

# ACCESS

**BASICS**

DAS ACCESS-MAGAZIN FÜR ALLE,  
DIE VON 0 AUF 100 WOLLEN



**In diesem Heft:**

**TABELLEN [BASICS]: INTEGRITÄT PER BEZIEHUNG (S. 3)**

**ABFRAGEN [BASICS]: KREUZTABELLENABFRAGEN PER ASSISTENT (S. 6)**

**ABFRAGEN [BASICS]: DATEN AUSWERTEN MIT KREUZTABELLEN (S. 15)**

**ABFRAGEN [BASICS]: GRUPPIERUNGEN NUTZEN (S. 23)**

**ABFRAGEN [BASICS]: 1:N-DATEN IN HAUPT- UND UNTERFORMULAR (S. 32)**

### Dateien im Access- Formular anzeigen

In unserem Shop gibt es ein tolles, neues Produkt: den amvExplorer!

Diese Lösung kannst Du problemlos in alle Access-Datenbanken integrieren. Sie zeigt ein Explorer-Fenster mit den Dateien eines Windows-Ordners an. Damit kannst Du die Dateien zum aktuellen Projekt, zum aktuellen Kunden oder zum aktuellen Produkt anzeigen.

Außerdem kannst Du per Drag and Drop Dateien, Ordner, E-Mails oder auch

Anlagen aus E-Mails hinzufügen. Du kannst sogar tracken, welche Dateioperationen durchgeführt wurden, um Aktionen auszulösen.

Das Ganze völlig ohne zusätzliche Komponenten wie ActiveX oder DLLs!

Überzeuge Dich selbst und hole Dir die Demo von unserer Seite unter folgendem Link:

[amvexplorer.minhorst.com](http://amvexplorer.minhorst.com)

Viel Spaß beim Ausprobieren!

André Minhorst



## IMPRESSUM

### ACCESS [BASICS] WIRD HERAUSGEGEBEN VON:

Minhorst und Minhorst GbR - André Minhorst Verlag  
Borkhofer Straße 17  
47137 Duisburg

Die hier veröffentlichten Texte sind urheberrechtlich geschützt. Übersetzung und Vervielfältigung bedürfen der ausdrücklichen schriftlichen Genehmigung des Verlages. Sämtliche Veröffentlichungen in Access [basics] erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes, auch werden Warennamen ohne Gewährleistung einer freien Verwendung benutzt. André Minhorst Fachverlag für Softwareentwicklung übernimmt für beschriebene oder zum Download bereitstehende Programme weder Gewähr noch Haftung, außer für Vorsatz

oder grobe Fahrlässigkeit. Bezugspreise erfahren Sie auf [www.access-basics.de](http://www.access-basics.de).

### REDAKTION:

André Minhorst (V.i.S.d.P)  
Telefon: 0203/4495577  
E-Mail: [info@access-basics.de](mailto:info@access-basics.de)  
Internet: [www.access-basics.de](http://www.access-basics.de)

Geschäftsführung, Herstellung, Text- und Schlussredaktion,  
Layout von Magazin und Webseite: André Minhorst

Autor: André Minhorst

Fach- und Sprachlektorat: Carsten Gromberg

ISSN: 2190-8761

## Tabellen [basics]: Integrität per Beziehung

Wenn wir im Datenmodell Beziehungen zwischen zwei Tabellen definieren, geschieht das meist unter Anwendung der sogenannten referenziellen Integrität. Was das im Detail bedeutet, haben wir bereits im Artikel **Tabellen [basics]: 1:n-Beziehungen** ([www.access-basics.de/571](http://www.access-basics.de/571)) erläutert. Dort haben wir allerdings den Aspekt etwas stiefmütterlich behandelt, dass ein Fremdschlüsselfeld in einer 1:n-Beziehung mit referenzieller Integrität auch den Wert Null enthalten kann. Was es damit auf sich hat, welche Folgen das haben kann und wie wir darauf reagieren können, erläutern wir im vorliegenden Artikel.

### Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **TabellenBasics\_IntegritaetPerBeziehung.accdb**.

### Beziehung mit referenzieller Integrität

Wir schauen uns in diesem Artikel beispielhaft die Beziehung zwischen den beiden Tabellen **tblBuecher** und **tblKategorien** der Beispieldatenbank an. Diese sind über das Fremdschlüsselfeld **KategorieID** der Tabelle **tblBuecher** und das Primärschlüsselfeld **KategorieID** der Tabelle **tblKategorien** miteinander verknüpft (siehe Bild 1).

An der 1 und dem Unendlich-Zeichen an den Enden der Verbindungslinie zwischen den beiden Tabellen erkennen wir, dass es sich um eine 1:n-Beziehung mit referenzieller Integrität handelt.

Genauer können wir uns das ansehen, indem wir doppelt auf den Beziehungspfeil klicken. Dies öffnet den Dialog **Beziehungen bearbeiten** aus Bild 2. Hier sehen wir die beteiligten Tabellen und Felder sowie die Information, dass referenzielle Integrität definiert ist.

### Nur Primärschlüsselwerte der verknüpften Tabelle oder Null

Dies bedeutet, dass das Fremdschlüsselfeld **KategorieID** der Tabelle nur Werte enthalten darf, die im gleichnamigen Primärschlüsselfeld der Tabelle **tblKategorien** vorkommen – oder den Wert **Null**. Wir können also entweder einen der Datensätze der Ta-

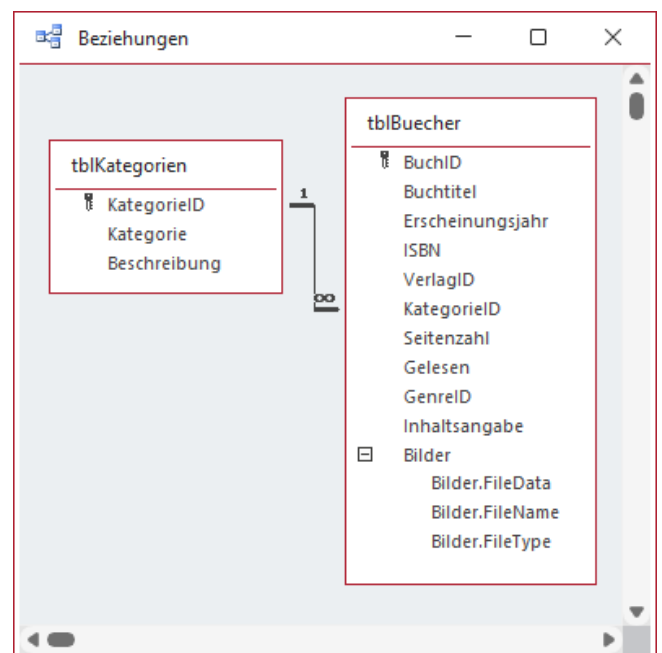


Bild 1: Zwei per 1:n-Beziehung unter referenzieller Integrität verknüpfte Tabellen

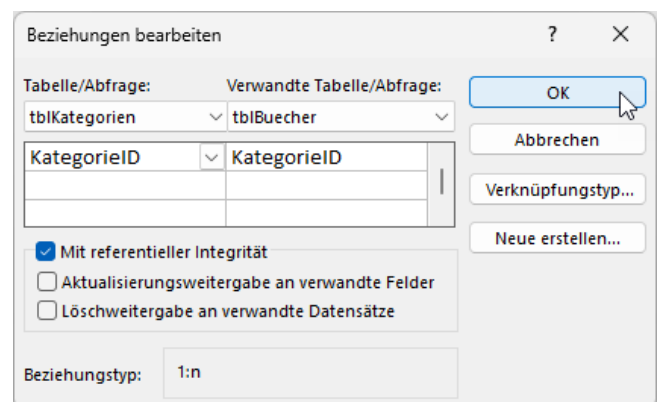


Bild 2: Eigenschaften der Beziehung

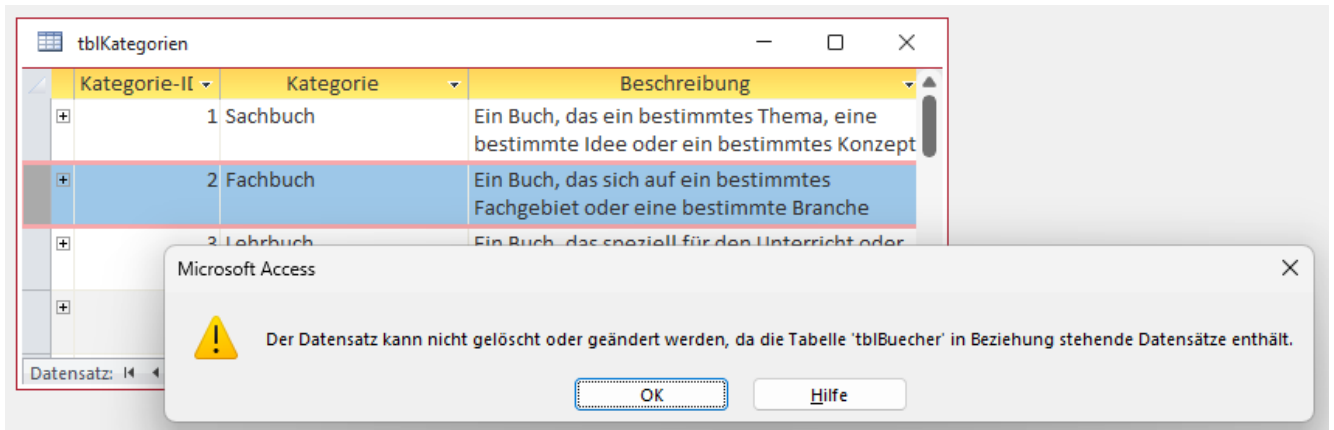


Bild 3: Meldung beim Versuch, einen verknüpften Datensatz zu löschen

belle **tblKategorien** mit einem Buch verknüpfen oder keinen – dann würde das Fremdschlüsselfeld **KategorieID** der Tabelle **tblBuecher** leer bleiben.

### Verknüpfte Datensätze löschen oder nicht

Aktuell ist die Eigenschaft **Löschweitergabe an verwandte Datensätze** nicht aktiviert, was bedeutet, dass wir keinen Eintrag der Tabelle **tblKategorien** löschen können, der bereits mit einem Datensatz der Tabelle **tblBuecher** verknüpft ist. Versuchen wir dies dennoch, lösen wir eine Meldung aus (siehe Bild 3).

Die Alternative wäre, die Option **Löschweitergabe an verwandte Datensätze** zu aktivieren – dann würden beim Löschen eines Datensatzes der Tabelle **tblKategorien** alle mit diesem Datensatz verknüpften Datensätze der Tabelle **tblBuecher** ebenfalls gelöscht werden.

### Der Sinn von Null-Werten in Fremdschlüsselfeldern

Wenn wir eine Beziehung mit dem Nachschlage-Assistenten anlegen, belässt dieser die Einstellungen immer so, dass wir auch bei aktivierter referenzieller Integrität den Wert **Null** in das Fremdschlüsselfeld eingeben können. Wobei wir das genauer formulieren müssen: Null ist genau genommen kein Wert, sondern das Fehlen eines Wertes. Das Feld ist also schlicht leer – was ein Unterschied etwa zu einem Textfeld mit einer leeren Zeichenkette ist.

Faktisch bedeutet dies, dass wir einen Datensatz in der Tabelle **tblBuecher** anlegen können, ohne diesem eine Kategorie zuzuweisen.

Hier muss man sich über die Konsequenzen bewusst werden. Wenn man nun beispielsweise in einer Abfrage alle Bücher mit ihren Kategorien ausgeben möchte, werden solche Datensätze, für die keine Kategorie festgelegt ist, ignoriert.

Es gibt vielleicht Fälle, in denen das sinnvoll ist.

### Einsatzzweck: Import und anschließende Validierung von Daten

Ein tatsächlich sinnvoller Einsatzzweck für eine derartige Gestaltung einer Beziehung ist der Import von Daten, bei denen ansonsten gültige Fremdschlüsselwerte angegeben sind, manche Datensätze aber keinen Wert im Fremdschlüsselfeld enthalten. Dann kann man die Daten zunächst importieren und sie in der Datenbank korrigieren, bevor man nach Abschluss dieser Arbeiten das Anlegen von Datensätzen ohne Fremdschlüsselwert verhindert.

### Wie können wir Null-Werte verhindern?

Die Eigenschaft, die entscheidet über Null-Wert oder nicht, heißt **Eingabe erforderlich**. Wir finden diese im Tabellenentwurf in den Feldeigenschaften des Fremdschlüsselfeldes (siehe Bild 4). Sie hat üblicherweise den Wert **Nein**. Wenn wir ihn auf **Ja** ändern, haben wir

# ACCESS

## BASICS

Schade! Du hast die letzte Seite der Leseprobe dieses Artikels erreicht. Wenn Du den Rest lesen möchtest oder sogar alle über 400 Artikel, die bisher in Access [basics] erschienen sind, hol Dir doch das Jahresabonnement!

[Zum Jahresabo ... hier klicken](#)

## Abfragen [basics]: Kreuztabellenabfragen per Assistent

Im Artikel [Abfragen \[basics\]: Kreuztabellenabfragen per Assistent \(www.access-basics.de/597\)](http://www.access-basics.de/597) haben wir uns angesehen, wie wir Kreuztabellenabfragen mit dem Assistent erstellen. Die Ergebnisse sind brauchbar – erst die Verwendung von Datumsbereichen hat uns an die Grenzen dieses Assistenten gebracht. Deshalb, und weil es noch andere Funktionen gibt, die man nur von Hand erstellen kann, schauen wir uns in diesem Artikel an, was hinter den Kulissen des Kreuztabellenabfragen-Assistenten geschieht und wie wir die Ergebnisse des Assistenten selbst erzeugen können.

### Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **AbfragenBasics\_KreuztabellenabfragenPerAssistent.accdb**.

### Vom Assistenten zum Entwurf

Im Artikel [Abfragen \[basics\]: Kreuztabellenabfragen per Assistent \(www.access-basics.de/597\)](http://www.access-basics.de/597)

haben wir mit dem Assistenten eine Kreuztabellenabfrage erstellt, die das Ergebnis aus Bild 1 liefert.

Nun wollen wir uns den Entwurf dieser Abfrage aus Bild 2 einmal genauer ansehen und nachvollziehen,

Spendenprojekt	Gesamtsum	Qrtl 1	Qrtl 2	Qrtl 3	Qrtl 4
Projekt 1	5.189,00 €	2.005,00 €	561,00 €	1.198,00 €	1.425,00 €
Projekt 2	7.197,00 €	1.673,00 €	1.690,00 €	2.951,00 €	883,00 €
Projekt 3	11.277,00 €	6.046,00 €	433,00 €	3.193,00 €	1.605,00 €
Projekt 4	7.121,00 €	2.533,00 €	2.945,00 €	1.289,00 €	354,00 €
Projekt 5	2.512,00 €	911,00 €	922,00 €		679,00 €
Projekt 6	2.283,00 €	873,00 €	724,00 €	431,00 €	255,00 €
Projekt 7	2.367,00 €	815,00 €	496,00 €	540,00 €	516,00 €
Projekt 8	3.776,00 €	1.700,00 €		1.119,00 €	957,00 €
Projekt 9	5.628,00 €	2.896,00 €	1.952,00 €	658,00 €	122,00 €

Bild 1: Die zu untersuchende Kreuztabelle

Feld:	Spendenprojekt	"Qrtl " & Format([Spendendatum];"q")	Spendenbetrag	Gesamtsumme von Spendenbetrag: Spendenbetrag
Tabelle:	tblSpenden	tblSpenden	tblSpenden	tblSpenden
Funktion:	Gruppierung	Gruppierung	Summe	Summe
Kreuztabelle:	Zeilenüberschrift	Spaltenüberschrift	Wert	Zeilenüberschrift
Sortierung:				
Kriterien:				
oder:				

Bild 2: Die zu untersuchende Kreuztabelle in der Entwurfsansicht

was der Assistent hier genau erledigt hat. Hier sehen wir, dass der Assistent vier Spalten angelegt hat. Außerdem fällt uns im Vergleich zu einer herkömmlichen Abfrage auf, dass wir gleich zwei neue Zeilen namens **Funktion** und **Kreuztabelle** vorfinden.

### Werte für die Zeile Kreuztabelle

Für die Zeile **Kreuztabelle** können wir die folgenden Werte auswählen (siehe Bild 3):

- **Zeilenüberschrift:** Legt die Spalten fest, deren Werte als Zeilenüberschriften ausgegeben werden. Hier können wir offensichtlich ein oder mehrere Spalten verwenden.
- **Spaltenüberschrift:** Gibt die Spalte an, deren Werte als Spaltenüberschriften verwendet werden sollen, in diesem Fall die Quartalsangaben.
- **Wert:** Spalte, welche die eigentlichen Werte liefert.

Im Assistenten durchlaufen wir diese Optionen in den ersten Schritten und legen die Felder fest, aus denen die Werte für die jeweiligen Bereiche stammen sollen. Diese Schritte können wir also auch gut ohne Assistenten durchführen.

### Werte für die Zeile Funktion

Darüber befindet sich die Zeile **Funktion**. Diese wird noch für Abfragen verwendet, in denen Gruppierungen eine Rolle spielen. Diese stellen wir in einem weiteren Artikel namens **Abfragen [basics]: Gruppierungen nutzen** ([www.access-basics.de/593](http://www.access-basics.de/593)) vor.

Im Kontext von Kreuztabellenabfragen werden die dortigen Optionen ähnlich eingesetzt:

- Für Zeilenüberschriften verwenden wir in der Regel den Wert **Gruppierung**.
- Für Spaltenüberschriften nutzen wir ebenfalls **Gruppierung**, außer wir wollen in einer Spalte eine Aggregatfunktion für das Feld oder den Ausdruck

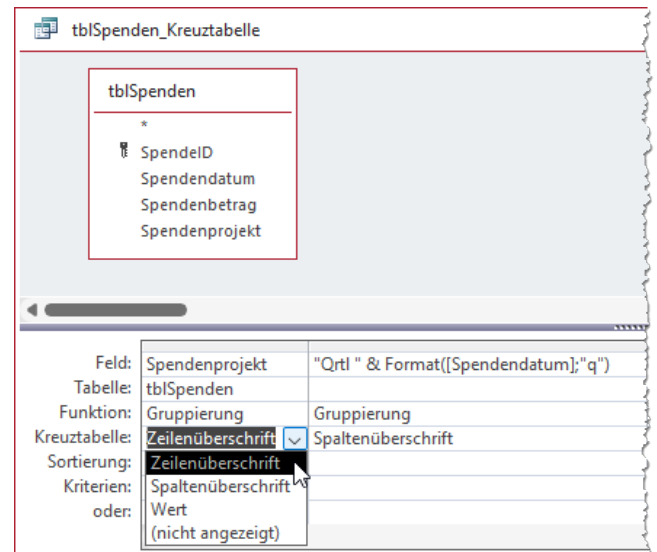


Bild 3: Mögliche Werte für die Zeile Kreuztabelle

anwenden, etwa um wie im Beispiel oben die Gesamtsumme der Spenden für ein Projekt auszugeben. Eine Spaltenüberschrift kombiniert mit einer Aggregatfunktion liefert immer das Ergebnis angewendet auf alle Spalten und die enthaltenen Werte.

- Für Werte können wir alle Aggregatfunktionen außer die Einträge **Gruppierung**, **Ausdruck** oder **Bedingung** nutzen.

### Aggregatfunktionen für die Werte und Spaltenüberschriften

Dabei stehen die folgenden Aggregatfunktionen zur Auswahl (siehe Bild 4):

- **Summe:** Berechnet die Summe aller Werte dieser Spalte entsprechend der Spaltenüberschrift.
- **Mittelwert:** Berechnet den Mittelwert.
- **Min:** Liefert den kleinsten Wert.
- **Max:** Liefert den größten Wert.
- **Anzahl:** Liefert die Anzahl.

- **StAbw.:** Ermittelt die Standardabweichung.
- **Varianz:** Ermittelt die Varianz.
- **ErsterWert:** Holt den Wert aus dem ersten Datensatz.
- **LetzterWert:** Holt den Wert aus dem letzten Datensatz.

### Einfache Kreuztabellenabfrage in der Entwurfsansicht anlegen

Wenn wir eine neue Kreuztabellenabfrage in der Entwurfsansicht erstellen wollen, beginnen wir mit einer neuen, leeren Abfrage in der Entwurfsansicht.

Dann wählen wir im Ribbon unter **Abfrageentwurf** im Bereich **Abfragetyp** den Eintrag **Kreuztabelle** aus (siehe Bild 5).

Dadurch blenden wir in der ansonsten noch leeren Abfrage die beiden Zeilen **Funktion** und **Kreuztabelle** im Entwurfsraster ein (siehe Bild 6).

### Tabellen oder Abfragen hinzufügen

Im Gegensatz zum Kreuztabellenabfrage-Assistenten zeigt sich hier bereits ein Vorteil: Während wir beim Kreuztabellenabfragen-Assistenten vor dem Erstellen bereits die vollständige Datenherkunft definiert haben müssen, was unter Umständen das Erstellen einer Abfrage mit den benötigten Tabellen erforderlich machte, brauchen wir die benötigten Tabellen und Abfragen beim Erstellen über die Entwurfsansicht nicht bereits vor-

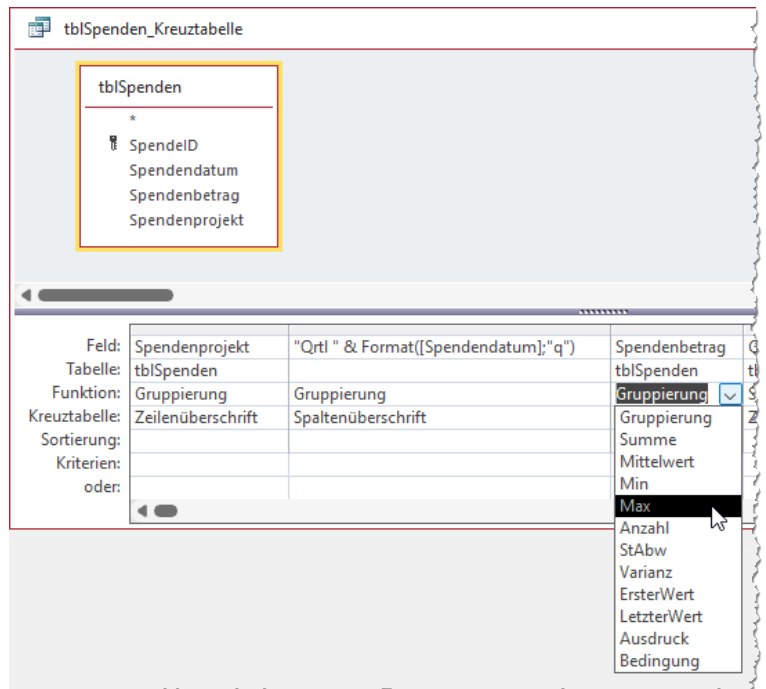


Bild 4: Mögliche Werte für die Aggregatfunktion der Werte

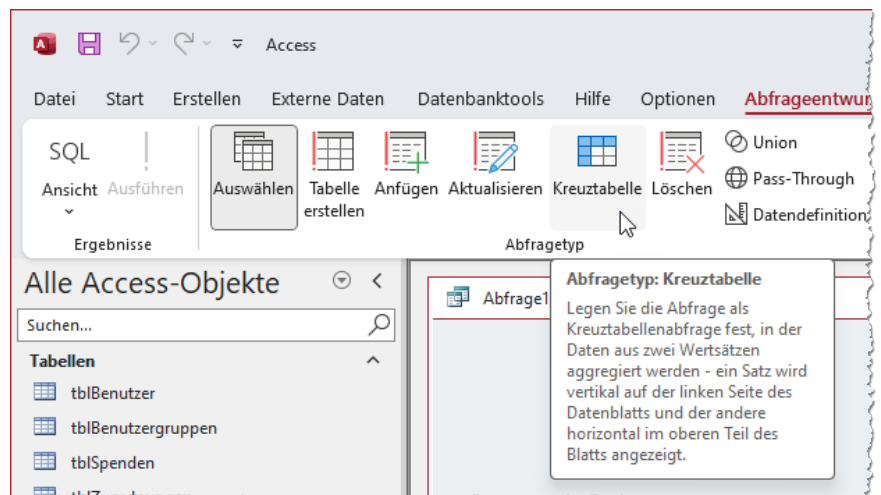


Bild 5: Anlegen einer Kreuztabellenabfrage in der Entwurfsansicht

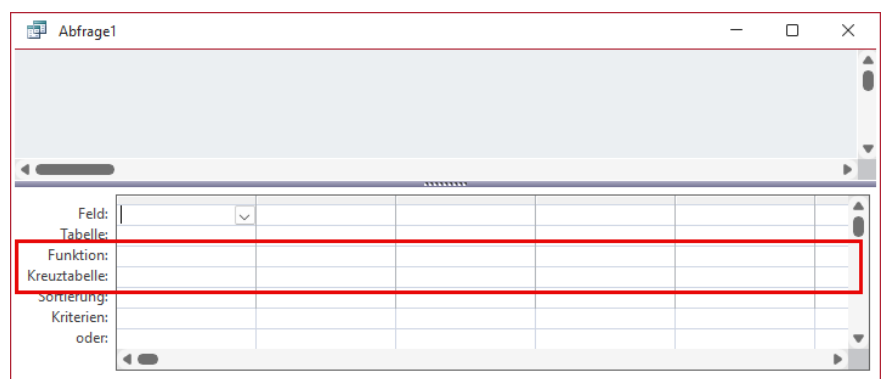


Bild 6: Zwei neue Zeilen im Entwurfsraster



her bereitzustellen. Wir können diese einzeln zur Entwurfsansicht hinzufügen.

### Tabellen oder Abfragen zum Abfrageentwurf hinzufügen

Im zweiten Schritt fügen wir die Tabellen oder Abfragen zur Abfrage hinzu, deren Felder wir für die Darstellung in der Kreuztabelleabfrage benötigen. Es kann sein, dass wir mit einer einzigen Tabelle auskommen, vielleicht benötigen wir aber auch Daten aus mehreren, miteinander verknüpften Tabellen.

Die benötigten Tabellen oder Abfragen können wir direkt aus dem Navigationsbereich in den Abfrageentwurf ziehen oder auch aus dem Bereich **Tabellen hinzufügen** (siehe Bild 7), den wir über den Ribbon-Eintrag **Abfrageentwurf|Abfragesetup|Tabellen hinzufügen** einblenden. Eine Suchfunktion zum Filtern nach dem Objektnamen bieten beide Ansichten.

In unserem Beispiel fügen wir die Tabelle **tblSpenden** zum Abfrageentwurf hinzu und speichern die Abfrage erst einmal unter dem Namen **qrySpendenNachDatumUndProjekt**.

Wir wollen diesmal die Datumsangaben als Zeilenüberschriften anzeigen, die Projekte als Spaltenüberschriften und die Spendenbeträge als Werte.

Also ziehen wir als Erstes das Feld **Spendendatum** in das Entwurfsraster und stellen in der Zeile **Funktion** den Wert

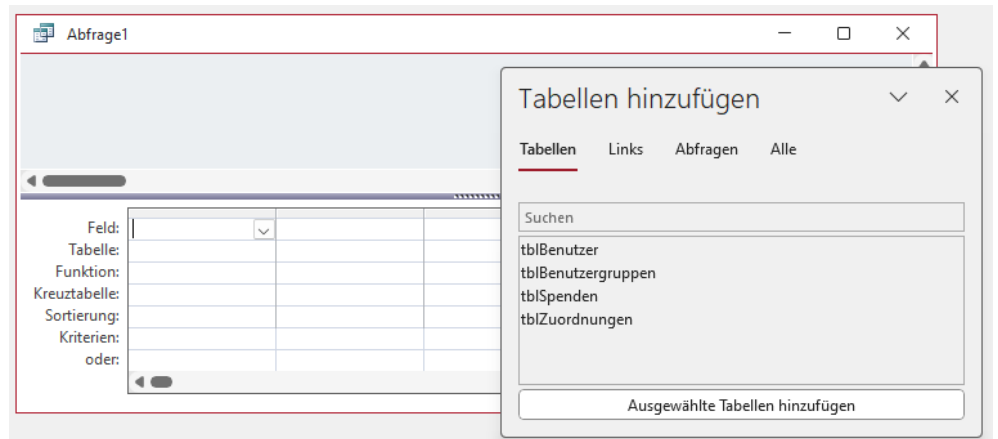


Bild 7: Tabellen und Abfragen zur Abfrage hinzufügen

**Gruppierung** und in der Zeile **Kreuztabelle** den Wert **Zeilenüberschrift** ein.

Dann ziehen wir das Feld **Spendenprojekt** hinein und stellen **Funktion** ebenfalls auf **Gruppierung** und **Kreuztabelle** auf **Spaltenüberschrift** ein. Schließlich fügen wir das Feld **Spendenbetrag** hinzu und legen für die beiden relevanten Zeilen die Werte **Summe** und **Wert** ein (siehe Bild 8).

Damit erhalten wir nach einem Wechsel in die Datenblattansicht die Ansicht aus Bild 9. Hier können wir

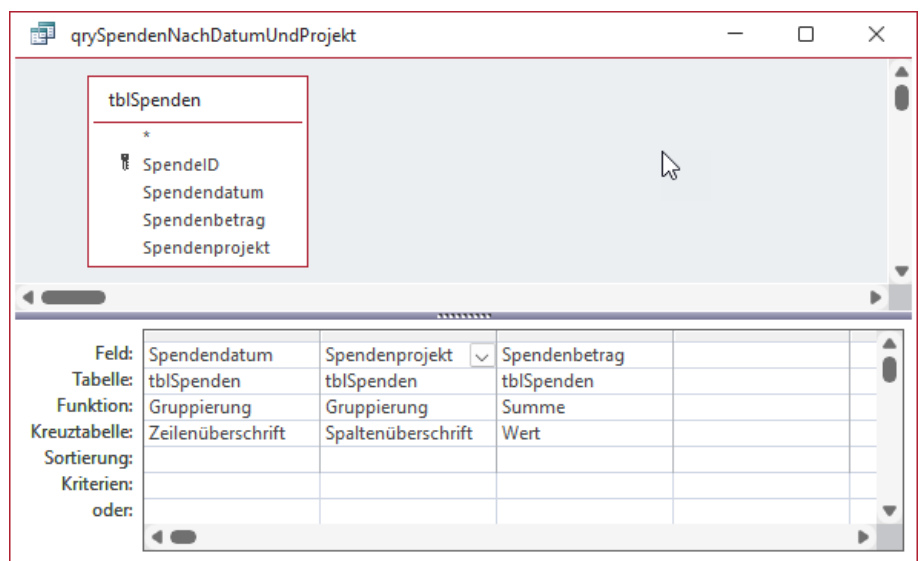


Bild 8: Felder zum Entwurfsraster hinzufügen

# ACCESS

## BASICS

Schade! Du hast die letzte Seite der Leseprobe dieses Artikels erreicht. Wenn Du den Rest lesen möchtest oder sogar alle über 400 Artikel, die bisher in Access [basics] erschienen sind, hol Dir doch das Jahresabonnement!

[Zum Jahresabo ... hier klicken](#)

## Abfragen [basics]: Daten auswerten mit Kreuztabellen

Die übliche Datenblattansicht von Access ist immer gleich aufgebaut: Sie enthält als Spaltenköpfe die Feldnamen und in den Zeilen die einzelnen Datensätze. In manchen Fällen ist jedoch eine andere Ansicht erforderlich: wenn man die Inhalte zweier Felder einer Tabelle als Zeilen- und Spaltenköpfe verwenden möchte und den Inhalt eines dritten Feldes als eigentliche Werte. Dabei haben wir sogar noch den Bonus, dass wenn wir mehrere Datensätze mit den gleichen Werten für die als Spalten- und Zeilenköpfe verwendeten Felder haben, die eigentlichen Werte sogar noch mit verschiedenen Funktionen aggregieren können. Wie wir solche Kreuztabellen einrichten, zeigen wir in diesem Artikel.

### Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **AbfragenBasics\_Kreuztabellenabfragen.accdb**.

### Warum Kreuztabellen?

Übliche Tabellen speichern Datensätze mit Werten für die verschiedenen Felder. Die Datenblattansicht reicht in vielen Fällen bereits für die Eingabe von Daten aus, oft wird jedoch ein Detailformular verwendet, das jeweils einen Datensatz übersichtlich anzeigt. Auch Kombinationen von beiden mit Daten aus mehreren Tabellen gleichzeitig sind möglich.

In Excel könnte man jedoch auch sogenannte Kreuztabellen einfach anlegen: Man trägt einfach Bezeichnungen für die Spalten- und Zeilenköpfe ein und die Werte landen in der dadurch aufgespannten Tabelle.

Ein bekanntes Beispiel aus der Welt des Sports sind die Tabellen, die alle Ergebnisse zwischen den Teams einer Liga darstellen. Zeilen- und Spaltenköpfe enthalten dabei jeweils eine Auflistung aller Teams, die eigentlichen Daten sind die Ergebnisse zwischen diesen Teams.

Andere einfache Beispiele sind Darstellungen von bestimmten Daten nach bestimmten Eigenschaften wie:

- Anzahl der Schulabschlüsse nach Geschlecht

- Wahlergebnisse für verschiedene Parteien nach Altersgruppen
- Umsätze je Produktkategorie pro Quartal
- Zuordnung von Benutzern zu Benutzergruppen

### Voraussetzung für eine Kreuztabelle

Um eine Kreuztabelle erstellen zu können, benötigen wir in einer einzelnen oder mehreren verknüpften Tabellen mindestens drei Felder:

- ein Feld für die Spaltenüberschriften
- ein Feld für die Zeilenüberschriften
- ein Feld, das die anzuzeigenden und gegebenenfalls zu aggregierenden Daten enthält

Wir starten mit einem einfachen Beispiel. Dabei wollen wir einfach die Benutzer zu Benutzergruppen zuordnen und in einer Kreuztabellenabfrage die Zugehörigkeiten übersichtlich abbilden. Dazu benötigen wir die drei Tabellen aus Bild 1.

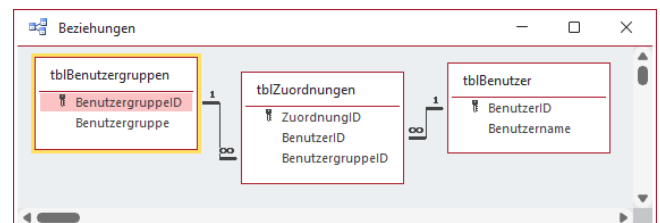


Bild 1: Beziehung zwischen den Beispieldatenbanken

Die Tabelle **tblBenutzer** nimmt die Namen der Benutzer auf, die Tabelle **tblBenutzergruppen** die Bezeichnungen der Benutzergruppen. Über die m:n-Beziehungstabelle verknüpfen wir die Datensätze der beiden Tabellen miteinander.

Ein Datensatz in dieser Tabelle enthält immer die Zuordnung eines Benutzers zu einer Benutzergruppe.

Damit jeder Benutzer nur einmal in jede Benutzergruppe eingetragen werden kann, definieren wir einen zusammengesetzten Index für die beiden Felder **BenutzerID** und **BenutzergruppeID**.

Nachdem wir den Tabellen **tblBenutzer** und **tblBenutzergruppen** einige Datensätze hinzugefügt haben, können wir diese über die dafür angelegten Nachschlagefelder in der Verknüpfungstabelle **tblZuordnungen** auswählen und somit die Zugehörigkeit der Benutzer zu den Benutzergruppen definieren (siehe Bild 2).

### Kreuztabelle per Assistent erstellen

In diesem Artikel schauen wir uns an, wie wir eine Kreuztabelle per Assistent erstellen.

Diesen finden wir, wenn wir im Ribbon unter **Erstellen** im Bereich **Abfragen**

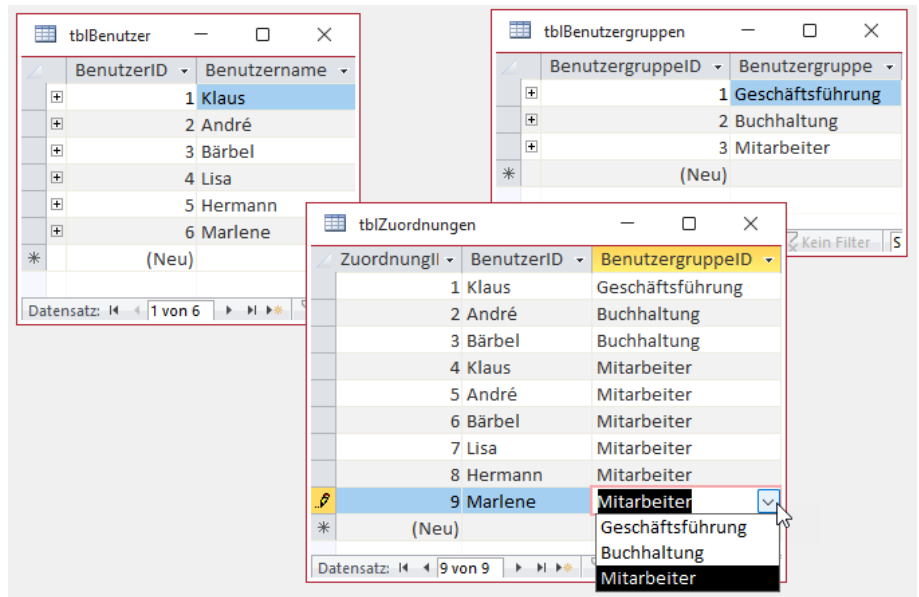


Bild 2: Daten in den drei Tabellen zur Benutzerverwaltung

gen auf die Schaltfläche **Abfrage-Assistent** klicken. Dies öffnet den Dialog **Neue Abfrage**, in dem wir den Eintrag **Kreuztabellen-Assistent** auswählen und bestätigen (siehe Bild 3).

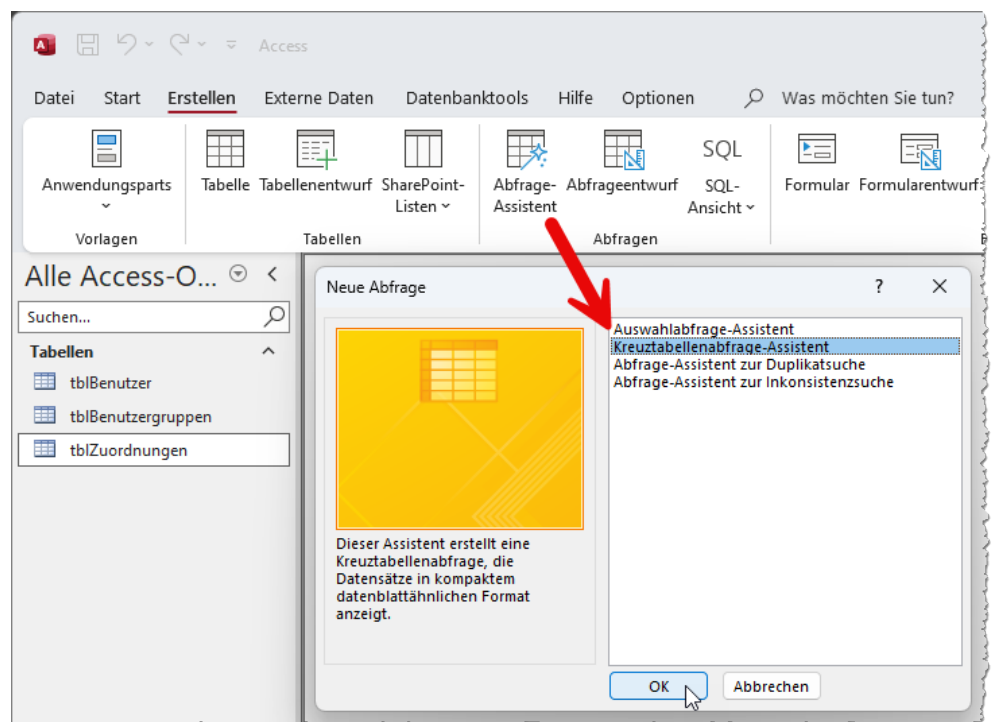


Bild 3: Start des Kreuztabellen-Assistenten

#### Tabelle auswählen

Im ersten Schritt des Assistenten wählen wir aus, welche Tabelle oder Abfrage die Daten für die Kreuztabelle liefern soll. Die Tabelle **tblZuordnungen** enthält grundsätzlich alle Informationen, die uns zur Verfügung stehen. Also wählen wir diese aus (siehe Bild 4).

Aber haben wir wirklich alle Felder, die wir brauchen? Das Feld **BenutzerID** können wir für die Zeilenüberschriften verwenden, das Feld **BenutzergruppeID** für die Spaltenüberschriften. Aber welches Feld soll die eigentlichen Werte abbilden? Dazu nutzen wir einfach das Primärschlüsselfeld **ZuordnungID**. Wie, das beschreiben wir gleich.

#### Feld für die Zeilenüberschriften auswählen

Nachdem wir die Tabelle festgelegt haben, fragt der Assistent uns nach dem Feld, das die Zeilenüberschriften enthält. Hier wählen wir wie in Bild 5 das Feld **BenutzergruppeID** aus.

Wie wir hier sehen, können wir auch mehr als ein Feld für die Spaltenüberschrift auswählen. Dieses wir dann einfach in der nächsten Spalte angezeigt.

#### Feld für die Spaltenüberschrift auswählen

Im nächsten Schritt legen wir fest, welches Feld die Spaltenüberschriften für die Kreuztabelle liefern soll. Das ist in unserem Fall

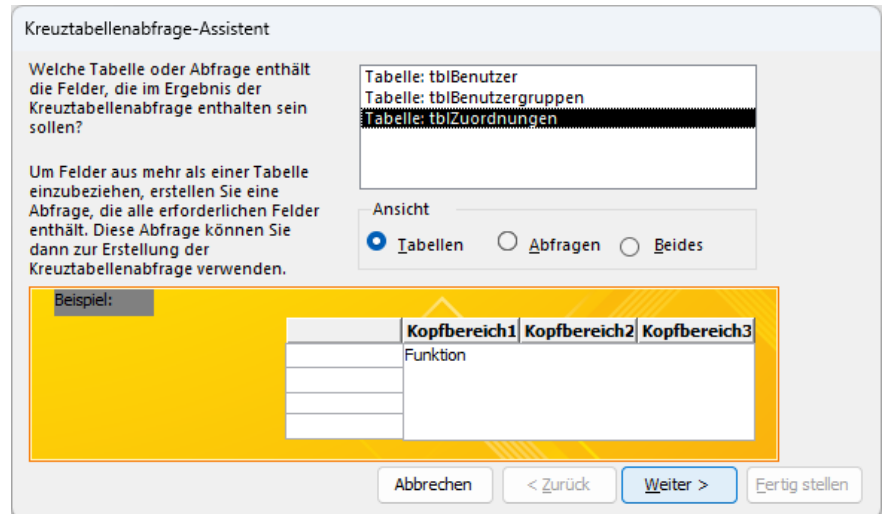


Bild 4: Auswahl der Tabelle mit den Quelldaten

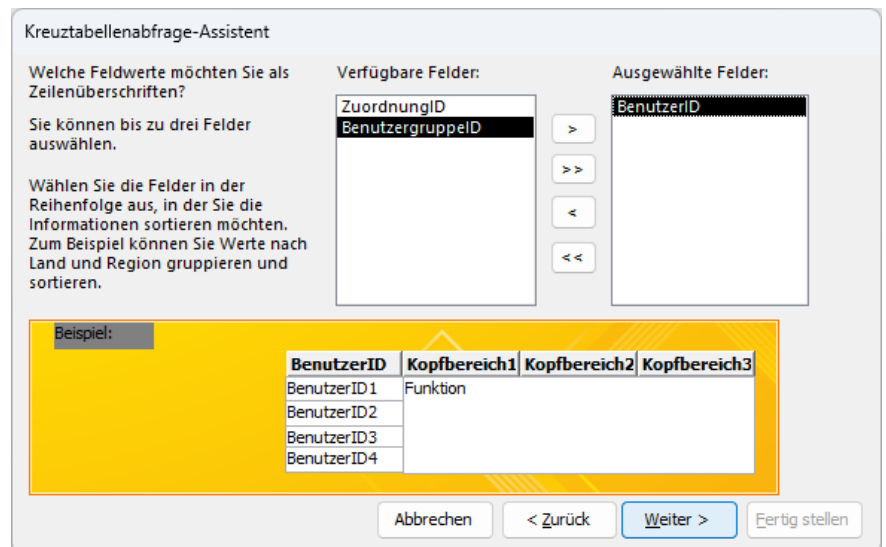


Bild 5: Auswahl des Feldes, das die Zeilenüberschriften liefern soll

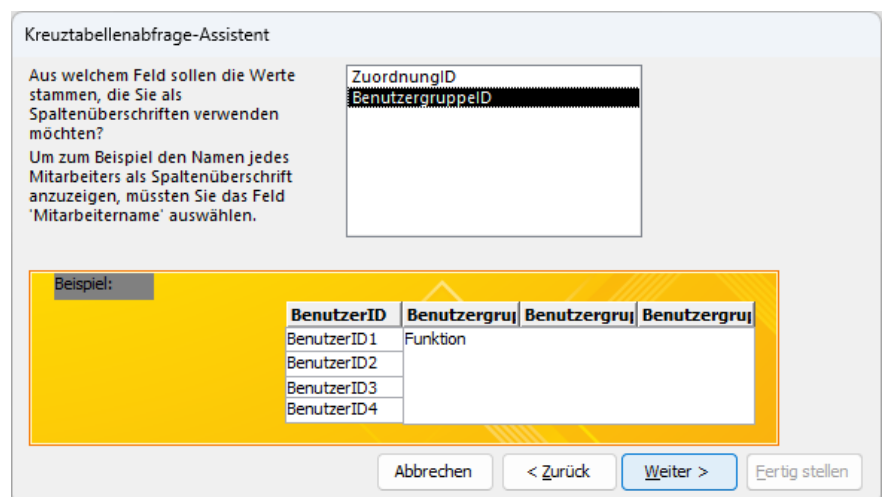


Bild 6: Auswahl des Feldes, das die Spaltenüberschriften liefern soll

das Feld **BenutzergruppeID**. Nach der Auswahl dieses Feldes zeigt der Assistent bereits die Zeilen- und Spaltenüberschriften in einer Vorschau an (siehe Bild 6).

### Feld für die anzuzeigenden Werte auswählen

Damit kommen wir zu dem Feld, dessen Daten zwischen den als Zeilen- und Spaltenüberschriften ausgewählten Feldern angezeigt werden soll. Eigentlich haben wir dazu gar kein eigenes Feld.

Aber da wir, wie wir in Bild 7 sehen, ohnehin eine Funktion festlegen müssen, können wir hier einfach das verbleibende Feld **ZuordnungID** wählen.

Nutzen wir dieses in Kombination mit der Funktion **Anzahl**, werden wir immer den Wert **1** an den Schnittpunkten der Zeilenüberschrift und der Spaltenüberschrift erhalten, wo der Benutzer der entsprechenden Benutzergruppe zugeordnet ist.

Da wir, wie oben beschreiben, einen eindeutigen Index für die Kombination der beiden Felder **BenutzerID** und **BenutzergruppeID** in der Tabelle **tblZuordnungen** definiert haben, werden wir immer nur keine oder eine Zuordnung für jede Kombination erhalten.

Also behalten wir das verbleibende Feld **ZuordnungID** mit der Funktion **Anzahl** bei und gehen zum nächsten Schritt weiter. Die Option **Gesamtsumme jeder Zeile berechnen** behalten wir zunächst bei, um zu sehen, wie sich dies auf das Ergebnis auswirkt.

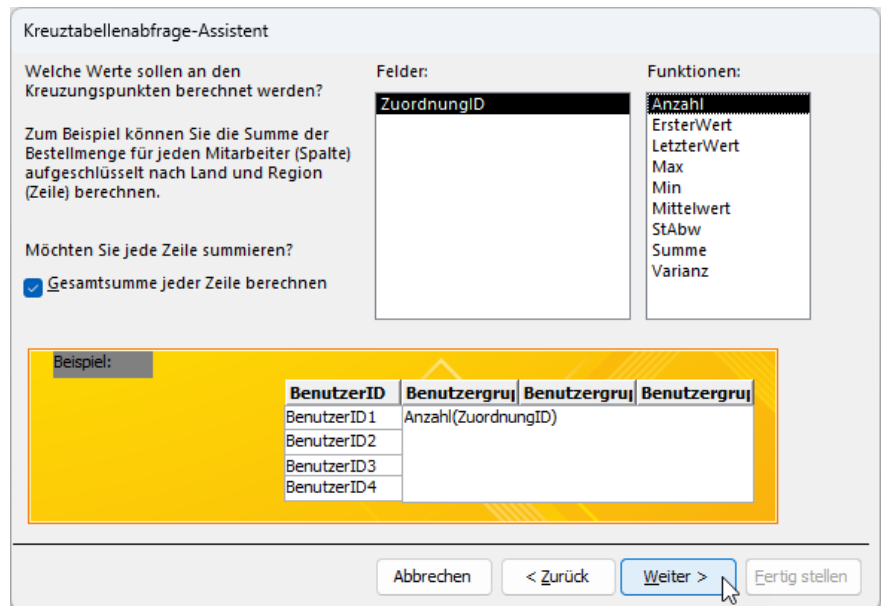


Bild 7: Auswahl des Feldes, das die eigentlichen Werte liefern soll

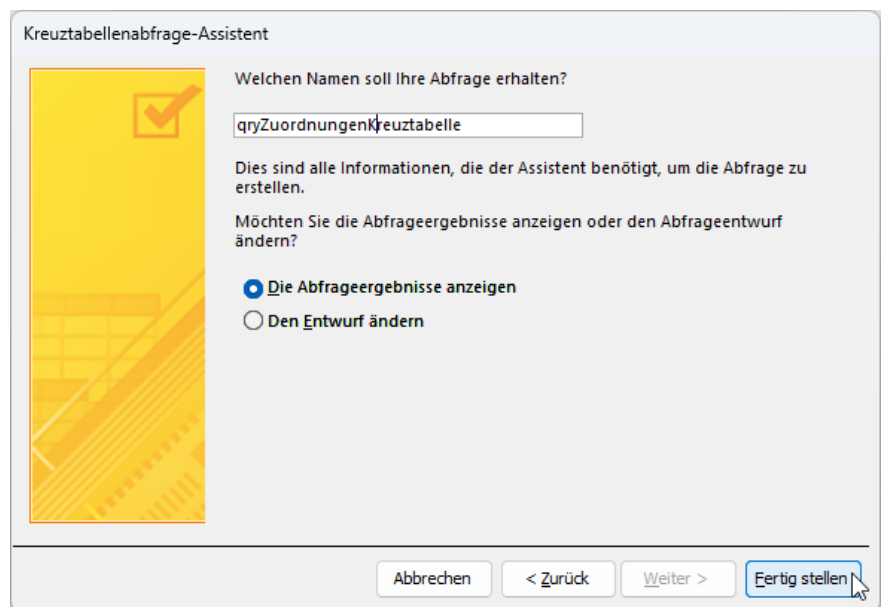


Bild 8: Letzer Schritt des Assistenten

### Abschluss des Assistenten

Damit landen wir auf der letzten Seite, wo wir nur noch festlegen können, wie die zu erstellende Abfrage heißen soll und ob wir die Abfrage im nächsten Schritt zur Anzeige der Ergebnisse oder in der Entwurfsansicht öffnen wollen. Wir entscheiden uns

# ACCESS

## BASICS

Schade! Du hast die letzte Seite der Leseprobe dieses Artikels erreicht. Wenn Du den Rest lesen möchtest oder sogar alle über 400 Artikel, die bisher in Access [basics] erschienen sind, hol Dir doch das Jahresabonnement!

[Zum Jahresabo ... hier klicken](#)

## Abfragen [basics]: Gruppierungen nutzen

Abfragen liefern meist immer je einen Datensatz entsprechend der Anzahl der Datensätze der zugrunde liegenden Tabellen. Dabei werden in der Regel die Werte aus den Feldern der Datenquelle abgebildet. Manchmal führen wir in Abfragen Operationen auf diesen Feldern aus, beispielsweise um die Inhalte mehrerer Felder zusammenzuführen oder diese zu formatieren. Wir können in einer Abfrage jedoch auch mehrere Datensätze der Datenquelle zusammenfassen – beispielsweise, um gleichartige Datensätze nur einmal abzubilden oder um die Werte eines oder mehrerer Felder ansonsten gleicher Datensätze zusammenzufassen. So können wir Summen, Mengen und andere Funktionen anwenden. Die Abfragen, mit denen wir solche Ansichten realisieren, nennen sich Gruppierungsabfragen. Dieser Artikel liefert die Grundlagen zum Erstellen solcher Abfragen.

### Beispieldatenbank

Die Beispiele dieses Artikels finden Sie in der Datenbank **AbfragenBasics\_Gruppierungen.accdb**.

### Beispieltabelle

Als Beispieltabelle verwenden wir die Tabelle **tblSpenden**, die wie in Bild 1 aussieht und neben dem Primärschlüsselfeld namens **SpendeID** drei weitere Felder namens **Spendendatum**, **Spendenbetrag** und **Spendenprojekt** enthält.

SpendeID	Spendendatum	Spendenbetrag	Spendenprojekt
1	18.02.2022	580,00 €	Projekt 3
2	17.03.2021	775,00 €	Projekt 2
3	16.01.2023	815,00 €	Projekt 3
4	07.03.2020	415,00 €	Projekt 2
5	28.02.2023	374,00 €	Projekt 5
6	27.06.2023	57,00 €	Projekt 6
7	15.06.2021	525,00 €	Projekt 9
8	19.03.2020	593,00 €	Projekt 8
9	11.03.2021	623,00 €	Projekt 9
10	20.01.2021	280,00 €	Projekt 4

Bild 1: Tabelle mit Beispieldaten für Gruppierungsabfragen

Diese bietet ausreichend Möglichkeiten zum Gruppieren, zum Beispiel nach dem Spendendatum oder nach dem Spendenprojekt.

### Basisabfrage erstellen

Als Grundlage für die folgenden Beispiele verwenden wir eine neue Abfrage, die wir über den Ribbonbefehl **Abfrageentwurf** des Bereichs **Abfragen** im Ribbontab **Erstellen** anlegen.

Dieser Abfrage fügen wir die Tabelle **tblSpenden** hinzu und

Feld:	SpendeID	Spendendatum	Spendenbetrag	Spendenprojekt
Tabelle:	tblSpenden	tblSpenden	tblSpenden	tblSpenden
Sortierung:				
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:				
oder:				

Bild 2: Basisabfrage als Basis für die Gruppierungen



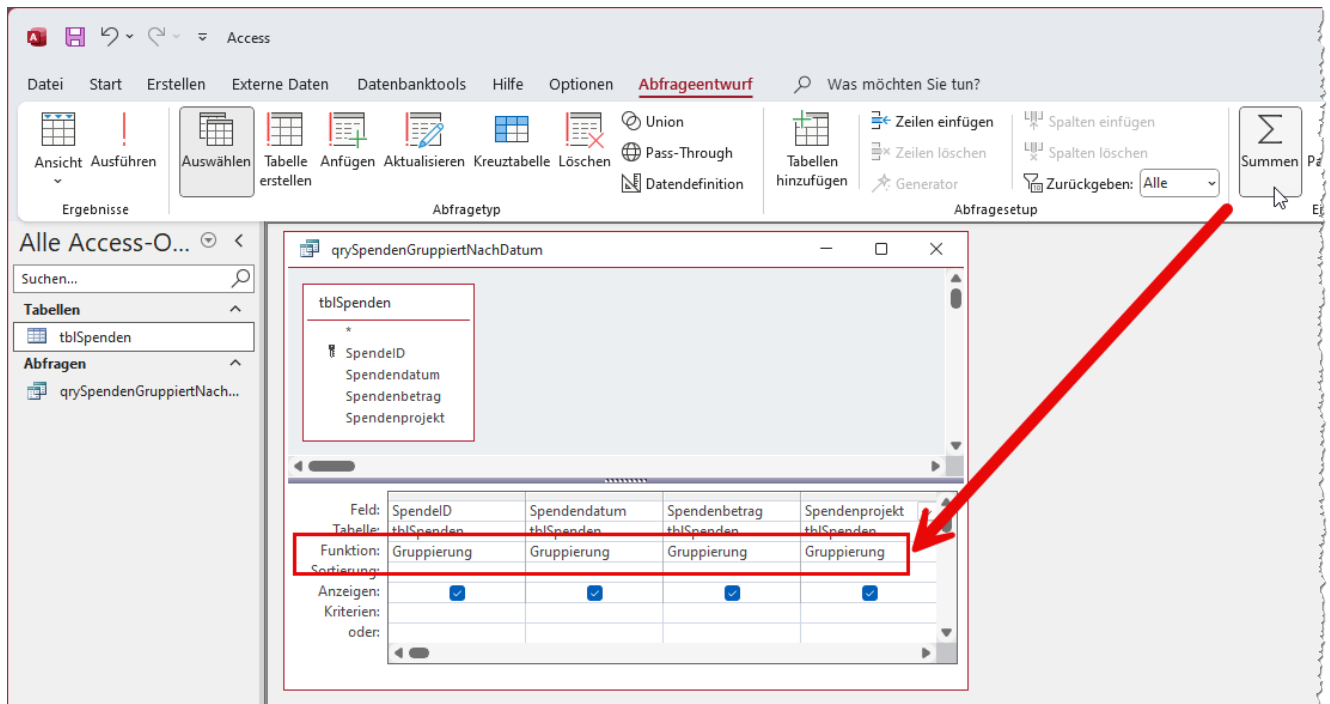


Bild 3: Einblenden der Zeile Funktion mit dem Wert Gruppierung

ziehen alle enthaltenen Felder in das Entwurfsraster (siehe Bild 2).

### Gruppieren-Funktion aktivieren

Nun stellt sich die Frage: Wie wollen wir die Datensätze dieser Abfrage gruppieren? Dazu wechseln wir zum Ribbontab **Abfrageentwurf** und wähle aus dem Bereich **Einblenden/Ausblenden** den Befehl **Summen** aus. Dies blendet die Zeile **Funktion** ein, die für alle Felder zunächst den Wert **Gruppierung** anzeigt (siehe Bild 3).

Wie wirkt sich das auf das Abfrageergebnis aus? In der aktuellen Konstellation gar nicht, wir erhalten alle 100 Datensätze, die wir auch in der Tabelle vorfinden. Wozu dann überhaupt gruppieren? In diesem Beispiel ist das Gruppieren für keinen Zweck zielführend, denn die Gruppierung bezieht sich immer auf die Gesamtheit der abgebildeten Felder. Und da wir aktuell das Primärschlüsselfeld in das Abfrageergebnis einbeziehen, bekommen wir immer alle Datensätze

zurückgeliefert, denn das Primärschlüsselfeld enthält immer eindeutige Werte.

### Gruppierte Abfragen können nicht bearbeitet werden

Was wir jedoch bereits feststellen können: Das Ergebnis einer Abfrage mit einer Gruppierung kann nicht mehr bearbeitet werden und wir können auch keine neuen Datensätze hinzufügen – siehe Bild 4.

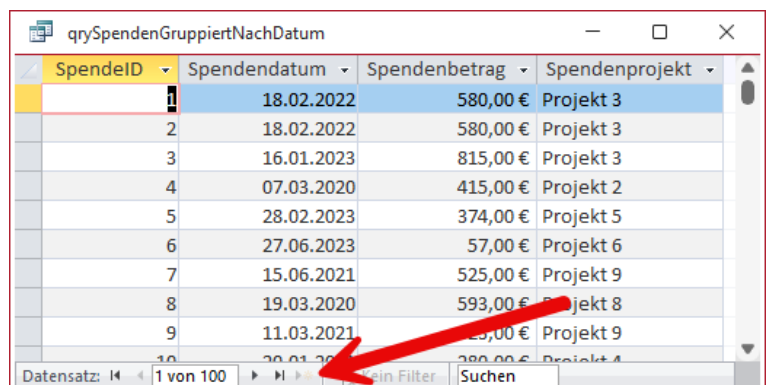


Bild 4: Die Datensätze einer gruppierten Abfrage können weder bearbeitet werden noch können wir neue Datensätze hinzufügen.

### Reine Gruppierungsfunktion

Wenn wir nur die Gruppierungsfunktion nutzen wollen, also Datensätze zusammenfassen wollen, deren abgebildete Felder alle gleich sind, müssen wir also das Primärschlüsselfeld zwingend aus dem Abfrageentwurf entfernen. In diesem Beispiel würde sich das allerdings erst auf das Ergebnis auswirken, wenn wir mindestens einen Datensatz verwenden, dessen Inhalte aller Felder mit dem eines weiteren Datensatzes übereinstimmen. Um das nachzustellen, haben wir die Inhalte der ersten beiden Datensätze angeglichen (siehe Bild 5).

Erstellen wir nun eine gruppierte Abfrage mit allen Feldern außer dem Primärschlüsselfeld, erhalten wir den Datensatz nur noch einmal im Abfrageergebnis (siehe Bild 6).

Auf diese Weise können wir beispielsweise ermitteln, wie viele eindeutige Datensätze es bezüglich der abgebildeten Felder gibt. Noch interessanter wäre es, wenn wir noch herausfinden könnten, wie viele Datensätze es von jeder Kombination gibt. Dazu haben wir von einem weiteren Datensatz drei identische Kopieren bezüglich der drei Felder Spendendatum, Spendenbetrag und Spendenprojekte erstellt (siehe Bild 7).

### Gruppierung mit Anzahl

Um die Anzahl der gleichen Datensätze bezüglich der drei Felder **Spendendatum**, **Spendenbetrag** und **Spendenprojekt** zu ermitteln, benötigen wir ein weiteres Feld. Dieses kann ein beliebiges Feld sein – sogar das Primärschlüsselfeld **SpendeID** kann verwendet werden. Also fügen wir dieses als viertes Feld hinzu. Für dieses Feld stellen wir nun in der Zeile **Funktion** den Wert **Anzahl** ein.

Damit wir die Gruppierungen mit der größten Anzahl gleich oben im Abfrageergebnis wiederfinden, stellen wir für die Zeile **Sortierung** in der Spalte **SpendeID**

SpendeID	Spendendatum	Spendenbetrag	Spendenprojekt
1	18.02.2022	580,00 €	Projekt 3
2	18.02.2022	580,00 €	Projekt 3
3	16.01.2023	815,00 €	Projekt 3
4	07.03.2020	415,00 €	Projekt 2
5	28.02.2023	374,00 €	Projekt 5

Bild 5: Zwei bis auf das Primärschlüsselfeld identische Datensätze

Spendendatum	Spendenbetrag	Spendenprojekt
20.12.2021	208,00 €	Projekt 3
25.01.2022	965,00 €	Projekt 3
17.02.2022	561,00 €	Projekt 9
18.02.2022	580,00 €	Projekt 3
02.03.2022	157,00 €	Projekt 4
02.03.2022	495,00 €	Projekt 1
04.03.2022	815,00 €	Projekt 7

Bild 6: Die beiden Datensätze werden nun so zusammengefasst.

SpendeID	Spendendatum	Spendenbetrag	Spendenprojekt
1	18.02.2022	580,00 €	Projekt 3
2	18.02.2022	580,00 €	Projekt 3
3	16.01.2023	815,00 €	Projekt 3
4	16.01.2023	815,00 €	Projekt 3
5	16.01.2023	815,00 €	Projekt 3
6	27.06.2023	57,00 €	Projekt 6
7	15.06.2021	525,00 €	Projekt 9

Bild 7: Ein Datensatz mit zwei und einer mit drei Dubletten.

Feld:	Spendendatum	Spendenbetrag	Spendenprojekt	SpendeID
Tabelle:	tblSpenden	tblSpenden	tblSpenden	tblSpenden
Funktion:	Gruppierung	Gruppierung	Gruppierung	Anzahl
Sortierung:				Absteigend
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:				
oder:				

Bild 8: Entwurf einer Abfrage mit der Anzahl eines Feldes

noch den Wert **Absteigend** ein (siehe Bild 8).

Nach dem Wechsel in die Datenblattansicht finden wir oben zuerst die Gruppe mit drei Vorkommen des gleichen Datensatzes und darunter die Gruppe mit den zwei gleichen Datensätzen vor (siehe Bild 9).

Spendendatum	Spendenbetrag	Spendenprojekt	AnzahlvonSpendeID
16.01.2023	815,00 €	Projekt 3	3
18.02.2022	580,00 €	Projekt 3	2
11.05.2021	496,00 €	Projekt 7	1
22.11.2020	585,00 €	Projekt 3	1
20.01.2021	280,00 €	Projekt 4	1
29.01.2021	57,00 €	Projekt 1	1
02.02.2021	674,00 €	Projekt 4	1

Bild 9: Anzahl der gleichen Datensätze

### Mögliche Aggregatfunktionen

Neben der Summe der Werte eines Feldes der Gruppierung können wir noch weitere Aggregatfunktionen auf die Felder einer Gruppierung anwenden:

- **Summe:** Berechnet die Summe aller Werte der Gruppierung.
- **Mittelwert:** Berechnet den Mittelwert aller Werte der Gruppierung.
- **Min:** Liefert den kleinsten Wert aller Werte der Gruppierung.
- **Max:** Liefert den größten Wert aller Werte der Gruppierung.
- **Anzahl:** Liefert die Anzahl aller Werte der Gruppierung.
- **StAbw.:** Ermittelt die Standardabweichung aller Werte der Gruppierung.
- **Varianz:** Ermittelt die Varianz aller Werte der Gruppierung.
- **ErsterWert:** Holt den Wert aus dem ersten Datensatz der Gruppierung.
- **LetzterWert:** Holt den Wert aus dem letzten Datensatz der Gruppierung.

### Spaltenüberschrift anpassen

Access erstellt für Spalten, in denen wir eine der oben genannten Aggregatfunktionen verwenden, eine individuelle und von dem Feldnamen abweichende Spaltenüberschrift. Diese setzt sich in der deutschen Version aus der Aggregatfunktion, **von** und Feldname zusammen, also zum Beispiel **AnzahlvonSpendeID**.

Das wollen wir direkt im Abfrageentwurf ändern, damit diese Änderung in darauf aufbauenden Formularen und Berichten direkt übernommen werden kann. Dazu aktivieren wir die Anzeige des Eigenschaftensblatts und markieren das Feld, dessen Eigenschaften wir manipulieren wollen. Für die Eigenschaft **Beschriftung** tragen wir nun die gewünschte Spalten-

Bild 10: Anpassen der Spaltenüberschrift für eine Spalte mit Aggregatfunktion

# ACCESS

## BASICS

Schade! Du hast die letzte Seite der Leseprobe dieses Artikels erreicht. Wenn Du den Rest lesen möchtest oder sogar alle über 400 Artikel, die bisher in Access [basics] erschienen sind, hol Dir doch das Jahresabonnement!

[Zum Jahresabo ... hier klicken](#)

## Abfragen [basics]: 1:n-Daten in Haupt- und Unterformular

Fast jede Anwendung mit mindestens einer Beziehung zwischen zwei Tabellen verlangt nach der Darstellung der Daten dieser beiden Tabellen in Haupt- und Unterformular. Diese Darstellung lässt sich mit kaum einem System einfacher realisieren als mit Microsoft Access. Wie das gelingt, zeigen wir in diesem Artikel. Dabei greifen wir auf eine sehr einfache Konstellation zurück, die sich umgekehrt auch als mit einem einfachen Formular samt Kombinationsfeld abbilden lässt: Die Verwaltung von Kategorien und den enthaltenen Elementen, in diesem Fall von Büchern. Damit greifen wir einen Teil des Datenmodells unseres Artikels **Formulare [basics]: Beispiel Bücherverwaltung** ([www.access-basics.de/624](http://www.access-basics.de/624)) auf.

### Beispieldatenbank

Die Beispiele dieses Artikels findest Du in der Datenbank **FormularBasics\_1nBeziehungenMitUnterformular.accdb**.

### Beispieltabellen

Konkret beziehen wir uns in diesem Artikel auf die beiden Tabellen **tblBuecher** und **tblKategorien**. Diese sind über eine 1:n-Beziehung derart verknüpft, dass wir über das Fremdschlüsselfeld **KategorieID** der Tabelle **tblBuecher** für jedes Buch einen Datensatz der Tabelle **tblKategorien** zuweisen können (siehe Bild 1).

Da die Beziehung zwischen den beiden Tabellen mit referenzieller Integrität definiert wurde, ist sichergestellt, dass wir nur Werte für dieses Fremdschlüsselfeld auswählen können, die im Feld **KategorieID** der Tabelle **tblKategorien** enthalten sind – oder, im Standardzustand eines solchen Feldes, den Wert **Null**.

Das kann später einmal zu einem Problem werden – wie das genau aussieht, schauen wir uns weiter unten an.

### Aufteilung in Hauptformular und Unterformular

Die Aufteilung in Haupt- und Unterformular ist nicht für jeden Access-Benutzer offensichtlich und notwendig. Immerhin könnte man auch einfach die Kategoriedaten und die Buchdaten in einer Abfrage zusammenstellen und diese in einem einzigen For-

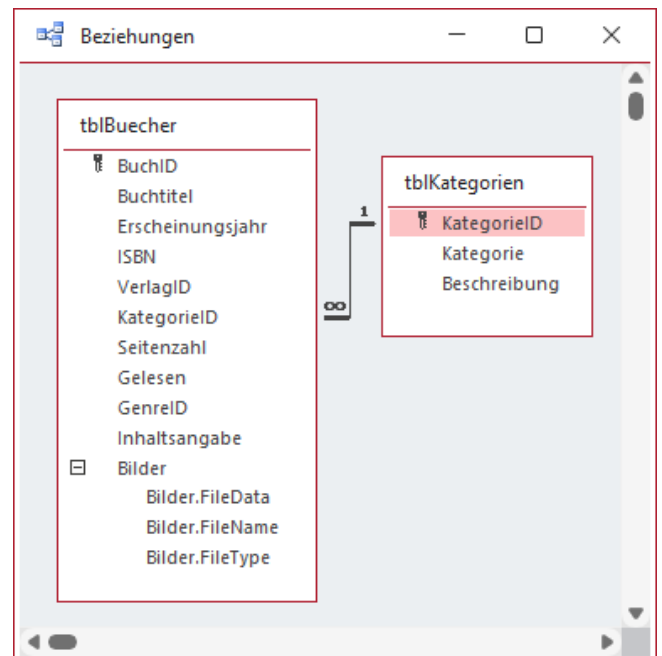


Bild 1: Abzubildende Tabellen

mular anzeigen. Es gibt nichts, was ich noch nicht gesehen hätte ... Jedoch hat die Aufteilung in Haupt- und Unterformular entscheidende Vorteile.

### Reihenfolge beim Erstellen der Formulare

In meiner Praxis hat es sich als günstig erwiesen, zunächst das Unterformular in der Datenblattansicht zu erstellen, dieses zu speichern und es dann nach der Erstellung des Hauptformulars in seinen Entwurf hinzuziehen. Wir starten also mit dem Unterformular.

### Erstellen des Unterformulars in der Datenblattansicht

Wir erstellen das Unterformular wie ein gewöhnliches Formular. Dieses Formular wird erst durch das Einfügen in ein anderes Formular zu einem Unterformular. Zum Erstellen rufen wir den Befehl **Formularentwurf** der Gruppe **Formular** unter dem Ribbontab **Erstellen** auf (siehe Bild 2).

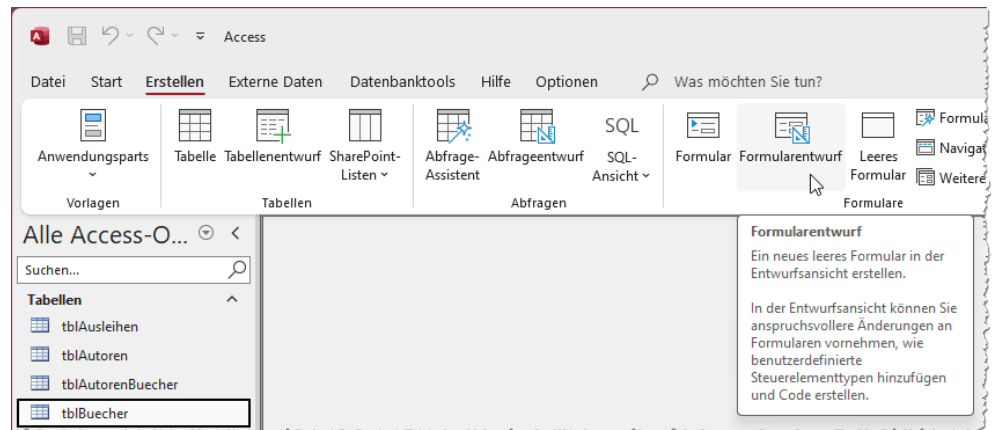


Bild 2: Erstellen eines Formulars in der Entwurfsansicht

Für das nun angezeigte Formular in der Entwurfsansicht blenden wir nun, soweit noch nicht sichtbar, das **Eigenschaftenblatt** ein (F4).

Das Formular soll die Daten der Tabelle **tblBuecher** anzeigen, also wählen wir diese Tabelle für die Eigenschaft **Datensatzquelle** aus (siehe Bild 3).

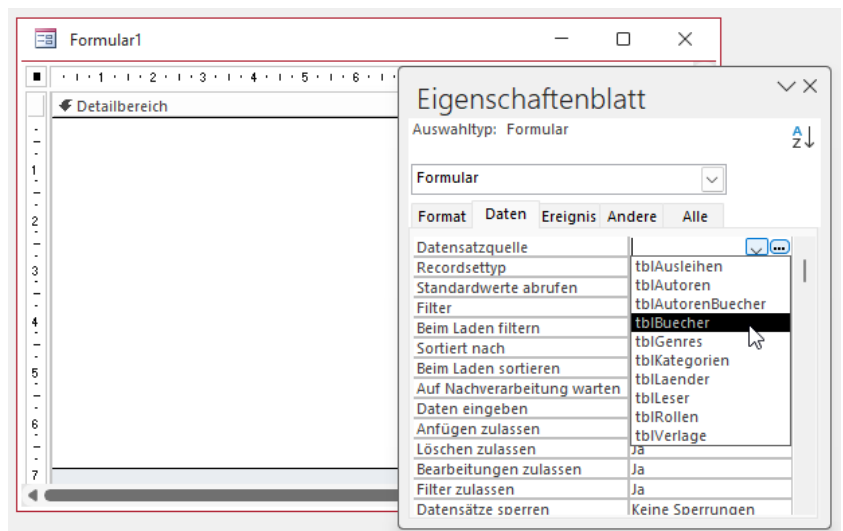


Bild 3: Auswählen der Datensatzquelle

Nun wollen wir die Felder dieser Tabelle in Form von Steuerelementen zum zukünftigen Unterformular hinzufügen. Dazu blenden wir mit dem Ribbon-Befehl **Formularentwurf|Tools|Vorhandene Felder hinzufügen** die Feldliste ein. Diese zeigt bereits alle Felder der als Datensatzquelle ausgewählten Tabelle an. Wir können bei gedrückter **Strg**-Taste alle Felder markieren, die wir zum Formular hinzufügen wollen. Unsere Auswahl umfasst nicht alle Felder (siehe Bild 4).

Das Feld **BuchID** ist allein zum Herstellen der Verknüpfung und als ein-

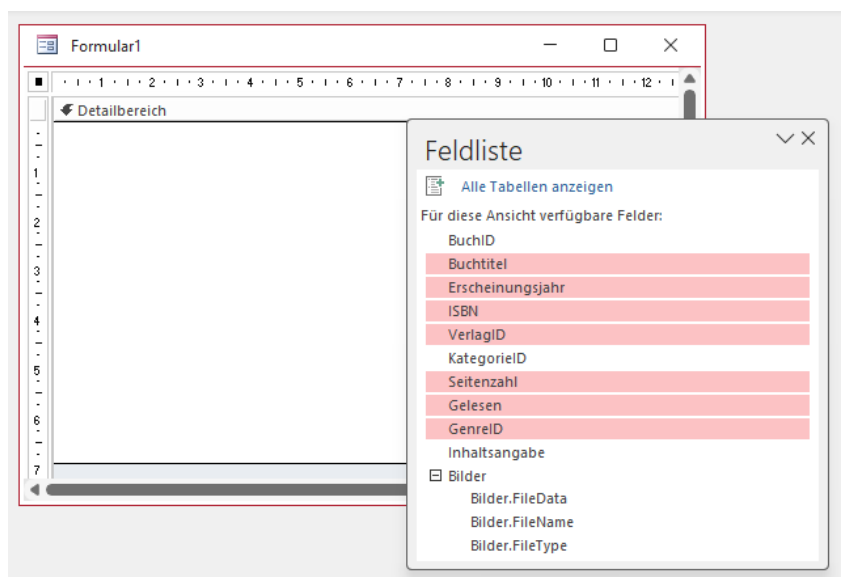


Bild 4: Arbeiten mit der Feldliste

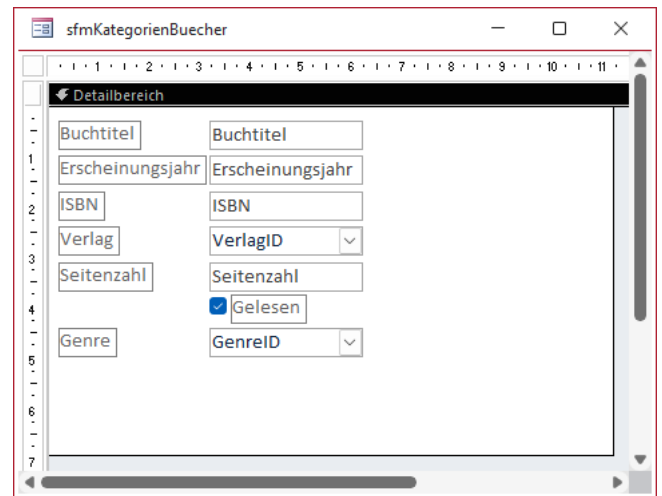
deutiges Merkmal eines jeden Datensatzes nötig, wir brauchen es dem Benutzer nicht zu zeigen. Das Feld **KategorieID** benötigen wir nicht, denn das Unterformular soll alle Bücher zu der im Hauptformular angezeigten Kategorie anzeigen – daher wissen wir schon, zu welcher Kategorie die Datensätze im Unterformular gehören. Das Feld **Inhaltsangabe** ist zu umfangreich, um in der Datenblattansicht angezeigt werden zu können. Und Bilder aus Anlagefeldern können wir in der Datenblattansicht ohnehin nicht anzeigen.

Wenn wir alle Felder ausgewählt haben, können wir diese auf einmal per Drag and Drop in den Detailbereich des Formularentwurfs hineinziehen. Das Ergebnis sehen wir in Bild 5. Hier sehen wir, dass Access gleich die Steuerelementtypen entsprechend dem Entwurf der Tabelle verwendet. Einfache Textfelder werden durch das Steuerelement **Textfeld** abgebildet, Nachschlagfelder durch Kombinationsfelder und **Ja/Nein**-Felder durch Kontrollkästchen.

Das Formular haben wir zwischenzeitlich unter dem Namen **sfmKategorienBuecher** gespeichert. Dies hat sich für uns als gute Praxis erwiesen: Als Unterformulare zu verwendende Formulare erhalten das Präfix **sfm** und der Rest des Namens besteht aus den Bezeichnungen der Elemente im Haupt- und im Unterformular in der Mehrzahl.

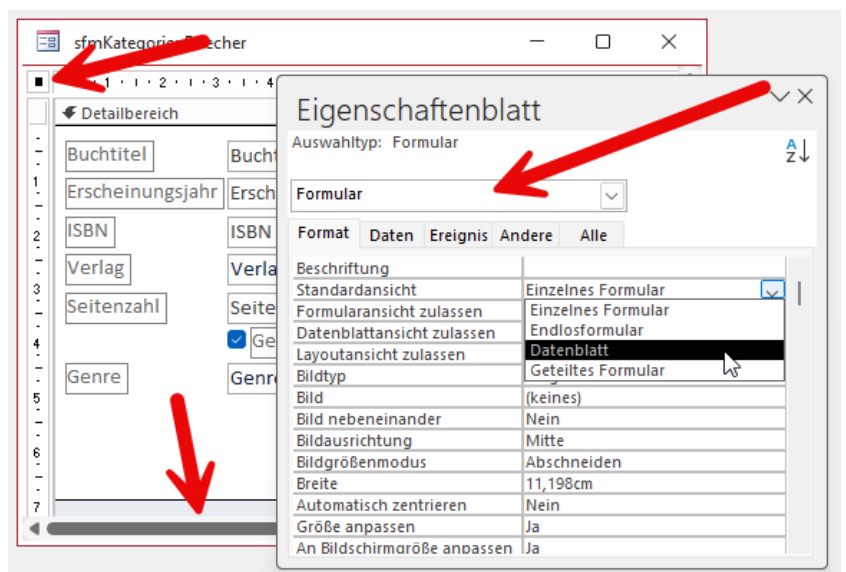
### Aktivieren der Datenblattansicht

Damit das Unterformular seine Daten in der Datenblattansicht anzeigt, müssen wir die Eigenschaft **Standardansicht** auf **Datenblatt** einstellen. Dazu müssen wir diese erst einmal im Eigenschaftensblatt vorfinden. Es handelt sich bei dieser Eigenschaft um eine Eigenschaft des Formulars selbst und nicht etwa eines der Formularbereiche oder Steuerelemente. Deshalb müssen wir zuerst das Formular markieren. Das erledigen wir auf eine der folgenden Methoden:



**Bild 5:** Hineinziehen der Felder in den Entwurf des Formulars

- Wenn das Lineal angezeigt wird: Klick auf den Schnittpunkt der Lineale oder
- Klick auf den grauen Bereich rechts neben oder unter dem weißen Formularhintergrund oder
- Auswahl des Eintrags **Formular** aus dem Kombinationsfeld oben im Eigenschaftensblatt.



**Bild 6:** Einstellen der Ansicht auf die Datenblattansicht

Buchtitel	Erscheinungsjahr	ISBN	Verlag	Seitenzahl	Gelesen	Genre
Anwendungen entwickeln mit Access	2012	978-3-944216-00-3	André Minhorst Verlag	528	<input type="checkbox"/>	Sachbuch
Access und SQL Server	2013	978-3-944216-01-0	André Minhorst Verlag	528	<input type="checkbox"/>	Sachbuch
Access und Office	2014	978-3-944216-02-7	André Minhorst Verlag	528	<input type="checkbox"/>	Sachbuch
*	0			0	<input type="checkbox"/>	

Bild 7: Test des Unterformulars in der Datenblattansicht

Alle drei Möglichkeiten sind in Bild 6 markiert. Nun können wir im Eigenschaftenblatt wie in der Abbildung im Bereich **Format** die Eigenschaft **Standardansicht** auf **Datenblatt** einstellen.

### Test des Unterformulars

Bevor wir uns an die Gestaltung des Hauptformulars begeben, testen wir das Unterformular. Dazu schließen wir es und speichern es gegebenenfalls und öffnen es dann erneut. Mit ein paar Beispieldatensätzen sieht das Formular nun wie in Bild 7 aus. Es sieht genauso aus wie eine Tabelle oder Abfrage in der Datenblattansicht und enthält auch alle der komfortablen Merkmale wie beispielsweise die Möglichkeiten zum Filtern oder Sortieren.

Wie wir hier sehen, spielt es auch keine Rolle, wie wir die Felder im Entwurf arrangieren – am Ende werden diese ohnehin wie im Datenblatt angezeigt. Die einzige Auswirkung ist die Reihenfolge, die beim Hinzufügen einfach von oben nach unten zugewiesen wird.

Damit können wir das Formular **sfrmKategorienBuecher** wieder schließen.

### Hauptformular erstellen

Auf die gleiche Weise wie das Unterformular erstellen wir nun das Hauptformular, also mit dem Ribbonbefehl **Erstellen|Formular-**

**re|Formularentwurf**. Danach wählen wir für die Eigenschaft **Datensatzquelle** die Tabelle **tblKategorien** aus.

Anschließend wechseln wir zur Feldliste und markieren die beiden Felder **Kategorie** und **Beschreibung** und ziehen diese wie in Bild 8 in den Detailbereich des Formularentwurfs. Anschließend ergänzen wir noch die Doppelpunkte in den Beschriftungsfeldern.

### Unterformular in das Hauptformular einfügen

Nun kommt der entscheidende Schritt: Wir fügen das Unterformular in das Hauptformular ein. Das geht leichter als man denkt, denn wir brauchen nur folgende Schritte zu erledigen:

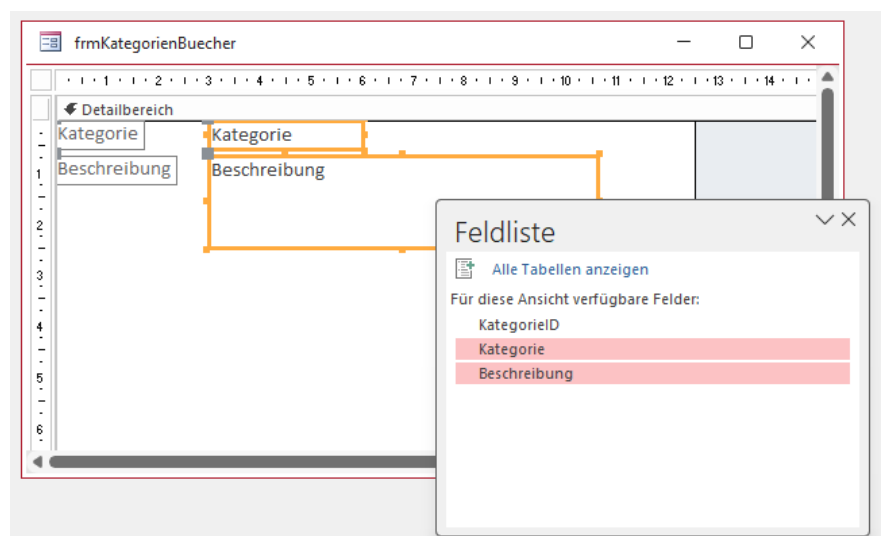


Bild 8: Hinzufügen der Felder zum Hauptformular



# ACCESS

## BASICS

Schade! Du hast die letzte Seite der Leseprobe dieses Artikels erreicht. Wenn Du den Rest lesen möchtest oder sogar alle über 400 Artikel, die bisher in Access [basics] erschienen sind, hol Dir doch das Jahresabonnement!

[Zum Jahresabo ... hier klicken](#)