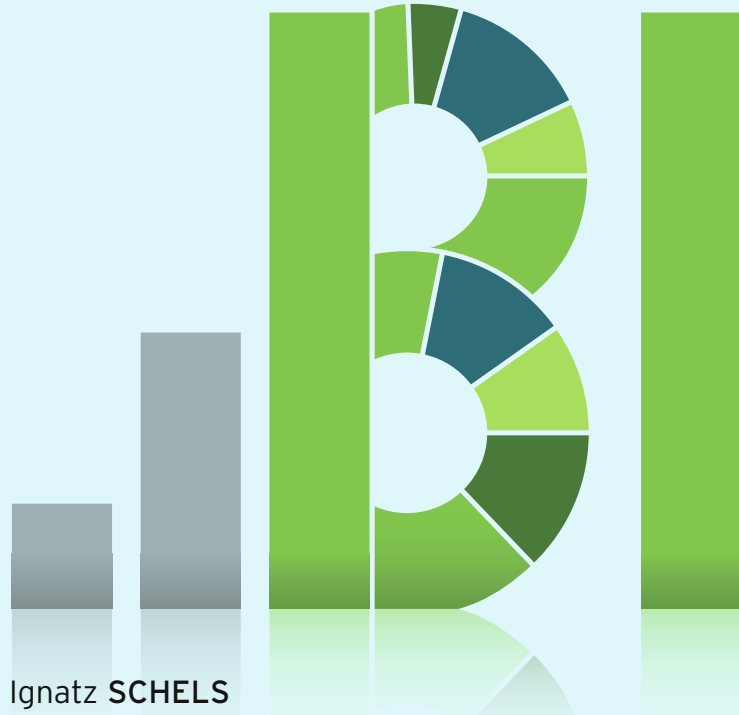


Für  
Microsoft  
**365**



Ignatz SCHELS

3. Auflage

# BUSINESS INTELLIGENCE MIT EXCEL

Datenanalyse und Reporting  
mit Power Query, Power Pivot  
und Power BI Desktop



Zusatzmaterial unter  
[plus.hanser-fachbuch.de](https://plus.hanser-fachbuch.de)

HANSER



## Business Intelligence mit Excel



### Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf [plus.hanser-fachbuch.de](https://plus.hanser-fachbuch.de) einfach diesen Code ein:

plus-23si4-X9u7m



### Bleiben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine.

Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

[www.hanser-fachbuch.de/newsletter](https://www.hanser-fachbuch.de/newsletter)





Ignatz Schels

# Business Intelligence mit Excel

Datenanalyse und Reporting  
mit Power Query, Power Pivot und  
Power BI Desktop

Für Microsoft 365

3., aktualisierte Auflage

HANSER

Der Autor:

*Ignatz Schels*, Wolnzach

Alle in diesem Werk enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Werk enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autoren und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die endgültige Entscheidung über die Eignung der Informationen für die vorgesehene Verwendung in einer bestimmten Anwendung liegt in der alleinigen Verantwortung des Nutzers.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – mit Ausnahme der in den §§ 53, 54 URG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2024 Carl Hanser Verlag München, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sylvia Hasselbach

Copy editing: Walter Saumweber, Sandra Gottmann

Umschlagdesign: Marc Müller-Bremer, [www.rebranding.de](http://www.rebranding.de), München

Umschlagrealisation: Max Kostopoulos

Satz: Eberl & Koesel Studio, Kempten

Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Göttingen

Printed in Germany

Print-ISBN: 978-3-446-47952-4

E-Book-ISBN: 978-3-446-48002-5

E-Pub-ISBN: 978-3-446-48003-2

# Inhalt

<b>Vorwort</b> .....	<b>XIII</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Excel im Business Intelligence – eine Chance .....	2
1.2 Werkzeuge und Methoden .....	3
1.2.1 Power Query .....	4
1.2.2 Power Pivot .....	5
1.2.3 Power BI .....	6
1.2.4 Power View .....	8
1.2.5 Power BI oder Excel? .....	8
1.2.6 Power Pivot oder Datenbank? .....	9
<b>2 Grundlagen, Verfahren und Konzepte</b> .....	<b>11</b>
2.1 Relationale und multidimensionale Datenbanken .....	11
2.2 Das Entity-Relationship-Modell (ERM) .....	12
2.3 Normalisierung und Faktentabellen .....	12
2.4 Sternschema .....	14
2.5 OLAP – die multidimensionale Datenbank .....	15
2.5.1 Von OLAP zu BI .....	15
2.5.2 Wie ein OLAP-Cube funktioniert .....	15
2.6 Datenimport und dynamische Verknüpfungen .....	20
2.6.1 ODBC .....	20
2.6.2 ODBC-Treiber verwalten .....	20
2.7 ODBC und Power Query .....	21
2.8 SQL .....	22
2.8.1 SQL-Abfragen mit Excel .....	22
<b>3 Das Datenmodell</b> .....	<b>35</b>
3.1 Regeln für Datenmodelle .....	35
3.2 Spezifikationen und Einschränkungen .....	36
3.2.1 Objekte im Datenmodell .....	36

3.3	Datenmodell anlegen .....	37
3.3.1	Über die PivotTable .....	37
3.3.2	Aus Tabellen über Power Pivot .....	39
3.3.3	Implizites und explizites Datenmodell .....	40
3.3.4	Aus externen Quellen .....	44
3.3.5	Von der Excel-Liste zum Datenmodell .....	46
3.3.6	Datenmodell entfernen .....	47
3.3.7	Datenmodelle zusammenführen .....	47
3.4	Datenmodelle komprimieren und Speicherplatz sparen .....	48
3.4.1	Nur importieren, was nötig ist .....	48
3.4.2	Weniger eindeutige Werte .....	49
3.4.3	SQL DISTINCT verwenden .....	49
3.4.4	Nur die benötigten Tabellen aus der Datenbank .....	49
3.4.5	Spaltenzahl reduzieren .....	50
3.5	Optionen für den Datenimport .....	53
<b>4</b>	<b>Abrufen und Transformieren (Power Query) .....</b>	<b>55</b>
4.1	Menüband und Symbole .....	55
4.2	ETL - der Prozess .....	56
4.3	Power Query installieren .....	58
4.4	Daten abrufen und transformieren .....	58
4.4.1	Aus Datei .....	59
4.4.2	Aus Datenbank .....	60
4.4.3	Aus Azure .....	60
4.4.4	Aus Power BI-Plattform .....	60
4.4.5	Aus Online-Diensten .....	61
4.4.6	Aus anderen Quellen .....	62
4.5	Datenquelleneinstellungen und Abfrageoptionen .....	65
4.5.1	Daten laden .....	65
4.5.2	Power Query-Editor .....	65
4.5.3	Sicherheit .....	66
4.5.4	Datenschutz .....	66
4.5.5	Regionale Einstellungen .....	67
4.5.6	Diagnose und Ablaufverfolgung .....	67
4.5.7	Aktuelle Arbeitsmappe .....	67
4.6	Abfrage erstellen .....	68
4.6.1	Eine neue Abfrage .....	69
4.6.2	Menü Abfrage/Abfrage bearbeiten .....	76
4.7	Der Power Query-Editor .....	76
4.7.1	Die Elemente .....	77
4.7.2	Abfragesprache einstellen .....	78
4.7.3	Eigenschaften .....	78
4.7.4	Abfrage abbrechen .....	78
4.7.5	Abfrage speichern .....	79



4.7.6	Transformationen anwenden	79
4.7.7	Gruppieren	85
4.7.8	Daten transformieren und entpivotieren	88
4.7.9	Abfrageschritte bearbeiten	90
4.7.10	Schritte benennen	97
4.7.11	Abfragen kombinieren	97
4.7.12	Abfragen kombinieren mit Fuzzy	103
4.7.13	Abfragen anfügen	107
4.8	Abfragen kopieren und transferieren	110
4.8.1	M-Code transferieren	110
4.8.2	Datenvorschau	111
4.8.3	Abfrage kopieren	111
4.8.4	Abfragen als ODC-Datei speichern	114
4.8.5	Abfrage an den Azure-Datenkatalog senden	116
4.9	Berechnete Spalten	119
4.9.1	IntelliSense aktivieren	119
4.9.2	Das Prinzip	121
4.10	Mit Parametern arbeiten	127
4.10.1	Abfrage mit Parameter auf Pfad und Dateiname	131
4.10.2	Abfrage verfeinern mit M-Code	139
4.10.3	Parameterabfrage mit M-Funktion	141
4.10.4	Parameterabfrage mit Funktion und WENN-Filter	141
4.11	Die Abfragesprache M	143
4.11.1	Die Bearbeitungsleiste	144
4.11.2	Erweiterter Editor	145
4.11.3	Die Syntax der M-Formel	146
4.11.4	Funktionsreferenzen und Hilfe	153
4.11.5	Abfrage auf Funktionsliste erstellen	153
4.11.6	Datumsberechnungen mit M	154
4.11.7	Kalendertabelle aus Start-/Enddatum erstellen	159
4.11.8	Eigene Funktionen schreiben	162
4.12	KI-Funktionen	164
4.12.1	Spalte aus Beispielen	164
4.12.2	KI-Insights	166
4.13	Power Query-Praxis	166
4.13.1	Fehlzeitenanalyse	166
4.13.2	Richtig filtern	172
4.13.3	Webabfrage mit Formularelement kombiniert	177
4.13.4	Vorsicht bei automatischer Typerkennung	184
4.13.5	Abfragen kombinieren mit Joins	185
4.13.6	Zellwerte in Abfragen verwenden	191
4.13.7	Daten aus Zellen im Kopfbereich übernehmen	196
4.13.8	Daten mit variablen Spaltenüberschriften	200
4.13.9	Dateien aus Ordnern	202

4.13.10	Zuletzt gespeicherte Datei importieren	206
4.13.11	Quelldaten mit unterschiedlichen Spaltenüberschriften	209
4.13.12	SAP-Berichte	214
4.14	Power Query und VBA	220
4.14.1	VBA lernen	220
4.14.2	Abfragen aktualisieren per VBA	221
4.14.3	Abfragen und Ereignisse	222
4.14.4	Abfrageskript per VBA bearbeiten	224

## **5 Power Pivot – Daten modellieren im Datenmodell** ..... 227

5.1	Grundlagen	227
5.1.1	Das kann Power Pivot	227
5.1.2	Technische Voraussetzungen	228
5.1.3	Office-Version	228
5.2	Power Pivot starten	229
5.2.1	Zurück zu Excel	229
5.3	Die Power Pivot-Menüs	230
5.3.1	Das Datei-Menü	230
5.3.2	Das Menü „Start“	231
5.3.3	Das Menü „Entwurf“	240
5.3.4	Das Menü „Erweitert“	244
5.4	Daten importieren	248
5.4.1	Zwischenablage	248
5.4.2	Aus Datenbank – Aus SQL Server	250
5.4.3	Aus Datenbank – Aus Access	258
5.4.4	Aus Datenbank – Analysis Services	261
5.4.5	Aus OLAP-Cubes per MDX	262
5.4.6	Aus Datendienst	266
5.4.7	Aus Azure SQL	272
5.4.8	Aus Excel	281
5.4.9	Aus Text oder CSV	284
5.4.10	Aus anderen Quellen	286
5.4.11	SQL-Abfragen	287
5.5	Tabelleneigenschaften	289
5.5.1	Datumstabelle	290
5.5.2	Die Tabellenregister	310
5.6	Arbeiten mit Spalten	311
5.6.1	Spalte einfügen	312
5.6.2	Spalten formatieren	314
5.6.3	Spalte löschen	316
5.6.4	Beschreibungen für Spalten und Measures	318
5.7	Hierarchien	318
5.7.1	Hierarchien erstellen	319
5.7.2	Hierarchien in PivotTables verwenden	321

5.8	Measures erstellen und verwalten	325
5.8.1	Implizite Measures	326
5.8.2	Diskrete Anzahl	329
5.8.3	Explizite Measures	332
5.9	Time-Intelligence und KPIs	339
5.10	PivotTable in Cube-Funktionen umwandeln	344
5.10.1	Cube-Funktionen-Übersicht	344
5.10.2	Beispiel: Firmenumsätze	345
5.11	Einführung in DAX	350
5.11.1	Was ist DAX?	350
5.11.2	DAX-Funktionen erstellen	351
5.11.3	DAX-Funktionen im Kontext	356
5.11.4	Die wichtigsten DAX-Funktionen	362
5.11.5	Hilfe zu DAX-Funktionen	365
5.12	Praxisbeispiel: Vorjahresvergleich	366
5.12.1	Die Datenquelle	367
5.12.2	Die Datumstabelle	368
5.12.3	Relationen	369
5.12.4	DAX-Measures	369
5.12.5	PivotTable und PivotChart	370
5.12.6	Visualisieren mit Power BI	371
<b>6</b>	<b>Karten und 3D-Karten</b>	<b>373</b>
6.1	Kartendiagramme	373
	Beispiel: Einwohner pro Bundesland	374
	Bundesländer mit Kategorie-Einfärbung	375
6.2	3D-Karten	377
<b>7</b>	<b>BI-Analysen mit PivotTables und PivotCharts</b>	<b>381</b>
7.1	Das Pivot-Prinzip	383
7.1.1	Von der Filtersumme zur PivotTable	383
7.1.2	Voraussetzungen	386
7.1.3	Auswertbare Daten	387
7.2	Datenquellen für PivotTables	389
7.2.1	Listen	389
7.2.2	Tabellen manuell erstellen	391
7.2.3	MS Query	393
7.2.4	Power Query	394
7.2.5	Power Pivot	397
7.3	PivotTable-Berichte erstellen	398
7.3.1	Berichte planen	398
7.3.2	PivotTables positionieren	399
7.4	Das Berichtslayout	402

7.4.1	Die Feldliste	403
7.4.2	Layouts und Formate	404
7.4.3	Layout als Standard speichern	407
7.4.4	Wertfeldeinstellungen	408
7.4.5	Zahlenformate	410
7.4.6	Bedingte Formatierung	410
7.4.7	PivotTables aktualisieren	411
7.4.8	Datenquelle ändern	414
7.4.9	PivotTable verschieben und löschen	414
7.4.10	Drilldown (Details anzeigen)	415
7.5	Datenschnitte	416
7.5.1	Ein neuer Datenschnitt	416
7.5.2	Datenschnitt formatieren	417
7.5.3	Datenschnittverbindungen	418
7.6	Die Zeitachse	419
7.6.1	Eine neue Zeitachse	419
7.6.2	Zeitachse bedienen	419
7.7	Berechnete Felder	420
7.7.1	Berechnete Felder und Elemente in der PivotTable	420
7.7.2	Berechnete Felder mit strukturierten Verweisen	424
7.7.3	Berechnete Felder mit Power Query	427
7.7.4	Berechnete Felder mit DAX in Power Pivot	432
7.8	Gruppieren	436
7.8.1	Textfeld gruppieren	436
7.8.2	Wertefeld gruppieren	439
7.8.3	Datumfelder gruppieren	439
7.9	PivotCharts	441
7.9.1	Das Prinzip	442
7.9.2	Diagrammtypen	445
7.9.3	Beispiel: Altersstrukturanalyse	446
7.9.4	Kennzahlcockpits mit PivotCharts	453
<b>8</b>	<b>Power BI</b>	<b>459</b>
8.1	Das Prinzip	460
8.2	Power BI und Excel	461
8.3	Power BI Desktop	462
8.3.1	Komponenten und Ansichten	463
8.3.2	Daten importieren	465
8.3.3	Daten modellieren mit Power Query	465
8.3.4	Eine Datumstabelle mit DAX	468
8.3.5	Beziehungen und Measures mit Power Pivot	470
8.3.6	Visualisierungen	471
8.3.7	Interaktive Filter	476

8.4	Mit Power BI Services in der Cloud .....	477
8.4.1	Bericht veröffentlichen .....	477
8.5	Von Excel zu Power BI .....	479
8.5.1	Excel-Datenmodell importieren .....	480
8.5.2	Excel-Arbeitsmappe veröffentlichen .....	481
8.5.3	Excel-Daten exportieren .....	484
8.5.4	Excel-Daten mit Power BI laden .....	485
8.5.5	Power BI in Excel analysieren .....	486
<b>9</b>	<b>Dynamische Arrays .....</b>	<b>491</b>
9.1	Das Prinzip .....	491
9.1.1	#Überlauf! .....	492
9.1.2	Implizierte Schnittmenge .....	492
9.1.3	#-Operator .....	493
9.2	Dynamische Array-Funktionen .....	493
9.2.1	EINDEUTIG() .....	494
9.2.2	SORTIEREN() .....	494
9.2.3	FILTER() .....	495
9.2.4	SPALTENWAHL() .....	496
9.2.5	SEQUENZ() .....	497
9.2.6	TEXTVOR(), TEXTNACH(), TEXTTEILEN() .....	498
9.2.7	Alternative zur PivotTable .....	499
9.2.8	Stapeln .....	500
9.3	Diagramme und dynamische Arrays .....	501
9.3.1	Beispiel: Umsatzanalyse Sport .....	502
9.3.2	Datenquelle berechnen .....	502
9.3.3	Diagrammdaten auswählen .....	503
<b>Index .....</b>	<b>505</b>	



# Vorwort

Liebe Lesende,

herzlich willkommen zur mittlerweile dritten Auflage von „Business Intelligence mit Excel“. Wir – Autor und Verlag – haben keinen Aufwand gescheut, um Ihnen die aktuellen Entwicklungen rund um das Thema „Business Intelligence mit Excel“ nahezubringen. Und das ist nicht immer einfach. Microsoft macht es uns Informationsbrokern nicht leicht, aktuell zu sein. Jährliche Software-Updates, neue Funktionen und geänderte Oberflächen sind eine Herausforderung für Autoren und Leser gedruckter Publikationen.

Sie haben sich also entschlossen, in „Business Intelligence“ einzusteigen. Vielleicht haben Sie ja schon erste Versuche mit BI-Tools gemacht und wollen Ihr Wissen jetzt vertiefen. Oder Sie gehören zu den klassischen Excel-Anwendern und suchen nach Lösungen, um Ihre Arbeit zu optimieren. Es kann auch sein, dass Sie den Auftrag bekommen haben, sich im Markt nach einer geeigneten Lösung für ein BI-basiertes Reporting umzusehen, und jetzt wollen Sie wissen, ob das auch mit Excel machbar ist und wo die Grenzen sind.

In jedem Fall sind Sie richtig beraten, sich mit den BI-Tools von Excel (und mit diesem Buch) zu beschäftigen. Die Tabellenkalkulation ist nach wie vor das wichtigste Werkzeug für Datenverarbeiter am PC und hat mit Business Intelligence massiv an Bedeutung gewonnen. Neue Tools sind an Bord – Power Query, Power Pivot, Power BI –, behalten Sie den Überblick und informieren Sie sich mit diesem Buch über Vor- und Nachteile, Einsatz und Nutzen der einzelnen Werkzeuge.

Für den praktischen Nutzen ist gesorgt. Wir sind seit Jahren zum Thema *Business Intelligence mit Excel* in Seminaren, Inhouse-Schulungen und Workshops unterwegs und haben bereits mehrere Projekte auf Basis der BI-Tools erfolgreich abgeschlossen. Schauen Sie auf unserer Webseite vorbei und schreiben Sie uns. Wir beraten Sie gerne:

Wir – das sind Ignatz Schels und Ignaz Schels junior. Von Ignaz Schels junior stammt das Buch zur neuen Programmiersprache für die BI-Tools,

*M – Daten abfragen und verarbeiten mit Excel und Power BI,*

das ebenfalls im Hanser-Verlag erschienen ist (ISBN 978-3-446-45588-7).



[www.schels.de](http://www.schels.de)

### Die Beispiele im Buch ...

... sind praxisnah und mit viel Erfahrung ausgearbeitet. Zu den wichtigsten Themen finden Sie immer eine oder mehrere Übungen mit Schritt-für-Schritt-Anleitungen, und um diese nachzuvollziehen, brauchen Sie Beispieldaten. Diese stehen auf der Webseite des Verlags zum Download bereit:



#### Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf [plus.hanser-fachbuch.de](http://plus.hanser-fachbuch.de) einfach diesen Code ein:

plus-23si4-X9u7m

Zu jeder Übung finden Sie eine von externen Verknüpfungen freie Basisdatei sowie eine Lösungsdatei. Da diese in der Regel Verbindungen zu Datenquellen enthält, beziehen sich alle Verknüpfungen auf ein einheitliches Verzeichnis. Kopieren Sie deshalb alle Beispieldaten in diesen Ordner auf Ihrem Root-Verzeichnis (C:), dann funktionieren auch die Lösungen problemlos:

`C:\HanserExcelBI\`

Beispieldateien haben beliebige Bezeichnungen wie *Umsatz2018.csv* oder *Getränkemarkt.xlsx*. Lösungen beginnen immer mit dem Wort *Lösungen*:

*Absatzanalyse Sportbekleidung.xlsx*

→ *Lösung Absatzanalyse Sportbekleidung.xlsx*

Jetzt wünsche ich Ihnen viel Spaß und viel Erfolg bei Ihrer Aufgabe, „Business Intelligence mit Excel“ kennenzulernen. Mit diesem Buch wird Ihnen das gelingen!

*Ihr Autor Ignatz Schels*



# 1

## Einleitung

### **Business Intelligence – was ist das?**

Die offizielle Erklärung liefert wie immer Wikipedia, das kluge Weltlexikon:

*Geschäftsanalytik, englisch Business Intelligence (Abkürzung BI) ist ein der Wirtschaftsinformatik zuzuordnender Begriff, der Verfahren und Prozesse zur systematischen Analyse des eigenen Unternehmens bezeichnet. Dies umfasst die Sammlung, Auswertung und Darstellung von Daten in elektronischer Form.*

Ziel des Business Intelligence ist, Wissen über den gegenwärtigen Status und die zukünftigen Perspektiven des eigenen Unternehmens wie auch des geschäftlichen Umfeldes zu erzeugen. Die Erkenntnisse dienen der Unterstützung von Managemententscheidungen, die Methoden umfassen das Sammeln der Daten (Data Mining), das Aufbereiten und Analysieren der Daten und das Visualisieren über Dashboards.

Der Hype um das Thema ist groß und wird die nächste Zeit noch größer werden. Seit 2016 steigt die Zahl der Unternehmen, die sich die Entwicklung und Bereitstellung eines BI-Systems auf die To-do-Liste geschrieben haben, rasant an. Analysten sagen voraus, dass die Unternehmen ihre IT in den nächsten 10 Jahren vollständig von zentrierter Berichtsplattform in moderne, BI-unterstützte Analyseplattformen umrüsten werden.

Auch den Anbietern von BI-Software werden goldene Zeiten prognostiziert. Firmen wie Tableau, QlickView oder prevero haben sich auf BI spezialisiert, ERP-Anbieter wie SAP und Oracle führen BI-Lösungen im Portfolio. Microsoft rüstet Dynamics und SQL Server mit BI-Fähigkeiten aus und bietet mit Power BI eine vollständige Lösung von der Desktop-Gestaltungssoftware bis zur Cloud-Integration. Microsoft Office und speziell Excel wird im Power BI-Konzept vollständig integriert und spielt in diesem eine wichtige Rolle.

### **... und Self-Service Business Intelligence?**

Das ist das Verfahren, bei dem der Anwender weitgehend eigenständig und unabhängig von der IT-Abteilung Daten importiert, aufbereitet und visualisiert. Die Voraussetzungen müssen natürlich geschaffen sein. Das Data Warehouse sollte verfügbar sein, Zugriffsrechte sollten eingeräumt und die Aktualität der Daten gesichert sein. Ist der Anwender mit den passenden Werkzeugen ausgestattet, entlastet er die IT-Abteilung von Routineaufgaben, bringt seine eigenen Daten mit ein und erhöht so die Aktualität und Qualität seiner Berichte.

Excel ist mehr dem *Self Service Business Intelligence* zuzuordnen. Die Einbindung der Office-App in die Cloud mit Power BI bietet größtmögliche Freiheit und Individualität in der Daten-

analyse und im Reporting. Damit sind auch kleinere Lösungen realisierbar, denn nicht jede Aufgabe erfordert gleich ein umfangreiches BI-Konzept.

## ■ 1.1 Excel im Business Intelligence – eine Chance

Für Excel-Anwender wird es höchste Zeit, aus der Bastellecke herauszukommen, und BI ist die Lösung dafür. Obwohl das allgemeine Know-how mit der Zeit gestiegen ist, sehen viele Kalkulationen immer noch wie „Malen nach Zahlen“ aus. Häufig werden Tabellen ohne Einhaltung simpelster Regeln der Datenbankentwicklung als Datenbanken missbraucht, mit dem Effekt, dass Excel-User immer noch einen großen Anteil ihrer Zeit damit verbringen, Daten per Copy-and-paste zwischen Zellen zu verschieben.

Die Tatsache, dass in vielen Tabellenmodellen die Verweisfunktionen (SVERWEIS(), XVERWEIS()) inflationär oft zum Einsatz kommt, deutet darauf hin, dass sehr häufig relationale Verbindungen zwischen Datenmengen benötigt würden, die Anwender aber weder die Ahnung noch die technischen Möglichkeiten haben, diese über eine Datenbank vorzubereiten und die darin professionell aufbereiteten Daten nach dem Client-Server-Prinzip dynamisch in ihre Auswertungen zu holen.

### Die Zukunft

Alle Daten, die in Fachbereichen wie Controlling, Finanzbereich, Personalverwaltung, Marketing u. a. benötigt werden, befinden sich an zentralen Stellen, die für jeden Anwender zugänglich sind. Das kann ein gemeinsamer Ordner im Netz sein, in dem Exportdateien, Excel-Arbeitsmappen, CSV-Listen, Textdaten und alle anderen Datenbestände abgelegt werden. Idealerweise richtet man eine Plattform unter dem Dokumentensystem SharePoint ein und vergibt Berechtigungen für alle Beteiligten.

Die Daten werden zentral gepflegt und aktualisiert, es gibt keine *Datasets* mehr und keine Altbestände oder Kopien. Alle Datenbestände sind tagesaktuell. Für den Datenabzug aus Datenbanken erhalten die Anwender die Berechtigungen, die Sie brauchen, um ihre Daten abzuholen. SQL Server, Access, Oracle bieten zu jeder Zeit Zugriff auf Abfragen und Sichten. In OLAP-Datenbanken wie *SQL Server Analysis Services (SSAS)* werden passende Cubes eingerichtet, aus denen Anwender zu jeder Zeit Daten extrahieren können. Für ERP-Systeme (SAP, Microsoft Dynamics, Salesforce etc.) richtet der Admin Online-Zugriffe ein oder er exportiert die Daten in regelmäßigem Turnus.

Mitarbeiter der Fachabteilungen dürfen dabei aber nicht gezwungen sein, sich mit IT-spezifischen Strukturen vertraut zu machen, um die auf unterschiedlichste Quellen verteilten Daten nutzen zu können. Es empfiehlt sich, qualifizierte Kräfte zwischen die IT-Abteilung und die Fachabteilungen zu schalten. Diese Mitarbeiter sollten die Sprache der IT-Profis ebenso beherrschen wie die der Fachabteilungen. So würde dann beispielsweise von der IT-Abteilung eine Cube-Analyse mit Dimensionen und Measures und detaillierten Tupeln angefordert und dem Controller ein Soll/Ist-Vergleich mit Year-to-date, Vorjahreswerten, Forecast und Budgets präsentiert.

## Risikovermeidung und Sicherheit

Business Intelligence eröffnet dem Excel-Anwender nicht nur neue Möglichkeiten, es zwingt ihn auch in eine Arbeitsweise, in der individuelle Tabellenkonstrukte keinen Platz mehr haben. Wirtschaftsprüfer und Revisoren können ein Lied singen von undurchschaubaren Excel-Patchworks mit Monsterformeln und wilden Bezügen. In Zukunft gibt es nur noch transparente Daten und ebensolche Visualisierungen. Das Risiko von Fehlberechnungen wird vermindert, Datenanalysen werden transparenter und sicherer.

## Umdenken: Vom Einzelkämpfer zum Teamplayer

Der berühmte Einzelkämpfer hat ausgedient. Excel-Spezialisten, die zwar ein hohes Niveau bezüglich Funktionen, Matrixtechniken und Formelkonstruktion haben, ihr Wissen aber wie die Goldreserven von Fort Knox hüten, sind nicht mehr angesagt. Teamarbeit ist angesagt. Alle Mitarbeiter müssen sich regelmäßig gegenseitig über Datenaktualität und Datenbestände informieren, Aktionen absprechen und Ergebnisse teilen. Dass bei dieser Teamarbeit so mancher Report nicht doppelt gemacht und so manche Power Query-Abfrage oder DAX-Funktion geteilt werden kann, ist ein gewollter Nebeneffekt.

## Neue Aufgaben – neue Ausbildung

Power Query, Power Pivot und Power BI sind für die meisten Anwender neu und unerforscht, und die vielen neuen Begriffe, die Aufgabe, neue Skriptsprachen (M) und neue Funktionen (DAX in Englisch!) lernen zu müssen, sind auch nicht gerade akzeptanzfördernd. Die Unternehmen sollten das durch gezielte Weiterbildung fördern, die nicht nur Excel, sondern das gesamte BI-Konzept zum Inhalt hat. Das klassische Excel-Seminar hat ausgedient, der BI-Workshop ist das neue Medium für Wissenstransfer.

# ■ 1.2 Werkzeuge und Methoden

Die Zeit, die der Excel-Anwender aufwendet, um seine Daten berichts-fähig zu machen, ist verlorene Zeit. Sie muss nämlich für jeden Bericht neu investiert werden. So manches lässt sich zwar schon automatisieren mit Excel-Funktionen (SVERWEIS!) und PivotTables, die nur noch kurz aktualisiert werden müssen. In der Praxis wollen die Berichte aber immer wieder auf die aktuellen Daten angepasst werden. Die eigentliche Aufgabe des Reportings, nämlich dem Berichtsempfänger die Informationen zu liefern, die er für die Entscheidungsfindung braucht, kommt in diesem Fall zu kurz. Neue Besen müssen also her, um die alten, überholten Arbeitstechniken auszukehren:

- Das Importieren der Daten und das Bereinigen, Ausputzen, Reparieren und Verschönern der Daten erledigt Power Query. Das Werkzeug ist bereits in Excel 2010 als Add-in verfügbar, ab Excel 2013 heißt es *Abrufen und Transformieren* (eine passende Beschreibung, aber der Begriff Power Query ist griffiger, weshalb wir in diesem Buch bei diesem bleiben).
- Mit *Power Pivot* werden Daten ähnlich wie in Datenbanken über Schlüssel miteinander verknüpft. Die Formelsprache DAX liefert die Funktionen für zusätzliche Berechnungen.

- *Power BI* ist der neue Star am Reporting-Himmel: Das Produkt gehört zwar weder zu Excel noch zum Microsoft Office- bzw. Microsoft 365-Paket, wird aber in Zukunft eine bedeutende Rolle bei *Business Intelligence mit Excel* spielen. Excel arbeitet eng mit *Power BI* zusammen. Excel-Daten werden in Dateiform als Datenquellen zur Verfügung gestellt oder dynamisch in Berichte und Dashboards eingebunden. Mit dem Add-in *Power BI Publisher* werden Excel-Daten eng mit *Power BI* verknüpft.

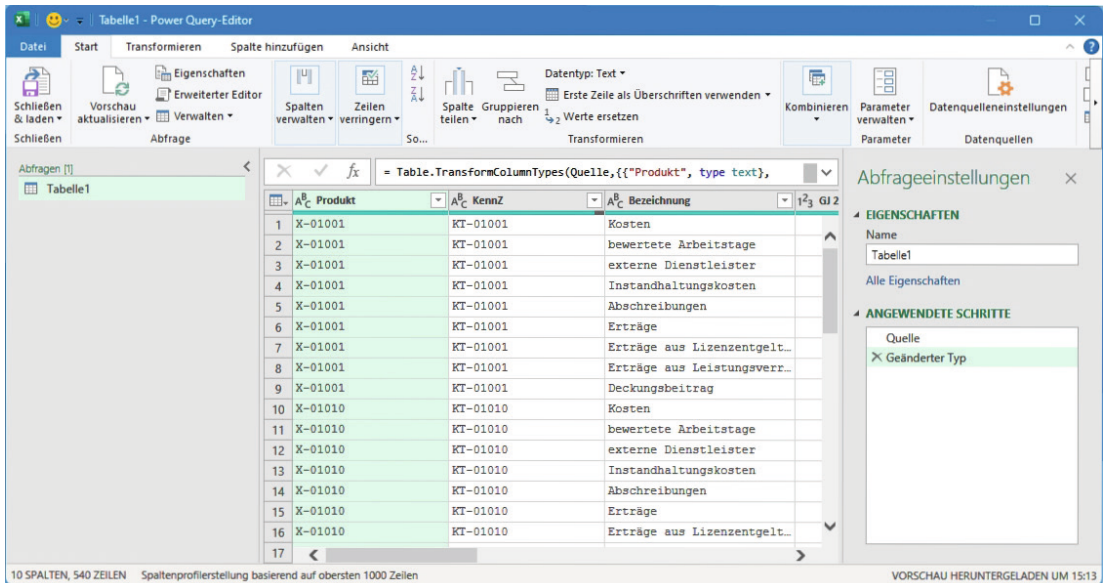
### 1.2.1 Power Query

Power Query ist das Datenabfrage-Tool, das für den ETL-Prozess generiert wurde und diesen mit genialen Werkzeugen unterstützt. ETL steht für *Extract, Transform* und *Load*. Mit Power Query werden die Daten aus unterschiedlichsten Datenquellen importiert (Extract), bereinigt, umgewandelt, modelliert, neu berechnet oder kombiniert (Transfer) und in das Datenmodell für die BI-Analyse geladen (Load). Aus dem Datenmodell entstehen im letzten Schritt Berichte und Dashboards, aber dafür ist wieder ein anderes Tool (Power BI) zuständig.

Power Query-Erzeugnisse generieren dynamische Abfragen auf die Daten. Für eine einmal modellierte Datenmenge genügt ein Knopfdruck, und alle Änderungen aus den Quelldaten werden automatisch aktualisiert.

Power Query kommt mit einer eigenen Programmiersprache namens M. Alle Schritte, die im Rahmen des ETL-Prozesses nötig sind, um die Daten bereitzustellen, werden in einer Art Protokoll aufgezeichnet, und für die Digitalisierung der einzelnen Aktionen verwendet M eine spezifische Sprache mit entsprechend großem Sprachumfang. Wer sich bereits mit VBA, der Makrosprache von Excel, beschäftigt hat, kennt den Makrorecorder. Er zeichnet die Aktionen des Anwenders auf und wandelt sie in VBA-Befehle um. Das Ergebnis ist ein Makro, das alle Aktionen „auf Knopfdruck“ wieder ausführt.

Power Query-Abfragen entstehen ähnlich: Der Anwender importiert Daten, bereinigt diese, filtert und sortiert, passt Datentypen an, entfernt Spalten und berechnet zusätzliche Spalten. Alles wird Schritt für Schritt aufgezeichnet, und zum Schluss hält der Anwender eine Abfrage in den Händen, die nur noch aktualisiert werden muss, um alle diese Schritte wieder neu auszuführen. Die Programmiersprache M bietet darüber hinaus noch zusätzlich Elemente wie Variablen, Parameter, logische Strukturen und eine Art Schleifen an, mit denen die aufgezeichneten Abfragen noch verfeinert und verbessert werden können.

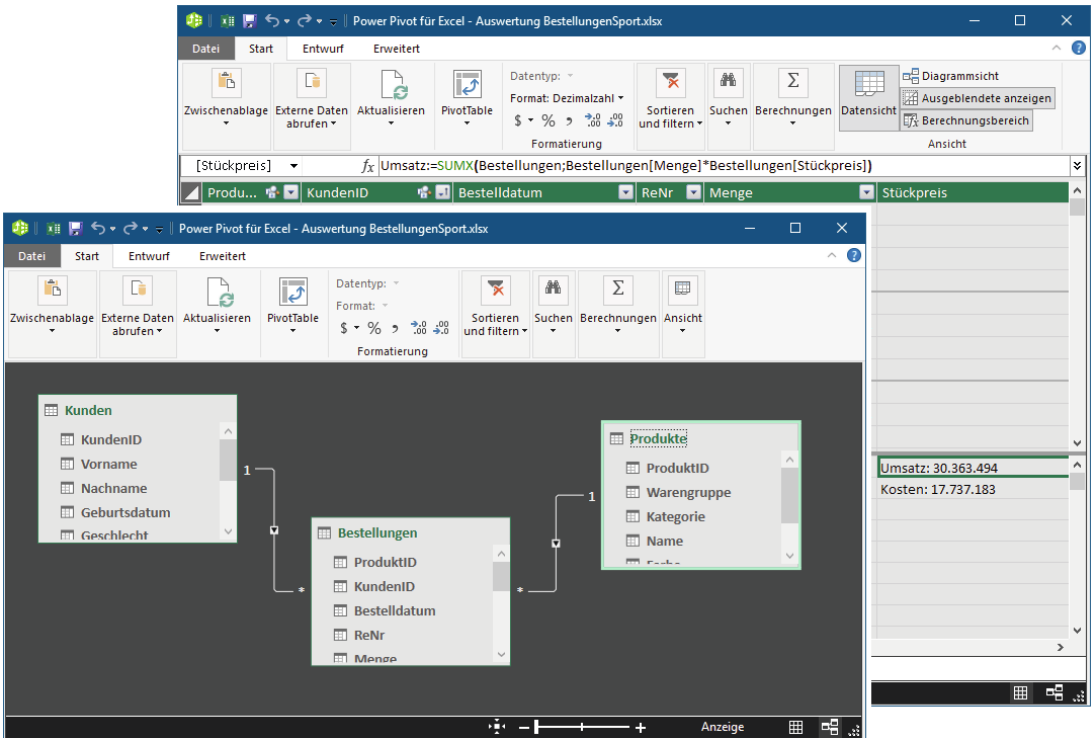


**Bild 1.1** Der Power Query-Editor

## 1.2.2 Power Pivot

Mit Power Pivot bietet Excel ein Tool für die Integration großer Datenmengen aus unterschiedlichsten Quellen (Datenbanken, Excel-Mappen, Text/CSV, Internet u. a.). Power Pivot verwaltet die Daten in Form von Tabellen. Verknüpfungen in importierten Datenbank-Daten werden automatisch erkannt und übernommen, mit Hilfe einer Diagrammansicht können unterschiedlichste Datenquellen aber auch manuell über Schlüsselfelder verknüpft werden. Damit stellt Power Pivot eine besonders nützliche Komponente für Self Service Business Intelligence zur Verfügung.

Power Pivot ist keine Weiterentwicklung der guten alten PivotTable. Technisch gesehen ist Power Pivot die Oberfläche eines SQL-Datenbankservers, und das Datenmodell, auf dem das Tool aufbaut, ist eine OLAP-Datenbank, die unsichtbar im Hintergrund aufgebaut wird. Wer sich entscheidet, in Zukunft mit Power Pivot zu arbeiten, sollte sich deshalb gleich mit den Prinzipien des Datenbankdesigns anfreunden, denn darauf basiert das Tool. Felder, KPIs, Hierarchien, 1:n-Beziehungen, Schlüsselfelder, linker und rechter Join, Datenbank-Speech – mit diesen Dingen sollten Sie sich vertraut machen. Dazu kommt die neue Formelsprache DAX mit englischsprachigen (!) Funktionen, die oft identisch mit vertrauten Excel-Funktionen (MAX, MIN, SUM) und doch so verschieden (RELATED, SUMX, PREVIOUSPERIOD) sind. Formeln heißen Measures, PivotTables sind die einzige Schnittstelle zu Excel. Das vertraute Tabellenblatt spielt mit Power Pivot keine Rolle mehr, es gibt nur noch das Datenmodell und die PivotTables.



**Bild 1.2** Power Pivot mit Daten- und Diagrammansicht

### 1.2.3 Power BI

Power BI ist im Verbund der BI-Tools für die Visualisierung der Daten zuständig. Aber damit nicht genug – das Programm bietet sogar die Funktionalität der beiden anderen Tools Power Query und Power Pivot. In der Praxis lässt sich mit Power BI also ein komplettes BI-System ganz ohne Excel-Unterstützung aufbauen. Power BI kann Daten aus unterschiedlichsten Quellen importieren, bietet die Möglichkeit, Daten ähnlich wie Power Query zu modellieren und zu transferieren, und genauso wie in Power Pivot können Tabellen auch über Schlüssel-felder relational verknüpft werden. Die Skriptsprache M steht ebenso wie in Power Query zur Verfügung, und das Datenmodell lässt sich über DAX-Funktionen mit zusätzlichen Kalkulationen erweitern.

Für die Visualisierung stehen zahlreiche Diagrammelemente, vom Balkendiagramm bis zur Geo-Map, und unterstützende Werkzeuge (Datenschnitte, Bild- und Zeichenwerkzeuge u. a.) zur Verfügung. Das Angebot an Visualisierungswerkzeugen wächst ständig, Entwickler können diese sogar selbst entwerfen und individuellen Anforderungen anpassen. Berichte können online und mobil abgefragt, bearbeitet, zu Dashboards zusammengefasst und mit Kollegen geteilt werden.

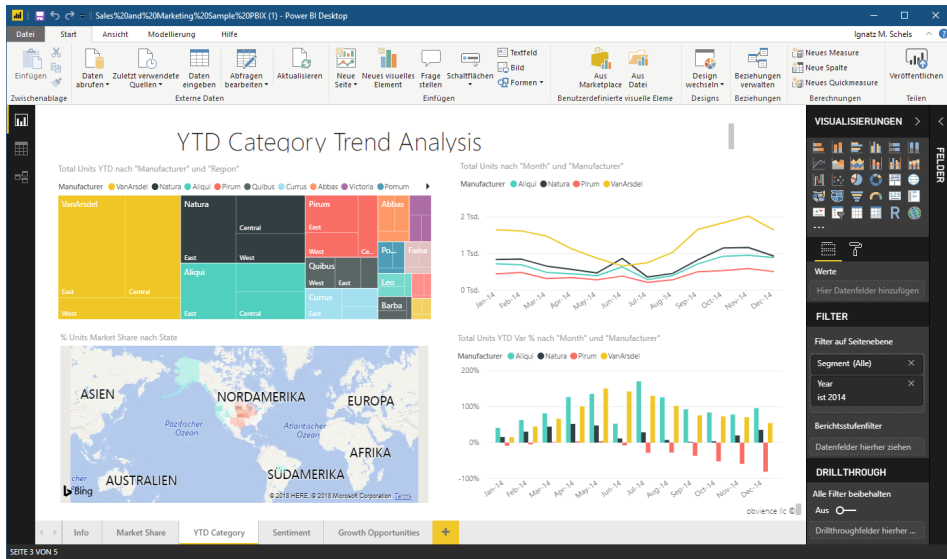


Bild 1.3 Power BI Desktop

Power BI integriert Excel in sein Cloud-Konzept. Mit der Möglichkeit, Excel-Daten in der Cloud zu veröffentlichen, an Berichte und Dashboards anzuhängen und dynamisch mit diesen auszutauschen, schafft Power BI alle Voraussetzungen, um Excel weiterhin als Client zu verwenden. Voraussetzung ist natürlich die Bereitschaft, die Cloud-Dienste von Microsoft zu nutzen (mit dem Report Server lässt sich auch eine Cloud-freie Lösung schaffen). Das Power BI-Konzept bietet sowohl für Einzelkämpfer als auch für Unternehmen jeder Größenordnung bezahlbare Pakete an.

Besonders vorteilhaft ist die Import-Funktion für Excel-Arbeitsmappen, die Power-Query-Abfragen und Power-Pivot-Datenmodelle enthalten. Excel-Anwender können so ihre BI-Anwendung in vertrauter Umgebung gestalten und anschließend an Power BI Desktop oder die Power BI-App in der Cloud übermitteln.

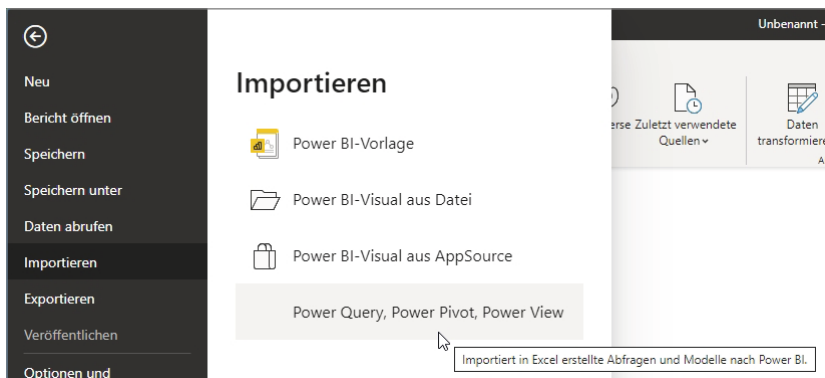


Bild 1.4 Import von BI-Daten aus Excel-Arbeitsmappen

### 1.2.4 Power View

Power View ist ein Add-in für Excel, das mit der Version 2013 eingeführt, in den Nachfolgeversionen aber nicht weitergepflegt wurde. Mit Power View wurden Daten in interaktiven Dashboards visualisiert. Power BI ist der designierte Nachfolger, und das hat folgende Gründe:

- Power View wurde im Original mit Silverlight erstellt, einem Browser-Plug-in für Web-Applikationen. Silverlight war als Konkurrenzprodukt zu Adobe Flash geplant. Microsoft hat nach zahlreichen Sicherheitslücken und mit der Ausrichtung auf HTML5 das Produkt 2015 eingestellt. Aus diesem Grund wird ActiveX und damit auch Silverlight in Edge, dem Windows 10-Browser, nicht mehr unterstützt. Google hat Silverlight ebenfalls aus seinem Browser Chrome verbannt.
- Power View wurde zwar noch auf HTML5 umgeschrieben, dabei gingen aber bereits einige Visualisierungen verloren. Zeitgleich startete die Entwicklung von Power BI, und auch wenn ein offizielles Statement dazu fehlt, wird Microsoft nur noch Power BI weiterentwickeln und Power View allmählich auslaufen lassen.
- In älteren Excel-Versionen (bis 2016) lässt sich Power View zwar noch in der Liste der COM-Add-ins aktivieren, das Tool funktioniert aber nicht.

### 1.2.5 Power BI oder Excel?

Warum sollte sich ein Anwender mit Power Query und Power Pivot in Excel anfreunden, wenn Power BI Desktop diese Werkzeuge auch anbietet? Wird Power BI Desktop kurz- oder langfristig Excel ersetzen?

Excel-Anwender werden sich langsam, aber stetig mit den beiden Excel-Tools Power Query und Power Pivot anfreunden, ihre Daten aber auch weiter mit den Standardwerkzeugen (Diagramme, PivotCharts, Sparklines etc.) visualisieren. Die Visualisierungselemente von Power BI Desktop sind optisch wesentlich anspruchsvoller und leistungsfähiger als die Excel-Werkzeuge, und mit der Zeit wird Power BI mit Excel zusammenwachsen und die Rolle der Visualisierung übernehmen. Aber – Power BI wird Excel niemals ersetzen können.

#### Vorteile von Power BI Desktop gegenüber Excel

- In Power BI Desktop sind die beiden ETL-Werkzeuge Power Query und Power Pivot enthalten, die Handhabung ist weitgehend mit Excel identisch. BI-Anwendungen in Excel-Arbeitsmappen können importiert werden.
- Power BI Desktop hat optisch anspruchsvollere und technisch ausgereifere Visualisierungswerkzeuge als Excel (Wasserfallcharts, Maps u. a.).
- Power BI Desktop ist die Basis für den Power BI Service. Mit diesem Werkzeug lassen sich schöne Visualisierungen erstellen und in Dateiform speichern. Um Reports im eigenen Unternehmen oder in der Cloud publizieren zu können, muss bei Microsoft ein kostenpflichtiges Konto beantragt werden. Die Möglichkeiten des Power BI Service reichen von automatischen Dashboards über Q&A (Frage und Antwort) bis zu KI (künstlicher Intelligenz).



## Nachteile von Power BI gegenüber Excel

- Wichtige Werkzeuge von Excel stehen mit Power BI Desktop nicht zur Verfügung. Power BI Desktop hat keine PivotTables. Reports können nur aus Tabellen erstellt werden, und Aggregationen müssen aufwendig mit DAX konstruiert werden.
- Power BI Desktop bietet nicht den Komfort eines Tabellenkalkulationsprogramms. Es gibt nur Datenquellen und Visualisierungen. Die Oberfläche lässt sich zwar flexibel ähnlich wie eine PowerPoint-Folie mit Tabellen, Diagrammen und grafischen Objekten aus schmücken, aber Excel-Tabellenblätter sind variantenreicher mit Rahmen, Farben, bedingten Formaten, Formularelementen, Bereichsnamen, Gliederungen, grafischen Objekten, SmartArts u. a. Für Layout und Druck gibt es zahlreiche Möglichkeiten, und in jeder Zelle lässt sich eine Berechnung unterbringen. Funktionen müssen sich auch nicht an Tabellenspalten und -zeilen orientieren.
- Dashboarding und Cloud-basierende Berichte sind die Zukunft, aber dieser Komfort will bezahlt werden. Power BI Services in der Cloud kosten Geld, und zwar nicht wenig, wenn ein Unternehmen sich entschließt, die BI-Services für seine Mitarbeiter oder Kunden einzurichten.
- Nicht zu vergessen: VBA. Die Makrosprache bietet nach wie vor unerreichten Komfort für den Dialog mit dem Anwender und die Automatisierung von Prozessen. Power BI hat (noch) nichts Gleichwertiges zu bieten.

## 1.2.6 Power Pivot oder Datenbank?

Macht es Sinn, Power Pivot an Stelle einer Datenbank einzusetzen? Wo sind die Vorteile dieses Tools, das sowohl in Excel als auch in Power BI Desktop zur Verfügung steht?

An den Datenmengen kann es nicht liegen. Power Pivot kann zwar wesentlich mehr als die 1 048 576 Zeilen von Excel einlesen, aber mehrere Millionen Datensätze sind auch für Access keine Herausforderung, und der große Bruder SQL Server hat damit noch weniger Probleme.

Mit dem DBMS Access lassen sich Daten wesentlich komfortabler aufbereiten als mit Power Pivot. Die Möglichkeit, Abfragen (Queries) zu erstellen und abzuspeichern, die Formulare als Dialogschnittstelle zum Benutzer, die Berichte und nicht zuletzt die gigantische VBA-Makrosprache bieten allen Komfort für die Entwicklung moderner Datenbanken.

Mit Access werden Daten erfasst, editiert und gelöscht (Read and Write). Power Pivot liefert nur eine Sicht auf gespeicherte oder per Verknüpfung zur Verfügung gestellte Daten (Read only). Um Daten abzuändern, muss der Benutzer Rechte und Zugriff auf externe Systeme (ERP, Datenbanken etc.) haben oder die Dateien (XLSX, CSV) öffnen und darin Änderungen vornehmen. Erst dann kann er seine Power Pivot-Auswertungen aktualisieren.

Access-Formulare können mit dem Einsatz von Formularwerkzeugen benutzerfreundlich gestaltet werden. Selbst unerfahrene Einsteiger werden mit Datenbankformularen beste Resultate erzielen. Die Power Pivot-Oberfläche lässt sich nicht gestalten, eine Automatisierung von Prozessen oder eine Oberfläche mit Formularelementen gibt es nicht.

Power Pivot ist also nur eine abgespeckte Version einer Datenbank, aber Beziehungen, Datentypen und berechnete Spalten sind wesentlich einfacher zu erlernen als Datenbankstrukturen, SQL-Abfragen und Formularelemente. Wer mit dem spärlichen Komfort leben kann und schnelle Resultate mit verknüpften Datenquellen braucht, wird Power Pivot auch den Vorzug vor Access & Co. geben. Und nicht zu vergessen seien die vielen Anwender, die von der IT ausgebremst werden, weil sie kein Access haben oder haben dürfen (Datenbanken machen Ärger und Arbeit!). Für sie ist Power Pivot nicht die beste, aber die einzige Lösung.

# 2

## Grundlagen, Verfahren und Konzepte

Business Intelligence mit Excel ist mehr als Excel. Neue Verfahren, neue Werkzeuge und neue Konzepte halten Einzug in die altherwürdige Tabellenkalkulation. Wer sich noch auf dem Wissensstand von vor 10 Jahren ausruht und seinen Horizont auf Zellen, Tabellenblättern und Arbeitsmappen mit Formatierungen und einigen Funktionen begrenzt, sollte umdenken. Das Wissen um grundlegende Datenbankkonzepte und -formate wird für Business Intelligence ebenso gebraucht wie der sichere Umgang mit den BI-Tools.

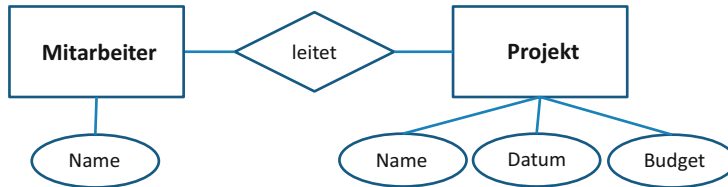
### ■ 2.1 Relationale und multidimensionale Datenbanken

Die relationale Datenbank ist eine Erfindung aus den 70er-Jahren. Dr. Edgar F. Codd, ein Programmierer bei IBM, erfand das Modell der relationalen Datenbank. Er verwendete die Abfragesprache SEQUEL, aus der SQL hervorging, die Sprache, die heute noch Kern und Basis aller Datenbanksysteme ist. Edgar F. Codd entwickelte auch die Normalformen und die 12 Evaluierungsregeln als Anforderung für ein OLAP-System (Online Analytic Processing). Oracle kam als erstes Unternehmen im Jahr 1979 mit einem relationalen Datenbankmanagementsystem auf den Markt.

Mitte der 90er-Jahre nutzte jedes größere Unternehmen relationale Datenbanken. Aber mit der Zeit tauchte ein Problem auf: Die Datenbanken mit relationalen Verknüpfungen waren zwar ideal geeignet, um Daten auch in größeren Mengen zu speichern. Für die Erstellung von Berichten aus Transaktionsdaten erwiesen sich die Datenbanken jedoch als zu schwerfällig. Musste zum Beispiel ein Bericht über monatliche Umsätze erstellt werden, suchte das System in Millionen von Transaktionen, um eine einzelne Summe zu ermitteln. Nicht selten ging dabei die gesamte Serverwelt „in die Knie“, und an schnelle, Just-in-time-Informationen war gar nicht zu denken.

## ■ 2.2 Das Entity-Relationship-Modell (ERM)

Das Entity-Relationship-Modell (ERM) ist die Grundlage für einen Datenbankentwurf. Die Grundelemente sind Entitäten, Beziehungen und Attribute. Im ERM-Modell werden die grundlegenden Beziehungen einer relationalen Datenbank aufgestellt.



**Bild 2.1** Das Entity-Relationship-Modell

## ■ 2.3 Normalisierung und Faktentabellen

Normalisierung ist die Aufteilung der Attribute (Tabellenspalten) in mehrere Relationen unter Berücksichtigung der Normalisierungsregeln und Normalformen mit dem Ziel, eine Form zu erhalten, die keine vermeidbaren Redundanzen aufweist.

### Erste Normalform

In der ersten Normalform sind alle Werte atomar, d. h. nicht mehr teilbar.

Rechnungsnr.	Name	Artikel	Anzahl	Preis
1234	Max Mustermann	Bleistift	10	1,20 €

↓

Rechnungsnr.	Name	Vorname	Artikel	Anzahl	Preis
1234	Mustermann	Max	Bleistift	10	1,20 €

## Zweite Normalform

In der 2. Normalform sind die Daten in Tabellen aufgeteilt, die Beziehungen zwischen den Daten werden über Schlüssel definiert.

Tabelle "Kunden"			
Kundennr.	Name	Vorname	
0815	Mustermann	Max	

Tabelle "Artikel"		
Artikelnr.	Artikel	Preis
A-01	Bleistift	1,20 €

Tabelle "Rechnungen"			
Rechnungsnr.	Kundennr.	Artikelnr.	Anzahl
1234	0815	A-01	10

## Beziehungsarten

1:1		Jeder Datensatz in Tabelle A ist genau mit einem Datensatz in Tabelle B verbunden (nicht zu empfehlen).
1:n		Jeder Datensatz in Tabelle A kann mit beliebig vielen Datensätzen in Tabelle B verbunden sein, einem Datensatz in Tabelle B ist aber nie mehr als ein Datensatz aus Tabelle A zugewiesen (häufigste Beziehungsart).
n:m		Jeder Datensatz in Tabelle A kann mit mehreren Datensätzen in Tabelle B und umgekehrt verbunden sein. Dazu wird eine dritte Tabelle benötigt, in der die Schlüssel aus Tabelle A und Tabelle B enthalten sind.

## ■ 2.4 Sternschema

Zu den wichtigsten Konzepten im Datenbankdesign gehört das Sternschema. Wer längere Erfahrung mit Excel hat, wird es gewohnt sein, die Daten in Listen- oder Tabellenform anzuordnen, um gezielt filtern, sortieren oder per PivotTable analysieren zu können. Relationen, d. h. Beziehungen zwischen zwei Tabellen lassen sich über Verweisfunktionen (SVERWEIS(), HVERWEIS() XVERWEIS()) realisieren, was aber nichts mit Datenbankmodellierung zu tun hat.

Für ein relationales Datenbankmodell, das für BI-Analysen geeignet ist, werden im ersten Schritt alle Tabellen gesammelt. Für jede Tabelle wird die erforderliche Anzahl Felder definiert. Hier gilt das Prinzip der Einfachheit: Daten, die nicht ins Reporting einfließen, haben im Datenmodell nichts verloren.

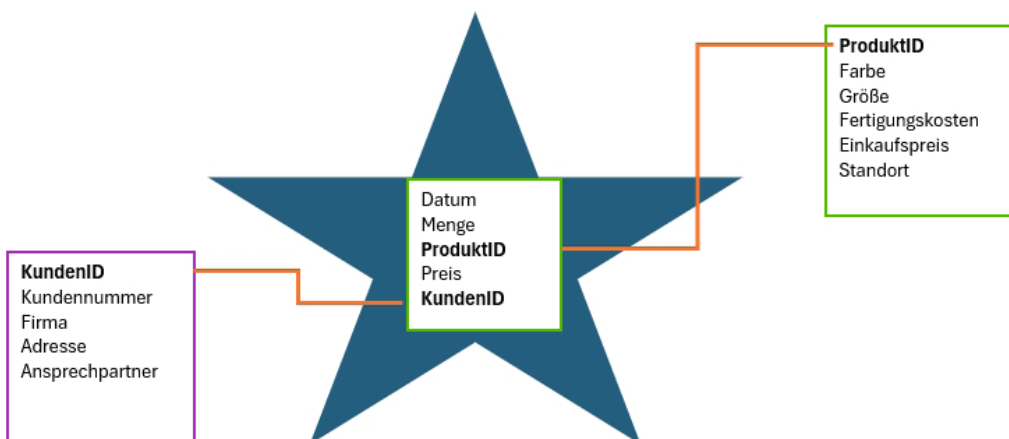
Bei der Sammlung der Daten sollte von Anfang an das Sternschema berücksichtigt werden. Dazu werden alle Daten in diese Tabellen unterteilt:

- Fakten
- Dimensionen

Fakten enthalten die „bewegten“ Daten: Verkäufe mit Mengenangaben, Umsätze mit Preisen, Bestellungen, Aufträge. Fast immer enthalten Faktentabellen Datumfelder, die in den Auswertungen auf Monate, Jahre etc. aggregiert werden.

Dimensionen enthalten die Details zu den Daten in den Faktentabellen. Die „Stammdaten“ sind Produktpreislisten mit Beschreibungen, Kategorien, Herstellerangaben, Mitarbeiterdaten mit Eintrittsdatum, Kundenlisten mit Adressen u. a. Wichtigstes Merkmal ist die Eindeutigkeit: Jeder Datensatz wird durch die Zuweisung einer ID einmalig, es gibt das Produkt ebenso wie den Kunden oder den Mitarbeiter nur einmal.

Im Sternschema wird die Faktentabelle zentriert, und die Dimensionen ordnen sich mit den ID-Verknüpfungen rund um die Fakten an.



**Bild 2.2** Fakten im Zentrum, Dimensionen rundum – das Sternschema

## ■ 2.5 OLAP – die multidimensionale Datenbank

Die multidimensionale Datenbank ist eine Weiterentwicklung des relationalen Modells. Edgar F. Codd entwickelte sie mit dem Ziel, die Server von Berechnungen zu entlasten, die für aktuelle Berichtserstellungen benötigt wurden. Dazu werden die Aggregationen (Summen) zu Zeiten, in denen die Server nicht beschäftigt sind (nachts) in speziellen Datenbanken gespeichert, sogenannte OLAP-Cubes. Cubes müssen weniger zeitaufwendige Berechnungen durchführen, da ihre Ergebnisse vorberechnet sind.

Ein OLAP-Cube ist ein „Snapshot“ auf die Daten zu einem bestimmten Zeitpunkt, zum Beispiel am Ende des Tages, der Woche, des Monats oder des Jahres (Year-to-date).

### 2.5.1 Von OLAP zu BI

Der Begriff Business Intelligence ist eng mit der OLAP-Technik verbunden. BI-Analysen sind in der Regel nichts anderes als Auswertungen von OLAP-Cubes, aber diese Analysen waren bisher wie die Herstellung und Wartung dieser Cubes nur von IT-Spezialisten und erfahrenen Programmierern durchführbar. Entsprechend zeitraubend und kostspielig gestalteten sich BI-Analysen.

Mit der Excel-PivotTable machte Microsoft den ersten Schritt in Richtung Self-Service-BI: Der einfache Excel-Anwender war in der Lage, Daten in unterschiedlichen Perspektiven darzustellen, zu filtern und mit arithmetischen oder statistischen Methoden (Summe, Anzahl, Mittelwert, Standardabweichung etc.) auszuwerten und diese in ein Datenmodell (OLAP-Cube) zu befördern.

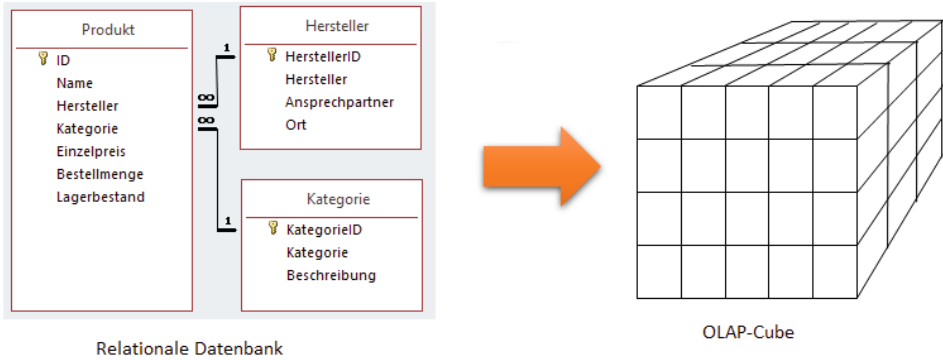
Technisch gesehen entsteht nämlich bei der Anlage einer PivotTable nichts anderes als ein OLAP-Cube.

### 2.5.2 Wie ein OLAP-Cube funktioniert

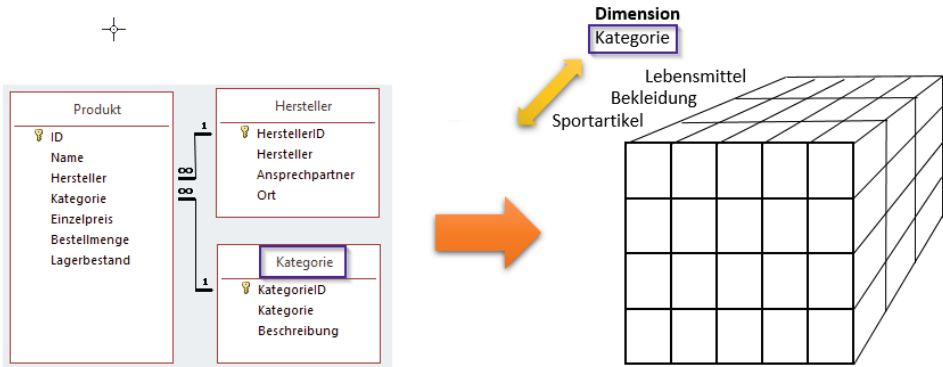
Ein Cube besteht aus Measures, Dimensionen und Dimensionsattributen.

- *Measures* sind numerische Datenwerte aus Formeln, die über Dimensionen hinweg Werte zählen, Summen berechnen, Durchschnitte oder Maximal-/Minimalwerte ermitteln.
- *Dimensionen* sind ähnlich wie Tabellen im relationalen Modell die Spalten- und Zeilenachsen des Cubes. Typische Dimensionen sind Produkte, Artikel, Kunden, Mitarbeiter, Vertriebsgebiete etc.
- *Dimensionsattribute* sind benannte Elemente innerhalb einer Dimension, ähnlich den Spalten einer Tabelle. Typische Dimensionsattribute für die Dimension *Vertriebsgebiete* wären zum Beispiel Europa, USA, EMEA (Europe and Middle East).

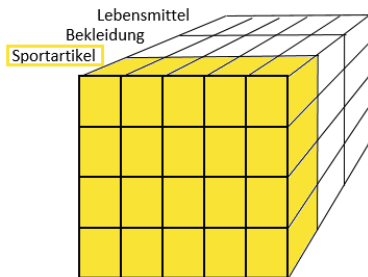
Im Prinzip besteht ein OLAP-Cube aus vielen einzelnen Boxen. Jede Box enthält eine Aggregation (Summe) aus den Daten einer Datenbank.



Fügen wir eine *Dimension* für die Daten aus dem Datenbankelement *Kategorie* hinzu.

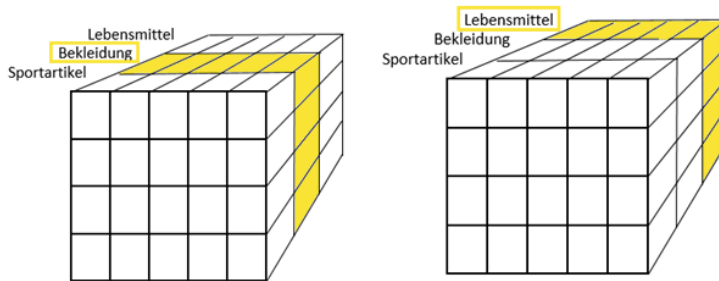


Die Vorderseite des Würfels enthält die Summen der ersten Kategorie.

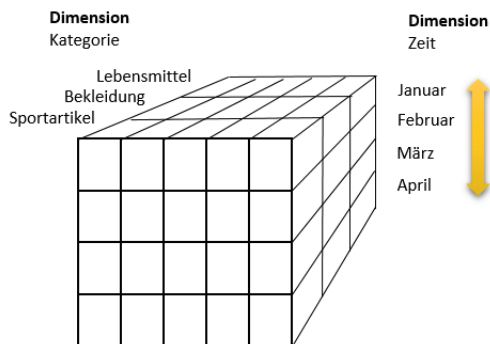




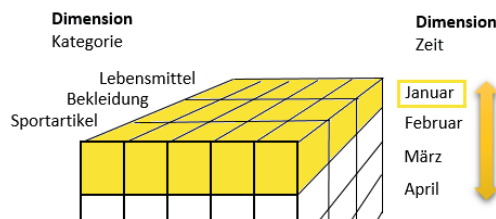
Die mittleren Boxen enthalten die Summen der zweiten Kategorie und die letzte Würfelreihe enthält die Summen der letzten Kategorie.



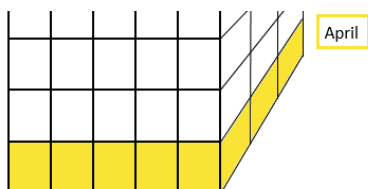
Fügen wir eine weitere Dimension *Zeit* mit den Monaten als Elemente hinzu.



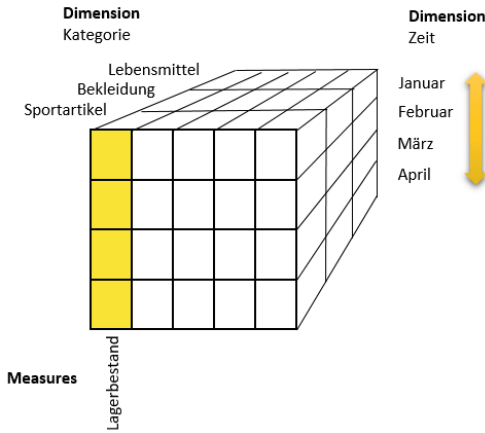
Jetzt enthält die erste horizontale Reihe die Summen des ersten Monats.



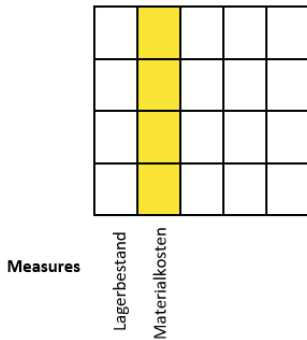
Und die letzte horizontale Reihe speichert die Summen des letzten Monats.



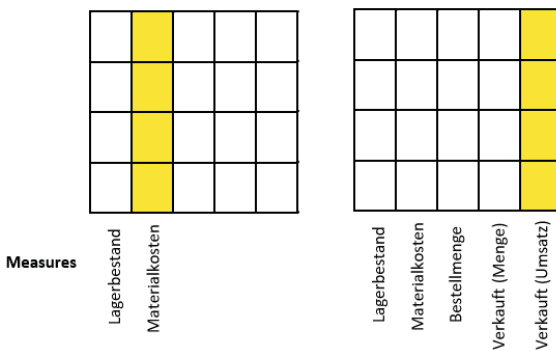
Das nächste Element aus der OLAP-Technik ist das *Measure*. Ein Measure bezeichnet die Aggregatfunktion oder einfacher gesagt, was im Würfel berechnet wird. Fügen wir auf der Unterseite des Würfels ein erstes Measure für den *Lagerbestand* hinzu.



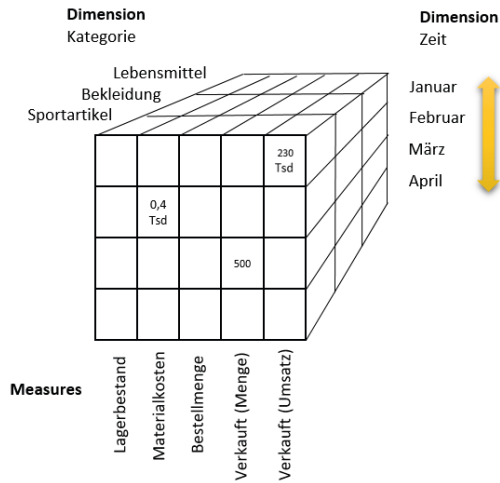
Die nächste Reihe bekommt das Measure *Materialkosten*.



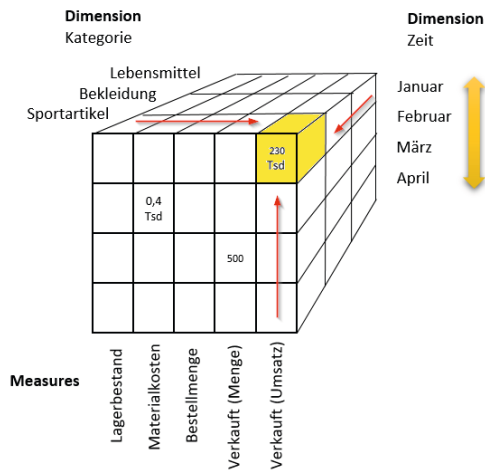
Und schließlich werden noch die Summen für die Bestellmenge und die Verkaufsdaten eingebaut.



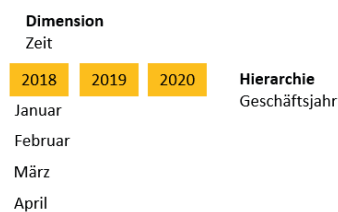
Füllen wir die einzelnen Boxen des Cubes anschließend mit Werten, lassen sich diese über den Schnittpunkt von Measure und Dimension eindeutig bezeichnen.



So lässt sich der *Umsatz* (Measure) für die Kategorie *Sportartikel* (Dimension 1) im Monat *Januar* (Dimension 2) eindeutig klassifizieren.



Noch ein Begriff aus der OLAP-Technologie: Eine Hierarchie wird eingeführt, um Dimensionen zu gruppieren und in Eltern/Kinder (parent/child) zu unterteilen. Damit bietet der Cube die Möglichkeit, Daten per Drilldown von einer oberen Ebene in eine tiefere Ebene zu analysieren.



## ■ 2.6 Datenimport und dynamische Verknüpfungen

Der Umgang mit Daten hat sich in Excel grundlegend gewandelt und wird sich mit Business Intelligence weiter in die eingeschlagene Richtung entwickeln. Daten werden nicht in Tabellenblättern und Arbeitsmappen gespeichert und auch nicht mehr mit Formelkonstruktionen und Funktionstricks kombiniert und ausgewertet. Die Verweisfunktionen SVERWEIS, WVERWEIS und XVERWEIS als klassische Werkzeuge für einfache Beziehungen zwischen Datenbeständen haben weitgehend ausgedient, an ihre Stelle treten dynamische Verknüpfungen auf externe Daten, Datenbanken und Webseiten. Client/Server heißt die Devise, Excel wird zum Auswertungstool für externe Daten.

Dabei ist die dynamische Verknüpfung auf externe Daten ein „alter Hut“ für erfahrene Excel-Anwender. Schon in Excel 5.0 (1995) gab es die DDE-Schnittstelle für dynamische Datenverbindungen, in Office 97 war MS Query der Garant für sichere SQL-Abfragen an Datenbanken. Mit Business Intelligence kommen neue Tools ins Spiel, Power Query verbindet den Datenimport mit der Möglichkeit, die Daten gleich zu transformieren und aufzubereiten und Power Pivot stellt Beziehungen wie Datenbankprogramme zur Verfügung.

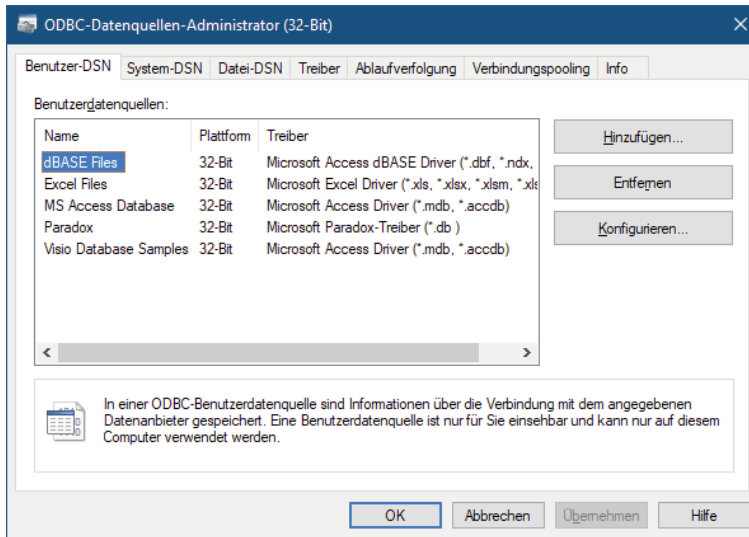
### 2.6.1 ODBC

ODBC (Abkürzung für „Open Database Connectivity“) ist eine 1992 von Microsoft entwickelte Software-Schnittstelle, die den Zugriff aus einem Anwendungsprogramm auf unterschiedliche Datenbanken gewährleistet. Jeder Datenbankhersteller (Provider), der sich an den ODBC-Standard hält, kann Treiber zur Verfügung stellen, damit Office-Programme wie Excel auf seine Datenbanken zugreifen können.

### 2.6.2 ODBC-Treiber verwalten

Für die Verwaltung dieser Treiber stellt Windows einen Dienst in der Systemsteuerung zur Verfügung:

- Wählen Sie in der Systemsteuerung *Verwaltung/Datenquellen (ODBC)* oder geben Sie bei Cortana (Windows 10) oder im Suchfeld von Windows 7 einfach „ODBC“ ein.
- Der ODBC-Datenquellen-Administrator verwaltet die installierten ODBC-Treiber. Holen Sie mit *Hinzufügen* weitere Treiber in die Liste.
- Benutzerdatenquellen sind spezifisch für das Microsoft Windows-Anmeldekonto, das zum Zeitpunkt des Erstellens der Datenquellen aktiv ist. Sie sind für keine anderen Anmeldekonto sichtbar. Für Anwendungen, die als Dienste auf einem Computer ausgeführt werden, sind sie nicht immer sichtbar.
- Systemdatenquellen sind für alle Anmeldekonto auf einem Client sichtbar. Für Anwendungen, die als Dienste auf einem Computer ausgeführt werden, sind sie immer sichtbar.
- Dateidatenquellen werden nicht in der Systemregistrierung gespeichert; sie werden in einer Datei auf dem Client gespeichert.



**Bild 2.3** ODBC-Datenquellen werden unter Windows verwaltet.

## ■ 2.7 ODBC und Power Query

ODBC ist eine alte Datenimport-Technik, die von Microsoft nicht mehr weiterentwickelt wird. Power Query bietet als Nachfolger von ODBC wesentlich mehr und deutlich bessere Verknüpfungen zu externen Datenquellen an. Da viele Anwender aber mit ODBC arbeiten, müssen die alten Werkzeuge auch in der neueren Version noch verfügbar sein. Excel hat ODBC in Version 2016 und den ersten Versionen von 2019 noch im Menüband in einer eigenen Gruppe neben der Power Query-Gruppe *Daten abrufen und transformieren* angeboten, diese dann aber mit Microsoft 365 geschlossen. ODBC-Treiber finden Sie noch unter *Daten/Daten abrufen und transformieren/Daten abrufen/Aus anderen Quellen/Aus ODBC*, diese Treiber funktionieren aber nicht mehr. Mit dem Legacy-Assistenten holen Sie ODBC zurück in die Menüstruktur:

- Wählen Sie *Datei/Optionen/Daten*.
- Schalten Sie unter *Legacy-Datenimport-Assistenten anzeigen* die alten ODBC-Treiber für Access, Webabfragen, Textdaten, SQL-Server, OData, XML und den alten Datenverbindungs-Assistenten ein.

Damit lassen sich die ODBC-Verbindungen über *Daten/Daten abrufen und Transformieren/Daten abrufen/Legacy-Assistenten* wieder starten.

## ■ 2.8 SQL

SQL (Structured Query Language) ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken und zur Erstellung und Bearbeitung von Abfragen auf darauf basierende Datenbestände. SQL-Abfragen sind das Kernstück der relationalen Datenbank. Mit der Abfrage (View) werden Daten eingefügt, editiert, gelöscht oder einfach abgefragt.

SQL lässt sich in Excel-Abfragen auf externe Datenquellen einsetzen. Der Abfrage-Assistent bietet die Möglichkeit, SQL-Abfragen zu editieren oder neue Abfragen zu generieren. In Abfragen, die mit Power Query erstellt wurden, lässt sich der SQL-Befehl nicht editieren, SQL kann aber für gezielte Abfragen an externe Datenquellen eingesetzt werden.

Hier eine Liste mit SQL-Anweisungen, die per ODBC an eine Excel-Mappe übermittelt werden können:

Abfrage	Erklärung
SELECT * FROM [Tabelle1\$]	Die Daten werden aus einem Tabellenblatt erhoben. \$ und [] sind hier Pflicht.
SELECT * FROM [Tabelle1\$A1:C10]	Die Daten werden aus einem Zellenbereich eines Tabellenblatts erhoben.
SELECT * FROM Name	Die Daten werden aus einem „definierten Namen“ im Gültigkeitsbereich „Arbeitsmappe“ erhoben. Der Bereich muss hier auf „Arbeitsmappe“ gestellt sein (Default-Wert), da ansonsten der Name nicht über ODBC angesprochen werden kann.
SELECT * FROM BereichA UNION SELECT * FROM BereichB	Die Daten werden per UNION SELECT aus zwei verschiedenen Bereichen eines Tabellenblatts erhoben.
SELECT * FROM Datei1.csv UNION SELECT * FROM Datei2.csv	Die Daten werden per UNION SELECT aus zwei verschiedenen CSV-Dateien erhoben. In der <i>.ini</i> -Datei wird der Ordner angegeben, in denen sich die Dateien befinden.
INSERT INTO [Tabelle1\$] VALUES (Wert1,Wert2,Wert3,...)	Es werden Daten in ein Tabellenblatt eingefügt. INSERT INTO ist nur über ein Worksheet möglich (nicht über einen Bereich).

### 2.8.1 SQL-Abfragen mit Excel

Die ODBC-Technik ist zwar mit der Einführung der BI-Tools etwas veraltet, SQL bietet aber die Möglichkeit, gezielte Anforderungen an externe Daten zu richten. Hier ein Beispiel aus der Praxis, wie mit ODBC und SQL-Abfragen Daten gezielt aus einer externen Quelle geholt werden.

Für die Abfrage verwenden Sie Microsoft Query. Das ist ein alter Abfrage-Assistent, der schon für Excel Version 3.0 entwickelt wurde, aber immer noch gute Dienste leistet.

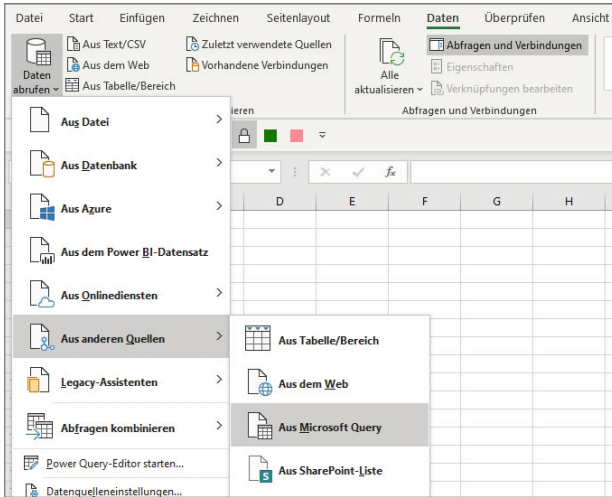
Die Mappe *Umsatz\_Golfshop.xlsx* enthält Produktumsätze in 12 Monatsstabellen von Januar bis Dezember.

	A	B	C	D	E	F	G	H	I	J	K
1	Date	ArticleNr	Description	Category	Manufacturer	Sales					
2	24.01.21	G-B-0101	Srixon AB	Bälle 3er Pack	Bridgestone	27108					
3	21.01.21	G-B-0102	Callaway Mx	Bälle 3er Pack	Callaway	38247					
4	14.01.21	G-B-0103	Taylor Made Gold	Bälle 3er Pack	Dunlop	48024					
5	25.01.21	G-B-0104	Dunlop DD	Bälle 3er Pack	Srixon	18015					
6	27.01.21	G-B-0105	Titleist CY	Bälle 3er Pack	TaylorMade	68045,96					
7	01.01.21	G-B-0106	Bridgestone DX	Bälle 3er Pack	Titleist	89664					
8	23.01.21	G-B-0107	Top Flite XL Distance	Bälle 3er Pack	Top Flite	9024,6					
9	25.01.21	G-B-0108	Top Flite Pro Spin	Bälle 3er Pack	Top Flite	6127,7					
10	28.01.21	G-CL-0101	Callaway FT-5	Driver	Callaway	254349					
11	30.01.21	G-CL-0102	Mizuno MX 560	Driver	Callaway	347304					
12	20.01.21	G-CL-0103	Titleist 905 R	Driver	Mizuno	39767					
13	28.01.21	G-CL-0104	Callaway Fti	Driver	Titleist	164498					
14	16.01.21	G-CL-0105	Callaway FT	Eisen	Callaway	661770					
15	24.01.21	G-S-0102	Mikado Dry	Golfschuhe	Mikado	90109,8					
16	18.01.21	G-CL-0122	Callaway Fusion FT	Holz	Callaway	256320					
17	15.01.21	G-CL-0123	Cleveland HiBore XLS	Holz	Cleveland	105194					
18	14.01.21	G-CL-0151	Zentaurus	Putter	Yamato	136484					
19	31.01.21	G-T-0101	Bad Boy OE-7	Trolley	Bad Boy	53535					
20	02.01.21	G-T-0102	Hartl Ecar	Trolley	Hartl	89679					
21	14.01.21	G-CL-0154	Forged CNC	Wedge	Komperdell	77745					
22	16.01.21	G-CL-0105	Callaway FT	Eisen	Callaway	661770					
23	24.01.21	G-S-0102	Mikado Dry	Golfschuhe	Mikado	90109,8					
24	18.01.21	G-CL-0122	Callaway Fusion FT	Holz	Callaway	77745					
25	15.01.21	G-CL-0123	Cleveland HiBore XLS	Holz	Cleveland	661770					
26	14.01.21	G-CL-0151	Zentaurus	Putter	Yamato	90109,8					
27	31.01.21	G-T-0101	Bad Boy OE-7	Trolley	Bad Boy	256320					
28	02.01.21	G-T-0102	Hartl Ecar	Trolley	Hartl	89679					
29	14.01.21	G-CL-0154	Forged CNC	Wedge	Komperdell	77745					
30	14.01.21	G-CL-0151	Zentaurus	Putter	Yamato	180219,6					
31	31.01.21	G-T-0101	Bad Boy OE-7	Trolley	Bad Boy	512640					
32	02.01.21	G-T-0102	Hartl Ecar	Trolley	Hartl	179358					
33	14.01.21	G-CL-0154	Forged CNC	Wedge	Komperdell	155490					
34											
35											
36											

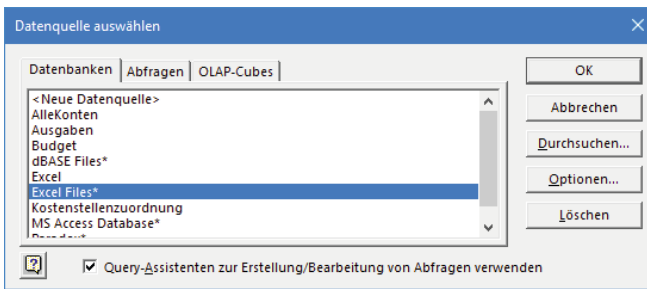
**Bild 2.4** Umsätze, aufgeteilt auf zwölf Monatsblätter

Legen Sie eine neue Arbeitsmappe an und starten Sie eine neue ODBC-Abfrage über MS Query. In Excel bis 2016 finden Sie dafür ein Symbol unter *Daten/Externe Daten abrufen*.

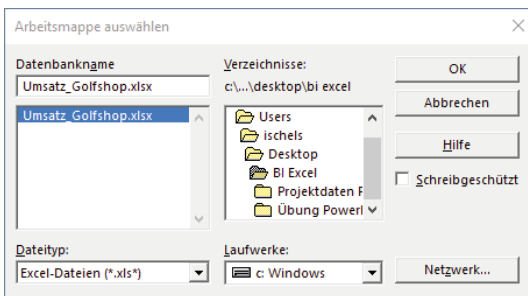
Ab Excel 2016 Pro finden Sie den Befehl in Power Query unter *Daten/Daten abrufen und transformieren/Daten abrufen/Aus anderen Quellen/Aus Microsoft Query*.



Wählen Sie den ODBC-Treiber für Excel.

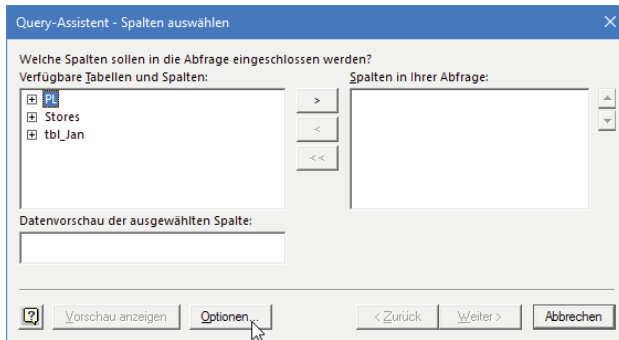


Holen Sie die Datei in die Abfrage.

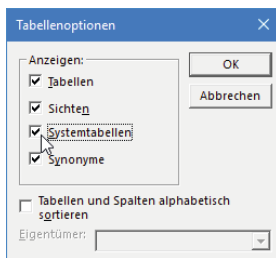


Die Quelldatei enthält weder Tabellen noch benannte Bereiche, sondern nur Listen. Wenn die Fehlermeldung „Diese Datenquelle enthält keine sichtbare(n) Tabellen“ angezeigt wird, bestätigen Sie mit OK und schalten Sie um auf *Optionen*.

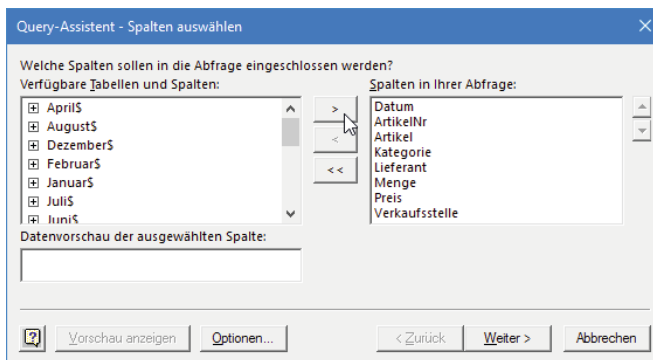




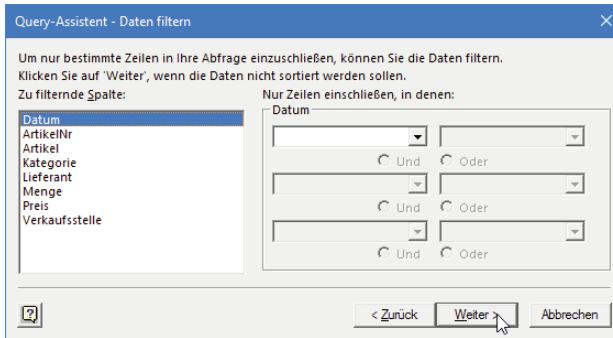
Schalten Sie die Systemtabellen ein.



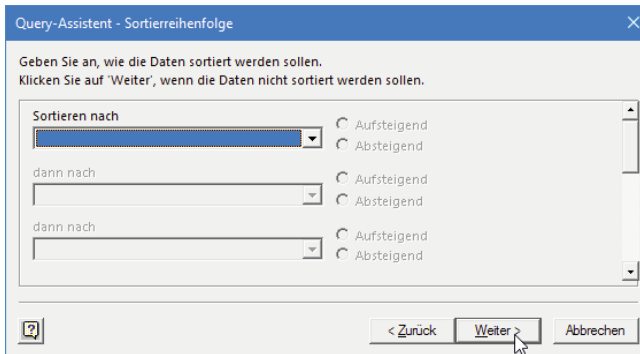
Jetzt werden auch die Tabellenblätter angezeigt, zu erkennen an einem \$-Zeichen am Blattnamen. Holen Sie alle Felder aus dem Tabellenblatt „Januar“ in die Abfrage.



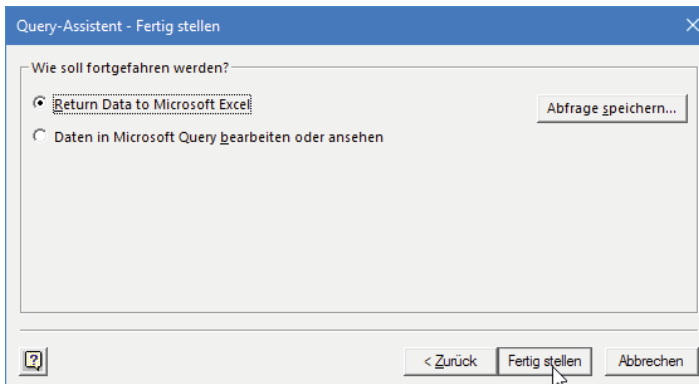
Filtern Sie die Abfrage bei Bedarf nach Datum oder Verkaufsstelle.



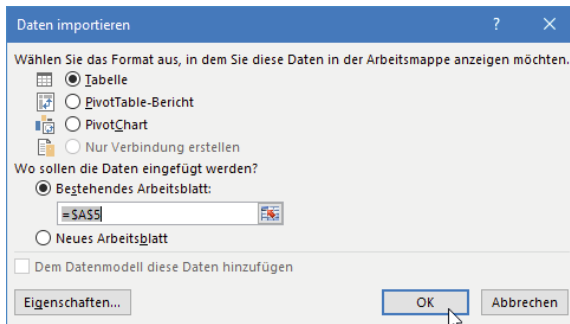
Sortieren Sie die Abfrage nach beliebigen Feldern.



Schicken Sie das Abfrageergebnis zurück zu Excel.



Geben Sie die Zelle A5 als Ziel für die neue Tabelle an.



Aktivieren Sie unter *Daten/Abfragen und Verbindungen* die Anzeige der Abfragen rechts am Bildschirmrand (Klick auf *Abfragen und Verbindungen*). Wählen Sie *Eigenschaften* im Kontextmenü der rechten Maustaste.

	A	B	C	D	E	F	G	H	I	J
1	Date	ArticleNr	Description	Category	Manufacturer	Sales				
2	24.01.2021 00:00	G-B-0101	Srixon AB	Bälle 3er Pack	Bridgestone	27108				
3	21.01.2021 00:00	G-B-0102	Callaway Mx	Bälle 3er Pack	Callaway	38247				
4	14.01.2021 00:00	G-B-0103	Taylor Made Gold	Bälle 3er Pack	Dunlop	48024				
5	25.01.2021 00:00	G-B-0104	Dunlop DD	Bälle 3er Pack	Srixon	18015				
6	27.01.2021 00:00	G-B-0105	Titleist CY	Bälle 3er Pack	TaylorMade	68045,96				
7	01.01.2021 00:00	G-B-0106	Bridgestone DX	Bälle 3er Pack	Titleist	89664				
8	23.01.2021 00:00	G-B-0107	Top Flite XL Distance	Bälle 3er Pack	Top Flite	9024,6				
9	25.01.2021 00:00	G-B-0108	Top Flite Pro Spin	Bälle 3er Pack	Top Flite	6127,7				
10	28.01.2021 00:00	G-CL-0101	Callaway FT-5	Driver	Callaway	254349				
11	30.01.2021 00:00	G-CL-0102	Mizuno MX 560	Driver	Callaway	347304				
12	20.01.2021 00:00	G-CL-0103	Titleist 905 R							
13	28.01.2021 00:00	G-CL-0104	Callaway Ft							
14	16.01.2021 00:00	G-CL-0105	Callaway FT							
15	24.01.2021 00:00	G-S-0102	Mikado Dry							

**Abfragen und Verbindungen**

Abfragen | **Verbindungen**

1 Verbindung

Abfrage von ...

- Aktualisieren
- Löschen
- Eigenschaften...

**Verbindungseigenschaften**

Verbindungsname: Abfrage von Excel Files

Beschreibung:

Verwendung | Definition | Verwendet in

**Aktualisierungssteuerung**

Zuletzt aktualisiert:

Aktualisierung im Hintergrund zulassen

Aktualisierung alle 60 Minuten

Aktualisieren beim Öffnen der Datei

Daten vor dem Speichern des Arbeitsblatts aus dem externen Datenbereich entfernen

Bei "Alle aktualisieren" diese Verbindung aktualisieren

**OLAP-Serverformatierung**

Bei Verwendung dieser Verbindung folgende Formate vom Server abrufen:

Zahlenformat  Füllfarbe

Schriftschnitt  Textfarbe

**OLAP-Drillthrough**

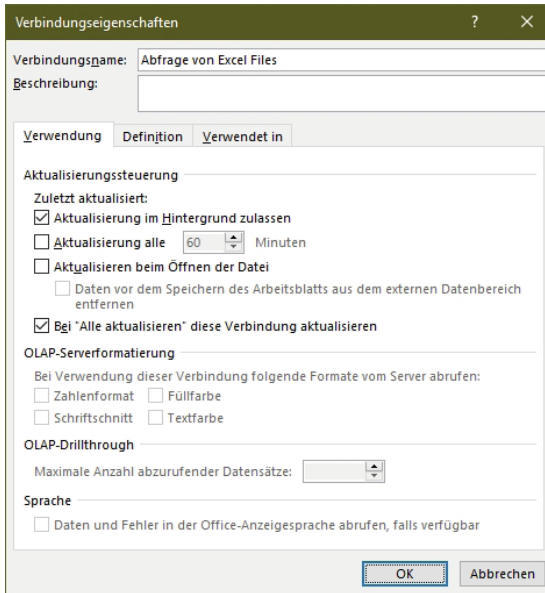
Maximale Anzahl abzurufender Datensätze:

**Sprache**

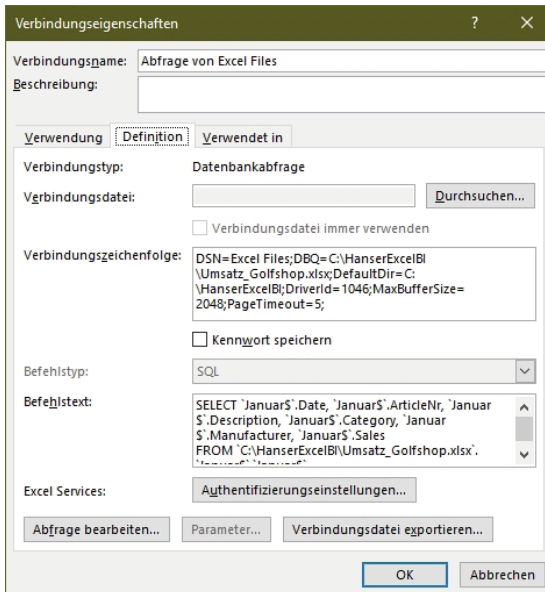
Daten und Fehler in der Office-Anzeigesprache abrufen, falls verfügbar

OK Abbrechen

Im ersten Register stellen Sie die Aktualisierungsparameter ein und bestimmen den Verbindungsnamen.



In der Definition sehen Sie die Verbindungsdaten und den SQL-Befehl.



Ändern Sie diesen, verwenden Sie die Kurzform für das Tabellenblatt:

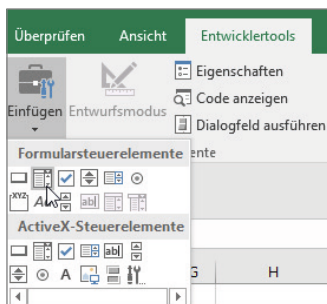
Befehlstyp:	SQL
Befehlstext:	SELECT * FROM [Januar\$]

[Tabellenblatt\$]  
SELECT \* FROM [Januar\$]

Legen Sie in einem neuen Tabellenblatt eine Monatsreihe an und geben Sie ihr den Bereichsnamen „Monate“.

Monate	
	A
1	Januar
2	Februar
3	März
4	April
5	Mai
6	Juni
7	Juli
8	August
9	September
10	Oktober
11	November
12	Dezember
13	

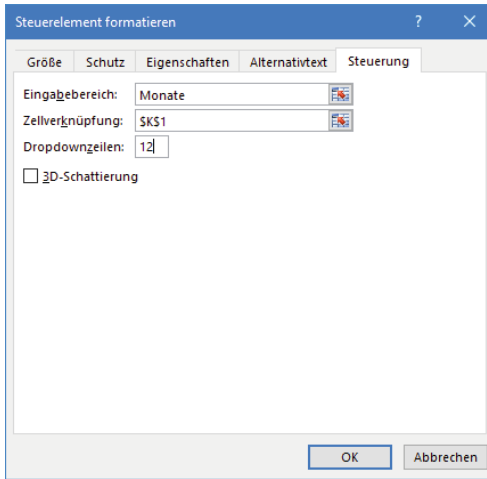
Fügen Sie im Tabellenblatt mit der Abfrage vier Zeilen über der Überschrift ein und zeichnen Sie ein Formularsteuerelement (Kombinationsfeld) über die Entwicklertools.



Formatieren Sie das Steuerelement über das Kontextmenü.

B	C	D	E	F	G
	Bitte Monat wählen:				
ArtikelNr	Artikel	Kategorie			/erl
G-B-0106	Bridgestone DX	Bälle 3er			P-10
G-T-0102	Hartl Ecar	Trolley			P-20
G-T-0102	Hartl Ecar	Trolley			P-30
G-T-0102	Hartl Ecar	Trolley			P-40
G-B-0103	Taylor Made Gold	Bälle 3er			P-10
G-CL-0151	Zentaurus	Putter	Yamato	120	P-20

Weisen Sie dem Steuerelement die Monate und als Ausgabeverknüpfung die Zelle K1 zu. *Dropdownzeilen* bestimmt die Größe der Liste nach dem Öffnen.

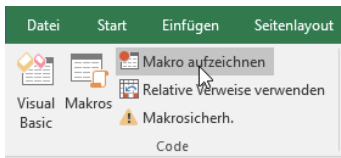


Schreiben Sie über eine Textformel den SQL-Befehl, der den gewählten Monat beinhaltet:

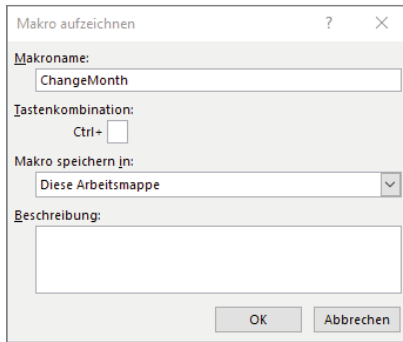
="SELECT * FROM ["&INDEX(Monate;\$K\$1;1)&"\$]"				
	J	K	L	M
		1		
		SELECT * FROM [Januar\$]		

```
"SELECT * FROM ["
&INDEX(Monate;$K$1;1)&"$]"
```

Zeichnen Sie ein Makro auf, das die ODBC-Verbindung der Abfrage auf die Golfshop-Daten ändert.



Geben Sie dem Makro den Namen *ChangeMonth* und starten Sie mit *OK*.



Makro aufzeichnen

Makroname: ChangeMonth

Tastenkombination: Ctrl+

Makro speichern in: Diese Arbeitsmappe

Beschreibung:

OK Abbrechen

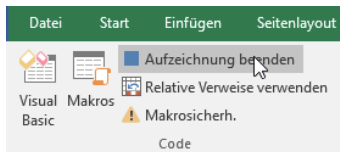
Tragen Sie einen neuen Verbindungstext in die ODBC-Verbindung ein. Löschen Sie den Rest, die Verbindungsinformationen zur Datei und zur Mappe sind in der Verbindungszeichenfolge hinterlegt.



Befehlstext: SELECT \* FROM [Februar\$]

```
SELECT * FROM [Februar$]
```

Stoppen Sie den Makrorecorder, beenden Sie die Makroaufzeichnung.



Schalten Sie mit **Alt+F11** in den Visual Basic Editor. Hier sehen Sie das aufgezeichnete Makro.

```

Microsoft Visual Basic for Applications - Mappe2 - [Modul1 (Code)]
Datei Bearbeiten Ansicht Einfügen Format Debuggen Ausführen Extras Add-Ins Fenster ?
Projekt - VBAProject (Allgemein) ChangeMonth
Option Explicit
Sub ChangeMonth()
    With ActiveWorkbook.Connections("Abfrage von Excel Files").ODBCConn
        .BackgroundQuery = True
        .CommandText = Array("SELECT * FROM [Februar$]")
        .CommandType = xlCmdSql
        .Connection = Array(Array(
            "ODBC;DSN=Excel Files;DBQ=C:\Users\ischels\Desktop\BI Excel\Ums
        ), Array("d=1046;MaxBufferSize=2048;PageTimeout=5;"))
        .RefreshOnFileOpen = False
        .SavePassword = False
        .SourceConnectionFile = ""
        .SourceDataFile = ""
        .ServerCredentialsMethod = xlCredentialsMethodIntegrated
        .AlwaysUseConnectionFile = False
    End With
    With ActiveWorkbook.Connections("Abfrage von Excel Files")
        .Name = "Abfrage von Excel Files"
        .Description = ""
    End With
    ActiveWorkbook.Connections("Abfrage von Excel Files").Refresh
    ActiveWindow.SmallScroll Down:=-6
End Sub

```

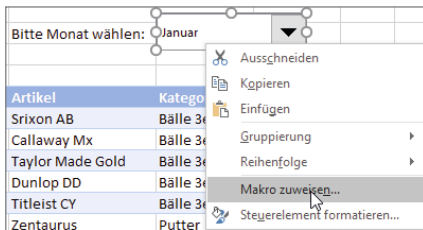
Ändern Sie den `CommandText`, weisen Sie ihm den Inhalt der Zelle K2 zu. Alle weiteren aufgezeichneten Befehle außer der Refresh-Anweisung können Sie zwischen `With` und `End With` löschen.

```

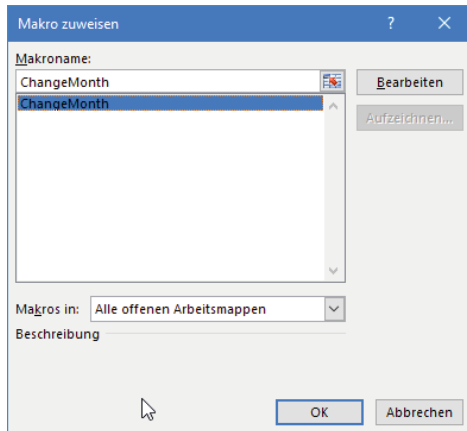
Mappe1 - Modul1 (Code) (Allgemein) ChangeMonth
Option Explicit
Sub ChangeMonth()
    With ActiveWorkbook.Connections("Abfrage von Excel Files").ODBCConnection
        .CommandText = Range("K2")
        ActiveWorkbook.Connections("Abfrage von Excel Files").Refresh
    End With
End Sub

```

Schalten Sie zurück zu Excel und weisen Sie der Schaltfläche über das Kontextmenü das Makro zu.







In den Eigenschaften der Abfrage finden Sie die Option *Spaltenbreiten anpassen*, schalten Sie diese aus. Erstellen Sie eine PivotTable und ein PivotChart mit der Abfrage als Datenquelle (Zeile: *Kategorie*, Wert: *Sales*). Für die Überschrift verknüpfen Sie den Titel mit einer Zelle, in der diese Formel steht:

```
= "Umsatz " & INDEX (Monate; $K$1; 1)
```

M1												
A	B	C	D	E	F	G	H	I	J	K	L	M
											3	Umsatz März
		März									SELECT * FROM [März\$]	
											Category	Summe von Sales
5	Date	ArticleNr	Description	Category	Manufacturer	Sales						
6	19.03.2021 00:00	G-B-0101	Srixon AB	Bälle 3er Pack	Bridgestone	27108					Bälle 3er Pack	199.440
7	11.03.2021 00:00	G-B-0102	Callaway Mx	Bälle 3er Pack	Callaway	38247					Putter	1.106.656
8	08.03.2021 00:00	G-B-0103	Taylor Made Gold	Bälle 3er Pack	Dunlop	48024					Trolley	143.214
9	17.03.2021 00:00	G-B-0104	Dunlop DD	Bälle 3er Pack	Srixon	18015					Wedge	77.745
10	05.03.2021 00:00	G-B-0105	Titleist CY	Bälle 3er Pack	TaylorMade	68045,96					Zubehör	61.853
11	07.03.2021 00:00	G-T-0101	Bad Boy OE-7	Trolley	Bad Boy	53535					<b>Gesamtergebnis</b>	<b>1.588.908</b>
12	01.03.2021 00:00	G-T-0102	Hartl Ecar	Trolley	Hartl	89679						
13	17.03.2021 00:00	G-CL-0154	Forged CNC	Wedge	Komperdell	77745						
14	26.03.2021 00:00	G-Z-0101	Foot Joy Cap	Zubehör	Foot Joy	17633,1						
15	13.03.2021 00:00	G-Z-0102	Foot Joy Regenanz	Zubehör	Foot Joy	44220						
16	25.03.2021 00:00	G-CL-0143	Kramski	Putter	Kramski	208083						
17	09.03.2021 00:00	G-CL-0144	Odyssey XG 2-Ball	Putter	Odyssey	181687						
18	07.03.2021 00:00	G-CL-0145	Optica	Putter	Cobra Golf	135850						
19	12.03.2021 00:00	G-CL-0146	Ping Redwood Z8	Putter	Ping	90447						
20	05.03.2021 00:00	G-CL-0147	Red Arrow	Putter	Yamato	176952						
21	20.03.2021 00:00	G-CL-0148	Scotty Studio	Putter	Scotty	197780						
22	25.03.2021 00:00	G-CL-0149	Torpedo	Putter	Torpedo	115857						
23												
24												
25												
26												
27												
28												

Mit diesem SQL-Befehl holen Sie alle Monatsumsätze in eine weitere Abfrage. Kopieren Sie dazu einfach die erste Abfrage und fügen Sie das Ergebnis in ein neues Blatt ein.

```
SELECT * FROM [Januar$]
UNION ALL SELECT * FROM [Februar$]
UNION ALL SELECT * FROM [März$]
UNION ALL SELECT * FROM [April$]
UNION ALL SELECT * FROM [Mai$]
```

```

UNION ALL SELECT * FROM [Juni$]
UNION ALL SELECT * FROM [Juli$]
UNION ALL SELECT * FROM [August$]
UNION ALL SELECT * FROM [September$]
UNION ALL SELECT * FROM [Oktober$]
UNION ALL SELECT * FROM [November$]
UNION ALL SELECT * FROM [Dezember$]
    
```

Jetzt können Sie eine PivotTable- und eine PivotChart-Auswertung über einzelne Monate oder über das gesamte Jahr erstellen. Für das PivotChart verwenden Sie das Datum im Zeilenbereich, gruppiert nach Monaten. Stellen Sie die Umsatzsumme zweimal in den Wertebereich und formatieren Sie die zweite Reihe mit der Wertfeldeinstellung *Laufende Summe in/Datum*. Als Diagrammtyp verwenden Sie ein Kombidiagramm mit Sekundärachse.

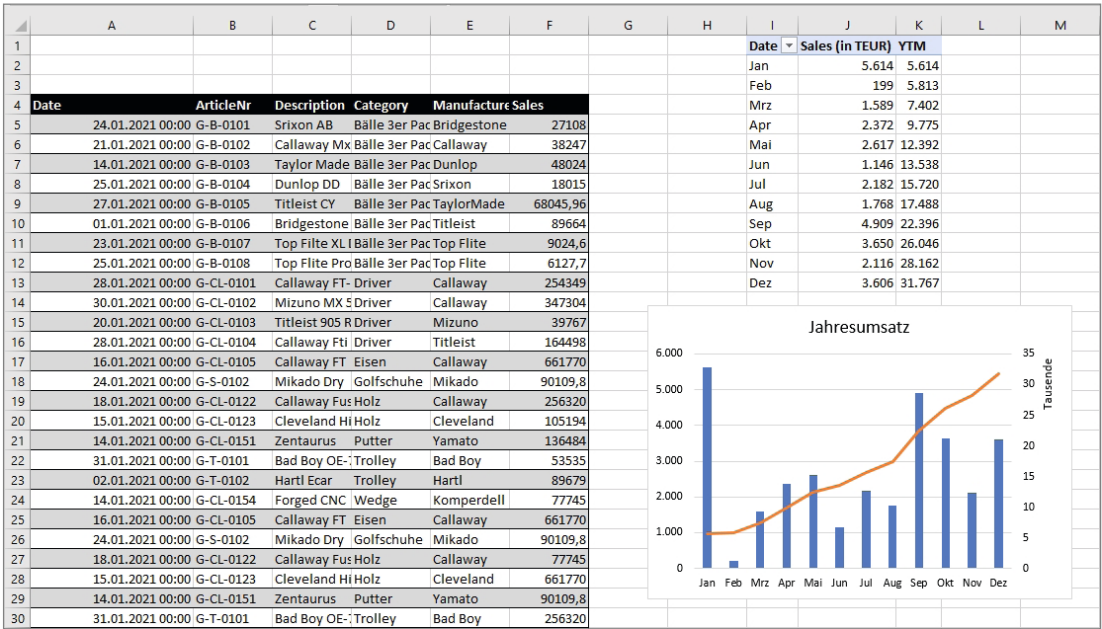


Bild 2.5 PivotTable und PivotChart für die Golfshop-Daten

# 3

## Das Datenmodell

Externe Daten sind in der Praxis selten einzelne Tabellen. Datenbanken zeigen ihre Stärke in den relationalen Beziehungen, OLAP-Systeme sind auf mehrfache Dimensionen und Measures aufgebaut. Excel wäre ein schlechter „Client“, wenn diese Beziehungen beim Import externer Daten nicht akzeptiert oder berücksichtigt würden.

Excel bietet die Möglichkeit, mehrere Tabellen gleichzeitig zu importieren und dabei bestehende Beziehungen zu berücksichtigen. Dazu wird im Arbeitsspeicher ein Datenmodell aufgebaut. Die dafür verwendete *xVelocity-in-memory analytics engine* ist der Nachfolger von Vertipaq, beide Werkzeuge wurden für den SQL Server entwickelt. Das Limit für Tabellen liegt bei 2 Millionen Zeilen und bei einer maximalen Dateigröße von 4 GByte.

### ■ 3.1 Regeln für Datenmodelle

- Ein Datenmodell kann eine einzelne Tabelle oder mehrere Tabellen enthalten. Die Anzahl der maximal verfügbaren Tabellen ist vom Speicherplatz abhängig.
- Dem Datenmodell können alle Tabellen aus einer Arbeitsmappe hinzugefügt werden.
- Zusätzlich können Daten aus externen Datenquellen, sowohl aus Datenbanken als auch aus Datenfeeds und Textdateien, hinzugefügt werden.
- Die Anzahl Zeilen in einer Tabelle im Datenmodell ist nur vom verfügbaren Speicherplatz begrenzt.
- Zwischen den Tabellen eines Datenmodells können bis zu fünf Beziehungen erstellt werden.
- Felder, die in den Quelltabellen nicht vorhanden sind, lassen sich als berechnete Felder (Measures) einfügen. Measures können in Power Pivot oder in PivotTables angelegt werden.

## ■ 3.2 Spezifikationen und Einschränkungen

### Excel 2013

- Die 32-Bit-Umgebung von Excel 2013 ist auf einen virtuellen Adressraum von 2 GByte beschränkt. Dieses Adressraum teilen sich Excel selbst, die Arbeitsmappe, die Add-ins und das Datenmodell, das ca. 500 bis 700 MByte beanspruchen kann.
- In der 64-Bit-Umgebung von Excel 2013 gibt es keine Einschränkungen.

### SharePoint Server

- Auf dem SharePoint Server sind die Standardwerte wesentlich geringer als die erlaubten Maximalwerte. Fragen Sie den SharePoint-Administrator nach den Höchstgrenzen für Excel-Dateien mit Datenmodellen.
- Die maximale Dateigröße einer Excel-Arbeitsmappe für das Hochladen in eine Dokumentenbibliothek:
  - 50 MByte (Standard)
  - 2 GByte (maximal)
- Für Webanwendungen muss die maximale Dateigröße vom SharePoint-Administrator konfiguriert werden.

### Microsoft 365

Arbeitsmappen können eine maximale Dateigröße von 250 MByte aufweisen (Werte können sich ändern bzw. in den unterschiedlichen Office 365-Versionen abweichen).

### Power BI

Bis maximal 250 MByte in Power BI Pro oder in der kostenlosen Online-Version.

### 3.2.1 Objekte im Datenmodell

Objekt	Obergrenze
Anzahl Zeichen im Tabellen- oder Spaltennamen	100
Anzahl Tabellen im Datenmodell	2.147.483.647
Anzahl Spalten in einer Tabelle	2.147.483.647
Gleichzeitige Aktualisierungsanforderungen pro Mappe	6
Anzahl Verknüpfungen	5
Anzahl unterschiedlicher Werte pro Spalte	1.999.999.997
Anzahl Zeilen pro Tabelle	1.999.999.997
Länge einer Zeichenfolge	536.870.912 Bytes (512 MB), entspricht 268 435 456 Unicode-Zeichen (256 Megazeichen)

## ■ 3.3 Datenmodell anlegen

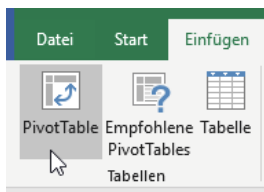
Es gibt mehrere Möglichkeiten, ein Datenmodell anzulegen. Voraussetzung ist immer eine vorliegende Datenquelle, und das kann eine Liste oder (besser) eine Tabelle sein, eine Datenbank oder eine externe Verbindung.

### 3.3.1 Über die PivotTable

Die Quelle kann eine Liste oder eine Tabelle sein. Sie können den Bereich der Liste, einen zugewiesenen Bereichsnamen oder einen Tabellennamen verwenden.

	A	B	C
1	PersNr	Nachname	Vorname
2	00001202	Apak	Elisabeth
3	00001228	Cernjjevic	Werner
4	00001229	Clauß	Wolfgang
5	00001257	Friedl	Fritz

Wählen Sie *Einfügen/Tabellen/PivotTable*.



Bestimmen Sie den Bereich, der für die PivotTable als Datenquelle bestimmt ist. Drücken Sie **F3**, wenn Sie einen Bereichsnamen abholen wollen. Tabellennamen werden automatisch angeboten, wenn der Zellzeiger in einer Tabelle steht.

Kreuzen Sie die Option *Dem Datenmodell diese Daten hinzufügen* an.

