

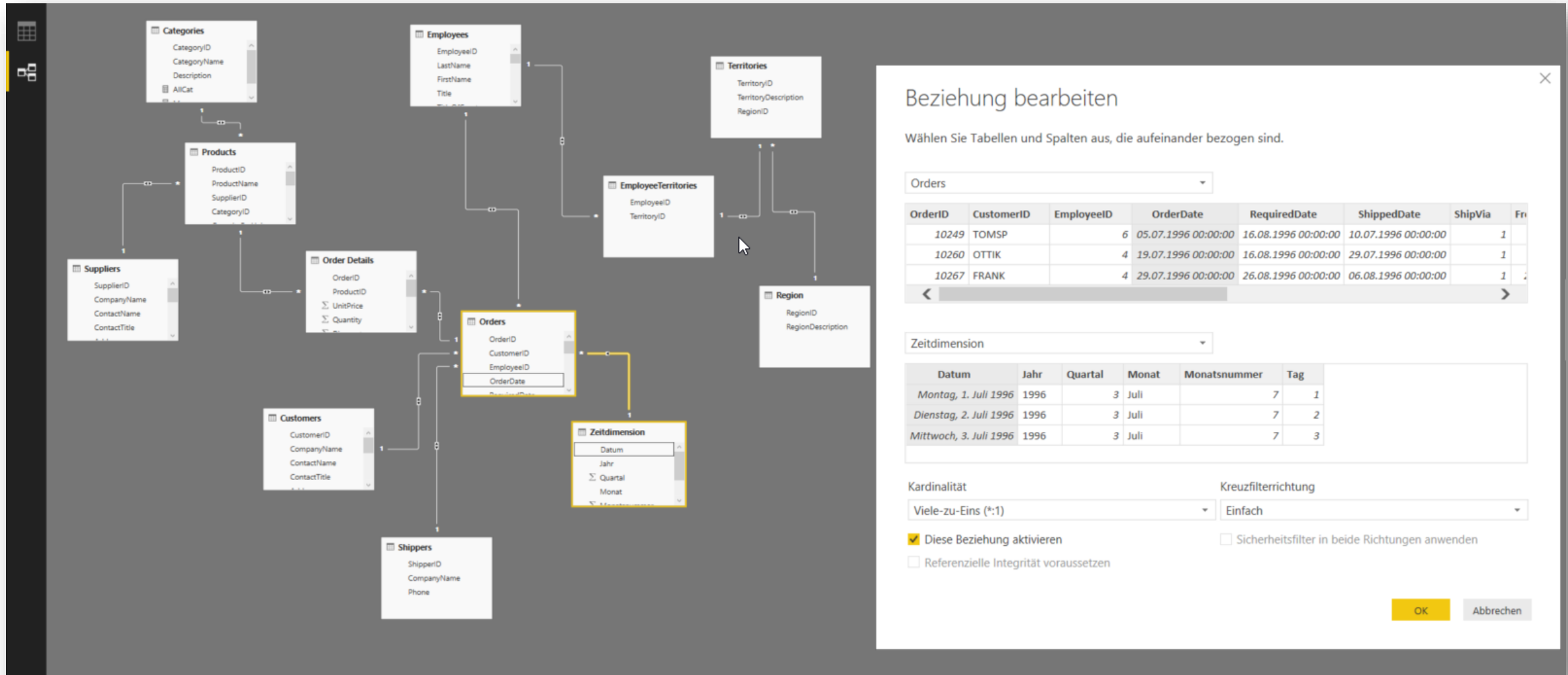
# Lab-Power BI Modul 3

Datenmodellierung, Berechnungen, Measures und Neue Tabellen

# Datenmodell erstellen

- In dieser Übung werden Daten aus zwei unterschiedlichen Datenquellen geladen:
  - SQL Server und Excel
- Um ein Datenmodell zu erzeugen, kann eine Datenquelle oder bei Bedarf auch mehrere unterschiedliche Datenquellen geladen und zusammengeführt werden
- Nachfolgend wird eine SQL-Datenbank mitsamt mehrerer Tabellen als Relationen ins Datenmodell geladen
- Anschließend noch eine Excel-Datei die mit einer SQL Tabelle verbunden wird

# Datenmodellierung



The screenshot displays the Microsoft Access database environment. On the left, a database schema is visible with tables: Categories, Products, Suppliers, Order Details, Customers, Shippers, Employees, Territories, EmployeeTerritories, and Region. The 'Orders' table is highlighted with a yellow border. On the right, the 'Beziehung bearbeiten' (Edit Relationship) dialog box is open, showing the relationship between the 'Orders' table and the 'Zeitdimension' table. The dialog includes a table of data for the 'Orders' table and a table for the 'Zeitdimension' table. Below the tables, there are options for cardinality and cross-filtering.

**Beziehung bearbeiten**

Wählen Sie Tabellen und Spalten aus, die aufeinander bezogen sind.

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight
10249	TOMSP	6	05.07.1996 00:00:00	16.08.1996 00:00:00	10.07.1996 00:00:00	1	5.45
10260	OTTIK	4	19.07.1996 00:00:00	16.08.1996 00:00:00	29.07.1996 00:00:00	1	3.10
10267	FRANK	4	29.07.1996 00:00:00	26.08.1996 00:00:00	06.08.1996 00:00:00	1	4.83

Zeitdimension

Datum	Jahr	Quartal	Monat	Monatsnummer	Tag
Montag, 1. Juli 1996	1996	3	Juli	7	1
Dienstag, 2. Juli 1996	1996	3	Juli	7	2
Mittwoch, 3. Juli 1996	1996	3	Juli	7	3

Kardinalität: Viele-zu-Eins (\*:1)

Kreuzfilterrichtung: Einfach

☒ Diese Beziehung aktivieren

☐ Sicherheitsfilter in beide Richtungen anwenden

☐ Referenzielle Integrität voraussetzen

OK Abbrechen

# Daten mit relationalen Beziehungen

- Nun wird eine SQL Datenbank verbunden und mehrere in Beziehung zueinander stehende Daten geladen
- Daten abrufen -> SQL Server ->
  - Server angeben (z.B. win10-base) ->
  - Datenbank **leer** oder „**Northwind**“ eingeben ->
  - Datenkonnektivitätsmodus „**Importieren**“ wählen
- Zuerst wird die Tabelle „**Order Details**“ markiert -> „**Verknüpfte Tabellen auswählen**“ anklicken -> „**Orders**“ und „**Products**“ werden markiert -> „**Verknüpfte Tabellen auswählen**“ -> „**Categories**“, „**Customers**“, „**Employees**“, „**Shippers**“, „**Suppliers**“
- Dann noch manuell „**Territories**“, „**Region**“ und „**EmployeeTerritories**“ markieren
- Abschließend alle markierten Tabellen per Klick auf „**Laden**“ ins Datenmodell laden

# Beziehungen definieren

Beziehung bearbeiten

Wählen Sie Tabellen und Spalten aus, die aufeinander bezogen sind.

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight
10249	TOMSP	6	05.07.1996 00:00:00	16.08.1996 00:00:00	10.07.1996 00:00:00	1	
10260	OTTIK	4	19.07.1996 00:00:00	16.08.1996 00:00:00	29.07.1996 00:00:00	1	
10267	FRANK	4	29.07.1996 00:00:00	26.08.1996 00:00:00	06.08.1996 00:00:00	1	

Zeitdimension

Datum	Jahr	Quartal	Monat	Monatsnummer	Tag
Montag, 1. Juli 1996	1996	3	Juli	7	1
Dienstag, 2. Juli 1996	1996	3	Juli	7	2
Mittwoch, 3. Juli 1996	1996	3	Juli	7	3

Kardinalität: Viele-zu-Eins (\*:1)

Kreuzfilterrichtung: Einfach

☒ Diese Beziehung aktivieren

☐ Sicherheitsfilter in beide Richtungen anwenden

☐ Referenzielle Integrität voraussetzen

OK Abbrechen

- Tabellen können verschiedene Beziehungen zu anderen Tabellen aufweisen, dabei spielen die „Kardinalität“ und die „Kreuzfilterrichtung“ eine Rolle
- Beziehungen können aktiviert oder wahlweise deaktiviert werden

# Beziehungen verändern

- Die geladenen Daten anzeigen per „**Modell**“ in der linken Leiste anzeigen
- Reiter Start -> „**Beziehungen verwalten**“ -> Kreuzfilterrichtung „**Beide**“ einstellen für alle Tabellen
- **Trainer Beispiel:** werden nicht alle Tabellen mit „Beide“ formatiert, kann es dazu führen, dass kumulierte Auswertungen über mehrere Tabellen hinweg nicht funktionieren werden!

# Measures und berechnete Spalten

- Measures sind Aggregationen auf die Spalten bezogen. Mittels Formel und Funktion können sowohl Measures als auch Berechnete Spalten erzeugt werden.
- Menü links -> „**Daten**“ wählen
- Rechts unter Felder -> Order Details aufklappen
- Im Menü Reiter „**Tabellentools**“ -> „**Neues Measure**“ anklicken
- Measures wie auf nächster und übernächster Folie abgebildet erstellen

# Berechnete Spalten

1 Sales = [UnitPrice]\*[Quantity]\*(1-[Discount])

OrderID	ProductID	UnitPrice	Quantity	Discount	Sales
10508	39	18 €	10	0	180,00 €
10521	35	18 €	3	0	54,00 €
10530	76	18 €	50	0	900,00 €
10546	35	18 €	30	0	540,00 €
10553	35	18 €	6	0	108,00 €
10566	76	18 €	10	0	180,00 €
10569	76	18 €	30	0	540,00 €
10575	76	18 €	10	0	180,00 €
10576	1	18 €	10	0	180,00 €
10577	39	18 €	10	0	180,00 €
10578	35	18 €	20	0	360,00 €
10582	76	18 €	14	0	252,00 €
10587	35	18 €	20	0	360,00 €
10589	35	18 €	4	0	
10590	1	18 €	20	0	
10609	1	18 €	3	0	
10611	1	18 €	6	0	
10612	76	18 €	80	0	

Order Details

- % Abweichung
- Discount
- Durchschnittlich Su...
- OrderID
- ProductID
- Quantity
- Sales
- SalesVJ
- SumSales
- UnitPrice



- Register „Modellierung“ oder „Tabellentools“
- Schaltfläche -> „**Neue Spalte**“
- Spaltenname vorweg  

$$\text{Sales} = [\text{UnitPrice}] * [\text{Quantity}] * (1 - [\text{Discount}])$$
- Das Symbol einer berechneten Spalte sieht so aus:





# Arbeiten mit Measures

1

1 SumSales = sum('Order Details'[Sales])

2

1 SalesVJ = CALCULATE([SumSales];SAMEPERIODLASTYEAR(Zeitdimension[Datum]))

3

1 % Abweichung = [SumSales]/[SalesVJ]

OrderID	ProductID	UnitPrice	Quantity	Discount	Sales
10508	39	18 €	10	0	180,00 €
10521	35	18 €	3	0	54,00 €
10530	76	18 €	50	0	900,00 €
10546	35	18 €	30	0	540,00 €
10553	35	18 €	6	0	108,00 €
10566	76	18 €	10	0	
10569	76	18 €	30	0	
10575	76	18 €	10	0	
10576	1	18 €	10	0	180,00 €
10577	39	18 €	10	0	180,00 €
10578	35	18 €	20	0	360,00 €
10582	76	18 €	14	0	252,00 €
10587	35	18 €	20	0	360,00 €
10589	35	18 €	4	0	72,00 €
10590	1	18 €	20	0	360,00 €
10609	1	18 €	3	0	54,00 €
10611	1	18 €	6	0	108,00 €



FELDER
Suchen
Calendar
Categories
Customers
Employees
EmployeeTerritories
Order Details
% Abweichung
Discount
Durchschnittlich Su...
OrderID
ProductID
Quantity
Sales
SalesVJ
SumSales
UnitPrice



- Register „Modellierung“
  - Schaltfläche -> „Neues Measure“
  - Measure-Name vorweg
1. SumSales = sum('Order Details'[Sales])
  2. SalesVJ = CALCULATE([SumSales]; SAMEPERIODLASTYEAR(Zeitdimension [Datum]))
  3. % Abweichung = [SumSales]/[SalesVJ]

# Tabelle anhand eines DAX-Ausdrucks erstellen

- Die DAX Formelsprache hat den großen Vorteil, dass statt des Imports und Pflege von bspw. Excel-Tabellen, dies wesentlich einfacher und performanter möglich ist, indem eine Tabelle auf Basis eine DAX-Ausdrucks entsteht.
- Auf der nächsten Folie ist dies näher beschrieben, inkl. der zu verwendenden DAX-Fuktion

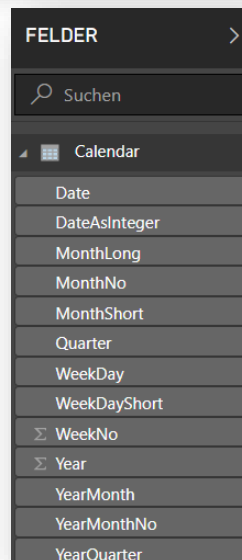
# Neue Tabelle mit DAX erstellen

1



2

```
1 Calendar = ADDCOLUMNS ( CALENDAR ( DATE(2017;1;1); DATE(2018;12;31)); "DateAsInteger"; FORMAT ( [Date]; "YYYYMMDD" ); "Year"; YEAR ( [Date] ); "MonthNo"; FORMAT ( [Date]; "MM" );
"YearMonthNo"; FORMAT ( [Date]; "YYYY/MM" ); "YearMonth"; FORMAT ( [Date]; "YYYY/mmm" ); "MonthShort"; FORMAT ( [Date]; "mmm" ); "MonthLong"; FORMAT ( [Date]; "mmmm" );
"WeekNo"; WEEKDAY ( [Date] ); "WeekDay"; FORMAT ( [Date]; "ddd" ); "WeekDayShort"; FORMAT ( [Date]; "dddd" ); "Quarter"; "Q" & FORMAT ( [Date]; "Q" ); "YearQuarter"; FORMAT (
[Date]; "YYYY" ) & "/Q" & FORMAT ( [Date]; "Q" ) )
```



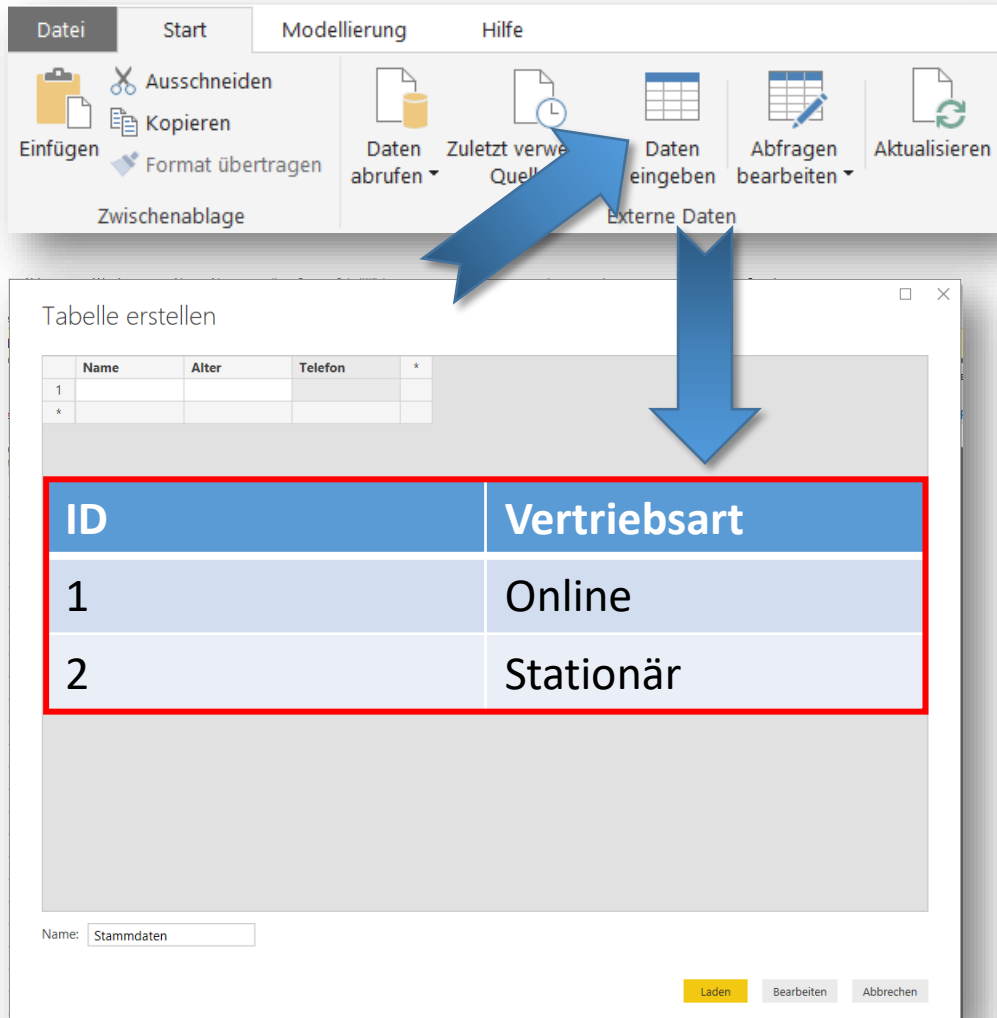
- Register „Tabellentools“
- Schaltfläche -> „Neue Tabelle“
- Tabellen-Name vorweg (Calendar)

```
Calendar = ADDCOLUMNS ( CALENDAR ( DATE(2017;1;1); DATE(2018;12;31)); "DateAsInteger"; FORMAT ( [Date]; "YYYYMMDD" ); "Year";
YEAR ( [Date] ); "MonthNo"; FORMAT ( [Date]; "MM" ); "YearMonthNo"; FORMAT ( [Date]; "YYYY/MM" ); "YearMonth"; FORMAT ( [Date];
"YYYY/mmm" ); "MonthShort"; FORMAT ( [Date]; "mmm" ); "MonthLong"; FORMAT ( [Date]; "mmmm" ); "WeekNo"; WEEKDAY ( [Date] );
"WeekDay"; FORMAT ( [Date]; "ddd" ); "WeekDayShort"; FORMAT ( [Date]; "dddd" ); "Quarter"; "Q" & FORMAT ( [Date]; "Q" );
"YearQuarter"; FORMAT ( [Date]; "YYYY" ) & "/Q" & FORMAT ( [Date]; "Q" ) )
```

# Datenstrukturen in Power BI erstellen

- Daten werden üblicherweise ins Power BI Datenmodell hineingeladen
- Gleichwohl ist es möglich diese ebenso in Power BI manuell einzugeben, was zugegebenermaßen kaum sinnvoll wäre
- Aus einer anderen Datenquelle kopieren und in Power BI hineinkopieren – das könnte schon mehr Sinn machen
- Es kann Sinn machen, wenn z.B. eine kleine Dimensionstabelle mit zwei Spalten und z.B. vier Zeilen benötigt würde, um Daten nach Regionen zu gruppieren.
- Erstelle nach dem auf der nächsten Folie dargestelltem Vorgehen eine kleine Tabelle

# Daten manuell eingeben



- Register „**Start**“
- Schaltfläche -> „**Daten eingeben**“
- Spaltenüberschriften definieren
- Inhalte händisch eingeben oder hineinkopieren, z.B. aus Excel
- Name der Tabelle -> „**Laden**“

Hinweis: sobald die Daten geladen worden sind, ist weiterhin eine nachträgliche Bearbeitung möglich!