



Power BI Data Analyst 2

The Five-Layer model for Business Intelligence



Share

Collaborate and distribute insights to others



Visualize

View results, get insights, and play with reports and dashboards



Analyze

Define business views and implement calculations in analytical models



Prepare

Transform, cleanse, and combine data

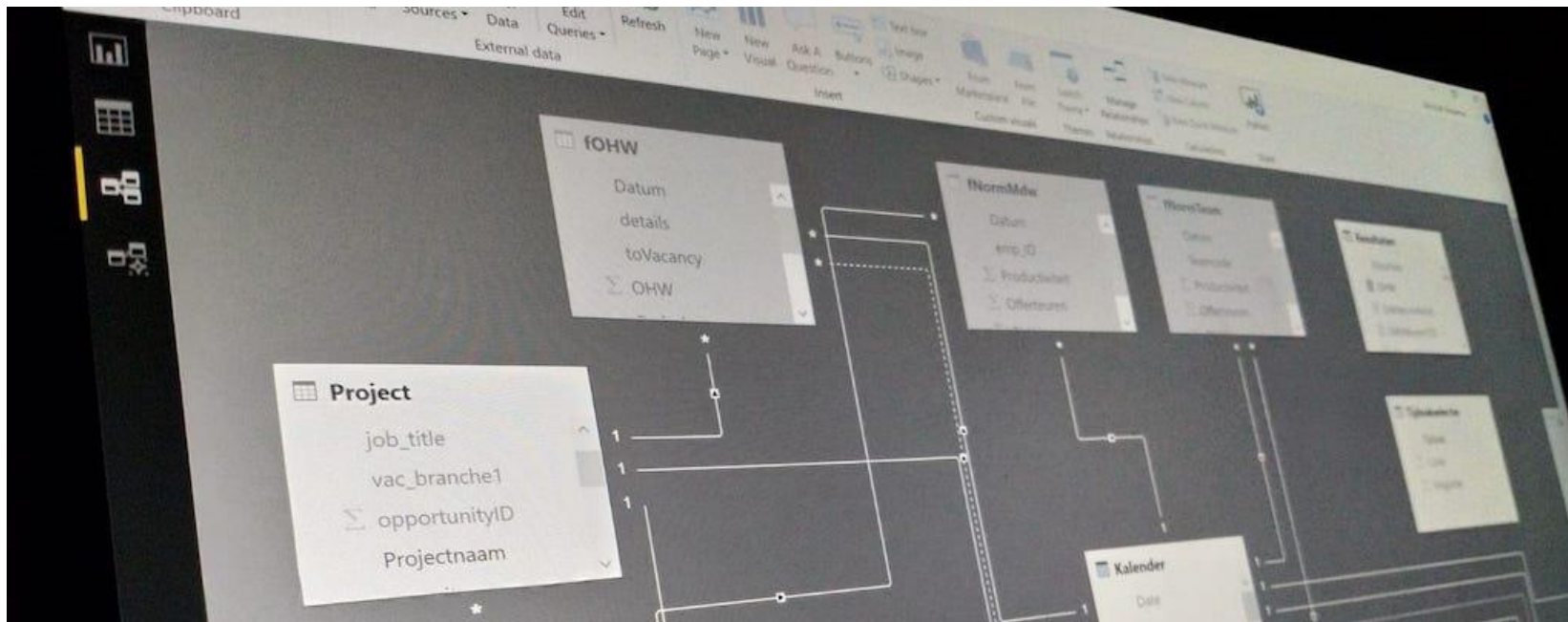


Connect

Discover data and extract data from source

Data analytics with Power BI

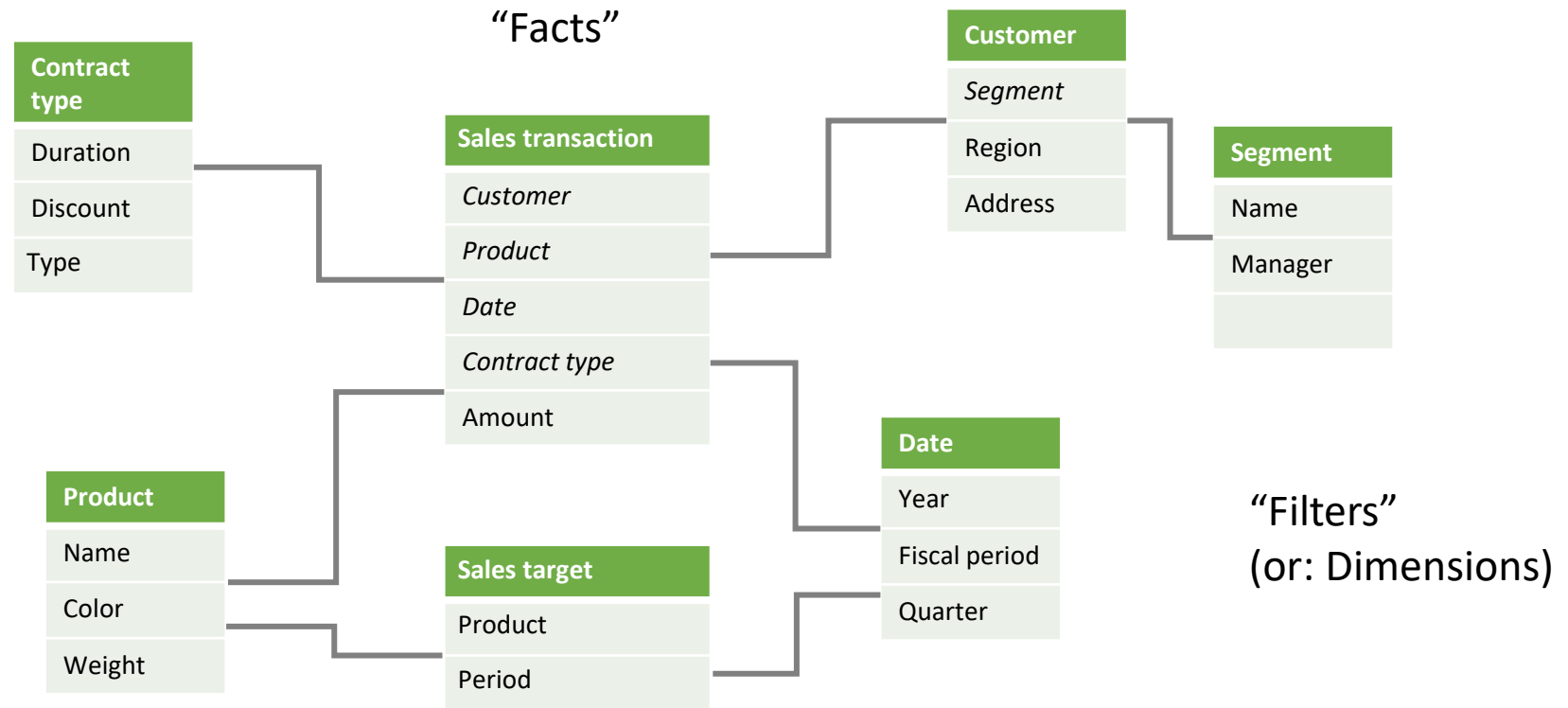
Power BI modeling



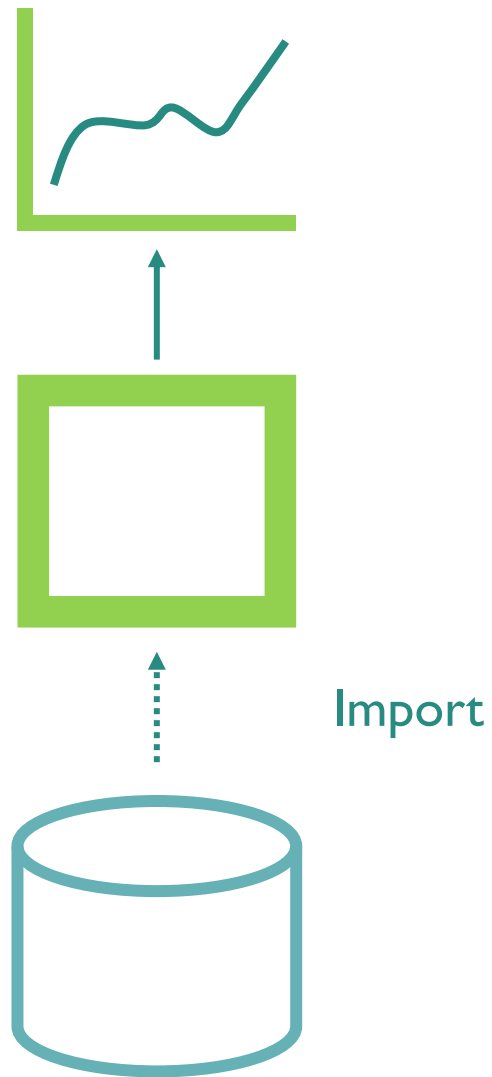
The Power BI/Tabular engine

- Power Pivot/Tabular is a database technology
- Data is stored in-memory
- Data is stored per column, instead of per row
- As a consequence, **hyper-compression** can be achieved
- And models are blazing fast
- But depending on the amount of RAM
 - And of course, you can mess things up!

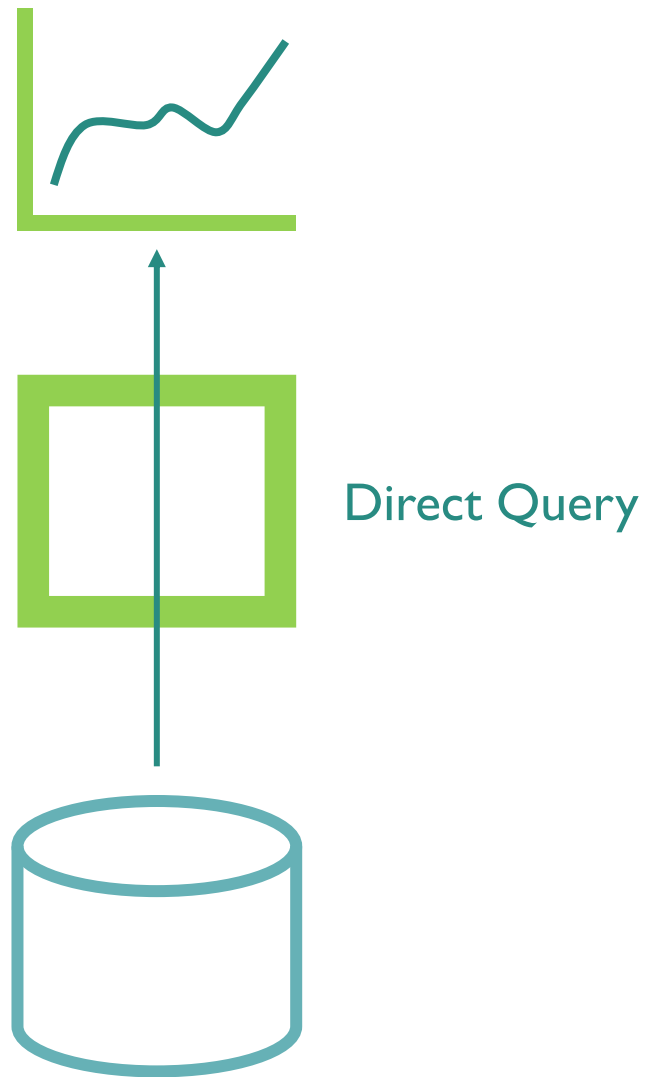
Structure of a model



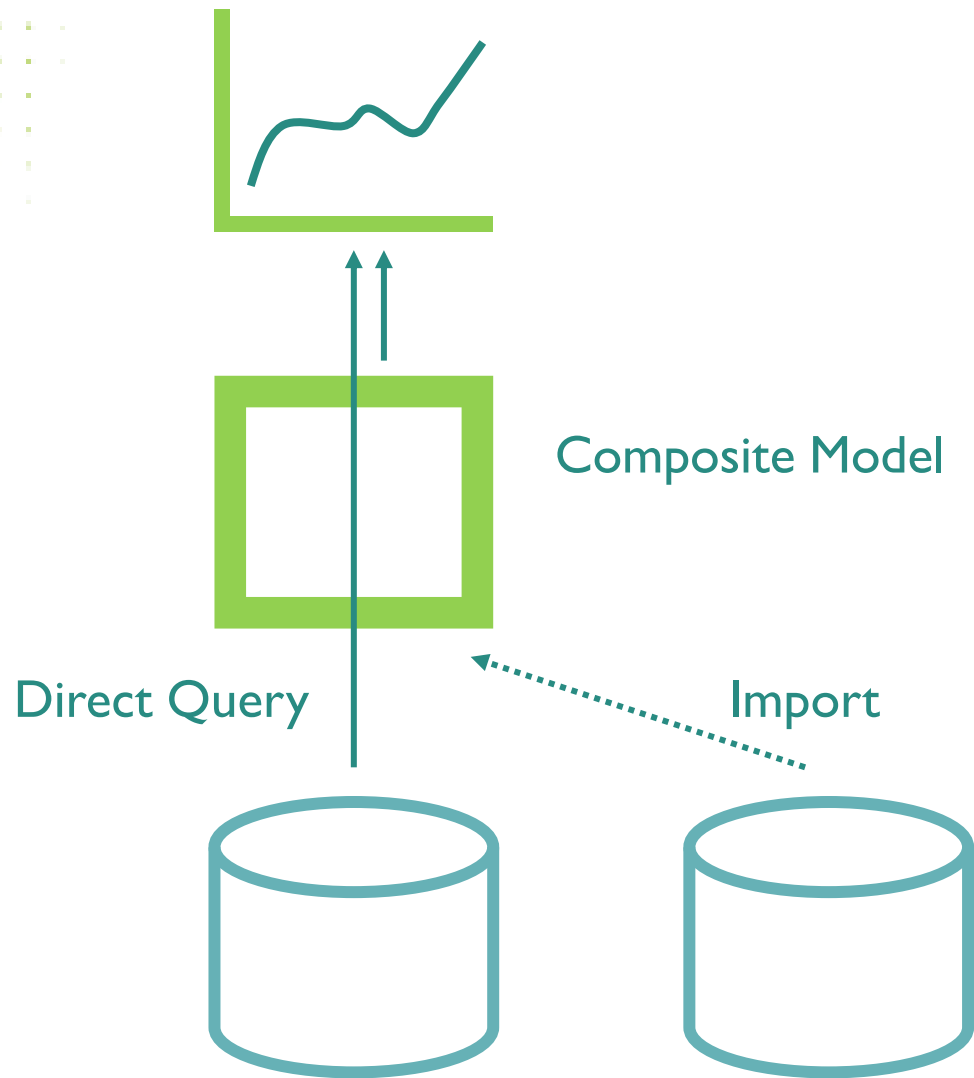
Power BI Data Model storage modes (I)



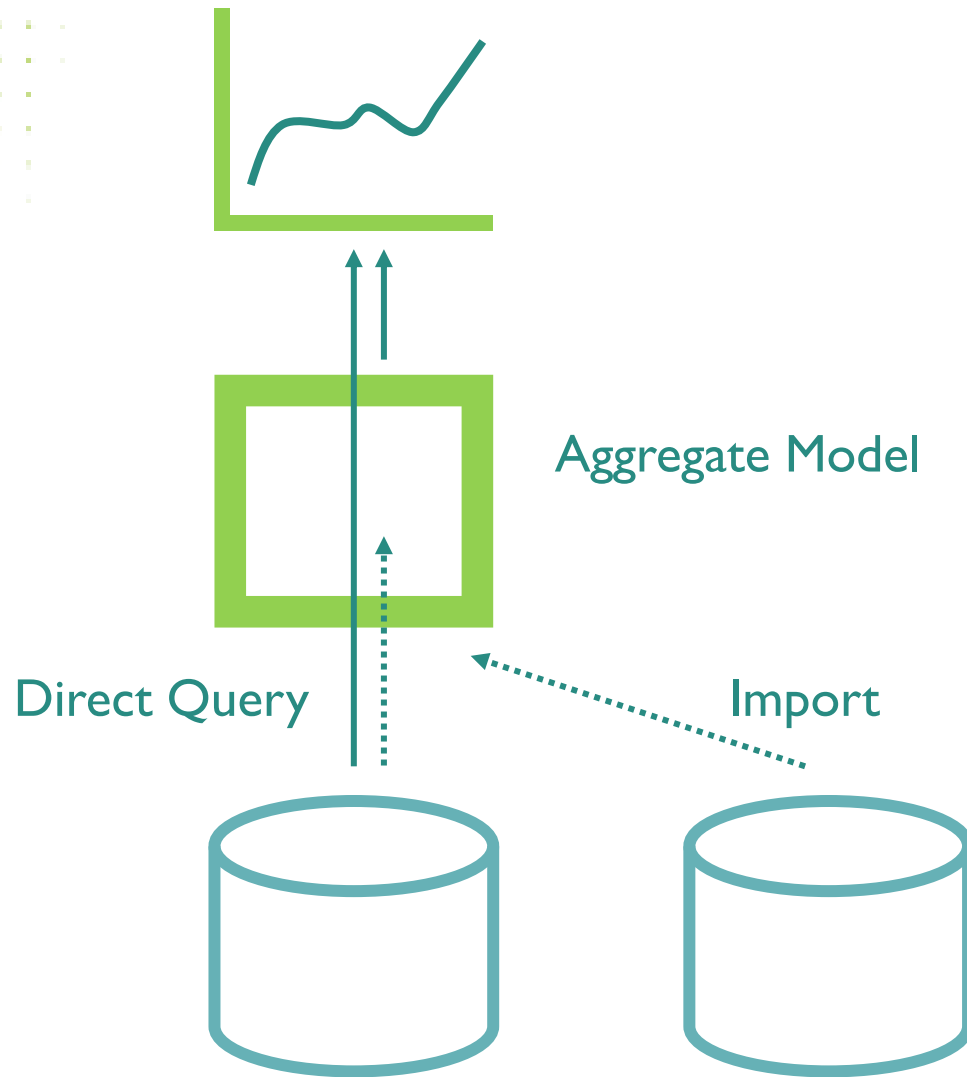
Power BI Data Model storage modes (2)



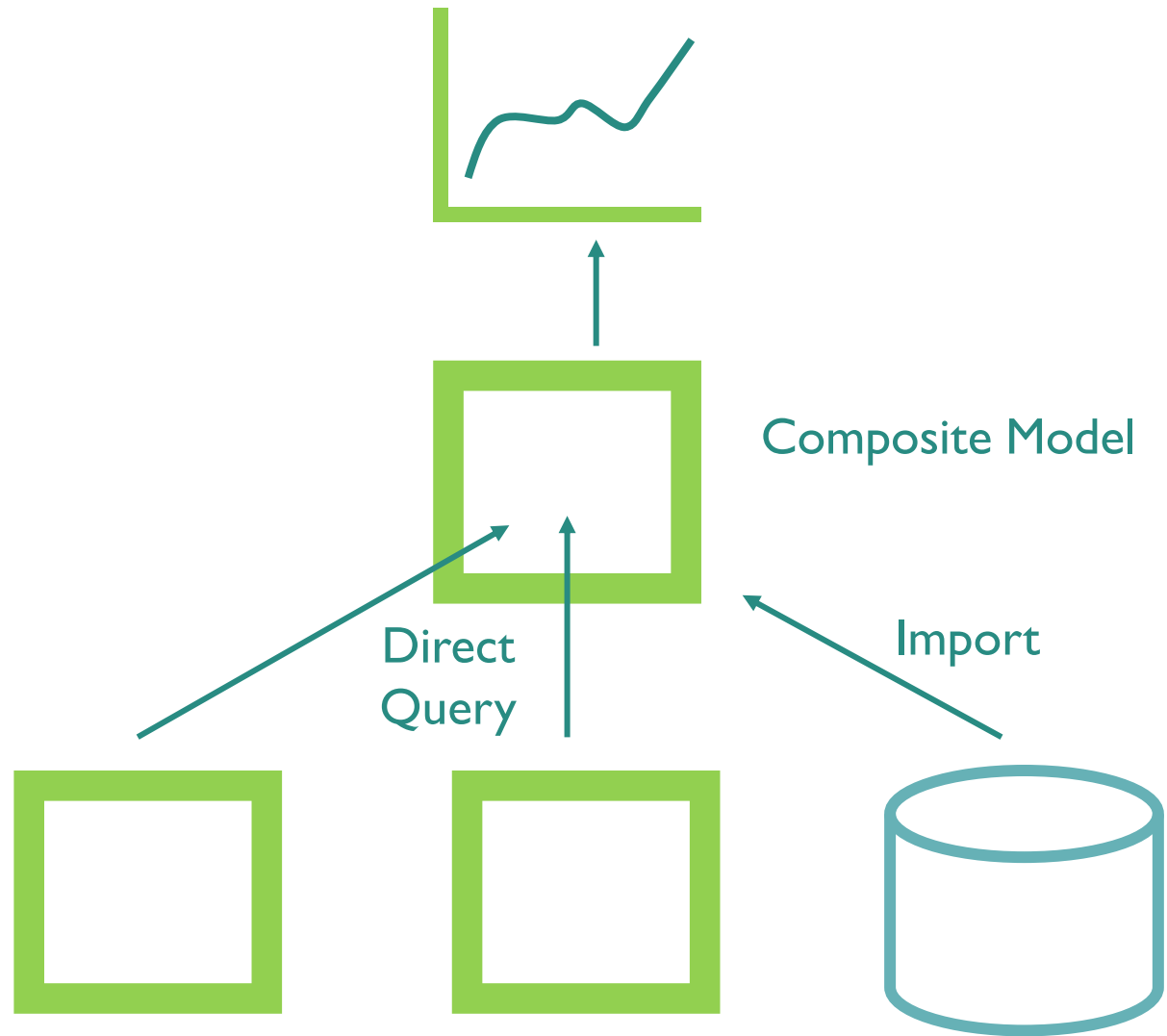
Power BI Data Model storage modes (3)



Power BI Data Model storage modes (4)



Power BI Data Model storage modes (5)



Considerations

- Don't load what you don't need NOW!
 - Adding columns is easy, but you'll probably never find time for cleaning up unused columns
 - Think about rows as well. Do you really need data from 2010?
- Think about the analyses needed (DAX calculations) before deciding on table design

How to complicate the life of a Power BI model

- Load (long) tables with many, many columns
- Have columns with many unique values
- Use expensive data types and outliers

And therefore...

- Implement calculations in calculated columns
- Try to implement all logic in a data warehouse
 - You can only load this through (many) columns

About Data Types

- Internally, the engine only works with whole numbers
- All data needs to be converted, or encoded, to whole numbers

“Good” data types

Whole number

Fixed decimal number

True/False

Date

“Bad” data types

Decimal number

Text

Date/Time

Binary

(Well, it's more subtle than this...)

Hash encoding

How to convert text values to numbers:


Customer ID	Seller	Active
3542	John	True
3544	Denise	True
8645	Richard	True
6987	Denise	False
2376	Anna	True
1875	Tom	True
3967	Denise	True

1	John
2	Anna
3	Richard
4	Denise
5	Tom

Customer ID	Seller	Active
3542	1	True
3544	4	True
8645	3	True
6987	4	False
2376	2	True
1875	5	True
3967	4	True

Dealing with outliers

Customer ID	Name	Active
1	Microsoft	True
2	Apple	True
3	Google	True
4	Amazon	False
5	Facebook	True
99999999	Unknown	True



1	1
2	2
3	3
4	4
5	5
6	99999999

The concept of relationships

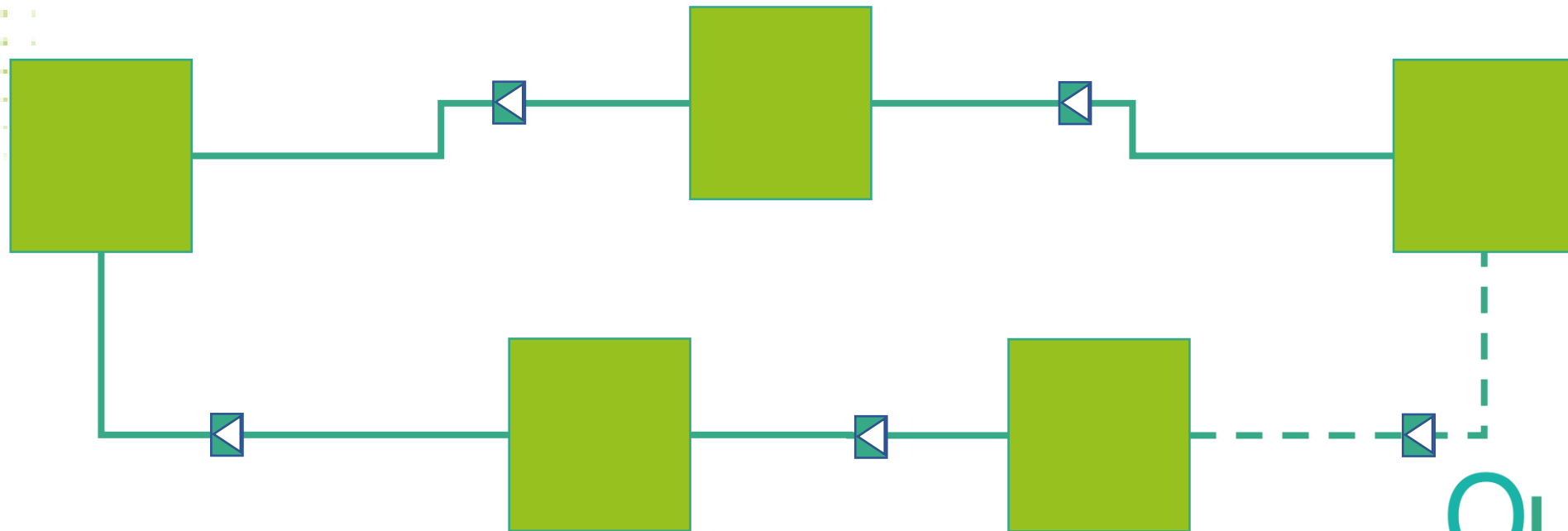
Product ID	Amount	Date
223	23,00	22-6-2019
223	24,50	23-6-2019
225	53,80	23-6-2019
223	12,10	24-6-2019
226	25,00	26-6-2019
228	33,75	27-6-2019
221	38,20	27-6-2019

Product ID	Product
221	Coffee
222	Tea
223	Lemonade
224	Water
225	Beer
226	Orange Juice
228	Tomato Juice



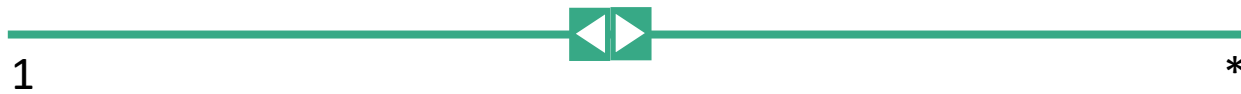
Active and inactive relationships

- Because relationships join table automatically, only one active path of relationships can exist between any two tables
 - Enable an inactive relationship/path with USERRELATIONSHIP function

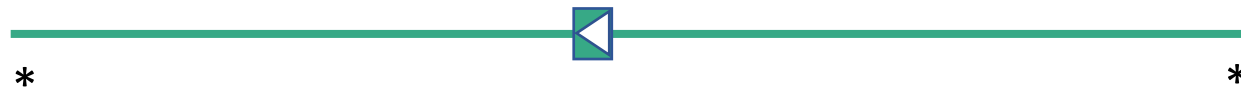


Other types of relationships

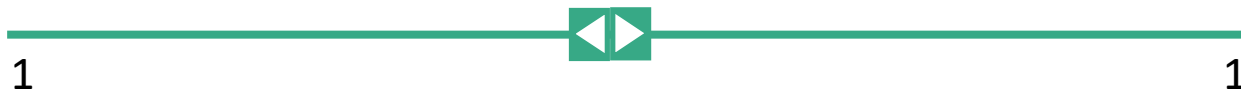
- Relationships with double crossfilter direction



- Many-to-many



- One-to-one



* Use these only for specific reasons and when you know what you're doing!

Considerations for relationships in Power BI

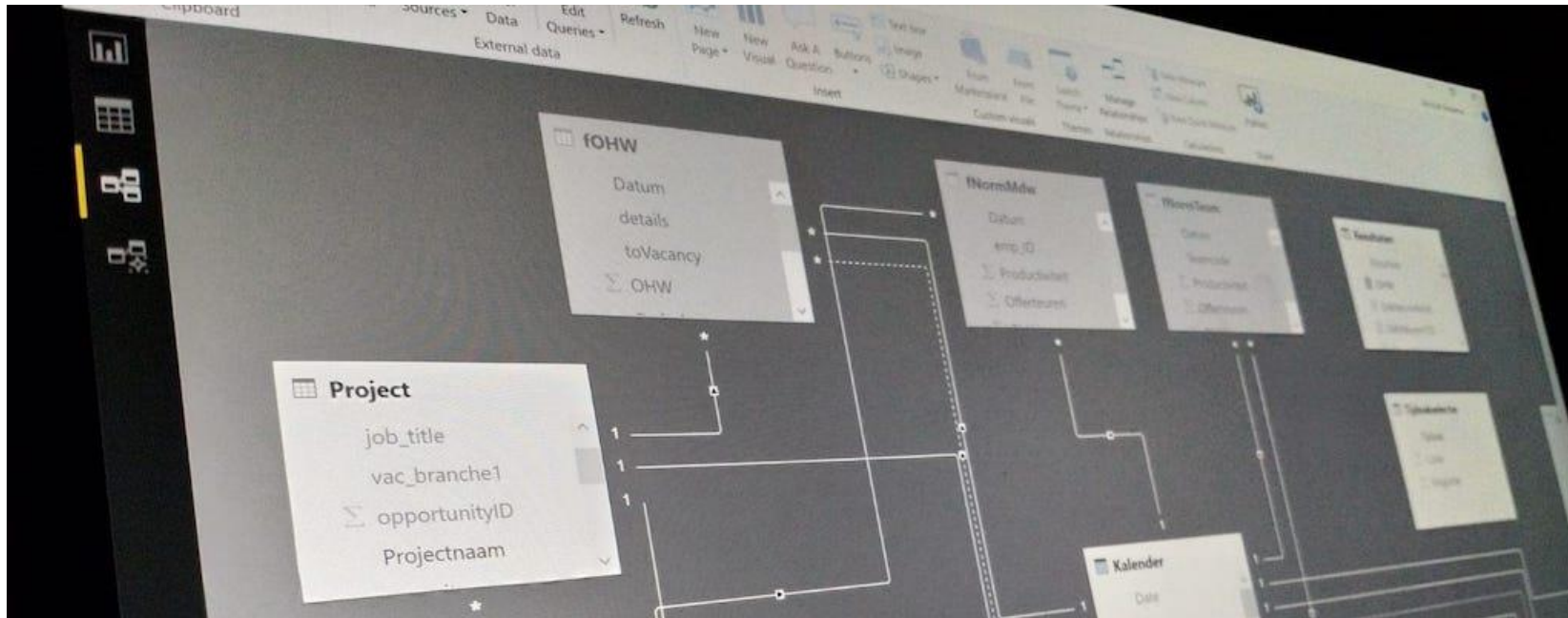
- Relationships are between single columns only
- For relationships, less key values is better (keep it < 100k)
- Use 'good' data types for relationship keys when possible
- Don't load data you don't need
 - E.g. Calendar running from 1900 to 2100 ?!
- Double crossfilter direction is slow, use only on small tables
 - Or even better, use it only in DAX measures with CROSSFILTER

Relationships in the data model

- Go from a source table to a lookup table, and from a source column to a lookup column
 - This is comparable to foreign keys and primary keys in relational databases
- The data model supports relationships on single columns only
- The lookup column must contain **unique values**, even if they're blank
- Not every value in the source table has to exist in the lookup table – missing values are linked to a blank row
 - So relationships are not constraints as in relational databases
- It is possible to have multiple relationships between tables, but only **one** can be **active**
- **Best practice**: hide the source column!

Data analytics with Power BI

Calculations with DAX



Calculations in tabular models: DAX

- Calculated columns
 - `Amount = [Price] * [Number]`
- Calculated tables
 - `Calendar = CALENDARAUTO()`
- Measures (calculated fields)
 - `Total sales = SUM(fSales[Amount])`
- Security filters
 - `= dEmployee[Login] = USERPRINCIPALNAME()`
- DAX Query
 - `EVALUATE(Customer)`

Avoid these!

Use for specific extensions of a model

Use every time!

Needed for security within a model

Not needed when working with Power BI reports

Calculated columns vs. Measures

Calculated column:

- Adds a column to a table, based on a calculation
- Is computed at time of creation or data refresh
- Is stored in the data model and thus makes the model larger

Measure:

- Adds an aggregation to the model that can be used in reports
- Is computed at time of use
- Doesn't take up space in the model (but does take processing time)

Tips and guidelines

- Create one or more measure tables for storing measures
 - We usually call them Results
 - Within the measure table, use display folders to group measures
- A measure can be used as an argument for a DAX function. Use this to nest measures, keep formulas short, and reuse logic
- Measures can be hidden - use this for intermediate calculations
- Think carefully about measure names: these must be unique in the model
- Provide a default format for each measures
 - Power BI will try to derive this from the columns referenced