

3 Datenbankmodellierung

In Microsoft Access haben wir bereits mit Datenbanken gearbeitet. In diesem Kapitel lernen wir, wie wir eine Datenbank erstellen. Zur Datenbankmodellierung wenden wir das Entity-Relationship-Modell an, das wir im Rahmen der relationalen Auflösung in eine Datenbank umwandeln. Mittels Indexierung und referenzieller Integrität optimieren wir unsere Datenbank. Bei der Erstellung neuer Datenbanken beschäftigen wir uns außerdem mit Beziehungen und der Festlegung von Schlüsselfeldern.

Lerneinheit 1: Modellierung

Lernen	92
1 Datenmodelle	92
2 Entity-Relationship-Modell	94
3 Zusammengesetzte Attribute	95
4 Schlüsselattribute	96
5 Virtuelle Attribute	97
6 Fehlende Werte	98
7 Kardinalität	99
Üben	100
Sichern	102
Wissen	102

Lerneinheit 2: Relationale Auflösung

Lernen	104
1 ER-Modell zeichnen	104
2 Relationale Auflösung	105
3 Normalisierung	108
4 Normalformen	109
5 Indexierung	110
6 Referenzielle Integrität	111
Üben	112
Sichern	114
Wissen	115

Lerneinheit 3: Erstellen einer Datenbank

Lernen	116
1 Datenbankschema	116
2 Datenbank anlegen	118
3 Primärschlüssel	119
4 Beziehungen erstellen	120
5 Verknüpfungseigenschaften	121
Üben	123
Sichern	125
Wissen	125
Kapitelrückblick	128



Ich bin Mr. What! Ich kenne mich ganz gut aus, habe aber immer wieder einige Zwischenfragen ...



Ich bin Ms. Check! Ich kenne mich sehr gut aus und habe auf jede Frage eine verständliche Antwort.

Lerneinheit 1 Modellierung

SbX

Alle SbX-Inhalte zu dieser Lerneinheit findest du unter der ID: 9907.

In dieser Lerneinheit lernen wir die Modellierung von Datenbanken mit Hilfe des Entity-Relationship-Modells kennen. Wir beschäftigen uns mit

- Entitäten, Beziehungen und Attributen,
- zusammengesetzten und virtuellen Attributen,
- Schlüsselattributen und fehlenden Werten sowie mit
- der Kardinalität von Beziehungen.

Lernen

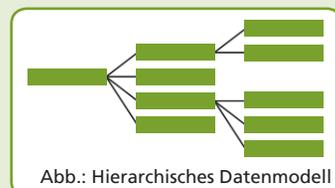
1 Datenmodelle Formen strukturierter Datenspeicherung

Anhand von **Hierarchien**, wie z.B. Abteilungen oder Profitcentern, werden Unternehmensorganisationen gegliedert.

Im zweiten Kapitel haben wir uns bereits intensiv mit dem Thema Datenbanken und mit deren Anwendung beschäftigt. Zur Beschreibung der Art und Weise, wie Daten in einer Datenbank gespeichert werden, gibt es verschiedene **Datenmodelle**. Einige dieser Modelle finden wir auch im Bereich der betriebswirtschaftlichen Organisationslehre, wie z.B. das **hierarchische Datenmodell** in Form der **Stablinienorganisation** oder das **Netzwerk-Datenmodell** bei der **Matrixorganisation**.

Arten von Datenmodellen

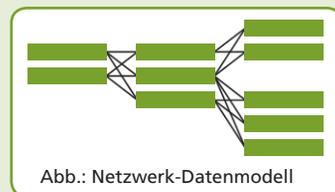
1 Hierarchisches Datenmodell



Das **hierarchische Datenmodell** kennen wir vom Dateisystem unserer Festplatte (Laufwerksbuchstabe, Ordner, Dateien). Wir nennen dieses Modell auch eine **Baumstruktur**. Ganz links befindet sich die **Wurzel** (Root). Von ihr sind alle Objekte abhängig, die weitere abhängige Objekte haben können.

Hierarchische Datenmodelle ermöglichen sehr rasches Suchen und Sortieren von Daten.

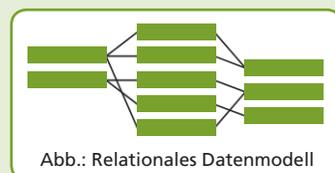
2 Netzwerk-Datenmodell



Beim **Netzwerk-Datenmodell** kann es **Abhängigkeiten** zwischen **allen Datenobjekten** geben. Wenn wir im Internet eine Webseite aufrufen, stellen wir eine Abhängigkeit zwischen unserem PC und einem Internet-Host (Server) her. Auf diese Art und Weise kann jeder PC mit jedem Host kommunizieren und umgekehrt.

Das **Netzwerk-Datenmodell** wird z.B. in Peer-to-Peer-Netzwerken eingesetzt.

3 Relationales Datenmodell

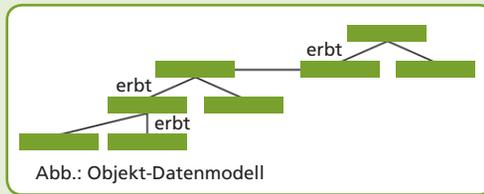


Wenn wir die ersten beiden Modelle kombinieren, so erhalten wir das **relationale Datenmodell**. Ein **Datenobjekt ist von einem oder mehreren Datenobjekten abhängig**. Daraus ergeben sich drei Arten von Abhängigkeiten, die wir als **Beziehungen** bezeichnen: **1:1**, **1:n** und **n:m**.

Das **relationale Datenmodell** ist das am häufigsten anzutreffende Datenmodell in Datenbanksystemen.

4 Objekt-Datenmodell

Ein wichtiges Prinzip beim **Objektmodell** ist die **Vererbung**, durch die ein effizienteres Programmieren möglich wird.



Objekte sind **modellhafte Abbilder der Wirklichkeit**. Das **Objekt-Datenmodell** wird vor allem im Bereich der **Softwareentwicklung** eingesetzt.



Ü 1:

Um welches Datenmodell handelt es sich in diesen Beispielen?

- Das österreichische Bundesheer möchte seine Befehlsstruktur speichern.
- Ein Supermarkt möchte alle Produkte in Produktgruppen und Produktuntergruppen gliedern und speichern.
- Ein Haubenrestaurant möchte die Rezepte und Zutaten verwalten.
- Ein Netzwerkadministrator möchte Benutzergruppen, Benutzer und Benutzerrechte verwalten.

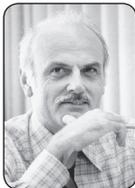


Abbildungen zu den Datenmodellen findest du in der PowerPoint-Präsentation unter der ID: 9908.

In **Datenbanken** (z.B. Microsoft Access) und Datenbanksystemen (z.B. Microsoft SQL-Server, Oracle, DB2 und MySQL) wird das **relationale Datenmodell** angewendet.

Relationales Datenmodell

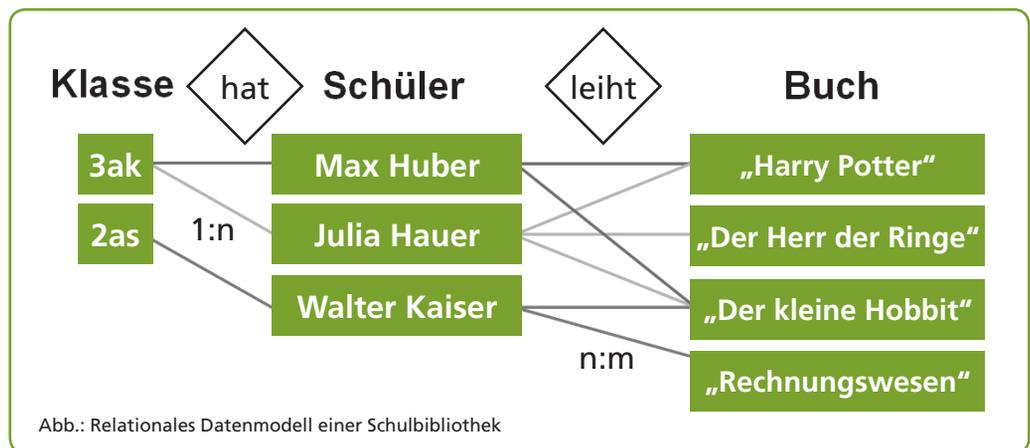
Das relationale Datenmodell wurde in den frühen 1970er Jahren vom englischen Mathematiker **Edgar F. Codd** entwickelt.



Dr. Edgar Frank Codd (1923–2003) war ein englischer Mathematiker und entwickelte bei IBM das relationale Datenmodell.



Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.



Mr. What und Ms. Check

Wie funktioniert das relationale Datenmodell?



Das Buch „Harry Potter“ wurde z.B. von Max Huber und Julia Hauer entliehen. Max Huber hat sich aber auch das Buch „Der kleine Hobbit“ ausborgt. Die Beziehung „Schüler leiht Buch“ ist eine n:m-Beziehung. In eine Klasse gehen mehrere Schüler, ein Schüler besucht jedoch nur eine Klasse. Das ist eine 1:n-Beziehung.



Zur Abbildung der Wirklichkeit in einem **relationalen Datenmodell** wurde das **Entity-Relationship-Modell** entwickelt.



Dr. Peter Chen entwickelte 1976 das ER-Modell und ist seit 1983 Professor für Informatik an der Louisiana State University, USA.

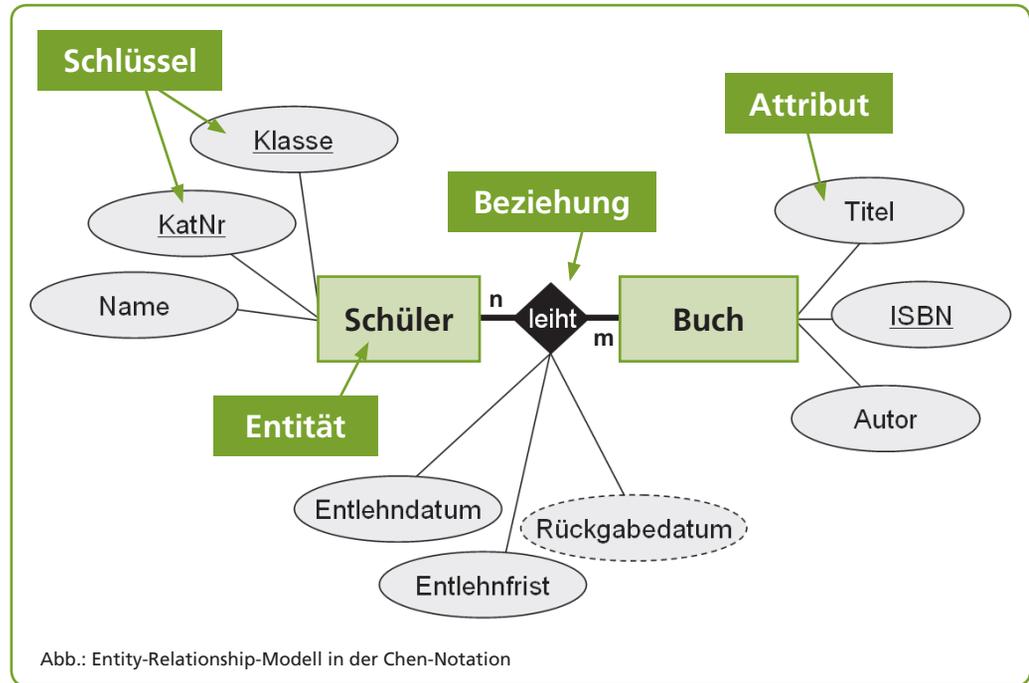
In der Chen-Notation werden Entitäten, Beziehungen und Attribute dargestellt. Der Typ der Beziehung *Schüler leiht Buch* ist als n:m-Beziehung dargestellt.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.

2 Entity-Relationship-Modell Modellierung einer relationalen Datenbank

Nachdem wir bereits einige Datenmodelle kennengelernt haben, steht nun die Modellierung von relationalen Datenbanken im Mittelpunkt. Ein **Modell** beschreibt ein Abbild der Wirklichkeit. Die Modellierung relationaler Datenbanken erfolgt mit dem von **Peter Chen** entwickelten **Entity-Relationship-Modell (ER-Modell, ERM)**.



Bestandteile des Entity-Relationship-Modells

1 Entitäten sind eindeutig unterscheidbare Objekte. Exemplare (Instanzen) sind konkrete Ausprägungen einer Entität.

Die Entität **Schüler** hat die Exemplare „Max Muster“ aus der 3as, „Hubert Auer“ aus der 3as, „Philipp Müller“ aus der 3bs usw.

2 Beziehungen sind Assoziationen zwischen Entitäten. Auch Beziehungen haben Exemplare (Instanzen).

Ein Schüler leiht ein Buch aus. Die Beziehung **leiht** hat die Exemplare „Max Muster leiht Harry Potter“, „Hubert Auer leiht Harry Potter“, „Hubert Auer leiht Wilhelm Tell“.

3 Attribute sind Eigenschaften von Entitäten oder Beziehungen.

Attribute von **Schüler** sind z.B. „Katalognummer“ und „Name“. Attribute der Beziehung **leiht** (Schüler leiht Buch) sind z.B. „Entlehndatum“ und „Entlehnfrist“.

4 Die Attribute einer Entität sind vom Schlüsselattribut abhängig.

Schüler „Max Muster, 3as“ hat die Katalognummer 1. Kein anderer Schüler der 3as hat diese Katalognummer. **Klasse** und **Katalognummer** sind das **zusammengesetzte Schlüsselattribut** eines Schülers.

SbX

Abbildungen zu den Bestandteilen des ER-Modells findest du in der PowerPoint-Präsentation unter der ID: 9908.



Ü 2:

Nimm ein leeres Blatt Papier zur Hand und zeichne

- a) die Entitäten **Lehrer** und **Schüler** sowie die Beziehung **unterrichtet**,
- b) die **Attribute** zu den Entitäten und zur Beziehung sowie
- c) die **Schlüsselattribute** zu den Entitäten.

Mr. What und Ms. Check

Warum werden Entitäten nicht als Tabellen bezeichnet?



Das ER-Modell ist ein Modell – keine Datenbank. Um diesen Unterschied auszudrücken, gibt es die Begriffe Entität und Tabelle. Im Rahmen der **relationalen Auflösung** werden Entitäten in Tabellen transformiert, Attribute werden zu Feldern der Tabelle.



Die Entität **Schüler** enthält das Attribut **Name**, das seinerseits aus dem Vor- und dem Zunamen besteht. In einer Datenbank sollte es möglichst keine zusammengesetzten Attribute geben, denn das Sortieren und Filtern sind bei zusammengesetzten Attributen meist schwierig.

3 Zusammengesetzte Attribute

Attribute müssen atomar sein

Zusammengesetzte Attribute werden im ER-Modell mit Linien zwischen den Attributteilen dargestellt.

In der Entität **Lehrer** gibt es das Attribut **Name**, z.B. mit dem Exemplar „Dr. Heinrich Specht“.

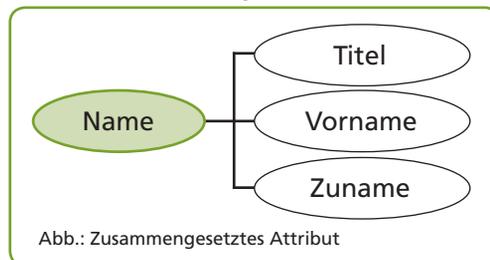


Abb.: Zusammengesetztes Attribut

Bei genauerer Betrachtung wird uns auffallen, dass das Attribut **Name** aus drei Teilen besteht: **Titel**, **Vorname** und **Zuname**.

Ein weiteres Beispiel ist die **Sozialversicherungsnummer**, die aus einem vierstelligen Code und dem Geburtsdatum besteht.

Auch **Schlüsselattribute** können zusammengesetzt sein, z.B. **Klasse** und **Katalognummer** der Entität **Schüler**.

Mr. What und Ms. Check

Ist die Katalognummer ein Schlüsselattribut für die Entität **Schüler**?

In einer Handelsakademie und Handelsschule lauten die Klassenbezeichnungen z.B. 3ak und 2bs. Handelt es sich hierbei um zusammengesetzte Attribute? Falls ja, warum?



Die Katalognummer allein genügt nicht, da in der Entität **Schüler** mehrere Klassen mit den gleichen Katalognummern vorkommen. Daher muss ein Schlüsselattribut für die Entität **Schüler** sowohl die Klasse als auch die Katalognummer enthalten. Es ist ein zusammengesetztes Schlüsselattribut.

Ja. Die Ziffer steht für den Jahrgang, der erste Buchstabe (a, b, c) für die Klasse und der zweite Buchstabe (k oder s) für die Schulform (Handelsakademie oder Handelsschule).



4 Schlüsselattribute

Identifikation für Exemplare

Ein Schlüsselattribut identifiziert ein Exemplar einer Entität oder Beziehung.

Primärschlüssel werden im ER-Modell unterstrichen dargestellt.

Arten von Schlüsselattributen

1 Ein Primärschlüssel identifiziert ein Exemplar einer Entität oder Beziehung eindeutig.

Die Entität **Schüler** hat z.B. folgende Exemplare:

Klasse	KatNr	Vorname	Zuname
<u>3as</u>	01	Klaus	Auer
<u>3as</u>	02	Robert	Bach
<u>3bs</u>	01	Julia	Adam

Der **Primärschlüssel** der Entität **Schüler** besteht aus der **Klasse** und der **Katalognummer**, z.B. „3as 01“, „3as 02“ und „3bs 01“.

Der Primärschlüssel muss eindeutig sein, um die Exemplare eindeutig identifizieren zu können. Daher erstellen wir bei der Anlage einer Tabelle in einer Datenbank für den Primärschlüssel einen **Index ohne Duplikate**. Dieser Index erlaubt nur eindeutige Schlüssel. Access und SQL-Server erledigen dies automatisch, sodass wir uns darum nicht explizit kümmern müssen.

Fremdschlüssel werden im ER-Modell üblicherweise nicht dargestellt.

2 Ein Fremdschlüssel verknüpft ein Exemplar mit dem Primärschlüssel eines anderen.

Der Primärschlüssel eines Lehrers ist seine Sozialversicherungsnummer. Jede Klasse hat einen Klassenvorstand, der ein Lehrer ist. Die Entität **Klassenvorstand** könnte dann z.B. folgende Exemplare aufweisen:

Klasse	Lehrer-Sozialversicherungsnummer
<u>3as</u>	1234-120368
<u>3bs</u>	3428-091159

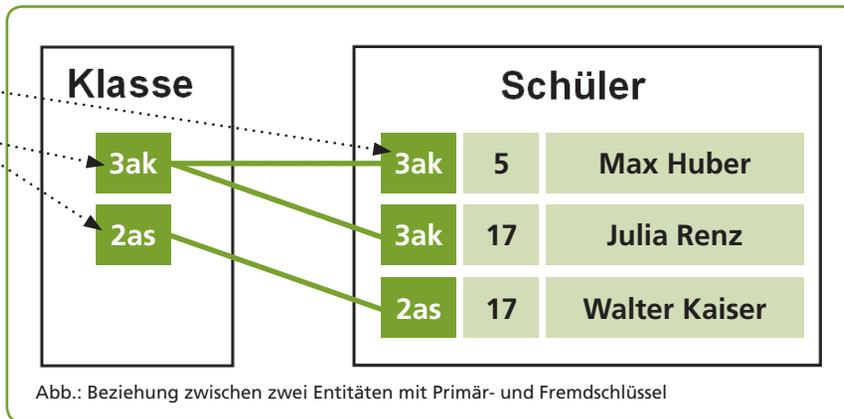
Die **Klasse** ist ein **Fremdschlüssel**, der die Exemplare von **Klassenvorstand** mit den **Schülerexemplaren** verknüpft.

Die **Lehrer-Sozialversicherungsnummer** ist der **Fremdschlüssel**, der die Exemplare von **Klassenvorstand** mit den **Lehrerexemplaren** verknüpft.

Über die **Verknüpfungen** ist es nun möglich, folgende Aussage zu formulieren: „Lehrer Dr. Heinrich Specht mit der Sozialversicherungsnummer 1234-120368 ist Klassenvorstand der Schüler Klaus Auer und Robert Bach.“ Diesen Zusammenhang kann eine **Abfrage** in Access bzw. eine **Sicht** in SQL-Server herstellen.

Die folgende Abbildung veranschaulicht das Zusammenwirken des **Primärschlüssels Klasse** der Entität **Klasse** und des **Fremdschlüssels Klasse** der Entität **Schüler**. Mehrere Schüler besuchen eine Klasse.

Fremdschlüssel
Primärschlüssel



SbX
Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.

Mr. What und Ms. Check

Wie lautet der Primärschlüssel der Entität *Klassenvorstand*?



Sofern ein Lehrer mehrere Klassen leitet, ist der Primärschlüssel die *Klasse* – sie muss eindeutig sein, denn zwei Klassenvorstände sind für eine Klasse nicht erlaubt. Leitet ein Lehrer nur eine Klasse, ist der Primärschlüssel ein zusammengesetztes Attribut aus *Klasse* und *Lehrer-Sozialversicherungsnummer*.

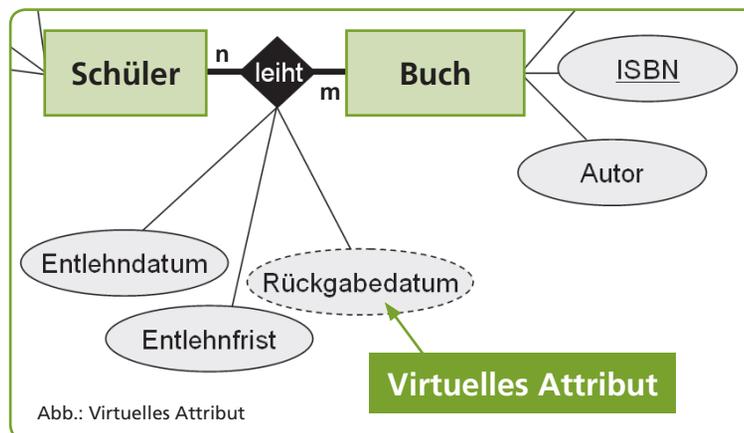


5 Virtuelle Attribute

Abgeleitete bzw. berechnete Attribute

Redundanz bedeutet, dass Daten mehrfach gespeichert werden. Virtuelle Attribute führen zu redundanten Daten.

Attribute, die aus anderen Attributen einer Entität abgeleitet bzw. berechnet werden können, werden **virtuelle Attribute** genannt. Sie werden strichliert in das ER-Modell eingezeichnet.



SbX
Eine Abbildung zu den virtuellen Attributen findest du in der PowerPoint-Präsentation unter der ID: 9908.

Anomalien können durch **Redundanz** entstehen.

Das **Rückgabedatum** kann aus den Attributen **Entlehndatum** und **Entlehnfrist** berechnet werden. Es ist ein **virtuelles Attribut**. Problematisch sind virtuelle Attribute deshalb, weil sie zu widersprüchlichen Daten führen können. Wir nennen dies eine **Anomalie**. Sie tritt z.B. auf, wenn das berechnete Rückgabedatum ein anderes Ergebnis liefert als das gespeicherte Datum.



Beachte

Virtuelle Attribute können zu **Anomalien** in einer Datenbank führen und sollten daher vermieden werden.

Auch das **Alter** einer **Person** ist ein **virtuelles Attribut**. Es resultiert aus der Differenz zwischen dem Tages- und dem Geburtsdatum. Das Alter zu speichern wäre falsch, da es sich ständig verändert. Wir müssen uns also gut überlegen, welche Attribute wir tatsächlich für unsere Datenbank verwenden.



L 1:

In einem Autohaus hat die Entität **Auto** folgende Exemplare:

FzNr	Marke	Type	max. Tankinhalt in l	Benzinvorrat in l	Verbrauch l/100 km	Reichweite in km
1	Audi	A4	65	50	6,2	806
2	BMW	530D	75	34	7,1	479
3	Kia	Sorento	70	66	8,9	742

Welches Attribut ist ein **virtuelles Attribut**?

Da sich bei jedem Autoexemplar die Reichweite durch das Fahren und Tanken ständig verändert, handelt es sich dabei um ein virtuelles Attribut. Die Reichweite ist von den Attributen **Benzinvorrat** und **Verbrauch** abhängig.



Ü 3:

Ein Elektrohändler möchte eine Datenbank zum Fakturieren der Aufträge erstellen. Folgende Entitäten wurden bereits definiert:

Kunde (KNr, Vorname, Zuname, Firma, Straße, PLZ, Ort, Telefon)

Rechnung (RNr, Datum, KNr)

Rechnungsposten (RNr, ANr, Menge, Verkaufspreis)

Artikel (ANr, Bezeichnung, Lagermenge, Einkaufspreis, Verkaufspreis)

In der Datenbank sollen nach Möglichkeit keine virtuellen Attribute vorkommen. Welche Gründe könnten dafür sprechen, dass das Attribut **Verkaufspreis** redundant gespeichert wird?

Mr. What und Ms. Check

Darf ich virtuelle Attribute in einer Datenbank speichern?



Generell solltest du das unterlassen, um Anomalien zu vermeiden. Es gibt aber Situationen, in denen Redundanz unbedingt erforderlich ist, da sonst falsche Ergebnisse entstehen könnten. Eine Preiserhöhung bei den Artikeln des Elektrohändlers aus Ü 3 darf die Preise auf den alten Rechnungen z.B. nicht beeinflussen.



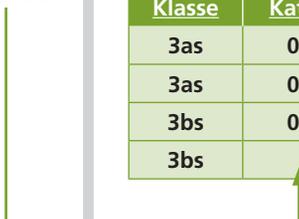
6 Fehlende Werte

Null ist nicht 0

Null (gesprochen „Nall“) bedeutet kein Wert.

Was passiert, wenn ein Attribut **keinen Wert** hat? Sehen wir uns dazu folgendes Beispiel an:

Klasse	KatNr	Vorname	Zuname
3as	01	Klaus	Auer
3as	02	Robert	Bach
3bs	01	Julia	Adam
3bs		Viola	Vogel



Die Schülerin **Viola Vogel, 3bs**, hat **keine Katalognummer**. Wie uns bereits bekannt ist, bilden die **Klasse** und **Katalognummer** gemeinsam den Primärschlüssel für die Entität **Schüler**. Was passiert mit unserer Verknüpfung, wenn es keine Katalognummer gibt?

Fehlende Werte

1 Wird für ein Attribut kein Wert eingegeben, ist der Wert des Attributs **Null**.

Die Katalognummer von Viola Vogel hat z.B. den Wert **Null**.

2 **Null** führt zu Fehlern, wenn es Teil einer Bedingung oder Berechnung ist.

Eine Bedingung oder Berechnung verlangt nach berechenbaren oder logisch auswertbaren Werten. Diese können numerisch oder alphanumerisch (Text) sein. Ist jedoch kein Wert vorhanden, so kann die Berechnung nicht durchgeführt werden. Die Folge ist ein Fehler.

In einer Abfrage soll z.B. das Alter aller Lehrer berechnet werden. Ist die Sozialversicherungsnummer bei einem Lehrer nicht vorhanden, z.B. weil das Geburtsdatum nicht eingegeben wurde, kann das Alter nicht berechnet werden. Die Abfrage funktioniert nicht.

3 Bei Schlüsselattributen ist **Null** unzulässig.

Eine Verknüpfung zwischen einem Primärschlüssel und einem Fremdschlüssel basiert auf einer Bedingung, nämlich der Gleichheit beider Schlüssel. Fehlt ein Schlüsselwert, so kann die Verknüpfung nicht aufgebaut werden. Das Ergebnis ist eine Anomalie. Schlüssel sind daher immer Pflichtfelder und dürfen niemals **Null** enthalten.

SbX
Eine Abbildung zu den fehlenden Werten findest du in der PowerPoint-Präsentation unter der ID: 9908.

Mr. What und Ms. Check

Wie kann ich **Null**-Werte in einer Tabelle verhindern?



In Microsoft Access gibt es für jedes Feld einer Tabelle die Eigenschaft „Eingabe erforderlich“. Wird sie aktiviert, sind **Null**-Werte unzulässig.

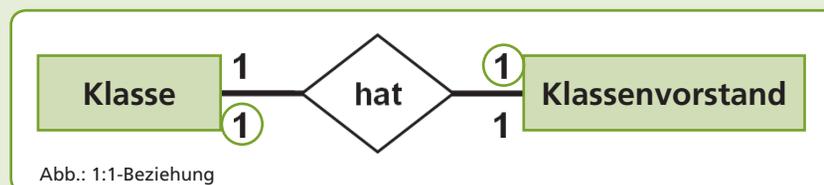


7 Kardinalität Beziehungstypen

Die Kardinalität oder auch Konnektivität gibt den Grad einer Beziehung an. Wir unterscheiden dabei drei Typen:

Grade von Beziehungen

1 1:1



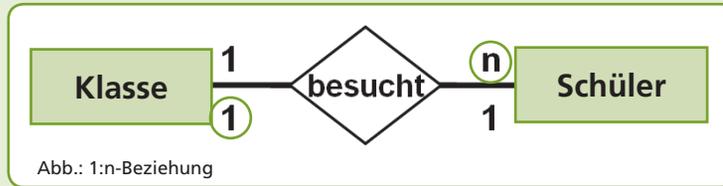
Eine Klasse hat wie viele Klassenvorstände? Einen.
Ein Klassenvorstand hat wie viele Klassen? Eine.

SbX
Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.

2 1:n

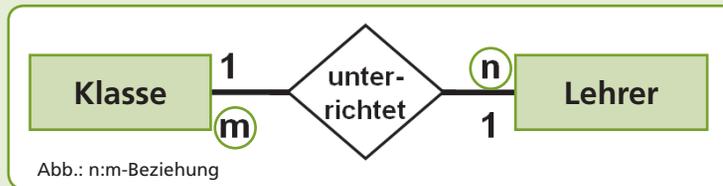


Eine Klasse hat wie viele Schüler? Mehrere.
Ein Schüler besucht wie viele Klassen? Eine.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9908.

3 n:m



Eine Klasse wird von wie vielen Lehrern unterrichtet? Von mehreren.
Ein Lehrer unterrichtet wie viele Klassen? Mehrere.

Mr. What und Ms. Check

Wie kann ich den Grad einer Beziehung feststellen?



Wir fragen, wie viele Instanzen der zweiten Entität (z.B. **Lehrer**) mit einer Instanz der ersten Entität (z.B. **Klasse**) verknüpft werden können: In einer Klasse unterrichten n Lehrer. Danach fragen wir umgekehrt. Ein Lehrer unterrichtet in n Klassen. Die größere Kardinalität bleibt auf beiden Seiten stehen: n Lehrer unterrichten m Klassen.



Üben

Übungsbeispiele

Ü 4:

In einer Schule gibt es folgende Entitäten:

Klasse (Klasse, Sozialversicherungsnummer)

Schüler (Klasse, Katalognummer, Name = Vorname + Zuname, Straße, PLZ, Ort)

Lehrer (Sozialversicherungsnummer, Name = Titel + Vorname + Zuname, Alter)

Fach (Fachkürzel, Fachbezeichnung)

Unterricht (Klasse, Sozialversicherungsnummer, Fachkürzel, Wochenstunden)

Eine Klasse besteht aus mehreren Schülern. Ein Schüler besucht eine Klasse. Ein Lehrer kann Klassenvorstand von maximal einer Klasse sein. Eine Klasse hat nur einen Lehrer, der auch Klassenvorstand ist. Ein Lehrer unterrichtet ein oder mehrere Fächer in einer Klasse. Eine Klasse hat mehrere Fächer und diese werden von mehreren Lehrern unterrichtet. Es kann auch vorkommen, dass ein Fach von mehreren Lehrern unterrichtet wird (z.B. Gruppenteilung in den Sprachen oder im EDV-Unterricht).

a) Nenne alle **Primär- und Fremdschlüssel**, die in diesem Beispiel vorkommen!

b) Nenne alle **zusammengesetzten Attribute**, die in diesem Beispiel vorkommen!

c) Nenne alle **virtuellen Attribute**, die in diesem Beispiel vorkommen!

d) Zeichne das **ER-Modell** mit den Beziehungen und Attributen in der Chen-Notation!

Ü 5:

In einer Großbäckerei werden folgende Entitäten vermutet:

Backware (BNr, Bezeichnung, Haltbarkeitsdauer, Verkaufspreis)

Produktion (PNr, PDatum, BNr, Menge, Ablaufdatum)

Auftrag (KNr, ADatum)

Kunde (KNr, Name, Adresse)

Ein Kunde erteilt pro Tag maximal einen Auftrag. Die von den Kunden beauftragten Backwaren werden an einem bestimmten Tag (Produktionsdatum) für die Produktion geplant. Das Ablaufdatum der produzierten Backwaren muss feststellbar sein. In einer Produktionscharge werden die Backwaren aus mehreren Kundenaufträgen zusammengefasst. Backwaren können in mehreren Produktionschargen vorkommen.

a) Unterstreiche die **Primärschlüssel** der Entitäten!

b) Nenne die **Fremdschlüssel** der Entitäten!

c) Nenne alle **Beziehungen** und ihre Kardinalität!

d) Entferne das **virtuelle Attribut** aus den Entitäten und nenne dessen Namen!

e) Zeichne das **ER-Modell** mit den Beziehungen und Attributen in der Chen-Notation!

Sichern

In dieser Lerneinheit haben wir uns mit dem Entity-Relationship-Modell und seinen Bestandteilen, den Entitäten, Beziehungen und Attributen, beschäftigt.

ER-Modell Das **Entity-Relationship-Modell (ERM)** wurde von Peter Chen zur **Modellierung relationaler Datenbanken** entwickelt. Es enthält **Entitäten, Beziehungen** mit ihrem Grad und **Attribute**. Zusätzlich werden zusammengesetzte Attribute, Primärschlüssel und virtuelle Attribute im ER-Modell eingezeichnet.

Entität Entitäten sind **eindeutig unterscheidbare Objekte**. Exemplare sind konkrete Ausprägungen einer Entität.

Beziehung Beziehungen sind **Assoziationen zwischen Entitäten**. Auch sie haben Exemplare.

Attribut Attribute sind **Eigenschaften von Entitäten oder Beziehungen**. Die Attribute einer Entität sind vom Primärschlüssel abhängig.

Zusammengesetztes Attribut Zusammengesetzte Attribute, wie z.B. **Name** oder **Sozialversicherungsnummer**, bestehen aus weiteren Attributen.

Schlüsselattribute Ein **Primärschlüssel** identifiziert ein Exemplar einer Entität oder Beziehung eindeutig. Ein **Fremdschlüssel** verknüpft ein Exemplar mit dem Primärschlüssel eines anderen.

Virtuelles Attribut Virtuelle Attribute sind **von anderen Attributen ableitbar**. Sie können zu **Anomalien** in einer Datenbank führen und sollten vermieden werden.

Null **Null** bedeutet **fehlender Wert** und ist für **Schlüsselattribute nicht zulässig**, da dies zu **Anomalien** führen könnte.

Kardinalität Die Kardinalität gibt den Grad einer Beziehung an: **1:1, 1:n oder n:m**.

 **SbX**
ID: 9910

Zusätzlich zu dieser Zusammenfassung findest du in SbX eine Bildschirmpräsentation.

Wissen

Wiederholungsfragen und -aufgaben

1. Nenne die wesentlichen Bestandteile eines ER-Modells!
2. Beschreibe ein Beispiel für eine Entität mit ihren Attributen!
3. Beschreibe ein Beispiel für ein Exemplar einer Entität mit seinen Attributen!
4. Erkläre den Unterschied zwischen Primär- und Fremdschlüssel!
5. Nenne ein Beispiel für ein zusammengesetztes Attribut!
6. Was versteht man unter einem virtuellen Attribut?
7. Warum kann der durchschnittliche Lagerbestand in einer Abfrage nicht berechnet werden, wenn in der Entität **Lager** bei einem Artikel der Lagerstand NULL ist?

8. Welche Grade von Beziehungen gibt es?
9. Gib die Kardinalität der folgenden Beziehungen in einer Pizzeria an!
 - a) Gast bestellt Pizza
 - b) Pizza enthält Zutat
 - c) Mitarbeiter produziert Pizza
 - d) Geschäftsführer ist Mitarbeiter



ID: 9911

Zusätzlich zu diesen Aufgaben findest du in SbX eine Internetaufgabe.

Lerncheck

Ich kann jetzt ...

- ... einige Datenmodelle aufzählen.
- ... die Bestandteile eines ER-Modells nennen.
- ... die Begriffe Entität, Beziehung, Exemplar und Attribut beschreiben.
- ... zusammengesetzte und virtuelle Attribute erkennen.
- ... das Zusammenspiel von Primär- und Fremdschlüssel erklären.
- ... die Konsequenzen der Verwendung von NULL und virtuellen Attributen absehen.
- ... die Kardinalität einer Beziehung richtig erkennen.

In der nächsten Lerneinheit wenden wir die gelernten Grundlagen an, indem wir ein ER-Modell zeichnen und in Tabellen auflösen. Außerdem beschäftigen wir uns mit der Indexierung und der referenziellen Integrität.

Lerneinheit 2

Relationale Auflösung

SbX

Alle SbX-Inhalte zu dieser Lerneinheit findest du unter der ID: 9912.

In dieser Lerneinheit erfahren wir, wie ein ER-Modell gezeichnet und in Tabellen aufgelöst wird. Wir beschäftigen uns mit

- der relationalen Auflösung,
- den Normalformen,
- der Indexierung sowie
- der referenziellen Integrität.

Lernen

1 ER-Modell zeichnen

Abbilden der Wirklichkeit

Für einen **Sushi-Lieferservice** soll ein ER-Modell erstellt werden. Wir überlegen uns zunächst, welche Entitäten benötigt werden:

1. Es gibt **Gäste**, die den Lieferservice in Anspruch nehmen.
2. Es gibt **Speisen**, z.B. Sushi, Sashimi, Maki, California Rolls, die von Gästen bestellt werden.
3. Die Speisen sind in die **Speisearten** Vorspeise, Hauptspeise und Dessert eingeteilt.

Schritt 1: Entitäten und Beziehungen

Wir erkennen die Entitäten **Gast**, **Speise** und **Speiseart**. Nun überlegen wir uns die **Kardinalitäten der Beziehungen** zwischen den Entitäten:

Ein **Gast** bestellt mehrere **Speisen**. Eine Speise kann von mehreren Gästen bestellt werden. Frau Gruber kann z.B. Maki und Sushi bestellen. Umgekehrt kann Sushi von Frau Gruber und Herrn Maier bestellt werden. Es handelt sich also um eine **n:m-Beziehung**.

Eine **Speiseart** gehört zu mehreren **Speisen**. Eine Speise gehört zu einer Speiseart. Zu den Vorspeisen gehören Sushi, Maki, Sashimi usw., aber Sushi ist auf jeden Fall eine Vorspeise. Es handelt sich um eine **1:n-Beziehung**.

Zwischen **Gast** und **Speiseart** gibt es keine sinnvolle Beziehung.

Nun können wir die Entitäten und ihre Beziehungen zeichnen:

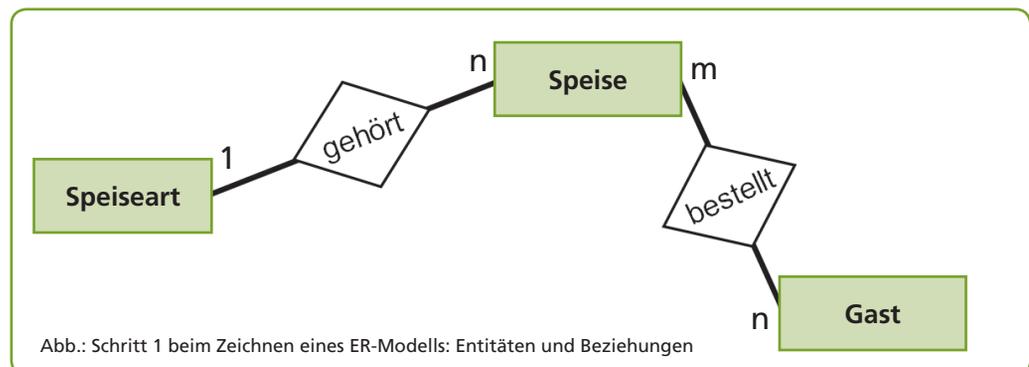


Abb.: Schritt 1 beim Zeichnen eines ER-Modells: Entitäten und Beziehungen

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Schritt 2: Attribute und Primärschlüssel

Nun überlegen wir uns die **Attribute** und **Primärschlüssel** der drei Entitäten:

Speiseart (Code, Bezeichnung)

Speise (Nr, Bezeichnung, Preis)

Gast-bestellt-Speise (Datum, Lieferzeit, Menge)

Gast (Nr, Name, Adresse)

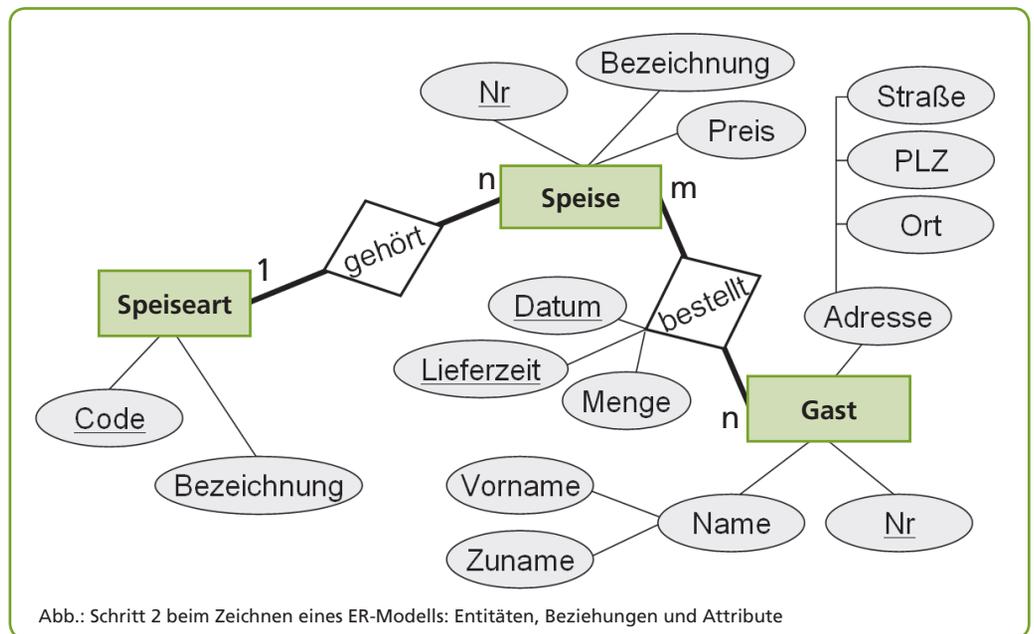
Als **Primärschlüssel** für **Speiseart**, **Speise** und **Gast** verwenden wir eine **eindeutige Nummer**.

Bei der Beziehung **Gast-bestellt-Speise** würde das Datum alleine dazu führen, dass ein Gast an einem Tag die gleiche Speise nur einmal bestellen könnte. Wenn ein Gast also nach einiger Zeit (am selben Tag) erneut die gleiche Speise bestellt, würde unsere Datenbank dies nicht zulassen. Daher verwenden wir hier einen **zusammengesetzten Primärschlüssel** aus **Datum** und **Lieferzeit**.

Die Entität **Gast** enthält die **zusammengesetzten Attribute Name** (= Vorname + Zuname) und **Adresse** (= Straße + PLZ + Ort).

Häufig wird das Einzeichnen der zusammengesetzten Attribute übergangen. Es werden dann sofort die atomaren Attribute eingezeichnet.

Ein atomares Attribut ist nicht mehr weiter zerlegbar.



SbX
Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Mr. What und Ms. Check

Warum wurden Maki und Sushi im ERM nicht eingezeichnet?



„Maki“ und „Sushi“ sind Exemplare der Entität **Speise**. Exemplare werden in ein ER-Modell nicht eingezeichnet. Daher wurden auch die Exemplare von **Speiseart**, „Vorspeise“, „Hauptspeise“ und „Dessert“ nicht eingezeichnet.



2 Relationale Auflösung

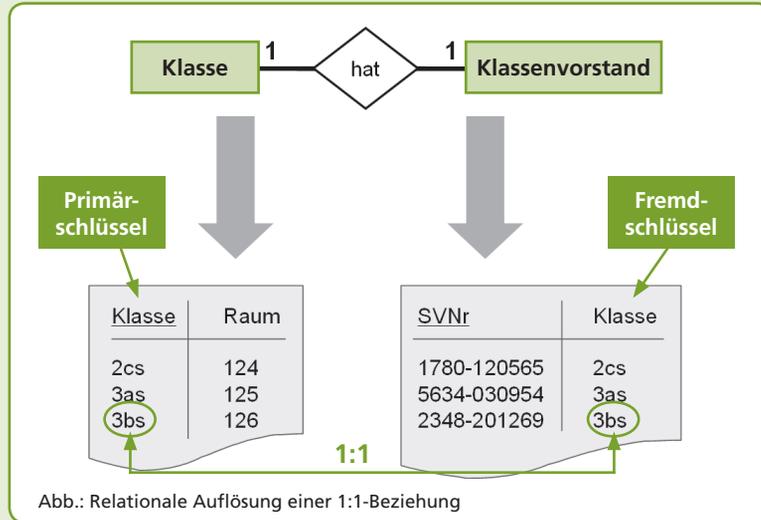
Entitäten werden zu Relationen

Relation bedeutet Tabelle.

Um das Entity-Relationship-Modell in eine Datenbank eingeben zu können, müssen wir die Entitäten in Relationen umwandeln. Unter der relationalen Auflösung verstehen wir die Umwandlung in Tabellen.

Beziehungen auflösen

1 Auflösung einer 1:1-Beziehung

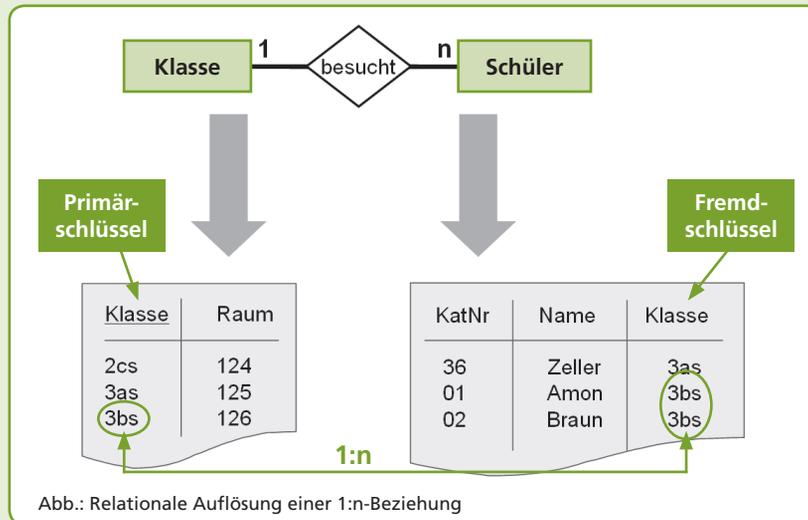


Die Entitäten **Klasse** und **Klassenvorstand** werden zu Tabellen. Die Attribute werden zu Feldern der Tabellen. Die Primärschlüssel werden als Primärschlüsselfelder in den Tabellen festgelegt (**Klasse** und **Sozialversicherungsnummer**).

Als Fremdschlüssel kann bei einer **1:1-Beziehung** einer der beiden Primärschlüssel in der jeweils anderen Tabelle verwendet werden. Hier wurde die Klasse als Fremdschlüssel in der Tabelle **Klassenvorstand** herangezogen.

Da es sich um eine **1:1-Beziehung** handelt, muss für die Klasse in der Tabelle **Klassenvorstand** ein **Index ohne Duplikate** angelegt werden. Dadurch wird verhindert, dass ein Lehrer in mehreren Klassen Klassenvorstand sein kann.

2 Auflösung einer 1:n-Beziehung



Die Entitäten **Klasse** und **Schüler** werden zu Tabellen. Die Attribute werden zu Feldern der Tabellen. Die Primärschlüssel werden als Primärschlüsselfelder in den Tabellen festgelegt (**Klasse** und **KatNr**).

Bei einer **1:n-Beziehung** wird der Primärschlüssel der 1-Tabelle als Fremdschlüssel der n-Tabelle festgelegt. Hier ist der Fremdschlüssel **Klasse** in der Tabelle **Schüler**.

Da es sich um eine **1:n-Beziehung** handelt, muss für **Klasse** in der Tabelle **Schüler** ein **Index mit Duplikaten** angelegt werden.

Relationale Transformation:
Entität → Relation
1:1-Beziehung bleibt
Attribut → Feld

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Der Index bestimmt den Grad der Beziehung in einer Datenbank.

Relationale Transformation:
Entität → Relation
1:n-Beziehung bleibt
Attribut → Feld

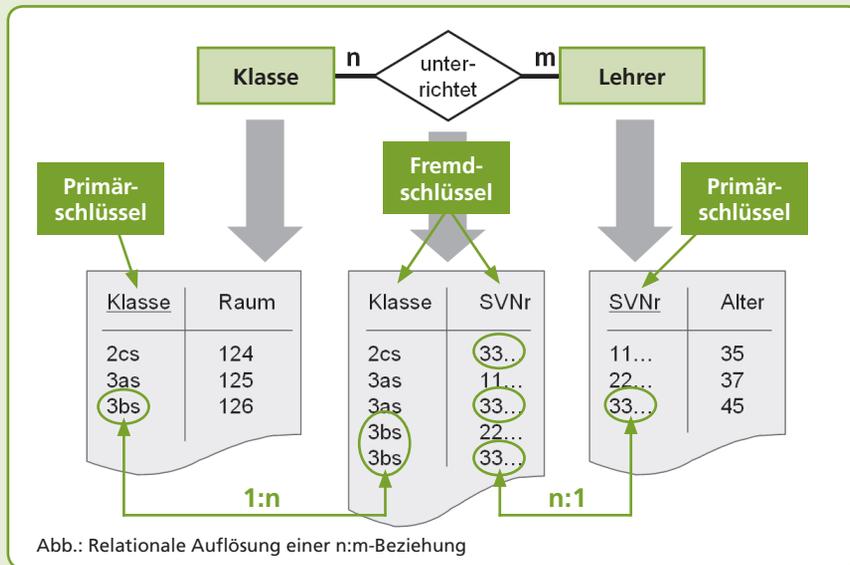
SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Bei einem **Schlüsselfeld** sollte aus Geschwindigkeitsgründen immer ein **Index** eingesetzt werden.

3 Auflösung einer n:m-Beziehung

Relationale Transformation:
 Entität -> Relation
 n:m-Beziehung -> Beziehungrelation
 Attribut -> Feld



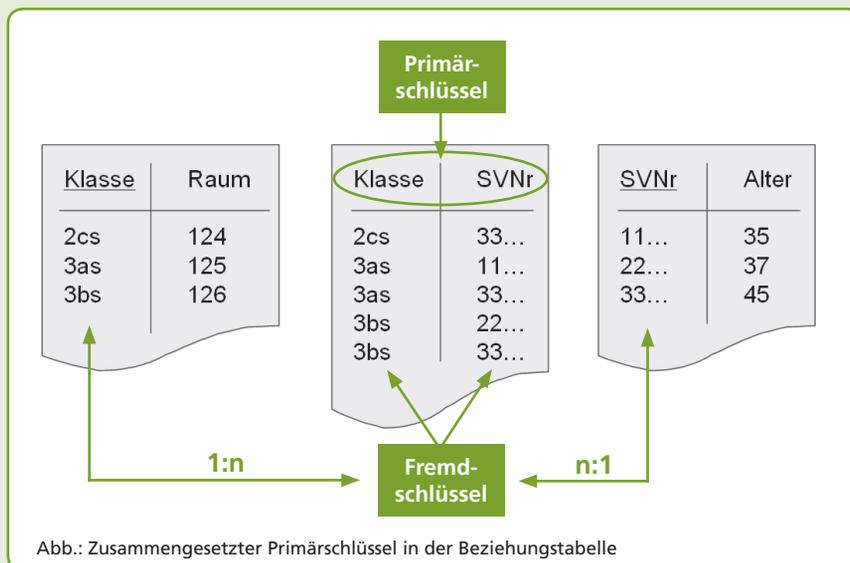
SbX
 Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Eine Datenbank erlaubt nur 1:1- und 1:n-Beziehungen zwischen Tabellen. Daher müssen wir eine **n:m-Beziehung in zwei 1:n-Beziehungen auflösen**:

Die Entitäten werden zunächst in Tabellen aufgelöst. Ihre Attribute werden zu Feldern der beiden Tabellen. Die Primärschlüssel sind die Schlüsselattribute, die immer einen **Index ohne Duplikate** erhalten. Hier sind das **Klasse** und **Lehrer**.

Im nächsten Schritt erstellen wir aus der Beziehung eine neue Tabelle: **die Beziehungstabelle**. Sie bekommt den Namen **KlasseUnterrichtetLehrer**, also eine Kombination aus den beiden Tabellen, die sie verknüpft, und dem Namen der Beziehung. In dieser Tabelle legen wir als Fremdschlüssel die beiden Primärschlüssel aus **Klasse** und **Lehrer** an. Die Fremdschlüssel repräsentieren die n-Teile (mehrere gleiche Schlüssel), daher erstellen wir einen **Index mit Duplikaten**.

Die **Beziehungstabelle** enthält in unserem Beispiel die beiden Fremdschlüssel **Klasse** und **Lehrer**. Aber wo befindet sich der Primärschlüssel?



Die **Beziehungstabelle** erhält immer die **n-Teile** der Beziehungen.

SbX
 Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

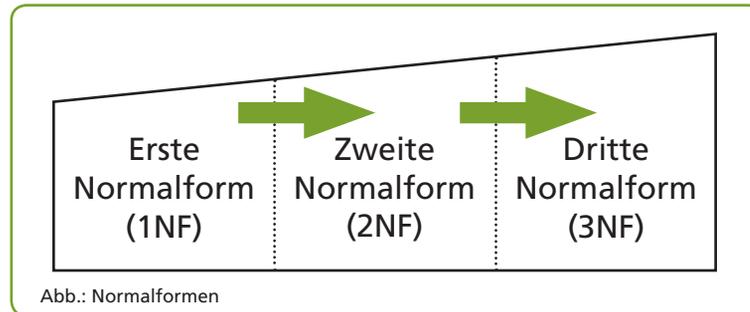
Der **Primärschlüssel** ist eine Kombination der beiden Fremdschlüssel, also ein **zusammengesetzter Schlüssel**. Da ein Primärschlüssel keine Duplikate erlaubt, die beiden Fremdschlüssel für sich betrachtet jedoch sehr wohl, ergibt sich daraus Folgendes: Ein Lehrer darf in einer Klasse nicht mehrfach angelegt werden. Da dies aber durchaus der Fall sein kann, nämlich wenn ein Lehrer unterschiedliche Fächer in einer Klasse unterrichtet, müssen wir das Fach noch zusätzlich in den Primärschlüssel einbeziehen.

3 Normalisierung Optimierung der Datenstrukturen

Im Zuge der Normalisierung wenden wir bestimmte Kriterien auf unsere Datenbank an, damit sie einen festgelegten Zustand erreicht, der Inkonsistenzen (Unstimmigkeiten) vermeidet. Dieser festgelegte Zustand ist eine **Normalform**.

Wir bringen unsere Datenbank nun Schritt für Schritt durch die Anwendung bestimmter Regeln von der ersten bis in die dritte Normalform. Danach gäbe es noch weitere Normalformen, z.B. die Boyce-Codd-Normalform, die vierte und die fünfte Normalform, die wir aber nicht behandeln. In der Praxis sollte jede Datenbank mindestens der dritten Normalform (3NF) entsprechen.

1NF = 1. Normalform
2NF = 2. Normalform
3NF = 3. Normalform



SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9913.

Beachte

Die wichtigsten **Gründe für das Normalisieren** einer Datenbank sind:

1. Vermeidung von Redundanz (doppelte Einträge)
2. Vermeidung von Anomalien (widersprüchliche Daten)
3. Konsistenz (Vollständigkeit und Richtigkeit)
4. Vereinfachung der Wartung

Mr. What und Ms. Check

Wie kann ich eine Normalisierung meiner Datenbank durchführen?



Indem du das ER-Modell verwendest, die relationale Auflösung durchführst und danach die Regeln der Normalformen anwendest.



Relationen-Notation

Für Entitäten, Beziehungen und Attribute haben wir bisher die grafische Darstellung des ER-Modells benutzt. Bei der relationalen Auflösung wird die Relationenschreibweise verwendet. Die Entitäten werden nun als Relationen bezeichnet.



L 1:

Die relationale Auflösung der Entitäten **Klassenvorstand**, **Klasse**, **Schüler** und **Lehrer** lautet:

$R_{\text{Klassenvorstand}} = \text{Klassenvorstand}$ (Klasse, SVNr)

$R_{\text{Klasse}} = \text{Klasse}$ (Klasse, Raum)

$R_{\text{Schüler}} = \text{Schüler}$ (Klasse, KatNr, Vorname, Zuname)

$R_{\text{Lehrer}} = \text{Lehrer}$ (SVNr, Titel, Vorname, Zuname)

$R_{\text{Unterricht}} = \text{unterrichtet}$ (SVNr, Klasse, Fach, Wochenstunden)

Im Rahmen der Normalisierung werden die Normalformen verwendet, um die Datenbank in einen **normalisierten Zustand** zu versetzen.

4 Normalformen

Zustände einer Datenbank

Die wichtigsten Normalformen sind die 1NF, 2NF und 3NF.

1 Erste Normalform: Jedes Attribut einer Tabelle ist unteilbar.

Eine Information in einem Attribut ist dann unteilbar, wenn sie nicht weiter in Einzelinformationen zerlegt werden kann. Als Attributwerte sind keine Aufzählungen oder Listen erlaubt.

Schüler (KatNr, Name)

→ Schüler (KatNr, Vorname, Zuname)

Abb.: Erste Normalform (1NF)

In diesem Beispiel muss das Feld **Name** in seine Bestandteile, z.B. Vorname und Zuname, aufgeteilt werden.

2 Zweite Normalform: 1NF und alle Attribute müssen vom gleichen Primärschlüssel abhängen.

Schüler (KatNr, Vorname, Zuname)

→ Schüler (Klasse, KatNr, Vorname, Zuname)

Abb.: Zweite Normalform (2NF)

In diesem Beispiel ist der Primärschlüssel **Katalognummer** für alle Schülernamen der unterschiedlichen Klassen nicht ausreichend, da es z.B. die Nummer 1 in jeder Klasse gibt. Die Klasse muss als weiteres Schlüsselfeld hinzugefügt werden.

3 Dritte Normalform: 2NF und es darf keine transitiven Abhängigkeiten geben.

Schüler (Klasse, KatNr, Vorname, Zuname, Klassenvorstand)

→ Schüler (Klasse, KatNr, Vorname, Zuname)
→ Klassenvorstand (Klasse, LehrerNr)

Abb.: Dritte Normalform (3NF)

In dem Beispiel ist der Klassenvorstand nicht dem Schüler, sondern der Klasse zuzuordnen. Daher muss für diese Abhängigkeit eine eigene Tabelle erstellt werden. Im Falle einer Änderung des Klassenvorstands muss nicht jeder Schüler aktualisiert werden. Es genügt, den neuen Lehrer in der Tabelle **Klassenvorstand** der Klasse zuzuordnen.

Mr. What und
Ms. Check

Muss eine Datenbank immer in die 3NF gebracht werden?



Generell ja. Es gibt aber Fälle, wo dies nicht sinnvoll ist, wie z.B. die redundante Speicherung von Preisen bei Artikeln und Rechnungsposten.



Im nächsten Schritt kann die Datenbank z.B. in Microsoft Access oder SQL-Server angelegt werden. Sehen wir uns an, welche besondere Rolle dabei der Indexierung von Fremdschlüsseln zukommt.

5 Indexierung Kardinalität von Fremdschlüsseln

Wir haben das ER-Modell der Schule in Relationen aufgelöst und diese in die dritte Normalform gebracht. Folgendes Ergebnis liegt vor:

Klassenvorstand (Klasse, SVNr)

Klasse (Klasse, Raum)

Lehrer (SVNr, Titel, Vorname, Zuname)

Schueler (Klasse, KatNr, Vorname, Zuname, Straße, PLZ, Ort)

Unterricht (Klasse, SVNr, FachKürzel)

Fach (FachKürzel, FachBezeichnung)

Wir starten Microsoft Access, erstellen eine neue Datenbank mit dem Namen **Schule** und legen die Tabellen an.

Nun legen wir die Beziehungen zwischen den Tabellen fest. Aber wie erhalten wir eine 1:1-Beziehung zwischen **Klassenvorstand** und **Klasse** bzw. **Klassenvorstand** und **Lehrer**?

Ein **Index** ist eine sortierte Liste eines Tabellenfeldes. Er wird dort eingesetzt, wo oft nach Werten gesucht, sortiert oder gefiltert wird, wie z.B. bei den **Schlüsselfeldern**.

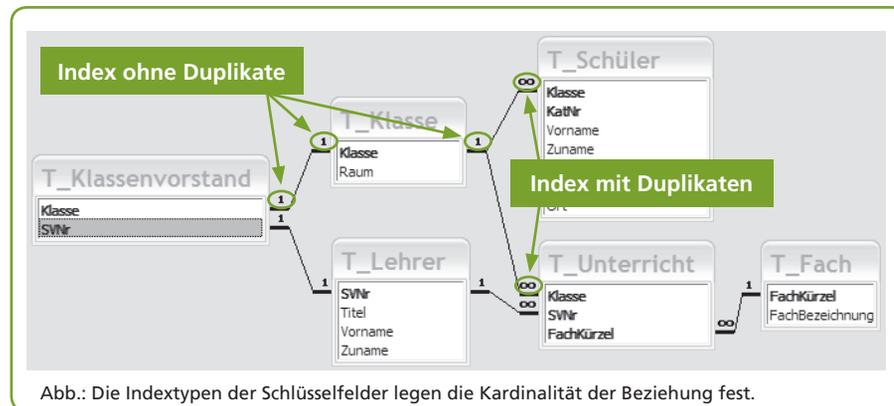


Abb.: Die Indextypen der Schlüsselfelder legen die Kardinalität der Beziehung fest.

Das Feld **Klasse** ist für die Tabelle **Klasse** der **Primärschlüssel** und erhält automatisch einen **Index ohne Duplikate**, weshalb dieses Feld eindeutig ist. Dies trifft auch auf das Feld **SVNr** bei der Tabelle **Lehrer** zu.

In der Tabelle **Klassenvorstand** sind die Felder **Klasse** und **SVNr** der **Fremdschlüssel**. Nur beide Felder gemeinsam bilden den Primärschlüssel. Da **Klasse** und **SVNr** keinen Index ohne Duplikate erhalten, können hier Werte mehrfach vorkommen. Wir erhalten 1:n-Beziehungen.

Beide Beziehungen sollten aber **1:1-Beziehungen** sein. Diese erhalten wir, indem wir für beide **Fremdschlüssel** einen **Index ohne Duplikate** verwenden.

Indextypen des Fremdschlüssels

1 Ein Index ohne Duplikate ergibt eine 1:1-Beziehung.

Die Beziehung zwischen **Klassenvorstand** und **Lehrer** ist eine 1:1-Beziehung, weil der Fremdschlüssel einen Index ohne Duplikate erhält.

2 Ein Index mit Duplikaten ergibt eine 1:n-Beziehung.

Im Feld **Klasse** der Tabelle **Schüler** wurde ein Index mit Duplikaten (bzw. kein Index) festgelegt. Da das Feld **Klasse** in der Tabelle **Klasse** ein Primärschlüssel mit einem Index ohne Duplikate ist, ergibt sich hieraus eine 1:n-Beziehung.

Aus Gründen der **Performance** sollte jeder **Schlüssel** einen **Index** haben.

Beachte

Bei Schlüsselfeldern legt der **Indextyp** die **Kardinalität der Beziehung** fest.

Damit die Beziehungen einer Datenbank immer einwandfrei funktionieren, müssen wir die referenzielle Integrität verwenden.

6 Referenzielle Integrität

Vermeidung von Anomalien

Eine Datenbank soll Fehler von sich aus so weit wie möglich vermeiden – sie soll **konsistent** sein. **Indexierung und referenzielle Integrität** tragen zur **Konsistenz** bei.

1 Die referenzielle Integrität sorgt für die Korrektheit der verwendeten Schlüssel.

Nehmen wir an, jemand würde in der Tabelle **Klassenvorstand** eine SV-Nummer eingeben, zu der es keinen Lehrer gibt. Oder jemand würde in der Tabelle **Schüler** eine Klasse eingeben, die nicht existiert. Die Regeln der referenziellen Integrität verbieten solche Falscheingaben. Eine typische Fehlermeldung von Access würde beispielsweise so aussehen:

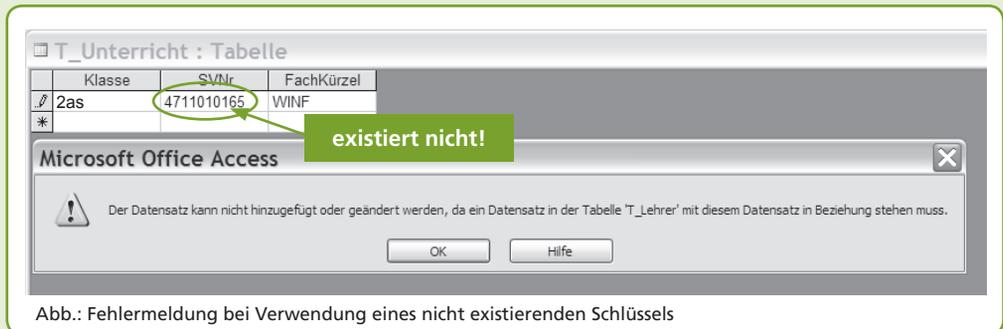


Abb.: Fehlermeldung bei Verwendung eines nicht existierenden Schlüssels

2 Ein Index ohne Duplikate verbietet Mehrfacheingaben, wo diese nicht sinnvoll oder nicht erlaubt sind.

Wenn jemand in der Tabelle **Klassenvorstand** eine Klasse irrtümlich zweimal eingibt, antwortet Access mit folgender Fehlermeldung:

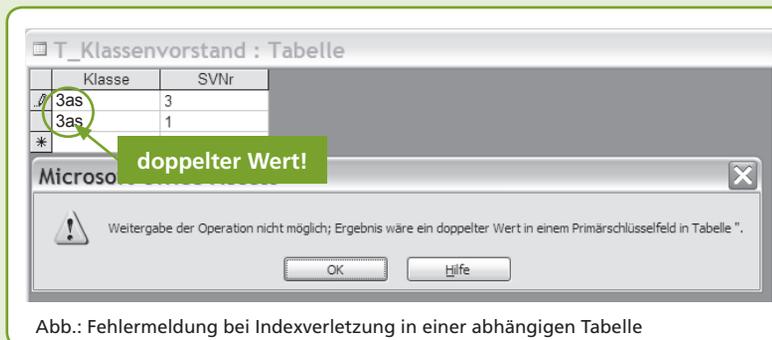


Abb.: Fehlermeldung bei Indexverletzung in einer abhängigen Tabelle

Beachte

Die **referenzielle Integrität** sorgt für eine **konsistente Datenbank**.

Aktualisierungs- und Löscheintrag

Die Aktualisierungseintrag **vermeidet Änderungsanomalien**. Der Name der Klasse **2as** soll z.B. in **3as** geändert werden. Diese Änderung wird in der Tabelle **Klasse** vorgenommen. Da aber die Klasse ein häufig verwendeter Schlüssel ist, muss diese Änderung in allen anderen betroffenen Tabellen ebenfalls stattfinden, nämlich in **Klassenvorstand**, **Schüler** und **Unterricht**. Access erledigt diese Aufgabe im Rahmen der Aktualisierungseintrag automatisch.

Was passiert mit den Schülerinnen und Schülern einer Klasse, wenn die Klasse gelöscht wird? Wenn wir die Löscheintrag nicht aktiviert haben, tritt eine **Löschanomalie** auf – eine Art „Datenleiche“. Die Löscheintrag bewirkt, dass in allen untergeordneten Tabellen, den zugehörigen n-Fremdschlüsseln, die Daten ebenfalls gelöscht werden.

Vorsicht: Unbeabsichtigtes Löschen kann weitreichende Folgen haben!

Die Aktualisierungseintrag und Löscheintrag sind nur bei aktivierter referenzieller Integrität verfügbar.



Animierte Folien zur referenziellen Integrität findest du in der PowerPoint-Präsentation unter der ID: 9913.

Üben

Übungsbeispiele

Ü 1:

Gabi will ihre DVD-Sammlung in einer Datenbank verwalten:

1. Auf jeder DVD befindet sich ein Film. Filme kommen nicht doppelt vor.
2. Zu jedem Film sollen Titel, Laufzeit, Sprache, Genre, DolbyDigital (ja/nein), die Schauspieler und der Regisseur gespeichert werden.
3. In einem Film spielen mehrere Schauspieler, ein Schauspieler spielt in mehreren Filmen.
4. Jeder Film hat einen Regisseur, ein Regisseur führt in mehreren Filmen Regie.
5. Jeder Film ist einem Genre zugeordnet. Für jedes Genre gibt es mehrere Filme.

Erledige folgende Aufgabenstellungen:

- a) Zeichne das ER-Modell für die DVD-Sammlung mit allen Entitäten, Beziehungen und Attributen in Chen-Notation.

- b) Führe die relationale Auflösung des ER-Modells durch und gib das Ergebnis in Relationen-Notation in der dritten Normalform an.

- c) Erstelle die Datenbank in Access.

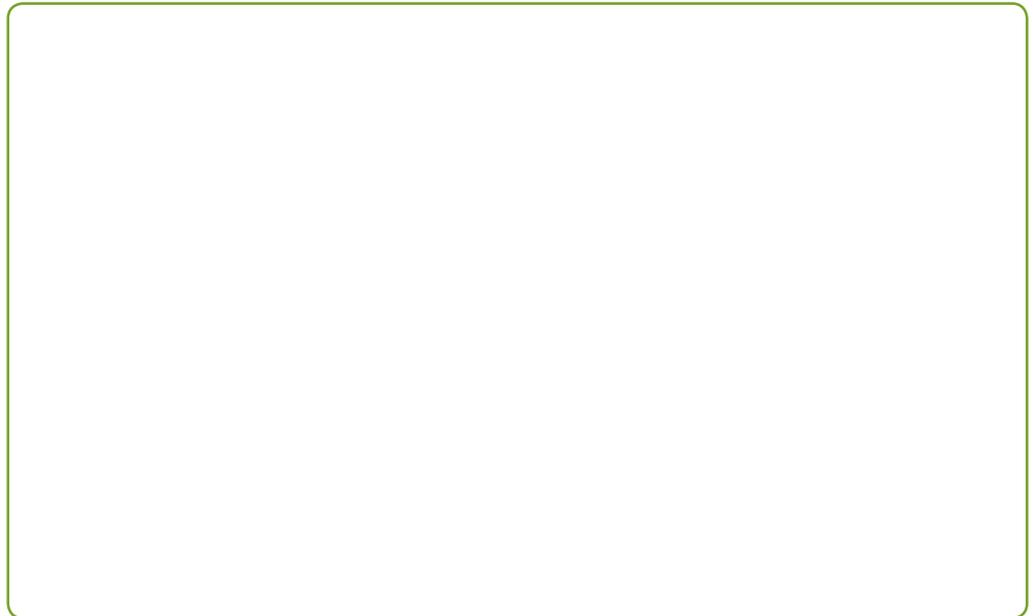
Ü 2:

Die Firma Univent möchte ihre Auftragsverwaltung in einer neuen Datenbank speichern:

1. Es gibt drei Kundengruppen: Privatkunden, Großkunden und Händler.
2. Jeder Kunde kann mehrere Aufträge erteilen. Ein Auftrag ist immer einem bestimmten Kunden zugeordnet.
3. Mit einem Auftrag werden mehrere Artikel bestellt. Jeder Artikel kann in mehreren Aufträgen bestellt werden.
4. Jeder Artikel ist einer Artikelgruppe zugeteilt. Eine Artikelgruppe besteht aus mehreren Artikeln.
5. Ein Auftrag wird in eine Rechnung übernommen. Auf einer Rechnung können mehrere Aufträge verrechnet werden.
6. Ein Kunde zahlt die Rechnung unter Umständen in Raten. Es kann mehrere Zahlungen zu einer Rechnung geben. Eine Zahlung kann sich auch auf mehrere Rechnungen beziehen.

Erledige folgende Aufgabenstellungen:

- a) Zeichne das ER-Modell für die Auftragsverwaltung mit allen Entitäten, Beziehungen und Attributen in Chen-Notation.



- b) Führe die relationale Auflösung des ER-Modells durch und beschreibe das Ergebnis in Relationen-Notation in der dritten Normalform.

- c) Erstelle die Datenbank in Access.

Ü 3:

Der Besitzer einer Videothek beschreibt sein Geschäftsmodell wie folgt:

1. Die Videothek verleiht Filme an Kunden. Jeder Kunde kann beliebig viele Filme ausleihen. Ein Film kann von vielen Kunden ausgeliehen werden. Ein Film kann auch mehrmals pro Tag verliehen werden.
2. Von den Kunden werden Kundennummer, Name und Adresse gespeichert. Jeder Kunde erhält einen Pin-Code, über den er/sie seine Bestellungen genehmigt.
3. Es gibt Filme auf Video und DVD. Zu jedem Film werden Filmtitel, Spieldauer und der Name des Verleihers gespeichert.
4. Jeder Film ist einer Kategorie zugeordnet. Kategorien sind z.B. Science Fiction, Heimatfilm, Kriegsfilm, Komödie etc.
5. Es gibt drei Preistarife: Aktion, Standard, Aktuell. Jedem Film ist ein Tarif mit einem Preis zugeordnet. Ein Tarif kann für mehrere Filme gelten.
6. In jedem Film spielen mehrere Schauspieler mit, ein Schauspieler spielt in mehreren Filmen.
7. Wenn für Filme die Nachfrage groß ist, werden mehrere Exemplare derselben DVD oder desselben Videos eingelagert. Jede Kassette erhält eine eindeutige Kassettensnummer. Zu einem Film kann es mehrere Kassetten geben (z.B. acht DVD- und drei Videokassetten zu „Lord of the Rings – Return of the King“).

Zeichne das ER-Modell für die DVD-Sammlung mit allen Entitäten, Beziehungen und Attributen in Chen-Notation und führe die relationale Auflösung bis zur 3NF durch.

 SbX
 ID: 9914

Zusätzlich zu diesen Übungen findest du in SbX eine Internetaufgabe.

Sichern

In dieser Lerneinheit haben wir uns mit der relationalen Auflösung von ER-Modellen, mit den Normalformen sowie der Bedeutung von Indexierung und referenzieller Integrität für die Konsistenz einer Datenbank beschäftigt.

Relationale Auflösung	Bei der relationalen Auflösung wird aus jeder Entität eine Relation . Zusätzlich muss eine n:m-Beziehung in eine Beziehungsrelation umgewandelt werden.
1. Normalform	Eine Tabelle ist in der ersten Normalform (1NF) , wenn jedes Attribut unteilbar ist.
2. Normalform	Die zweite Normalform (2NF) liegt vor, wenn die Tabelle in der ersten Normalform ist und alle Attribute vom gleichen Primärschlüssel abhängig sind.
3. Normalform	Eine Datenbank ist in der dritten Normalform (3NF) , wenn sie in der zweiten Normalform ist und keine transitiven Abhängigkeiten vorliegen.
Indextyp	Der Indextyp der Schlüsselfelder legt die Kardinalität der Beziehung zwischen den Tabellen fest.
Referenzielle Integrität	Die referenzielle Integrität sorgt für eine konsistente Datenbank . Durch Aktualisierungs- und Löschanomalien werden Änderungs- und Löschanomalien verhindert.

 SbX
 ID: 9915

Zusätzlich zu dieser Zusammenfassung findest du in SbX eine Bildschirmpräsentation.

Wissen

Wiederholungsfragen und -aufgaben

1. Was ist eine Beziehungsrelation?
2. Warum muss eine n:m-Beziehung aufgelöst werden?
3. Erkläre die Unterschiede zwischen 1NF, 2NF und 3NF!
4. Was ist ein Index?
5. Welche Bedeutung hat der Indextyp für die Kardinalität einer Beziehung?
6. Was versteht man unter einer Anomalie?
7. Wie kann die Konsistenz einer Datenbank hergestellt werden?
8. Welche Aufgaben haben die Aktualisierungs- und die Löschweitergabe?
9. Zwischen den Tabellen **Artikelgruppe** und **Artikel** gibt es eine 1:n-Beziehung, ebenso zwischen **Artikel** und **Rechnungsposten**. Welche Auswirkung hat das Löschen einer Artikelgruppe auf die Artikel und die Rechnungsposten, wenn die Löschweitergabe für beide Beziehungen aktiviert wurde? Begründe deine Antwort!

 ID: 9916

Zusätzlich zu diesen Aufgaben findest du in SbX eine Internetaufgabe.

Lerncheck

Ich kann jetzt ...

- ... ein ER-Modell eigenständig zeichnen.
- ... ein ER-Modell in Relationen auflösen.
- ... Relationen in die erste, zweite und dritte Normalform bringen.
- ... die Kardinalität einer Beziehung mit Hilfe des Indextyps festlegen.
- ... die referenzielle Integrität zur Vermeidung von Anomalien anwenden.
- ... die Funktionsweise der Aktualisierungs- und Löschweitergabe erklären.

In der nächsten Lerneinheit erstellen wir mit Access eine neue Datenbank und beschäftigen uns eingehender mit Schlüsselfeldern sowie Beziehungen.

Lerneinheit 3

Erstellen einer Datenbank

SbX

Alle SbX-Inhalte zu dieser Lerneinheit findest du unter der ID: 9917.

In dieser Lerneinheit beschäftigen wir uns mit der Strukturierung und Erstellung von Tabellen in einer Datenbank. Wir verwenden Primärschlüssel zur Identifikation von Datensätzen und Fremdschlüssel zur Herstellung von Beziehungen.

Wir beschäftigen uns mit

- der Erstellung von Tabellen und Beziehungen in einer neuen Datenbank,
- der Definition von Primär- und Fremdschlüsseln,
- der Festlegung von Beziehungen zwischen Tabellen sowie
- den verschiedenen Verknüpfungstypen von Beziehungen.

Lernen

1 Datenbankschema

Tabellen und Beziehungen erstellen

Die **Tabellen und Beziehungen** einer Datenbank werden als **Datenbankschema** bezeichnet. Zur Erstellung einer Datenbank müssen diese identifiziert und geplant werden.

Ein **Tupel** enthält die zusammengehörige Information in einer Zeile einer Tabelle oder Abfrage.

Eine **Einfachinformation** ist vom Primärschlüssel *IDSong* eindeutig abhängig, z.B. der Titel.

Eine **Mehrfachinformation** ist mehreren Primärschlüsseln zuzuordnen, z.B. die Musikrichtung.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

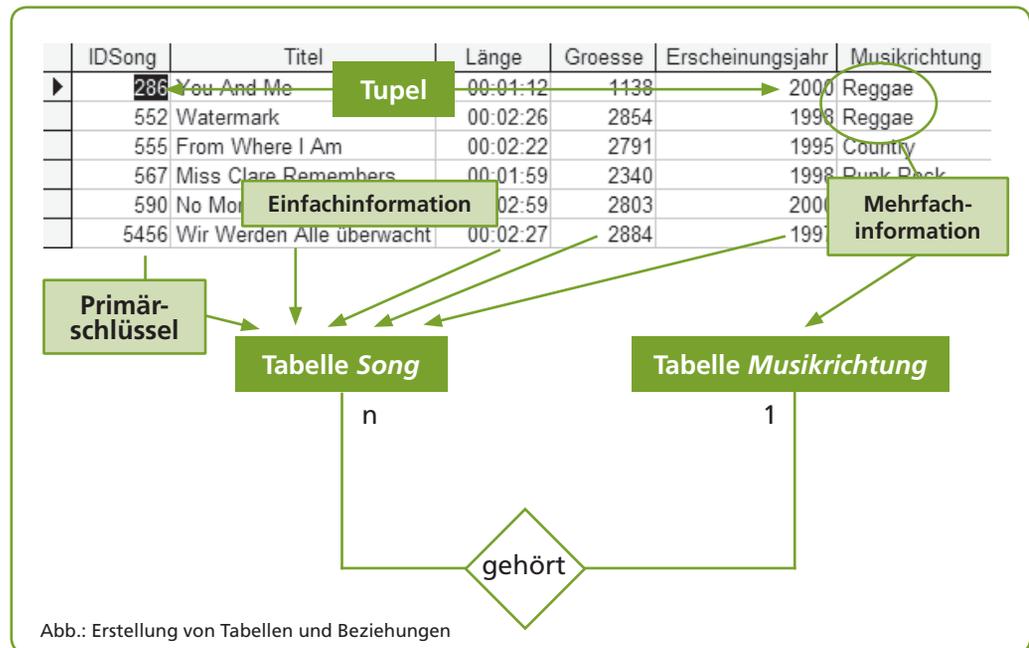


Abb.: Erstellung von Tabellen und Beziehungen

Einfachinformationen sind 1:1-Beziehungen und werden als **Felder einer Tabelle** gespeichert. Für **Mehrfachinformationen** werden **separate Tabellen** erstellt, die mit einer **1:n-Beziehung** verknüpft werden.

Ü 1:

Eine Gärtnerei möchte eine Datenbank zur Verwaltung der Kundenbestellungen erstellen. Ein Kunde bestellt pro Auftrag verschiedene Blumen in unterschiedlichen Stückzahlen. Welche Tabellen und Beziehungen planst du für die Datenbank?

Tupel = Datensatz

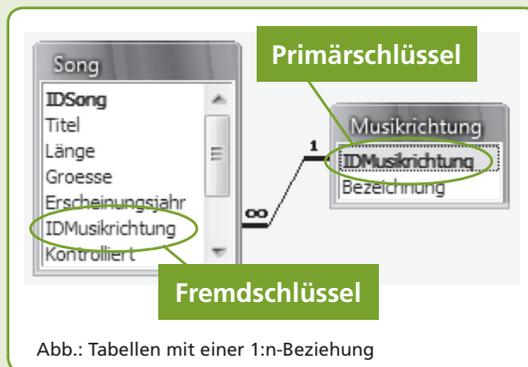
1 In einer relationalen Datenbank darf es nur 1:1- und 1:n-Beziehungen geben.

Einfachinformationen werden in den Feldern einer Tabelle gespeichert, die über **1:1-Beziehungen** miteinander verknüpft sind und einen gemeinsamen Primärschlüssel, z.B. **IDSong**, haben. **Mehrfachinformationen** werden **in unterschiedlichen Tabellen** gespeichert, die über **1:n-Beziehungen** miteinander verknüpft sind. Die Zuordnung der verknüpften Tabellen erfolgt über den Primär- und den Fremdschlüssel.

2 Jedes Tupel einer Tabelle erhält einen Primärschlüssel, der es eindeutig identifiziert.

Der **Primärschlüssel**, z.B. **IDSong**, verhindert Verwechslungen aufgrund mehrfach vorkommender Feldinhalte, wie z.B. beim Titel oder bei der Größe. Über den Primärschlüssel kann jeder Song eindeutig identifiziert werden.

3 Bei einer 1:n-Beziehung zwischen zwei Tabellen wird der Primärschlüssel der 1-Tabelle als Fremdschlüssel in der n-Tabelle verwendet.



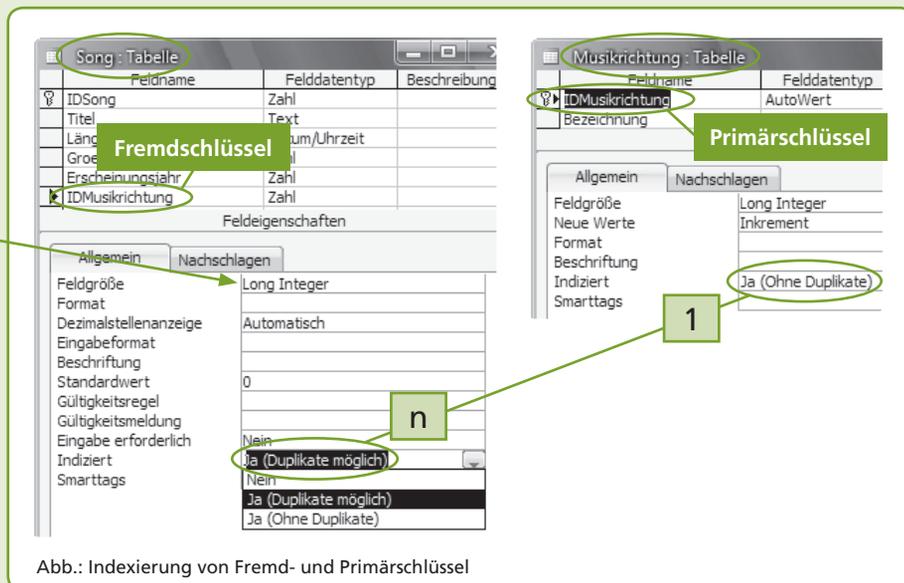
Der Primärschlüssel **IDMusikrichtung** der Tabelle **Musikrichtung** wird als **Fremdschlüssel** in der Tabelle **Song** definiert. Der Fremdschlüssel muss **den gleichen Datentyp** wie der Primärschlüssel haben.

4 Für jedes Schlüsselfeld sollte ein Index definiert werden, um die Performance der Datenbank zu verbessern.

Access legt für einen **Primärschlüssel** automatisch einen **Index ohne Duplikate** fest, für den **Fremdschlüssel** können wir den Index wählen:

- Index mit Duplikaten** → 1:n-Beziehung
- Index ohne Duplikate** → 1:1-Beziehung

Der Datentyp **Autowert** von **IDMusikrichtung** in der Tabelle **Musikrichtung** ist **Long Integer**, ebenso **IDMusikrichtung** in der Tabelle **Song**.





L 1:

Erstelle eine neue Datenbank für eine Autowerkstatt: Jedes Auto hat eine Marke, z.B. Audi, VW, und einen Typ, z.B. A4, A6, Golf, sowie eine Fahrgestellnummer als Primärschlüssel. Es gibt mehrere Autos einer Marke bzw. eines Typs. Ein Kunde kann mehrere Autos reparieren lassen, ein Auto gehört aber nur einem Kunden. Plane die Tabellen und Beziehungen.

Tabelle **Marke** (IDMarke, Bezeichnung)

Tabelle **Typ** (IDTyp, *IDMarke*, Bezeichnung)

Tabelle **Auto** (Fahrgestellnummer, *IDTyp*, *IDKunde*)

Tabelle **Kunde** (IDKunde, Zuname, Vorname, Straße, PLZ, Ort, Telefon)

Primärschlüssel sind unterstrichen, Fremdschlüssel kursiv dargestellt.

Mit dem **Entity Relationship Modell**, das wir bereits besprochen haben, können Relationen und Beziehungen gezeichnet und anschließend in Tabellen aufgelöst werden. Ein besonderes Problem stellt dabei eine **n:m-Beziehung** dar. Diese muss in eine **Beziehungstabelle** aufgelöst werden, es entstehen zwei 1:n-Beziehungen. In einer Access-Datenbank können keine n:m-Beziehungen erstellt werden.

2 Datenbank anlegen

Aufbau einer neuen Datenbank

In Microsoft Access werden **alle Objekte**, wie z.B. Tabellen und deren Daten, Beziehungen, Abfragen, Formulare, Berichte, Makros und Module, die zu einer Datenbank gehören, in einer Datei mit der Endung **.mdb** gespeichert.



Datei | Neu

1 Über den Menüpunkt Datei | Neu wird eine leere Datenbank erstellt.

Die leere Datenbank dient als Container für alle Datenbankobjekte.

2 Im nächsten Schritt werden die Tabellen angelegt.

Bei der Erstellung der Tabellen achten wir besonders auf die Anlage der **Primär- und Fremdschlüssel**. Fremdschlüssel müssen den gleichen Datentyp wie die korrespondierenden Primärschlüssel und aus Performancegründen einen Index erhalten.

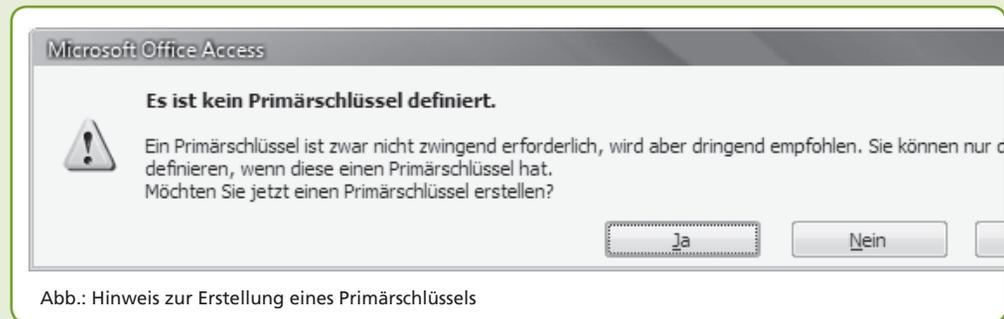


Abb.: Hinweis zur Erstellung eines Primärschlüssels

3 Bei der Erstellung der Beziehungen sollte die referenzielle Integrität verwendet werden.

Die **referenzielle Integrität** prüft, ob die verwendeten **Primär- und Fremdschlüssel zusammenpassen**. Es ist z.B. unzulässig, dass ein nicht existierender Primärschlüssel als Fremdschlüssel verwendet wird.

Zur Verhinderung von falschen Daten und Anomalien verwenden wir die referenzielle Integrität und **aktivieren die Aktualisierungs- und die Löschweitergabe**.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

Mr. What und Ms. Check

Was passiert, wenn ich für ein Feld einen Index erstelle?



Access erstellt intern eine sortierte Liste für das indizierte Feld. Das Suchen nach einem bestimmten Eintrag geht in einer sortierten Liste deutlich schneller als in einer nicht sortierten.



3 Primärschlüssel

Identifikation eines Tupels

Alle Informationen eines Datensatzes innerhalb einer Tabelle hängen vom gleichen Primärschlüssel ab. Ein Primärschlüsseleintrag darf innerhalb einer Tabelle nicht mehrfach vorkommen, weshalb er immer einen Index ohne Duplikate erhält.

1 Jede Tabelle sollte einen Primärschlüssel haben.

Gutes Datenbankdesign zeichnet sich durch die Vermeidung von Redundanz und Anomalien aus. Wir wollen also weder Daten doppelt speichern noch sollen falsche Daten oder Zuordnungen in einer Datenbank enthalten sein. An Beziehungen zwischen Tabellen ist immer mindestens ein Primärschlüssel beteiligt.

2 Ein Primärschlüssel erhält zur Kennzeichnung ein Schlüsselssymbol und einen Index ohne Duplikate.

	Feldname	Felddatentyp
	IDInterpret	Zahl
	Interpret/Gruppe	Text

Abb.: Primärschlüssel

Ein Feld wird zum Primärschlüssel, indem wir auf das Schlüsselssymbol in der Symbolleiste klicken.

3 Ein zusammengesetzter Primärschlüssel sind mehrere als Primärschlüssel definierte Felder in einer Tabelle.

	Feldname	Felddatentyp
	Vorname	Text
	Zuname	Text
	Geburtsdatum	Datum/Uhrzeit
	SV_Nummer	Zahl

Abb.: Zusammengesetzter Primärschlüssel

Ein typisches Beispiel für einen **zusammengesetzten Primärschlüssel** ist die **Sozialversicherungsnummer**, die aus einer vierstelligen Zahl und dem Geburtsdatum besteht. Dieser Primärschlüssel wird häufig zur eindeutigen Identifikation von Personen, z.B. Mitarbeitern oder Patienten, verwendet.

Zur Erstellung eines zusammengesetzten Primärschlüssels werden vor dem Aktivieren des Schlüsselssymbols in der Symbolleiste die entsprechenden Felder markiert.



Primärschlüssel in der Symbolleiste



Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

Beachte

Bei einem **zusammengesetzten Primärschlüssel** können die einzelnen Felder mehrfach vorkommende Werte enthalten, z.B. das gleiche Geburtsdatum. Daher erstellt Access in diesem Fall **keinen Index**. Der **gewünschte Index**, z.B. mit Duplikaten, muss bei den Feldern eines zusammengesetzten Primärschlüssels **manuell gesetzt** werden.

Soll eine Tabelle mit einem **zusammengesetzten Primärschlüssel** mit einer anderen Tabelle verknüpft werden, so müssen darin alle Bestandteile des Primärschlüssels als Fremdschlüssel angelegt werden. Die Beziehung wird in diesem Fall über den Dialog **Beziehungen bearbeiten** erstellt.

4 Beziehungen erstellen Tabellen verknüpfen

Damit wir in Abfragen, Formularen und Berichten Daten aus verschiedenen Tabellen darstellen können, müssen wir die Tabellen über ihre Primär- und Fremdschlüssel miteinander verknüpfen.



Symbol zum Hinzufügen von Tabellen in die Beziehungsansicht.

Sbx

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

Die Aktualisierungsweitergabe bewirkt, dass Änderungen in den Detailfeldern ebenfalls geändert werden.

Löschweitergabe:
Ein Löschen in der Haupttabelle bewirkt auch ein Löschen in der Detailtabelle.

1 Über Extras | Beziehungen gelangen wir in die Beziehungsansicht der Tabellen.

In der **Beziehungsansicht** werden über das Symbol **Hinzufügen von Tabellen** die Tabellen und deren bereits existierende Beziehungen angezeigt. Eine **neue Beziehung** erstellen wir durch **Ziehen des Primärschlüssels auf den Fremdschlüssel** in der Zieltabelle.



Abb.: Beziehungsansicht

2 Im Dialogfenster *Beziehungen bearbeiten* aktivieren wir die referenzielle Integrität sowie die Aktualisierungs- und die Löschweitergabe.

Für die Festlegung der **referenziellen Integrität** müssen beide Felder den **gleichen Datentyp** und korrespondierende Schlüssel aufweisen. Gibt es zu einem Fremdschlüssel keinen passenden Primärschlüssel, ist die referenzielle Integrität für die Beziehung nicht aktivierbar. Dieses Problem kann nur bei Tabellen auftreten, die falsche Daten enthalten, weshalb **Beziehungen immer vor einer Dateneingabe erstellt werden** sollten.

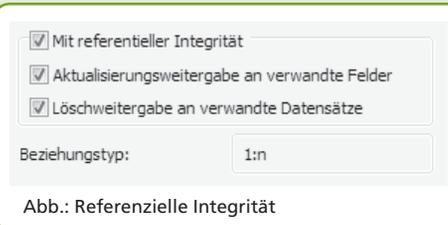


Abb.: Referenzielle Integrität

3 Tabellen mit zusammengesetzten Schlüsseln werden über eine Mehrfachbeziehung verknüpft, die über den Dialog *Beziehungen bearbeiten* erstellt wird.

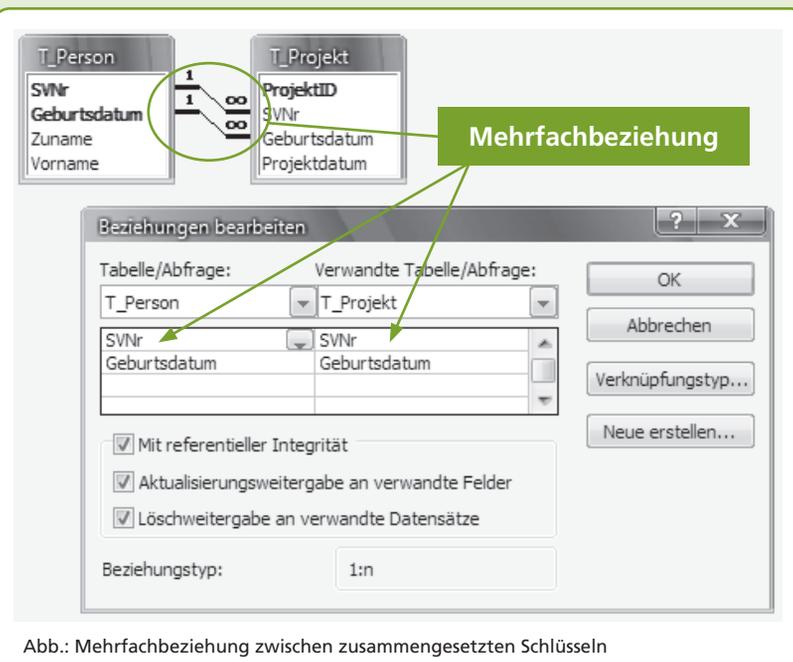


Abb.: Mehrfachbeziehung zwischen zusammengesetzten Schlüsseln

Mr. What und
Ms. Check

Warum erhalte ich bei der Erstellung einer Beziehung die Fehlermeldung „Für die Beziehung ist dieselbe Anzahl an Feldern mit denselben Datentypen erforderlich.“?

Um zwei Tabellen zu verknüpfen, muss der Datentyp beim Primär- und beim Fremdschlüssel ident sein. Falls du zusammengesetzte Schlüssel verwendest, müssen auf beiden Seiten die gleichen Schlüsselfelder vorhanden sein.

Welcher Fehler liegt vor, wenn ich bei der Aktivierung der referenziellen Integrität die Fehlermeldung „Es wurde kein eindeutiger Index für das in Beziehung stehende Feld der Primärtabelle angegeben.“ erhalte?

In diesem Fall gibt es in der n-Tabelle Fremdschlüssel, zu denen kein Primärschlüssel in der 1-Tabelle passt. Lösche die inkonsistenten Daten in der n-Tabelle und aktiviere danach die referenzielle Integrität.



Die Erstellung des Datenbankschemas ist abgeschlossen, sobald wir alle Tabellen mit den erforderlichen Beziehungen erstellt haben. Ein optimales Datenbankdesign enthält für alle Beziehungen zwischen den Tabellen eine referenzielle Integrität.

Bei der Erstellung von Abfragen werden die Beziehungen zwischen den Tabellen automatisch verwendet und in der Entwurfsansicht angezeigt – außer wir erstellen eine Abfrage, die nicht nur Tabellen, sondern zusätzlich weitere Abfragen enthält, die miteinander verknüpft werden. Für diesen Fall müssen wir innerhalb der Abfrage eine Beziehung erstellen.

5 Verknüpfungseigenschaften Gleichheits-, Inklusions- und Reflexivverknüpfung

Mit Hilfe der **Verknüpfungseigenschaften** können wir **in Abfragen** festlegen, wie die Daten zwischen Tabellen und Unterabfragen miteinander verknüpft werden.

Die **Inner-Join-Verknüpfung** ist die Standardverknüpfung zwischen Tabellen bzw. Abfragen.

1 Die Gleichheitsverknüpfung (**Inner Join**) verknüpft Datensätze mit identen Schlüsselfeldern auf beiden Seiten.

Die folgende Auswahlabfrage verknüpft die Tabellen **Song** und **InterpretSingtSong** mit der Auswahlabfrage **A_Interpret_Songanzahl**, die alle Interpreten mit mehr als fünf Liedern enthält. Die Abfrage mit der Gleichheitsverknüpfung gibt alle Interpreten mit ihrer Anzahl an Songs sowie ihren Titeln aus.

The screenshot shows the 'Abfrage' (Query) design view in Microsoft Access. It features three objects: the 'Song' table, the 'InterpretSingtSong' table, and the 'A_Interpret_Songanzahl' query. The 'Song' table and 'A_Interpret_Songanzahl' query are connected by a double line, indicating an inner join. The 'InterpretSingtSong' table is also connected to the 'A_Interpret_Songanzahl' query. The design grid below shows the fields and their relationships:

Feld:	Interpret/Gruppe	AnzahlvonTitel	Titel
Tabelle:	A_Interpret_Songa	A_Interpret_Songa	Song
Sortierung:			
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:			
oder:			

Abb.: Auswahlabfrage mit einer Gleichheitsverknüpfung (**Inner Join**)

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

Beachte

Eine **Abfrage innerhalb einer Abfrage** muss immer **manuell verknüpft** werden.

Left Join und Right Join werden auch als Outer-Join-Verknüpfungen bezeichnet.

Null (gesprochen „Null“) bedeutet kein Wert.



Eine Inklusivverknüpfung ist durch den Pfeil am Ende der Verknüpfung erkennbar.

Ein Doppelklick auf die Beziehung öffnet den Dialog Verknüpfungseigenschaften.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

2 Eine Inklusivverknüpfung (Left Join bzw. Right Join) enthält alle Datensätze der einen Seite sowie die passenden der anderen. Gibt es keine passenden Datensätze auf der anderen Seite, ist der Feldinhalt Null.

Die folgende Abfrage zeigt alle Interpreten, denen keine Lieder zugeordnet sind. Die **Left-Join-Verknüpfung** zeigt **alle** Datensätze der Tabelle **Interpreten** sowie die **passenden Datensätze der Tabelle InterpretSingtSong oder Null** (= kein Wert) an. Durch das Kriterium **Ist Null** werden nur die Interpreten ohne Songs ausgegeben. Eine **Right-Join-Verknüpfung** wirkt in die entgegengesetzte Richtung, der Pfeil erscheint dann links bei der Beziehung.

Verknüpfungseigenschaften

Linker Tabellename: Interpreten
Rechter Tabellename: InterpretSingtSong

Linker Spaltenname: IDInterpret
Rechter Spaltenname: IDInterpret

1. Beinhaltet nur die Datensätze, bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.

2. Beinhaltet **ALLE** Datensätze aus 'Interpreten' und nur die Datensätze aus 'InterpretSingtSong', bei denen die Inhalte der verknüpften Tabellen gleich sind.

3. Beinhaltet **ALLE** Datensätze aus 'InterpretSingtSong' und nur die Datensätze aus 'Interpreten', bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.

OK Abbrechen Neu

Inklusivverknüpfung

Feld:	IDInterpret	Interpret/Gruppe	IDInterpret
Tabelle:	Interpreten	Interpreten	InterpretSingtSong
Sortierung:			
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:			Ist Null

Abb.: Auswahlabfrage mit einer Inklusivverknüpfung (Left Join)

3 Eine Reflexivverknüpfung verknüpft eine Tabelle oder Abfrage mit sich selbst.

Die Tabelle **T_Mitarbeiter** speichert die Namen der Mitarbeiter einer Firma unter dem **Primärschlüssel MitarbeiterID**. Der Vorgesetzte eines Mitarbeiters ist mit seiner **MitarbeiterID** im Feld **VorgesetzterID** eingetragen, z.B. ist der Vorgesetzte von Verkäufer A der Mitarbeiter mit der Nummer 2, also der Leiter der Verkaufsabteilung. Der Geschäftsführer hat keinen Vorgesetzten.

MitarbeiterID	Name	VorgesetzterID
1	Geschäftsführer	0
2	Abteilungsleiter Verkauf	1
3	Verkäufer A	2
4	Verkäufer B	2
5	Abteilungsleiter Einkauf	1
6	Einkäufer A	5
7	Einkäufer B	5

Abb.: Tabelle T_Mitarbeiter

Die Auswahlabfrage gibt eine Liste aller Mitarbeiter mit den Namen ihrer Vorgesetzten aus.

MitarbeiterID	Name	Vorgesetzter
1	Geschäftsführer	
2	Abteilungsleiter Verkauf	Geschäftsführer
5	Abteilungsleiter Einkauf	Geschäftsführer
3	Verkäufer A	Abteilungsleiter Verkauf
4	Verkäufer B	Abteilungsleiter Verkauf
6	Einkäufer A	Abteilungsleiter Einkauf
7	Einkäufer B	Abteilungsleiter Einkauf

Zur Erstellung der Reflexivverknüpfung wird die Tabelle **T_Mitarbeiter** zweimal in die Abfrage übernommen.

SbX

Diese Abbildung findest du in der PowerPoint-Präsentation unter der ID: 9918.

Reflexivverknüpfung

Feld:	MitarbeiterID	Name	Vorgesetzter: Name
Tabelle:	T_Mitarbeiter	T_Mitarbeiter	T_Mitarbeiter_1
Sortierung:			
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:			

Der Verknüpfungstyp ist hier ein **Right Join**, da die Vorgesetzten zu allen Mitarbeitern angezeigt werden sollen.

Abb.: Abfrage mit Reflexivverknüpfung

Üben

Übungsbeispiele

Ü 2:

Eine Bekannte eröffnet in der Innenstadt ein Sushilokal. Bestellungen von Gästen werden auch via Telefon entgegengenommen und anschließend ausgeliefert. Die Verrechnung des Lieferservices soll über eine Datenbank mit dem Namen „Sushi.mdb“ durchgeführt werden. Erstelle die Datenbank mit den folgenden Tabellen.

	Feldname	Felddatentyp
PK	GastNr	AutoWert
	Name	Text
	Straße	Text
	PLZ	Text
	Ort	Text

Feldname	Datentyp und Feldgröße
GastNr	Autowert
Name	Text, 50 Zeichen
Straße	Text, 50 Zeichen
PLZ	Text, 50 Zeichen
Ort	Text, 50 Zeichen

	GastNr	Name	Straße	PLZ	Ort
▶	1	Namamoto Sari	Wiener Str. 5/3	3100	St. Pölten
	2	Kaiser Wilhelm	Kremser Str. 90	3100	St. Pölten
	3	Allg. Versicherung AG	Mariazeller Platz 6	3100	St. Pölten
	4	Gruber Max	Humboltstr. 3	3100	St. Pölten
	5	Sumo San	Europaplatz 1	3100	St. Pölten
	6	Dr. VanHelsing	Draculstr. 3	3500	Krems
	7	Pesto Bueno	Spaghettistr. 8	3504	Stein

	Feldname	Felddatentyp
PK	Nr	AutoWert
	Bezeichnung	Text
	SpeiseCode	Zahl
	Preis	Zahl

Feldname	Datentyp und Feldgröße
Nr	Autowert
Bezeichnung	Text, 50 Zeichen
SpeiseCode	Zahl, Long Integer
Preis	Zahl, Double

	Nr	Bezeichnung	SpeiseCode	Preis
▶	1	California Rolls 8 Stk.	1	8
	2	Sushi klein 5 Stk.	1	4
	3	Sushi groß 10 Stk.	1	7
	4	Maki 8 Stk.	1	7
	5	Sushi Menü Lachs	2	13
	6	Sushi Menü Thunfisch	2	13
	7	Nori klein	1	4
	8	Lachs Sashimi	1	6
	9	Thunfisch Sashimi	1	6
	10	Gemischtes Sashimi	1	6

	Feldname	Felddatentyp
PK	Code	Text
	Speiseart	Text

Feldname	Datentyp und Feldgröße
Code	Text, 1 Zeichen
Speiseart	Text, 50 Zeichen

	Code	Speiseart
▶	A	Vorspeise
	B	Menü
	C	Dessert
	D	Getränke

Sb+

www.wissenistmanz.at

SbX

Die Datenbank „Sushi.mdb“ findest du unter der ID: 9919.

Ü 3:

Erstelle zwischen den Tabellen in der Datenbank „Sushi.mdb“ die erforderlichen Beziehungen und löse die bestehenden Probleme im Datenbankschema.

- a) Die Tabelle **Speiseart** lässt sich nicht mit der Tabelle **Speise** verknüpfen. Finde den Grund dafür heraus und beseitige das Problem. Aktiviere für die Beziehung die referenzielle Integrität mit Aktualisierungsweitergabe.
- b) Von einem Gast sollen mehrere Speisen bestellt werden können. Außerdem muss eine Speise natürlich von mehreren Gästen bestellbar sein. Zusätzlich sollen das Bestelldatum und ein Liefertermin (Zeitpunkt) gespeichert werden. Beachte dabei, dass der gleiche Gast an verschiedenen Tagen die gleiche Speise erneut bestellen kann. Verändere die Tabellenstruktur so, dass sie den Anforderungen entspricht.
- c) Die California Rolls kosten 8,40 EUR, alle Sashimi kosten jeweils 6,50 EUR. Doch die Preise werden offenbar falsch gespeichert. Finde die Ursache dafür heraus, löse das Problem und korrigiere die Preise. Gib folgende Bestellungen in der neu angelegten Tabelle **T_Bestellung** ein.

GastNr	SpeiseNr	Datum	Lieferzeit	Zahlung
1	1	01.03.2007	20:00	3
1	3	01.03.2007	21:36	3
1	4	01.03.2007	21:37	2
2	3	01.03.2007	21:37	1
2	7	01.03.2007	21:37	2
2	8	01.03.2007	21:37	1
3	3	01.03.2007	21:37	1
3	3	01.03.2007	21:37	3
3	9	01.03.2007	21:37	2
3	10	01.03.2007	21:37	3
4	4	28.02.2007	21:38	3
4	5	01.03.2007	22:00	1
4	6	01.03.2007	22:00	1
5	5	01.03.2007	22:00	2
6	3	01.03.2007	21:37	1
6	4	01.03.2007	21:37	1
8	4	28.05.2007	16:45	3
8	11	28.05.2007	16:45	3
8	12	28.05.2007	16:45	3
0	0			0

Finde die Ursache dafür heraus, löse das Problem und korrigiere die Preise. Gib folgende Bestellungen in der neu angelegten Tabelle **T_Bestellung** ein.

Sb+

www.wissenistmanz.at

SbX

Die Datenbank „MP3Sammlung.mdb“ findest du unter der ID: 9919.

Ü 4:

Erstelle in der MP3-Datenbank die Abfrage **A_Oldies** (wie abgebildet), die die Musikrichtungen aller Lieder jener Interpreten enthält, die vor 1950 geboren wurden. Kopiere die Abfrage nach **A_NewAge** und ändere das Kriterium beim Geburtsdatum in „>1980“.

Erstelle die Abfrage **A_Musikrichtungsvergleich**, die beide Abfragen kombiniert.

- a) Wie viele Musikrichtungen kommen in beiden Abfragen gemeinsam vor?

- b) Welche Musikrichtungen gibt es nur bei den Oldies aber nicht bei NewAge?

SbX

ID: 9919

Zusätzlich zu diesen Aufgaben findest du in SbX eine Internetaufgabe.

Sichern

In dieser Lerneinheit haben wir uns mit der Erstellung des Datenbankschemas, das aus Tabellen und Beziehungen besteht, beschäftigt. Wir haben Beziehungen mit Hilfe von Primär- und Fremdschlüsseln erstellt und die verschiedenen Verknüpfungseigenschaften kennengelernt.

Datenbankschema	Das Datenbankschema umfasst Tabellen und deren Beziehungen . Eine Access-Datenbank beinhaltet neben dem Datenbankschema auch Abfragen, Formulare, Berichte, Makros und Module.
Tabellendesign	Zur Erstellung einer Tabelle werden die Felder mit ihren Datentypen und Feldgrößen in der Entwurfsansicht angelegt. Jede Tabelle erhält einen Primärschlüssel , der ein Tupel (= Datensatz) eindeutig identifiziert .
Primärschlüssel	Ein Primärschlüsselfeld erhält einen Index ohne Duplikate und kann daher nur eindeutige Werte innerhalb einer Tabelle aufnehmen. Bei einem zusammengesetzten Primärschlüssel werden mehrere Felder als Schlüssel definiert.
Fremdschlüssel	Ein Fremdschlüssel ist das Pendant zum Primärschlüssel in einer anderen Tabelle und muss daher den gleichen Datentyp und die gleiche Feldgröße aufweisen. Aus Performancegründen sollte ein Fremdschlüssel einen Index mit bzw. ohne Duplikate erhalten, je nachdem, welche Beziehung vorliegt – 1:n bzw. 1:1 .
Beziehung	Über den Primär- und den Fremdschlüssel werden Tabellen miteinander verknüpft , indem eine Beziehung zwischen den Schlüsseln erstellt wird. Den Grad der Beziehung (1:n, 1:1) steuert der Index der Schlüsselfelder.
Referenzielle Integrität	Die referenzielle Integrität stellt sicher, dass als Fremdschlüssel nur solche Werte verwendet werden, die es auch als Primärschlüssel gibt. Mit Hilfe der Aktualisierungs- und der Löschweiteregabe werden Änderungen beim Primärschlüssel automatisch auf alle verknüpften Fremdschlüssel übertragen.
Inner Join	Standardmäßig werden Tabellen mit der Gleichheitsverknüpfung verbunden. Dabei werden nur jene Datensätze angezeigt, bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.
Outer Join	Bei der Inklusionsverknüpfung (Left Join, Right Join) werden alle Daten der einen sowie die übereinstimmenden Daten der verknüpften Tabelle oder Null angezeigt.
Reflexivverknüpfung	Bei einer Reflexivverknüpfung wird eine Tabelle mit sich selbst verknüpft .

 ID: 9920

Zusätzlich zu dieser Zusammenfassung findest du in SbX eine Bildschirmpräsentation.

Wissen

Wiederholungsfragen und -aufgaben

1. Was machst du beim Design einer Datenbank mit Mehrfachinformationen in Tupeln?
2. Welche Grade von Beziehungen können in einer Datenbank verwendet werden?
3. Warum sollte jede Tabelle einen Primärschlüssel haben?

4. Was musst du hinsichtlich der Indexierung bei einem zusammengesetzten Primärschlüssel beachten?
5. Erkläre die Wirkungsweise der referenziellen Integrität!
6. Warum sollte für einen Fremdschlüssel ein Index festgelegt werden?
7. Wie funktioniert ein Index?
8. Kann die referenzielle Integrität verwendet werden, wenn eine der beiden Tabellen keinen Primärschlüssel hat?
9. Ordne die Schritte zur Erstellung einer Datenbank in chronologischer Reihenfolge:

Aufgabe	Reihenfolge
Mehrfachinformationen auf Tabellen aufteilen	
Primärschlüssel festlegen	
Leere Datenbankdatei .mdb erstellen	
Felder mit Datentypen und Feldgrößen anlegen	
Tabellen verknüpfen	
Informationen sammeln, die in der Datenbank gespeichert werden sollen	

10. Erkläre den Unterschied zwischen einer Gleichheits- und einer Inklusionsverknüpfung!
11. Was ist eine Reflexivverknüpfung?
12. Kreuze die richtigen Antworten an!

Aussage	Richtig
Die Dateierweiterung einer Access-Datenbank ist „.mbd“.	
Access erkennt am ersten Datensatz in der Datenblattansicht, um welchen Datentyp es sich handelt und vergibt diesen automatisch.	
Ein Feld mit dem Datentyp <i>Autowert</i> kann nur mit Zahlen gefüllt werden.	
Für eine mit dem Assistenten erstellte Tabelle kann der Name festgelegt werden.	
Wenn kein Primärschlüssel vergeben wird, verwendet Access automatisch das erste Datenfeld als Primärschlüssel.	
Der Primärschlüssel verhindert doppelte Datensätze.	
Ein Primärschlüssel hat immer einen Index mit Duplikaten.	
Das Sortieren und Filtern geht bei indizierten Feldern schneller.	
Die Verwendung von Indizes benötigt keinen zusätzlichen Speicherplatz.	
Eine Beziehung ist eine Verknüpfung von zwei Tabellen mit Hilfe gemeinsamer Felder.	
Die referenzielle Integrität kann nur eingestellt werden, wenn die verknüpften Felder den gleichen Datentyp haben. Die Feldgröße ist dabei nicht relevant.	
Mit der referenziellen Integrität bleiben die Fremdschlüssel in einer Datenbank konsistent.	
Redundanz steht für das sofortige Speichern von Daten.	
Eine n:m-Beziehung wird mit Hilfe einer Inklusionsverknüpfung erstellt.	
Beziehungen können auch ohne referenzielle Integrität erstellt werden.	

 SbX
 ID: 9921

Zusätzlich zu diesen Aufgaben findest du in SbX eine Drag&Drop-Aufgabe.

Lerncheck

Ich kann jetzt ...

- ... Einfach- und Mehrfachinformationen bei Tupeln unterscheiden und daraus ein Datenbankschema mit Tabellen und Beziehungen erstellen.
- ... eine neue Access-Datenbank mit Tabellen, Primär- und Fremdschlüsseln sowie Beziehungen anlegen.
- ... die Indexierung für zusammengesetzte Primärschlüssel sowie für Fremdschlüssel zur Verbesserung der Datenbankperformance anwenden.
- ... **Inner-** und **Outer-Join**-Verknüpfungen bei Beziehungen verwenden, um z.B. Abfragen innerhalb von Abfragen zu verwenden.

Kapitelrückblick

In diesem Kapitel haben wir die Datenmodellierung mit Hilfe des ER-Modells, die relationale Auflösung und die Normalformen kennengelernt. Außerdem haben wir uns mit der Erstellung einer Datenbank beschäftigt.

Was genau sollte ich nun eigentlich wissen?



Anhand des Lernchecks hier am Kapitelende kannst du leicht überprüfen, ob du wirklich alles verstanden hast!

Lerncheck

Ich kann jetzt ...

- ... verschiedene Datenmodelle, insbesondere das relationale Modell, beschreiben.
- ... ein ER-Modell zur Darstellung von Entitäten, Beziehungen und Attributen verwenden.
- ... die Kardinalität von Beziehungen bestimmen.
- ... Entitäten und Beziehungen in Relationen auflösen.
- ... die erste, zweite und dritte Normalform erklären und auf Relationen anwenden.
- ... die Indexierung und die referenzielle Integrität in einer Datenbank richtig einsetzen.
- ... in Access neue Datenbanken mit Tabellen und Beziehungen anlegen.

SbX

ID: 9922

Zusätzlich findest du in SbX eine zusammenfassende Bildschirmpräsentation.