

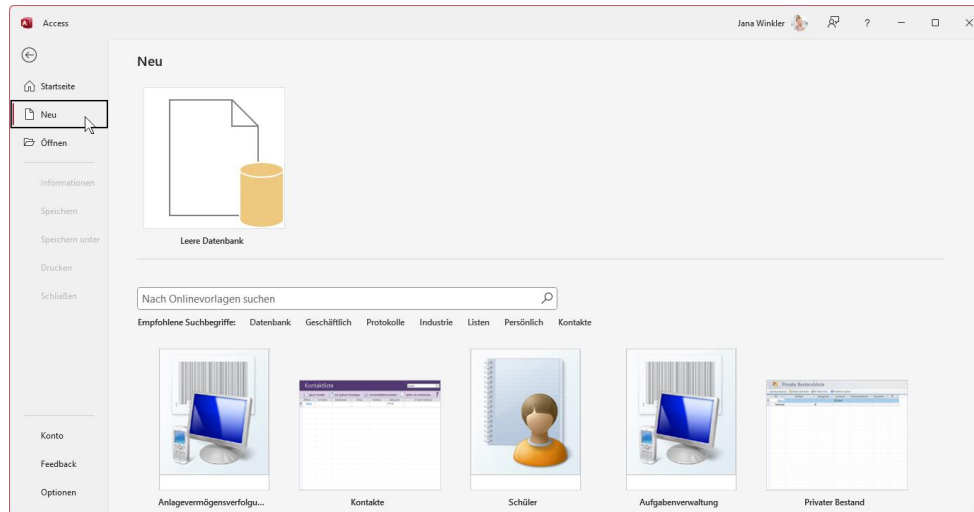
**Unsere Schulungsunterlagen  
bieten professionelle Hilfestellungen  
im Umgang mit Access 2021, wahlweise  
als Ausdruck nutzbar oder im Intranet**

Aus technischen Gründen wurde in dieser  
Musterunterlage dieses Deckblatt zusätzlich eingefügt,  
und anders als im Original-Worddokument haben wir  
das Inhaltsverzeichnis am Ende platziert.  
Darüber hinaus entsprechen hier auch die Kopf- und  
Fußzeilen nicht dem Original.

## 2 Der Datenbankentwurf

Ein Datenbankentwurf ist immer dann sinnvoll, wenn die Daten aus unterschiedlichen Bereichen stammen und doch in Beziehung zueinander stehen. Dabei ist auch ein Ziel, die Fehler so früh wie möglich zu minimieren, da bei jedem nachfolgenden Schritt im Datenbankentwurf meistens höhere Kosten anfallen, um einen Fehler zu beheben!

Beim Erstellen einer neuen Datenbank können Sie auch auf Muster-Datenbanken zurückgreifen. Dazu wählen Sie auf der Seite **Neu** eine **Vorlage** aus:



Eine Vorlage auswählen

### 2.1 Schritte beim Entwurf einer Datenbank

Der Entwurf einer Datenbank erfordert verschiedene Schritte:

**Informationsbedarfsanalyse:** Die Anforderungen an das zu realisierende System erheben.

**Konzeptueller Entwurf:** Enger Bezug zur realen Welt, statt zum Computer. Die später zu entwerfenden Datenstrukturen spielen noch keine Rolle: formale Beschreibung der Daten.

**Logischer Entwurf:** logische Strukturen auf dem Papier. Das Ergebnis kann später im Fenster **Beziehungen** betrachtet werden (z.B. Seite 27).

**Physischer Entwurf:** Implementierung der Hardware, des Betriebssystems und der Software, Erstellung der Datenbank und Eingabe der Daten. Danach schließt sich die Nutzungsphase an.

**Nutzungsphase:** Die Anwender arbeiten mit der Datenbank, um die erforderlichen Informationen zu bekommen. Darüber hinaus sind laufend die Daten zu pflegen und zu sichern und es müssen Hard- und Software gewartet werden.

### 2.2 Tabellen

#### Vorüberlegungen für eine neue Tabelle

Sowohl bei Papier-Karteikarten als auch bei Computer-Tabellen bedeutet eine nachträgliche Änderung der Karteikarten bzw. Tabellenstrukturen eventuell einen erheblichen Aufwand. Manchmal muss bei einer Änderung auch der ganze bestehende Informationsbestand angepasst werden. Sie sollten sich gut überlegen, wie die Struktur einer neuen Tabelle aussehen soll. Nehmen Sie sich dafür Zeit. Es zahlt sich mehrfach wieder aus.

## Schulungsunterlage Access 2021 Weiterführung

Stellen Sie sich die folgenden Fragen:

- Welche Informationseinheiten brauche ich?  
Machen Sie sich eine Liste mit allen Informationen, die Sie in der Datenbank ablegen wollen. Sie können in diesem Schritt durchaus schon Feldnamen zuordnen. Denken Sie dabei ruhig an den Karteikasten mit dem vorgedruckten Formular. Bei einem Adressverzeichnis könnten die Informationseinheiten z.B. Name, Vorname, Anrede, Titel, Straße, Postleitzahl (PLZ), Ort, Telefonnummer und Land heißen.
- Auf welche Datenfelder will ich später zugreifen?  
Markieren Sie auf der Liste alle Informationseinheiten, nach denen Sie später einmal suchen oder über die Sie Auswertungen anfertigen wollen. Bei dem gewählten Beispiel dürften Name, PLZ und Ort interessante Felder für eine Suche sein.
- Nach welchen Kriterien sollen die Informationen sortiert werden?  
Prüfen Sie, nach welchen Feldern Sie Ihre Tabelle später sortieren wollen. Sortieren könnten Sie ein Adressbuch ebenfalls nach Name, PLZ, Ort und Land.
- Von welchem Typ sind die jeweiligen Informationen?  
Ordnen Sie jedem Feld einen Felddatentyp zu (Text, Zahl usw.) Beispielsweise ist es in einem Adressbuch sicher richtig, zunächst allen Feldern den Typ **Text** zuzuordnen.

Wenn Sie auf den Nachnamen zugreifen wollen, ist es sinnvoll, für Vor- und Nachnamen zwei getrennte Datenfelder vorzusehen. Dagegen kann der Straßenname mit der Hausnummer in einem Feld abgespeichert werden. Postleitzahl und Ort sollten wiederum in getrennten Feldern gespeichert werden.

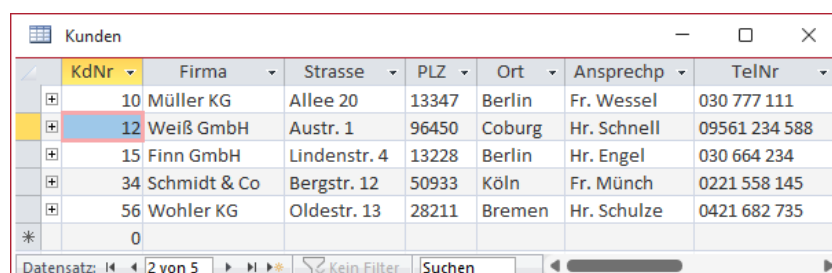
Legen Sie in Zweifelsfällen die Informationen in getrennten Datenfeldern ab! Eine Datenbank kann später aber auch geändert werden. Allerdings ist es manchmal sehr mühsam, dann die Informationen richtig in die jeweiligen Felder einzuordnen oder sie aus anderen Feldern zu isolieren.

Bei einer Datenbank-Struktur sind die Spaltenüberschriften einer Tabelle bekannt. Es werden verschiedene Informationen und die Beziehungen dieser Informationen in einer Menge dargestellt. Das nachfolgende Bild zeigt die Informationen über die Kunden eines Unternehmens an. Zu diesen Informationen gehören u.a. die Kundennummer, der Firmenname, der Ort und die Straße.

Jede Zeile enthält Informationen über einen Kunden. In der Datenbanksprache ist eine Zeile ein Datensatz.

Zeile = Datensatz

Spalte = Feldinhalt des entsprechenden Feldnamens  
(z.B. stehen alle Kundennummern in der ersten Spalte, der Feldname lautet **KdNr**)



| KdNr | Firma        | Strasse      | PLZ   | Ort    | Ansprechp   | TelNr         |
|------|--------------|--------------|-------|--------|-------------|---------------|
| 10   | Müller KG    | Allee 20     | 13347 | Berlin | Fr. Wessel  | 030 777 111   |
| 12   | Weiß GmbH    | Austr. 1     | 96450 | Coburg | Hr. Schnell | 09561 234 588 |
| 15   | Finn GmbH    | Lindenstr. 4 | 13228 | Berlin | Hr. Engel   | 030 664 234   |
| 34   | Schmidt & Co | Bergstr. 12  | 50933 | Köln   | Fr. Münch   | 0221 558 145  |
| 56   | Wohler KG    | Oldestr. 13  | 28211 | Bremen | Hr. Schulze | 0421 682 735  |
| 0    |              |              |       |        |             |               |

Elemente einer Tabelle

## 2.3 Primär- und Sekundärschlüssel

### Primärschlüssel

Um Datensätze eindeutig identifizieren zu können, benötigen diese ein Merkmal, das in einer Tabelle nicht doppelt vorkommt. In der untenstehenden Tabelle **Kunden** könnten sowohl der Firmenname als auch der Ort durchaus mehrmals auftreten. Die Kundennummer hingegen ist eindeutig, d.h. es gibt nicht mehrere Kunden mit der gleichen Kundennummer. Aufgrund dieser Kundennummer kann also der Kunde eindeutig aus der Tabelle herausgefunden werden. Das Feld **KdNr** wird deshalb als Primärschlüssel definiert.

#### Merkmale eines Feldes mit Primärschlüssel:

- Der Inhalt des Feldes ist eindeutig.
- In jedem Datensatz müssen Daten stehen.
- Ein Primärschlüssel kann sich auch aus mehreren Feldern zusammensetzen (z.B. Ort und PLZ).

#### Primärschlüssel definieren



Schaltfläche **Primärschlüssel**



Schaltfläche **Entwurfsansicht**

Die Definition eines Primärschlüssels wird in der **Entwurfsansicht** über die Schaltfläche **Primärschlüssel** vorgenommen:

In der Entwurfsansicht  
definierter Primärschlüssel

| Kunden   |              |
|----------|--------------|
| Feldname | Felddatentyp |
| KdNr     | Zahl         |
| Firma    | Kurzer Text  |
| Strasse  | Kurzer Text  |
| PLZ      | Kurzer Text  |
| Ort      | Kurzer Text  |

| Kunden |      |              |              |       |        |             |               |
|--------|------|--------------|--------------|-------|--------|-------------|---------------|
|        | KdNr | Firma        | Strasse      | PLZ   | Ort    | Ansprechp   | TelNr         |
|        | 10   | Müller KG    | Allee 20     | 13347 | Berlin | Fr. Wessel  | 030 777 111   |
|        | 12   | Weiß GmbH    | Austr. 1     | 96450 | Coburg | Hr. Schnell | 09561 234 588 |
|        | 15   | Finn GmbH    | Lindenstr. 4 | 13228 | Berlin | Hr. Engel   | 030 664 234   |
|        | 34   | Schmidt & Co | Bergstr. 12  | 50933 | Köln   | Fr. Münch   | 0221 558 145  |
|        | 56   | Wohler KG    | Oldestr. 13  | 28211 | Bremen | Hr. Schulze | 0421 682 735  |
| *      | 0    |              |              |       |        |             |               |

Tabelle mit definiertem Primärschlüssel

Primärschlüssel werden häufig auch **Identifikationsschlüssel** genannt.

### Sekundärschlüssel bzw. Index

In Access wird der **Sekundärschlüssel** als **Index** bezeichnet. Indizes werden definiert, um die Datensortierung zu beschleunigen. Wenn Sie beim Entwurf der Tabelle bereits wissen, nach welchen Spalten bzw. Feldern sortiert werden soll, so können Sie diese Felder mit der rechtsstehenden Schaltfläche indizieren.



## Merkmale von Indizes:

- Auf jede Spalte der Tabelle kann ein Index gesetzt werden, außer auf die Spalte, die als Primärschlüssel definiert ist.
- Indizes können über mehrere Spalten gesetzt werden.
- Sekundärschlüssel können mehrfach vorkommen.
- In einer Tabelle können beliebig viele Sekundärschlüssel definiert werden.

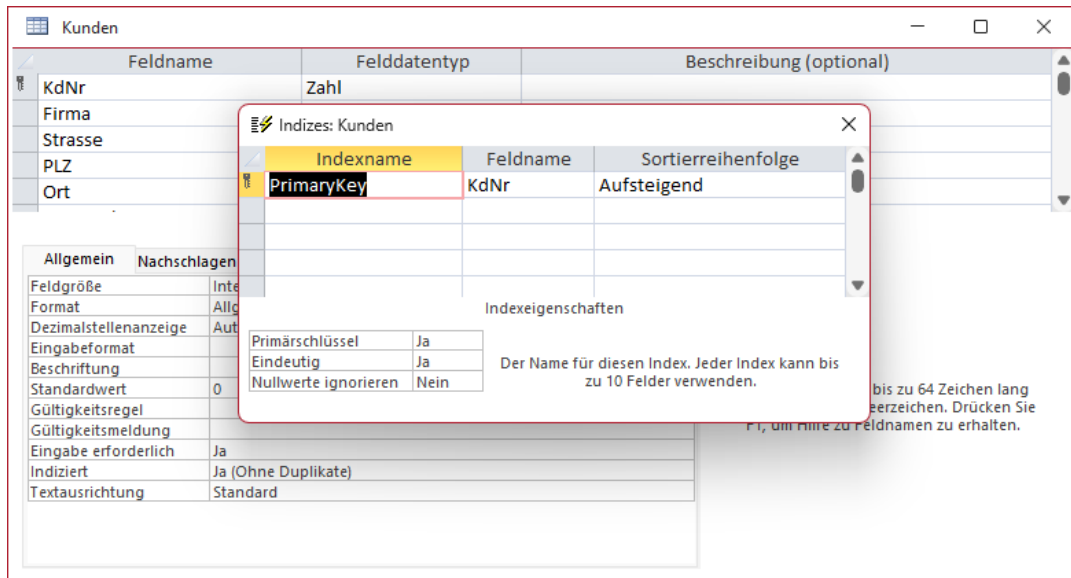


Tabelle in der Entwurfsansicht mit eingeblendetem Indexfenster

## 2.4 Beispiel: Datenbank Auftrag

Angenommen, Sie speichern alle Kunden-, Bestell- und Artikeldaten in einer einzigen Tabelle, so müssen Sie bei Bestellungen von unterschiedlichen Kunden jeweils alle Bestell- und Artikeldaten nochmals erfassen. Bestellt ein Kunde mehrmals, so wiederholt sich die Eingabe aller kunden-spezifischen Daten.

In der folgenden Abbildung wiederholen sich die Bestellungen der Artikel mit den Artikelnummern 5534 und 6643.

| Kunden |              |                              |             |               |           |       |             |       |
|--------|--------------|------------------------------|-------------|---------------|-----------|-------|-------------|-------|
| KdNr   | Firma        | Adresse                      | Ansprechp   | TelNr         | BestellNr | ArtNr | ArtName     | Menge |
| 10     | Müller KG    | Darwinstr. 20, 01109 Dresden | Fr. Wessel  | 0351 777 111  | 132       | 5534  | Damenhose   | 300   |
| 12     | Weiß GmbH    | Austr. 1, 96450 Coburg       | Hr. Schnell | 09561 234 588 | 55        | 6643  | Sweatshirt  | 170   |
| 15     | Finn GmbH    | Lindenstr. 4, 13228 Berlin   | Hr. Engel   | 030 664 234   | 67        | 3367  | Damenschuhe | 500   |
| 34     | Schmidt & Co | Bergstr. 12, 50933 Köln      | Fr. Münch   | 0221 558 145  | 34        | 6643  | Sweatshirt  | 200   |
| 56     | Wohler KG    | Oldestr. 13, 28211 Bremen    | Hr. Schulze | 0421 682 735  | 33        | 5534  | Damenhose   | 250   |
| *      |              |                              |             |               |           |       |             |       |

Tabelle Kunden mit allen Kunden- und Bestelldaten

Diese Tabelle ist aus den oben genannten Gründen wenig sinnvoll.

### Probleme bei der Datenbearbeitung dieser Tabelle:

- Falls Sie eine Bestellung löschen, dann entfernen Sie gleichzeitig die dazugehörigen Firmendaten. Hat ein Kunde keine weiteren Bestellungen getätigt, so gehen die Firmendaten verloren.

- Ändert sich der Artikelname (z.B. Damenschuhe in Damenstiefel), so müssen alle Datensätze geändert werden, in denen dieser Artikel vorkommt.

Ziel eines optimalen Datenbankentwurfs ist es, die Tabellen so zu strukturieren, dass sich Informationen in Tabellen nicht wiederholen und trotzdem alle wichtigen Informationen aus den Tabellen gelesen werden können. Diesen Prozess bezeichnet man als

## Normalisierung.

Eine Normalisierung wird durch eine Datenerfassung in mehreren getrennten Tabellen und das sinnvolle Verbinden dieser Tabellen erreicht.

## 2.5 Normalisierung

Durch die Normalisierung soll erreicht werden, dass keine Datenredundanz auftritt und trotzdem alle Informationen aus den Tabellen abrufbar sind. Um dieses Ziel zu erreichen, wird die Normalisierung *stufenweise* durchgeführt.

### Die 1. Normalform



Eine Tabelle befindet sich nur dann in der 1. Normalform, wenn in jedem Feld nur eine einzige Information steht.

In unserem Beispiel ist die vorher zu sehende Tabelle **Kunden** nicht in der 1. Normalform, da das Feld **Adresse** mehrere Informationen aufweist, nämlich die Informationen Strasse, Postleitzahl und Ort.

Sie bringen diese Tabelle in die 1. Normalform, indem Sie das Feld **Adresse** in die Felder **Strasse\***, **PLZ** und **Ort** aufteilen. Nach dieser Normalisierung ist jedem Feld nur jeweils eine Information zugeordnet.

| Kunden |              |              |       |        |             |               |           |       |             |       |
|--------|--------------|--------------|-------|--------|-------------|---------------|-----------|-------|-------------|-------|
| KdNr   | Firma        | Strasse      | PLZ   | Ort    | Ansprechp   | TeleNr        | BestellNr | ArtNr | ArtName     | Menge |
| 10     | Müller KG    | Allee 20     | 13347 | Berlin | Fr. Wessel  | 030 777 111   | 132       | 5534  | Damenhose   | 300   |
| 12     | Weiß GmbH    | Austr. 1     | 96450 | Coburg | Hr. Schnell | 09561 234 588 | 55        | 6643  | Sweatshirt  | 170   |
| 15     | Finn GmbH    | Lindenstr. 4 | 13228 | Berlin | Hr. Engel   | 030 664 234   | 67        | 3367  | Damenschuhe | 500   |
| 34     | Schmidt & Co | Bergstr. 12  | 50933 | Köln   | Fr. Münch   | 0221 558 145  | 34        | 6643  | Sweatshirt  | 200   |
| 56     | Wohler KG    | Oldestr. 13  | 28211 | Bremen | Hr. Schulze | 0421 682 735  | 33        | 5534  | Damenhose   | 250   |
| *      |              |              |       |        |             |               |           | 0     | 0           | 0     |

Tabelle Kunden in der 1. Normalform

### Problem der Tabelle Kunden in der 1. Normalform:

Die Tabelle enthält Redundanzen. Bestellt ein Kunde mehrere Artikel, so werden in *jedem* Datensatz alle Kundendaten gespeichert.

### Die 2. Normalform

Die Tabelle in der 1. Normalform enthält nun in jedem Feld eine einzige Information. Belassen Sie aber diese Tabelle für Ihre Datenbank so wie oben dargestellt, so können so genannte **Anomalien** bzw. **Dateninkonsistenzen** auftreten. Ändern Sie z.B. nur im ersten Datensatz für den Artikelnamen **Damenhose** die dazugehörige Artikelnummer und vergessen diese Änderung im

\* Bei einem Datenaustausch kann es nachteilig sein, wenn in den Feldnamen nationale Sonderzeichen (ä ü ö ß, hier bei Straße) verwendet werden.

fünften Datensatz, so treten Unstimmigkeiten auf, da dem Artikelnamen **Damenhose** unterschiedliche Artikelnummern zugeordnet werden.

Dieses Problem tritt deswegen auf, weil die Tabelle gleichzeitig verschiedenartige Informationen (Kundendaten, Bestelldaten, Artikeldaten) enthält, diese aber unabhängig voneinander geändert werden können.

Im Einführungskurs haben Sie gesehen, dass es sinnvoll ist, jede Tabelle mit einem Primärschlüssel zu versehen, um einen Datensatz eindeutig zu identifizieren. Ein Primärschlüssel besteht aus einer Spalte oder aus einer Kombination mehrerer Spalten. Der Inhalt des Primärschlüssels darf in der Tabelle nur ein einziges Mal vorkommen.

### Kennzeichen eines Primärschlüssels:

- Der Inhalt des Feldes ist eindeutig.
- In jedem Datensatz muss ein Primärschlüssel zu finden sein.



Um eine Tabelle in die 2. Normalform bringen zu können, muss jedes Feld dieser Tabelle voll von einem Primärschlüssel abhängig sein.

In der Tabelle **Kunden** wurde in der 1. Normalform der Primärschlüssel auf die Kundennummer gesetzt, weil diese eindeutig ist und somit einen Kunden klar identifiziert. Da die Artikel- und Bestelldaten jedoch mit dem Primärschlüssel **KdNr** nichts zu tun haben, müssen diese Daten aus der Tabelle herausgelöst und in eine eigene Tabelle gebracht werden, die ebenfalls die Kriterien für die 1. und 2. Normalform erfüllen.

Die Tabelle **Kunden** wird nun durch einen Normalisierungsprozess in die Tabellen **Kunden**, **Bestellungen** und **Artikel** aufgeteilt. Diese Tabellen enthalten danach keine Redundanzen mehr, d.h. es gibt keine Daten, die in mehreren Tabellen gleichzeitig gespeichert werden. In diesen Tabellen sind die Felder jeweils vollständig vom jeweiligen Primärschlüssel abhängig.

In den folgenden drei Abbildungen sehen Sie die Tabellen in der 2. Normalform mit den Primärschlüsseln **KdNr**, **BestellNr** und **ArtNr**.

| KdNr | Firma        | Strasse         | PLZ   | Ort     | Ansprechp   | TelNr         |
|------|--------------|-----------------|-------|---------|-------------|---------------|
| 10   | Müller KG    | Darwinstr. 20   | 01109 | Dresden | Fr. Wessel  | 0351 777 111  |
| 12   | Weiß GmbH    | Austr. 1        | 96450 | Coburg  | Hr. Schnell | 09561 234 588 |
| 34   | Schmidt & Co | Bergstr. 12     | 50933 | Köln    | Fr. Münch   | 0221 558 145  |
| 56   | Wohler KG    | Oldestr. 13     | 28211 | Bremen  | Hr. Schulze | 0421 682 735  |
| 63   | Keller OHG   | Albstr. 24      | 89081 | Ulm     | Fr. Haber   | 0731 248 345  |
| 74   | Brick GmbH   | Kolumbusring 48 | 18106 | Rostock | Hr. Berger  | 0381 122 935  |

Tabelle Kunden in der 2. Normalform mit dem Primärschlüssel KdNr

| BestellNr | KdNr | ArtNr | Menge               | Zahlungsart | Kennz |
|-----------|------|-------|---------------------|-------------|-------|
| 13        | 10   | 3367  | 20 Rechnung         |             | 1     |
| 33        | 56   | 5534  | 250 Kreditkarte     |             | 2     |
| 34        | 34   | 6643  | 200 Kreditkarte     |             | 2     |
| 55        | 12   | 6643  | 170 Electronic Cash |             | 3     |
| 64        | 34   | 3367  | 80 Kreditkarte      |             | 2     |
| 67        | 15   | 3367  | 500 Nachnahme       |             | 4     |
| 132       | 10   | 5534  | 300 Rechnung        |             | 1     |

Tabelle Bestellungen in der 2. Normalform mit dem Primärschlüssel BestellNr



| ArtNr | ArtName     | Preis/St |
|-------|-------------|----------|
| 3367  | Damenschuhe | 89,00 €  |
| 5534  | Damenhose   | 79,00 €  |
| 6643  | Sweatshirt  | 59,90 €  |
| *     | 0           | 0,00 €   |

Tabelle Artikel in der 2. Normalform mit dem Primärschlüssel ArtNr

## Tabellen-Analyse

**Tabelle Kunden:** Diese Tabelle enthält nur kundenspezifische Daten. Alle Kundendaten können über die Kundennummer (KdNr = Primärschlüssel) in Erfahrung gebracht werden.

**Tabelle Bestellungen:** In dieser Tabelle werden lediglich die Bestelldaten gespeichert. Über die Kundennummer kann aus der Tabelle **Kunden** der Kunde ermittelt werden. Eine Bestellung kann über die Bestellnummer (BestellNr = Primärschlüssel) ermittelt werden.

**Tabelle Artikel:** Diese Tabelle enthält die Zuordnung zwischen Artikelnummer, Artikelname und Preis je Stück. Ein Artikel ist über die Artikelnummer (ArtNr = Primärschlüssel) eindeutig identifiziert.

## Die 3. Normalform

In der Tabelle **Bestellungen** kann noch ein Fehler auftreten und zwar deshalb, weil das Feld **Zahlungsart** erscheint, obwohl sich die Zahlungsart aus dem Kennzeichen ergibt. Ändern Sie nun das Kennzeichen, welches der Rechnung zugeordnet ist, auf eine 2, so ist das Kennzeichen nicht mehr eindeutig, da 2 dann sowohl der Rechnung als auch der Kreditkarte zugeordnet ist.

Um solche Fehler zu vermeiden, müssen Sie Ihre Tabellen in die 3. Normalform überführen, die da lautet:



Ein Feld, das nicht vom Primärschlüssel abhängig ist, muss in einer eigenen Tabelle gespeichert werden.

In der Tabelle **Bestellungen** bedeutet dies, dass das Feld **Zahlungsart** überflüssig ist, da die Zahlungsart durch das Kennzeichen eindeutig wird.

In der neu erstellten Tabelle **Zahlungsart** erfolgt die Zuordnung zwischen den Feldern **Kennz** und **Zahlungsart**.

| BestellNr | KdNr | ArtNr | Menge | Kennz |
|-----------|------|-------|-------|-------|
| 13        | 10   | 3367  | 20    | 1     |
| 33        | 56   | 5534  | 250   | 2     |
| 34        | 34   | 6643  | 200   | 2     |
| 55        | 12   | 6643  | 170   | 3     |
| 64        | 34   | 3367  | 80    | 2     |
| 67        | 15   | 3367  | 500   | 4     |
| 132       | 10   | 5534  | 300   | 1     |

Tabelle Bestellungen mit dem Feld Kennz für die Festlegung der Zahlungsart

| Kennz | Zahlungsart     |
|-------|-----------------|
| 1     | Rechnung        |
| 2     | Kreditkarte     |
| 3     | Electronic Cash |
| 4     | Nachname        |
| 5     | Barzahlung      |

Tabelle Zahlungsart als Zuordnungstabelle zwischen Kennzeichen und Zahlungsart

Die gesamte Datenbank mit den Tabellen **Kunden**, **Bestellungen**, **Artikel** und **Zahlungsart** befindet sich nun in der 3. Normalform. Damit treten keine Datenredundanzen auf und die Gefahr von Anomalien ist nicht mehr gegeben.





Die Tabelle **Bestellungen** befindet sich nur dann in der 3. Normalform, wenn bei jeder Bestellung nur ein einziger Artikel bestellt werden kann. Da dies jedoch nicht der Realität entspricht, muss diese Tabelle noch geändert werden.

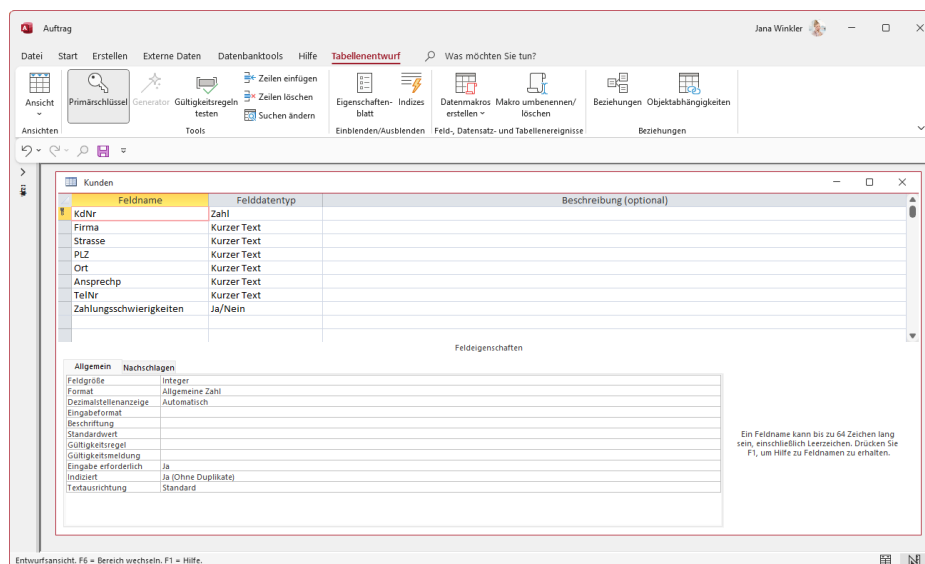
## Normalisierungsgründe:

- Das Datenmodell wird für alle verständlich.
- Dateninkonsistenzen werden vermieden.
- In Datenfeldern werden Nullwerte vermieden.
- Es treten keine Redundanzen auf.
- Normalisierte Modelle sind änderungsfreundlich.

## 2.6 Übung

Erstellen Sie die normalisierte Datenbank **Auftrag.accdb** mit den Tabellen **Kunden**, **Bestellungen**, **Artikel** und **Zahlungsart**.

Das nachfolgende Bild zeigt die Datenstruktur und den Primärschlüssel der Tabelle **Kunden** in der Entwurfsansicht.



Datenstruktur der Tabelle Kunden in der Entwurfsansicht

### Datenstrukturen und Primärschlüssel

Nachfolgend sind die Tabellen mit ihrer Struktur und den dazugehörigen Primärschlüsseln aufgelistet:

#### Tabelle **Kunden**

| Feldname | Felddatentyp  | Feldlänge | Primärschlüssel |
|----------|---------------|-----------|-----------------|
| KdNr     | Zahl; Integer |           | ja              |
| Firma    | Kurzer Text   | 50        |                 |
| Strasse  | Kurzer Text   | 50        |                 |
| PLZ      | Kurzer Text   | 10        |                 |

| Feldname                | Felddatentyp | Feldlänge | Primärschlüssel |
|-------------------------|--------------|-----------|-----------------|
| Ort                     | Kurzer Text  | 50        |                 |
| Ansprechp               | Kurzer Text  | 50        |                 |
| TelNr                   | Kurzer Text  | 20        |                 |
| Zahlungsschwierigkeiten | Ja/Nein      |           |                 |

Tabelle **Bestellungen**

| Feldname  | Felddatentyp  | Feldlänge | Primärschlüssel |
|-----------|---------------|-----------|-----------------|
| BestellNr | Zahl; Integer |           | ja              |
| KdNr      | Zahl; Integer |           |                 |
| ArtNr     | Zahl; Integer |           |                 |
| Menge     | Zahl; Integer |           |                 |
| Kennz     | Zahl; Byte    |           |                 |

Tabelle **Artikel**

| Feldname | Felddatentyp  | Feldlänge | Primärschlüssel |
|----------|---------------|-----------|-----------------|
| ArtNr    | Zahl; Integer |           | ja              |
| ArtName  | Kurzer Text   | 50        |                 |
| Preis/St | Währung       |           |                 |

Tabelle **Zahlungsart**

| Feldname    | Felddatentyp | Feldlänge | Primärschlüssel |
|-------------|--------------|-----------|-----------------|
| Kennz       | Zahl; Byte   |           | ja              |
| Zahlungsart | Kurzer Text  | 50        |                 |

Übertragen Sie die Daten in der Tabellenansicht in die jeweiligen Tabellen bzw. kopieren Sie die Daten aus den bereitgestellten Tabellen (Datenbank **Auftrag.accdb**).

## 2.7 Die Beziehungen zwischen den Tabellen bestimmen

Um aus einer Datenbank mit den Tabellen **Kunden**, **Bestellungen**, **Artikel** und **Zahlungsart** Informationen zu erhalten, die in mehreren Tabellen gespeichert sind, müssen zwischen den Tabellen Beziehungen bestehen. Beziehungen bzw. Relationen legen die Verknüpfungen der Datensätze einer Tabelle mit den Datensätzen einer anderen Tabelle fest.

Welche Beziehungen bestehen nun zwischen den Tabellen und wie tragen Sie diese Beziehungen in Access ein?

### Theoretische Grundlagen

Neben der Datenhaltung und dem Zugriff auf vorhandene Daten gibt es noch Beziehungen der einzelnen Tabellen untereinander.

## Beziehungstypen

In der folgenden Liste finden Sie die Abkürzungen für die unterschiedlichen Zuordnungsarten.

| Beziehungstyp | Zuordnung                            |
|---------------|--------------------------------------|
| <b>1</b>      | Exakt eine Zuordnung                 |
| <b>n</b>      | Keine, eine oder mehrere Zuordnungen |
| <b>m</b>      | Mindestens eine Zuordnung            |
| <b>c</b>      | Keine oder eine Zuordnung            |

**1:1-Beziehung:** Stehen zwei Tabellen in Beziehung zueinander, in denen ein Datensatz der einen Tabelle genau einem Datensatz der anderen Tabelle zugeordnet ist, so spricht man von einer 1:1-Beziehung. Dieser Beziehungstyp ist eher selten anzutreffen.

**1:n-Beziehung:** In einer 1:n-Beziehung ist ein Datensatz der einen Tabelle keinem bzw. einem Datensatz oder mehreren Datensätzen der anderen Tabelle zugeordnet.

**n:m-Beziehung:** Beliebige viele Datensätze der einen Tabelle sind beliebig vielen Datensätzen der anderen Tabelle zugeordnet.

## Beispiele:

| Tabelle 1   | Tabelle 2   | Beziehungstyp | Zuordnung   |
|-------------|-------------|---------------|---|
| Leser       | Bücher      | 1:n           | Ein Leser entleiht kein Buch, ein Buch oder mehrere Bücher.                                   |
| Frauen      | Männer      | c:c           | Heirat: Frauen und Männer sind entweder verheiratet oder nicht.                               |
| Mitarbeiter | Abteilungen | m:1           | Viele Mitarbeiter sind einer Abteilung zugeordnet.  |
| Kinder      | Ehepaare    | n:1           | Ehepaare haben kein Kind, ein Kind oder mehrere Kinder.                                       |
| Personen    | Personen    | n:m           | Freundschaften: Personen pflegen keine, eine oder mehrere Freundschaften zu anderen Personen. |

**In einer normalisierten Datenbank gibt es nur noch 1:n-Beziehungen.**

Zum besseren Überblick beim Datenbank-Entwurf ist es sinnvoll, Beziehungen zwischen den Tabellen grafisch darzustellen.

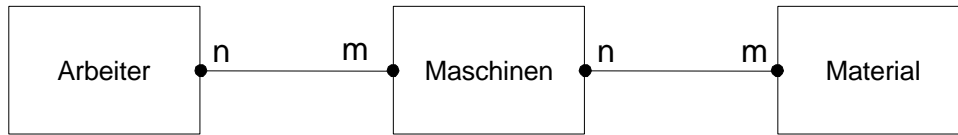
## Ein Beziehungsmodell

### Beschreibung des Problems:

1. In einer Firma werden in der Produktion den Arbeitern Maschinen zugeteilt.
2. Diese Arbeiter können verschiedene Maschinen bedienen.
3. An mehreren Maschinen können unterschiedliche Materialien verarbeitet werden.

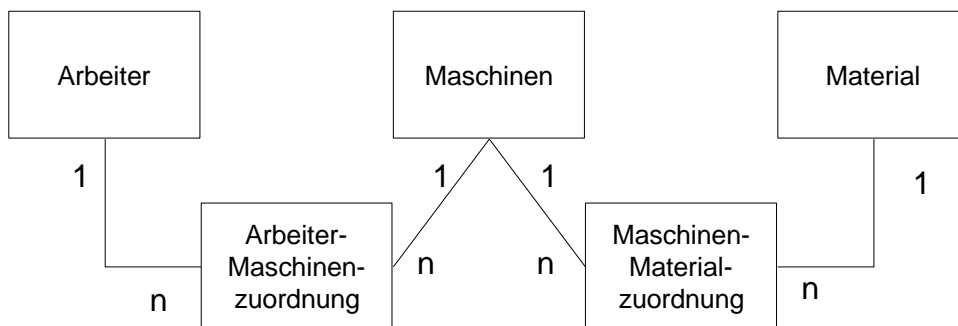
4. Das Datensystem soll nun verwalten, welche Arbeiter an welchen Maschinen welches Material verarbeiten.

Es entstehen zunächst drei Tabellen: **Arbeiter**, **Maschinen** und **Material**. Da jedoch sowohl die Arbeiter mehrere Maschinen bedienen können, als auch auf den Maschinen mehrere Materialien verarbeitet werden können, bestehen n:m-Beziehungen.



Datenmodell, nicht normalisiert

Durch das Einfügen von Hilfstabellen wird eine Normalisierung erzielt. Es gibt nur noch 1:n Beziehungen:



Datenmodell, normalisiert

Dieses Datenmodell könnte in Tabellenform folgende Struktur haben:

### Vor der Normalisierung:

Tabelle **Arbeiter**: PersNr, Name, Vorname, Strasse, PLZ, Wohnort, MaschName  
Tabelle **Maschine**: MaschNr, MaschName, PersNr, MaterialNr  
Tabelle **Material**: MaterialNr, MaschNr, MatName, Eigenschaft1, Eigenschaft2

### Problematik:

1. Da die Arbeiter unterschiedliche Maschinen bedienen, werden alle Mitarbeiterdaten für jede Maschine gespeichert.
2. Da die Materialien an unterschiedlichen Maschinen verarbeitet werden, sind sämtliche Materialdaten für jede Maschine gespeichert.

### Nach der Normalisierung:

Tabelle **Arbeiter**: PersNr, Name, Vorname, Strasse, PLZ, Wohnort  
Tabelle **PersMasch**: PersNr, MaschNr  
Tabelle **Maschine**: MaschNr, MaschName  
Tabelle **MaschMat**: MaschNr, MaterialNr  
Tabelle **Material**: MaterialNr, MatName, Eigenschaft1, Eigenschaft2

Die unterstrichenen Datenfelder sind als Primärschlüssel definiert.

### Erläuterung:

- In der Tabelle **PersMasch** stehen nur die Zuordnungen, welcher Arbeiter an welcher Maschine jeweils eingesetzt werden kann. Es wird lediglich die **PersNr** gespeichert. Die dazugehörigen Personaldaten werden über die eindeutige Personalnummer aus der Tabelle **Arbeiter** entnommen.
- In der Tabelle **MaschMat** stehen nur die Zuordnungen, welche Materialien an welcher Maschine jeweils eingesetzt werden können. Es wird lediglich die Materialnummer gespeichert. Die dazugehörigen Materialdaten werden über die eindeutige Materialnummer aus der Tabelle **Material** entnommen.

Und nun zurück zu unserer Beispieldatenbank **Auftrag.accdb**.

### Beziehung zwischen den Tabellen Bestellungen und Kunden

Betrachten Sie die Tabellen **Bestellungen** und **Kunden**, so stellen Sie fest, dass in beiden Tabellen das Feld **KdNr** vorkommt. In der Tabelle **Kunden** ist dieses Feld der Primärschlüssel, d.h. es ist eindeutig. Die Datenblattansicht verdeutlicht die Beziehung zwischen den Tabellen **Bestellungen** und **Kunden**:

| BestellNr | KdNr | ArtNr | Menge | Kennz |
|-----------|------|-------|-------|-------|
| 13        | 10   | 3367  | 20    | 1     |
| 33        | 56   | 5534  | 250   | 2     |
| 34        | 34   | 6643  | 200   | 2     |
| 55        | 12   | 6643  | 170   | 3     |
| 64        | 34   | 3367  | 80    | 2     |
| 67        | 15   | 3367  | 500   | 4     |
| 132       | 10   | 5534  | 300   | 1     |

| KdNr | Firma        | Strasse         | PLZ   | Ort     | Ansprechp   | TelNr         |
|------|--------------|-----------------|-------|---------|-------------|---------------|
| 10   | Müller KG    | Darwinstr. 20   | 01109 | Dresden | Fr. Wessel  | 0351 777 111  |
| 33   | Weiß GmbH    | Austr. 1        | 96450 | Coburg  | Hr. Schnell | 09561 234 588 |
| 34   | Schmidt & Co | Bergstr. 12     | 50933 | Köln    | Fr. Münch   | 0221 558 145  |
| 56   | Wohler KG    | Oldestr. 13     | 28211 | Bremen  | Hr. Schulze | 0421 682 735  |
| 63   | Keller OHG   | Albstr. 24      | 89081 | Ulm     | Fr. Haber   | 0731 248 345  |
| 74   | Brick GmbH   | Kolumbusring 48 | 18106 | Rostock | Hr. Berger  | 0381 122 935  |

Tabellen Bestellungen und Kunden

Die Tabelle **Kunden** enthält die Kundendaten und den eindeutigen Primärschlüssel **KdNr**. Die Tabelle **Bestellungen** enthält alle Daten für eine Bestellung, unter anderem auch die Kundennummer. Diese Information stellt die Beziehung zwischen den beiden Tabellen her. Bei jeder Bestellung kann über die Kundennummer die dazugehörige Adresse aus der Tabelle **Kunden** herausgelesen werden. Die vollständige Kundenadresse wird jedoch nur ein einziges Mal abgespeichert.

Da ein Kunde mehrere Bestellungen tätigen kann, jedoch eine Bestellung nicht gleichzeitig auf mehrere Kunden aufgeteilt wird, handelt es sich hier um eine **1:n-Beziehung**.

## Beziehung zwischen den Tabellen **Bestellungen** und **Zahlungsart**

Die Tabellen **Zahlungsart** und **Bestellungen** sind über das Feld **Kennz** miteinander verknüpft. In der Tabelle **Zahlungsart** ist dieses Feld der Primärschlüssel.

Jeder Bestellung ist genau ein Zahlungskennzeichen zugeordnet. Jedoch kann jedes Zahlungskennzeichen in mehreren Bestellungen auftreten. Es handelt sich deshalb auch hier um eine 1:n-Beziehung.

| BestellNr | KdNr | ArtNr | Menge | Kennz |
|-----------|------|-------|-------|-------|
| 13        | 10   | 3367  | 20    | 1     |
| 33        | 56   | 5534  | 250   | 2     |
| 34        | 34   | 6643  | 200   | 2     |
| 55        | 12   | 6643  | 170   | 3     |
| 64        | 34   | 3367  | 80    | 2     |
| 67        | 15   | 3367  | 500   | 4     |
| 132       | 10   | 5534  | 300   | 1     |

| Kennz | Zahlungsart     |
|-------|-----------------|
| 1     | Rechnung        |
| 2     | Kreditkarte     |
| 3     | Electronic Cash |
| 4     | Nachname        |
| 5     | Barzahlung      |

Tabellen Bestellungen und Zahlungsart

## Beziehung zwischen den Tabellen **Bestellungen** und **Artikel**

Die Tabellen **Artikel** und **Bestellungen** sind über das Feld **ArtNr** miteinander verknüpft. In der Tabelle **Artikel** ist dieses Feld der Primärschlüssel.

| BestellNr | KdNr | ArtNr | Menge | Kennz |
|-----------|------|-------|-------|-------|
| 13        | 10   | 3367  | 20    | 1     |
| 33        | 56   | 5534  | 250   | 2     |
| 34        | 34   | 6643  | 200   | 2     |
| 55        | 12   | 6643  | 170   | 3     |
| 64        | 34   | 3367  | 80    | 2     |
| 67        | 15   | 3367  | 500   | 4     |
| 132       | 10   | 5534  | 300   | 1     |

| ArtNr | ArtName     | Preis/St |
|-------|-------------|----------|
| 3367  | Damenschuhe | 89,00 €  |
| 5534  | Damenhose   | 79,00 €  |
| 6643  | Sweatshirt  | 59,90 €  |
| 0     |             | 0,00 €   |

Tabellen Bestellungen und Artikel



Die Eingabe von gleichen Bestellnummern ist nur möglich, weil bei der Definition der Tabelle die Bestellnummer *nicht* als Primärschlüssel definiert wurde.

Da eine Bestellung aus mehreren Artikeln bestehen und gleichzeitig ein Artikel in mehreren Bestellungen vorkommen kann, handelt es sich hier um eine n:m-Beziehung.

In der Tabelle **Bestellungen** kann durch diese n:m-Beziehung kein Primärschlüssel vergeben werden bzw. die Datensätze sind bei Abfragen nicht eindeutig gekennzeichnet. Die Festlegung eines Primärschlüssels ist jedoch im Normalisierungsprozess beim Datenbankentwurf notwendig, um Datenredundanz zu vermeiden.

Um dieses n:m-Problem zu lösen, definiert man eine neue Tabelle, die zu den Tabellen **Bestellungen** und **Artikel** jeweils eine 1:n-Beziehung bildet.

## Auflösung einer n:m-Beziehung

Die Einführung einer dritten Tabelle **Art-Best** dient als Bindeglied, mit dessen Hilfe die n:m-Beziehung in zwei 1:n-Beziehungen aufgelöst werden kann. In die Tabelle **Art-Best** werden die beiden Felder der Grundtabellen **Bestellungen** und **Artikel** eingefügt, so dass hieraus eindeutig hervorgeht, welche Artikel unter welcher Bestellnummer bestellt wurden. Zusätzlich kommt in die neue Tabelle **Art-Best** noch die Bestellmenge.

Tabelle ART-BEST

| BestellNr | ArtNr | Menge |
|-----------|-------|-------|
| 13        | 3367  | 20    |
| 33        | 3367  | 100   |
| 33        | 5534  | 250   |
| 34        | 6643  | 200   |
| 55        | 6643  | 170   |
| 64        | 3367  | 80    |
| 67        | 3367  | 500   |
| 132       | 5534  | 300   |

Tabelle BESTELLUNGEN mit dem Primärschlüssel BestellNr

| BestellNr | KdNr | Kennz |
|-----------|------|-------|
| 13        | 10   | 1     |
| 33        | 56   | 2     |
| 34        | 34   | 2     |
| 55        | 12   | 3     |
| 64        | 34   | 2     |
| 67        | 15   | 4     |
| 132       | 1    | 1     |

Tabelle ARTIKEL mit dem Primärschlüssel ArtNr

| ArtNr | ArtName     | Preis/St |
|-------|-------------|----------|
| 3367  | Damenschuhe | 89,00 €  |
| 5534  | Damenhose   | 79,00 €  |
| 6643  | Sweatshirt  | 59,90 €  |
| *     | 0           | 0,00 €   |

## Beziehungen zwischen den Tabellen

Die nachfolgende Tabelle zeigt nochmals die Beziehungen, die zwischen den Tabellen der Beispieldatenbank bestehen.

| Haupttabelle | Detailtabelle | Beziehung |
|--------------|---------------|-----------|
| Kunden       | Bestellungen  | 1:n       |
| Zahlungsart  | Bestellungen  | 1:n       |
| Bestellungen | Art-Best      | 1:n       |
| Artikel      | Art-Best      | 1:n       |

## 2.8 Übung

1. Kopieren Sie die Tabelle **Bestellungen** unter dem neuen Namen **Art-Best**. Dazu können Sie beispielsweise die zwei Befehle **Kopieren** und **Einfügen** im Kontextmenü verwenden.
2. Entfernen Sie in der Tabelle **Art-Best** den Primärschlüssel der Spalte **BestellNr**.
3. Löschen Sie in der Tabelle **Art-Best** die Felder **KdNr** und **Kennz**.
4. Löschen Sie in der Tabelle **Bestellungen** die Felder **ArtNr** und **Menge**.

Ihre Tabellen sollten nach Abschluss der Übung die gleiche Struktur wie in dieser Unterlage aufweisen.



## 2.9 Übung

Datenbank **FTW.acddb**

Eine Firma, die Betonfertigteile für Bauunternehmen produziert, möchte die Produktion und Auslieferung dieser Fertigteile effektiv verwalten.

### Vorüberlegungen

- Ein Werk für Betonfertigteile (FTW = Fertigteilewerk) produziert Fertigteile für die Bauwirtschaft.
- Diese Fertigteile werden nach Bauprojekten aufgeteilt und abgerechnet.
- In einem Bauprojekt (z.B. Neubau eines Bürogebäudes) wird für jedes Bauteil ein Bauplan erstellt, wobei diese Bauteile in einem Bauprojekt mehrfach einsetzbar sind. So kann man z.B. gleiche Fassadenplatten in einem Bürogebäude mehrfach einbauen.
- Da jedes Fertigteil pro Projekt mehrfach produziert werden kann, ist für jedes produzierte Fertigteil ein Fertigungs- und ein Lieferdatum zu erfassen.
- In die Fertigteile kommen mehrere Einbauteile (EBT) zum Einbau, die über Artikelnummern von Fremdfirmen gekauft werden.

In der Tabelle **Baustelle** sind die für die Produktion der Fertigteile wichtigen Daten gespeichert. Neben den Daten des Auftrages werden auch für jeden Plan gespeichert: Plannummer, Planinhalt, Länge, Breite, Höhe des Bauteils sowie Fertigungs- und Lieferdatum. Da es möglich ist, jeden Plan innerhalb eines Bauvorhabens mehrfach zu fertigen, müssen für jeden Einzelplan das Fertigungs- und Lieferdatum gespeichert werden.

| Kst   | Bauvorhaben      | Auftraggeber     | PlanNr  | Planinhalt        | Anz | Länge | Breite | Höhe | Fertigungsdat | Lieferdat  |
|-------|------------------|------------------|---------|-------------------|-----|-------|--------|------|---------------|------------|
| 95233 | Bürogebäude      | Fa. Mühlherr     | 67      | Winkel L1         | 2   | 2,53  | 1,17   | 0,2  | 13.01.2024    | 30.01.2024 |
| 95233 | Bürogebäude      | Fa. Mühlherr     | 67      | Winkel L1         | 2   | 2,53  | 1,17   | 0,2  | 17.01.2024    | 31.01.2024 |
| 95233 | Bürogebäude      | Fa. Mühlherr     | 821     | Bodenplatte       | 1   | 7,35  | 8,55   | 0,2  | 12.02.2024    | 25.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 11      | Podest            | 3   | 4,49  | 1      | 1    | 03.01.2024    | 15.01.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 11      | Podest            | 3   | 4,49  | 1      | 1    | 07.01.2024    | 17.01.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 11      | Podest            | 3   | 4,49  | 1      | 1    | 28.12.2023    | 15.01.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 31      | Fundament-03      | 2   | 3,13  | 7,8    | 1,5  | 25.11.2023    | 05.12.2023 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 35      | Frostriegel       | 4   | 2,99  | 0,15   | 0,7  | 30.12.2023    | 03.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 35      | Frostriegel       | 4   | 2,99  | 0,15   | 0,7  | 23.12.2023    | 04.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 35      | Frostriegel       | 4   | 2,99  | 0,15   | 0,7  | 20.12.2023    | 01.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 35      | Frostriegel       | 4   | 2,99  | 0,15   | 0,7  | 14.12.2023    | 01.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 40      | Treppe Keller     | 3   | 3,13  | 1,1    | 2,5  | 14.01.2024    | 03.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 40      | Treppe Keller     | 3   | 3,13  | 1,1    | 2,5  | 12.01.2024    | 30.01.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 40      | Treppe Keller     | 3   | 3,13  | 1,1    | 2,5  | 20.12.2023    | 28.01.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 66      | Treppe Eingang    | 1   | 3,6   | 0,9    | 2,5  | 01.02.2024    | 11.04.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 78      | Fassadenplatte 01 | 1   | 3,13  | 0,24   | 1,28 | 10.12.2023    | 11.03.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | 9       | Bodenplatte       | 1   | 8,05  | 12     | 0,1  | 12.12.2023    | 03.02.2024 |
| 96344 | Granitwerk Mölln | Fa. Weber        | SF 03   | Fundament-03      | 2   | 3,13  | 7,8    | 1,5  | 21.11.2023    | 05.12.2023 |
| 97940 | Neubau Büro      | Bau GmbH Niedorf | FS      | Fenstersturz      | 3   | 2     | 0,3    | 0,2  | 02.01.2024    | 10.01.2024 |
| 97940 | Neubau Büro      | Bau GmbH Niedorf | FS      | Fenstersturz      | 3   | 2     | 0,3    | 0,2  | 12.01.2024    | 17.01.2024 |
| 97940 | Neubau Büro      | Bau GmbH Niedorf | FS      | Fenstersturz      | 3   | 2     | 0,3    | 0,2  | 08.01.2024    | 10.01.2024 |
| 97975 | MFH Berlin       | Fa. Köhn GmbH    | 1 OG/H2 | Decke OG          | 3   | 12,9  | 8,75   | 1,8  | 30.11.2023    | 03.12.2023 |
| 97975 | MFH Berlin       | Fa. Köhn GmbH    | 1 OG/H2 | Decke OG          | 3   | 12,9  | 8,75   | 1,8  | 02.12.2023    | 20.12.2023 |
| 97975 | MFH Berlin       | Fa. Köhn GmbH    | 1 OG/H2 | Decke OG          | 3   | 12,9  | 8,75   | 1,8  | 07.12.2023    | 20.12.2023 |

Tabelle Baustelle, nicht strukturiert

# Schulungsunterlage Access 2021 Weiterführung

In der Tabelle **Teile** sind für jeden Plan die erforderlichen Einbauteile festgehalten:

| Kst   | Bauvorhaben      | Auftraggeber     | PlanNr  | Planinhalt        | Einbauteil       | Ar | ArtNr-Ef |
|-------|------------------|------------------|---------|-------------------|------------------|----|----------|
| 95233 | Bürogebäude      | Fa. Mühlherr     | 121     | Abdeckplatte 67/2 | Platte 40/40     | 3  | 1333     |
| 95233 | Bürogebäude      | Fa. Mühlherr     | 67      | Winkel L1         | Rippenfußanker   | 2  | 1203     |
| 95233 | Bürogebäude      | Fa. Mühlherr     | 821     | Bodenplatte       | Hüllrohr 5/7     | 1  | 1107     |
| 95233 | Bürogebäude      | Fa. Mühlherr     | 821     | Bodenplatte       | Gewindehülse M16 | 1  | 1423     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 11      | Podest            | Rippenfußanker   | 3  | 1203     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 31      | Fundament-03      | Hüllrohr 6/8     | 2  | 1589     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 35      | Frostriegel       | Rippenfußanker   | 4  | 1203     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 40      | Treppe Keller     | Gewindehülse M16 | 3  | 1423     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 66      | Treppe Eingang    | Rippenfußanker   | 1  | 1203     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 78      | Fassadenplatte 01 | Hüllrohr 5/7     | 1  | 1107     |
| 96344 | Granitwerk Möllr | Fa. Weber        | 9       | Bodenplatte       | Rippenfußanker   | 1  | 1203     |
| 96344 | Granitwerk Möllr | Fa. Weber        | SF 03   | Fundament-03      | Rippenfußanker   | 2  | 1203     |
| 97940 | Neubau Büro      | Bau GmbH Niedorf | FS      | Fenstersturz      | Platte 20/20     | 3  | 1830     |
| 97940 | Neubau Büro      | Bau GmbH Niedorf | FS      | Fenstersturz      | Platte 30/20     | 3  | 1835     |
| 97975 | MFH Berlin       | Fa. Köhn GmbH    | 1 OG/H2 | Decke OG          | Winkel L1/102    | 3  | 789      |
| 97975 | MFH Berlin       | Fa. Köhn GmbH    | 1 OG/H2 | Decke OG          | Winkel L3        | 3  | 801      |
| *     |                  |                  |         |                   |                  |    |          |

Tabelle Teile, nicht strukturiert

1. Normalisieren Sie diese Beispieldatenbank auf dem Papier.
2. Tragen Sie die normalisierten Tabellen in eine Access-Datenbank ein und speichern Sie diese unter dem Namen **FTW2** ab. Achten Sie auf Datenkonsistenz und Datenintegrität.

Ein gemeinsamer Lösungsweg für die Übung von dieser Seite und der Übung von der Seite 32 (Referentielle Integrität) steht ab der Seite 152.

## 3 Tabellen verbinden

Im vorherigen Kapitel haben Sie Beziehungen zwischen Tabellen erarbeitet, um Datenredundanzen und Anomalien zu vermeiden. In diesem Kapitel sollen diese Beziehungen in Access festgelegt werden.

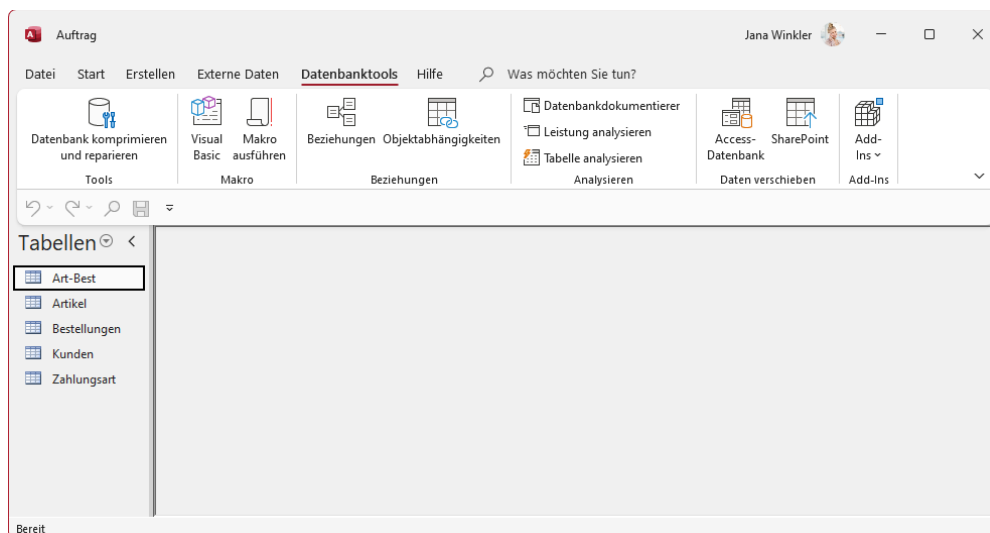
### 3.1 Haupt- und Detailtabelle

In der Datenbank **Auftrag** ist in der Tabelle **Kunden** die Kundennummer einmalig, d.h. jeder Datensatz kann durch die Kundennummer eindeutig identifiziert werden. In Access bezeichnet man eine solche Tabelle als **Haupttabelle** oder auch **Mastertabelle**.

Nimmt eine andere Tabelle Bezug auf die Kundennummer in der Tabelle **Kunden**, so bezeichnet man diese Tabelle als **Detailtabelle**. In einer Detailtabelle kann die Information Kundennummer mehrmals auftreten.

### 3.2 Beziehungen zwischen Tabellen herstellen

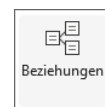
In dem nachfolgenden Bild sind die Tabellen der Datenbank **Auftrag.accdb** geschlossen. Den Navigationsbereich mit den Tabellennamen am linken Rand können Sie beispielsweise mit der **F11**-Taste verkleinern und vergrößern.



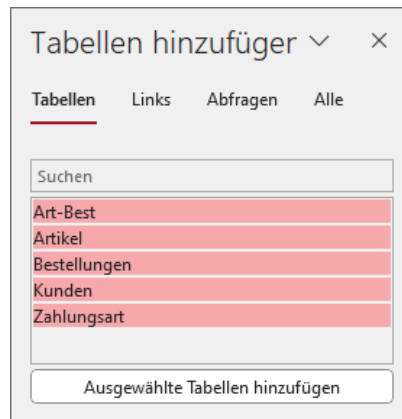
Datenbank Auftrag mit dem Navigationsbereich

Nun erstellen Sie die Beziehungen zwischen den Tabellen:

1. Klicken Sie auf der Registerkarte **Datenbanktools** in der Gruppe **Beziehungen** auf die Schaltfläche **Beziehungen**. Das zunächst noch leere Fenster **Beziehungen** und die rechte Seitenleiste **Tabellen hinzufügen** werden geöffnet.
2. Markieren Sie in der rechten Seitenleiste **Tabellen hinzufügen** mit einem Mausclick den Tabellennamen in der ersten Zeile und klicken Sie bei gedrückter **⇧**-Taste auch auf die letzte Zeile in dieser Liste. Damit sind alle Tabellennamen markiert.
3. Klicken Sie auf **Ausgewählte Tabellen hinzufügen** und schließen Sie die Seitenleiste X:

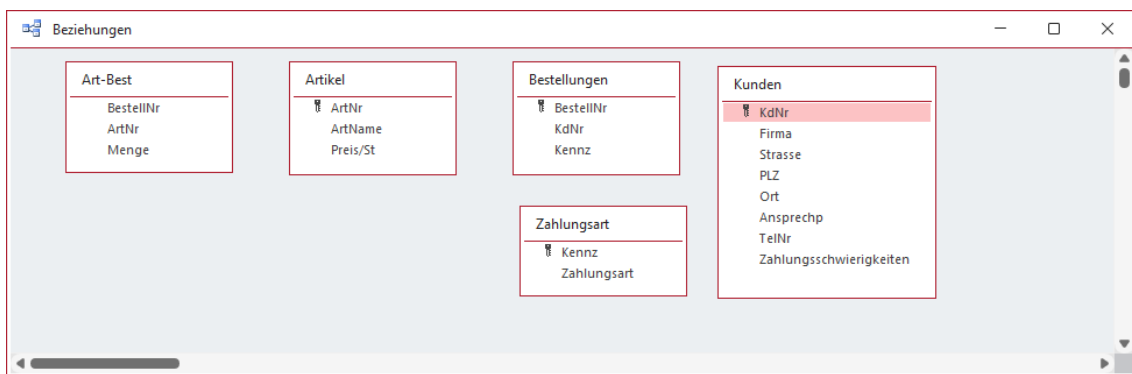


Auf der Registerkarte Datenbanktools



Dialogfeld Tabelle anzeigen

4. In dem Fenster **Beziehungen** sind jetzt die hinzugefügten Tabellen mit allen Feldern jeweils in einem kleinen Rahmen platziert. Die Primärschlüssel der Tabellen erkennen Sie an den kleinen Schlüsseln vor dem Feldnamen.
5. Sie können die kleinen Tabellenrahmen umstellen: Ziehen Sie bei gedrückter linker Maustaste die Titelleiste eines Rahmens an eine andere Position.



Fenster Beziehungen mit den hinzugefügten Tabellen, jedoch noch ohne Beziehungen zwischen den Tabellen

So erstellen Sie zwischen den Tabellen **Kunden** und **Bestellungen** eine Beziehung:

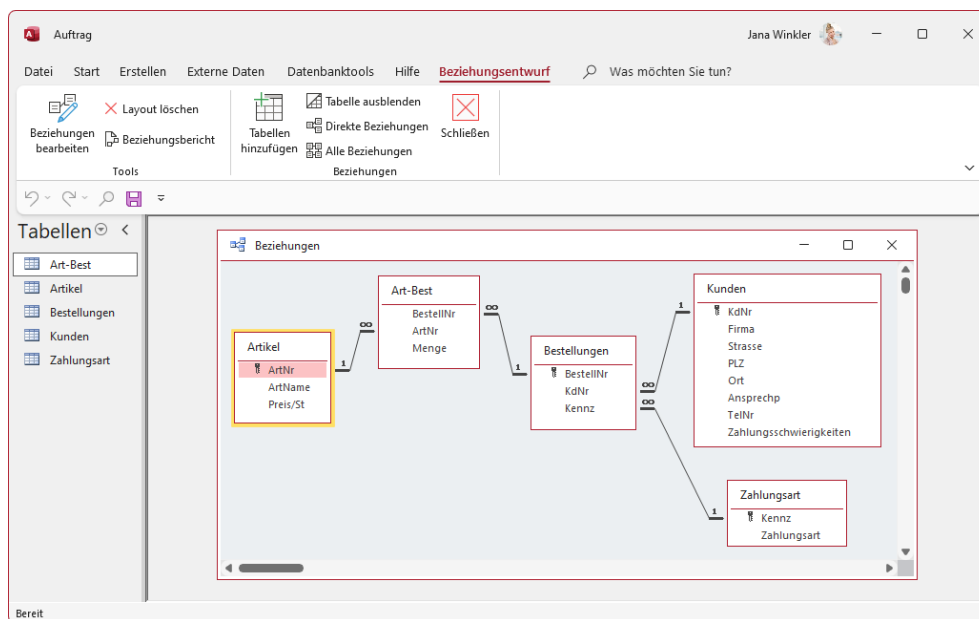
1. Im Fenster **Beziehungen** klicken Sie in der Tabelle **Kunden** mit der Maus auf das Feld **KdNr**. Lassen Sie die Maustaste gedrückt und ziehen Sie das Maussymbol auf das Feld **KdNr** in der Tabelle **Bestellungen**. Das Dialogfeld, das jetzt erscheint, hat den Namen **Beziehungen bearbeiten** (siehe unten). Hier werden als Haupt- und Detailtabelle die beiden Tabellen angezeigt, die Sie durch das Mausziehen miteinander verbunden haben.
2. Überprüfen Sie, dass in beiden Listenfeldern das Feld **KdNr** ausgewählt ist.
3. Klicken Sie auf die Schaltfläche **Erstellen**. Die Beziehung zwischen den Tabellen **Kunden** und **Bestellungen** ist nun festgelegt. Access zeigt dies grafisch durch eine Verbindungslinie zwischen den Tabellenfeldern an.

Dialogfeld Beziehungen bearbeiten

Erstellen Sie nun auf die gleiche Weise die Beziehungen für folgende Tabellen:

| Haupttabelle | Detailtabelle | Feld in Haupttabelle | Feld in Detailtabelle |
|--------------|---------------|----------------------|-----------------------|
| Zahlungsart  | Bestellungen  | Kennz                | Kennz                 |
| Bestellungen | Art-Best      | BestellNr            | BestellNr             |
| Artikel      | Art-Best      | ArtNr                | ArtNr                 |

Wenn Sie alle Beziehungen festgelegt haben, sieht Ihr Beziehungs-Layout folgendermaßen aus:



Die Beziehungen zwischen den Tabellen in der Datenbank Auftrag.accdB mit der Registerkarte Beziehungsentwurf

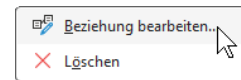


Die Positionen der Tabellen ändern Sie durch Ziehen der Titelleisten.

## 3.3 Beziehungen bearbeiten

Im Menüband ist die Kontext-Registerkarte **Beziehungsentwurf** geöffnet. Einmal erstellte Beziehungen können Sie jederzeit bearbeiten:

- Zeigen Sie mit der Maus auf die Verbundlinie und klicken Sie mit der *rechten* Maustaste. Im Kontextmenü wählen Sie den Befehl **Beziehung bearbeiten**.
- Oder doppelklicken Sie auf die Verbundlinie.
- Oder markieren Sie mit einem Mausklick die Verbundlinie für die Beziehung, die Sie bearbeiten möchten, und klicken Sie auf der Registerkarte **Beziehungsentwurf** auf die rechtsstehende Schaltfläche.



Kontextmenü

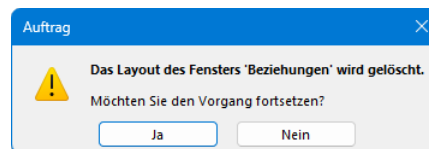


Auf der Registerkarte  
Beziehungsentwurf

Das Dialogfeld **Beziehungen bearbeiten** wird wieder angezeigt. Nehmen Sie die gewünschten Änderungen vor und schließen Sie das Dialogfeld.

### 3.4 Beziehungen löschen

Um den *gesamten Inhalt* des Fensters **Beziehungen** zu löschen, klicken Sie auf der Registerkarte **Beziehungsentwurf** auf die Schaltfläche **Layout löschen**. Bestätigen Sie die nachfolgende Meldung:



Löschen?

Die Änderung ist aber nur wirksam, wenn Sie speichern. Falls Sie aus Versehen das Layout gelöscht haben, schließen Sie das Fenster **Beziehungen** über das Schließen-Symbol **x** oben rechts und klicken bei der Frage nach dem Speichern auf **Nein**.

### Einzelne Beziehungen löschen

Möchten Sie nur eine einzelne Beziehung löschen,

- klicken Sie mit der *rechten* Maustaste auf die Verbundlinie. Im Kontextmenü wählen Sie den Befehl **Löschen**.
- Oder markieren Sie mit einem Mausklick die Verbundlinie für die Beziehung, die Sie löschen möchten, und drücken Sie die **[Entf]**-Taste.

Wenn Sie die darauf folgende Rückfrage bestätigen, wird die Beziehung in dem Fenster gelöscht. Aber erst nach dem Speichern ist die Löschung dauerhaft wirksam.

### 3.5 Referentielle Integrität

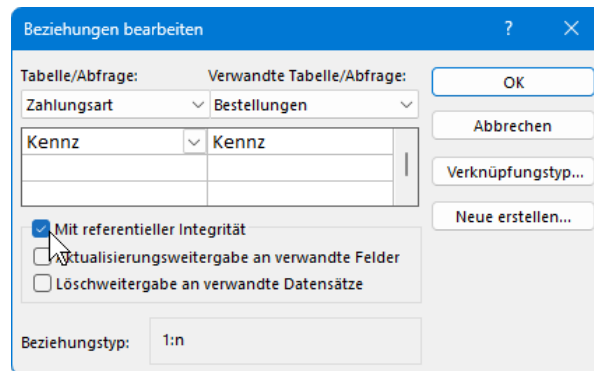
**Referentielle Integrität** bedeutet, dass zwischen zwei Tabellen eine feste Verbindung hergestellt wird, so dass bei der Datenbearbeitung (Änderung, Löschung) zunächst in der Verbindungstabelle geprüft wird, ob ein Bearbeitungsvorgang überhaupt zulässig ist. Innerhalb der referentiellen Integrität sind zwei Einstellungen möglich, die am besten anhand eines Beispiels erklärt werden.

Wir arbeiten mit der Datenbank **Auftrag.accdb**. Falls geschlossen, öffnen Sie das Fenster **Beziehungen**: Klicken Sie auf der Registerkarte **Datenbanktools** in der Gruppe **Beziehungen** auf die Schaltfläche **Beziehungen**.



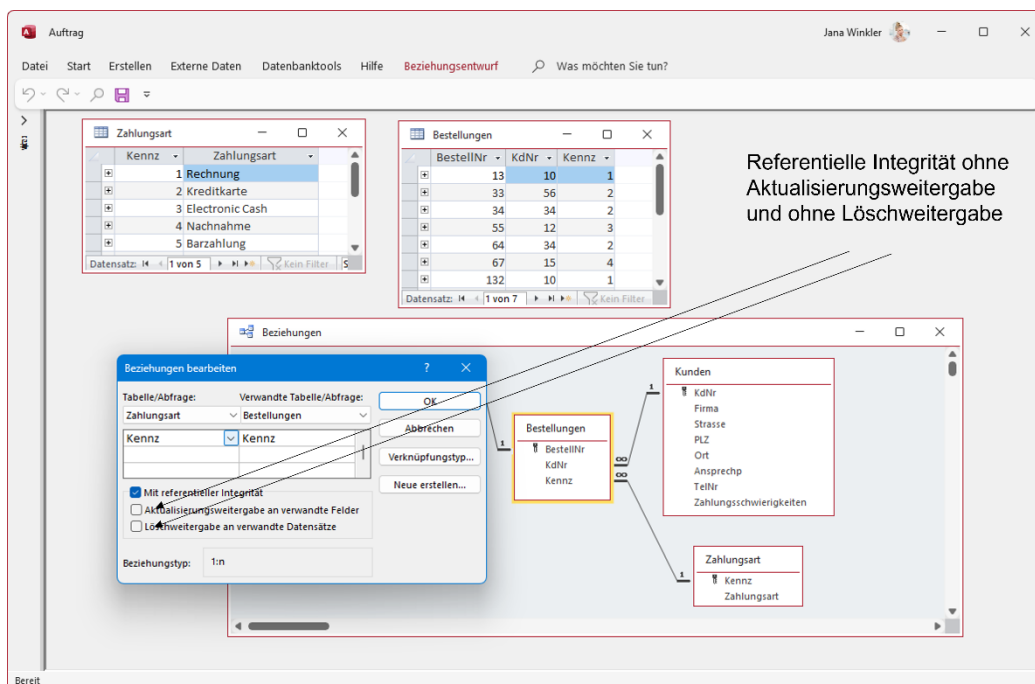
## Aktivieren der referentiellen Integrität

1. Markieren Sie mit einem Mausklick die Verbindungslinie zwischen den Tabellen **Zahlungsart** und **Bestellungen**. Diese beiden Tabellen stehen über das Feld **Kennz** miteinander in Beziehung.
2. Wählen Sie über das Kontextmenü (*rechte Maustaste*) den Befehl **Beziehung bearbeiten**. Es wird das nachfolgende Dialogfeld angezeigt.
3. Aktivieren Sie das Kontrollkästchen ☒ **Mit referentieller Integrität**.



Mit referentieller Integrität

4. Bestätigen Sie Ihre Eingabe mit .



Beziehungen und referentielle Integrität

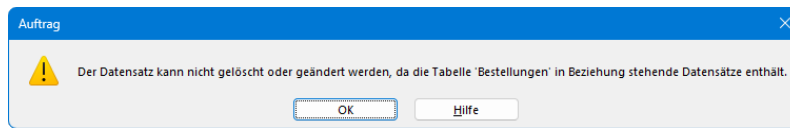
5. Öffnen Sie im Navigationsbereich (Seite 25) mit einem Doppelklick die Tabelle **Zahlungsart**.
6. Versuchen Sie hier den Datensatz mit dem Kennzeichen 1 und der Zahlungsart **Rechnung** zu löschen.

Access öffnet anschließend eine Meldung, dass dieser Datensatz nicht gelöscht werden kann. Denn das Kennzeichen 1 wird ja in der Tabelle **Bestellungen** verwendet. Würde Access diesen



# Schulungsunterlage Access 2021 Weiterführung

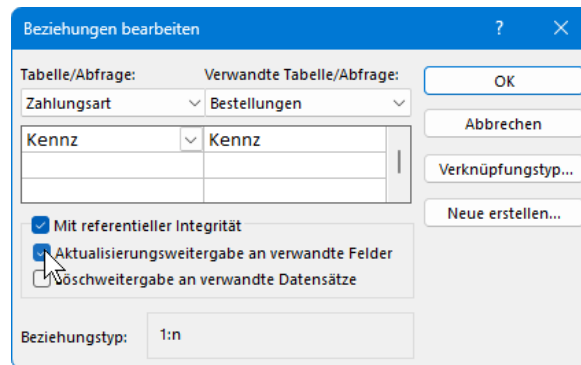
Datensatz löschen, so stünde zwar in der Tabelle **Bestellungen** unter **Kennz** eine **1**, die Zuordnung zwischen Kennzeichen und Zahlungsart würde jedoch fehlen.



Microsoft Access Hinweis

**Aktivieren der referentiellen Integrität mit Aktualisierungsweitergabe an verwandte Felder**  
Die Aktualisierungsweitergabe bewirkt, dass Access im Fall einer Aktualisierung eines Primärschlüssels automatisch alle Felder, die auf den Primärschlüssel verweisen, auf den neuesten Stand bringt.

1. Schließen Sie alle Fenster, ausgenommen das Fenster **Beziehungen**.
2. Klicken Sie wieder mit der *rechten* Maustaste auf die Verbindungslinie zwischen den Tabellen **Zahlungsart** und **Bestellungen** und wählen Sie im Kontextmenü den Befehl **Beziehung bearbeiten**. Das Dialogfeld wird angezeigt.
3. Aktivieren Sie das Kontrollkästchen ☒ Aktualisierungsweitergabe an verwandte Felder:



Die Aktualisierungsweitergabe an verwandte Felder aktivieren

4. Bestätigen Sie Ihre Eingabe mit .
5. Öffnen Sie im Navigationsbereich die Tabelle **Zahlungsart** und ändern Sie hier den Datensatz mit dem Kennzeichen **1** und der Zahlungsart **Rechnung**, indem Sie das Kennzeichen **1** in **6** ändern.
6. Öffnen Sie auch die Tabelle **Bestellungen**: Access hat nun auch in dieser Detailtabelle alle Daten in eine **6** geändert, in denen als Kennzeichen vorher die **1** stand. Diese Änderung ist auch notwendig, da ansonsten in der Tabelle **Bestellungen** ein Kennzeichen stehen würde, welches in der Tabelle **Zahlungsart** nicht mehr existiert.

Ergebnis: Referentielle Integrität mit Aktualisierungsweitergabe an das Detailfeld

7. Machen Sie nun die vorgenommene Änderung wieder rückgängig. Die Zahlungsart **Rechnung** hat dann wieder das Kennzeichen 1.

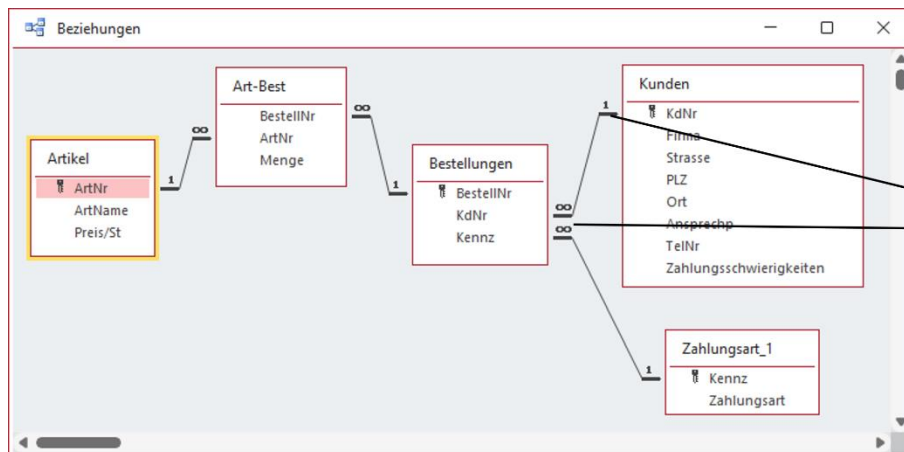
## Aktivieren der referentiellen Integrität mit Löschweitergabe an verwandte Datensätze

Ist im Dialogfeld **Beziehungen bearbeiten** das Kontrollkästchen ☒ **Löschweitergabe an verwandte Datensätze** aktiviert, so wird beim Löschen eines Datensatzes in der Tabelle **Zahlungsart** jeder Datensatz in der Tabelle **Bestellungen** gelöscht, der dasselbe Kennzeichen hat.

**Ihre Aufgabe:** Aktivieren Sie nur die **referentielle Integrität** für *alle* Beziehungen in unserer Beispieldatenbank **Auftrag.accdb** und speichern Sie ab.



Danach werden auch die Beziehungstypen an der Verbindungslinie angezeigt:

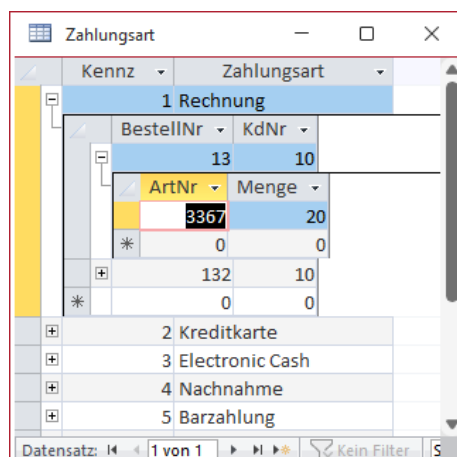


1:n-Beziehung zwischen den Tabellen Kunden und Bestellungen, da ein Kunde mehrere Bestellungen vornehmen kann. Jede Bestellung bezieht sich aber auf genau einen Kunden.

Beziehungen zwischen Tabellen der Datenbank Auftrag.accdb mit Anzeige der Beziehungstypen

## 3.6 Unterdatenblätter

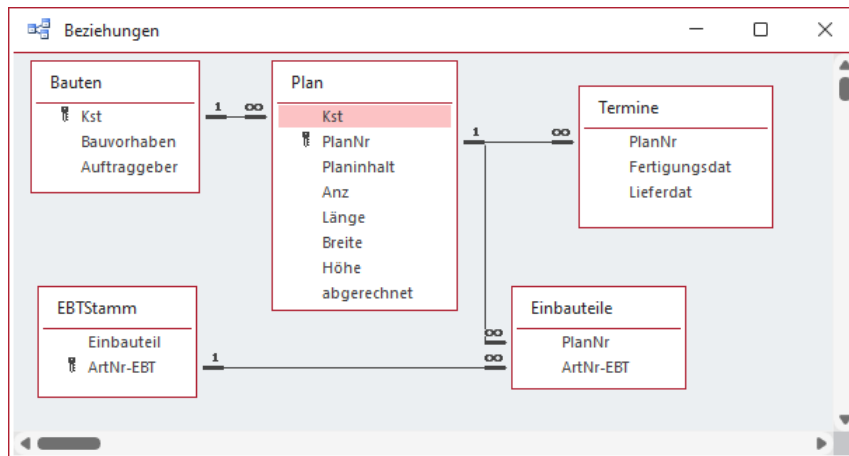
Bei verknüpften Daten in einer Tabelle hat Access am linken Tabellenrand eine neue Spalte eingefügt für die Anzeige von Unterdatenblättern. Mit einem Mausklick auf das Erweiterungssymbol öffnen Sie ein Unterdatenblatt mit den verknüpften Daten, die zu dem aktuellen Datensatz gehören. Zum Schließen klicken Sie auf das -Symbol:



Unterdatenblätter bei verknüpften Daten

### 3.7 Übung

Erstellen Sie für die bereits normalisierte Datenbank **FTW2.accdb** die Beziehungen jeweils mit referentieller Integrität. Eventuelle Dateninkonsistenzen sollten Sie bereinigen.



Beziehungsschema der Datenbank FTW2.accdb

Ein gemeinsamer Lösungsweg für die Übung von der Seite 23 (Normalisierung der Datenbank FTW) und der Übung von dieser Seite steht ab der Seite 152.

## Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Theoretische Datenbank-Grundlagen .....</b>       | <b>5</b>  |
| 1.1      | Datenbankmodelle .....                               | 6         |
| 1.2      | Wichtige Grundsätze einer Datenbank.....             | 6         |
| 1.3      | Konventionelle Datenverarbeitung und Datenbank ..... | 8         |
| <b>2</b> | <b>Der Datenbankentwurf .....</b>                    | <b>9</b>  |
| 2.1      | Schritte beim Entwurf einer Datenbank.....           | 9         |
| 2.2      | Tabellen .....                                       | 9         |
| 2.3      | Primär- und Sekundärschlüssel .....                  | 11        |
| 2.4      | Beispiel: Datenbank Auftrag .....                    | 12        |
| 2.5      | Normalisierung .....                                 | 13        |
| 2.6      | Übung .....  | 16        |
| 2.7      | Die Beziehungen zwischen den Tabellen bestimmen..... | 17        |
| 2.8      | Übung .....  | 22        |
| 2.9      | Übung .....  | 23        |
| <b>3</b> | <b>Tabellen verbinden .....</b>                      | <b>25</b> |
| 3.1      | Haupt- und Detailtabelle .....                       | 25        |
| 3.2      | Beziehungen zwischen Tabellen herstellen .....       | 25        |
| 3.3      | Beziehungen bearbeiten.....                          | 27        |
| 3.4      | Beziehungen löschen .....                            | 28        |
| 3.5      | Referentielle Integrität .....                       | 28        |
| 3.6      | Unterdatenblätter .....                              | 31        |
| 3.7      | Übung .....  | 32        |
| <b>4</b> | <b>Formeln und Funktionen .....</b>                  | <b>33</b> |
| 4.1      | Wissenswertes über Formeln und Funktionen .....      | 33        |
| 4.2      | Zusammenfassung.....                                 | 36        |
| 4.3      | Funktionen .....                                     | 36        |
| <b>5</b> | <b>Abfragen .....</b>                                | <b>43</b> |
| 5.1      | Abfragen über mehrere Tabellen .....                 | 43        |
| 5.2      | Berechnungen in Abfragen .....                       | 45        |
| 5.3      | Feldeigenschaften in Abfragen .....                  | 46        |
| 5.4      | Übung .....  | 49        |
| 5.5      | Aktionsabfragen .....                                | 49        |
| 5.6      | Löschabfragen .....                                  | 57        |
| 5.7      | Parameterabfrage .....                               | 59        |
| 5.8      | Abfragen mit besonderen Verknüpfungen (Joins) .....  | 60        |
| <b>6</b> | <b>Abfragen mit SQL.....</b>                         | <b>64</b> |
| 6.1      | Grundstruktur des SELECT-Befehls .....               | 64        |
| 6.2      | Abfrage mit einem SQL-Befehl erstellen .....         | 66        |
| 6.3      | Komponenten des SELECT-Befehls .....                 | 67        |
| 6.4      | Die JOIN-Operation.....                              | 71        |
| 6.5      | Prädikate.....                                       | 72        |
| <b>7</b> | <b>Weitere Abfrage-Anwendungsmöglichkeiten .....</b> | <b>73</b> |
| 7.1      | Abfrage zur Duplikatsuche.....                       | 73        |
| 7.2      | Spitzen- und Tiefstwerte in Abfragen anzeigen .....  | 75        |
| 7.3      | Gruppierungsabfragen mit Aggregatfunktionen .....    | 76        |
| <b>8</b> | <b>Formulare .....</b>                               | <b>78</b> |
| 8.1      | Neues Formular erstellen .....                       | 78        |
| 8.2      | Datensatzquelle .....                                | 80        |
| 8.3      | Die Steuerelemente.....                              | 81        |

|           |   |            |
|-----------|---|------------|
| 8.4       | Steuerelemente einfügen .....                                 | 84         |
| 8.5       | Übung .....   | 89         |
| 8.6       | Objekt einfügen .....   | 96         |
| 8.7       | Linien und Rechtecke .....                                    | 98         |
| 8.8       | Ausrichtungshilfen .....                                      | 98         |
| 8.9       | Eigenschaften .....   | 99         |
| 8.10      | Feldreihenfolge festlegen .....                               | 100        |
| 8.11      | Übung .....   | 101        |
| 8.12      | Geteiltes Formular .....                                      | 102        |
| 8.13      | Unterformulare .....  | 103        |
| 8.14      | Übung .....   | 107        |
| <b>9</b>  | <b>Datenanalysen.....</b>                                     | <b>108</b> |
| 9.1       | Kreuztabellen .....   | 108        |
| 9.2       | Übung .....   | 110        |
| 9.3       | PivotTabellen in Excel erstellen .....                        | 111        |
| 9.4       | Übung .....   | 116        |
| <b>10</b> | <b>Berichte .....</b>   | <b>117</b> |
| 10.1      | Einen neuen Bericht erstellen .....                           | 117        |
| 10.2      | Die Berichtsbereiche .....                                    | 118        |
| 10.3      | Datensatzquelle .....   | 119        |
| 10.4      | Bericht gruppieren .....                                      | 119        |
| 10.5      | Mehrspaltiger Bericht .....                                   | 121        |
| 10.6      | Datenfelder einfügen .....                                    | 122        |
| 10.7      | Die Berichts-Ansichten .....                                  | 124        |
| 10.8      | Eigenschaften in Berichten .....                              | 126        |
| 10.9      | Berechnende Felder und Funktionen einfügen.....               | 127        |
| 10.10     | Übung .....   | 129        |
| <b>11</b> | <b>Automatisierung durch Makros.....</b>                      | <b>130</b> |
| 11.1      | Makros erstellen .....  | 130        |
| 11.2      | Eigene Menübefehle (Makros) einer Registerkarte zufügen ..... | 133        |
| 11.3      | Standardstartformular erstellen.....                          | 135        |
| 11.4      | Standardstartformular aktivieren.....                         | 139        |
| <b>12</b> | <b>Datenaustausch .....</b>                                   | <b>141</b> |
| 12.1      | Import – Struktur einer Datenbank .....                       | 141        |
| 12.2      | Export .....  | 144        |
| 12.3      | Verknüpfung.....  | 146        |
| 12.4      | Tabellenverknüpfungs-Manager .....                            | 148        |
| 12.5      | Übung .....   | 149        |
| <b>13</b> | <b>Datenschutz und Datensicherheit .....</b>                  | <b>150</b> |
| 13.1      | Datensicherheit .....   | 150        |
| 13.2      | Datenschutz .....   | 150        |
| <b>14</b> | <b>Lösungswege für einige Übungen .....</b>                   | <b>152</b> |
| <b>15</b> | <b>Stichwortverzeichnis .....</b>                             | <b>161</b> |