# TIPS

* **Axis Value – Display units**
* **Show Constant Line name - Data label – Style – Name**
* **Custom visuals - Microsoft AppSource**
* **Format column – Column Tools**
* **Navigate in - Drill Down and Drill Mode (form of cross filtering)**
  + **Two down arrows - move down to the next level of the hierarchy**
  + **Split arrows down - expand down to the next level of the hierarchy**
  + **Single down arrow - enable Drill Mode – selecting a level will drill down and expand all elements in that level**
* **Drill mode – form of cross filtering - Format – Edit interactions – Turn off interactions**
* **Row context iterator in Measures E.g. Orders Above Target Profit margin = SUMX(Orders,IF(Orders[Profit\_Margin]>Orders[Target Profit Margin],1,0)) – iterates over each row and count number of orders with profit margin above target profit margin**
* **Create a table as placeholder for measures – Home – Enter Data – Measures - \_Calculations – Delete measures Column**
* **Apply bin size - right-click Field - Choose New group – Choose Bin size**
* **SWITCH() – Create Column with DAX to categorize - evaluate expression against list of values and return one of multiple result expressions - used to avoid multiple nested IF statement**
* **Use Legends in charts to break out categories**
* **Nested DAX IF statements to Categorize 3 – E.g. Demographics = IF('Databel - Data'[Age]>=65, "Senior", (IF('Databel - Data'[Age]<30, "Under 30", "Other")))**
* **Nested DAX IF statements to Categorize 3 E.g. Grouped Consumption = IF('Databel - Data'[Avg Monthly GB Download]<5, "Less than 5 GB", (IF('Databel - Data'[Avg Monthly GB Download]> 10, "10 or more GB", "Between 5 and 10 GB")))**
* **Turn off Stepped Layout in Matrix**
* **Conditional formatting Table - Cell elements - choose column - enable for font colour - conditional formatting button- Add Rule**
* **Conditional formatting Visual – Format – Colours – fx**
* **Best Practice – Overview page canvas settings – 1640px width x 960px height**
* **Use gradient fx for bubbles**
* **Add multiple fields for Slicers / Filters**
* **Add Markers – Style**
* **Fixed Decimal Number – more appropriate for currencies**
* **ADD FILTERS IN DATA PREVIEW**
* **DISTINCT - THE TOTAL NUMBER OF DIFFERENT VALUES REGARDLESS OF HOW MANY TIMES IT APPEARS IN THE DATASET - A NAME THAT APPEARS IN THE LIST MULTIPLE TIMES IS COUNTED AS 1 DISTINCT COUNT**
* **UNIQUE - TOTAL NUMBER OF VALUES THAT ONLY APPEAR ONCE - E.G. THERE ARE 740 DISTINCT NAMES IN THE DATASET AND OUT OF IT THERE ARE 485 NAMES HAS 1 RECORD ONLY**
* **Power Query Editor - FILL Transformation – Up / Down – Takes column and transverses through the values in it to fill any null values in the next rows until it finds a new value**

* **FILL only useful if sorting by meaningful column in ascending order**
* **Always TRIM & CLEAN text data**
* **Transform – Standard – Divide (‘000)**
* **Transform – Scientific – ABSOLUTE VALUE**
* **Transform - Date (date functions) – text data – ALWAYS Duplicate column first**
* **Transform – Round**
* **Multiply by – 1 to have negative values**
* **Define base changes in Power Query and perform fine-tuning steps in Power BI**
* **Sort column by another column – Power BI > Sort Column**
* **Preview – Shape Maps Azure Maps**
* **Hide Tables in Power BI Report View**
* **Keep characters before 2nd delimiter – First extract last x characters – Extract again using delimiter**
* **Report View - Home - Manage Relationships**
* **Snowflake Schema - Dimension Tables that join to each other do not join with Fact Table**
* **Snowflake Schema - Join to the lowest hierarchy**
* **Relationship view - dotted line - make relationship active**
* **Demote / Promote headers before transposing / transforming / pivoting / unpivoting**
* **Merge Queries - Second table converted to column – Expand – Choose columns**
* **Merge Queries – Can merge 2 tables more than once using different keys**
* **Merge Queries - Can merge more than 1 column (shift + select columns)**
* **Ranking - Index Column – Add Column – Dropdown – From 1**
* **DIVIDE function has the advantage of not throwing an error when the denominator is zero**
* **EDIT RELATIONSHIP IN MODEL VIEW**
* **DOTTED LINE – INACTIVE RELATIONSHIP**

# INTRODUCTION TO POWER BI

## Getting Started with Power BI

## Star Schema (common database structure)

* Fact tables – events or transactions
* Dimension tables - more information about each transaction
* Create relationships through key columns

### Fact table and Dimension tables

* **Must have a main fact table**
* **And related dimension tables**

### 3 Views (top left)

1. Report (default) – create reports and visuals
2. Data – see associated data
3. Model – see and manage relationships among table in data model

### Report View

* Canvas area – visualizations are created and arranged
* Filters pane – filter data visualizations
* Visualizations pane – add, change or customize visualizations
* Fields pane – shows available fields
* Can drag filters and visualizations panes to the canvas
* Change filters and axes
* Drag between X and Y axis to segment
* Search in Fields Pane
* Select graph and change in Viz Pane

### Slicers and Tables

* Add additional fields to existing reports
* Slicer – automatically modifies all connected visualizations
* Use multiple filters and slicers
* Dimension table
* Match page name to title
* Format Slicer (filter) in Visualizations pane
* Chart titles are automatically generated
* Use eraser to clear selection
* **Filter types : Visual Level | Page Level | Report Level| Drill through**

## Transforming Data

### Dataset Errors

* Unneeded columns
* Inconvenient, inconsistent data
* Extra characters
* Blank rows

### Cleaning Data

* Transform before loading
* Power Query Editor (M language) - format dataset
* Opens in separate window – need to close and apply
* Applied steps tracks changes – can revert to earlier version
* Use first row as headers
* Edit query in Report view
* Power Bi automatically sums visualizations – can change aggregation in Visualizations pane

### Transforming and Formatting Data

* Power Query Editor Menu options
* Replace data type / blanks
* Format in Column Tools
* Set Summarization

### Map Visualization

* Set Data Category
* Map Visualization – Globe icon
* Set bubble size

## Visualizing Data

### Visualization Options – Specific data series values across different categories

* Bar | Column charts – look at specific data series values across different categories
* Bar chart – vertical
* Column chart – horizontal
* Stacked Bar | Column chart – includes multiple elements - entire Bar | Column is the total
* Clustered Bar | Column chart – includes multiple elements located next to each other
* 100% Stacked Bar | Column chart – total of each stacked bar always equals 100%
* Combo chart – combines column chart + line chart
* Line chart – multiple lines in one chart – overall shape of entire value series, usually over time
* Area charts – based on line-charts with the area between the line and axis filles in
* Pie | Doughnut charts – relationships of parts to a whole
* Tree maps - relationships of parts to a whole

### Visualization Options – One, two or more values for showing the overall level of performance

* Cards – single value
* Mulli-row cards – multiple values
* Gauge | KPI – actual data compared to budgeted data
* Table - detailed text data in tabular format – grid of related data in multiple columns
* Matrix - detailed text data in tabular format – can be expanded and collapsed by rows or columns

### Change and edit different visualizations

* Click visualization and chose another visualization icon
* Change properties of visualizations - Formatting icon Paintbrush – above visualizations pane
* Fonts – under values
* Edit table column widths in the table

### Sorting and more formatting

* Click on 3 dots for more options – sort / sort axis / sort by
* Modify in data view – select column – Sort by column options
* Change size – Formatting options – general – size
* Change slicer – format – visual – slicer styles
* **Change theme – View menu - similar to Office apps**

## Filtering

### Drilling up and down and expanding

* Must have hierarchies
* Hover over visual to use icons in action bar or right-click
* **Drill down all fields at once** - Select double arrow drill-down goes to next level in hierarchy -
* Each step in the path shows new information
* To go up a level in the hierarchy - select the single upward arrow drill-up icon
* **Drill down one field at a time –** Select drill-down icon to turn it on - option of drilling-down one field at a time
* **Selecting a visual element without turning on drill-down option cross-filters other charts on report page rather than drilling down**
* Expand option adds another hierarchy level to current view
* **Expand all fields at once -** Confirm drill-down option on and select expand down icon - Discrete
* **Expand one field at a time -** Confirm drill-down option off and select expand down icon
* **Show the data as a table to look at how drilling works**
* **Navigate in Drill Down and Drill Mode (form of cross filtering)**
  + **Two down arrows - move down to the next level of the hierarchy**
  + **Split arrows down - expand down to the next level of the hierarchy**
  + **Single down arrow - enable Drill Mode – selecting a level will drill down and expand all elements in that level**
  + **Drill mode – form of cross filtering**

### Filtering

* Display data based on selected criteria
* Filter out unnecessary information
* Levels of filters – Visualization level | Page level | Report level
* **Turn off filtering – Don’t allow end user to change filters – turn off interactions**
* **Select a Report and look at Interaction icons on other reports - Turn off so other reports do not change**
* **Select visualization - Format – Edit interactions – Turn off interactions**
* Use Filter Pane – drag values
* Possible to filter on single report or all pages
* Some Filter options – Top n
* Drag any Field to Filter pane

### Underlying data and hierarchies

* 3 dots on visualization – show as table
* 3 dots – export data
* Create hierarchy – Data view or Report View – Right click column name – Create hierarchy
* Rename hierarchy – double click
* Add columns to hierarchy by right clicking on column names
* Edit hierarchy order – Model view - Properties pane
* Use hierarchy – Select visualization – change axis to new hierarchy
* New controls appear with drill down controls – turns black if enabled
* **Show Date without hierarchy – Values - Right click – Select Date instead of Date Hierarchy**

# INTRODUCTION TO DAX IN POWER BI

## Getting started with DAX

### DAX

* Data Analysis Expressions
* Formula language to create calculations in columns, tables, measures
* Based on Excel
* Used on other Microsoft tools like Power Pivot and Analysis Services

### DAX functions

* Predefined formulas – arguments
* Specific syntax – order of arguments
* Over 200 different functions in different categories

### Calculated Columns

* Expands existing dataset – without editing source data
* Evaluates at row level
* Adds new column to existing table
* Calculated at data load – when data is refreshed

### Calculated Measures

* Enabled complex calculations
* Aggregates multiple rows
* Adds new field that can be added to visualizations
* Calculated at query time – as interact and filter – not run every time the table is accessed
* Write a DAX formula or use Quick Measure tool

### Creating Calculated Columns and Measures

* Report or Data view
* Select correct table
* New Column – Input formula
* **Create a table as placeholder for measures – Home – Enter Data – Measures - \_Calculations**
* New Measure - Input formula

## Context in DAX Formulas

### Introduction to Context

* Enables dynamic analysis - results of a formula change to reflect the selected data
* There are three types of contexts - row, filter and query

### Introduction to Row Context

* Row context - "the current row" – e.g. Calculated column
* Calculated column - context includes values from all columns within current row

### Row context can also be used in measures

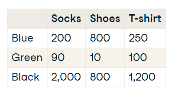
* **ONLY when using iterator functions**
* Loops through every row in a table and runs the formula for each row
* Iterator function - identified by “X” at the end of function name – e.g. SUMX
* Syntax: SUMX(<table>, <expression>)
* E.g. SUMX (Sales, Sales[Price] + (Sales[Price] \* Sales[Tax]) – Iterates over each row and performs the measure calculation without creating a Calculated Column
* E.g. Orders Above Target Profit margin = SUMX(Orders,IF(Orders[Profit\_Margin]>Orders[Target Profit Margin],1,0)) – iterates over each row and count number of orders with profit margin above target profit margin

### Introduction to Filter Context

* Filter context - set of filters that have been applied before the calculation is carried out
* Applied in several ways, including attributes in a row or column, via a slider, through the filter pane, or in a calculated measure

|  |  |
| --- | --- |
| **Colour** | **Quantity** |
| Blue | 1,250 |
| Green | 200 |
| Black | 4,000 |

* E.g. Table of data - apply a filter of blue
* For Power BI to return a value for colour blue, it will go to the base table and apply a filter where colour is equal to blue.
* Can extend this further by creating a matrix of colour versus product category



* This changes the filter context again, because for each data point that is returned Power BI is applying filter context
* For Blue Socks - a filter is being applied for colour being equal to Blue and product category being equal to Socks

### CALCULATE Function

* Calculate allows to evaluate an expression with one or more filter contexts
* Takes two arguments - one argument is required, and the other argument is optional
* **SYNTAX: CALCULATE(<expression> [, <filter> [, <filter2> [, …]]])**
* First argument contains the expression to be evaluated - this must return a single value
* Second argument for adding filters is optional - but need to ensure that filters evaluate as a table, and do not clash with one another
* E.g. Sales City is equal to London and Sales Country is not equal to United Kingdom
* Filters inside calculate function will always override any filters from visualization
* Example: Total sales in EMEA region - CALCULATE (SUM(Sales), Sales[Region] = “EMEA”)

### Variables

* Simplify code
* Increase visibility and performance
* Create and reuse measure inside Calculate

## Working with Dates

### Working with dates

* Example date: 2020/09/30 12:52
* **Date and Time Functions**
* YEAR (<date>) = 2020
* QUARTER (<date>) = 3
* MONTH (<date>) = 9
* **Format Functions**
* Weekday: FORMAT (<date>, <”dddd”>) = Friday
* Time: FORMAT (<date>, <”h:nn:ss”>) = “12:52:00”
* **Time Intelligence Functions**
* LASTDATE()
* DATESBETWEEN ()
* DATEADD ()
* Evaluate data in time-series
* **Time intelligence from only transactional tables – wrong results for gaps**

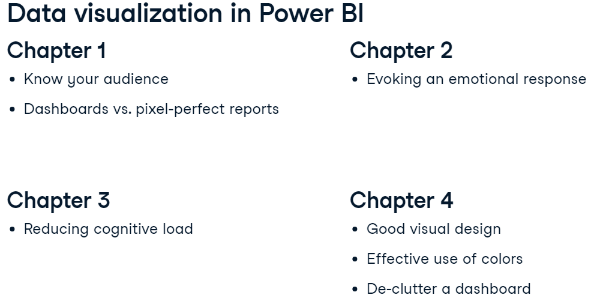
### Create dedicated Standalone Date Table

* For accurate reporting with time-series and time-intelligence functions
* Filter multiple date attributes
* Custom calendar views
* Use time-intelligence functions
* Two ways to create Date table in DAX
* **CALENDAR ()**
* SYNTAX: CALENDAR (<start\_date>, <end\_date>)
* Dates = CALENDAR (MIN (Sales[OrderDate]), MAX(Sales[OrderDate]))
* Returns table with single continuous date column inclusive of start and end dates
* **CALENDARAUTO ()**
* SYNTAX: CALENDARAUTO (<fiscasl\_year\_end\_month>)
* Returns table with single continuous date column inclusive of earliest and latest dates in model
* E.g. CALENDARAUTO (12)
* fiscal\_year\_end\_month needs to be specified for the last fiscal month you after the last date
* E.g. If last date was '2020-07-27' and we specified 12, we'd see dates until the end of the year

### Date and Quick Measures

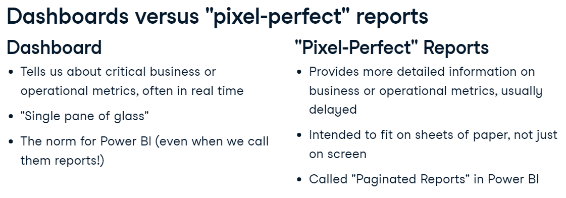
* For accurate reporting with time-series and time-intelligence functions
* Create new table – Create relationship
* Create new calculated column
* Quick Measures – create complex DAX functions by drag and drop
* E.g. Rolling Average –long term trend
* **3-year period = Periods after = 3**

# DATA VISUALIZATION IN POWER BI



## The Audience is King

### Dashboards vs Pixel-Perfect Reports

* Power BI offers two methods of showing data to users: **Dashboards | Pixel-Perfect Reports**
* 

### Intended audience

## Getting an Emotional Response

### Getting an emotional response

* Users are happy because everything looks fine
* Happy users don't always need to act, which makes them even happier
* Users are angry because something has gone wrong
* Angry users are angry in part because they now need to do something.
* Worst-case scenario is if a person has zero emotional response - that's a sign that the dashboard is not relevant to the user

### Bar and Column Charts

* Currently, three types of column charts are available in Power BI
* **Clustered bar charts - easy to show variety of data over single period of time**
* Include appropriate titles and axis values
* Use Format (Painter) options
* Use Analytics menu lines
* **Stacked bar chart – easy to see aggregates**
* **100% stacked bar chart – easy to compare proportions**
* Use legends to break out categories
* Use page level filters
* Use small multiples fields
* Use show as table (right click) to clarify more data
* Use Small Multiples field to segment and analyse
* **Tip – Axis Value – Display units**
* **Tip – show Constant Line name - Data label – Style – Name**

### Small multiples

* Show comparison information across two categorical dimensions
* Saves space
* No need for individual charts

## Reduce Cognitive Load

### Reducing cognitive load

* Tailor page to audience
* Focus one story per page
* Balance between information-rich visuals vs confusing audience
* Use custom visuals

### Line and Area charts

* Line charts – Easy to track historical changes in small number of features

### Combination charts

* Combination chart - Easy to compare a rate variable and a counting variable over time

### Tornado Chart

* Tornado chart – easy to compare two values for each category

## Less is More

### 3 Keys to good visual design

* Does this contribute to the story?
* Is this the right visual element?
* Is it necessary?
* Less control as user can use filters and slicers
* Static text is not useful
* Colour- ‘Pre-attentive attribute’ emphasis
* Negative space (without any information or visual effect) – easy to differentiate sections

### Shares- Pie charts | Tree maps

* Pie chart – easy to show simple shares of static total
* Tree map – easy to show shares hierarchical categorical data
* Tree map - Use data labels
* Tree map - Add more categories to category to create hierarchy
* Navigate in - Drill Down and Drill Mode (form of cross filtering)
* Two down arrows - move down to the next level of the hierarchy
* Split arrows down - expand down to the next level of the hierarchy
* Single down arrow - enable Drill Mode – selecting a level will drill down and expand all elements in that level

### Gauge

* Gauge – easy to show current value with target value and maximum value
* Create Measures in DAX for target and maximum values

### Card

* Card - easy to show single value – clean – important metrics

### Key Performance Indicators (KPIs)

* Track performance over time vs expectations
* KPI visual
* Set target as measure
* Format - Set trend axis direction E.g. Low is good

### Conditional formatting

* Conditional formatting – easy to display colour when certain conditions are met
* Useful when a condition is particularly good or bad – colour draws the eye
* E.g. change font colour in table for any values more than xxx
* Table - Cell elements - choose column - enable for font colour - conditional formatting button- Add Rule
* Visual – Format – Colours – fx
* Use Max, Min

# CASE STUDY: ANALYZING CUSTOMER CHURN IN POWER BI

## Exploratory Analysis

### Data analysis flow in Power BI

* Five different steps in the data analysis flow

1. Data check - to make sure the data makes sense and is ready to work with
   1. E.g. check for duplicate values or missing values
   2. sense check with other internal data sources.
2. Explore Data - ask different questions
   1. E.g. Does an increase in revenue also lead to an increase in profit?
   2. Build your first visualizations in this step.
3. Analyze & Visualize Data - key to choose the right visualization to convey a message
   1. also enables to dig deeper into certain topics to make sure not missed any insights
4. Dashboarding - portray analysis clearly in one or more dashboards
5. Communicate Insights - with stakeholders

### Case Study

* The problem is customer churn.
* Telecom provider Databel.
* Analyze why customers are churning (leaving Databel)
* Analyze a snapshot of the database at a specific moment in time- no time dimension

### Defining churn (Investopedia)

* Churn rate / Rate of attrition / Customer churn
* **Rate at which customers stop doing business with an entity**
* Compare churn with the leaky bucket problem
* Can fill with more water (new customers)
* But overall revenue won't increase if existing customers are leaving
* Easier to retain customers than to attract new customers
* For many companies it's a priority to reduce churn

### Calculating churn

* Simplified formula for churn
* **Divide customers lost by the total number of customers**
* E.g. If total 100 customers in certain period and 10 end up leaving - churn rate = 10%
* Multiple methods to calculate churn
* Makes sense for company to slightly alter formula depending on the industry
* E.g. A traditional e-commerce platform could consider a certain customer a churner if they haven’t made a purchase in last 12 months

### The Databel dataset

* 29 different columns
* 1 row per customer
* Customer\_id - unique ID identifies individual customer
* Churn Label - indicates if customer churned with "Yes" and "No" labels
* Contains various other dimensions – i.e. demographic fields, premium plans
* Measures
  + Total Charges column - sum of all monthly charges billed to a customer
  + Other measures – use Metadata sheet

## Investigating Churn Patterns

### Analyzing the demographics of Databel

* Insights discovered so far
* Average churn rate is around 27%
* Main reason why customers churn is related to competitors
* This could raise questions such as "Is Databel competitive enough?"
* Churn rate in California is abnormally high at 63.24%,
* Don’t have a clear explanation yet for the relatively high churn rate of 27%,
* Still so many columns o analyze

### Metadata sheet

* Holistic analysis plan
* 29 columns in the database
* Good practice to have a structured approach
* Grouped in different categories - create different pages to analyze the different topics

### Advice to Databel

* Discovered that churn rate for customers who pay for an international plan but don't call internationally is sky-high
* Contact customers who are on an international plan but have not called internationally and propose they downgrade their plan.

## Visualizing your Analysis

### Creating a cohesive story

* Build a series of different dashboard-style pages which in combination will tell a story
* Build the pages in such a way to enable easy communication of insights
* Unable to build dashboards in Power BI Desktop - available only in Power BI Service
* Try and create a similar experience for users within Power BI Desktop report

### Structuring analysis

* Not easy to structure an analysis piece in a cohesive way
* It's not informative to publish a bunch of visualizations stored on different pages
* Need to combine the information in such a way that it fits well together
* Ideally, show all discovered insights in a few pages

### Create narrative for stakeholders

* Overview dashboard or page - DataCamp colour scheme - main insights from analysis
* Interactive - revealing even deeper insights for stakeholders
* Contains main KPIs - Number of customers, number of churned customers, churn rate

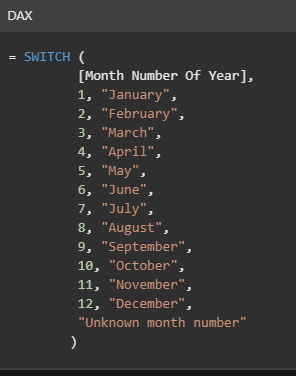
### Interactivity makes a report powerful

* End user clicks on Month-to-Month - churn rate goes up to 46%
* Almost 1600 out of 1800 churners come from this group
* End user selects the Two-Year contract - churn rate drops to 2.78%

### Overview Page

* Collate information for best possible insights into key concerns
* Building dashboard-style pages - makes sense to build new graphs
* **Best Practice – Overview page canvas settings – 1640px width x 960px height**

### SWITCH () DAX Function

* Create new column with DAX SWITCH() to categorize
* Evaluates expression against list of values and returns one of multiple result expressions
* Used to avoid multiple nested IF statements
* SYNTAX: SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])
* E.g.
* 

# DATA PREPARATION IN POWER BI

## Profiling your Data and Introduction to Power Query

### Introduction to Power Query

* Output will only be as good as the input dataset - Garbage in, garbage out
* Make sure data is of high quality and clean
* Won't waste time by working on dirty data.

### What is clean data?

* Free from any missing values – nulls
* No typos or data entry errors
* Remove duplicate entries - prevent skewing statistics
* Focus only on relevant data - remove any data which will not help answer questions
* Outliers should be treated with care - not necessarily removed - can be important
* Apply a floor or ceiling to outliers - not to skew statistics - Cap outliers at a certain value
* Columns should have correct data types – to allow calculations
* Column / table names - short & descriptive

### How does data preparation work in Power Query?

* Power Query - data manipulation framework – to connect to, load and transform data
* Apply series of transformations one after the other
* Sequence is shown on the right of power query interface - applied steps – saved in query

### Data types in Power Query

1. Numbers
2. Date / Time
3. Text
4. Logical Data
5. Binary Data - encoded in base64 - used to store images

* Always make sure to select the correct data type for each column
* Power BI will also usually automatically detect the data type of each column

### Basic structural transformation of data

* Transforming data structure - won't change data stored in tables - only how it is organized
* E.g. promoting headers - make the first row of the data into column headers
* Change order and sort columns as well as remove any redundant columns or rows
* Rename columns to be more descriptive

### Introduction to Power Query demo

* Transform Data option - Home ribbon Power BI desktop
* New Source - Web connector - URL https://www.bankrate.com/retirement/best-and-worst-states-for-retirement/
* Web view - how the website looks on a browser
* Power Query window - see query loaded
* Some applied steps will be automatically generated and applied
  + promoting headers
  + auto detecting the column types
* Click any of steps - preview data as it was right after that transformation was applied
* Useful to understand the effects of applied transformations
* In csv and excel files - the column headers are in the first row
* Make first row into column headers - use promote first row
* Do opposite – use demote column headers
* Remove top rows - enter number of rows
* Rename – double-click column name
* Undo – X on each step
* Or select step – right-click delete until end
* Reposition columns - drag & drop

## Data Preview features in Power Query

### What is data preview

* Quickly summarize key characteristics of dataset
* Find errors and inconsistencies - Identify and fix common errors
* investigate dataset to diagnose what transformations are needed
* Apply transformations step-by-step - see how characteristics change - if transformation had desired effect
* Add filters
* Data Preview - enable in view ribbon
* 3 types of data previews

1. Column Distribution
2. Column Quality
3. Column Profile

* Default - data preview will only analyze data in the first 1000 rows
* Change option - click Column Profiling based on the top 1000 rows box - bottom-left

### How to use data preview?

* Column Distribution
  + Preview distribution of columns – histograms
  + Useful to check duplicate values
  + Shows count of unique and distinct values
* Column Quality
  + Check if data contains any missing values or error values
  + Calculate percentage for each of these and show underneath column names
* Column Profile
  + Combines aspects from previous features
  + Key difference - only works for one column at a time
  + Displays much more detailed statistics i.e. avg, min, max, std
  + Distribution and number of unique and distinct values
  + Only works when a column is selected

## Additional uses for data preview

* Quick access to most common transformations to deal with errors
* Hover over output - tooltip - suggested transformation marked with lightbulb
* Also click on 3 dots to expand a menu containing additional useful transformations
* **It is always better to do replacements in the source file rather than in Power Query - because if multiple rows have the same value, they will all be replaced with the specified value - which will sometimes not be the intent**

## Transformation

* FILL – Up / Down – Takes column and transverses through the values in it to fill any null values in the next rows until it finds a new value
* FILL only useful if sorting by meaningful column in ascending order

## Data Manipulation

### Transforming text in Power Query

* Data entry errors | inconsistencies | typos | incorrect values in them
* Better to fix at source file if possible
* Consistent - should not represent same data point in more than one way E.g., Country column have both USA as well as United States
* Same kind of capitalization for all data points.
* Remove all blank space that appears before and after data
* Remove punctuation and control characters before and after data
* General paradigm - each column should only represent one piece of information - May have to split or combine columns to achieve this

### How to clean text data?

* Transform ribbon - Text section
* Format - different options for capitalization
* Access to Trim & Clean - should generally be applied to every single text column in dataset
* TRIM – removes all trailing and leading whitespace & extra blank space before and after data
* If there is no trailing or leading whitespace, the data is left as is
* CLEAN – removes all control characters such as new lines or carriage returns
* This is less common than whitespace

### Splitting and combining columns

* Specificity of data
* Always split column into two or more that each contain a distinct piece of information
* E.g. Splitting Address field into Building, Street, City, Country columns
* Sometimes two or more columns are more useful when combined
* E.g. Combining First Name and Last Name into a single Name column

### Text Transformation

* Duplicate column
* Extract length
* Trim & Clean
* Merge & extract length
* Split & add prefix

## Numerical transformations in Power Query

### Why should you clean data?

* Rule of thumb: 1-10-100 rule - Costs $1 to verify single row of data - costs $10 to clean row costs $100 if data is left as

### What is clean numerical data?

* Completely free from missing values and errors
* Outliers also affect analysis - requires more advanced tools
* Absolute value transformation – removes all negative values in a column
* Logarithmic transformations - translate exponential relationship into linear one
* Multiply by or add a certain value
* Save memory and make columns more readable by rounding to reduce decimal places (especially recommended if many rows in dataset)

### Date / Time columns

* Special types of numerical columns - standard numerical transformations cannot be applied
* Extract specific properties of a date column like year and month
* Extract derived properties such as the start or end of a particular year
* Extract the age from a date column

### Numerical transformations

* Transform – Standard – Divide (‘000)
* Transform – Scientific – Power | Absolute Value
* Transform - Date (date functions) – text data – Duplicate column first
* Transform – Round
* Multiply by – 1 to have negative values

# DATA MODELLING IN POWER BI

## Defining Tables

### Data modelling and table properties

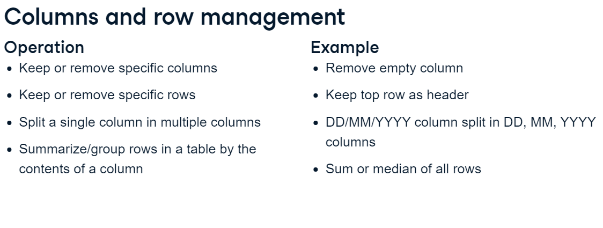
* Data model - conceptual representation of data elements and relationships between them
* In the form of a diagram
* Include tables, columns, and relationships between tables
* Can also include information on data types and keys
* Process of creating a data model - critical part of Power BI development
* Incoming data usually needs clean-up and substantial alterations to find the right insights
* Includes changing columns to join together disparate tables, removing non-data rows, or comprehensive changes like pivoting tables
* Can reduce disk space - improves speed
* Makes the model easier for business users to interpret

### Data modelling in Power BI & Power Query

* Handled by Power Query - query editor for Power BI and other MS products like Excel
* Main purpose - manage queries which drive dashboards and reports, also assist with data modelling
* Power Query directly changes data i.e. removing columns
* Define base changes in Power Query and perform fine-tuning steps in Power BI

### Columns and row management

* Key data modelling steps in Power Query and Power BI – on column and row level



### Data types

* Columns can be assigned to data types in Power Query and Power BI
* Constrain data - ensuring values match expectations
* Optimize storage: numeric value 1,000,000 fits in four bytes but takes nine bytes as a string
* Functionality limited to specific types – i.e. date arithmetic possible on date / time columns
* Power Query usually correctly infers the data type - using the first few hundred rows

### Rounding

* Specify decimal places in Power Query - changes loaded data – not a formatting choice
* Useful for strict rounding rules
* However - round after performing all calculations - rounding in Power Query is uncommon
* Formatting within Power BI changes data's appearance but doesn't affect stored data
* Rounding happens after any calculations - prefer rounding in Power BI than in Power Query

### Load and transform data

* Power Query – Changed loaded data
  + Remove first row - Remove Rows > Remove Top Rows - enter no. of rows
  + Use First Row as Headers – changes data types automatically
  + Change data types - Replacing is default - minimizes data type conversion steps
  + Undo - click X mark next to a step in the list
  + Change case – Transform > Format
  + Add suffix / prefix
  + Perform arithmetic on numerical values - right-click or Scientific in Transform menu
  + Transform using the natural logarithm
  + Double click header to change names of columns
* Power BI – Changes appearance
  + Data view – select column - Column tools - options to alter visual representations
  + Changing the formatting
  + Currency
  + Rounding
  + Data category
  + Change summarization - E.g. Don't summarize
  + Data category for columns E.g. geographic data - State or Province
  + Change sorting
  + **Sort column by another column**
  + Sort Geographic ID for example
  + Hide columns – if no use for end user

## Shaping Tables

### Database normalization

* Set of rules and processes for data modeling
* Similar to organizing or designing a database
* Normalization has a few key goals
* First - remove redundant data - when same information is stored in multiple places
* Reducing redundancy can shrink down file sizes and avoids data inconsistency
* Final goal is to build out a design which reflects the real world as it is
* Including how business entities inter-relate
* Data is separated over several tables - connected through relationships
* Relationships are defined in Power BI.

### Data shaping in Power Query

* Power Query includes several data shaping operations to get closer to normalized data model.
* Four key techniques for shaping data
  + splitting columns
  + extracting values from columns
  + merging together queries
  + appending queries to one another
* Note that all of these are always performed in Power Query
* These are the most common practices, but not the only ones possible.

### Column splitting

* Column splitting breaks one column out into multiple columns based on split criteria
* It could be after a delimiter, such as a comma or tilde
* For serial numbers - split after a number of characters
* E.g. the first 5 characters representing the brand and the next 3, the product

### Column extraction

* Column extraction takes columns from one table and breaks them out into multiple tables
* When extracting a column - define keys for the new tables
* A key is a column or set of columns which make a particular row unique in a table
* On the original table - retain that key column so to create a relationship between the tables
* In Power BI - define these relationships in the Model view
* With column extraction - retain just the distinct values of these breakout tables, which shrinks the total data model size and reduces redundancy

### Query appending

* Append a query to combine the contents of two or more tables into a single table
* It matches rows based on the column names, so watch out for differing headers
* **It is equivalent to a UNION ALL statement in SQL**

### Query merging

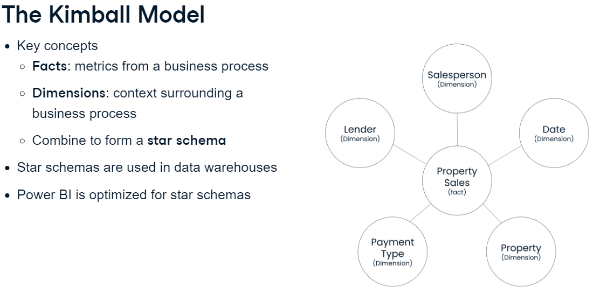
* There are different ways to join tables together using keys - similar to joining tables in SQL

### Merging and appending queries

* Append the second table - select New Source
* Both tables visible in Queries
* Create new table with above tables appended - select Append Queries in Combine section
* Append Queries - replaces the current table with the appended one
* Append Queries as New - creates a new table where the two tables are merged together
* The new table - Append1 - contains data from two tables
* Append as many tables as needed
* Rename Append1 for clarity
* Break out a column into its own table - Duplicate table - Remove all but needed column - Remove Duplicates - the table is now a reference or lookup table
* Extract values from column - different options extract data based on the number of characters, their position, or a specified delimiter
* E.g. keep the last two characters from a column – Transform – Extract - Last Characters

## Dimensional Modelling

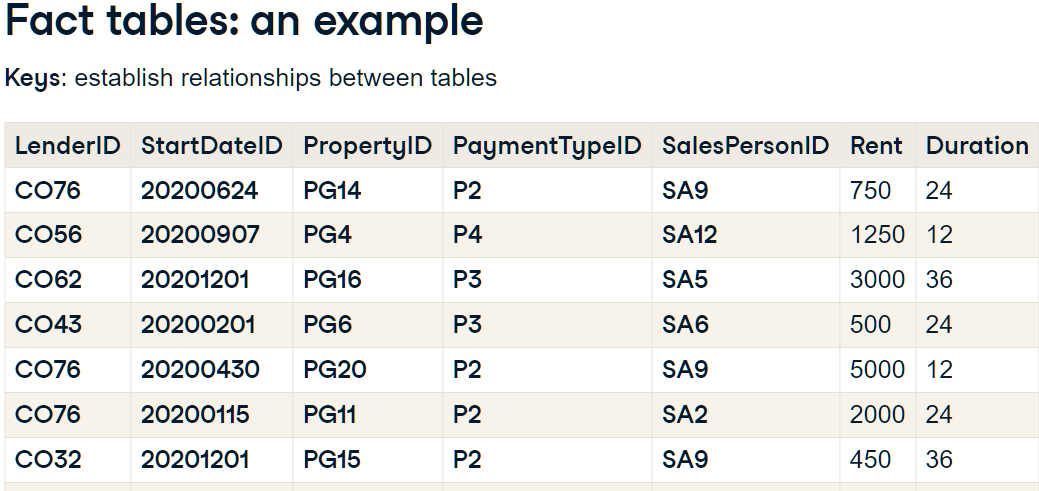
### Dimensional model / Kimball model

* Two key concepts
* Facts - metrics from business process
* Dimensions – provide context surrounding a business process
* Combine to form a star schema - name comes from way facts and dimensions connect
* E.g. Measures related to Property Sales are stored in the fact table and Lender, Salesperson, Date, Property, and Payment Type tables provide more context about each property sale
* 
* Huge amounts of data are organized in star schema in data warehouses
* Power BI is optimized to use star schemas over any other way of loading data
* Power BI is faster and easier to use with a dimensional approach

### Fact tables

* Typically has two types of columns
* Facts - measures or metrics from business process
* E.g. sales, employee count, number of website visits
* Generally dates and numbers which can be aggregated in some way
* Keys - relationships between fact tables and dimension tables are established
* Fact tables - tall and narrow - lots of rows - try to minimize number / size of columns

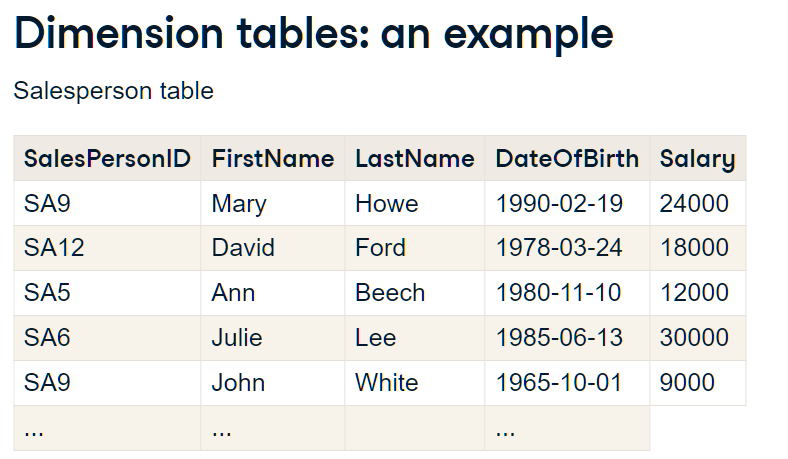
### Fact tables: an example

* Each row represents a property that was rented at a specific date
* First five columns contain keys or IDs that are used to link to each of the dimension tables
* Last two columns hold the measures
* .

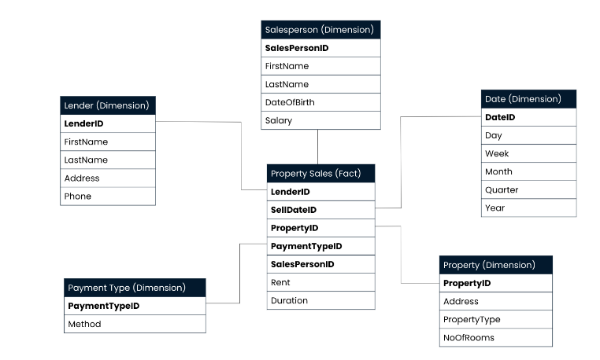
### Dimension tables

* Provide context around facts
* Fact may tell how much or how often - dimensions give the rest of the details
* E.g. Who did it, how they did it, where they did it, and so on
* Dimensions are shared business concepts - usually noun such as Person, Employee, Customer, and Vendor
* Dimensions contain static or slowly-changing data - names, dates of birth, and height
* Short and wide - don't contain many rows - do contain large amount of context for the facts

### Dimension tables: an example

* E.g. Salesperson table
* First column contains same key as fact table - used to combine information from both tables
* The table also stores additional attributes about each salesperson in the company
* 

### Star schema

* Kimball model - dimensions are often used in multiple facts
* E.g. dimensions surrounding Property Sales could provide context to a different fact table
* Dimensions do not link to other dimensions
* 

### Creating a star schema example

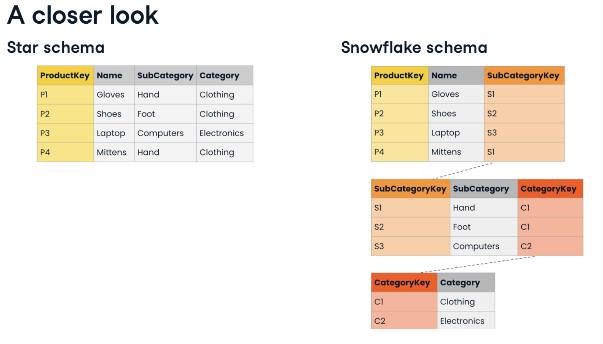
* Take a data file and break it out into a fact and a dimension tables
* Keep keys in fact table
* Remove duplicates in dimension tables
* Model view – connect fact table keys to dimension tables

## Star and Snowflake schemas

### Star schema

* Star schema consists of one or more fact tables surrounded by dimension tables.

### Snowflake schema

* Snowflake schema is like star schema - except it allows relationships between dimensions
* Biggest difference - how they handle hierarchical data
* Star dimensions - all levels of a hierarchy in the same table
* Snowflake dimensions - hierarchy levels are explicitly broken out into multiple tables
* Snowflake Schema - Dimension Tables that join to each other do not join with Fact Table
* Snowflake Schema - Join to the lowest hierarchy
* E.g.
* Product dimension table - each product has name - belongs to a product subcategory - belongs to a product category
* Star schema - all levels of hierarchy and their attributes are on the same product dimension
* Snowflake schema - each level of the hierarchy becomes its own table - joined with keys
* 

### Comparison

* Dimensional modeling theory - strongly prefer star schemas over snowflake schemas
* Key reason - star schemas are easier for business users to understand
* Quite a few business intelligence tools have been optimised for the star schemas
* Snowflake schemas common in some data warehouses
* Key reason - star schemas duplicate data - storage costs - performance impact
* Star schemas are not ideal for frequently updated data - especially with large dimensions
* E.g. Millions of rows containing country name - When country name changes - update large number of rows in star schema vs update a single row in snowflake schema

### Star and snowflakes in Power BI

* Both schemas work
* Can import the data as-is from snowflake schema in existing warehouse
* Power BI prefers star schemas for same reason as dimensional modeling theory
* Easier for users to understand
* Optimizations in Power BI which make performance much less of a concern

### The performance analyzer

* Power BI has built-in performance analyzer - Optimize - Performance Analyzer
* Keeps track of at least three key measures on each visual
* How long to read data from its internal database and perform any DAX operations
* How long for visual to render
* Everything else (waiting time for other operations, including for cross-filtering operations

### Performance tuning advice

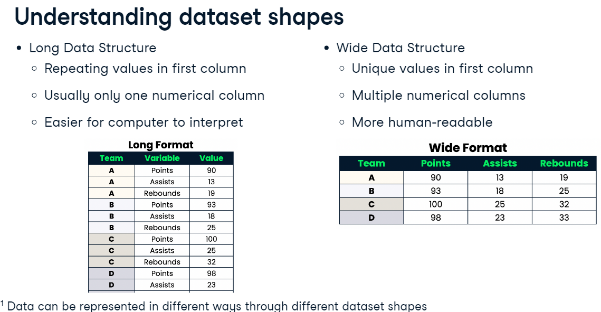
* Number of ways to improve performance
* Tune DAX operations or improve data loading performance
* Could include improving data model
* Use less complicated visuals and show less information on the screen
* Power BI renders each data point - plotting tens of thousands of points may take long
* Reduce number of visuals on the page if Other value is the cause of slowness

# DATA TRANSFORMATION IN POWER BI

## Reshaping Data

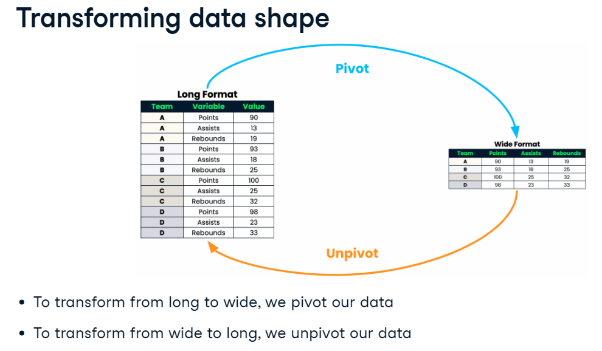
### Understanding dataset shapes

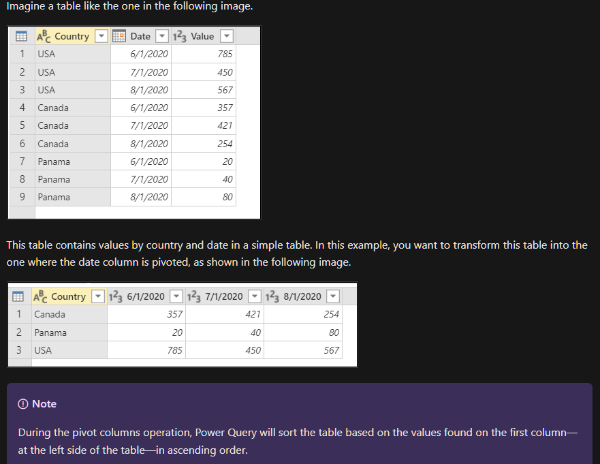
* Two main structures/shapes:
* Long - repeating categorical/id values in first column
* Only one column containing numerical value - easier for computers to interpret and work with
* Wide - contains unique values in first column and multiple numerical columns
* One for each variable
* More human-readable - represents a summarization of the long data structure

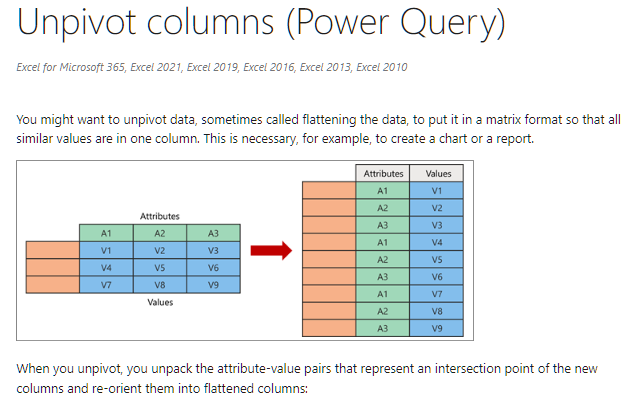


### Transforming data shape

* Pivot - constructs new columns based on values contained in selected column
* Changes structure of dataset from long to wide
* Unpivot - Select one or more columns to flatten
* Transforms selected columns into a category-value pair
* Changes structure of dataset from wide to long

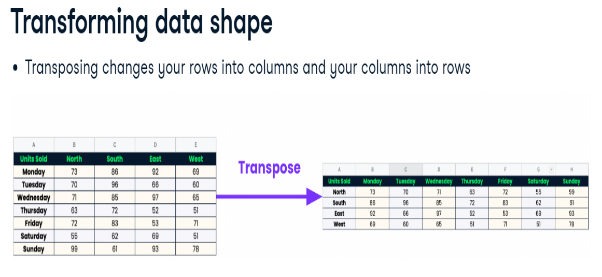




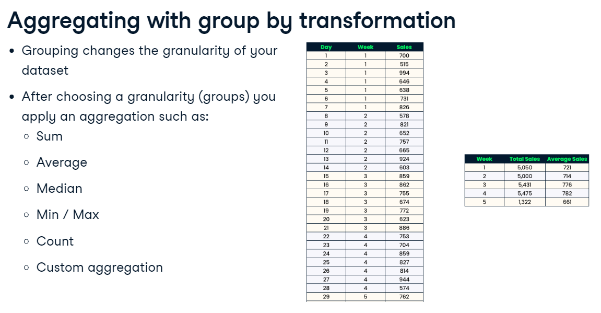


### Transpose table

* Switches rows and columns
* Changes orientation without any aggregations on underlying data



### Aggregating with group by transformation

* Work with summarized version of dataset by changing the granularity
* Define one or more columns that contain desired grouping hierarchy
* Apply an aggregation function on the relevant columns to compute statistics for those groups
* E.g. sum, average, median, minimum, maximum, count, custom aggregation functions
* E.g. interested in total and average weekly sales from dataset containing daily sales
* Select week column to "group" data by
* Aggregate sales by using the sum and average aggregations
* 

## Combining Data in Power Query

### Combining Data in Power Query

* Append query - data that has several tables - there are one or more common columns
* Most often in data models with a star or snowflake schema
* Merge queries - combine tables together vertically to make a "flatter" file structure

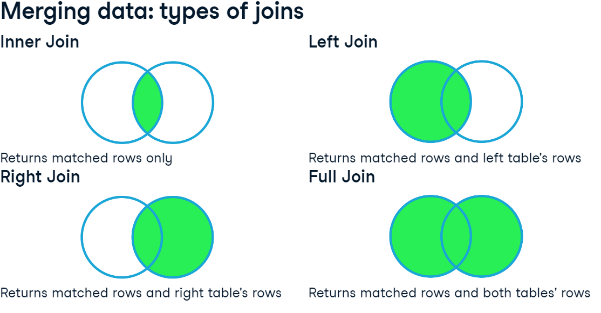
### Appending data

* Add rows to data
* Vertically stack tables
* Column names in tables must be exactly the same
* All tables must have the same number of columns

### Merging data

* Familiar to SQL joins or Vlookups
* Choose one or more columns from one table to compare to the columns in another table
* Link or join tables together horizontally through unique identifier keys

### Merging data: types of joins

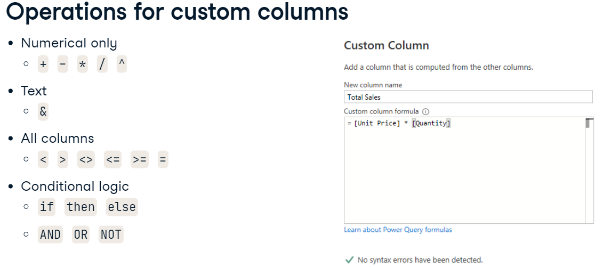
* Inner join - only matched rows are returned
* Left join - returns all matched rows and all other rows in left table that don't have a match
* Right join - returns all matched rows and all other rows in right table that don't have a match
* Full outer join - returns all matched rows and all unmatched rows in both tables
* Second table converted to column – Expand – Choose columns
* **Can merge 2 tables more than once using different keys**
* **Can merge more than 1 column (shift + select columns)**
* 

## Custom Columns

### Custom columns in Power Query

* Modify tables like an Excel spreadsheet
* Modify query with scripting M language (data Mashup)
* Similar to DAX but case sensitive
* E.g. Sum up two columns into one new column
* Extend functionality of conditional column feature
* Logical operands AND, OR, and NOT to check several columns for desired conditions

### Operations for custom columns

* Basic arithmetic operations to combine numerical columns together
* Adding, subtracting, multiplying, dividing, or exponentiating
* Text columns can be concatenated together with the & operation
* Comparative logic operations such as less than and greater than to compare column values together with conditional logic operations such as "and, or, and not"
* Data type mismatches result in errors
* **If any rows have null value - custom column will also contain a null value for that row**
* **Default data type is ANY**
* 

### Group by with custom columns

* Combining Custom column with Group by transformation
* Usually aggregates values in a column using aggregation functions i.e. sum or average
* Using "All Rows" aggregation - apply advanced transformations with custom columns
* "All Rows" aggregation divides data into several smaller tables - according to the grouping
* Alter data in these tables using Table M functions
* E.g. Group sales data into product categories, then ranking each product according to sales performance in its own category

### Adding Custom Columns

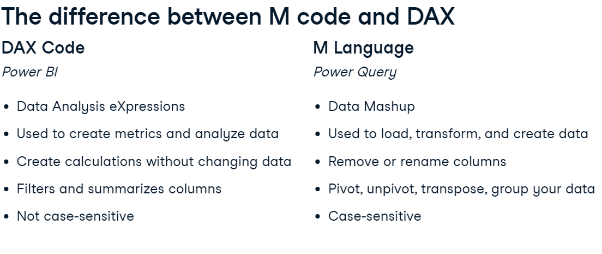
* Ranking - Index Column – Add Column – Dropdown – From 1

## Advanced Editor

### Introduction to Advanced Editor

* M Code - language of Power Query
* See and change source code of query
* Power Query works by translating each transformation step to a line of M code
* Similar to recording macros in Excel and gets translated to Visual Basic for applications
* DAX and M Language are both parts of Power BI but they have important differences
* DAX to create measures and analyze data
* M language to extract, transform, and load data

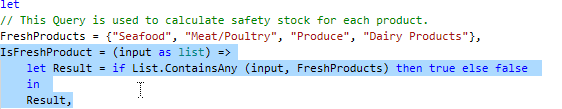
### The difference between M code and DAX

* DAX - Data Analysis eXpressions - used to create metrics such as sums, averages
* These metrics do not actually change data - use data to calculate new values
* Underlying data remains in original state
* DAX functions are not case-sensitive (like Excel)
* Power Query or M Language (data Mashup) - load, edit, create data
* Many transformations can be applied once loaded
* I.e. removing columns or renaming them, pivoting tables, and merging queries
* M code is case-sensitive
* 

### M Language and the applied steps

* M code is a translation of steps applied into a sequential programming language
* Any applied steps are translated to a corresponding line of M code
* Any line of written M code gets translated to an applied step
* 1:1 relationship

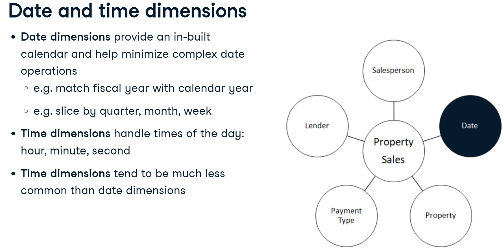
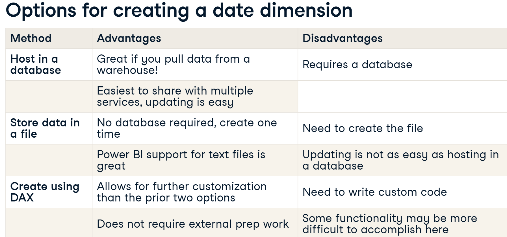
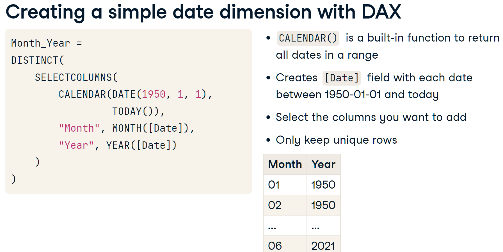
### Writing M language

* Store variables and define functions to be used in the context of query
* Write custom functions
* Add comments to M code – use // double forward slash prefix
* M Language defines several value types for use in data transformations
* Analogous to the data types in Power Query such as Number, Text, Logical, and Date
* More abstract value types such as lists and tables are used by Power Query to group data
* Lists can be used to store important values for functions
* Tables are the most common data structure in M Language
* Tables consist of column names and lists of values that are organized into those columns
* Top-right of Advanced Editor window - dropdown Display Options
* Set of modifications that make it easier to interface with the Advanced editor
* Each step of query will always reference name of previous step in the code
* Represents sequential transformations - ensures steps are executed in correct order
* Always keep referential structure intact
* E.g. Store Variable list: Food = {“a”, “b”, “c”},
* Store Local Function:
* 

# INTERMEDIATE DATA MODELING IN POWER BI

## Date Dimensions and Relationships

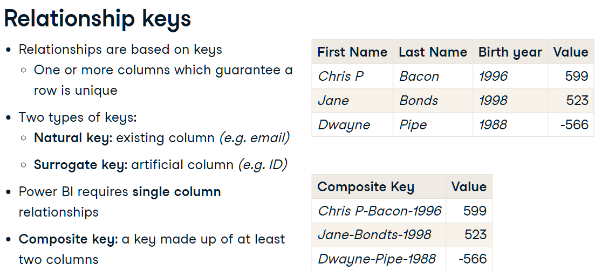
### Date and Time Dimensions

* 
* 
* 
* E.g. DAX Syntax to create “Year” table with single column “Year” from 1961 – today
* Year = DISTINCT(SELECTCOLUMNS(CALENDAR(DATE(1961, 1, 1), TODAY()), "Year", YEAR([DATE])))
* E.g. Extract decade
* Decade = 'Year'[Year] - MOD('Year'[Year], 10)

### Defining Relationships

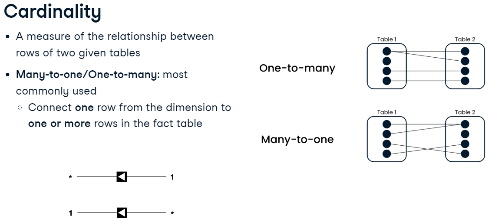
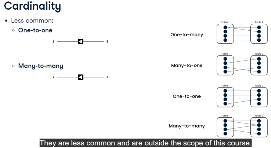
* Date or other dimension tables need to be linked to other tables
* Achieved by creating a relationship
* Easily propagate filters across tables and allow for cross-table calculations
* Power BI automatically guesses relationships based on column names
* Customize relationships manually

### Relationship Keys

* 

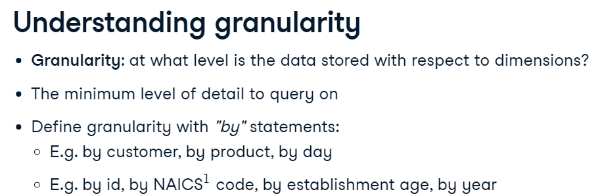
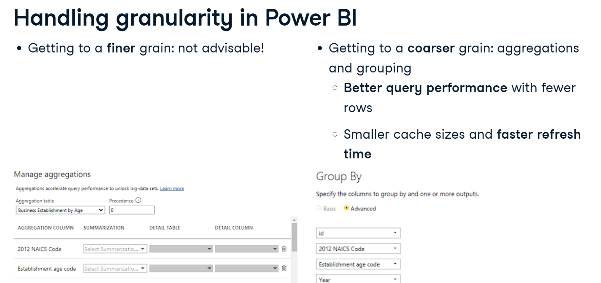
### Composite key relationships

### Cardinality

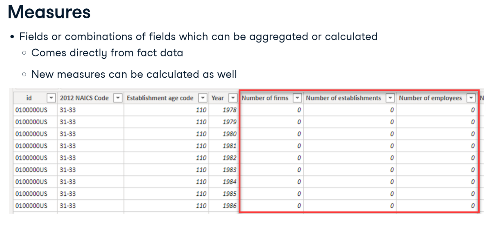
* 
* 

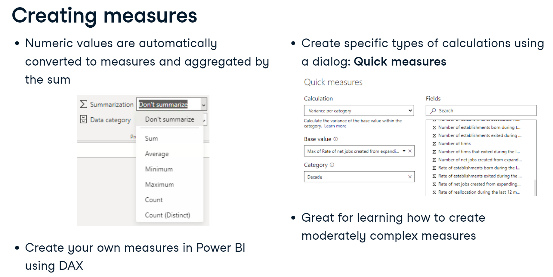
## Granularity, Measures, and Hierarchies

### Granularity

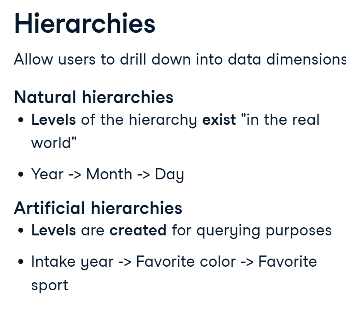
* 
* 

### Measures

* 



### Hierarchies

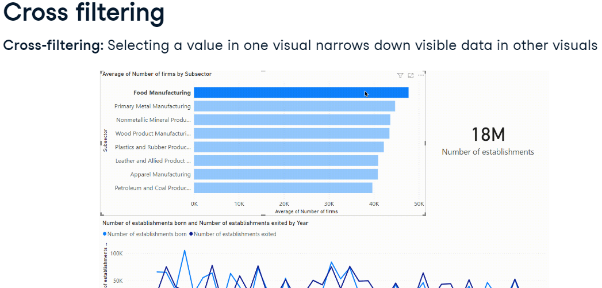
* 

### Hierarchies and measures in Power BI

* Data view - Right click on the dimension - select Create hierarchy
* Can rename the hierarchy by double clicking
* Right click field – Add to hierarchy
* Best practice - hide other columns.
* Changing the granularity of the data by aggregating measures happens in Power Query
* E.g. only need the sum of firms and employees per year - but aw data takes too long to load
* Aggregate these measures in Power Query - speed up the process
* Duplicate Business Establishment by Age - Transform menu - select Group by - click Advanced
* Grouping per year – set on top
* Aggregations are sum of number of firms, and sum of number of employees.
* Creating new measures in Power BI is mostly handled by DAX
* E.g. Create new measure called Employees per Firm. - use DIVIDE to divide sum of employees by sum of firms
* **DIVIDE function has the advantage of not throwing an error when the denominator is zero**
* New calculated measures do not show up in this table - only run when used in visualization
* Quick measures E.g. Calculate a running total of the number of employees, per year
* Select the calculation (Running total), the Base value (Sum of Employees), and the Field (Year)

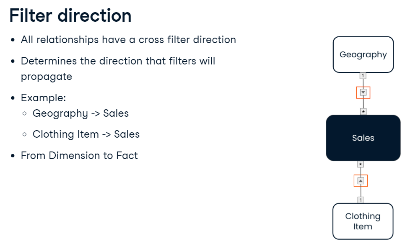
## Advanced Data Modeling

### Cross filtering

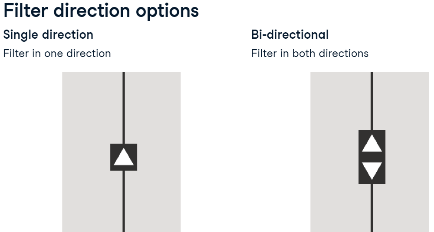
* Selecting data point on one visual changes other visuals based on that selection
* 

### Filter direction

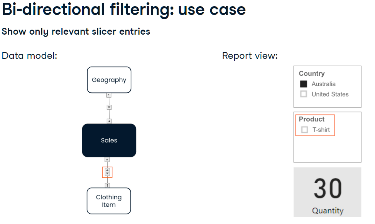
* Each relationship must be defined with a cross filter direction
* This determines the direction that filters will propagate
* E.g. Any column from the Geography dimension can filter the data in the Sales fact table
* The same is true for the Clothing Item dimension
* The filter direction usually goes from the one to the many, or from the dimension to the fact



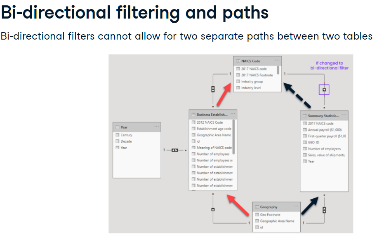
### Filter direction options

* Single direction filters allow to filter in one direction
* Sometimes have to filter in both directions - bi-directional filter
* 

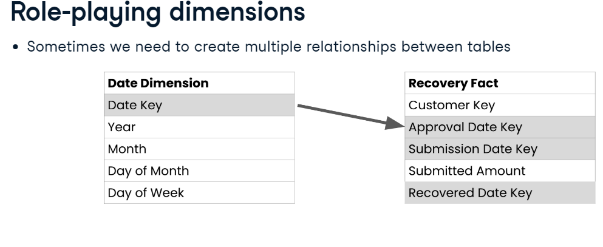
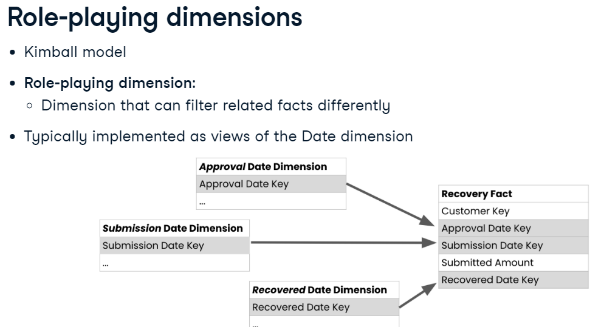
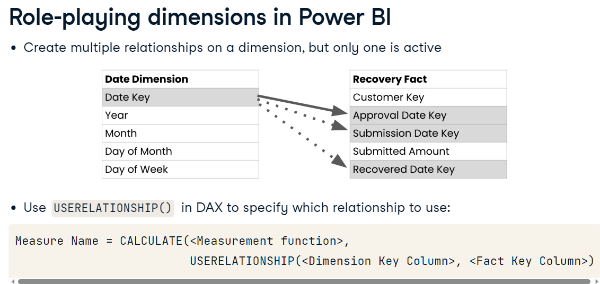
### Bi-directional filtering: use case

* Bi-directional relationships can deliver slicers that limit items to where data exists
* There are two single direction filters leading from the dimensions to the fact
* Add two slicers, one on country from the Geography dimension and one on Product from the Clothing Item dimension
* The Product slicer displays all options when Australia is selected
* However, if we look at the fact table, we can see that only sweaters were sold in Australia
* So it would make more sense if the Product slicer only showed the Sweater option
* To achieve that, we have to add a bi-directional filter between Sales and Clothing Item
* That way there is a direct filter path from Geography to Sales to Clothing Item
* 

### Bi-directional filtering and paths

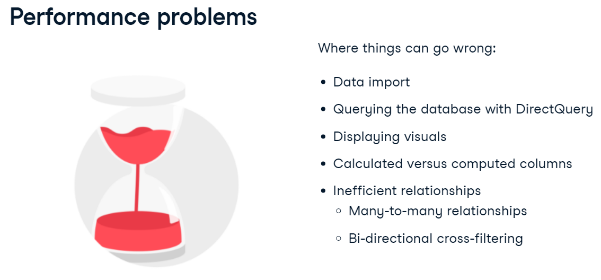
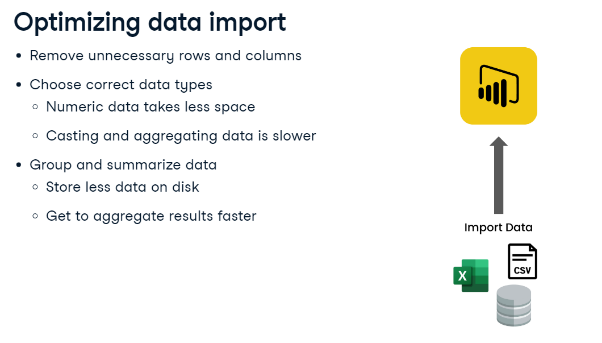
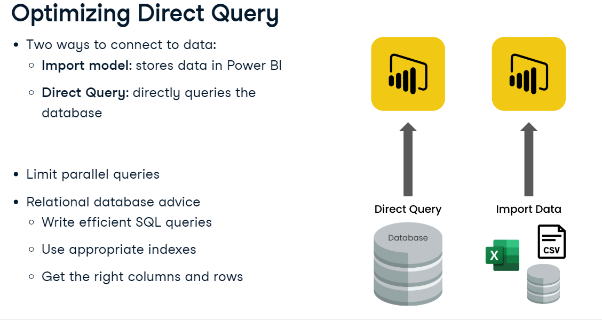
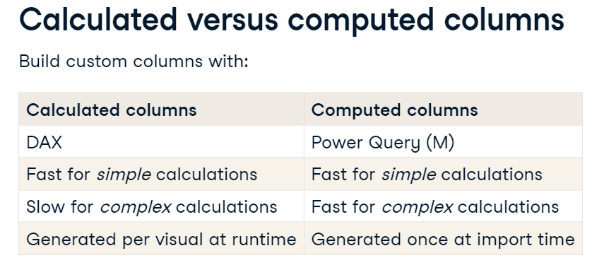
* Bi-directional filters are allowed in Power BI
* Cannot combine them in such a way that they allow two separate paths to let one table filter another table
* **EDIT RELATIONSHIP IN MODEL VIEW**
* **DOTTED LINE – INACTIVE RELATIONSHIP**
* 

### Role-playing dimensions

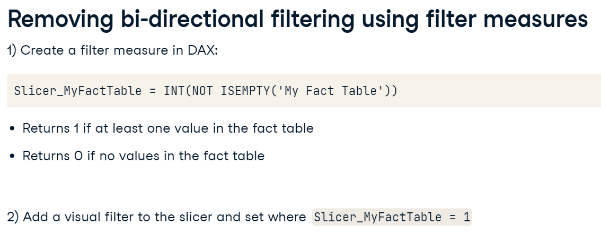
* Sometimes need to create multiple relationships between tables
* E.g. Recovery fact has three separate date keys - can only create single active relationship between Date dimension and the Recovery fact
* Problem - cannot filter by submission month if tie to Date dimension is approval date
* **Classical answer in the Kimball design is to create role-playing dimensions**
* **Table with multiple valid relationships between itself and another table**
* Typically implemented in the database as a view, or copy, of the base dimension
* Connect each view individually to its proper key in the Recovery fact and filter by submission month
* In Power BI this design is imitated by creating multiple relationships between two tables
* First, inactive relationships are created from the Date dimension to the individual dates on the Recovery fact
* Then, create DAX measures that specify which relationship to use
* The **USERELATIONSHIP()** DAX function allows you to specify, just for the given measure, an inactive relationship to use in place of the active relationship
* 
* 
* 

## Identifying Performance Problems

### Performance Problems

* 
* 
* 
* 

### Removing bi-directional filtering using filter measures

* **Slicer\_MyFactTable**
* 

### Displaying visuals

* 