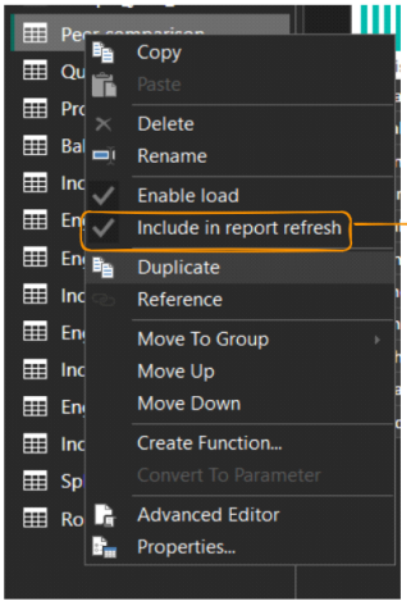
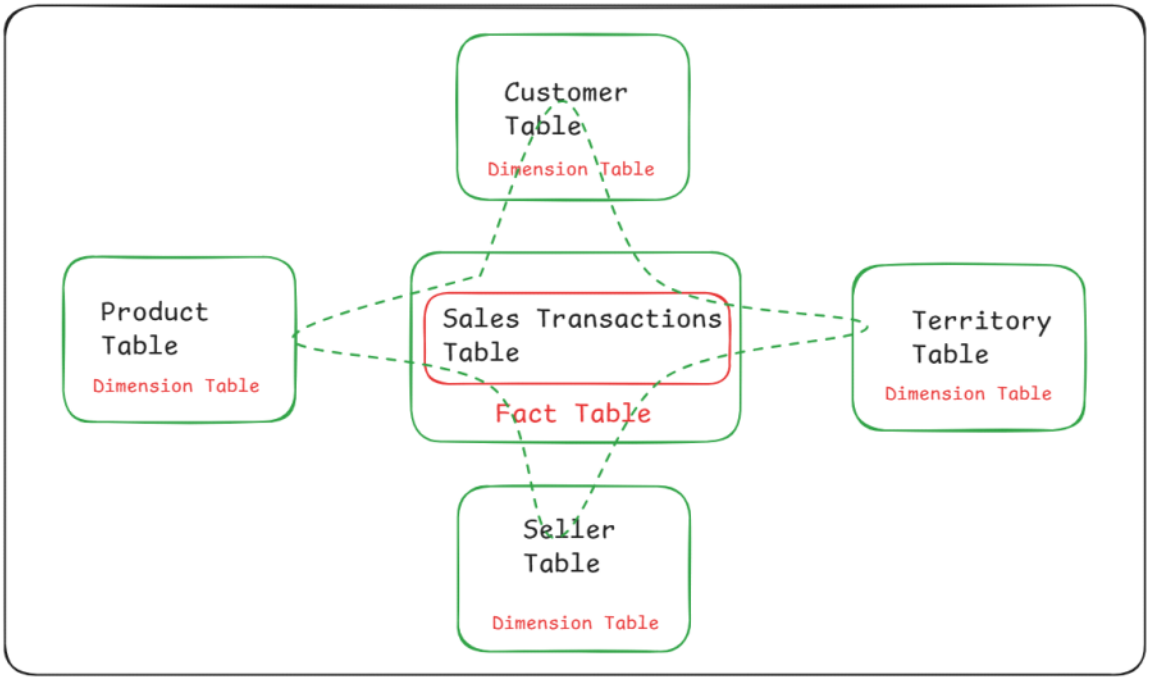
Data modelling & DAX Functions in Power BI

Learning Goals:

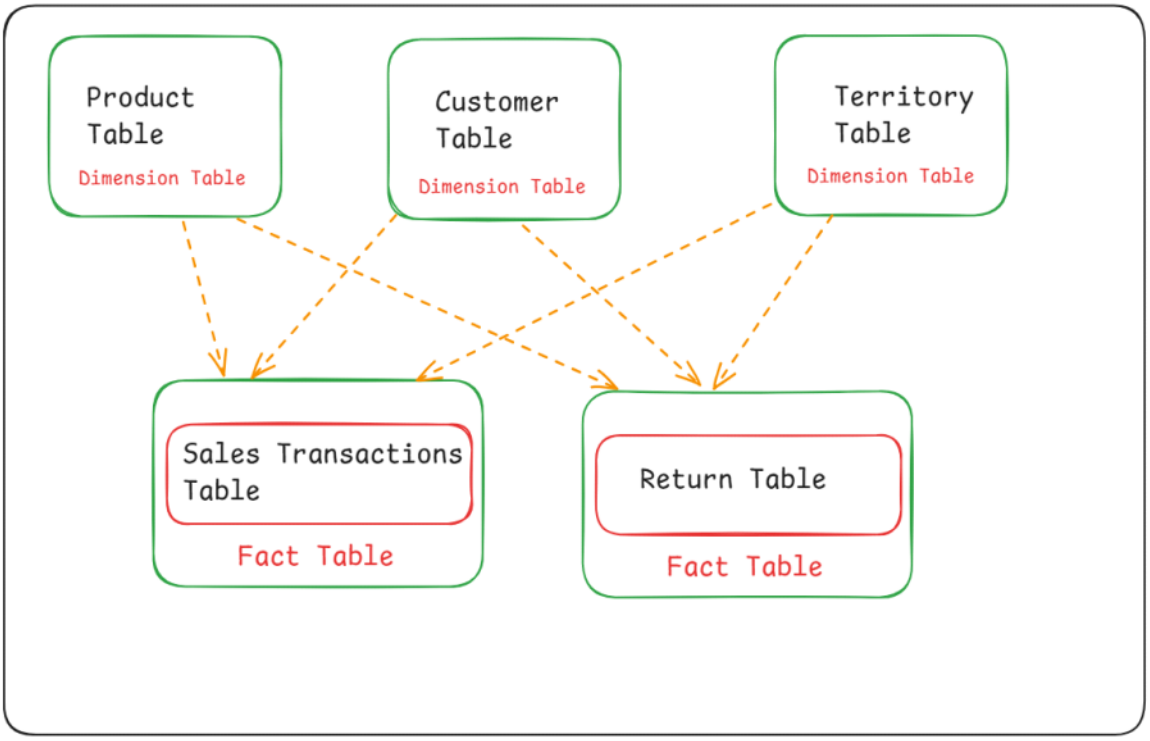
- What is DAX.
- DAX vs M-code.
- Calculated Column vs Measure.
- Implicit measure vs explicit measure.
- Filter Context vs Row Context.
- Different Categories of DAX Functions.
- Understanding Syntax Of Formula Language



Star Schema → Star schema is a relationship between atleast one fact table with multiple dimension table.

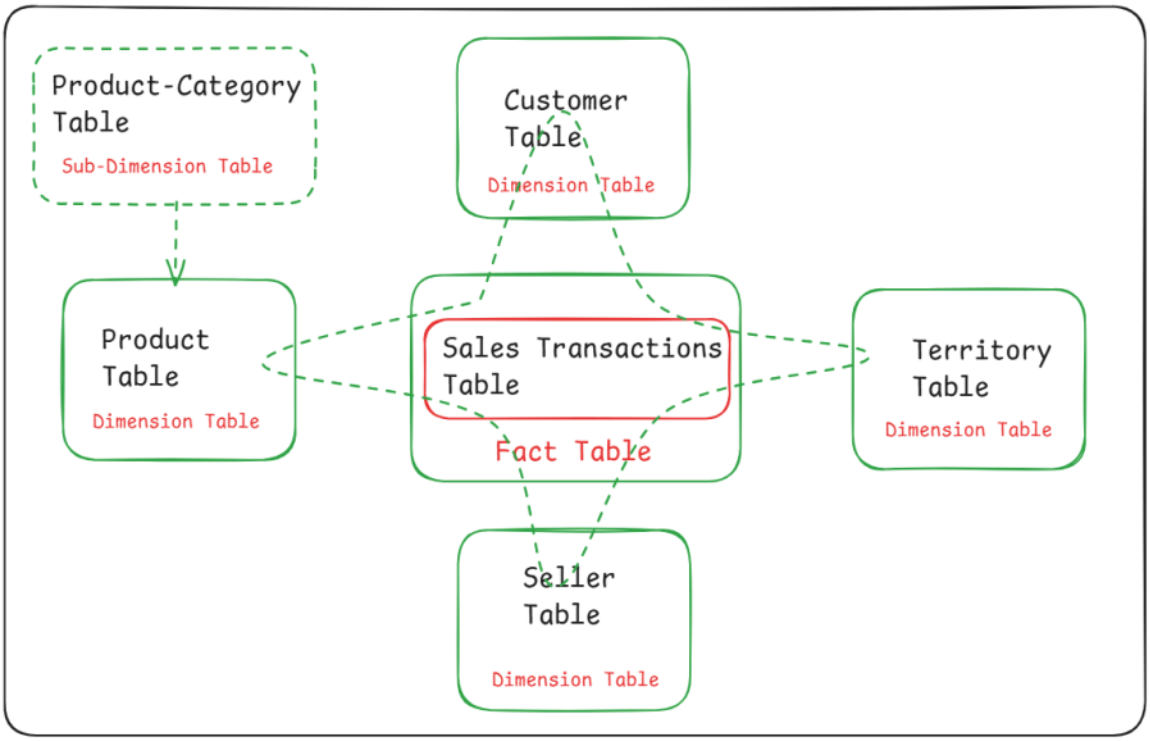


Star Schema → Star schema is a relationship between atleast one fact table with multiple dimension table.



Star schema with 2 Fact Table

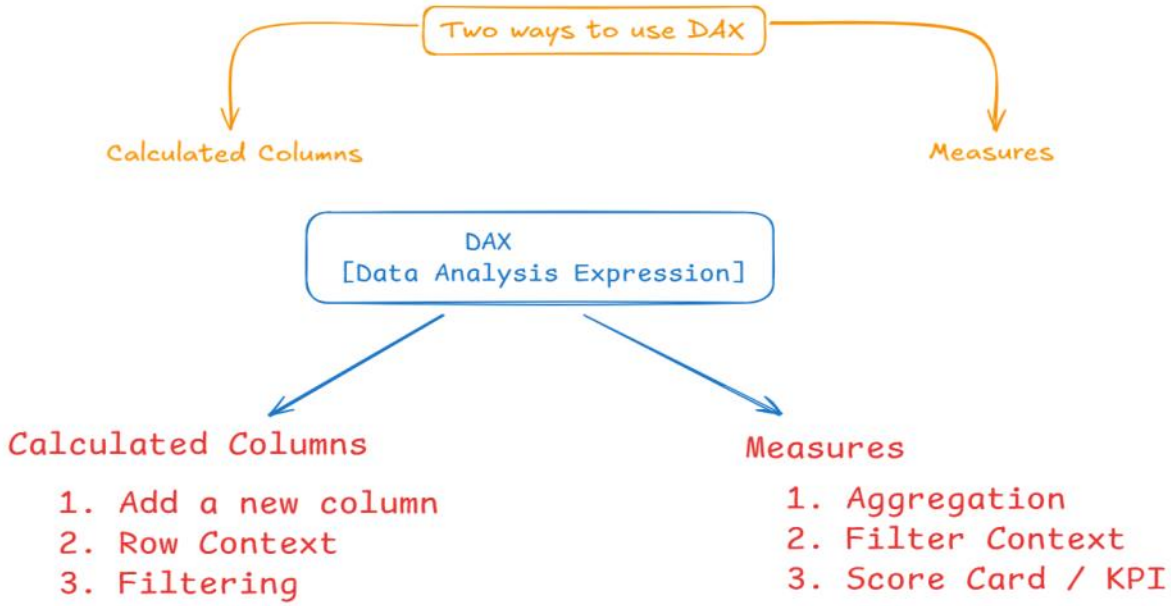
Snowflake Schema → snowflake schema is an extended part of star schema, which consist of sub-dimension table.



MEET DAX

Data Analysis Expressions (commonly known as DAX) is the formula language that drives the Power BI front-end. With DAX, you can:

- Go beyond the capabilities of traditional spreadsheet formulas, with powerful and flexible functions built specifically to work with relational data models.
- Add calculated columns (for filtering) and measures (for aggregation) to enhance data models.



M VS. DAX

M and DAX are two distinct functional languages used within Power BI Desktop:

- M is used in the Power Query editor, and is designed specifically for extracting, transforming and loading data.
- DAX is used in the Power BI front-end, and is designed specifically for analyzing relational data models

CALCULATED COLUMNS

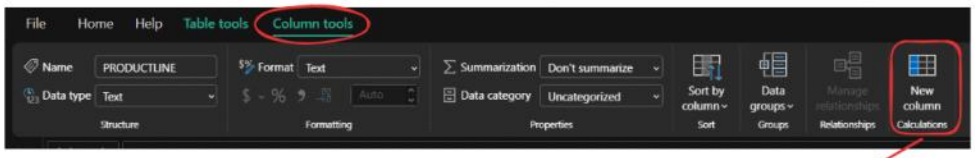
Calculated columns allow you to add new, formula-based columns to tables in a model

- Calculated columns refer to entire tables or columns (no A1-style cell references).
- Calculated columns generate values for each row, which are visible within tables in the Data view.
- Calculated columns understand row context; they're great for defining properties based on information in each row, but generally useless for aggregation (sum, count, etc.)

Data View

Row Context

QUANTITYORDERED	PRICEEACH
20	100
20	100
34	100
42	100
39	100
41	100
46	100
54	100
47	100
33	100
29	100
36	100
40	100
38	100
39	100
41	100
24	100
40	100
41	100
29	100
27	100
40	100
43	100
40	100
43	100
29	100



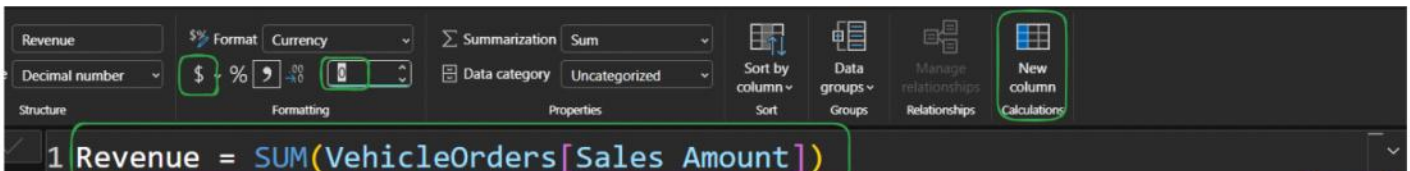

```
Sales Amount = VehicleOrders[QUANTITYORDERED] * VehicleOrders[PRICEEACH]
```

QUANTITYORDERED	PRICEEACH
20	$100
20	$100
34	$100
42	$100
39	$100
41	$100
46	$100

Sales Amount
$2,000
$2,000
$3,400
$4,200
$3,900
$4,100
$4,600

calculated column

row context



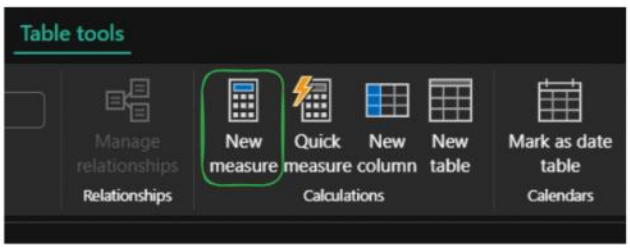

```
1 Revenue = SUM(VehicleOrders[Sales Amount])
```

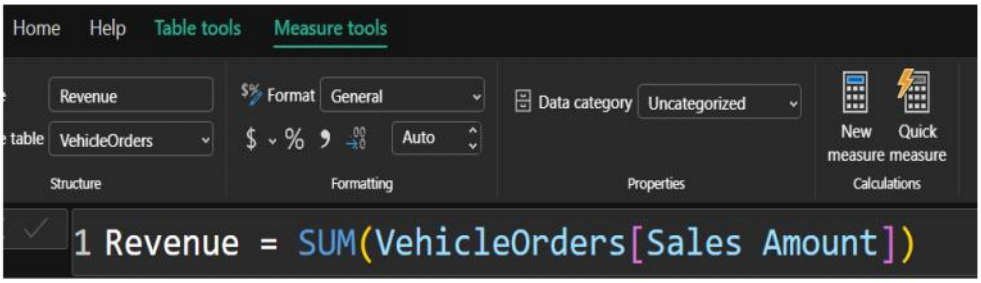
STATUS	PRODUCTLINE	CITY	STATE	POSTALCODE	COUNTRY	TERRITORY	CONTACTLASTNAME	CONTACTFIRSTNAME	DEALSIZE	Sales Amount	Revenue
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$2,000	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$2,000	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$3,400	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$4,200	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$3,900	$82,90,887
Shipped	Trucks and Buses	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$4,100	$82,90,887
Shipped	Trucks and Buses	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$4,600	$82,90,887
Shipped	Trucks and Buses	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$5,400	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$4,700	$82,90,887
Shipped	Classic Cars	Madrid	Unknown	28034	Spain	EMEA	Freyre	Diego	Medium	$3,300	$82,90,887

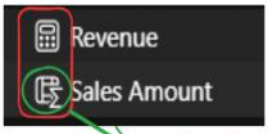
Memory Increases ↑
Efficiency Decreases ↓

All Values are same. That means Calculated columns are useless for aggregation.

In actual , its a Measure, that needs to be dropped in Report View







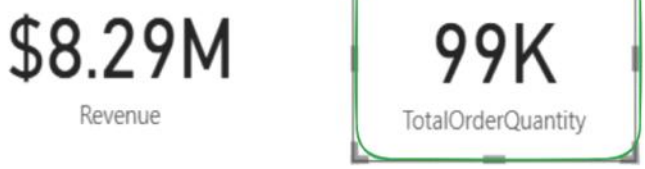
calculated column

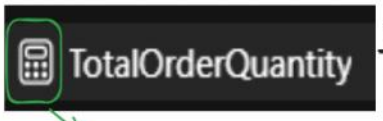
DAX MEASURES

Measures are DAX formulas used to generate new calculated values

- Like calculated columns, measures reference entire tables or columns (no A1-style cell references).
- Unlike calculated columns, measures aren't visible within tables; they can only be "seen" within a visualization like a chart or matrix (similar to a calculated field in a PivotTable).
- Measures evaluate based on filter context, which means they recalculate when the fields or filters around them change.

```
1 TotalOrderQuantity = SUM(VehicleOrders[QUANTITYORDERED])
```





rename to QuantitySold

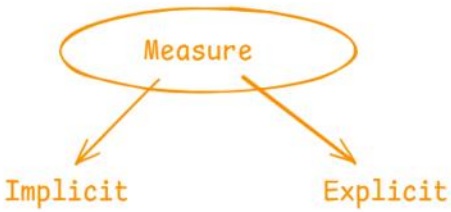
```
QuantitySold = SUM(VehicleOrders[QUANTITYORDERED])
```

Measure

What if I have to calculate TotalOrder [Unique Count of Invoice]

```
TotalOrders = DISTINCTCOUNT(VehicleOrders[ORDERNUMBER])
```

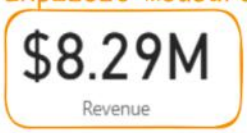
IMPLICIT VS. EXPLICIT MEASURES

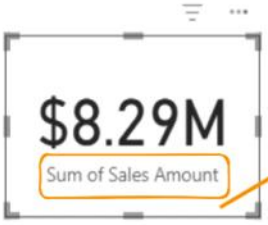


Implicit measures are created when you drag raw numerical fields into a report visual and manually select an aggregation mode (Sum, Average, Min, Max, Count, etc.)

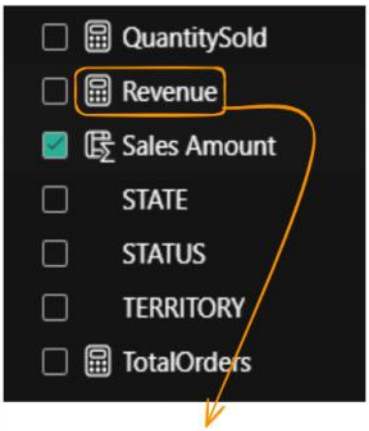
Explicit measures are created when you actually write a DAX formula and define a new measure that can be used within the model

Explicit Measure

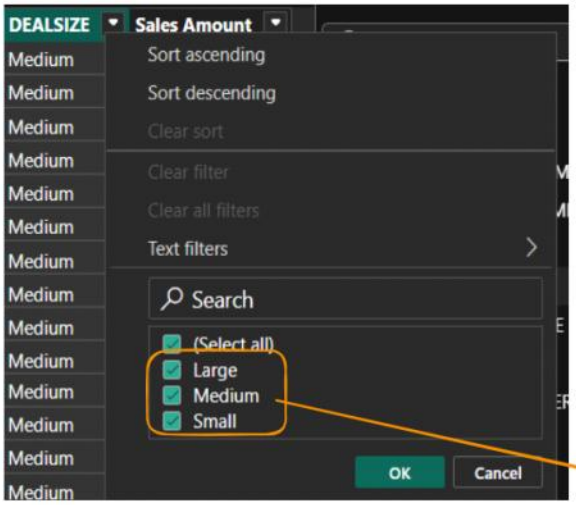




Implicit Measure



Reusing this explicit measures can be possible to write the complex measures



Individual Revenue

```
Medium Deal Size Revenue =
CALCULATE(
    [Revenue],
    VehicleOrders[DEALSIZE] = "Medium"
)
```

$4.96M

Medium Deal Size Revenue

RECAP: CALCULATED COLUMNS VS. MEASURES

CALCULATED COLUMNS	MEASURES
• Values are calculated based on information from each row of a table (row context)	• Values are calculated based on information from any filters in the report (filter context)
• Appends static values to each row in a table and stores them in the model (which increases file size)	• Does not create new data in the tables themselves (doesn't increase file size)
• Recalculate on data source refresh or when changes are made to component columns	• Recalculate in response to any change to filters within the report
• Primarily used for filtering data in reports	• Primarily used for aggregating values in report visuals