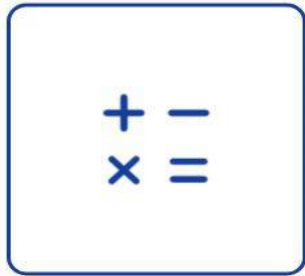


DAX Grundlagen

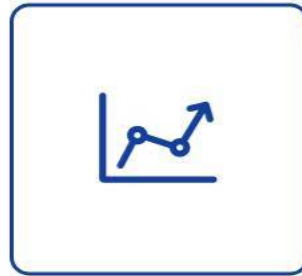
Berechnungen mit DAX durchführen

Einführung in DAX – Was ist DAX?



Formelsprache für Power BI

DAX (Data Analysis Expressions) ist eine spezielle Formelsprache, die in Power BI verwendet wird, um komplexe Berechnungen und Analysen durchzuführen.



Ermöglicht komplexe Analysen

Mit DAX können Benutzer fortgeschrittene Berechnungen, Aggregationen und Filterungen auf Daten in Power BI durchführen, um tiefere Einblicke zu gewinnen.



Steigerung der Datenanalyse-Fähigkeiten

Die Verwendung von DAX in Power BI ermöglicht es Analyseteams, komplexe Fragestellungen zu beantworten und aussagekräftige Erkenntnisse aus ihren Daten zu gewinnen.

Zusammenfassend ist DAX eine leistungsfähige Formelsprache, die Power BI-Anwender in die Lage versetzt, tiefgehende und detaillierte Datenanalysen durchzuführen und so fundierte Entscheidungen auf Basis ihrer Unternehmensdaten zu treffen.

Vorteile von DAX

- Komplexe Berechnungen und Analysen

DAX ermöglicht die Durchführung komplexer mathematischer Berechnungen und datenanalytischer Operationen, wie z.B. fortgeschrittene statistische Analysen oder komplexe Modellierungen.

- Fortgeschrittene Funktionen

DAX unterstützt eine Vielzahl an fortgeschrittenen Funktionen, wie Zeitintervalle, Rollierende Fenster, Hierarchische Beziehungen und leistungsfähige Filtermöglichkeiten, die die Flexibilität und Aussagekraft von Power BI-Berichten deutlich erhöhen.

- Erhöhte Flexibilität und Leistungsfähigkeit

Durch den Einsatz von DAX können Power BI-Berichte deutlich flexibler und leistungsfähiger gestaltet werden, da komplexe Zusammenhänge in den Daten besser abgebildet und analysiert werden können.

Wichtige DAX-Funktionen

- **SUM**
Summiert numerische Werte in einer Spalte oder Tabelle.
- **COUNT**
Zählt die Anzahl der Zeilen in einer Tabelle oder einer Spalte.
- **AVERAGE**
Berechnet den Durchschnittswert der numerischen Werte in einer Spalte oder Tabelle.
- **MAX**
Gibt den höchsten Wert in einer Spalte oder Tabelle zurück.
- **MIN**
Gibt den niedrigsten Wert in einer Spalte oder Tabelle zurück.
- **CALCULATE**
Ermöglicht das Erstellen komplexer Berechnungen, die auf Filterkontext basieren.
- **COUNTROWS**
Zählt die Anzahl der Zeilen in einer Tabelle, auch wenn die Tabelle keine numerischen Werte enthält.
- **RELATED**
Ermöglicht den Zugriff auf Werte aus einer verknüpften Tabelle.

Eine besondere Funktion – CALCULATE()

Was ist die CALCULATE()-Funktion?

Die CALCULATE()-Funktion in Power BI ist ein leistungsfähiges Werkzeug, das es Ihnen ermöglicht, komplexe Berechnungen und Filterungen durchzuführen. Mit dieser Funktion können Sie den Kontext einer Datenabfrage gezielt anpassen und so Ihre Analyse präziser und aussagekräftiger gestalten.



Die Calculate() - Funktion



WAS IST CALCULATE()?

Einführung in die CALCULATE()-Funktion, ihre Verwendung und ihre Bedeutung in Power BI



WIE FUNKTIONIERT CALCULATE()?

Erklärung der Arbeitsweise und Logik hinter der CALCULATE()-Funktion



WANN SOLLTE CALCULATE() VERWENDET WERDEN?

Empfehlungen zur richtigen Anwendung von CALCULATE() in verschiedenen Szenarien



BEISPIELE FÜR CALCULATE()

Praxisbeispiele zur Veranschaulichung der Funktionsweise von CALCULATE()

ZUSAMMENFASSEND BIETET DIE CALCULATE()-FUNKTION IN POWER BI LEISTUNGSSTARKE MÖGLICHKEITEN ZUR KONTEXTBEZOGENEN DATENANALYSE. DURCH DIE RICHTIGEN ANWENDUNG KÖNNEN ENTWICKLER KOMPLEXE BERECHNUNGEN UND ABFRAGEN ERSTELLEN, UM AUSSAGEKRÄFTIGE ERKENNTNISSE AUS IHREN DATEN ZU GEWINNEN.

Wann wird die CALCULATE()-Funktion verwendet?

- **KONTEXTUALISIERUNG VON BERECHNUNGEN**

Die CALCULATE()-Funktion ermöglicht es, Berechnungen in einem spezifischen Kontext durchzuführen, wodurch die Daten präziser und aussagekräftiger dargestellt werden können.

- **FILTERUNG UND GRUPPIERUNG**

Mit CALCULATE() können Daten basierend auf komplexen Filterkriterien gefiltert und gruppiert werden, um detailliertere Einblicke in die Daten zu erhalten.

- **BERECHNUNG VON ABWEICHUNGEN**

Die Funktion ermöglicht es, Differenzen zwischen Werten in unterschiedlichen Kontexten zu berechnen, z.B. um Veränderungen gegenüber dem Vorjahr darzustellen.

- **DYNAMISCHE BERECHNUNG**

CALCULATE() erlaubt es, Berechnungen dynamisch an den aktuellen Kontext anzupassen, was die Flexibilität und Aussagekraft der Analysen erhöht.

- **VERBESSERUNG DER LEISTUNG**

In komplexen Power BI-Modellen kann CALCULATE() die Leistung von Berechnungen optimieren, indem es unnötige Berechnungen vermeidet.

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

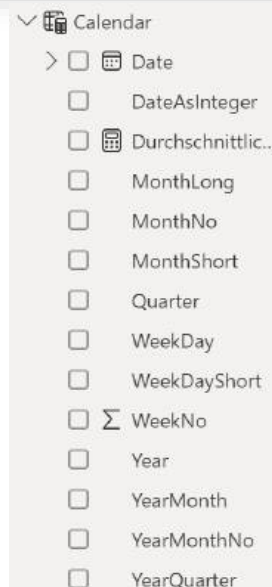
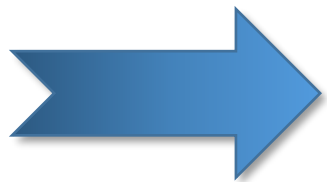
Neue Tabelle mit DAX erstellen (Kalendertabelle)

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]; "mmm" ); "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 „Modellierung“
- 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]; "mmm" ); "WeekNo"; WEEKDAY ( [Date] ); "WeekDay"; FORMAT ( [Date]; "ddd" ); "WeekDayShort"; FORMAT ( [Date]; "dddd" ); "Quarter"; "Q" & FORMAT ( [Date]; "Q" ); "YearQuarter"; FORMAT ( [Date]; "YYYY" ) & "/Q" & FORMAT ( [Date]; "Q" ) )
```