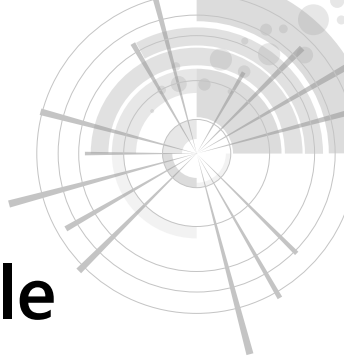


Segmentierungsmodelle



In Kapitel 1 haben Sie gelernt, wie Sie Ihre Daten mit Standardbeziehungen modellieren: Zwei Tabellen werden über eine einzelne Spalte miteinander verknüpft. Später haben wir sogar m:n-Beziehungen mithilfe von Standardbeziehungen eingerichtet. In diesem Kapitel erfahren Sie, wie Sie mithilfe von DAX mit noch anspruchsvolleren Beziehungen zwischen Tabellen umgehen. In Tabellenmodellen lassen sich zwischen den Tabellen einfache und bidirektionale Beziehungen einrichten, was jedoch eine Beschränkung darstellt. Mit DAX können Sie aber weit aufwendigere Modelle mit praktisch allen möglichen Arten von Beziehungen aufbauen, auch virtuellen. In sehr anspruchsvollen Situationen spielt DAX eine große Rolle bei der Definition des Datenmodells.

Als Beispiele zur Veranschaulichung dieser Art von Beziehungen verwenden wir Datenmodelle, bei denen es hauptsächlich um die Segmentierung der Daten geht. Die Segmentierung ist ein gängiges Modellierungsmuster, das immer dann auftritt, wenn Sie Ihre Daten auf der Grundlage einer Konfigurationstabelle schichten wollen. Nehmen wir an, Sie wollen Ihre Kunden nach dem Altersbereich oder nach dem erzielten Umsatz gruppieren oder Ihre Produkte nach dem Verkaufsbetrag.

Wir werden Ihnen in diesem Kapitel keine fertigen Muster an die Hand geben, die Sie auf Ihre eigenen Modelle anwenden können. Stattdessen wollen wir Ihnen ungewöhnliche Vorgehensweisen für den Aufbau komplexer Modelle mit DAX zeigen, Ihre Kenntnisse über Beziehungen vertiefen und Ihnen vorführen, was Sie mit DAX-Formeln alles erreichen können.

Mehrspaltige Beziehungen

Bei der ersten Art von Beziehungen, die wir Ihnen zeigen möchten, handelt es sich um berechnete physische Beziehungen. Der einzige Unterschied zu Standardbeziehungen besteht darin, dass sich der Schlüssel für die Beziehung in einer berechneten Spalte befindet. Wenn Ihnen für eine Beziehung ein Schlüssel fehlt oder wenn Sie ihn mit komplizierten Formeln berechnen müssen, können Sie auf berechnete Spalten zurückgreifen. Das Ergebnis ist immer noch eine physische Beziehung.

Die Tabular-Engine lässt nur Beziehungen auf der Grundlage einer einzigen Spalte zu. Allerdings können Beziehungen, die auf mehreren Spalten beruhen, sehr nützlich sein und treten in vielen Datenmodellen auf. Bei solchen Modellen haben Sie die beiden folgenden Möglichkeiten:

- Definieren Sie eine berechnete Spalte mit einer Kombination der gewünschten Schlüssel und verwenden Sie diese Spalte als neuen Schlüssel für die Beziehung.
- Denormalisieren Sie die Spalten der Zieltabelle (der 1-Seite in der 1:n-Beziehung) mithilfe der Funktion LOOKUPVALUE.

Nehmen wir an, Sie haben die Aktion »Produkt des Tages«, bei der einzelne Produkte an einem bestimmten Tag einen besonderen Rabatt erhalten (siehe Abbildung 10–1).

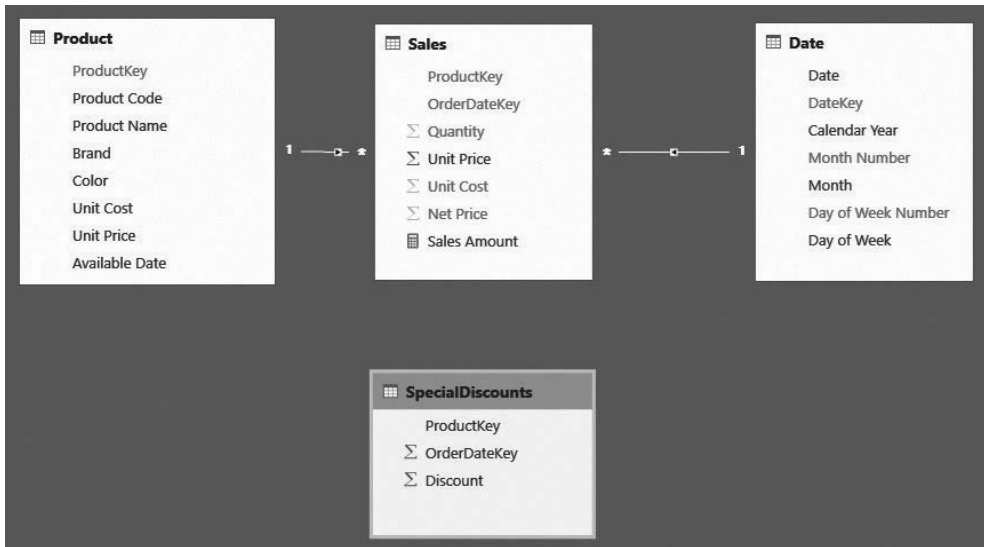


Abbildung 10–1 Die Tabelle *SpecialDiscounts* benötigt eine Beziehung auf der Grundlage von zwei Spalten in *Sales*.

Die Rabatttabelle *SpecialDiscounts* enthält die drei Spalten *ProductKey*, *OrderDateKey* und *Discount*. Bei der Berechnung eines Rabatts haben Sie das Problem, dass dieser Rabatt bei jedem Verkauf sowohl von *ProductKey* als auch von *OrderDateKey* abhängt. Die erforderliche Beziehung zwischen *Sales* und *SpecialDiscounts* können Sie jedoch nicht einrichten, da sie zwei Spalten umfasst, während Tabular nur Beziehungen über eine Spalte zulässt.

Um dieses Problem zu lösen, nutzen Sie die Tatsache, dass es nicht verboten ist, eine Beziehung auf eine berechnete Spalte zu stützen. Sie können also eine neue Spalte erstellen, die die beiden gewünschten Spalten kombiniert, und über diese neue Spalte die Beziehung einrichten. Diese neue Spalte legen Sie sowohl in *SpecialDiscount* als auch in *Sales* an. In *Sales* verwenden Sie dazu den folgenden Code:

```
Sales[SpecialDiscountKey] = Sales[ProductKey] & "-" & Sales[OrderDateKey]
```

In *SpecialDiscount* verwenden Sie einen ähnlichen Ausdruck. Anschließend können Sie die Beziehung zwischen den beiden Tabellen einrichten. Dadurch erhalten Sie das Modell aus Abbildung 10–2.

Diese Lösung ist einfach und funktioniert einwandfrei. Es gibt jedoch Situationen, in denen sie nicht die beste Lösung darstellt, da Sie dazu zwei berechnete Spalten mit vielen verschiedenen Werten erstellen müssen, was im Hinblick auf die Leistung nicht ratsam ist.

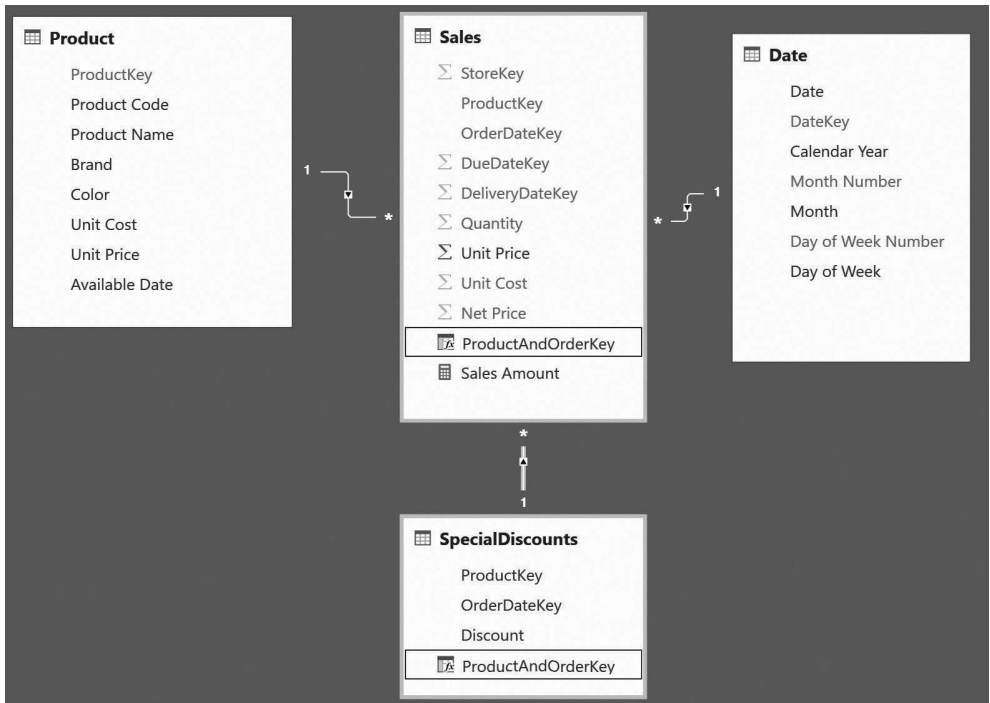


Abbildung 10-2 Als Grundlage für eine Beziehung kann auch eine berechnete Spalte dienen.

Eine andere Möglichkeit besteht darin, die Funktion LOOKUPVALUE zu verwenden. Damit können Sie den Rabatt unmittelbar in die Faktentabelle denormalisieren, indem Sie mit dem folgenden Code eine neue berechnete Spalte in *Sales* definieren:

```
Sales[SpecialDiscount] =
LOOKUPVALUE (
    SpecialDiscounts[Discount],
    SpecialDiscounts[ProductKey], Sales[ProductKey],
    SpecialDiscounts[OrderDateKey], Sales[OrderDateKey]
)
```

Bei dieser Vorgehensweise erstellen Sie keine Beziehung, sondern verschieben den Rabattwert in die Faktentabelle und führen einen Nachschlagevorgang durch. Technisch ausgedrückt denormalisieren Sie den Wert von *SpecialDiscount* von der Tabelle *SpecialDiscounts* in *Sales*.

Beide Vorgehensweisen sind gut, und für welche Sie sich entscheiden, hängt von mehreren Faktoren ab. Wenn *Discount* die einzige Spalte ist, die Sie von der Tabelle *SpecialDiscounts* benötigen, dann stellt die Denormalisierung die bessere Methode dar. Dabei wird nur eine einzige berechnete Spalte mit weniger einzigartigen Werten erstellt, und nicht zwei mit vielen einzigartigen Werten. Das verringert die Speichernutzung und macht den Code einfacher.

Enthält *SpecialDiscounts* dagegen viele Spalten, die Sie in Ihrem Code benötigen, würde es eine Speicherverschwendung und möglicherweise auch Leistungseinbußen bedeuten, wenn

Sie sie alle in die Faktentabelle denormalisieren. In diesem Fall ist die berechnete Spalte mit dem neuen zusammengesetzten Schlüssel die bessere Methode.

Dieses erste, einfache Beispiel ist vor allem deshalb wichtig, weil es eine wichtige Eigenschaft von DAX veranschaulicht, nämlich die Möglichkeit, Beziehungen auf der Grundlage von berechneten Spalten zu erstellen. Dadurch können Sie jede mögliche Art von Beziehung einrichten, sofern Sie sie nur berechnen und in einer berechneten Spalte materialisieren lassen können. Mit dem nächsten Beispiel zeigen wir Ihnen, wie Sie Beziehungen auf der Grundlage statischer Bereiche einrichten. Durch die Erweiterung dieses Prinzips lassen sich fast alle Arten von Beziehungen aufstellen.

Statische Segmentierung

Statische Segmentierung kommt sehr häufig vor, wenn Sie einen Wert in einer Tabelle haben und weniger an der Analyse des Wertes an sich interessiert sind (da es Hunderte und Tausende von möglichen Werten geben kann), sondern ihn in Segmente aufteilen wollen. Zwei gängige Beispiele dafür sind die Analyse von Verkäufen nach dem Alter der Kunden oder nach dem Listenpreis. Es wäre sinnlos, die Verkaufsbeträge auf alle möglichen Werte des Listenpreises aufzuteilen, da es zu viele davon gibt. Wenn Sie die verschiedenen Preise aber zu Bereichen gruppieren, können Sie aus der Untersuchung dieser Gruppen wertvolle Erkenntnisse gewinnen.

Betrachten Sie als Beispiel die Tabelle *PriceRanges* mit den Preisbereichen. Wie Sie in Abbildung 10–3 sehen, legen Sie dabei für jeden Bereich die Grenzen fest.

PriceRangeKey	PriceRange	MinPrice	MaxPrice
1	VERY LOW	0	10
2	LOW	10	30
3	MEDIUM	30	80
4	HIGH	80	150
5	VERY HIGH	150	99999

Abbildung 10–3 Die Konfigurationstabelle für die Preisbereiche

Wie im vorherigen Beispiel können Sie auch hier keine direkte Beziehung zwischen der Faktentabelle mit den Verkäufen und der Konfigurationstabelle *PriceRanges* aufstellen, da der Schlüssel in der Konfigurationstabelle auf einer Bereichsbeziehung beruht, diese Art von Beziehung in DAX aber nicht zulässig ist. Daher besteht die beste Lösung darin, den Preisbereich mithilfe einer berechneten Spalte direkt in die Faktentabelle zu denormalisieren. Das Codemuster dafür ähnelt dem vorherigen, wobei der Hauptunterschied in der folgenden Formel besteht:

```
Sales[PriceRange] =
CALCULATE (
    VALUES ( PriceRanges[PriceRange] ),
    FILTER (
        PriceRanges,
        AND (
            PriceRanges[MinPrice] <= Sales[Net Price],
```

```

        PriceRanges[MaxPrice] > Sales[Net Price]
    )
)
)

```

Beachten Sie, dass VALUES in diesem Code zum Abruf eines einzelnen Werts verwendet wird, obwohl diese Funktion normalerweise eine Tabelle zurückgibt. Wenn eine Tabelle jedoch nur aus einer einzigen Zeile und einer einzigen Spalte besteht, wird sie automatisch in einen Skalar umgewandelt, wenn das in dem Ausdruck erforderlich ist.

Aufgrund der Art und Weise, wie die Funktion FILTER das Ergebnis berechnet, gibt sie immer eine einzelne Zeile aus der Konfigurationstabelle zurück. Daher gibt auch VALUES immer eine einzige Zeile zurück, und das Ergebnis von CALCULATE ist die Beschreibung des Preisbereichs mit dem aktuellen Nettopreis. Dieser Ausdruck funktioniert problemlos, wenn die Konfigurationstabelle ordnungsgemäß aufgebaut ist. Sollten die Preisbereiche aber aus irgendeinem Grund Lücken oder Überlappungen aufweisen, gibt VALUES mehrere Zeilen zurück, sodass der Ausdruck zu einem Fehler führt.

Um den vorstehenden Code zu verbessern, können Sie eine Fehlerbehandlungsfunktion nutzen, die eine unsaubere Konfiguration erkennt und eine entsprechende Meldung ausgibt:

```

Sales[PriceRange] =
VAR ResultValue =
    CALCULATE (
        IFERROR (
            VALUES ( PriceRanges[PriceRange] ),
            "Overlapping Configuration"
        ),
        FILTER (
            PriceRanges,
            AND (
                PriceRanges[MinPrice] <= Sales[Net Price],
                PriceRanges[MaxPrice] > Sales[Net Price]
            )
        )
    )
)
RETURN
    IF (
        ISEMPTY ( ResultValue ),
        "Wrong Configuration",
        ResultValue
    )

```

Dieser Code erkennt sowohl sich überlappende Werte (mit dem internen IFERROR) als auch Lücken in der Konfiguration (durch Überprüfung des Ergebnisses mit ISEMPTY vor dem Rücksprung zum Aufrufer). Da hiermit garantiert ist, dass immer ein guter Wert zurückgegeben wird, ist der Code jetzt viel sicherer als in der vorherigen Version.

Berechnete physische Beziehungen sind ein sehr vielseitiges Instrument für die Modellierung in Power BI und Excel, da Sie damit sehr anspruchsvolle Beziehungen einrichten können. Außerdem erfolgt die Berechnung der Beziehung während der Aktualisierung der Daten und nicht bei der Abfrage des Modells. Unabhängig von der Komplexität der Beziehungen ergibt sich dadurch eine sehr gute Leistung.

Dynamische Segmentierung

In vielen Situationen können Sie die logische Beziehung zwischen den Tabellen nicht statisch einrichten. In diesen Fällen ist es nicht möglich, berechnete statische Beziehungen zu verwenden. Stattdessen müssen Sie die Beziehung in dem berechneten Feld definieren, um sie dynamisch handhaben zu können. Da die Beziehungen dann nicht Bestandteile des Modells sind, sprechen wir hier im Gegensatz zu den bisher betrachteten physischen Beziehungen von *virtuellen Beziehungen*.

In unserem nächsten Beispiel führen wir eine virtuelle Beziehung als Variante der zuvor gezeigten statischen Segmentierung vor. Bei der statischen Segmentierung hatten wir jeden Verkauf mithilfe einer berechneten Spalte einem bestimmten Segment zugeordnet. Bei der dynamischen Segmentierung erfolgt diese Zuordnung dynamisch.

Nehmen wir an, Sie wollen Ihre Kunden nach dem Verkaufsvolumen gruppieren. Da das Verkaufsvolumen aber jeweils von den in dem Bericht verwendeten Datenschnitten abhängt, kann die Segmentierung nicht statisch sein. Beispielsweise kann ein Kunde bei der Filterung nach einem einzelnen Jahr zu der einen Gruppe gehören und bei der Filterung nach einem anderen Jahr zu einer anderen. Daher können Sie zur Lösung keine physische Beziehung heranziehen oder das Datenmodell abwandeln, um den DAX-Code zu vereinfachen. Die einzige Möglichkeit in einem solchen Fall besteht darin, die Ärmel hochzukrempeln und anspruchsvollen DAX-Code zur Berechnung des Wertes zu schreiben.

Als Erstes definieren Sie die Konfigurationstabelle *Segments*, die Sie in Abbildung 10–4 sehen.

SegmentCode	Segment	MinSale	MaxSale
1	Very Low	0	75
2	Low	75	100
3	Medium	100	500
4	High	500	1000
6	Very High	1000	9999999

Abbildung 10–4 Die Konfigurationstabelle für die dynamische Segmentierung

Was wir berechnen wollen, ist die Anzahl der Kunden, die zu einer bestimmten Gruppe gehören, wobei alle Filter im aktuellen Filterkontext berücksichtigt werden müssen. Die folgende Formel sieht zwar harmlos aus, erfordert aber doch Aufmerksamkeit, da sie einen Kontextübergang enthält:

```

CustInSegment :=
COUNTROWS (
    FILTER (
        Customer,
        AND (
            [Sales Amount] > MIN ( Segments[MinSale] ),
            [Sales Amount] <= MAX ( Segments[MaxSale] )
        )
    )
)

```

Um das Verhalten dieser Formel zu verstehen, sehen Sie sich den Bericht aus Abbildung 10–5 an, der in den Zeilen die Segmente und in den Spalten die Kalenderjahre anzeigt.

Segment	CY 2007	CY 2008	CY 2009	Total
Very Low	351	266	255	810
Low	141	14	12	166
Medium	365	76	52	485
High	250	36	35	311
Very High	302	132	160	581
Total	1,409	524	514	2,353

Abbildung 10–5 Diese Pivottabelle zeigt die dynamische Segmentierung in Aktion.

Schauen Sie sich die Zelle an, die für das Jahr 2008 in der Gruppe mit mittlerem Verkaufsvolumen (*Medium*) 76 Kunden ausgibt. Die Formel iteriert über *Customer* und prüft für jeden Kunden, ob der Wert von *Sales Amount* zwischen das MIN von *MinSale* und das MAX von *MaxSale* fällt. Der Wert von *Sales Amount* stellt aufgrund des Kontextübergangs die Verkäufe für den einzelnen Kunden dar. Wie zu erwarten, ist das resultierende berechnete Feld für die Segmente und Kunden additiv, für alle anderen Dimensionen aber nicht additiv.

Die Formel funktioniert nur, wenn Sie alle Segmente auswählen. Bei der Auswahl von beispielsweise nur *Very Low* und *Very High* (also bei Wegfall der drei Zwischensegmente) geben MIN und MAX nicht die korrekte Auswahl wieder, da sie alle Kunden einbeziehen. Das führt zu falschen Gesamtsummen, wie Abbildung 10–6 zeigt.

Segment	Segment	CY 2007	CY 2008	CY 2009	Total
<input checked="" type="checkbox"/> Very Low	Very Low	351	266	255	810
<input type="checkbox"/> Low					
<input type="checkbox"/> Medium	Very High	302	132	160	581
<input type="checkbox"/> High	Total	1,409	524	514	2,353
<input checked="" type="checkbox"/> Very High					

Abbildung 10–6 Falsche Werte bei der Verwendung eines Datenschnitts mit nicht zusammenhängender Auswahl

Wollen Sie den Benutzer einzelne Segmente auswählen lassen, müssen Sie die Formel daher wie folgt umschreiben:

```

CustInSegment :=
SUMX (
    Segments,

```

```

COUNTROWS (
    FILTER (
        Customer,
        AND (
            [Sales Amount] > Segments[MinSale],
            [Sales Amount] <= Segments[MaxSale]
        )
    )
)

```

Diese Version der Formel leidet zwar nicht unter dem möglichen Problem einer Teilauswahl der Segmente, doch aufgrund der doppelten Iteration über die Tabellen kann die Leistung leiden. Wie Sie in Abbildung 10–7 sehen, sind die Ergebnisse jetzt aber korrekt.

Segment	Segment	CY 2007	CY 2008	CY 2009	Total
■ Very Low	Very Low	351	266	255	810
□ Low	Very High	302	132	160	581
□ Medium	Total	653	398	415	1,391
□ High					
■ Very High					

Abbildung 10–7 Das berechnete Feld zeigt jetzt andere Gesamtsummen, da die Teilauswahl der Segmente diesmal korrekt berücksichtigt wird.

Virtuelle Beziehungen bieten viele Möglichkeiten. Auch wenn sie den Benutzern als echte Beziehungen erscheinen, sind sie keine Bestandteile des Modells, sondern werden jedes Mal mit dem DAX-Code berechnet. Ist die dazu erforderliche Formel sehr aufwendig oder das Modell zu umfangreich, kann die Leistung beeinträchtigt werden. Für Modelle mittlerer Größe sind solche Beziehungen jedoch hervorragend geeignet.



Versuchen Sie, diese Prinzipien auf Ihr Geschäftsfeld zu übertragen, um zu sehen, ob dieses Muster auch für die von Ihnen benötigten Schichtungen geeignet ist.

ABC-Analyse

Berechnete Spalten werden in der Datenbank gespeichert. Für unsere Zwecke ist das von enormer Bedeutung, da es neue Möglichkeiten für die Modellierung eröffnet. In diesem Abschnitt sehen wir uns einige Aufgaben an, die sich mit berechneten Spalten sehr wirtschaftlich lösen lassen.

Als Beispiel dafür führen wir Ihnen die ABC-Analyse mit Power BI vor. Die ABC-Analyse basiert auf dem Pareto-Prinzip und ist eine weitverbreitete Technik, um das Kerngeschäft eines Unternehmens zu bestimmen, gewöhnlich in Form der besten Produkte oder besten Kunden. In diesem Beispiel wollen wir uns auf die Produkte konzentrieren.

Das Ziel der ABC-Analyse besteht darin, die Produkte herauszufinden, die eine erhebliche Auswirkung auf das Gesamtgeschäft haben, sodass die Manager ihre Anstrengungen darauf

konzentrieren können. Dazu wird jedes Produkt wie folgt einer der drei Kategorien A, B oder C zugewiesen:

- Produkte in Klasse A machen 70 % der Erträge aus.
- Produkte in Klasse B machen 20 % der Erträge aus.
- Produkte in Klasse C machen 10 % der Erträge aus.

Die Klasse eines Produkts muss in einer berechneten Spalte gespeichert werden, da Sie sie zur Analyse der Produkte verwenden und die Informationen danach filtern wollen. Ein Beispiel dafür sehen Sie in Abbildung 10–8. Sie zeigt eine einfache Pivottabelle, deren Zeilen für die einzelnen Klassen stehen.

ABC Class	NumOfProducts	Margin
A	215	\$1,411,868.11
B	285	\$404,299.99
C	2,017	\$202,448.10
Total	2,517	\$2,018,616.20

Abbildung 10–8 Dieser Bericht zeigt die Anzahl der Produkte und die Gewinnmargen nach Klasse an.

Wie so häufig bei der ABC-Analyse befinden sich auch hier nur wenige Produkte in Klasse A. Sie bilden das Kerngeschäft von Contoso. Produkte in Klasse B sind weniger bedeutend, für das Unternehmen aber immer noch unverzichtbar. Die Produkte in Klasse C dagegen bieten sich zur Herausnahme aus dem Angebot an, da sie im Vergleich zu den Kernprodukten nur sehr wenig zu den Einnahmen beitragen, obwohl sie so zahlreich sind.

Wie Sie in Abbildung 10–9 sehen, ist das Datenmodell für dieses Beispiel recht einfach. Sie brauchen dazu lediglich die Verkäufe und die Produkte.

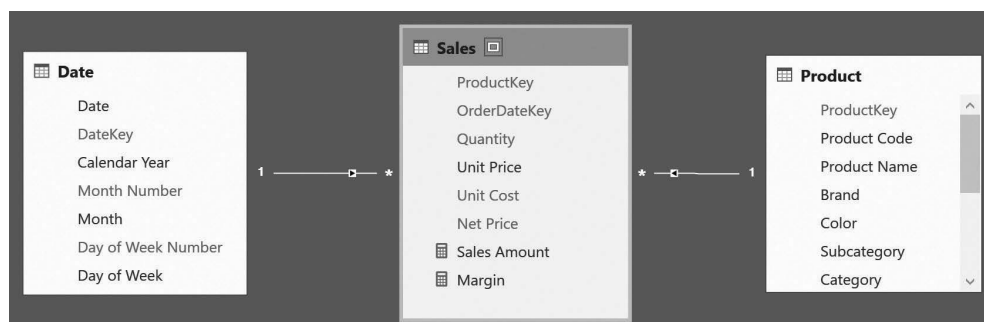


Abbildung 10–9 Das Datenmodell zur Berechnung der Produktklassen ist sehr einfach.

Wir wollen dieses Modell jetzt ändern, indem wir weitere Spalten hinzufügen. Neue Tabellen oder Beziehungen brauchen wir dagegen nicht. Um die Klasse der Produkte zu bestimmen, müssen Sie jeweils die Gesamtmenge für ein Produkt berechnen und mit dem Gesamtertrag vergleichen. Dadurch erhalten Sie den prozentualen Anteil des Produkts am Gesamtertrag. Anschließend sortieren Sie alle Produkte nach diesem prozentualen Anteil und ermitteln eine kumulierte Summe. Sobald diese kumulierte Summe 70 % beträgt, haben Sie die Produkte in Klasse A identifiziert. Die restlichen Produkte fallen in Klasse B, bis 90 % erreicht sind (70 + 20),

und alle weiteren Produkte gehören zu Klasse C. Die Berechnung erfolgt ausschließlich mithilfe berechneter Spalten.

Als Erstes benötigen Sie in der Tabelle *Product* eine berechnete Spalte mit der Marge für jedes Produkt. Das können Sie mit dem folgenden Ausdruck leicht erledigen:

```
Product[TotalMargin] =
SUMX (
    RELATEDTABLE( Sales ),
    Sales[Quantity] * ( Sales[Net Price] - Sales[Unit Cost] )
)
```

Abbildung 10–10 zeigt die Tabelle *Product* mit der neuen berechneten Spalte. Die Daten sind hier in absteigender Reihenfolge nach *TotalMargin* geordnet.

Product Name	TotalMargin ↑
Adventure Works 26" 720p LCD HDTV M140 Silver	\$81,856.27
Contoso Telephoto Conversion Lens X400 Silver	\$53,464.04
Fabrikam Refrigerator 24.7CuFt X9800 White	\$51,574.26
A. Datum SLR Camera X137 Grey	\$51,459.16
Litware Refrigerator 24.7CuFt X980 Brown	\$29,756.33
Litware Refrigerator 24.7CuFt X980 White	\$28,256.56
NT Washer & Dryer 27in L2700 Blue	\$26,591.59
Proseware Projector 1080p DLP86 Black	\$25,065.45
NT Washer & Dryer 24in M2400 Green	\$24,472.50
SV 16xDVD M360 Black	\$20,989.22
Contoso Projector 1080p X980 White	\$19,648.68

Abbildung 10–10 *TotalMargin* ist eine berechnete Spalte in der Tabelle *Product*.

Der nächste Schritt besteht darin, für die gesamte nach *TotalMargin* geordnete Tabelle *Product* die kumulierte Summe von *TotalMargin* zu berechnen. Die kumulierte Summe für ein Produkt ist dabei jeweils die Summe der Produktmargen aller Produkte, deren *TotalMargin*-Wert größer oder gleich dem des aktuellen Produkts ist. Dazu verwenden Sie die folgende Formel:

```
Product[MarginRT] =
VAR
    CurrentTotalMargin = 'Product'[TotalMargin]
RETURN
SUMX (
    FILTER (
        'Product',
        'Product'[TotalMargin] >= CurrentTotalMargin
    ),
    'Product'[TotalMargin]
)
```

In Abbildung 10–11 sehen Sie die Tabelle *Product* mit dieser neuen Spalte.

Product Name	TotalMargin	↑ MarginRT
Adventure Works 26" 720p LCD HDTV M140 Silver	\$81,856.27	\$81,856.27
Contoso Telephoto Conversion Lens X400 Silver	\$53,464.04	\$135,320.31
Fabrikam Refrigerator 24.7CuFt X9800 White	\$51,574.26	\$186,894.56
A. Datum SLR Camera X137 Grey	\$51,459.16	\$238,353.72
Litware Refrigerator 24.7CuFt X980 Brown	\$29,756.33	\$268,110.06
Litware Refrigerator 24.7CuFt X980 White	\$28,256.56	\$296,366.62
NT Washer & Dryer 27in L2700 Blue	\$26,591.59	\$322,958.20
Proseware Projector 1080p DLP86 Black	\$25,065.45	\$348,023.65
NT Washer & Dryer 24in M2400 Green	\$24,472.50	\$372,496.15
SV 16xDVD M360 Black	\$20,989.22	\$393,485.38
Contoso Projector 1080p X980 White	\$19,648.68	\$413,134.06

Abbildung 10–11 *MarginRT* enthält die kumulierte Summe (»running total«) für die nach *TotalMargin* sortierten Zeilen.

Nun müssen Sie die kumulierte Summe als prozentualen Anteil an der Gesamtmarge berechnen. Das lässt sich mit einer weiteren berechneten Spalte leicht erledigen. Fügen Sie mit der folgenden Formel die Spalte *RunningPct* hinzu:

$$\text{Product}[\text{MarginPct}] = \text{DIVIDE} (\text{'Product'}[\text{MarginRT}], \text{SUM} (\text{'Product'}[\text{TotalMargin}]))$$

Abbildung 10–12 zeigt die neue berechnete Spalte. Um das Ergebnis aussagekräftiger zu machen, wurde sie als Prozentangabe formatiert.

Product Name	TotalMargin	↑ MarginRT	MarginPct
Adventure Works 26" 720p LCD HDTV M140 Silver	\$81,856.27	\$81,856.27	4.06 %
Contoso Telephoto Conversion Lens X400 Silver	\$53,464.04	\$135,320.31	6.70 %
Fabrikam Refrigerator 24.7CuFt X9800 White	\$51,574.26	\$186,894.56	9.26 %
A. Datum SLR Camera X137 Grey	\$51,459.16	\$238,353.72	11.81 %
Litware Refrigerator 24.7CuFt X980 Brown	\$29,756.33	\$268,110.06	13.28 %
Litware Refrigerator 24.7CuFt X980 White	\$28,256.56	\$296,366.62	14.68 %
NT Washer & Dryer 27in L2700 Blue	\$26,591.59	\$322,958.20	16.00 %
Proseware Projector 1080p DLP86 Black	\$25,065.45	\$348,023.65	17.24 %
NT Washer & Dryer 24in M2400 Green	\$24,472.50	\$372,496.15	18.45 %
SV 16xDVD M360 Black	\$20,989.22	\$393,485.38	19.49 %
Contoso Projector 1080p X980 White	\$19,648.68	\$413,134.06	20.47 %

Abbildung 10–12 *MarginPct* berechnet den prozentualen Anteil der kumulierten Summe am Gesamtertrag.

Als Letztes müssen Sie jetzt noch die Prozentangaben in die Klassenzugehörigkeit übersetzen. Mit den zuvor genannten Werten 70, 20 und 10 ergibt sich die folgende einfache Formel:

```
Product[ABC Class] =
IF (
  'Product'[MarginPct] <= 0.7,
  "A",
  IF (
    'Product'[MarginPct] <= 0.9,
    "B",
    "C"
  )
)
```

Das Ergebnis sehen Sie in Abbildung 10–13.

Product Name	TotalMargin	MarginRT	MarginPct	ABC Class
Adventure Works 26" 720p LCD HDTV M140 Silver	\$81,856.27	\$81,856.27	4.06 %	A
Contoso Telephoto Conversion Lens X400 Silver	\$53,464.04	\$135,320.31	6.70 %	A
Fabrikam Refrigerator 24.7CuFt X9800 White	\$51,574.26	\$186,894.56	9.26 %	A
A. Datum SLR Camera X137 Grey	\$51,459.16	\$238,353.72	11.81 %	A
Litware Refrigerator 24.7CuFt X980 Brown	\$29,756.33	\$268,110.06	13.28 %	A
Litware Refrigerator 24.7CuFt X980 White	\$28,256.56	\$296,366.62	14.68 %	A
NT Washer & Dryer 27in L2700 Blue	\$26,591.59	\$322,958.20	16.00 %	A
Proseware Projector 1080p DLP86 Black	\$25,065.45	\$348,023.65	17.24 %	A
NT Washer & Dryer 24in M2400 Green	\$24,472.50	\$372,496.15	18.45 %	A
SV 16xDVD M360 Black	\$20,989.22	\$393,485.38	19.49 %	A
Contoso Projector 1080p X980 White	\$19,648.68	\$413,134.06	20.47 %	A

Abbildung 10–13 Das Ergebnis der Formel für die Klasseneinteilung ist die berechnete Spalte *ABC Class*.

Da *ABC Class* eine berechnete Spalte ist, wird sie in der Datenbank gespeichert und kann für Datenschnitte, Filter, Zeilen und Spalten verwendet werden, um interessante Berichte zu erstellen.

Wie dieses Beispiel zeigt, können Sie in dem Modell anspruchsvolle Berechnungen speichern, indem Sie berechnete Spalten verwenden und diese systematisch ausführen. Es braucht etwas Erfahrung, bis Sie erkennen können, ob für eine Berechnung eine berechnete Spalte oder ein berechnetes Feld besser geeignet ist, aber sobald Sie sich damit vertraut gemacht haben, können Sie die Möglichkeiten nutzen, die berechnete Spalten bieten.



Weitere Informationen zur ABC-Analyse finden Sie auf <https://de.wikipedia.org/wiki/ABC-Analyse>.

Zusammenfassung

In diesem Kapitel sind wir einen Schritt über Standardbeziehungen hinausgegangen, indem wir Segmentierungstechniken betrachtet haben, die erheblichen Gebrauch von DAX machen. Die wichtigsten Punkte sind:

- Mit berechneten Spalten können Sie berechnete Beziehungen erstellen. Das ermöglicht es Ihnen, die Beziehung auf eine beliebige Berechnung zu stützen, nicht nur auf Gleichheitsverknüpfungen wie bei den Beziehungen, die die Engine normalerweise anbietet.
- Wenn Sie eine benötigte Beziehung nicht einrichten können, da sie dynamisch ist und von den im Bericht verwendeten Filtern und Datenschnitten abhängt, können Sie virtuelle Beziehungen nutzen. Für den Benutzer sehen sie wie Standardbeziehungen aus, allerdings werden sie im laufenden Betrieb berechnet. Die Leistung kann dabei leiden, aber das wird durch die Flexibilität wettgemacht, die Sie dadurch gewinnen.
- Berechnete Spalten bilden eine hervorragende Ergänzung zu den Modellierungsmöglichkeiten von Tabular. Mit nur wenigen berechneten Spalten können Sie äußerst anspruchsvolle Segmentierungen erreichen. Berechnet werden diese Spalten außerdem bei der Aktualisierung des Modells. Dadurch gewinnen Sie sowohl Geschwindigkeit und Flexibilität, was es Ihnen erlaubt, äußerst leistungsfähige Modelle aufzubauen.

Wir hoffen, Ihnen mit diesen wenigen Beispielen neue Einsichten darin gegeben zu haben, wie Sie mit ein wenig Kreativität hervorragende Modelle gestalten können.