

3.2

Einführung in das Erstellen von Measures mithilfe von DAX

- DAX = Data Analysis Expressions
- DAX wird benutzt, um neue (berechnete) Spalten, Measures und Tabellen zu definieren
- funktioniert ähnlich zu Excel-Formeln
$$\text{Umsatz gesamt} = \text{'Sales'[Quantity]} * \text{'Sales'[Unit Price]}$$

Es gibt drei Möglichkeiten, eine neue Spalte zu erstellen:

- direkt in den Quelldaten, z.B. eine neue View in SQL
- Beim Importieren in PowerBI mittels Power Query Editor (Sprache M):
Spalte aus Beispielen oder benutzerdefinierte Spalte
- eine berechnete Spalte mittels DAX definieren

```
Umsatz gesamt = 'Sales'[Quantity] * 'Sales'[Unit Price]
```

Mit DAX berechnete Spalten sind praktisch, allerdings nicht so ressourcenschonend wie vorher erstellte Spalten.

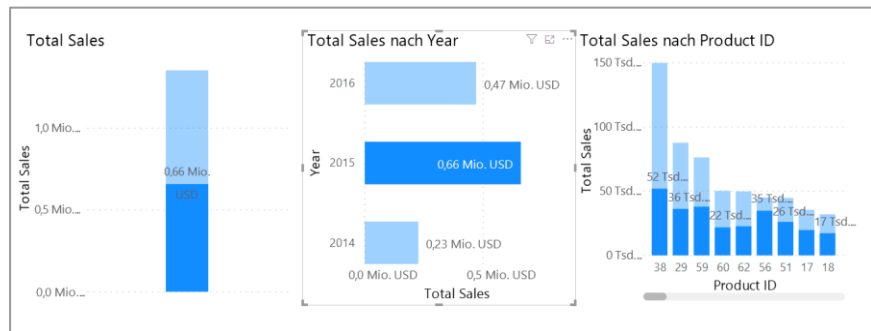
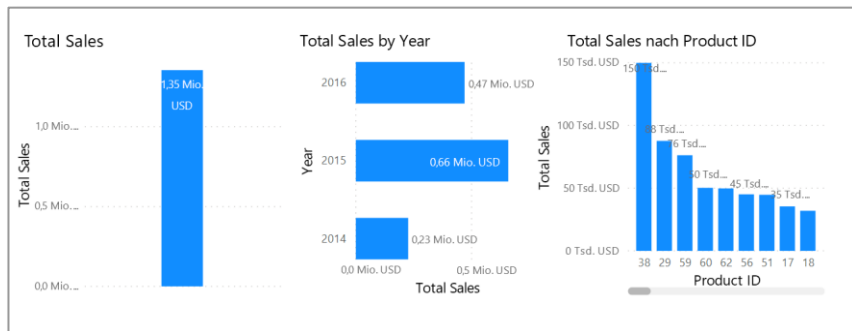
Durch sogenannte X-Funktionen (SUMX, COUNTX) können Measures z.T. auch ohne extra Spalten berechnet werden

Measures sind aggregierte Kennzahlen, z.B. der Gesamtumsatz pro Jahr, Filiale oder Produkt

- Quickmeasures erlauben eine eingeschränkte Auswahl von Aggregationen. Anschließend kann man sich den DAX-Code ansehen.
- **Measures** werden in DAX geschrieben

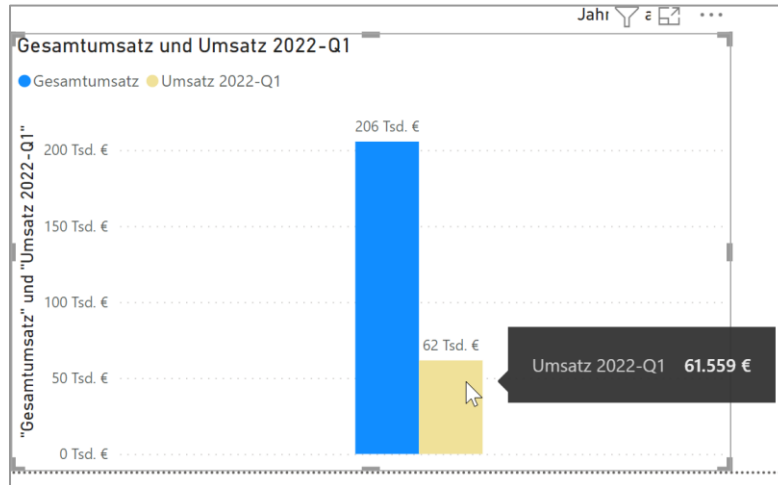
```
Umsatz_gesamt = sum('Sales'[Total Price])
```
- Berechnete Spalten enthalten für jede Zeile einer Tabelle einen Wert. Measures hingegen werden bei Bedarf berechnet, sie berücksichtigen die aktuelle Filterung der Daten

Wie ein Measure dargestellt wird, hängt vom Kontext ab. Die drei Visualisierungen verwenden das gleiche Measure (Total Sales), werden aber unterschiedlich gruppiert. Filterungen haben ebenfalls Einfluss.



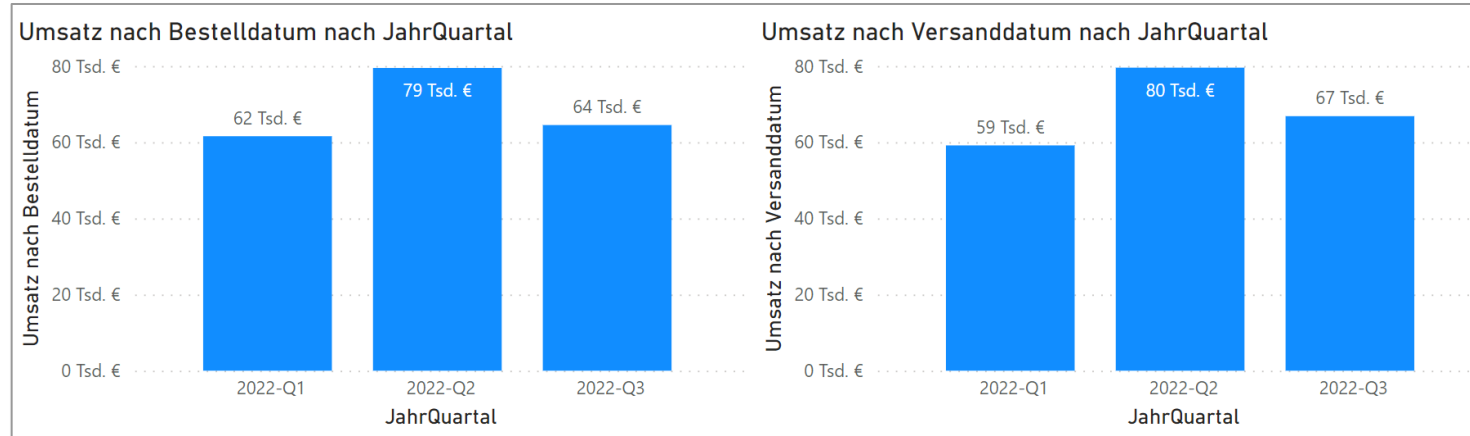
Die DAX-Funktion CALCULATE erlaubt die Einschränkung von Daten, unabhängig von gewählten Filterungen

```
Umsatz 2022-Q1 = calculate(sum('Sales'[Total Price]),  
                           Datum[JahrQuartal]="2022-Q1")
```



Die DAX-Funktion USERELATIONSHIP erlaubt das Verwenden von deaktivierten Beziehungen. Z.B. sind Bestell- und Versanddatum mit der Datumstabelle verbunden, aber nur die Beziehung zum Bestelldatum aktiv.

Umsatz nach Versanddatum = `calculate(sum('Sales'[Total Price]),
userelationship(Sales[Versanddatum], Datum[Datum]))`



Die DAX-Funktion LASTDATE erlaubt das Verwenden des letzten Werts anhand eines Datums, z.B. bei Lagerbeständen ist ein Summieren über mehrere Produkte ggf. sinnvoll, aber nicht über mehrere Tage.

```
Lagerbestand aktuell = calculate(  
    sum('Warehouse'[Lagerbestand]),  
    lastdate(Datum[Datum]))
```


DAX bietet einige Funktionen, die typische Zeit-Berechnungen ermöglichen und somit komplexe Berechnungen einfach machen

- **TOTALYTD** (ytd = year to date, seit Jahresbeginn) erlaubt die Summierung von Jahresbeginn an (analog MTD = month to date, QTD = quarter to date)
`YTD Umsatz = totalytd(sum(Sales[Sales]), Datum[Datum])`
- **PREVIOUSMONTH** erlaubt das Vergleichen mit einem früheren Wert
`Umsatz Vormonat = calculate(sum(Sales[Sales]),
previousmonth(Datum[Datum]))`