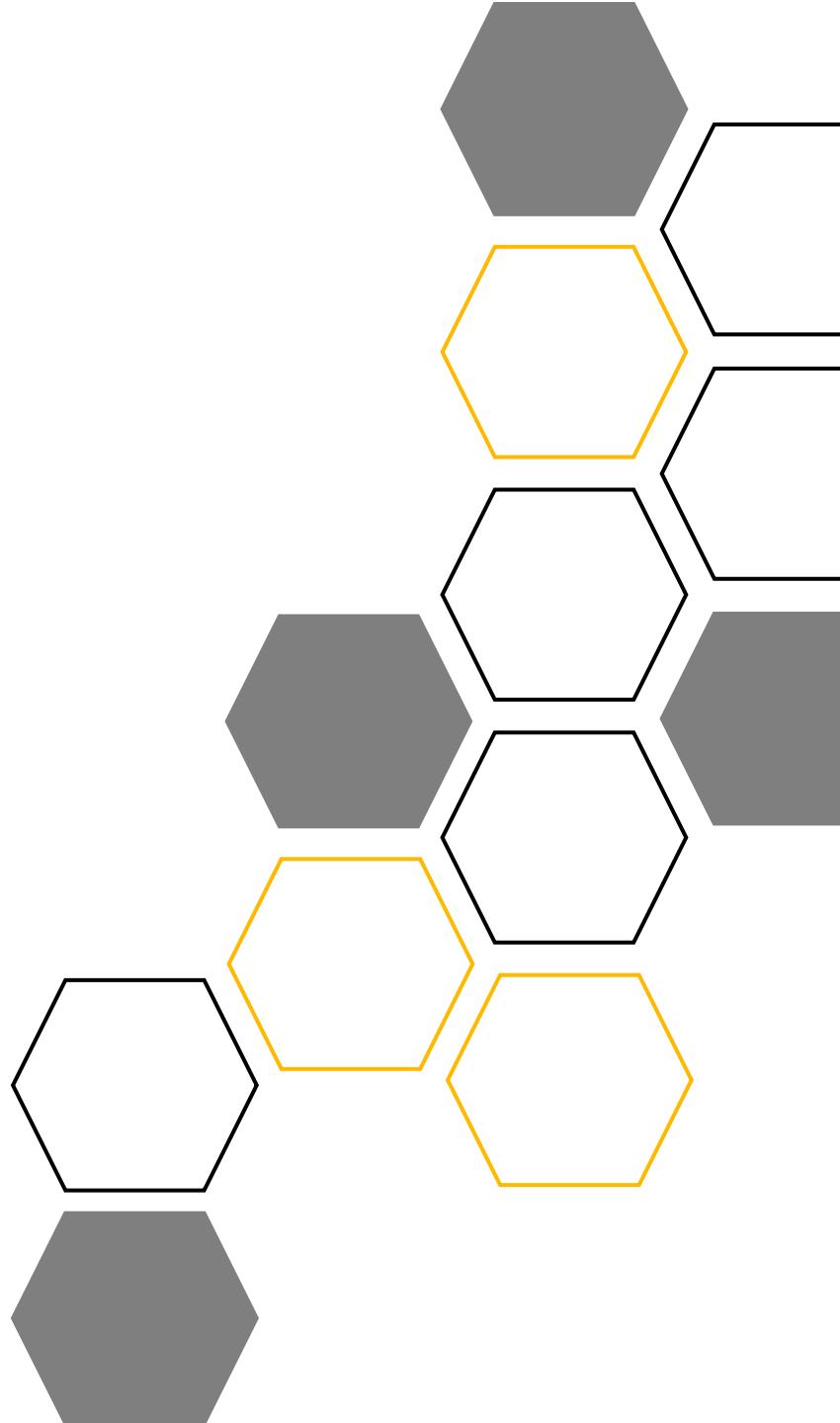
If you decide to append these, you
will get (blanks) when you use one of
the unmatched columns

Strategies for working with multiple fact tables

Jeroen ter Heerdt



Date and time dimensions

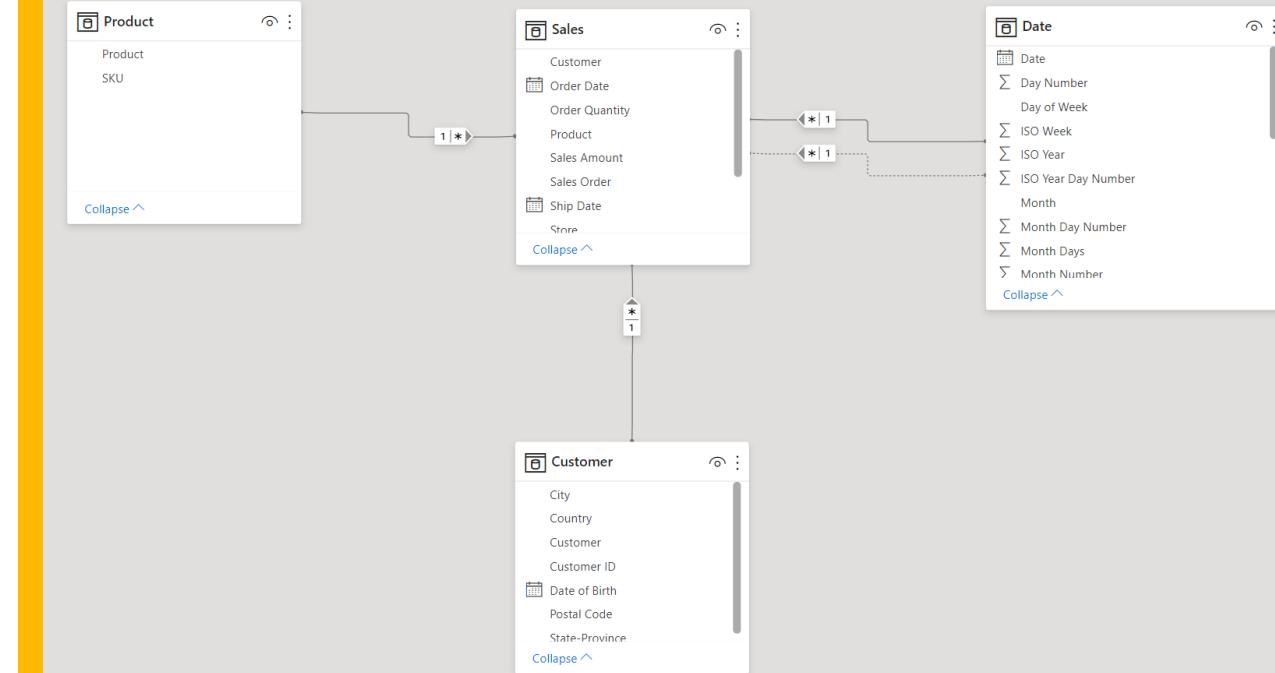


Auto date/time

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

