



Alberto Ferrari · Marco Russo

# Datenanalyse mit Microsoft Power BI und Power Pivot für Excel



dpunkt.verlag

# Inhalt

**Cover**

**Titel**

**Impressum**

**Inhaltsverzeichnis**

**Einleitung**

Zielgruppe

Voraussetzungen

Aufbau dieses Buches

Begleitende Inhalte

Schreibweisen und Hinweiskästen

Danksagungen

Errata und Support

**Kapitel 1 Einführung in die Datenmodellierung**

Arbeiten mit einer einzelnen Tabelle

Datenmodelle

Sternschemata

Die Wichtigkeit von Namen

Zusammenfassung

**Kapitel 2 Header/Detail-Tabellen**

Einführung

Werte aus dem Header aggregieren

Header/Detail-Tabellen reduzieren

Zusammenfassung

**Kapitel 3 Mehrere Faktentabellen**

Denormalisierte Faktentabellen

Dimensionsübergreifende Filterung

Mehrdeutigkeit von Modellen

Bestellungen und Rechnungen

- Gesamtrechnungsbetrag für einen Kunden

- Gesamtbetrag der Rechnungen für die ausgewählte Bestellung bzw. den ausgewählten Kunden

- Betrag der in Rechnung gestellten Bestellungen

Zusammenfassung

## **Kapitel 4 Datum und Uhrzeit**

Eine Datumsdimension erstellen

Automatische Zeitdimensionen

- Automatische Zeitgruppierung in Excel

- Automatische Zeitgruppierung in Power BI Desktop

Mehrere Datumsdimensionen

Umgang mit Datum und Uhrzeit

Zeitinformationsberechnungen

Geschäftskalender

Berechnungen mit Arbeitstagen

- Arbeitstage in einer einzigen Region

- Arbeitstage in mehreren Regionen

Besondere Zeiträume im Jahr

- Sich nicht überlappende Zeiträume

- Zeiträume relativ zu heute

- Sich überlappende Zeiträume

Wochenkalender

Zusammenfassung

## **Kapitel 5 Historische Attribute**

Einführung in langsam veränderliche Dimensionen

Langsam veränderliche Dimensionen verwenden

Langsam veränderliche Dimensionen laden

Die Granularität der Dimension korrigieren

Die Granularität der Faktentabelle korrigieren

Schnell veränderliche Dimensionen

Die richtige Modellierungstechnik wählen

Zusammenfassung

## **Kapitel 6 Snapshots**

Einführung

Snapshots aggregieren

Abgeleitete Snapshots

Übergangsmatrizen

Zusammenfassung

## **Kapitel 7 Datums- und Zeitintervalle**

Einführung in Zeitdaten

Aggregationen mit einfachen Intervallen

Datumsübergreifende Intervalle

Schichten und Zeitversatz modellieren

Laufende Ereignisse analysieren

Unterschiedliche Dauern vermischen

Zusammenfassung

## **Kapitel 8 m:n-Beziehungen**

Einführung

Das bidirektionale Muster

Nicht additive Berechnungen

Kaskadierende m:n-Beziehungen

Zeitliche m:n-Beziehungen

Prozentuale Zuordnungsfaktoren

m:n-Beziehungen materialisieren

Die Faktentabelle als Brücke verwenden

Überlegungen zur Leistung

Zusammenfassung

## **Kapitel 9 Unterschiedliche Granularitäten**

Einführung in Granularität

Beziehungen zwischen Tabellen unterschiedlicher Granularität

Vorhersagedaten analysieren

DAX-Code zum Verschieben von Filtern

Filterung über Beziehungen

Werte mit falscher Granularität ausblenden

Werte einer feineren Granularität mithilfe von Zuordnungsfaktoren berechnen

Zusammenfassung

## **Kapitel 10 Segmentierungsmodelle**

Mehrsplätige Beziehungen

Statische Segmentierung

Dynamische Segmentierung

ABC-Analyse

Zusammenfassung

## **Kapitel 11 Währungsumrechnung**

Die möglichen Situationen

Mehrere Quellwährungen, eine Berichtswährung

Eine Quellwährung, mehrere Berichtswährungen

Mehrere Quellwährungen, mehrere Berichtswährungen

Zusammenfassung

## **Anhang Grundlagen der Datenmodellierung**

Tabellen

Datentypen

Beziehungen

Filter und Kreuzfilter

Unterschiedliche Arten von Modellen

    Sternschema

    Schneeflockenschema

    Modelle mit Brückentabellen

Additivität berechneter Felder

    Additive berechnete Felder

    Nicht additive berechnete Felder

    Halbadditive berechnete Felder

**Index**

## KAPITEL 4

# Datum und Uhrzeit

In Geschäftsmodellen werden oft Werte seit Jahresbeginn, Jahresvergleiche und prozentuales Wachstum berechnet. In wissenschaftlichen Modellen kann es vorkommen, dass Voraussagen auf der Grundlage früherer Daten gemacht werden oder die Genauigkeit von Werten im Zeitverlauf untersucht wird. Zeitberechnungen sind in allen solchen Modellen erforderlich, weshalb wir ihnen ein ganzes Kapitel gewidmet haben.

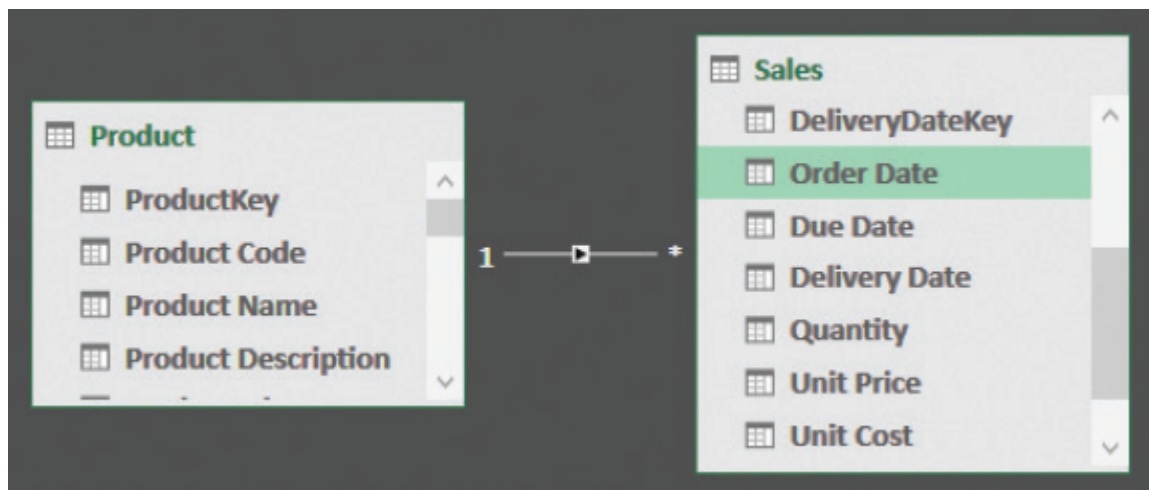
Technisch behandeln wir die Zeit als Dimension, was bedeutet, dass wir gewöhnlich eine Kalendertabelle heranziehen, um Daten nach Jahr, Monat oder Tag zu filtern. Allerdings ist die Zeit eine besondere Dimension, die Sie auf richtige Weise gestalten und bei der Sie besondere Umstände berücksichtigen müssen.

In diesem Kapitel zeigen wir Ihnen verschiedene Situationen und stellen dafür jeweils ein Datenmodell vor. Einige Beispiele sind sehr einfach, während andere ziemlich komplizierten DAX-Code zur Lösung erfordern. Wir wollen Ihnen Beispiele von Datenmodellen zeigen, damit Sie eine bessere Vorstellung davon bekommen, wie Sie Datums- und Zeitangaben richtig modellieren.

## Eine Datumsdimension erstellen

---

Zeit ist eine Dimension. Es reicht nicht aus, in Ihrer Faktentabelle eine einfache Spalte mit dem Datum des Ereignisses vorzusehen. Wenn Sie beispielsweise einen Bericht auf der Grundlage des Modells in Abbildung 4–1 erstellen müssten, würden Sie sehr schnell feststellen, dass das Datum allein nicht genug ist, um einen aussagekräftigen Bericht zu bekommen.



**Abbildung 4-1** Die Tabelle *Sales* enthält die Spalte *Order Date* mit dem Bestelldatum.

Anhand der Datumsspalte in *Sales* können Sie die Werte nach Datum filtern. Für eine Aggregation nach Jahr oder Monat bräuchten Sie jedoch weitere Spalten. Dieses Problem können Sie einfach lösen, indem Sie einen Satz berechneter Spalten direkt in die Faktentabelle aufnehmen. (Allerdings ist dies keine ideale Lösung, da Sie dann die Zeitinformationsfunktionen nicht nutzen können.) Um etwa drei Spalten für Jahr, Monatsname und Monatsnummer zu erstellen, können Sie die folgenden einfachen Formeln verwenden:

```
Sales[Year] = YEAR ( Sales[Order Date] )
```

```
Sales[Month] = FORMAT ( Sales[Order Date], "mmm" )
```

```
Sales[MonthNumber] = MONTH ( Sales[Order Date] )
```

Die Monatsnummern sind nützlich, um die Monatsnamen in die richtige Reihenfolge zu bekommen. Dazu können Sie die Funktion zum Sortieren von Spalten (*Sort by Column*) verwenden, die in Power BI Desktop und im Excel-Datenmodell verfügbar ist. Wie Sie in Abbildung 4-2 sehen, können Sie mit diesen Spalten Berichte erstellen, bei denen die Werte, z. B. die Verkaufsbeträge, nach der Zeit gefiltert sind.

Row Labels	Sales Amount
[-] 2007	1,459,215.95
January	101,097.12
February	108,553.20
March	119,707.83
April	121,085.74
May	123,413.41
June	121,707.44
July	139,381.00
August	87,384.31
September	155,275.94
October	99,872.65
November	122,522.86
December	159,214.45
[+] 2008	1,122,535.05
[+] 2009	1,242,534.61
<b>Grand Total</b>	<b>3,824,285.61</b>

**Abbildung 4-2** Der Bericht nutzt berechnete Spalten in der Faktentabelle, um nach Datum gefilterte Verkäufe zu zeigen.

Es gibt jedoch einige Probleme bei diesem Modell. Wenn Sie beispielsweise auch die Einkäufe nach Datum filtern wollen, müssen Sie die gleichen berechneten Spalten auch in die Tabelle *Purchases* einfügen. Da die Spalten zu den Faktentabellen gehören, können Sie die Jahre in *Sales* nicht zur Filterung von *Purchases* verwenden. Wie Sie in Kapitel 3 gelernt haben, brauchen Sie eine Dimension, um zwei Faktentabellen gleichzeitig zu filtern. Außerdem enthält eine Datumsdimension gewöhnlich viele Spalten, z. B. für Geschäftsjahr und Monate, für Feiertage und Arbeitstage. Es ist von großem Vorteil, all diese Informationen in einer einzigen, leicht zu handhabenden Tabelle zu speichern.

Es gibt noch einen wichtigeren Grund für die Verwendung von Dimensionen: Die Verwendung von Spalten in der Faktentabelle erschwert es, Zeitinformationsberechnungen zu schreiben. Mit Datumsdimensionen werden solche Formeln viel einfacher.

Sehen wir uns das an einem Beispiel an. Nehmen wir an, Sie wollen die aufgelaufenen Verkaufsbeträge seit Jahresbeginn (»year-to-date«, daher die Bezeichnung Sales YTD in dem folgenden Code) berechnen. Wenn Sie dazu Spalten in der Faktentabelle nutzen, wird die Formel ziemlich kompliziert:

```
Sales YTD :=
```

```
VAR CurrentYear = MAX ( Sales[Year] )
```

```
VAR CurrentDate = MAX ( Sales[Order Date] )
```

```
RETURN
```

```
CALCULATE (
```

```
    [Sales Amount],
```

```
    Sales[Order Date] <= CurrentDate,
```

```
    Sales[Year] = CurrentYear,
```

```
    ALL ( Sales[Month] ),
```

```
    ALL ( Sales[MonthNumber] )
```

```
)
```

Im Einzelnen muss der Code Folgendes tun:

1. Einen Datumsfilter anwenden, um nur die Daten vor dem letzten sichtbaren Datum auszuwählen.

2. Einen Jahresfilter anwenden, um nur das letzte sichtbare Jahr zu zeigen, falls im Filterkontext mehrere vorhanden sind.
  3. Jegliche Filter für den Monat in *Sales* entfernen.
  4. Jegliche Filter für die Monatsnummer in *Sales* entfernen.
- 



**Wenn Sie sich mit DAX noch nicht so gut auskennen, sollten Sie herauszufinden versuchen, wie diese Formel funktioniert, um sich besser damit vertraut zu machen, wie Filterkontexte und Variablen zusammenwirken.**

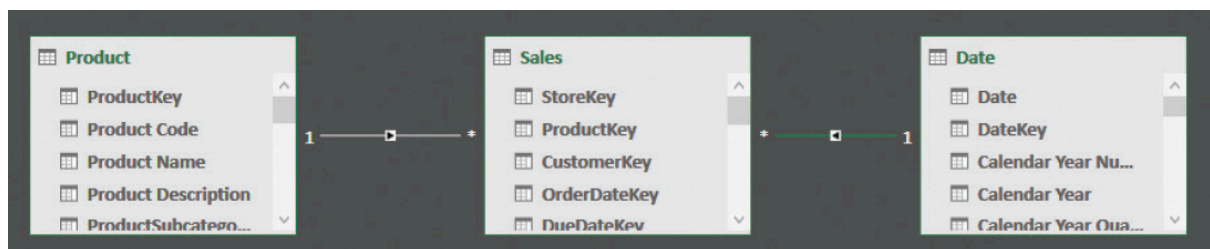
---

Wie Abbildung 4-3 zeigt, funktioniert dieser Code durchaus korrekt – allerdings ist er unnötig kompliziert. Das größte Problem bei dieser Formel besteht darin, dass Sie die integrierten DAX-Funktionen für Zeitinformationsberechnungen nicht nutzen können, denn für diese Funktionen müssen eigene Datumstabellen vorhanden sein.

Row Labels	Sales Amount	Sales YTD
2007	1,459,215.95	1,459,215.95
January	101,097.12	101,097.12
February	108,553.20	209,650.32
March	119,707.83	329,358.16
April	121,085.74	450,443.90
May	123,413.41	573,857.31
June	121,707.44	695,564.75
July	139,381.00	834,945.75
August	87,384.31	922,330.06
September	155,275.94	1,077,606.00
October	99,872.65	1,177,478.64
November	122,522.86	1,300,001.50
December	159,214.45	1,459,215.95
2008	1,122,535.05	1,122,535.05
2009	1,242,534.61	1,242,534.61
<b>Grand Total</b>	<b>3,824,285.61</b>	<b>1,242,534.61</b>

**Abbildung 4-3** Der Bericht über die Verkaufsbeträge seit Jahresbeginn ist korrekt, aber der Code dafür ist zu kompliziert.

Wenn Sie dem Datenmodell wie in Abbildung 4-4 eine Datumsdimension hinzufügen, lässt sich die Formel viel einfacher schreiben.



**Abbildung 4-4** Mit einer Datumsdimension im Modell lässt sich der Code viel einfacher schreiben.

Jetzt können Sie die vordefinierten Zeitinformationsfunktionen nutzen und die Formel damit wie folgt schreiben:

Sales YTD :=

```
CALCULATE (  
  
    [Sales Amount],  
  
    DATESYTD ( 'Date' [Date] )  
  
)
```



**Dies gilt nicht nur für die Berechnung von seit Jahresbeginn aufgelaufenen Beträgen. Alle Zeitinformationsformeln lassen sich leichter schreiben, wenn eine Datumsdimension vorhanden ist.**

---

Durch die Verwendung einer Datumsdimension haben Sie Folgendes erreicht:

Sie haben das Schreiben von berechneten Feldern vereinfacht.

Sie haben alle für Ihre Berichte erforderlichen Spalten mit Zeitinformationen an einer zentralen Stelle gespeichert.

Sie haben die Leistung der Abfragen verbessert.

Sie haben das Modell übersichtlicher gemacht.

Das sind die Vorteile. Gibt es aber auch Nachteile? Nein. Verwenden Sie stets eine Zeitdimension, denn dies bringt nur Vorteile. Gewöhnen Sie es sich an, bei jedem Datenmodell auch eine Kalenderdimension anzulegen, und lassen Sie sich nicht zu der vermeintlich einfacheren Lösung verführen, berechnete Spalten zu verwenden, denn das werden Sie schon sehr bald bereuen.

## **Automatische Zeitdimensionen**

---

Sowohl in Excel 2016 als auch Power BI Desktop hat Microsoft ein automatisches System für den Umgang mit Zeitinformationen eingebaut. Allerdings verwenden

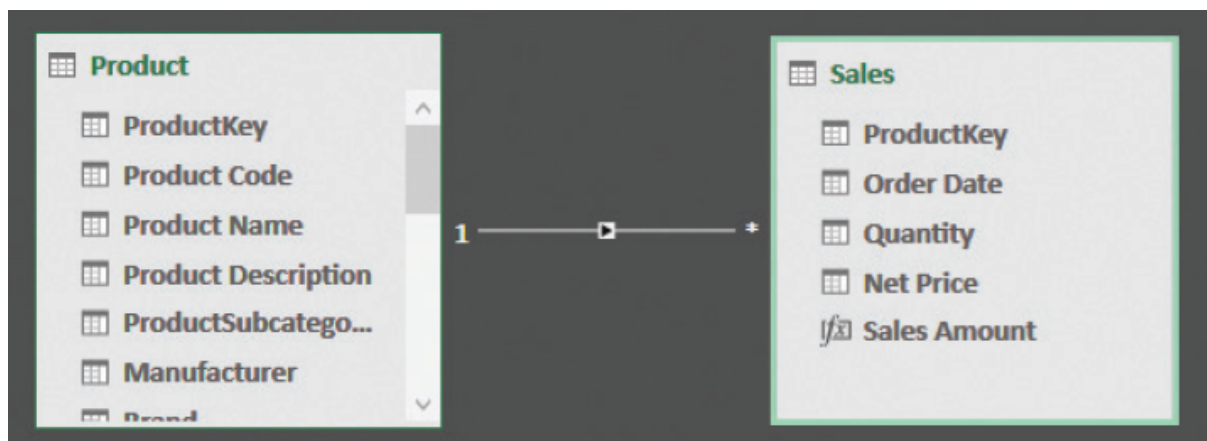
die beiden Programme dafür unterschiedliche Mechanismen. Wir werden uns beide in diesem Abschnitt ansehen.



**Wie Sie in diesem Abschnitt sehen werden, raten wir davon ab, diese Systeme zu verwenden, da sie weder die Flexibilität noch die Benutzerfreundlichkeit bieten, die Sie für Ihre Modelle benötigen.**

## Automatische Zeitgruppierung in Excel

Wenn Sie eine Pivottable für ein Excel-Datenmodell verwenden und ihr eine Datumsspalte hinzufügen, erstellt Excel automatisch eine Reihe von Spalten für automatische Datumsberechnungen in der Tabelle. Nehmen wir an, Sie haben das Modell aus Abbildung 4–5, in dem die Tabelle *Sales* nur eine einzige Datumsspalte hat, nämlich *Order Date*.



**Abbildung 4–5** Die Tabelle *Sales* enthält nur eine einzige Datumsspalte, nämlich *Order Date*, und keine Spalten für Jahr oder Monat.

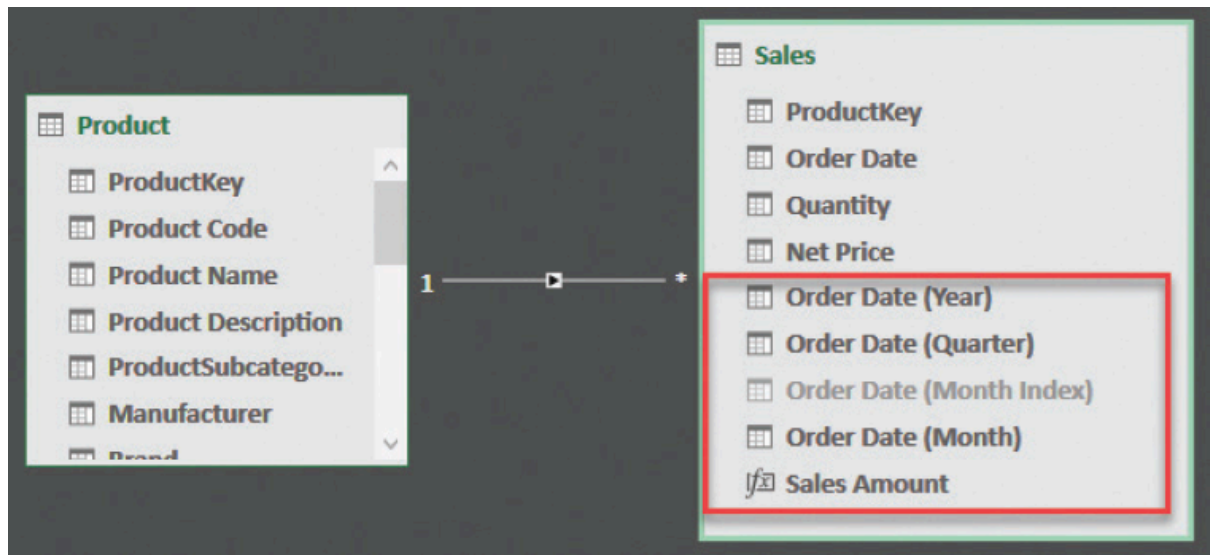
Wenn Sie nun eine Pivottable mit *Sales Amount* im Wertebereich und *Order Date* in den Spalten erstellen, verzögert sich der Vorgang ein wenig. Zu Ihrer Überraschung sehen Sie anschließend nicht etwa eine Pivottable mit *Order Date*, sondern die aus Abbildung 4–6.

Row Labels	Sales Amount
- 2007	1,459,215.95
+ Qtr1	329,358.16
+ Qtr2	366,206.60
+ Qtr3	382,041.25
+ Qtr4	381,609.95
+ 2008	1,122,535.05
+ 2009	1,242,534.61
<b>Grand Total</b>	<b>3,824,285.61</b>

**Abbildung 4-6** Diese Pivottabelle filtert das Datum nach Jahr und Quartal, auch wenn es gar keine entsprechenden Spalten in dem Modell gibt.

Damit diese Pivottabelle einen Datenschnitt nach dem Jahr vornehmen kann, hat Excel automatisch einige Spalten zur Tabelle *Sales* hinzugefügt, was Sie sehen können, wenn Sie das Datenmodell erneut öffnen. In Abbildung 4-7 sind die von Excel hinzugefügten neuen Spalten hervorgehoben.

Excel macht hier genau das, wovon wir Ihnen abgeraten haben: Es erstellt einen Satz von Spalten für die Zeitfilterung direkt in der Tabelle mit der Datumsspalte. Wenn Sie die gleiche Operation an einer anderen Faktentabelle durchführen, erhalten Sie einen weiteren Satz von Spalten. Es ist allerdings nicht möglich, einen dieser beiden Sätze von Spalten zur Filterung für beide Tabellen einzusetzen. Da die Spalten in der Faktentabelle angelegt werden, erfordert dieser Vorgang bei großen Datenmengen viel Zeit und nimmt auch viel Platz in der Excel-Datei ein. Weitere Informationen über diese Funktion finden Sie auf <https://blogs.office.com/2015/10/13/time-grouping-enhancements-in-excel-2016>. Dieser Artikel enthält auch einen Link zu einer Anleitung, mit der Sie die Funktion in der Registrierung ausschalten können. Sofern Sie nicht ausschließlich mit sehr einfachen Modellen arbeiten, empfehlen wir Ihnen, die automatische Zeitgruppierung nach dieser Anleitung auszuschalten und die korrekte manuelle Vorgehensweise zu erlernen, die wir Ihnen in diesem Kapitel zeigen.



**Abbildung 4-7** Die Tabelle *Sales* enthält neue Spalten, die Excel automatisch erstellt hat.

## Automatische Zeitgruppierung in Power BI Desktop

Power BI Desktop versucht, Zeitberechnungen durch Automatisierung einiger Schritte zu vereinfachen. Das funktioniert hier schon ein wenig besser als bei Excel, doch ist Power BI Desktop trotzdem nicht die ideale Lösung für Zeitinformationen.

Wenn Sie in Power BI Desktop ebenfalls das Datenmodell aus Abbildung 4-5 verwenden und eine Matrix mit der Spalte *Order Date* bilden, erhalten Sie die Visualisierung aus Abbildung 4-8.

Year ▼	Quarter	Month	Day	Sales Amount
2009	Qtr 1	January	1	2.198,95
			2	1.325,89
			3	1.775,52
			4	2.167,90
			5	511,70
			6	907,89
			7	332,37
			8	4.605,52
			9	4.442,14
			10	82,70
			11	60,44

**Abbildung 4-8** Die Matrix zeigt Spalten für Jahr, Quartal und Monat an, die gar nicht zu dem Modell gehören.

Ebenso wie Excel baut auch Power BI Desktop eine Kalenderhierarchie auf, allerdings mit einer anderen Technik. In der Tabelle *Sales* werden Sie keine neuen berechneten Spalten finden. Stattdessen legt Power BI Desktop für jede Spalte mit einem Datum eine verborgene Tabelle an und richtet die erforderlichen Beziehungen ein. Wenn Sie nun einen Datenschnitt nach dem Datum durchführen, nutzt das Programm die Hierarchie in der verborgenen Tabelle. Diese Vorgehensweise ist zwar etwas geschickter als die von Excel, weist aber die folgenden Nachteile auf:

Die automatisch generierte Tabelle ist verborgen, weshalb Sie keine Möglichkeit haben, ihren Inhalt zu ändern, also z. B. die Spaltennamen, die Sortierung der Daten oder den Beginn des Geschäftsjahrs.

Power BI Desktop legt eine Tabelle pro Spalte an. Wenn Sie mehrere Faktentabellen haben, werden diese daher mit verschiedenen Datumstabellen verknüpft, weshalb es nicht möglich ist, mehrere Tabellen mithilfe eines einzigen Kalenders zu filtern.

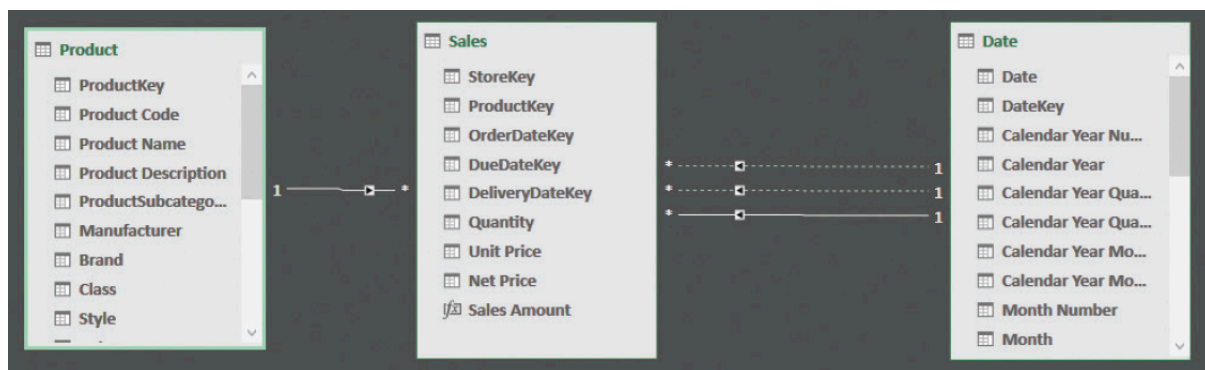
Mit der Zeit haben wir uns angewöhnt, die automatische Kalendergenerierung in Power BI Desktop auszuschalten. (Klicken Sie dazu auf die Registerkarte *Datei* und dann auf *Optionen und Einstellungen*, um das Dialogfeld *Optionen* zu öffnen. Wählen Sie die Seite *Daten laden* und deaktivieren Sie das Kontrollkästchen *Autom. Datum/Uhrzeit*.) Wir sind immer bereit, eine eigene Kalendertabelle einzurichten, die voll und ganz unserer Kontrolle unterliegt, um damit alle

Faktentabellen im Modell zu filtern. Diese Vorgehensweise empfehlen wir Ihnen ebenfalls.

## Mehrere Datumsdimensionen

Eine Faktentabelle kann mehrere Datumsangaben enthalten. Das kommt sogar recht häufig vor. Beispielsweise gibt es in der Contoso-Datenbank zu jeder Bestellung drei verschiedene Datumsangaben: Bestelldatum (*OrderDate*), Fälligkeitsdatum (*DueDate*) und Lieferdatum (*DeliveryDate*). Außerdem können auch mehrere Faktentabellen Datumsangaben enthalten. Daher ist die Anzahl der Datumsangaben in einem Datenmodell oft ziemlich hoch. Was ist die richtige Möglichkeit, um ein solches Modell zu gestalten? Die Antwort darauf lautet schlicht: Abgesehen von einigen Ausnahmefällen sollte es im ganzen Modell nur eine einzige Datumsdimension geben. In diesem Abschnitt wollen wir den Grund dafür erklären.

Wie bereits erwähnt, gibt es in *Sales* drei Datumsangaben, die Sie verwenden können, um *Sales* mit *Date* zu verknüpfen. Sie könnten versucht sein, für die drei Spaltenpaare drei Beziehungen zwischen den Tabellen einzurichten, doch dabei wird nur die erste Beziehung aktiviert, während die beiden anderen inaktiv bleiben (siehe Abbildung 4–9).



**Abbildung 4–9** Von den drei Beziehungen zwischen *Sales* und *Date* ist nur eine aktiv (durchgezogene Linie). Die beiden anderen sind inaktiv (gestrichelt).

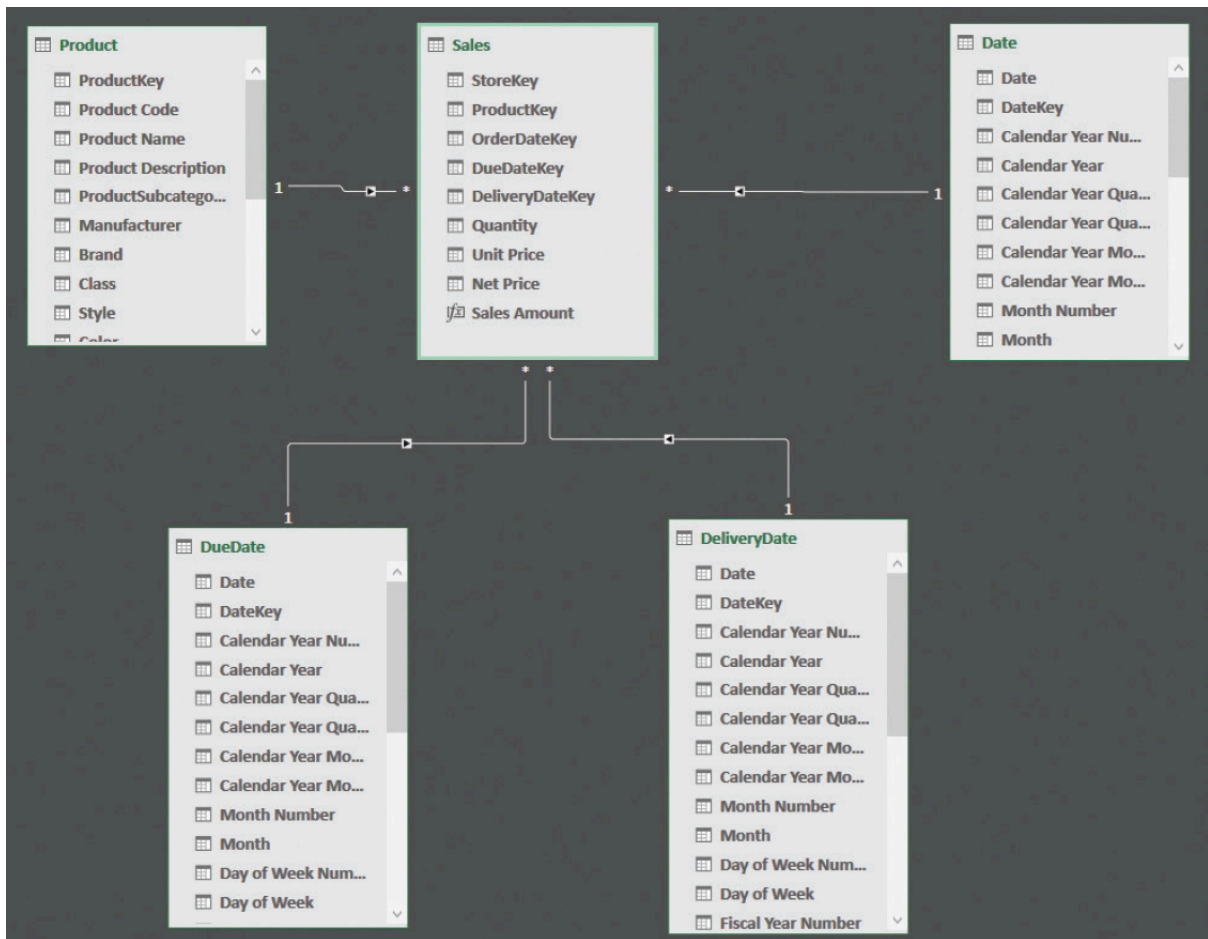
Mit der Funktion `USERELATIONSHIP` können Sie inaktive Beziehungen vorübergehend aktivieren, aber das ist eine Technik, die wir für einige besondere Formeln reservieren. Wenn Sie dieses Datenmodell für eine Pivottabelle oder einen Report verwenden, werden die inaktiven Beziehungen für keine Ihrer Formeln genutzt. Ein Benutzer kann Excel nicht anweisen, eine bestimmte Beziehung etwa in einer Pivottabelle zu aktivieren.



**Es ist nicht möglich, alle Beziehungen zu aktivieren, da die Engine keine mehrdeutigen Datenmodelle zulässt. Solche Mehrdeutigkeiten treten auf, wenn es mehrere Pfade von einer Tabelle (hier *Sales*) zu einer anderen (hier *Date*) gibt. Nehmen wir an, Sie wollten in *Sales* eine berechnete Spalte mit der Formel `RELATED ( Date[Calendar Year] )` aufnehmen. Wären dann alle drei Beziehungen aktiv, könnte DAX nicht erkennen, welche davon verwendet werden soll. Daher kann immer nur eine Beziehung aktiv sein. Dadurch wird das Verhalten von `RELATED`, `RELATEDTABLE` und der automatischen Weiterleitung des Filterkontexts bestimmt.**

---

Die Verwendung inaktiver Beziehungen ist nicht die richtige Vorgehensweise. Daher versuchen wir, das Modell zu ändern, indem wir die Dimension duplizieren. In unserem Beispiel laden wir dazu die Tabelle *Date* drei Mal: einmal für das Bestelldatum, einmal für das Fälligkeitsdatum und einmal für das Lieferdatum. Dadurch erhalten Sie das eindeutige Modell aus Abbildung 4–10.



**Abbildung 4-10** Durch dreifaches Laden der Datumstabelle wird das Modell eindeutig.

Mit diesem Modell können wir jetzt Berichte erstellen, die beispielsweise die Beträge der Verkäufe anzeigen, die in einem Jahr bestellt, aber in einem anderen geliefert wurden (siehe Abbildung 4-11).

Sales Amount	Column Labels				
Row Labels	2007	2008	2009	2010	Grand Total
2007	1,412,267.47	46,948.48			1,459,215.95
2008		1,092,620.19	29,914.86		1,122,535.05
2009			1,193,051.71	49,482.90	1,242,534.61
<b>Grand Total</b>	<b>1,412,267.47</b>	<b>1,139,568.67</b>	<b>1,222,966.57</b>	<b>49,482.90</b>	<b>3,824,285.61</b>

**Abbildung 4-11** Der Bericht zeigt die Beträge der Verkäufe, die in einem Jahr bestellt, aber in einem anderen geliefert wurden.

Die Pivottabelle in Abbildung 4-11 ist auf den ersten Blick unübersichtlich. Es ist schwer zu erkennen, ob das Lieferjahr in den Spalten oder den Zeilen angegeben wird. Dass das Lieferjahr in den Spalten steht, können Sie herausfinden, indem Sie sich die Zahlen genauer ansehen, da die Lieferung grundsätzlich nach der

Bestellung erfolgt. Allerdings ist der Aufbau dieser Tabelle nicht so offensichtlich, wie er in einem guten Bericht sein sollte.

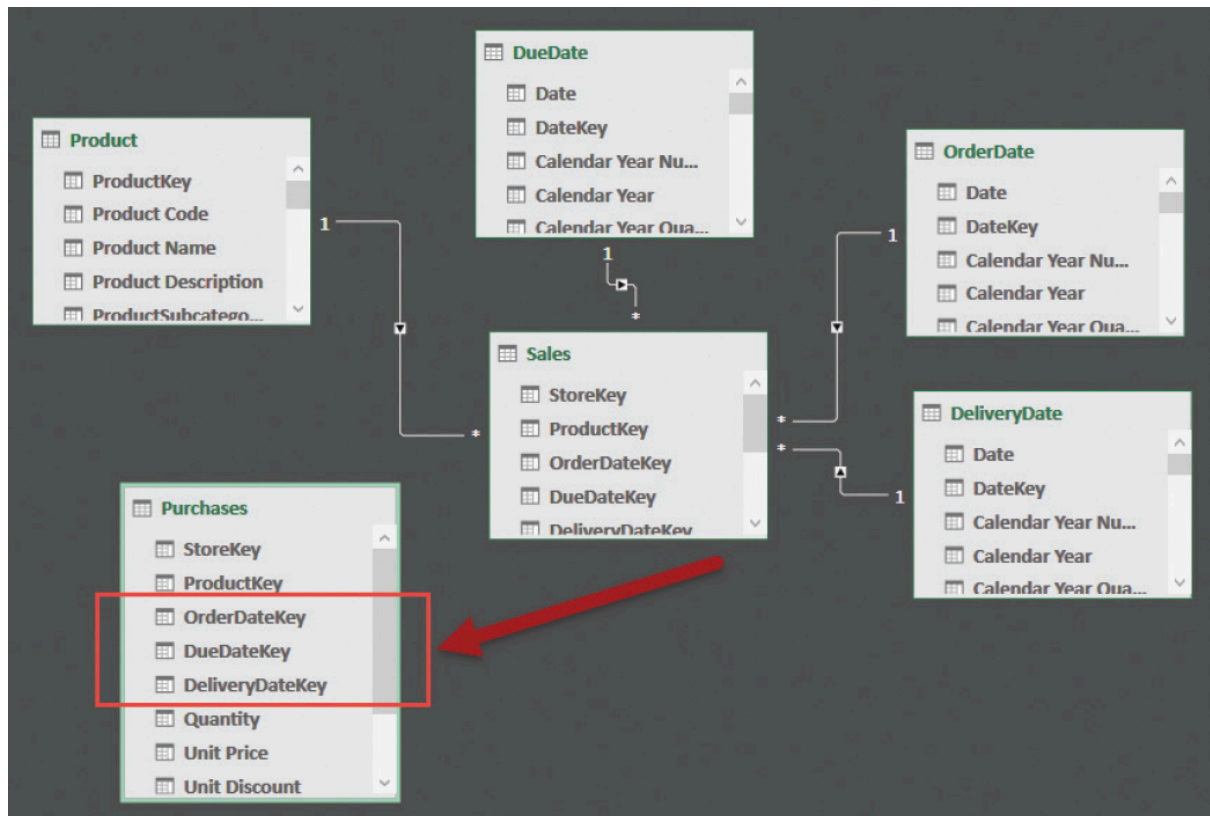
Um die Übersichtlichkeit zu erhöhen, können wir einfach den Zeilen- und Spaltenbezeichnungen für die Jahre die Präfixe *OY* für das Bestelljahr bzw. *DY* für das Lieferjahr voranstellen. Dazu muss die Abfrage der Tabelle *Calendar* geändert werden, es macht aber den Bericht viel übersichtlicher, wie Sie in Abbildung 4–12 sehen.

<b>Sales Amount</b>	<b>Column Labels</b> ▼				
<b>Row Labels</b> ▼	<b>DY 2007</b>	<b>DY 2008</b>	<b>DY 2009</b>	<b>DY 2010</b>	<b>Grand Total</b>
OY 2007	1,412,267.47	46,948.48			1,459,215.95
OY 2008		1,092,620.19	29,914.86		1,122,535.05
OY 2009			1,193,051.71	49,482.90	1,242,534.61
<b>Grand Total</b>	<b>1,412,267.47</b>	<b>1,139,568.67</b>	<b>1,222,966.57</b>	<b>49,482.90</b>	<b>3,824,285.61</b>

**Abbildung 4–12** Mit Präfixen für das Bestell- und das Lieferjahr wird der Bericht viel übersichtlicher.

Das mag bis jetzt so aussehen, als könnten Sie zur Handhabung von mehreren Datumsangaben einfach die Datumsdimension so oft wie nötig kopieren, sofern Sie darauf achten, die Spalten umzubenennen und Präfixe für die Spaltenwerte zu verwenden, um den Bericht übersichtlich zu halten. In gewisser Weise stimmt das auch. Aber wenn Sie dem Datenmodell eine weitere Faktentabelle hinzufügen, beispielsweise *Purchases*, wird die Situation viel komplizierter, wie Sie in Abbildung 4–13 sehen.

Allein durch Hinzufügen der Tabelle *Purchases* erhält das Modell drei weitere Datumsangaben, da es auch für Einkäufe ein Bestell-, ein Liefer- und ein Fälligkeitsdatum gibt. Die richtige Modellierung ist jetzt schon viel kniffliger. Sie könnten dem Modell jetzt drei weitere Datumsdimensionen hinzufügen, sodass es insgesamt sechs hätte. Diese Vielzahl von Datumsangaben aber ist für Benutzer verwirrend. Auch wenn das Modell sehr leistungsfähig sein mag, ist es schwer zu verwenden und nicht benutzerfreundlich. Versuchen Sie sich außerdem vorzustellen, was passiert, wenn Sie irgendwann noch eine weitere Faktentabelle hinzufügen müssen! Diese explosionsartige Vermehrung von Datumsdimensionen ist keine gute Sache.



**Abbildung 4-13** Die Tabelle *Purchases* weist drei weitere Datumsangaben auf.

Eine andere Möglichkeit besteht darin, die drei bereits in dem Modell vorhandenen Dimensionen zur Filterung sowohl der Einkäufe als auch der Verkäufe zu verwenden. Das bedeutet, dass *OrderDate*, *DueDate* und *DeliveryDate* sowohl *Sales* als auch *Purchases* nach dem Bestell-, Fälligkeits- bzw. Lieferdatum filtern können. Dadurch erhalten wir das Datenmodell aus Abbildung 4-14.

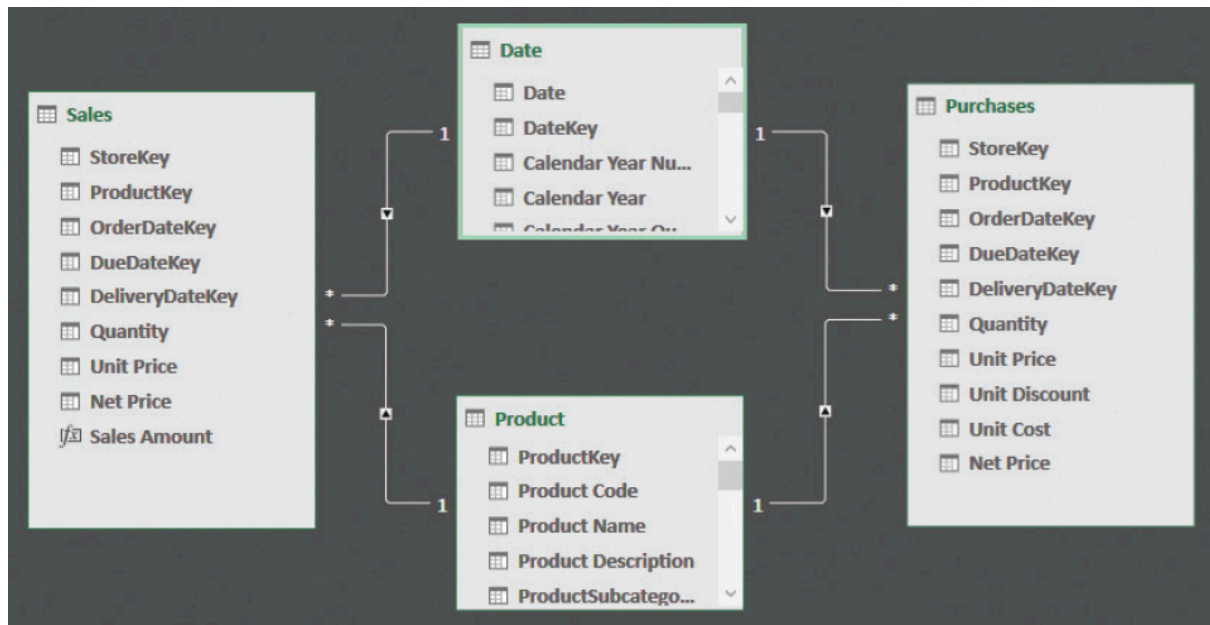
Das Modell in Abbildung 4-14 lässt sich schon viel leichter handhaben, ist aber immer noch zu kompliziert. Vor allem aber haben wir bei den neuen Dimensionen unverschämtes Glück gehabt, denn *Purchases* verwendet die gleichen Datumsangaben wie *Sales*, was in der Praxis nicht so häufig vorkommt. Es ist viel wahrscheinlicher, dass neue Faktentabellen mit Datumsangaben keine Gemeinsamkeiten mit bereits vorhandenen Faktentabellen haben. In einem solchen Fall müssten Sie entscheiden, ob Sie weitere Datumsdimensionen erstellen wollen, was das Modell unübersichtlicher macht, oder die neue mit einer verknüpfen möchten, die die vorhandenen Datumsangaben enthält. Da die Namen aber wahrscheinlich nicht hundertprozentig übereinstimmen, kann das die Verwendung durch die Benutzer erschweren.

Das Problem lässt sich viel einfacher lösen, wenn Sie der Versuchung widerstehen, mehrere Datumsdimensionen in dem Modell anzulegen. Wenn Sie bei einer einzigen Datumsdimension bleiben, ist das Modell viel übersichtlicher und verständlicher, wie Sie in Abbildung 4–15 sehen.

Durch die Verwendung einer einzigen Datumsdimension wird das Modell sehr leicht zu handhaben. Es ist unmittelbar einsichtig, dass *Date* sowohl *Sales* als auch *Purchases* jeweils nach deren Hauptdatumsspalte filtert, nämlich dem Bestelldatum. Auf den ersten Blick mag dieses Modell weniger leistungsfähig aussehen als das vorherige, und in gewissem Sinne ist dieser Eindruck auch korrekt. Bevor wir jedoch verkünden, dass das Modell tatsächlich weniger leistungsfähig ist, sollten wir uns zunächst einmal genauer ansehen, welche Unterschiede es bei den Analysemöglichkeiten zwischen einem Modell mit mehreren und einem Modell mit einer einzigen Datumsdimension gibt.



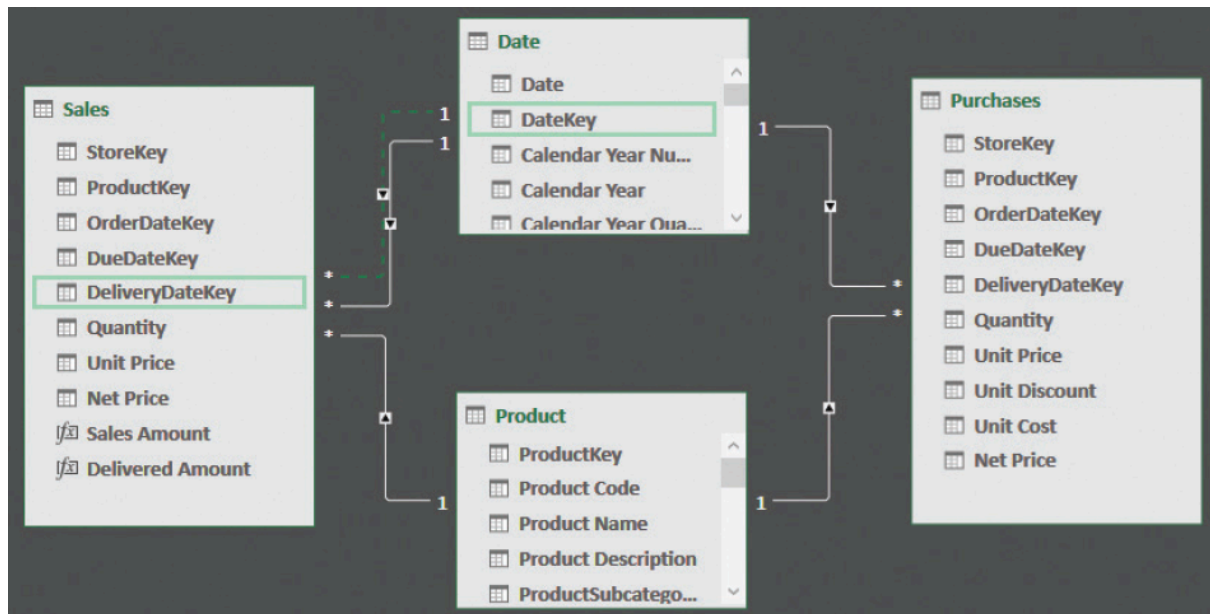
**Abbildung 4–14** Mit der Verwendung derselben Dimensionen zur Filterung beider Faktentabellen ist das Modell leichter zu handhaben.



**Abbildung 4-15** In diesem einfacheren Modell gibt es eine einzige Datumsdimension, die mit *OrderDate* in beiden Faktentabellen verknüpft ist.

Den Benutzern mehrere Datumstabellen zur Verfügung zu stellen, ermöglicht es, Berichte zu erstellen, die mehrere Datumsangaben auf einmal verwenden. Im vorherigen Beispiel haben Sie gesehen, dass das praktisch sein kann, um die Bestellbeträge mit den Beträgen der tatsächlich gelieferten Waren zu vergleichen. Allerdings müssen Sie sich hier fragen, ob wirklich mehrere Datumsangaben notwendig sind, um diese Information zu gewinnen. Die Antwort lautet nein. Sie können die gleiche Aufgabe mit berechneten Feldern lösen, die die erforderlichen Werte bestimmen, ohne dass dazu eine Änderung des Datenmodells notwendig ist.

Wenn Sie beispielsweise die Bestell- und die Lieferbeträge vergleichen wollen, können Sie eine inaktive Beziehung zwischen *Sales* und *Date* auf der Grundlage von *DeliveryDateKey* aufrechterhalten und für ein einzelnes berechnetes Feld aktivieren. Mit dieser zusätzlichen inaktiven Beziehung zwischen *Sales* und *Date* erhalten wir das Datenmodell aus Abbildung 4-16.



**Abbildung 4-16** Die Beziehung zwischen *DeliveryDateKey* und *DateKey* ist im Modell vorhanden, aber inaktiv.

Wenn diese Beziehung vorhanden ist, können Sie das berechnete Feld *Delivered Amount* wie folgt schreiben:

Delivered Amount :=

CALCULATE (

[Sales Amount],

USERELATIONSHIP ( Sales[DeliveryDateKey],  
'Date'[DateKey] )

)

Dieser Code aktiviert die inaktive Beziehung zwischen *Sales* und *Date* nur für die Dauer der Berechnung. Dadurch können Sie Ihre Daten anhand der Tabelle *Date* filtern und trotzdem Informationen mit Bezug zum Lieferdatum gewinnen, wie der Bericht in Abbildung 4-17 zeigt. Durch die Wahl eines geeigneten Namens

für das berechnete Feld beseitigen Sie auch alle Unklarheiten bei der Verwendung.

Die einfache Regel lautet daher, eine einzige Datumsdimension für das gesamte Modell zu erstellen. Dies ist natürlich keine strenge Regel. Es gibt Situationen, in denen es durchaus sinnvoll sein kann, mehrere Datumsdimensionen zu verwenden. Die Gründe dafür müssen aber gewichtig genug sein, um die Schwierigkeiten beim Umgang mit mehreren Datumstabellen zu rechtfertigen.

Row Labels	Sales Amount	Delivered Amount
CY 2007	1,459,215.95	1,410,787.80
CY 2008	1,122,535.05	1,145,421.73
CY 2009	1,242,534.61	1,221,566.90
CY 2010		46,509.18
<b>Grand Total</b>	<b>3,824,285.61</b>	<b>3,824,285.61</b>

**Abbildung 4-17** Das berechnete Feld *Delivered Amount* nutzt die Beziehung auf der Grundlage des Lieferdatums. Die Logik dahinter ist allerdings in dem berechneten Feld verborgen.

Nach unserer Erfahrung sind mehrere Datumstabellen in den meisten Datenmodellen nicht erforderlich. Eine reicht. Wenn Sie Berechnungen durchführen müssen, die unterschiedliche Datumsangaben nutzen, können Sie dafür berechnete Felder erstellen und inaktive Beziehungen nutzen. Meistens werden zusätzliche Datumsdimensionen nur hinzugefügt, weil die Erfordernisse des Modells nicht richtig erkannt sind. Wenn Sie vorhaben, das Modell um eine neue Datumsdimension zu ergänzen, sollten Sie sich daher immer fragen, ob Sie sie *wirklich* brauchen oder ob Sie dieselben Werte auch mit DAX-Code berechnen können. Wenn Letzteres der Fall ist, sollten Sie sich für mehr DAX-Code und weniger Datumsdimensionen entscheiden. Diese Entscheidung werden Sie nie bereuen.

## Umgang mit Datum und Uhrzeit

---

In fast jedem Modell ist eine Datumsdimension erforderlich, doch eine Uhrzeitdimension ist viel weniger häufig. Es gibt jedoch Situationen, in denen sowohl Datum als auch Uhrzeit wichtige Dimensionen sind, und in solchen Fällen müssen Sie genau wissen, wie Sie damit umzugehen haben.

Der erste wichtige Punkt ist, dass eine Datumstabelle keine Uhrzeitangaben enthalten kann. Wenn Sie eine Tabelle als Datumstabelle kennzeichnen (was Sie tun müssen, um die Zeitinformationsfunktionen nutzen zu können), müssen Sie die Anforderungen von DAX dafür erfüllen. Dazu gehört es auch, dass die Spalte, die den `datetime`-Wert enthält, nur eine Granularität bis zur Ebene einzelner Tage aufweisen darf, aber keine Uhrzeitangaben. Wenn Sie eine Datumstabelle mit Uhrzeitangaben nutzen wollen, erhalten Sie eine Fehlermeldung von der Engine. Allerdings kann die Engine die Zeitinformationsfunktionen auch nicht korrekt nutzen, wenn ein Datum mehrmals auftaucht.

Was können Sie also tun, um auch mit Uhrzeitangaben zu arbeiten? Die einfachste und effizienteste Lösung besteht darin, eine Dimension für das Datum und eine weitere für die Uhrzeit zu erstellen. Letztere können Sie mit dem folgenden einfachen M-Code in Power Query anlegen:

Let

```
StartTime = #datetime(1900,1,1,0,0,0),

Increment = #duration(0,0,1,0),

Times = List.DateTimes(StartTime, 24*60,
Increment),

TimesAsTable =
Table.FromList(Times,Splitter.SplitByNothing()),

RenameTime = Table.RenameColumns(TimesAsTable,
{"Column1", "Time"}),

ChangedDataType =
Table.TransformColumnTypes(RenameTime,{"Time",
type time}),
```

```
AddHour = Table.AddColumn(  
  
    ChangedDataType,  
  
    "Hour",  
  
    each  
    Text.PadStart(Text.From(Time.Hour([Time])), 2,  
    "0" )  
  
),
```

```
AddMinute = Table.AddColumn(  
  
    AddHour,  
  
    "Minute",  
  
    each  
    Text.PadStart(Text.From(Time.Minute([Time])),  
    2, "0" )  
  
),
```

```
AddHourMinute = Table.AddColumn(  
  
    AddMinute,  
  
    "HourMinute", each [Hour] & ":" & [Minute]
```

```

    ),

    AddIndex = Table.AddColumn(

        AddHourMinute,

        "TimeIndex",

        each Time.Hour([Time]) * 60 +
        Time.Minute([Time])

    ),

    Result = AddIndex

in

    Result

```

Dieses Skript erstellt eine Tabelle wie die in Abbildung 4–18. Sie enthält die Spalte *TimeIndex* (mit den Zahlen von 0 bis 1439), die Sie zur Verknüpfung mit der Faktentabelle nutzen können, und einige weitere Spalten zur Filterung Ihrer Daten. Wenn in Ihrer Tabelle eine andere Spalte für die Uhrzeit verwendet wird, können Sie das vorstehende Skript einfach abwandeln, um eine Zeitspalte als Primärschlüssel zu erstellen.

Time	Hour	Minute	TimeIndex	HourMinute
00.00.00	00	00	0	00:00
00.01.00	00	01	1	00:01
00.02.00	00	02	2	00:02
00.03.00	00	03	3	00:03
00.04.00	00	04	4	00:04
00.05.00	00	05	5	00:05
00.06.00	00	06	6	00:06
00.07.00	00	07	7	00:07
00.08.00	00	08	8	00:08
00.09.00	00	09	9	00:09
00.10.00	00	10	10	00:10
00.11.00	00	11	11	00:11
00.12.00	00	12	12	00:12
00.13.00	00	13	13	00:13
00.14.00	00	14	14	00:14
00.15.00	00	15	15	00:15

**Abbildung 4-18** Eine einfache Uhrzeittabelle, erstellt mit Power Query

Zur Berechnung des Zeitindex wird die Stunde mit 60 multipliziert und die Anzahl der Minuten addiert. Dadurch können Sie den Zeitindex leicht als Schlüssel in Ihre Faktentabelle aufnehmen. Diese Berechnung sollte in der Datenquelle erfolgen, die die Tabelle speist.

Durch die Verwendung einer eigenen Uhrzeittabelle können Sie Ihre Daten nach Stunden, Minuten oder sonstigen Spalten filtern, die Sie dieser Tabelle hinzufügen. Sehr häufig werden auch Tageszeiten (Vormittag, Nachmittag, Nacht) oder Zeiträume, z. B. Stundenintervalle, verwendet (wie in Abbildung 4-19).

Sales Amount	Column Labels <input type="text"/>			
Row Labels	<input type="text"/> CY 2007	CY 2008	CY 2009	Grand Total
From 06:00 to 07:00	79,518.54	47,981.26	67,825.28	195,325.08
From 07:00 to 08:00	27,692.12	39,036.69	37,788.93	104,517.75
From 08:00 to 09:00	54,368.20	52,602.75	56,441.84	163,412.79
From 09:00 to 10:00	69,017.22	55,524.44	55,657.92	180,199.58
From 10:00 to 11:00	63,355.05	49,343.90	44,823.41	157,522.37
From 11:00 to 12:00	51,625.55	49,563.19	42,184.88	143,373.62
From 12:00 to 13:00	52,189.03	29,557.34	45,533.05	127,279.42
From 13:00 to 14:00	47,517.66	35,557.72	46,450.37	129,525.75
From 14:00 to 15:00	74,327.75	52,080.15	45,249.45	171,657.35
From 15:00 to 16:00	48,098.40	46,725.44	37,985.21	132,809.04
From 16:00 to 17:00	43,919.20	36,328.50	53,859.38	134,107.08
From 17:00 to 18:00	62,586.62	47,388.08	47,082.21	157,056.92
From 18:00 to 19:00	64,856.62	40,480.32	74,347.68	179,684.62
From 19:00 to 20:00	68,391.58	32,012.46	52,723.16	153,127.20
<b>Grand Total</b>	<b>807,463.55</b>	<b>614,182.24</b>	<b>707,952.78</b>	<b>2,129,598.57</b>

**Abbildung 4-19** Die Uhrzeitdimension ist praktisch für Berichte, die beispielsweise Verkäufe nach Stunden anzeigen.

Es gibt jedoch auch Situationen, in denen es nicht nötig ist, Zahlen auf Zeiträume zu verteilen. Das ist beispielsweise der Fall, wenn Sie Werte basierend auf dem Zeitunterschied zwischen zwei Ereignissen berechnen wollen oder wenn Sie an der Anzahl der Ereignisse zwischen zwei Uhrzeiten interessiert sind, etwa an der Anzahl der Kunden, die Ihr Geschäft zwischen 8.00 Uhr am 1. Januar und 13.00 Uhr am 7. Januar betreten haben. Solche Fälle sind etwas anspruchsvoller und werden in Kapitel 7 behandelt.

## Zeitinformationsberechnungen

Wenn Sie Ihr Datenmodell auf die richtige Weise vorbereitet haben, lassen sich Zeitinformationsberechnungen leicht schreiben. Dazu müssen Sie auf die Kalendertabelle einen Filter anwenden, der die Zeilen für die gewünschten Zeiträume zeigt. Es gibt eine Menge von Funktionen, die Sie einsetzen können, um solche Filter zu erstellen. Beispielsweise können Sie aufgelaufene Beträge seit Jahresbeginn wie folgt schreiben:

Sales YTD :=

```
CALCULATE (  
  
    [Sales Amount],  
  
    DATESYTD ( 'Date' [Date] )  
  
)
```

DATESYTD gibt die Menge der Datumsangaben vom 1. Januar des zurzeit ausgewählten Zeitraums bis zum letzten Datum in dem Kontext zurück. Weitere praktische Funktionen sind SAMEPERIODLASTYEAR, PARALLELPERIOD und LASTDAY. Durch die Kombination dieser Funktionen können Sie noch vielschichtigere Aggregationen erreichen.

Wenn Sie beispielsweise die aufgelaufenen Beträge im Vergleichszeitraum des vorherigen Jahres berechnen wollen, können Sie dazu die folgende Formel verwenden:

Sales PYTD :=

```
CALCULATE (  
  
    [Sales Amount],  
  
    DATESYTD ( SAMEPERIODLASTYEAR ( 'Date' [Date] ) )  
  
)
```

Eine weitere praktische Zeitinformationsfunktion ist DATESINPERIOD, die die Datumsangaben in einem bestimmten Zeitraum zurückgibt. Das können Sie wie im folgenden Beispiel für die Berechnung gleitender Durchschnitte nutzen. Hier gibt DATESINPERIOD die letzten zwölf Monate zurück, wobei das letzte Datum im Filterkontext als Bezugspunkt fungiert.

```
Sales Avg12M :=
```

```
CALCULATE (

    [Sales Amount] / COUNTROWS ( VALUES (
        'Date' [Month] ) ),

    DATESINPERIOD (

        'Date' [Date],

        MAX ( 'Date' [Date] ),

        -12,

        MONTH

    )

)
```

Das Ergebnis sehen Sie in Abbildung 4–20.

Row Labels	Sales Amount	Sales Avg12M
+ 2007	1,459,215.95	121,601.33
+ 2008	1,122,535.05	93,544.59
- 2009	1,242,534.61	103,544.55
January	71,828.15	94,146.79
February	59,980.01	94,048.68
March	71,327.93	94,596.90
April	103,551.11	93,559.09
May	160,137.28	96,306.46
June	93,484.82	96,631.05
July	145,604.22	101,094.13
August	98,972.35	97,565.15
September	90,457.03	95,765.51
October	91,665.16	98,482.67
November	133,481.80	100,581.92
December	122,044.75	103,544.55
<b>Grand Total</b>	<b>3,824,285.61</b>	

**Abbildung 4-20** Das berechnete Feld ermittelt den Durchschnitt über zwölf Monate.

## Geschäftskalender

Ein weiterer sehr guter Grund, aus dem es sinnvoll ist, Ihre eigene Kalendertabelle aufzustellen, besteht darin, dass Sie dadurch sehr leicht mit einem Geschäftskalender arbeiten können. Es ist damit sogar möglich, noch anspruchsvollere Sonderformen wie Wochen- oder Saisonkalender zu nutzen.

Für einen Geschäftskalender fügen Sie nicht der Faktentabelle weitere Spalten hinzu, sondern der Datumstabelle, sodass Sie Datenschnitte sowohl nach dem Kalender- als auch nach dem Geschäftsjahr durchführen können. Nehmen wir als Beispiel an, Sie brauchen einen Geschäftskalender mit dem Juli als ersten Monat. Er geht also vom 1. Juli bis zum 30. Juni. In diesem Fall müssen Sie den Kalender so abändern, dass er die Geschäftsmonate zeigt, und einige Berechnungen anpassen, damit sie den Geschäftskalender nutzen.

Als Erstes müssen Sie geeignete Spalten für die Geschäftsmonate hinzufügen (falls sie noch nicht vorhanden sind). Manche verwenden die Monatsnamen wie

hier Juli als Bezeichnungen für die Geschäftsmonate, während andere lieber Zahlen angeben, also *Geschäftsmonat 01* statt *Juli*. In diesem Beispiel bleiben wir jedoch bei den üblichen Monatsnamen.

In jedem Fall aber brauchen Sie zur Sortierung eine zusätzliche Spalte mit der Bezeichnung des Geschäftsmonats. In Standardkalendern haben Sie eine Spalte für den Monatsnamen, die nach der Monatsnummer sortiert wird, sodass Januar an der ersten Stelle steht und Dezember an der letzten. In unserem Geschäftskalender soll jedoch Juli der erste Monat sein und Juni der letzte. Da es nicht möglich ist, eine Spalte nach unterschiedlichen Aspekten zu sortieren, müssen Sie die Spalte mit dem Monatsnamen für die Geschäftsmonate duplizieren und eine neue Sortierungsspalte für die gewünschte Anordnung erstellen.

Anschließend können Sie das Modell nach den Spalten in der Kalendertabelle durchsuchen und die Monate auf die gewünschte Weise sortieren lassen. Allerdings funktionieren einige Berechnungen dabei nicht wie erwartet. Schauen Sie sich beispielsweise die Berechnung der aufgelaufenen Verkäufe seit Jahresbeginn (*Sales YTD*) in Abbildung 4-21 an.

Wenn Sie sich die Pivottabelle genau ansehen, können Sie erkennen, dass die Werte für die aufgelaufenen Beträge im Januar 2008 zurückgesetzt werden statt im Juli. Das liegt daran, dass die Zeitinformationsfunktionen mit dem Standard- und nicht mit dem Geschäftskalender arbeiten. Einige Funktionen verfügen über einen zusätzlichen Parameter, mit dem Sie festlegen können, ob ein Geschäftskalender verwendet werden soll. Dazu gehört auch die Funktion *DATESYTD* zur Berechnung aufgelaufener Beträge seit Jahresbeginn. Damit sie den Geschäftskalender nutzt, fügen Sie ihr einen zweiten Parameter hinzu, der angibt, an welchem Tag welchen Monats der Kalender endet:

```
Sales YTD Fiscal :=
```

```
CALCULATE (
```

```
    [Sales Amount],
```

```
    DATESYTD ( 'Date'[Date], "06/30" )
```

)

Row Labels	Sales Amount	Sales YTD
<b>[-] FY 2007</b>	<b>695,564.75</b>	<b>695,564.75</b>
January	101,097.12	101,097.12
February	108,553.20	209,650.32
March	119,707.83	329,358.16
April	121,085.74	450,443.90
May	123,413.41	573,857.31
June	121,707.44	695,564.75
<b>[-] FY 2008</b>	<b>1,286,923.00</b>	<b>523,271.80</b>
July	139,381.00	834,945.75
August	87,384.31	922,330.06
September	155,275.94	1,077,606.00
October	99,872.65	1,177,478.64
November	122,522.86	1,300,001.50
December	159,214.45	1,459,215.95
January	64,601.67	64,601.67
February	61,157.39	125,759.06
March	64,749.27	190,508.33
April	116,004.84	306,513.17
May	127,168.83	433,682.00
June	89,589.80	523,271.80
<b>[+] FY 2009</b>	<b>1,159,572.55</b>	<b>560,309.30</b>
<b>[+] FY 2010</b>	<b>682,225.31</b>	
<b>Grand Total</b>	<b>3,824,285.61</b>	

**Abbildung 4-21** Die Berechnung aufgelaufener Beträge seit Jahresbeginn funktioniert im Geschäftskalender nicht richtig.

Die Pivottabelle in Abbildung 4-22 zeigt nebeneinander die aufgelaufenen Beträge seit Beginn des Kalenderjahres und seit Beginn des Geschäftsjahres.

Row Labels	Sales Amount	Sales YTD	Sales YTD Fiscal
<b>FY 2007</b>	<b>695,564.75</b>	<b>695,564.75</b>	<b>695,564.75</b>
January	101,097.12	101,097.12	101,097.12
February	108,553.20	209,650.32	209,650.32
March	119,707.83	329,358.16	329,358.16
April	121,085.74	450,443.90	450,443.90
May	123,413.41	573,857.31	573,857.31
June	121,707.44	695,564.75	695,564.75
<b>FY 2008</b>	<b>1,286,923.00</b>	<b>523,271.80</b>	<b>1,286,923.00</b>
July	139,381.00	834,945.75	139,381.00
August	87,384.31	922,330.06	226,765.31
September	155,275.94	1,077,606.00	382,041.25
October	99,872.65	1,177,478.64	481,913.89
November	122,522.86	1,300,001.50	604,436.75
December	159,214.45	1,459,215.95	763,651.19
January	64,601.67	64,601.67	828,252.87
February	61,157.39	125,759.06	889,410.25
March	64,749.27	190,508.33	954,159.53
April	116,004.84	306,513.17	1,070,164.36
May	127,168.83	433,682.00	1,197,333.20
June	89,589.80	523,271.80	1,286,923.00
<b>FY 2009</b>	<b>1,159,572.55</b>	<b>560,309.30</b>	<b>1,159,572.55</b>
<b>FY 2010</b>	<b>682,225.31</b>		<b>682,225.31</b>
<b>Grand Total</b>	<b>3,824,285.61</b>		

**Abbildung 4-22** Sales YTD Fiscal setzt die Beträge jetzt korrekt auf Ende Juni zurück.

Die verschiedenen möglichen Berechnungen erfordern natürlich jeweils eine eigene Vorgehensweise, aber die in DAX enthaltenen Standardfunktionen für Zeitinformationen lassen sich leicht an Geschäftskalender anpassen. Im letzten Abschnitt dieses Kapitels werden wir uns mit einer weiteren nützlichen Variante

beschäftigen, nämlich mit Wochenkalendern. Wenn Sie andere Bedürfnisse haben oder noch anspruchsvollere Kalender benötigen, ist wahrscheinlich eine aufwendigere Vorgehensweise notwendig. Schauen Sie sich dazu die Zeitinformationsmuster auf <http://www.daxpatterns.com/time-patterns> an.

Der wichtige Punkt im Hinblick auf das Thema dieses Buches ist jedoch, dass Sie keine zusätzlichen Tabellen brauchen, um Geschäftskalender nutzen zu können. Wenn Sie Ihre Datumstabelle auf die richtige Weise gestaltet haben und verwenden, ist die Handhabung verschiedener Kalender sehr simpel und kann einfach durch Anpassung der Kalendertabelle erreicht werden.

Wenn Sie zulassen, dass Power BI Desktop oder Excel die Zeitinformationsspalten für Sie erstellen, können Sie diese einfachen Techniken jedoch nicht einsetzen. Sie müssen dann ganz auf sich allein gestellt herausfinden, wie Sie die richtigen Formeln schreiben.

## **Berechnungen mit Arbeitstagen**

---

Nicht alle Tage sind Arbeitstage, und sehr oft müssen Sie diesen Unterschied in Ihren Berechnungen berücksichtigen, etwa wenn Sie die Anzahl der Arbeitstage zwischen zwei Kalenderdaten oder in einem bestimmten Zeitraum bestimmen müssen. In diesem Abschnitt beschäftigen wir uns mit der Datenmodellierung für Berechnungen mit Arbeitstagen.

Die erste (und wichtigste) Überlegung ist die Frage, ob ein bestimmter Tag immer ein Arbeitstag ist oder ob das von anderen Faktoren abhängt. Wenn Sie beispielsweise mehrere Bundesländer oder andere Staaten berücksichtigen müssen, kann es sein, dass ein bestimmter Tag in einer Region ein Arbeitstag ist und in der anderen ein Feiertag. Wie Sie noch sehen werden, brauchen Sie für die Berücksichtigung von Feiertagen nach Land oder Region aufwendigere Modelle. Zu Anfang wollen wir uns jedoch mit einem einfacheren Modell befassen, in dem wir nur mit den Feiertagen für eine einzige Region zu tun haben.

### **Arbeitstage in einer einzigen Region**

Als Ausgangspunkt verwenden wir ein einfaches Datenmodell mit den Tabellen *Date*, *Product* und *Sales*, wobei wir uns jedoch auf *Date* konzentrieren. Diese Tabelle sieht zu Anfang so aus wie in Abbildung 4-23.

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week
1/1/05	20050101	CY 2005	1	January	7	Saturday
2/1/05	20050102	CY 2005	1	January	1	Sunday
3/1/05	20050103	CY 2005	1	January	2	Monday
4/1/05	20050104	CY 2005	1	January	3	Tuesday
5/1/05	20050105	CY 2005	1	January	4	Wednesday
6/1/05	20050106	CY 2005	1	January	5	Thursday
7/1/05	20050107	CY 2005	1	January	6	Friday
8/1/05	20050108	CY 2005	1	January	7	Saturday
9/1/05	20050109	CY 2005	1	January	1	Sunday
10/1/05	20050110	CY 2005	1	January	2	Monday
11/1/05	20050111	CY 2005	1	January	3	Tuesday
12/1/05	20050112	CY 2005	1	January	4	Wednesday
13/1/05	20050113	CY 2005	1	January	5	Thursday

**Abbildung 4-23** Ausgangspunkt für die Arbeitstaganalyse ist eine einfache Datumstabelle.

Diese Tabelle enthält keine Angaben darüber, ob ein Tag ein Arbeitstag ist oder nicht. Für unsere Zwecke gibt es zwei Arten von arbeitsfreien Tagen, nämlich Wochenenden und Feiertage. Wenn das Wochenende in Ihrem Land auf Samstag und Sonntag fällt, können Sie mit dem folgenden Code ganz einfach eine berechnete Spalte erstellen, die besagt, ob der jeweilige Tag zum Wochenende gehört oder nicht. Fällt das Wochenende auf andere Tage, müssen Sie die Formel entsprechend anpassen.

```
'Date'[IsWorkingDay] =
```

```
INT (
```

```
    AND (
```

```
        'Date'[Day of Week Number] <> 1,
```

```
        'Date'[Day of Week Number] <> 7
```

```
    )
```

)

Wir haben die boolesche Bedingung in einen Integer umgewandelt, damit wir ihren Wert summieren und damit die Anzahl der Arbeitstage zählen können. Diese Anzahl können Sie mit dem folgenden berechneten Feld auf einfache Weise für gegebene Zeiträume ermitteln:

`NumOfWorkingDays = SUM ( 'Date'[IsWorkingDay] )`

Dieses berechnete Feld gibt uns schon eine gute Übersicht, wie Sie in Abbildung 4-24 sehen.

Calendar Year	Month	NumOfWorkingDays
CY 2005	January	21
	February	20
	March	23
	April	21
	May	22
	June	22
	July	21
	August	23
	September	22
	October	21
	November	22
	December	22
	<b>Total</b>	<b>260</b>

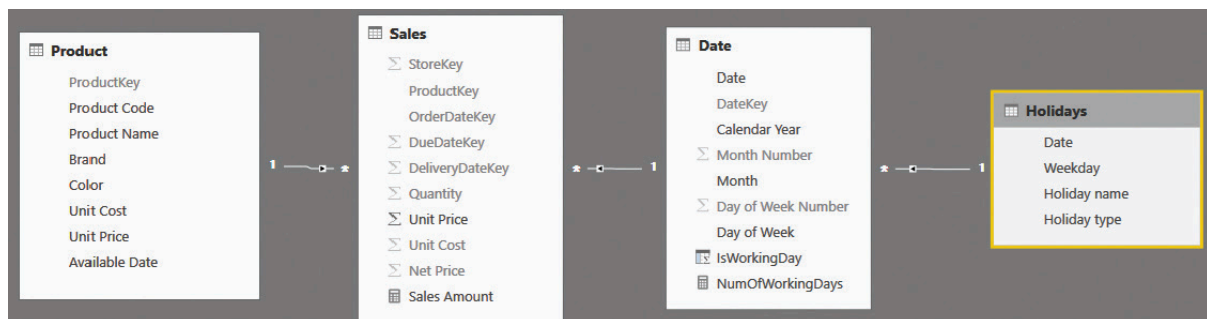
**Abbildung 4-24** NumOfWorkingDays berechnet die Anzahl der Arbeitstage in dem ausgewählten Zeitraum.

Damit haben wir die Samstage und Sonntage berücksichtigt, aber noch nicht die Feiertage. Für das folgende Beispiel haben wir die US-Bundesfeiertage des Jahres 2009 von [www.timeanddate.com](http://www.timeanddate.com) bezogen und anschließend die Tabelle aus Abbildung 4-25 mit dem Abfrage-Editor von Power BI Desktop erstellt.

Date	Weekday	Holiday name	Holiday type
01/01/2009	Thursday	New Year's Day	Federal Holiday
19/01/2009	Monday	Martin Luther King Day	Federal Holiday
16/02/2009	Monday	Presidents' Day	Federal Holiday
25/05/2009	Monday	Memorial Day	Federal Holiday
03/07/2009	Friday	Independence Day observed	Federal Holiday
04/07/2009	Saturday	Independence Day	Federal Holiday
07/09/2009	Monday	Labor Day	Federal Holiday
12/10/2009	Monday	Columbus Day	Federal Holiday
11/11/2009	Wednesday	Veterans Day	Federal Holiday
26/11/2009	Thursday	Thanksgiving Day	Federal Holiday
25/12/2009	Friday	Christmas Day	Federal Holiday

**Abbildung 4-25** Die Tabelle *Holidays* zeigt eine Liste der US-Bundesfeiertage.

Je nachdem, ob die Spalte *Date* in der Tabelle *Holidays* ein Schlüssel ist oder nicht, haben Sie jetzt zwei Möglichkeiten. Ist sie ein Schlüssel, so können Sie eine Beziehung zwischen *Date* und *Holidays* einrichten, um das Modell aus Abbildung 4-26 zu erstellen.



**Abbildung 4-26** Wenn *Date* ein Primärschlüssel ist, kann *Holidays* auf einfache Weise mit dem Modell verknüpft werden.

Nach dem Einrichten der Beziehung können Sie den Code für die berechnete Spalte *IsWorkingDay* um eine zusätzliche Überprüfung erweitern. Dadurch wird ein Tag als Arbeitstag gekennzeichnet, wenn er nicht auf einen Samstag oder Sonntag fällt und nicht in der Tabelle *Holidays* erscheint. Dazu dient der folgende Code:

```
'Date' [IsWorkingDay] =
```

```

INT (

    AND (

        AND (

            'Date'[Day of Week Number] <> 1,

            'Date'[Day of Week Number] <> 7

        ),

        ISBLANK ( RELATED ( Holidays[Date] ) )

    )

)

```

Das Modell ähnelt sehr stark einem Sternschema. Es ist eine Schneeflocke, aber da sowohl *Date* als auch *Holidays* klein sind, gibt es keine Leistungsprobleme.

Es kann jedoch auch vorkommen, dass die Spalte *Date* in *Holidays* kein Schlüssel ist. Wenn mehrere Feiertage auf denselben Tag fallen, dann enthält *Holidays* mehrere Spalten mit demselben Datum. In diesem Fall müssen Sie eine 1:n-Beziehung mit *Date* als Ziel und *Holidays* als Quelle einrichten (in der Tabelle *Date* ist die Spalte *Date* definitiv ein Primärschlüssel) und den Code wie folgt ändern:

```
'Date'[IsWorkingDay] =
```

```
INT (
```

```

AND (
    AND (
        'Date'[Day of Week Number] <> 1,
        'Date'[Day of Week Number] <> 7
    ),
    ISEMPTY ( RELATEDTABLE ( Holidays ) )
)
)

```

Die einzige Änderung betrifft die Zeile, in der geprüft wird, ob das Datum in der Tabelle *Holidays* aufgeführt ist. Anstatt der schnelleren Funktion `RELATED` müssen Sie hier `RELATEDTABLE` verwenden und prüfen, ob *Holidays* leer ist. Da wir hier mit einer berechneten Spalte arbeiten, ist der geringe Leistungsverlust völlig akzeptabel.

## Arbeitstage in mehreren Regionen

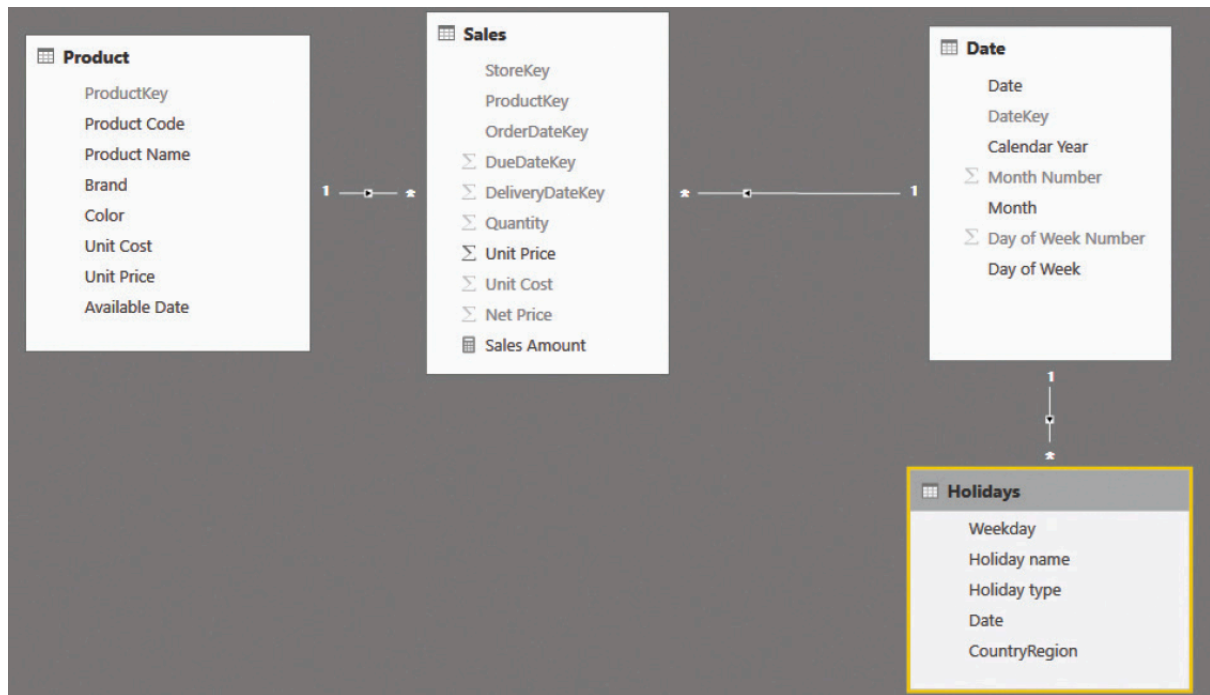
Wie Sie gesehen haben, ist die Modellierung von Feiertagen ziemlich einfach, solange Sie es nur mit einer einzigen Region zu tun haben. Es wird jedoch komplizierter, wenn Sie Feiertage in unterschiedlichen Ländern berücksichtigen müssen, denn dann können Sie sich nicht mehr auf berechnete Spalten verlassen. Je nach gewähltem Land kann die Spalte *IsHoliday* unterschiedliche Werte aufweisen.

Wenn Sie nur zwei Länder berücksichtigen müssen, ist es am einfachsten, zwei Spalten für *IsHoliday* zu erstellen, z. B. *IsHolidayGermany* und *IsHolidayChina*, und für die berechneten Felder dann jeweils die passende Spalte zu nehmen. Bei mehr als zwei Ländern geht das jedoch nicht mehr. Sehen wir uns eine solche Situation einmal ausführlich an. Beachten Sie, dass die Tabelle *Holidays* jetzt einen anderen Inhalt hat als zuvor, nämlich den aus Abbildung 4–27. Insbesondere enthält sie die neue Spalte *CountryRegion*, die die Region angibt, in der der Feiertag abgehalten wird. Das Datum ist jetzt auf jeden Fall kein Schlüssel mehr, da auf ein und dasselbe Datum Feiertage in mehreren Ländern fallen können.

Das neue Datenmodell in Abbildung 4–28 stellt eine leichte Abwandlung des vorherigen Modells dar. Der Hauptunterschied besteht darin, dass die Beziehung zwischen *Date* und *Holidays* jetzt in umgekehrter Richtung verläuft.

Date	Weekday	Holiday name	Holiday type	CountryRegion
01/01/2009	Thursday	New Year's Day	National holiday	China
01/01/2009	Thursday	New Year's Day	Federal Holiday	United States
01/01/2009	Thursday	New Year's Day	National holiday	Germany
01/05/2009	Friday	Labour Day	National holiday	China
01/05/2009	Friday	May Day	National holiday	Germany
01/06/2009	Monday	Whit Monday	National holiday	Germany
01/10/2009	Thursday	National Day	National holiday	China
02/10/2009	Friday	National Day Golden Week holiday	National holiday	China
03/07/2009	Friday	Independence Day observed	Federal Holiday	United States
03/10/2009	Saturday	National Day Golden Week holiday	National holiday	China
03/10/2009	Saturday	Mid-Autumn Festival	National holiday	China
03/10/2009	Saturday	Day of German Unity	National holiday	Germany
04/07/2009	Saturday	Independence Day	Federal Holiday	United States

**Abbildung 4–27** Die Tabelle *Holidays* mit Feiertagen verschiedener Länder



**Abbildung 4-28** Das Datenmodell für mehrere Länder sieht dem Modell für ein einziges Land sehr ähnlich.

Das Problem bei der Berücksichtigung mehrerer Länder besteht darin, dass wir genau wissen müssen, was die produzierten Zahlen bedeuten sollen. Die einfache Frage: »Wie viele Arbeitstage gibt es im Januar?«, hat keine klare Bedeutung mehr. Die Anzahl der Arbeitstage können Sie jetzt nur berechnen, wenn Sie auch das Land angeben.

Um die Problematik besser zu verstehen, schauen Sie sich Abbildung 4-29 an. Das berechnete Feld in diesem Bericht zählt (mithilfe von COUNTROWS) die Zeilen in der Tabelle *Holidays* und berechnet damit die Anzahl der Feiertage in jedem Land.

Die Zahlen für die einzelnen Länder sind korrekt, aber die Summe pro Monat gibt lediglich die Summe der Werte in den einzelnen Zellen an, ohne zu berücksichtigen, dass ein Tag in einem Land ein Feiertag sein mag und in einem anderen nicht. So gibt es beispielsweise im Februar einen Feiertag in den USA, aber nicht in China und Deutschland. Wie viele Feiertage gibt es also insgesamt im Februar? Diese Frage ist nicht sehr sinnvoll, die Gesamtzahl an Feiertagen über alle betrachteten Länder hinweg sagt überhaupt nichts aus. Eine sinnvolle Antwort muss jeweils auf ein Land bezogen sein.

Month	China	Germany	United States	Total
January	4	1	2	7
February			1	1
April	1	1		2
May	2	2	1	5
June		1		1
July			2	2
September			1	1
October	4	1	1	6
November			2	2
December		2	1	3
<b>Total</b>	<b>11</b>	<b>8</b>	<b>11</b>	<b>30</b>

**Abbildung 4-29** Anzahl der Feiertage pro Monat in den einzelnen Ländern

Bei der Definition des Modells müssen Sie jetzt die Bedeutung der Frage, ob ein Tag ein Arbeitstag ist oder nicht, genauer festlegen. Mit dem Muster `IF ( HASONEVALUE ( ) )` von DAX können Sie vor der Berechnung prüfen, ob in dem Bericht ein einzelnes Land ausgewählt wurde.

Es gibt jedoch noch einen weiteren Punkt, den Sie beachten müssen, bevor Sie die endgültige Formel schreiben können. Sie berechnen die Anzahl der Arbeitstage, indem Sie die Anzahl der Feiertage (aus der Tabelle *Holidays*) von der Gesamtzahl der Tage ohne die Samstage und Sonntage subtrahieren. Wenn ein Feiertag aber auf ein Wochenende fällt, dürfen Sie ihn nicht berücksichtigen. Dieses Problem können Sie dadurch lösen, dass Sie eine bidirektionale Filterung vornehmen und die Tage zählen, die weder Samstage oder Sonntage sind noch in der Tabelle *Holidays* aufgeführt sind. Die Formel lautet daher wie folgt:

`NumOfWorkingDays :=`

`IF (`

`OR (`

```

HASONEVALUE ( Holidays[CountryRegion] ),

ISEMPTY ( Holidays )

),

CALCULATE (

    COUNTRROWS ( 'Date' ),

    AND (

        'Date'[Day of Week Number] <> 1,

        'Date'[Day of Week Number] <> 7

    ),

    EXCEPT ( VALUES ( 'Date'[Date] ), VALUES (
Holidays[Date] ) )

)

)

```

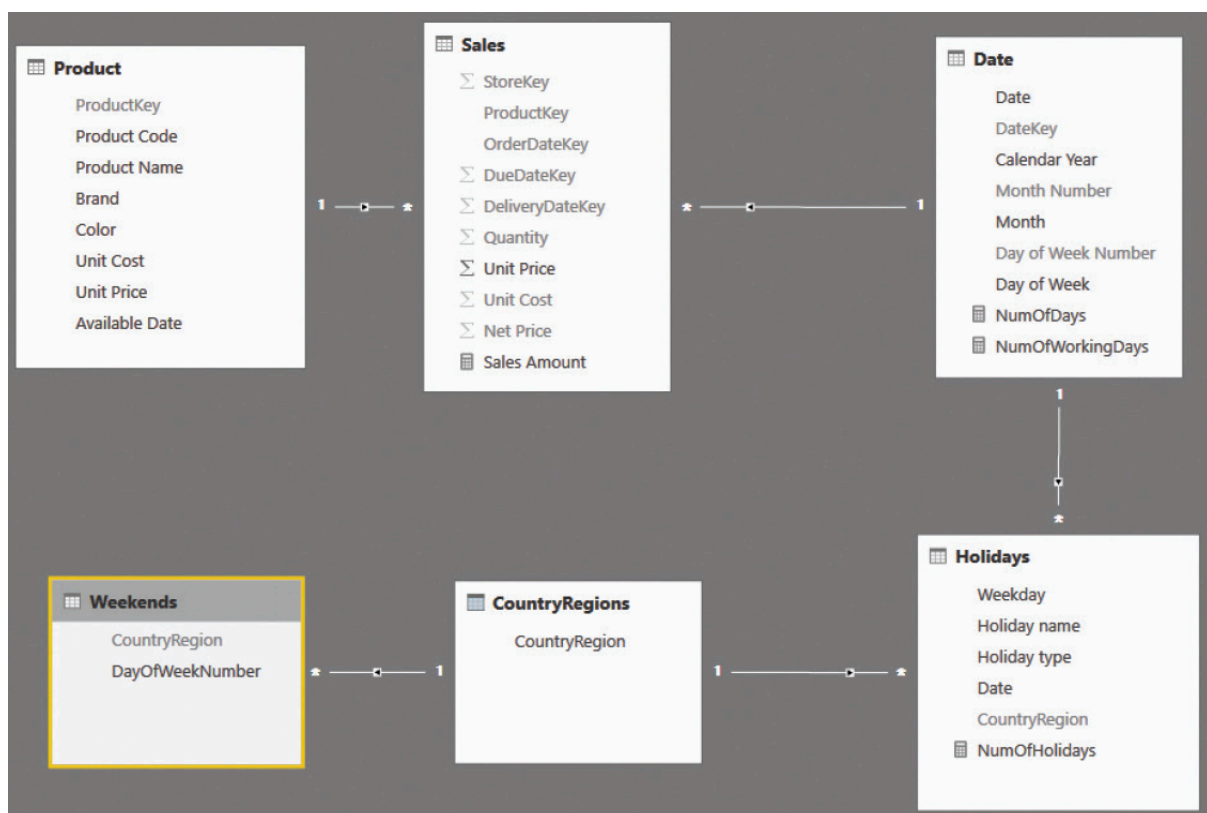
In der Formel sind zwei bemerkenswerte Punkte in Fettdruck hervorgehoben:

Sie müssen prüfen, ob es einen einzelnen Wert für *CountryRegion* gibt, um zu verhindern, dass das berechnete Feld Werte anzeigt, wenn mehrere Länder oder Regionen ausgewählt wurden. Außerdem müssen Sie prüfen, ob die

Tabelle *Holidays* leer ist, denn für Monate ohne Feiertage enthält die Spalte *CountryValue* keine Werte, sodass *HASONEVALUE* den Wert *False* zurückgibt.

Als Filter für *CALCULATE* können Sie die Funktion *EXCEPT* verwenden, um die Kalenderdaten abzurufen, die keine Feiertage sind. Diese Tage werden über *AND* mit den Tagen verknüpft, die nicht zum Wochenende gehören, sodass wir schließlich das korrekte Ergebnis erhalten.

Das Modell ist jedoch noch nicht perfekt. Wir setzen hier voraus, dass das Wochenende immer auf Samstag und Sonntag fällt, doch das ist nicht in allen Ländern der Fall. Wenn Sie diesen Umstand ebenfalls berücksichtigen müssen, wird das Modell noch etwas aufwendiger. Sie brauchen dann eine weitere Tabelle mit den Wochentagen, die in den verschiedenen Ländern jeweils das Wochenende bilden. Da Sie damit zwei verschiedene Tabellen haben, die nach dem Land gefiltert werden, müssen Sie die Länder zu einer eigenen Dimension machen. Das vollständige Modell sehen Sie in Abbildung 4–30.



**Abbildung 4–30** Das vollständige Modell mit einer eigenen Tabelle für die Wochenenden und einer Dimension für die Länder und Regionen

Der Code wird sogar etwas einfacher, auch wenn er etwas schwieriger zu lesen ist:

```

NumOfWorkingDays :=

IF (

    HASONEVALUE ( CountryRegions[CountryRegion] ),

    CALCULATE (

        COUNTRROWS ( 'Date' ),

        EXCEPT (

            VALUES ( 'Date'[Day of Week Number] ),

            VALUES ( Weekends[Day of Week Number] )

        ),

        EXCEPT ( VALUES ( 'Date'[Date] ), VALUES (
        Holidays[Date] ) )

    )

)

```

Diese Formel nutzt das EXCEPT-Muster für die Feiertage nun auch für die Wochenenden. Dabei berücksichtigt sie, welche Wochentage in dem ausgewählten Land nicht als Arbeitstage gelten.



**Wenn das Modell komplizierter wird, müssen Sie auch komplizierteren DAX-Code schreiben. Vor allem aber müssen Sie genau klären, wie die Zahlen berechnet werden. Wenn Sie mehrere Länder berücksichtigen müssen, funktionieren die einfachen Formeln für einzelne Länder nicht mehr. Als Datenmodellierer müssen Sie sich stärker anstrengen, um sinnvolle Formeln zu finden.**

---

## **Besondere Zeiträume im Jahr**

---

Im Rahmen von Zeitinformationen müssen Sie oft besondere Zeiträume im Jahr berücksichtigen. Für die Analyse von Hotelbuchungen kann beispielsweise die Osterzeit von Bedeutung sein, sodass Sie die Zahlen für Ostern in verschiedenen Jahren vergleichen wollen. Das Problem ist jedoch, dass Ostern jedes Jahr auf ein anderes Datum fällt. Um Vergleiche anstellen zu können, müssen Sie daher einen variablen Zeitraum definieren.

Eine weitere häufige Anforderung besteht darin, Berichte oder Dashboards zu erstellen, deren Inhalt automatisch an einem bestimmten Datum aktualisiert wird. Nehmen wir an, Sie haben ein Dashboard, das die Verkäufe des laufenden Monats mit denen des vorherigen vergleicht. Was der »aktuelle Monat« ist, hängt jedoch vom laufenden Datum ab. Heute kann der aktuelle Monat der April sein, aber einen Monat später ist es der Mai. Es wäre jedoch ziemlich lästig, die Filter des Dashboards jeden Monat anzupassen.

Wie bei den Arbeitstagen gibt es Unterschiede in den Datenmodellen je nachdem, ob die zu analysierenden Zeiträume sich überlappen oder nicht.

### **Sich nicht überlappende Zeiträume**

Wenn sich die Zeiträume, die Sie analysieren wollen, nicht überlappen, lässt sich das Datenmodell relativ einfach erstellen. Ähnlich wie bei den Feiertagen in den vorhergehenden Abschnitten brauchen Sie zunächst eine Konfigurationstabelle, in der Sie die zu analysierenden Zeiträume speichern. Als Beispiel haben wir eine Tabelle mit den Oster- und Weihnachtsterminen der Jahre 2008 bis 2010

erstellt. Im Gegensatz zu den Feiertagen geht es hier nicht um einzelne Tagesdaten, sondern um Zeiträume. Die Konfigurationstabelle sehen Sie in Abbildung 4-31.

Date	DaysBefore	DaysAfter	Description
April 12, 2009	4	3	Easter
March 23, 2008	4	3	Easter
April 4, 2010	4	3	Easter
December 25, 2009	2	1	Christmas
December 25, 2008	2	1	Christmas
December 25, 2010	2	1	Christmas

**Abbildung 4-31** Definition der besonderen Zeiträume in der Tabelle *SpecialPeriods*

Jeder der Zeiträume beginnt einige Tage vor einem gegebenen Datum und endet einige Tage danach. Auch wenn die Tabelle *SpecialPeriods* ein Datum als Primärschlüssel enthält, wäre es nicht sinnvoll, eine Beziehung darüber aufzubauen. Die einzig relevante Information in der Tabelle ist der Name des Zeitraums, den wir analysieren würden. Daher ist es besser, die Bezeichnung des Zeitraums in eine berechnete Spalte in der Tabelle *Date* zu denormalisieren. Das erledigen Sie mit dem folgenden Code:

```
'Date'[SpecialPeriod] =
CALCULATE (
    VALUES ( SpecialPeriods[Description] ),
    FILTER (
        SpecialPeriods,
        AND (
```

```
SpecialPeriods[Date] -  
SpecialPeriods[DaysBefore] <= 'Date'[Date],
```

```
SpecialPeriods[Date] +  
SpecialPeriods[DaysAfter] > 'Date'[Date]
```

)

)

)

Diese Spalte enthält den Namen eines der besonderen Zeiträume, wenn das aktuelle Datum zwischen die beiden folgenden Zeitpunkte fällt:

Das Datum des Zeitraums abzüglich der Anzahl der Tage davor

Dasselbe Datum plus die Anzahl der Tage danach

In Abbildung 4–32 sehen Sie das Ergebnis dieser berechneten Spalte für Ostern 2008.

Diese Spalte dient als Filter für verschiedene Zeiträume in verschiedenen Jahren. Damit ist es möglich, etwa die Verkäufe in einem vergleichbaren Zeitraum mehrerer Jahre zu vergleichen, ohne sich Gedanken darüber zu machen, wann dieser Zeitraum jeweils aufgetreten ist. Das können Sie in Abbildung 4–33 erkennen.

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week	SpecialPeriod
3/11/2008	20080311	CY 2008	3	March	3	Tuesday	
3/12/2008	20080312	CY 2008	3	March	4	Wednesday	
3/13/2008	20080313	CY 2008	3	March	5	Thursday	
3/14/2008	20080314	CY 2008	3	March	6	Friday	
3/15/2008	20080315	CY 2008	3	March	7	Saturday	
3/16/2008	20080316	CY 2008	3	March	1	Sunday	
3/17/2008	20080317	CY 2008	3	March	2	Monday	
3/18/2008	20080318	CY 2008	3	March	3	Tuesday	
3/19/2008	20080319	CY 2008	3	March	4	Wednesday	Easter
3/20/2008	20080320	CY 2008	3	March	5	Thursday	Easter
3/21/2008	20080321	CY 2008	3	March	6	Friday	Easter
3/22/2008	20080322	CY 2008	3	March	7	Saturday	Easter
3/23/2008	20080323	CY 2008	3	March	1	Sunday	Easter
3/24/2008	20080324	CY 2008	3	March	2	Monday	Easter
3/25/2008	20080325	CY 2008	3	March	3	Tuesday	Easter
3/26/2008	20080326	CY 2008	3	March	4	Wednesday	
3/27/2008	20080327	CY 2008	3	March	5	Thursday	

**Abbildung 4-32** Fällt ein Datum in einen der besonderen Zeiträume, wird es mit der Bezeichnung dieses Zeitraums gekennzeichnet.

SpecialPeriod	CY 2008	CY 2009	Total
Christmas	6,940.20	30,625.28	<b>37,565.48</b>
Easter	18,426.43	32,875.65	<b>51,302.07</b>
<b>Total</b>	<b>25,366.63</b>	<b>63,500.93</b>	<b>88,867.55</b>

**Abbildung 4-33** Dieser Bericht zeigt die Verkäufe für Weihnachten und Ostern in den Jahren 2008 und 2009.

Diese Technik funktioniert sehr gut und lässt sich einfach einrichten, allerdings weist sie eine bedeutende Einschränkung auf: Die Zeiträume dürfen sich nicht überlappen. Wenn Sie versuchen, sich überlappende Zeiträume in der Konfigurationstabelle zu speichern, gibt die Spalte eine Fehlermeldung für alle Zeilen an, die zu mehreren Zeiträumen gehören. In Situationen, in denen diese Einschränkung keine Rolle spielt, ist diese Vorgehensweise jedoch die einfachste Möglichkeit zur Handhabung besonderer Zeiträume. Wie Sie mit sich überlappenden Zeiträumen umgehen, erfahren Sie im übernächsten Abschnitt, »Sich überlappende Zeiträume«.

## Zeiträume relativ zu heute

Im vorherigen Abschnitt haben Sie gesehen, wie Sie sich nicht überlappende Zeiträume handhaben, indem Sie die entsprechenden Angaben in einer Konfigurationstabelle speichern. Eine ähnliche Technik können Sie anwenden, um Berichte zu erstellen, deren Inhalte dynamisch aktualisiert werden. Nehmen wir an, Sie wollen ein Dashboard wie das aus Abbildung 4-34 erstellen, das die

Verkäufe für verschiedene Marken über verschiedene Zeiträume hinweg anzeigt und ein Diagramm ausgibt, das die Verkäufe von heute mit denen von gestern vergleicht.



**Abbildung 4-34** Der Bericht enthält ein Diagramm, das die Verkäufe von heute mit denen von gestern vergleicht.

Was »heute« bedeutet, hängt davon ab, wann der Bericht zuletzt aktualisiert wurde. Selbstverständlich wollen Sie das Datum nicht direkt in die Formeln hineinschreiben. Stattdessen soll das Modell bei jeder Aktualisierung automatisch prüfen, was das letzte verfügbare Datum ist, und seinen Inhalt entsprechend anpassen. Dazu können Sie eine Variante des vorherigen Datenmodells verwenden, wobei die Zeiträume diesmal jedoch dynamisch berechnet werden.

Als Erstes müssen Sie dazu die Konfigurationstabelle aus Abbildung 4-35 aufstellen. Darin sind die Bezeichnungen der Zeiträume und die Abstände von heute in Tagen enthalten.

Description	DaysTo	DaysFrom	Code
Today	0	-1	1
Yesterday	1	0	2
Last 7 days	7	1	3
Last 30 days	30	7	4
Older	99999999	30	5

**Abbildung 4-35** Die Konfigurationstabelle *RelativePeriods* zeigt die besonderen Zeiträume relativ zum heutigen Tag.

Für jeden Zeitraum sind die Anzahl der Tage vor dem heutigen Datum, eine Bezeichnung und ein Code angegeben. Die Tage, die relativ zum aktuellen Datum zwischen *DatesFrom* und *DaysTo* fallen, werden mit der Bezeichnung markiert. Der Code dient hauptsächlich zur Sortierung. Aus dieser Tabelle können Sie den Code (zum Sortieren) und die Bezeichnung abrufen, um alle

Kalenderdaten zu markieren, die zu dem entsprechenden Zeitraum gehören. Das lässt sich leicht durch zwei berechnete Spalten in *Date* erreichen. Die erste bestimmt den Code des relativen Zeitraums:

```
'Date'[RelPeriodCode] =  
  
VAR LastSalesDateKey =  
  
    MAX ( Sales[OrderDateKey] )  
  
VAR LastSaleDate =  
  
    LOOKUPVALUE( 'Date'[Date], 'Date'[DateKey],  
                LastSalesDateKey )  
  
VAR DeltaDays =  
  
    INT ( LastSaleDate - 'Date'[Date] )  
  
VAR ValidPeriod =  
  
    CALCULATETABLE(  
  
        RelativePeriods,  
  
        RelativePeriods[DaysTo] >= DeltaDays,  
  
        RelativePeriods[DaysFrom] < DeltaDays  
  
    )
```

RETURN

```
CALCULATE ( VALUES ( RelativePeriods[Code] ),  
ValidPeriod )
```

Dieser Code nutzt für alle Schritte Variablen. Am Anfang ruft er mit dem letzten Eintrag der Spalte *OrderDateKey* in *Sales* den Schlüssel für das letzte verfügbare Datum ab. Dieses Datum wird als »heute« angesehen. Anschließend bestimmt er mit LOOKUPVALUE das Datum, das zu dem Schlüssel gehört. *DeltaDays* gibt den Abstand zwischen heute und dem vorliegenden Datum an. All diese Werte werden schließlich in CALCULATETABLE genutzt, um die Zeile der Tabelle *RelativePeriods* zu bestimmen, bei der *DeltaDays* zwischen *DaysFrom* und *DaysTo* liegt. Diese Zeile wird dann in der Variablen *ValidPeriod* gespeichert.

Das Ergebnis dieser Formel ist der Code für den relativen Zeitraum, zu dem das gegebene Datum gehört. Mit dieser berechneten Spalte können Sie die Bezeichnung des Zeitraums nun wie folgt ermitteln:

```
'Date'[RelPeriod] =
```

```
VAR RelPeriod =
```

```
LOOKUPVALUE(
```

```
    RelativePeriods[Description],
```

```
    RelativePeriods[Code],
```

```
    'Date'[RelPeriodCode]
```

```
)
```

## RETURN

IF ( ISBLANK ( RelPeriod ), "Future", RelPeriod )

Die beiden Spalten *RelPeriodCode* und *RelPeriod* in der Tabelle *Date* sehen Sie in Abbildung 4-36.

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week	RelPeriodCode	RelPeriod
15/8/08	20080815	CY 2008	8	August	6	Friday		Future
14/8/08	20080814	CY 2008	8	August	5	Thursday		Future
13/8/08	20080813	CY 2008	8	August	4	Wednesday		Future
12/8/08	20080812	CY 2008	8	August	3	Tuesday		Future
11/8/08	20080811	CY 2008	8	August	2	Monday	1	Today
10/8/08	20080810	CY 2008	8	August	1	Sunday	2	Yesterday
9/8/08	20080809	CY 2008	8	August	7	Saturday	3	Last 7 days
8/8/08	20080808	CY 2008	8	August	6	Friday	3	Last 7 days
7/8/08	20080807	CY 2008	8	August	5	Thursday	3	Last 7 days
6/8/08	20080806	CY 2008	8	August	4	Wednesday	3	Last 7 days
5/8/08	20080805	CY 2008	8	August	3	Tuesday	3	Last 7 days
4/8/08	20080804	CY 2008	8	August	2	Monday	3	Last 7 days
3/8/08	20080803	CY 2008	8	August	1	Sunday	4	Last 30 days
2/8/08	20080802	CY 2008	8	August	7	Saturday	4	Last 30 days
1/8/08	20080801	CY 2008	8	August	6	Friday	4	Last 30 days
31/7/08	20080731	CY 2008	7	July	5	Thursday	4	Last 30 days

**Abbildung 4-36** Die beiden letzten Spalten werden mit den in den vorstehenden Absätzen beschriebenen Formeln berechnet.

Da es sich um berechnete Spalten handelt, werden sie bei jeder Aktualisierung des Datenmodells neu berechnet. Dadurch ändern sich entsprechend die Bezeichnungen, die den Tagen zugewiesen werden. Den Bericht müssen Sie nicht aktualisieren, da er stets den zuletzt verarbeiteten Tag als heute anzeigt, den Tag davor als gestern usw.

## Sich überlappende Zeiträume

Die Techniken aus den vorherigen Abschnitten funktionieren tadellos, weisen aber eine bedeutende Einschränkung auf: Die Zeiträume dürfen sich nicht überlappen. Das liegt daran, dass Sie das Attribut zur Definition des Zeitraums in einer berechneten Spalte speichern und der Spalte nur ein Wert zugewiesen werden kann.

Es gibt jedoch Situationen, in denen diese Vorgehensweise nicht ausreicht. Nehmen wir an, Sie haben Produktkategorien, die jeweils in unterschiedlichen

Zeiträumen im Jahr zum Verkauf angeboten werden. In einem Zeitraum können dabei mehrere Produktkategorien verfügbar sein, es kann aber auch sein, dass eine Kategorie in mehreren Zeiträumen zum Verkauf steht. In einer solchen Situation können Sie die Verkaufszeiträume nicht in *Products* oder *Date* speichern.

Wenn viele Zeilen (Kategorien) mit vielen anderen Zeilen (Kalenderdaten) in Beziehung stehen, haben wir ein *m:n-Modell*. Solche Modelle lassen sich nicht so leicht handhaben, sind aber äußerst nützlich für die Analyse und eine Beschreibung wert. Eine ausführlichere Erörterung von m:n-Modellen finden Sie in Kapitel 8. In diesem Abschnitt wollen wir lediglich zeigen, dass der Code schwieriger zu schreiben ist, wenn wir es mit m:n-Beziehungen zu tun haben.

Als Beispiel verwenden wir die Konfigurationstabelle *Discounts*, die Sie in Abbildung 4-37 sehen.

Description	DateStart	DateEnd	Category
January Sales	1/1/2007	1/31/2007	Computers
January Sales	1/1/2008	1/31/2008	Computers
Start with Audio	1/1/2007	1/15/2007	Audio
Start with Audio	1/1/2008	1/15/2008	Audio
Summer Music	8/1/2007	8/15/2007	Audio
Summer Music	8/1/2008	8/15/2008	Audio
Holidays calls home	7/15/2007	8/15/2007	Cell phones
Holidays calls home	7/15/2007	8/15/2007	Cell phones

**Abbildung 4-37** Die Konfigurationstabelle *Discounts* mit verschiedenen Verkaufszeiträumen für verschiedene Kategorien

Wie Sie sehen, werden in der ersten Januarwoche von 2007 und 2008 mehrere Kategorien zum Verkauf angeboten (Computer und Audio). Das Gleiche gilt für die ersten beiden Augustwochen (hier mit Audiogeräten und Mobiltelefonen). In einer solchen Situation können Sie keine Beziehungen mehr verwenden, sondern müssen DAX-Code schreiben, der den aktuellen Filter des Verkaufszeitraums mit dem bereits vorhandenen Filter in *Sales* kombiniert. Das erreichen Sie mit der folgenden Formel:

`SalesInPeriod :=`

```

SUMX (
    Discounts,
    CALCULATE (
        [Sales Amount],
        INTERSECT (
            VALUES ( 'Date'[Date] ),
            DATESBETWEEN ( 'Date'[Date],
                Discounts[DateStart], Discounts[DateEnd] )
        ),
        INTERSECT (
            VALUES ( 'Product'[Category] ),
            CALCULATETABLE ( VALUES (
                Discounts[Category] ) )
        )
    )
)

```

Mit dieser Formel können Sie den Bericht aus Abbildung 4–38 aufstellen.

Category	Description	CY 2007	CY 2008	Total
Computers	January Sales	\$9,842.11	\$20,450.43	\$30,292.54
	<b>Total</b>	<b>\$9,842.11</b>	<b>\$20,450.43</b>	<b>\$30,292.54</b>
Cell phones	Holidays calls home	\$1,556.69		\$1,556.69
	<b>Total</b>	<b>\$1,556.69</b>		<b>\$1,556.69</b>
Audio	Start with Audio	\$492.22		\$492.22
	Summer Music	\$35.07		\$35.07
	<b>Total</b>	<b>\$527.30</b>		<b>\$527.30</b>
<b>Total</b>		<b>\$11,926.09</b>	<b>\$20,450.43</b>	<b>\$32,376.53</b>

**Abbildung 4–38** Bei der Verwendung sich überlappender Zeiträume können Sie verschiedene Zeiträume im selben Jahr untersuchen.

Der Bericht in Abbildung 4–38 zeigt die Verkäufe in verschiedenen Kategorien über mehrere Jahre, auch mit sich überlappenden Zeiträumen. In diesem Fall ist das Modell relativ einfach geblieben, da es keine möglichen Änderungen gab, mit denen wir für einfacheren Code hätten sorgen können. In Kapitel 7 werden Sie weitere Beispiele dieser Art finden, aber auch andere Arten von Datenmodellen, die es erlauben, einfacheren (und möglicherweise schnelleren) Code zu schreiben. m:n-Beziehungen sind im Allgemeinen sehr leistungsfähig und leicht verwendbar, aber Code dafür zu schreiben ist manchmal (wie hier) schwierig.

Wir zeigen Ihnen dieses Beispiel nicht, um Sie abzuschrecken oder um Ihnen eine Situation vorzuführen, in der das Modell keinen einfacheren Code zulässt. Es ist einfach so, dass Sie früher oder später auch anspruchsvollen DAX-Code schreiben müssen, wenn Sie anspruchsvolle Berichte aufstellen wollen.

## Wochenkalender

---

Wenn Sie mit Standardkalendern arbeiten, können Sie berechnete Felder wie aufgelaufene Beträge seit Jahres- oder Monatsbeginn oder Zeiträume im Jahresvergleich leicht aufstellen, da DAX einen Satz vordefinierter Funktionen dafür bietet. Bei anderen Arten von Kalendern werden solche Berechnungen jedoch etwas schwieriger.

Das betrifft alle Kalender, die nicht der üblichen Unterteilung in zwölf Monate mit einer eigenen Tageszählung pro Monat folgen. Beispielsweise arbeiten viele Unternehmen mit Wochen statt mit Monaten. Leider lassen sich Wochen nicht einfach zu Monaten oder Jahren aggregieren, denn die einzelnen Monate und Jahre setzen sich aus unterschiedlich vielen Wochen zusammen. Obwohl es verschiedene übliche Techniken zur Handhabung von Jahren auf der Grundlage von Wochen gibt, ist keine davon ein Standard, der sich in eine DAX-Funktion übertragen ließe. Daher bietet DAX keine Funktionen für solche abweichenden Kalender an. Wenn Sie damit arbeiten müssen, sind Sie auf sich allein gestellt.

Glücklicherweise gibt es einige Modellierungstechniken, mit denen Sie Zeitinformationsberechnungen auch in anderen als Standardkalendern durchführen können. In diesem Abschnitt werden wir jedoch nicht alle davon behandeln, sondern wollen Ihnen einfach einige Beispiele zeigen, die Sie wahrscheinlich noch an Ihre eigenen Bedürfnisse anpassen müssen. Wenn Sie an weiteren Informationen zu diesem Thema interessiert sind, schauen Sie sich die vollständige Darstellung auf <http://www.daxpatterns.com/time-patterns> an.

Als Beispiel erläutern wir im Folgenden Berechnungen in einem Wochenkalender nach ISO 8601. Wenn Sie mehr über diese Art von Wochenangaben erfahren wollen, schlagen Sie auf [https://en.wikipedia.org/wiki/ISO\\_week\\_date](https://en.wikipedia.org/wiki/ISO_week_date) nach.

Der erste Schritt besteht darin, eine ISO-Kalendertabelle zu erstellen. Dazu gibt es mehrere Möglichkeiten. Wahrscheinlich haben Sie bereits einen wohldefinierten ISO-Kalender in Ihrer Datenbank. In diesem Beispiel jedoch erstellen wir selbst einen ISO-Kalender mithilfe von DAX-Standardcode und einer Nachschlagetabelle, denn dabei können Sie noch einige weitere Modellierungstricks lernen.

Unser Kalender beruht auf Wochen. Eine Woche beginnt immer am Montag, und ein Jahr beginnt immer mit dem Anfang der ersten Woche. Daher kann es durchaus am 29. Dezember des kalendarischen Vorjahrs oder erst am 2. Januar des laufenden Kalenderjahrs beginnen. Dazu fügen Sie einer Standardkalendertabelle berechnete Spalten hinzu, um die ISO-Wochenummer und die ISO-Jahresnummer zu bestimmen. Mit den folgenden Definitionen erstellen Sie die Tabelle aus Abbildung 4–39 mit den Spalten *Calendar Week*, *ISO Week* und *ISO Year*:

```
'Date'[Calendar Week] = WEEKNUM ( 'Date'[Date], 2 )
```

```
'Date'[ISO Week] = WEEKNUM ( 'Date'[Date], 21 )
```

```
'Date'[ISO Year] =
```

```
CONCATENATE (
```

```
    "ISO ",
```

```
    IF (
```

```
        AND ( 'Date'[ISO Week] < 5, 'Date'[Calendar  
Week] > 50 ),
```

```
        YEAR ( 'Date'[Date] ) + 1,
```

```
        IF (
```

```
            AND ( 'Date'[ISO Week] > 50, 'Date'[Calendar  
Week] < 5 ),
```

```
            YEAR ( 'Date'[Date] ) - 1,
```

```
            YEAR ( 'Date'[Date] )
```

```
        )
```

```
    )
```

)

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week	ISO Week	Calendar Week	ISO Year
1/1/2005	20050101	CY 2005	1	January	7	Saturday	53	1	ISO 2004
1/2/2005	20050102	CY 2005	1	January	1	Sunday	53	1	ISO 2004
1/3/2005	20050103	CY 2005	1	January	2	Monday	1	2	ISO 2005
1/4/2005	20050104	CY 2005	1	January	3	Tuesday	1	2	ISO 2005
1/5/2005	20050105	CY 2005	1	January	4	Wednesday	1	2	ISO 2005
1/6/2005	20050106	CY 2005	1	January	5	Thursday	1	2	ISO 2005
1/7/2005	20050107	CY 2005	1	January	6	Friday	1	2	ISO 2005
1/8/2005	20050108	CY 2005	1	January	7	Saturday	1	2	ISO 2005
1/9/2005	20050109	CY 2005	1	January	1	Sunday	1	2	ISO 2005
1/10/2005	20050110	CY 2005	1	January	2	Monday	2	3	ISO 2005
1/11/2005	20050111	CY 2005	1	January	3	Tuesday	2	3	ISO 2005
1/12/2005	20050112	CY 2005	1	January	4	Wednesday	2	3	ISO 2005
1/13/2005	20050113	CY 2005	1	January	5	Thursday	2	3	ISO 2005
1/14/2005	20050114	CY 2005	1	January	6	Friday	2	3	ISO 2005
1/15/2005	20050115	CY 2005	1	January	7	Saturday	2	3	ISO 2005
1/16/2005	20050116	CY 2005	1	January	1	Sunday	2	3	ISO 2005

**Abbildung 4-39** Da das ISO-Jahr immer an einem Montag beginnt, kann es vom Kalenderjahr abweichen.

Wochen und Monate lassen sich leicht in einfachen berechneten Spalten bestimmen, aber der ISO-Monat erfordert etwas mehr Arbeit. Im ISO-Standard gibt es verschiedene Möglichkeiten, um die Monatsnummer zu berechnen. Eine besteht darin, die vier Quartale aufzuteilen. Jedes Quartal besteht aus drei Monaten, wobei die Monate nach einer der drei möglichen Gruppierungen 445, 454 oder 544 aufgebaut sind. Die Ziffern stehen dabei jeweils für die Anzahl der Wochen in jedem Monat. So umfassen bei 445 die ersten beiden Monate im Quartal jeweils vier Wochen und der dritte fünf. Anstatt nach einer komplizierten mathematischen Formel zu suchen, um die Monate zu berechnen, zu denen eine Woche je nach verwendetem Standard gehört, ist es einfacher, eine Nachschlagetabelle wie die aus Abbildung 4-40 aufzustellen.

Week	Period445	Period454	Period544
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	2	2	1
6	2	2	2
7	2	2	2
8	2	2	2
9	3	2	2
10	3	3	3
11	3	3	3
12	3	3	3
13	3	3	3
14	4	4	4
15	4	4	4
16	4	4	4
17	4	4	4
18	5	5	4
19	5	5	5
20	5	5	5

**Abbildung 4-40** Die Nachschlagetabelle *Weeks To Months* ordnet die Wochennummern den Monatsnummern zu. Dabei werden für die drei verschiedenen Verteilungsmuster drei verschiedene Spalten verwendet.

Um in dieser Tabelle nachzuschlagen, verwenden Sie den folgenden Code mit der Funktion LOOKUPVALUE:

'Date' [ISO Month] =

CONCATENATE

```

"ISO M",

RIGHT (

    CONCATENATE (

        "00",

        LOOKUPVALUE (

            'Weks To Months' [Period445],

            'Weks To Months' [Week],

            'Date' [ISO Week]

        ),

        2

    )

)

```

Die resultierende Tabelle enthält das Jahr und den Monat (siehe Abbildung 4-41).

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week	ISO Week	Calendar Week	ISO Year	ISO Month
1/1/2005	20050101	CY 2005	1	January	7	Saturday	53	1	ISO 2004	ISO M12
1/2/2005	20050102	CY 2005	1	January	1	Sunday	53	1	ISO 2004	ISO M12
1/3/2005	20050103	CY 2005	1	January	2	Monday	1	2	ISO 2005	ISO M01
1/4/2005	20050104	CY 2005	1	January	3	Tuesday	1	2	ISO 2005	ISO M01
1/5/2005	20050105	CY 2005	1	January	4	Wednesday	1	2	ISO 2005	ISO M01
1/6/2005	20050106	CY 2005	1	January	5	Thursday	1	2	ISO 2005	ISO M01
1/7/2005	20050107	CY 2005	1	January	6	Friday	1	2	ISO 2005	ISO M01
1/8/2005	20050108	CY 2005	1	January	7	Saturday	1	2	ISO 2005	ISO M01
1/9/2005	20050109	CY 2005	1	January	1	Sunday	1	2	ISO 2005	ISO M01
1/10/2005	20050110	CY 2005	1	January	2	Monday	2	3	ISO 2005	ISO M01
1/11/2005	20050111	CY 2005	1	January	3	Tuesday	2	3	ISO 2005	ISO M01
1/12/2005	20050112	CY 2005	1	January	4	Wednesday	2	3	ISO 2005	ISO M01
1/13/2005	20050113	CY 2005	1	January	5	Thursday	2	3	ISO 2005	ISO M01
1/14/2005	20050114	CY 2005	1	January	6	Friday	2	3	ISO 2005	ISO M01
1/15/2005	20050115	CY 2005	1	January	7	Saturday	2	3	ISO 2005	ISO M01
1/16/2005	20050116	CY 2005	1	January	1	Sunday	2	3	ISO 2005	ISO M01
1/17/2005	20050117	CY 2005	1	January	2	Monday	3	4	ISO 2005	ISO M01

**Abbildung 4-41** Der ISO-Monat lässt sich leicht mit einer Nachschlagetabelle bestimmen.

Da Sie nun alle erforderlichen Spalten haben, können Sie eine Hierarchie aufbauen und das Modell durch eine Aufteilung nach ISO-Jahren, -Monaten und -Wochen untersuchen. Allerdings sind die Bestimmung von Werten wie den seit Jahres- oder Monatsanfang aufgelaufenen Beträgen und andere Zeitinformationsberechnungen in einem solchen Kalender eine ziemliche Herausforderung. Die integrierten DAX-Funktionen sind nur für gregorianische Standardkalender gedacht und lassen sich nicht auf andere Kalender anwenden.

Daher müssen Sie Zeitinformationsberechnungen auf andere Weise durchführen, also ohne auf vordefinierte Funktionen zurückzugreifen. Um beispielsweise die aufgelaufenen Beträge seit ISO-Jahresbeginn zu ermitteln, können Sie das folgende berechnete Feld verwenden:

```
Sales ISO YTD :=
```

```
IF (
```

```
    HASONEVALUE ( 'Date'[ISO Year] ),
```

```
    CALCULATE (
```

```
        [Sales Amount],
```

```

ALL ( 'Date' ),

FILTER ( ALL ( 'Date'[Date] ), 'Date'[Date] <=
MAX ( 'Date'[Date] ) ),

VALUES ( 'Date'[ISO Year] )

)

)

```

Das Herz dieses berechneten Feldes bilden die Filter, die Sie auf die Kalendertabelle anwenden müssen, um die richtigen Datumsangaben für die Bestimmung der aufgelaufenen Beträge seit Jahresbeginn zu finden. Abbildung 4-42 zeigt das Ergebnis.

ISO Year	ISO Month	Sales Amount	Sales ISO YTD
ISO 2007	ISO M01	97,104.79	97,104.79
ISO 2007	ISO M02	97,133.79	194,238.58
ISO 2007	ISO M03	144,911.08	339,149.66
ISO 2007	ISO M04	106,741.46	445,891.12
ISO 2007	ISO M05	118,319.01	564,210.13
ISO 2007	ISO M06	131,504.47	695,714.60
ISO 2007	ISO M07	115,924.73	811,639.33
ISO 2007	ISO M08	96,647.19	908,286.52
ISO 2007	ISO M09	169,319.47	1,077,606.00
ISO 2007	ISO M10	91,261.55	1,168,867.54
ISO 2007	ISO M11	117,902.14	1,286,769.69
ISO 2007	ISO M12	169,009.75	1,455,779.43
ISO 2008	ISO M01	67,106.83	67,106.83
ISO 2008	ISO M02	49,376.15	116,482.98

**Abbildung 4-42** Das berechnete Feld *ISO YTD* berechnet die aufgelaufenen Beträge seit Jahresbeginn für den ISO-Kalender.

Es ist tatsächlich sehr einfach, die aufgelaufenen Beträge seit Monats- oder Quartalsbeginn oder berechnete Felder zu bestimmen, die einem ähnlichen Muster folgen. Komplizierter wird es jedoch bei Berechnungen, die beispielsweise identische Zeiträume über mehrere Jahre hinweg vergleichen. Da Sie die Funktion SAMEPERIODLASTYEAR nicht nutzen können, müssen Sie mehr Arbeit in das Datenmodell und in den DAX-Code investieren.

Um den identischen Vorjahreszeitraum zu berechnen, müssen Sie das Datum bestimmen, das zurzeit im Filterkontext ausgewählt ist, und die gleiche Menge von Datumsangaben im Vorjahr finden. Dazu können Sie nicht die Spalte *Date* verwenden, da das ISO-Datum eine ganz andere Struktur aufweist als das Kalenderdatum. Daher müssen Sie als Erstes eine neue Spalte mit der ISO-Tagesnummer für das Jahr in die Kalendertabelle aufnehmen. Das können Sie ganz leicht mit der folgenden berechneten Spalte erreichen:

$$\text{Date[ISO Day Number]} = ( \text{'Date'[ISO Week]} - 1 ) * 7 + \text{WEEKDAY}( \text{'Date'[Date]}, 2 )$$

Die resultierende Spalte sehen Sie in Abbildung 4-43.

Date	DateKey	Calendar Year	Month Number	Month	Day of Week Number	Day of Week	ISO Week	ISO Year	ISO Day Number
1/1/05	20050101	CY 2005	1	January	7	Saturday	53	ISO 2004	370
2/1/05	20050102	CY 2005	1	January	1	Sunday	53	ISO 2004	371
3/1/05	20050103	CY 2005	1	January	2	Monday	1	ISO 2005	1
4/1/05	20050104	CY 2005	1	January	3	Tuesday	1	ISO 2005	2
5/1/05	20050105	CY 2005	1	January	4	Wednesday	1	ISO 2005	3
6/1/05	20050106	CY 2005	1	January	5	Thursday	1	ISO 2005	4
7/1/05	20050107	CY 2005	1	January	6	Friday	1	ISO 2005	5
8/1/05	20050108	CY 2005	1	January	7	Saturday	1	ISO 2005	6
9/1/05	20050109	CY 2005	1	January	1	Sunday	1	ISO 2005	7
10/1/05	20050110	CY 2005	1	January	2	Monday	2	ISO 2005	8

**Abbildung 4-43** Die Spalte *ISO Day Number* zeigt die inkrementelle Tagesnummer im Jahr laut ISO-Standard.

Um den identischen Zeitraum im Vorjahr zu finden, können Sie dank dieser Spalte nun den folgenden Code nutzen:

Sales SPLY :=

IF (

```

HASONEVALUE ( 'Date'[ISO Year Number] ),

CALCULATE (

    [Sales Amount],

    ALL ( 'Date' ),

    VALUES ( 'Date'[ISO Day Number] ),

    'Date'[ISO Year Number] = VALUES ( 'Date'[ISO
    Year Number] ) - 1

)

)

```

Die Formel entfernt alle Filter von der Tabelle *Date* und ersetzt sie durch zwei neue Bedingungen:

Die ISO-Jahresnummer muss das aktuelle Jahr minus 1 sein.

Die ISO-Tagesnummer muss identisch sein.

Unabhängig von der Auswahl im aktuellen Filterkontext – ob Tag, Woche oder Monat – gehen Sie dadurch ein Jahr zurück.

In Abbildung 4–44 sehen Sie das berechnete Feld *Sales SPLY* («same period last year») in einem Bericht, der nach Jahr und Monat filtert.

Mit sehr ähnlichen Techniken können Sie auch vergleichbare Berechnungen für den Vormonat oder für das prozentuale Wachstum gegenüber dem identischen Zeitraum im Vorjahr schreiben. In letzterem Fall wird es extrem einfach, die Werte zu berechnen, wenn Sie dem Modell eine Spalte für die ISO-Tagesnummer hinzufügen. Ohne diese Spalte dagegen wird es nahezu unmöglich, eine Formel aufzustellen. Weitere Informationen zu diesem Thema

erhalten Sie auf <http://www.sqlbi.com/articles/week-based-time-intelligence-in-dax>.

ISO Year	ISO Month	Sales Amount	Sales SPLY
ISO 2007	ISO M01	97.104,79	
ISO 2007	ISO M02	97.133,79	
ISO 2007	ISO M03	144.911,08	
ISO 2007	ISO M04	106.741,46	
ISO 2007	ISO M05	118.319,01	
ISO 2007	ISO M06	131.504,47	
ISO 2007	ISO M07	115.924,73	
ISO 2007	ISO M08	96.647,19	
ISO 2007	ISO M09	169.319,47	
ISO 2007	ISO M10	91.261,55	
ISO 2007	ISO M11	117.902,14	
ISO 2007	ISO M12	169.009,75	
ISO 2008	ISO M01	67.106,83	97.104,79
ISO 2008	ISO M02	49.376,15	97.133,79
ISO 2008	ISO M03	72.586,47	144.911,08
ISO 2008	ISO M04	111.633,80	106.741,46
ISO 2008	ISO M05	98.019,09	118.319,01
ISO 2008	ISO M06	125.591,98	131.504,47
ISO 2008	ISO M07	87.074,22	115.924,73
ISO 2008	ISO M08	114.295,00	96.647,19
ISO 2008	ISO M09	137.011,56	169.319,47
ISO 2008	ISO M10	60.111,91	91.261,55
ISO 2008	ISO M11	96.580,97	117.902,14
ISO 2008	ISO M12	101.328,95	169.009,75

**Abbildung 4-44** Das berechnete Feld *Sales SPLY* zeigt für die Monate im ISO-Jahr 2008 jeweils die Verkaufszahlen für die identischen Zeiträume von 2007 an.

## Zusammenfassung

---

Zeitinformationen bilden ein sehr umfangreiches und interessantes Thema. In nahezu jeder BI-Lösung, die Sie schreiben müssen, wird es wenigstens einen Teil geben, der damit zu tun hat. In diesem Kapitel haben wir die folgenden wichtigen Punkte vorgestellt:

Die meisten (wenn nicht gar alle) Zeitinformationsberechnungen erfordern das Vorhandensein einer Datumstabelle in dem Modell.

Beim Erstellen einer Kalendertabelle müssen Sie Einzelheiten wie die Reihenfolge der Monate beachten.

Enthält das Modell mehrere Datumsangaben, so bedeutet das nicht, dass Sie auch mehrere Datumstabellen brauchen. Die Verwendung einer einzigen Datumstabelle im Modell macht alle Berechnungen viel einfacher. Wenn Sie mehrere Datumsangaben brauchen, müssen Sie die Datumstabelle wahrscheinlich mehrfach laden.

Aus Gründen der Leistung und der Modellierung müssen Datum und Uhrzeit voneinander getrennt werden.

Die weiteren Abschnitte dieses Kapitels waren verschiedenen Situationen gewidmet, auf die Sie bei Zeitberechnungen stoßen können: Ermitteln der Arbeitstage in einem oder mehreren Ländern, Berechnen der Werte für besondere Zeiträume im Jahr mithilfe neuer Spalten in der Datumstabelle oder neuen Tabellen im Modell sowie Berechnungen in ISO-Kalendern.

Aufgrund der Vielzahl der Möglichkeiten bei Zeitinformationsberechnungen wird wahrscheinlich keines der Beispiele genau dem entsprechen, was Sie für Ihre Arbeit benötigen. Allerdings können Sie diese Beispiele als Anregung für Ihre eigenen Berechnungen nutzen. Gewöhnlich müssen Sie dabei zusätzliche Spalten in die Datumstabelle aufnehmen und DAX-Code mittlerer Komplexität schreiben.

# Index

1:n-Beziehungen

Definition 10

Faktentabellen 50

## A

ABC-Analyse 210

Abgeleitete Snapshots

Definition 122

Überblick 128

Additive berechnete Felder

Faktentabellen 54

Snapshots aggregieren 124

Überblick 241

Aggregation

additive berechnete Felder 124

Dauer 141

halbadditive berechnete Felder 124

Header/Detail-Tabellen 26

Snapshots 122

ALL 42

Anzeigen

Beziehungsdiagramm 8

Power Pivot 10

Werte 196

Arbeitstage 77, 80

Ausblenden

Beziehungsdiagramm 8

Power Pivot 10

Werte 196

## Automatische Zeitdimensionen

erstellen 62

Excel 62

Power BI Desktop 64

## **B**

### Berechnete Felder

additiv 124, 241

halbadditiv 124, 242

m:n-Beziehungen 170

nicht additiv 171, 241

Snapshots aggregieren 122

### Berechnete Spalten 210

#### Berechnungen

berechnete Spalten 210

CALCULATE 39, 46, 83, 135, 169, 196, 207, 236

CALCULATETABLE 87, 135

Zeitinformationen 73

#### Berichtswährungen

eine Berichtswährung, mehrere Quellwährungen 218

mehrere Berichtswährungen, eine Quellwährung 222

mehrere Berichtswährungen, mehrere Quellwährungen 227

#### Beziehungen

1:n-Beziehungen 10

Beziehungsdiagramm 8

bidirektionale Filterung 194

Datenmodelle 8

denormalisieren 13, 19

Dimensionen 17

Faktentabellen 17, 49

Filter verschieben 192

Fremdschlüssel 10

Mehrdeutigkeit 18, 45

mehrere Tabellen 12

- normalisieren 13
- OLTP 14
- Primärschlüssel 9
- Quelltabelle 9
- Segmentierung 203
- Tabellen 8
- Überblick 233
- vorhersagen 189
- Werte ausblenden 196
- Ziel 10
- Zuordnungsfaktor 200

Beziehungsdiagramm 8

Bidirektionale Filterung

- CROSSFILTER 45, 56, 106, 169, 173, 182, 236

- Faktentabelle 43

- Granularität 193

- Header/Detail-Tabellen 28

- m:n-Beziehungen 168

- Überblick 234

Brückentabelle 56

- Definition 238

- Faktentabelle als Brückentabelle 182

- filtern 176

- kaskadierende m:n-Beziehungen 173

- Leistung 183

- Materialisierung 181

- mehrere 173

- m:n-Beziehungen 167, 181

- prozentuale Zuordnungsfaktoren 179

- Super-Brückentabelle 175

- Überblick 240

## **C**

CALCULATE 39, 46, 83, 135, 169, 196, 207, 236

CALCULATETABLE 87, 135  
CLOSINGBALANCELASTQUARTER 242  
CONTAINS 192  
COUNTRROWS 81, 105, 236  
CROSSFILTER 45, 56, 107, 169, 173, 182, 236

## **D**

### Datenmodelle

- 1:n-Beziehungen 10
- ABC-Analyse 210
- Aggregation 26
- berechnete Spalten 210
- Beziehungen über mehrere Spalten 203
- bidirektionale Filterung 28
- denormalisieren 13, 19
- Dimensionen 25
- dynamische Segmentierung 208
- Einzeltabellen 2
- Faktentabellen 49
- Fremdschlüssel 10
- Granularität 30
- Hierarchien 25
- normalisieren 13
- OLTP 14
- Primärschlüssel 9
- Quelltabellen 9
- reduzieren 33
- Schneeflocke 19, 238
- Segmentierung 203
- statische Segmentierung 206
- Sternschemata 15, 238
- Ziel 10

Datenschnitte 134

Datentypen 233

DATESINPERIOD 74

DATESYTD 73, 75, 242

## Datum

Aggregation bei Zeiträumen 141

Arbeitstage 77, 80

CLOSINGBALANCELASTQUARTER 242

DATESINPERIOD 74

DATESYTD 73, 75, 242

Datumsdimensionen 59

Dauer 139

Formel-Engine für Zeiträume 154

Geschäftskalender 75

LASTDATE 125, 154, 242

LASTDAY 74

laufende Ereignisse 149

mehrere Datumsangaben 143

mehrere Datumsdimensionen 65

nicht überlappende Zeiträume 85

PARALLELPERIOD 74

SAMEPERIODLASTYEAR 74, 94

TOTALYTD 242

Trennung von Datum und Uhrzeit 71, 143

überlappende Zeiträume 89

Uhrzeitdimensionen 71

unterschiedliche Dauern mischen 159

Wochenkalender 90

zeitliche m:n-Beziehungen 175

Zeiträume 84

Zeiträume relativ zu heute 86

Zeitversatz 148

## Datumsdimensionen

erstellen 59

mehrere 65

verwenden mit Uhrzeitdimensionen 71

## Dauer

- Aggregation 141
- Formel-Engine 154
- laufende Ereignisse 149
- mehrere Datumsangaben 143
- Überblick 139
- unterschiedliche Dauern mischen 159
- zeitliche Beziehungen materialisieren 181
- zeitliche m:n-Beziehungen 175
- Zeitversatz 148
- Zuordnungsfaktoren 178

## Denormalisierung

- Datenmodelle 13, 19
- Faktentabellen 37
- reduzieren 33

## Diagramme 8

## Dimensionen

- automatische Zeitdimensionen 62
- Beziehungen 17
- bidirektionale Filterung 43
- Datumsdimensionen 59
- Definition 16, 238
- Header/Detail-Tabellen 25
- langsam veränderliche Dimensionen 99
- Mehrdeutigkeit 18, 45
- mehrere Datumsdimensionen 65
- Namen 21
- schnell veränderliche Dimensionen 114
- Uhrzeitdimensionen 71

DISTINCT 42

DISTINCTCOUNT 104

Dynamische Segmentierung 208

## **E**

Einzeltabellen 2

Ereignisse 149

Excel

    automatische Zeitdimensionen 62

    Power Pivot 10

EXCEPT 83

## **F**

Faktentabellen

    additive berechnete Felder 54

    Aggregation 26

    als Brückentabelle verwenden 182

    Beziehungen 17

    bidirektionale Filterung 28, 43

    Brückentabellen 56

    Definition 16, 237

    denormalisieren 37

    Dimensionen 25

    Granularität 30

    Header/Detail-Tabellen 25, 52

    Hierarchien 25

    langsam veränderliche Dimensionen 113

    Mehrdeutigkeit 18, 45

    mehrere 37

    m:n-Beziehungen 49, 55

    Namen 21

    reduzieren 33

    Überblick 37

FILTER 161, 195, 207, 242

Filterung

    CROSSFILTER 56, 107, 169, 173, 182, 236

    Faktentabellen 43

    FILTER 161, 195, 207, 242

    Filter verschieben 192

- Granularität 193
- Header/Detail-Tabellen 28
- m:n-Beziehungen 168
- Überblick 234
- Formel-Engine 154
- Fremdschlüssel 10
- Funktionen
  - ALL 42
  - CALCULATE 39, 46, 83, 135, 169, 196, 207, 236
  - CALCULATETABLE 87, 135
  - CLOSINGBALANCELASTQUARTER 242
  - CONTAINS 192
  - COUNTROWS 81, 105, 236
  - CROSSFILTER 45, 56, 106, 169, 173, 182, 236
  - DATESINPERIOD 74
  - DATESYTD 73, 75, 242
  - DISTINCT 42
  - DISTINCTCOUNT 104
  - EXCEPT 83
  - FILTER 161, 195, 207, 242
  - HASONEVALUE 83, 219
  - IF 82
  - IFERROR 207
  - INTERSECT 39, 135, 192
  - ISEMPTY 207
  - LASTDATE 125, 154, 242
  - LASTDAY 74
  - List.Numbers 108
  - LOOKUPVALUE 87, 93, 203
  - MAX 109, 112, 209
  - MIN 209
  - PARALLELPERIOD 74
  - RELATED 46, 66, 79
  - RELATEDTABLE 66, 80

SAMEPERIODLASTYEAR 74, 94  
SUM 123, 146, 157, 169  
SUMMARIZE 180  
SUMX 172  
TOTALYTD 242  
TREATAS 193  
UNION 42  
USERRELATIONSHIP 46, 66  
VALUES 207

## **G**

Geschäftskalender 75

Granularität

bidirektionale Filterung 194  
Einzeltabellen 4  
Faktentabellen 113  
Filter verschieben 192  
Header/Detail-Tabellen 30  
langsam veränderliche Dimensionen 110  
mehrere Tabellen 12  
Snapshots 128  
Überblick 187  
vorhersagen 189  
Werte ausblenden 196  
Zuordnungsfaktor 200

## **H**

Halbadditive berechnete Felder

Snapshots aggregieren 124  
Überblick 242

HASONEVALUE 83, 219

Header/Detail-Tabellen

Aggregation 26  
bidirektionale Filterung 28  
Dimensionen 25

Granularität 30

Hierarchien 25

reduzieren 33

Überblick 25

Heute 86

Hierarchien 25

## I

IF 82

IFERROR 207

INTERSECT 39, 135, 192

ISEMPTY 207

## K

Kalender

    Geschäftskalender 75

    Wochenkalender 90

Kaskadierende m:n-Beziehungen 172

Kreuzfilterung *siehe* Bidirektionale Filterung

## L

Langsam veränderliche Dimensionen

    Arten 100

    Granularität 110, 112

    laden 107

    Modellierungstechniken 117

    schnell veränderliche Dimensionen 114

    Überblick 99

    Versionen 104

    verwenden 104

LASTDATE 125, 154, 242

LASTDAY 74

Laufende Ereignisse 149

Leistung 182

List.Numbers 108

LOOKUPVALUE 87, 93, 203

## **M**

Materialisieren 181

Matrizen

    Datenschnitte 134

    Snapshots 130

MAX 109, 112, 209

Measures *siehe* Berechnete Felder

Mehrdeutigkeit 18, 45

Menüband 10

MIN 209

m:n-Beziehungen

    bidirektionale Filterung 168

    Brückentabellen 182

    Faktentabellen 50, 55

    kaskadierend 172

    Leistung 182

    materialisieren 180

    nicht additive berechnete Felder 171

    Überblick 167

    zeitliche m:n-Beziehungen 175

    Zuordnungsfaktoren 178

## **N**

Namen

    Dimensionen 21

    Faktentabellen 21

    Spalten 21

    Tabellen 21

Natürliche Snapshots 122

Nicht additive berechnete Felder

    m:n-Beziehungen 170

    Überblick 241

Nicht überlappende Zeiträume 85

Normalisieren 13

## **O**

OLTP 14

Online Transactional Processing 14

## **P**

PARALLELPERIOD 74

Power BI Desktop 64

Power Pivot 10

Primärschlüssel 9

## **Q**

Quelltabellen 9

Quellwährungen

    eine Quellwährung, mehrere Berichtswährungen 222

    mehrere Quellwährungen, eine Berichtswährung 218

    mehrere Quellwährungen, mehrere Berichtswährungen 227

## **R**

Reduzieren 33

RELATED 46, 66, 79

RELATEDTABLE 66, 80

## **S**

SAMEPERIODLASTYEAR 74, 94

SCDs *siehe* Langsam veränderliche Dimensionen

Schemata

    Schneeflocke 19, 238

    Sternschemata 15, 238

Schlüssel 9

Schneeflocke 19, 238

Schnell veränderliche Dimensionen 114

Segmentierung

    ABC-Analyse 210

    berechnete Spalten 210

    Beziehungen über mehrere Spalten 203

    dynamisch 208

statisch 206

Überblick 203

## Snapshots

abgeleitet 122, 128

additive berechnete Felder 124

Aggregation 122

Arten 122

Datenschnitte 134

Granularität 128

halbadditive berechnete Felder 124

natürlich 122

Überblick 121

Übergangsmatrix 130

## Spalten

berechnete Spalten 210

Beziehungen über mehrere Spalten 203

Fremdschlüssel 10

Namen 21

Primärschlüssel 9

Statische Segmentierung 206

Sternschemata 15, 238

SUM 123, 146, 157, 169

SUMMARIZE 180

SUMX 172

## T

### Tabellen

1:n-Beziehungen 10

abgeleitete Snapshots 122, 128

additive berechnete Felder 124

Aggregation 26, 122

Arten 122

berechnete Spalten 210

Beziehungen über mehrere Spalten 203

bidirektionale Filterung 28  
Brückentabellen 238, 240  
CALCULATETABLE 87, 135  
Datenschnitte 134  
denormalisieren 13, 19  
Dimensionen 25  
Einzeltabellen 2  
Faktentabellen 49  
Fremdschlüssel 10  
Granularität 30, 128  
halbadditive berechnete Felder 124  
Header/Detail-Tabellen 25  
Hierarchien 25  
m:n-Beziehungen 181  
Namen 21  
natürliche Snapshots 122  
normalisieren 13  
OLTP 14  
Primärschlüssel 9  
Quelltabellen 9  
reduzieren 33  
RELATEDTABLE 66, 80  
Segmentierung 203  
Snapshots 121  
Überblick 231  
Übergangsmatrix 130  
Ziel 10  
TOTALYTD 242  
TREATAS 193

## **U**

Übergangsmatrix  
    Datenschnitte 134  
    Snapshots 130

Überlappende Zeiträume 89

Uhrzeit

Aggregation 141

automatische Zeitdimensionen 62

datumsübergreifend 143

Formel-Engine 154

laufende Ereignisse 149

trennen von Datum 71, 143

Überblick 139

verwenden mit Datumsdimensionen 71

Zeitversatz 148

UNION 42

USERRELATIONSHIP 46, 66

## **V**

VALUES 207

Versionen 104

Vorhersagen 189

## **W**

Währungsumrechnung

eine Berichtswährung, mehrere Quellwährungen 218

eine Quellwährung, mehrere Berichtswährungen 222

mehrere Berichtswährungen, eine Quellwährung 222

mehrere Berichtswährungen, mehrere Quellwährungen 227

mehrere Quellwährungen, eine Berichtswährung 218

mehrere Quellwährungen, mehrere Berichtswährungen 227

Überblick 217

Werte

ausblenden 196

HASONEVALUE 83, 219

LOOKUPVALUE 87, 93, 203

VALUES 207

Zuordnungsfaktor 200

Wochenkalender 90

## Z

### Zeitinformationen

- Arbeitstage 77, 80
- automatische Zeitdimensionen 62
- Berechnungen 73
- Datumsdimensionen 59
- Datums- und Uhrzeitdimensionen 71
- Geschäftskalender 75
- mehrere Datumsdimensionen 65
- nicht überlappende Zeiträume 85
- überlappende Zeiträume 89
- Uhrzeitdimensionen 71
- Wochenkalender 90
- Zeiträume 84
- Zeiträume relativ zu heute 86

### Zeitliche m:n-Beziehungen 175

- materialisieren 180
- Zuordnungsfaktoren 178

### Zeiträume

- DATESINPERIOD 74
- nicht überlappend 85
- PARALLELPERIOD 74
- relativ zu heute 86
- SAMEPERIODLASTYEAR 74, 94
- Überblick 84
- überlappend 86, 89

### Zeitversatz 148

### Zieltabellen 10

### Zuordnungsfaktoren 178, 200