

Die traditionellen Grenzen von Excel genauer verstehen

Datenbankterminologie übernehmen

Einen Blick auf Beziehungen werfen

Kapitel 1

Denken wie eine Datenbank

Mit der Einführung von Business-Intelligence-(BI-)Tools wie Power Pivot und Power Query wird es für Excel-Analysten immer wichtiger, grundlegende Datenbankprinzipien zu verstehen. Bei den traditionellen Excel-Konzepten ist der Lösungsansatz meist relativ intuitiv. Um Power Pivot und Power Query optimal nutzen zu können, brauchen Sie dagegen ein grundlegendes Verständnis der Datenbankterminologie und -architektur. Dieses Kapitel stellt Ihnen ein paar grundlegende Konzepte vor, die Sie kennen sollten, bevor Sie den Rest des Buchs in Angriff nehmen.

Die Grenzen von Excel ausloten – und wie hilfreich Datenbanken sein können

Im Laufe seiner Tätigkeit als Berater hatte es der Autor mit Managern, Buchhaltern und Analysten zu tun, die alle irgendwann eine einfache Tatsache akzeptieren mussten: Ihre Anforderungen an Datenanalysen waren über Excel hinausgewachsen. Sie alle standen vor grundlegenden Herausforderungen, die sich aus einem oder mehreren der drei Problembereiche von Excel ergaben: Skalierbarkeit, Transparenz der Analyseprozesse und Trennung von Daten und ihrer Präsentation.

Skalierbarkeit

Skalierbarkeit ist die Fähigkeit einer Anwendung, sich flexibel an Anforderungen im Hinblick auf Wachstum und Komplexität anzupassen. Im Rahmen dieses Kapitels verstehen wir Skalierbarkeit als die Fähigkeit von Excel, stetig wachsende Datenmengen zu verarbeiten.

Stellen Sie sich vor, Sie arbeiten in einem kleinen Unternehmen und verwenden Excel, um die täglichen Transaktionen zu analysieren. Im Laufe der Zeit haben Sie einen

stabilen Prozess mit allen Formeln, Pivot-Tabellen und Makros entwickelt, die Sie benötigen, um die Daten zu analysieren, die in Ihrem sauber gepflegten Arbeitsblatt gespeichert sind.

Wenn die Datenmenge wächst, werden Sie zunächst Leistungsprobleme feststellen. Die Kalkulationstabelle wird langsam geladen und noch langsamer berechnet. Wie ist es dazu gekommen? Es hat mit der Art und Weise zu tun, wie Excel mit dem Speicher umgeht. Wenn eine Excel-Datei geladen wird, wird die gesamte Datei in den Arbeitsspeicher geladen. Auf diese Weise will Excel eine schnelle Datenverarbeitung und einen schnellen Zugriff ermöglichen. Der Nachteil des Ganzen besteht darin, dass Excel jedes Mal, wenn sich die Daten in Ihrer Tabelle ändern, das gesamte Dokument erneut in den Arbeitsspeicher laden muss. Bei einer großen Kalkulationstabelle wird also sehr viel Arbeitsspeicher benötigt, um selbst die kleinste Änderung zu verarbeiten. Irgendwann geht jeder Aktion, die Sie in Ihrem riesigen Arbeitsblatt vornehmen möchten, eine quälende Wartezeit voraus.

Ihre Pivot-Tabellen benötigen größere Pivot-Caches, wodurch sich die Dateigröße der Excel-Arbeitsmappe fast verdoppelt. Letztendlich wird die Arbeitsmappe zu groß, um sie einfach weiterzugeben. Womöglich denken Sie sogar darüber nach, die Arbeitsmappe in mehrere kleinere Arbeitsmappen aufzuteilen (möglicherweise eine für jede Region). Dies führt letztlich dazu, dass Sie Ihre Arbeit duplizieren.

Mit der Zeit werden Sie vielleicht an die Grenze von 1.048.576 Zeilen im Arbeitsblatt stoßen. Was geschieht dann? Legen Sie ein neues Arbeitsblatt an? Wie können Sie zwei Datensätze auf zwei verschiedenen Arbeitsblättern als eine Einheit analysieren? Sind Ihre Formeln noch geeignet? Müssen Sie neue Makros schreiben?

Alles gute Fragen, auf die Sie gegebenenfalls Antworten finden müssen.

Natürlich gibt es auch die Excel-Power-Kunden, die clevere Möglichkeiten finden, diese Einschränkungen zu umgehen. Letztendlich werden diese Methoden aber immer nur provisorische Lösungen sein. Und mit der Zeit werden selbst diese Power-Kunden beginnen, weniger über die effektivste Art und Weise nachzudenken, wie sie die Analyse ihrer Daten durchführen und präsentieren können, sondern mehr darüber, wie sie die Daten in Excel »unterbringen« können, ohne ihre Formeln und Funktionen unbrauchbar zu machen. Excel ist flexibel genug, dass ein geübter Kunde die meisten Dinge gut anpassen kann. Wenn Kunden jedoch nur in Excel denken, schränken sie sich zweifellos selbst ein, wenn auch auf eine unglaublich funktionale Weise.

Außerdem zwingen diese Kapazitätseinschränkungen Excel-Kunden oft dazu, sich die Daten aufbereiten zu lassen. Das heißt, ein Drittanbieter extrahiert große Datenmengen aus einer riesigen Datenbank, fasst sie zusammen und bereitet sie für die Verwendung in Excel auf. Sollte ein seriöser Analyst tatsächlich immer von anderen abhängig sein, wenn er Daten benötigt? Wie wäre es, wenn man einem Analysten die Werkzeuge an die Hand geben könnte, um auf große Datenmengen zuzugreifen, ohne von anderen abhängig zu sein, die ihm die Daten bereitstellen? Könnte dieser Analyst für das Unternehmen wertvoller sein? Könnte sich dieser Analyst auf die Genauigkeit der Analyse und die Qualität der Präsentation konzentrieren, statt ständig Excel-Daten pflegen zu müssen?

Ein relationales Datenbanksystem (wie Access oder SQL Server) ist ein logischer nächster Schritt für den Analysten, der mit einem stetig wachsenden Datenbestand konfrontiert ist.

Datenbanksysteme haben in der Regel keine Leistungseinbußen bei großen gespeicherten Datenmengen und sind auf große Datenvolumen ausgelegt. Ein Analyst kann dann größere Datenmengen verarbeiten, ohne dass die Daten zusammengefasst oder für Excel aufbereitet werden müssen. Wenn ein Prozess für das Unternehmen immer wichtiger wird und in einer besser geeigneten Umgebung verfolgt werden muss, ist es außerdem einfacher, diesen Prozess weiterzuentwickeln und zu erweitern, wenn er bereits in einem relationalen Datenbanksystem gespeichert ist.

Transparenz analytischer Prozesse

Eine der attraktivsten Eigenschaften von Excel ist seine Flexibilität. Jede einzelne Zelle kann Text, eine Zahl, eine Formel oder praktisch alles andere enthalten, was der Kunde definiert. Dies ist in der Tat einer der Hauptgründe, warum Excel ein effektives Werkzeug für die Datenanalyse ist. Kunden können benannte Bereiche, Formeln und Makros verwenden, um ein kompliziertes System ineinandergreifender Berechnungen, verknüpfter Zellen und formatierter Zusammenfassungen zu erstellen, die in ihrer Gesamtheit eine endgültige Analyse ergeben.

Wo liegt also das Problem? Das Problem ist, dass es keine Transparenz der analytischen Prozesse gibt. Es ist äußerst schwierig, herauszufinden, was in einer Kalkulationstabelle tatsächlich vor sich geht. Jeder, der schon einmal mit einer Kalkulationstabelle gearbeitet hat, die von jemand anderem erstellt wurde, weiß nur zu gut, wie frustrierend es ist, die verschiedenen Berechnungen und Verknüpfungen zu entschlüsseln, die zur Durchführung von Analysen verwendet werden. Bereits kleine Kalkulationstabellen, die nur begrenzte Analysen durchführen, sind mühsam zu entziffern, und bei großen, aufwendigen Arbeitsmappen mit mehreren Arbeitsblättern wird es praktisch unmöglich, sie zu durchschauen, sodass man oft von vorne anfangen muss.

Im Vergleich zu Excel mögen Datenbanksysteme starr, streng und unnachgiebig in ihren Regeln erscheinen. Diese Unnachgiebigkeit hat jedoch auch einen Vorteil.

Da nur bestimmte Aktionen zulässig sind, können Sie leichter verstehen, was in strukturierten Datenbankobjekten wie Abfragen oder gespeicherten Prozeduren geschieht. Wenn ein Datensatz bearbeitet, eine Zahl berechnet oder ein Teil des Datensatzes als Teil eines analytischen Prozesses beeinflusst wird, können Sie diese Aktion leicht erkennen, indem Sie die Abfragesyntax oder den Code der gespeicherten Prozedur überprüfen. In einem relationalen Datenbanksystem gibt es keine versteckten Formeln, keine versteckten Zellen und keine toten benannten Bereiche.

Trennung von Daten und Präsentation

Daten sollten von der Präsentation getrennt sein. Sie wollen nicht, dass die Daten zu sehr an eine bestimmte Art der Präsentation gebunden werden. Wenn Sie zum Beispiel eine Rechnung von einem Unternehmen erhalten, gehen Sie nicht davon aus, dass die Finanzdaten auf dieser Rechnung die wahre Quelle Ihrer Daten sind. Es ist eine *Präsentation* Ihrer Daten, also eine spezifische *Darstellung*. Daten können Ihnen auf unterschiedliche Arten in

Diagrammen oder auf Websites präsentiert werden, aber solche Darstellungen sind nie die eigentliche Quelle der Daten.

Was genau hat dieses Konzept mit Excel zu tun? Menschen, die Datenanalysen mit Excel durchführen, neigen oft dazu, Daten, Analyse und Präsentation zu verschmelzen. Kennen Sie auch diese Excel-Arbeitsmappen mit 12 Arbeitsblättern, die jeweils einen Monat darstellen? Auf jedem Arbeitsblatt sind die Daten für einen Monat zusammen mit Formeln, Pivot-Tabellen und Zusammenfassungen aufgeführt. Was passiert, wenn Sie eine Zusammenfassung nach Quartal erstellen sollen? Fügen Sie weitere Formeln und Arbeitsblätter hinzu, um die Daten auf den einzelnen Monatsarbeitsblättern zu konsolidieren? Das grundlegende Problem in diesem Szenario besteht darin, dass die Arbeitsblätter eigentlich Datenwerte darstellen, die in die Präsentation der Analyse einfließen.

Hier geht es darum, dass Daten nicht an eine bestimmte Darstellung gebunden sein sollten, egal wie logisch oder nützlich diese auch erscheinen mag. In Excel geschieht dies jedoch ständig.

Da alle Arten und Phasen der Analyse direkt in einer Kalkulationstabelle durchgeführt werden können, kann Excel, wie bereits weiter oben in diesem Kapitel erwähnt, keine angemessene Transparenz für die Analyse bieten. Jede Zelle kann Formeln enthalten, ausgeblendet werden und Verknüpfungen zu anderen Zellen enthalten. Damit verschwimmt in Excel die Grenze zwischen Analyse und Daten, sodass schwer zu erkennen ist, was in einer Kalkulationstabelle vor sich geht. Außerdem ist es sehr mühsam, durch manuelle Pflege sicherzustellen, dass sich Bearbeitungen und unvorhergesehene Änderungen nicht auf frühere Analysen auswirken.

Relationale Datenbanksysteme trennen analytische Komponenten von Natur aus in Tabellen, Abfragen und Berichte. Durch die Trennung dieser Elemente machen Datenbanken Daten weniger anfällig für Änderungen und schaffen eine Datenanalyseumgebung, in der Sie problemlos auf neue Analyseanforderungen reagieren können, ohne frühere Analysen zu zerstören.

Vielleicht stellen Sie ja bereits fest, dass Sie Excel-Funktionen manipulieren, um sich diesem Datenbankverhalten anzunähern. Wenn Sie also die Funktionen von Excel nutzen, damit es sich wie eine Datenbankanwendung verhält, sollten Sie sich vielleicht überlegen, ob nicht eine echte Datenbankanwendung von Vorteil wäre. Die Nutzung von Datenbanken zur Datenspeicherung und für analytische Zwecke würde die Datenanalyse insgesamt verbessern und es den Excel-Kunden ermöglichen, sich auf die Darstellung in ihren Tabellen zu konzentrieren.

In Zeiten von Big Data verlangen die Kunden nicht weniger, sondern mehr komplexe Datenanalysen. Excel-Analysten müssen ihr Repertoire um weitere Tools erweitern, um nicht nur »Tabellenkalkulationsmechaniker« zu sein. Mit Excel lässt sich so gut wie alles machen, aber die Pflege solcher kreativen Lösungen kann mühsam sein, wenn sie manuell durchgeführt werden muss. Der attraktive Aspekt der Datenanalyse liegt also nicht in der routinemäßigen Datenverwaltung in Excel, sondern in der Nutzung von BI-Tools, um Kunden die beste Lösung für jede Situation zu bieten.

Die Datenbankterminologie

Die Begriffe *Datenbank*, *Tabelle*, *Datensatz*, *Feld* und *Wert* stehen für eine Hierarchie vom Größten zum Kleinsten. Diese Begriffe werden bei praktisch allen Datenbanksystemen verwendet, Sie sollten sie also gut kennen.

Datenbanken

Im Allgemeinen ist das Wort *Datenbank* ein Computerbegriff für eine Sammlung von Informationen zu einem bestimmten Thema oder einer Geschäftsanwendung. Eine Datenbank hilft Ihnen, diese zusammenhängenden Informationen in einer logischen Weise zu organisieren, um einen einfachen Zugriff und Abruf zu ermöglichen. In einigen älteren Datenbanksystemen wurde der Begriff Datenbank zur Beschreibung einzelner Tabellen verwendet. Die heutige Verwendung des Begriffs Datenbank bezieht sich auf alle Elemente eines Datenbanksystems.

Datenbanken sind nicht nur etwas für Computer. Manuelle Datenbanken werden manchmal auch als manuelle Ablagesysteme oder manuelle Datenbanksysteme bezeichnet. Diese Ablagesysteme bestehen normalerweise aus Menschen, Unterlagen auf Papier, Ordnern und Aktenschränken – Papier ist der Schlüssel zu einem manuellen Datenbanksystem. In einem realen manuellen Datenbanksystem haben Sie wahrscheinlich Posteingangs- und Postausgangskörbe und eine Art formale Ablagemethode. Sie greifen manuell auf Informationen zu, indem Sie einen Aktenschrank öffnen, einen Aktenordner herausnehmen und das richtige Blatt Papier suchen. Kunden füllen Papierformulare aus und geben die auf den Formularen gedruckten Informationen vielleicht über eine Tastatur ein. Sie finden Informationen, indem Sie die Papiere manuell sortieren oder Informationen von vielen Blättern auf ein anderes Blatt (oder sogar in eine Excel-Tabelle) übertragen. Sie können eine Tabellenkalkulation oder einen Taschenrechner verwenden, um die Daten zu analysieren oder sie auf neue und interessante Weise darzustellen.

Tabellen

Eine Datenbank speichert Informationen in einer sorgfältig definierten Struktur, die als *Tabelle* bezeichnet wird. Eine Tabelle ist nichts anderes als ein Behälter für Rohinformationen (die *Daten*), ähnlich einem Ordner in einem manuellen Ablagesystem. Jede Tabelle in einer Datenbank enthält Informationen über eine einzelne Einheit, zum Beispiel eine Person oder ein Produkt, und die Daten in der Tabelle sind in Zeilen und Spalten organisiert. Ein relationales Datenbanksystem speichert Daten in miteinander verknüpften Tabellen. So kann beispielsweise eine Tabelle mit Mitarbeiterdaten (Namen und Adressen) mit einer Tabelle mit Gehaltsabrechnungsdaten (Gehaltsdatum, Gehaltsbetrag und Schecknummer) verknüpft sein.

In der Datenbanksprache ist eine Tabelle ein *Objekt*. Beim Entwurf und bei der Arbeit mit Datenbanken ist es wichtig, jede Tabelle als spezifische Einheit zu betrachten und zu überlegen, in welcher Beziehung jede Tabelle mit den anderen Objekten in der Datenbank steht.

In den meisten Datenbanksystemen können Sie den Inhalt einer Tabelle in einer der Kalkulationstabelle ganz ähnlichen Form anzeigen, in einem sogenannten *Datenblatt*, das aus Zeilen und Spalten besteht (auch als *Datensätze* beziehungsweise *Felder* bezeichnet – siehe

folgender Abschnitt). Obwohl ein Datenblatt und eine Kalkulationstabelle oberflächlich betrachtet ähnlich sind, handelt es sich bei einem Datenblatt um eine ganz andere Art von Objekt. In der Regel können Sie in einer Tabelle nicht direkt Änderungen vornehmen oder Berechnungen hinzufügen. Ihre Interaktion mit Tabellen erfolgt in erster Linie in Form von Abfragen oder Ansichten – siehe den späteren Abschnitt »Abfragen«.

Datensätze, Felder und Werte

Eine Datenbanktabelle ist in Zeilen (*Datensätze* genannt) und Spalten (*Felder* genannt) unterteilt, wobei die erste Zeile (die Überschrift über jeder Spalte) die Namen der Felder in der Datenbank enthält.

Jede Zeile ist ein einzelner Datensatz mit Feldern, die sich auf diesen Datensatz beziehen. Stellt man sich das wieder im Vergleich zu einem manuellen System dar, sind die Zeilen einzelne Formulare (Papierblätter), und die Felder entsprechen den freien Bereichen auf einem ausgedruckten Formular, die Sie ausfüllen.

Jede Spalte ist ein Feld, das viele Eigenschaften enthält. Diese Eigenschaften beschreiben die Art der im Feld enthaltenen Daten sowie die Art und Weise, wie die Datenbank die Daten des Felds verarbeiten soll. Zu diesen Eigenschaften gehören der Name des Felds (zum Beispiel Firma) und die Art der Daten im Feld (zum Beispiel Text). Ein Feld kann auch andere Eigenschaften enthalten. Die Eigenschaft »Größe« des Felds »Adresse« könnte der Datenbank zum Beispiel die maximal zulässige Anzahl von Zeichen für die Adresse mitteilen.

An der Schnittstelle zwischen einem Datensatz und einem Feld befindet sich ein *Wert* – das eigentliche Datenelement. In einem Feld mit dem Namen »Firma« würde zum Beispiel ein in dieses Feld eingegebener Firmenname einen Datenwert darstellen.



Bei Microsoft Access wird der Begriff *Feld* verwendet, um ein in einem Datensatz gespeichertes Attribut zu bezeichnen. In vielen anderen Datenbanksystemen, einschließlich SQL Server, ist der Ausdruck *Spalte* anstelle von *Feld* gebräuchlicher – »Feld« und »Spalte« bedeuten dasselbe. Welche Terminologie genau verwendet wird, hängt vom Kontext des Datenbanksystems ab, das der Tabelle mit dem Datensatz zugrunde liegt.

Abfragen

Die meisten relationalen Datenbanksysteme ermöglichen die Erstellung von Abfragen (manchmal auch Ansichten genannt). Eine *Abfrage* extrahiert Informationen aus den Tabellen in der Datenbank – sie wählt und definiert eine Gruppe von Datensätzen, die eine bestimmte Bedingung erfüllen. Die meisten Datenbankausgaben basieren auf Abfragen, die Daten kombinieren, filtern oder sortieren, bevor sie angezeigt werden. Abfragen werden oft von anderen Datenbankobjekten wie gespeicherten Prozeduren, Makros oder Codemodulen aufgerufen. Neben der Extraktion von Daten aus Tabellen können Abfragen auch zum Ändern, Hinzufügen oder Löschen von Datenbankeinträgen verwendet werden.

Ein Beispiel für eine Abfrage ist, wenn eine Person im Vertriebsbüro der Datenbank mitteilt: »Zeige mir alle Kunden in alphabetischer Reihenfolge nach Namen, die in Weinheim ansässig sind und in den letzten sechs Monaten etwas gekauft haben.« Oder: »Zeigen Sie mir alle Kunden, die in den letzten sechs Monaten unser Modell ›Winterzauber‹ gekauft haben, sortiert nach Kundenname und dann nach Verkaufsdatum.«

Statt solche Fragen in natürlicher Sprache zu formulieren, wird eine spezielle Syntax verwendet, zum Beispiel SQL (Structured Query Language), um der Datenbank die Abfrage in einer für sie verständlichen Form mitzuteilen.

Beziehungen verstehen

Nachdem Sie die grundlegende Terminologie von Datenbanken verstanden haben, ist es nun an der Zeit, sich mit einer ihrer nützlichsten Funktionen zu befassen: Beziehungen (auch als *Relationen* bezeichnet! Durch das Erstellen einer *Beziehung* lassen sich separate Tabellen miteinander verknüpfen. Sie können sich eine Beziehung als eine Art SVERWEIS vorstellen, bei dem Sie die Daten in einem Datenbereich mit den Daten in einem anderen Datenbereich über einen Index oder einen eindeutigen Bezeichner verknüpfen. In Datenbanken bewirken Beziehungen das Gleiche, allerdings ohne das mühsame Schreiben von Formeln.

Beziehungen sind wichtig, weil die meisten Daten, mit denen Sie arbeiten, eine Art multidimensionale Hierarchie bilden. Sie könnten zum Beispiel eine Tabelle haben, in der Kunden gespeichert werden, die Produkte kaufen. Diese Kunden benötigen Rechnungen, die Rechnungsnummern enthalten. Diese Rechnungen enthalten mehrere Transaktionszeilen, in denen die gekauften Produkte aufgeführt sind. Es entsteht eine Hierarchie.

In der Welt der eindimensionalen Tabellenkalkulation würden diese Daten normalerweise in einer flachen Tabelle gespeichert, wie in Abbildung 1.1 dargestellt.

Da Kunden mehr als eine Rechnung erhalten, müssen die Kundeninformationen (in diesem Beispiel Kundennummer und Kundenname) wiederholt werden. Dies führt zu einem Problem, wenn diese Daten aktualisiert werden müssen.

Stellen Sie sich zum Beispiel vor, dass der Name des Unternehmens Aaron Fitz Elektro in Fitz und Söhne Elektro geändert wird. In Abbildung 1.1 erkennen Sie, dass mehrere Zeilen diesen alten Namen enthalten. Sie müssten sicherstellen, dass jede Zeile, die den alten Firmennamen enthält, aktualisiert wird, um die Änderung widerzuspiegeln. Jede Zeile, die Sie übersehen, wird nicht dem richtigen Kunden zugeordnet.

Wäre es nicht logischer und effizienter, den Namen und die Daten des Kunden nur einmal zu erfassen? Statt dieselben Kundeninformationen immer wieder erfassen zu müssen, könnte man dann einfach eine Art Kundenreferenznummer verwenden.

Genau das ist die Idee hinter Beziehungen. Sie können Kunden und Rechnungen voneinander trennen, indem Sie sie in jeweils eigenen Tabellen ablegen. Anschließend können Sie einen eindeutigen Bezeichner (zum Beispiel Kundennummer) verwenden, um sie miteinander zu verknüpfen.

	A	B	C	D	E	F
1	Kundennummer	Kundenname	Rechnungsnummer	Rechnungsdatum	Menge	Stückpreis
2	BAKERSEM0001	Bäckerei Blaubeer	ORDST2023000017	03.01.2023	1	19,95
3	BAKERSEM0001	Bäckerei Blaubeer	ORDST2023000018	04.01.2023	5	1759,95
4	BAKERSEM0001	Bäckerei Blaubeer	ORDST2023000019	05.01.2023	4	9,95
5	BAKERSEM0001	Bäckerei Blaubeer	STDINV2023000020	07.02.2023	4	9,95
6	AARONFIT0001	Aaron Fitz Elektro	ORDST2023000020	07.02.2023	5	9,95
7	AARONFIT0001	Aaron Fitz Elektro	ORDST2023000021	07.02.2023	3	1759,95
8	AARONFIT0001	Aaron Fitz Elektro	ORDST2023000022	07.02.2023	2	79,95
9	AARONFIT0001	Aaron Fitz Elektro	STDINV2023000028	13.03.2023	3	1759,95
10	AARONFIT0001	Aaron Fitz Elektro	STDINV2023000029	13.03.2023	5	9,95
11	METROPOL001	Metropolitan Fiber	ORDST2023000030	10.04.2023	1	9,95
12	AARONFIT0001	Aaron Fitz Elektro	INV1024	15.02.2023	1	119,95
13	AARONFIT0001	Aaron Fitz Elektro	INV1025	16.02.2023	1	109,95
14	LECLERC000001	LeClerc & Partner	ORDPH2023000022	13.03.2023	2	189,95
15	MAGNIFIC0001	Magnificent Office	ORD1000	08.05.2023	1	359,95
16	HOLLINGC0001	Holling Te-Kom	ORD1001	10.05.2023	2	59,95
17	MAHLERST0001	Mahler State Univer.	ORDST1008	10.05.2023	1	5999,95

Abbildung 1.1: In einer Excel-Tabelle werden Daten in einem flachen Tabellenformat gespeichert.

Abbildung 1.2 zeigt, wie diese Daten in einer relationalen Datenbank aussehen könnten. Die Daten würden in drei getrennte Tabellen aufgeteilt: Kunden, Rechnungskopf und Rechnungsdetails. Diese Tabellen würden dann über eindeutige Bezeichner (in diesem Fall Kundennummer und Rechnungsnummer) miteinander verknüpft.

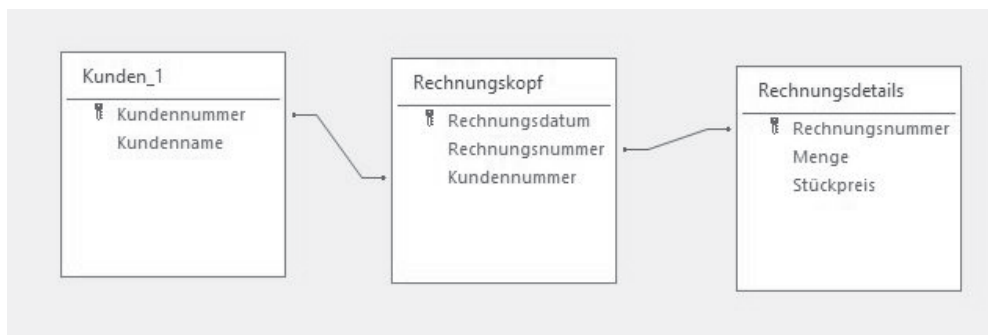


Abbildung 1.2: Datenbanken verwenden Beziehungen, um Daten in eindeutigen Tabellen zu speichern und diese Tabellen einfach zueinander in Beziehung zu setzen.

Die Tabelle Kunden würde für jeden Kunden einen einzigen Datensatz enthalten. Wenn Sie also den Namen eines Kunden ändern müssen, müssen Sie die Änderung nur in diesem Datensatz vornehmen. Im wirklichen Leben würde die Tabelle Kunden natürlich auch andere Attribute enthalten, zum Beispiel die Adresse des Kunden, seine Telefonnummer und das Datum, an dem er seine Tätigkeit aufnahm. Alle diese anderen Attribute könnten ebenfalls problemlos in der Tabelle Kunden gespeichert und verwaltet werden.

Der häufigste Beziehungstyp ist eine 1:n-Beziehung. Das heißt, ein Datensatz in einer Tabelle kann mit vielen Datensätzen in einer anderen Tabelle verknüpft werden. Beispielsweise könnte eine Rechnungskopftabelle mit einer Rechnungsdetailtabelle verknüpft sein.

Die Rechnungskopftabelle hat einen eindeutigen Bezeichner: Rechnungsnummer. In den Rechnungsdetails wird die Rechnungsnummer für jeden Datensatz verwendet, der ein Detail dieser spezifischen Rechnung darstellt.

Eine andere Art Beziehung ist die 1:1-Beziehung: Für jeden Datensatz in einer Tabelle gibt es genau einen zugehörigen Datensatz in einer anderen Tabelle. Daten aus verschiedenen Tabellen in einer 1:1-Beziehung können technisch in einer einzigen Tabelle kombiniert werden.

Bei einer n:n-Beziehung schließlich können Datensätze in beiden Tabellen beliebig viele passende Datensätze in der anderen Tabelle haben. Eine Datenbank bei einem Geldinstitut kann beispielsweise eine Tabelle mit den verschiedenen Kreditarten (Hauskredit, Autokredit und so weiter) und eine Tabelle mit den Kunden haben. Ein Kunde kann viele Arten von Krediten haben. Gleichzeitig kann jede Kreditart an viele Kunden vergeben werden.

Wenn Ihnen bei all diesen Informationen über Datenbanken der Kopf schwirrt, machen Sie sich keine Sorgen. Sie müssen kein Experte für Datenbankmodelle sein, um Power Pivot zu verwenden. Aber es ist wichtig, diese Konzepte zu verstehen. Je besser Sie verstehen, wie Daten in Datenbanken abgelegt und verwaltet werden, desto effektiver können Sie Power Pivot für Ihre Berichte nutzen.

