

# 10 Informatik



**TEILDRUCK**  
GENEHMIGTE AUFLAGE  
ERSCHEINT IM FESTEINBAND



Gymnasium Bayern  
NTG

Alle folgenden Kapitel haben dieselbe Struktur und sind aus denselben Gliederungseinheiten aufgebaut.

**Auftaktseite**  
Abholen im Alltag und Ausblick auf die neuen Kompetenzen

### Verknüpfen von Tabellen 1

**Erläuterung**  
Im Projekt der Software-AG wurde eine Datenbank für die Verwaltung der Schule entwickelt. Die folgenden zwei Tabellen sind Teil der Datenbank.

- Erläutere den Aufbau der beiden Tabellen und wie sie zusammenhängen.
- Beschreibe das Problem, wenn gleichzeitig die Schulleiter, die Klassenleiter und die Klassenleiterinnen ausgewählt werden soll.

Schüler		Lehrkraft	
Matr.	Name	Matr.	Name
101	Max	1001	Meier
102	Paul	1002	Müller
103	Anna	1003	Schmidt
104	Tom	1004	Wagner
105	Lena	1005	Koch
106	Tim	1006	Beck
107	Julia	1007	Hahn
108	Ben	1008	Hecht
109	Sara	1009	Huber
110	David	1010	Fischer

Am Ende dieses Kapitels hast du gelernt...

- SQL-Aufgaben über mehrere Tabellen zu erstellen.
- Beziehungen zwischen Klassen darzustellen und zu analysieren.

## Lernen braucht Struktur – Das Doppelseitenprinzip

Alle neuen Inhalte werden auf gleich strukturierten Doppelseiten aufbereitet. Gleich aufgebaute Themenseiten behandeln weitere fakultative interessante Inhalte.

**Einstieg**  
Motivierende Fragen zum neuen Thema

**Erarbeitung**  
Kleinschrittige Erarbeitung in mehreren Blöcken mithilfe von:

- ▶ kompakten Informationstexten mit Erklärungen am Beispiel
- ▶ praxisorientierten und auch materialbasierten Arbeitsaufträgen

### 14 1.4 Beziehungen zwischen Klassen

**EINSTIEG**  
Niels ist in der Software-AG und hat sich folgende Notizen gemacht.

```

Schüler > besucht > Schulklassen
Lehrkraft < hat als Klassenleiter < Schulklassen
    
```

Beschreibe mögliche Überlegungen, die Niels bei diesen Diagrammen hatte. Gib die Bedeutung der Zeichen < und > an. Erkläre die Funktion des Punkts im ersten Diagramm. Formuliere die untere Beziehung in beiden Richtungen in jeweils einem Satz, der mit „Ein(e) ...“ beginnt.

**ERARBEITUNG**  
„beliebig viele“ stimmt im Sachzusammenhang natürlich nicht, das ist aber für die Modellierung unerheblich.

Bedeutungen:  
n: kein oder ein  
1: genau ein  
0..1: kein oder ein

```

bisher:
Schüler > besucht > Schulklassen
jetzt:
Schüler > besucht > Schulklassen
    
```

Diese Angaben nennt man **Kardinalität**. n bedeutet, dass n Objekte der Klasse Schüler in Beziehung stehen können. In diesem Fall spricht man von einer **1:n-Beziehung**.

**A1 1:n-Beziehungen analysieren**

- Suche weitere Paare von Klassen, z. B. aus dem Umfeld der Schule, zwischen denen eine 1:n-Beziehung besteht, und zeichne jeweils das Klassendiagramm mit den jeweiligen Kardinalitäten.
- Ein Zug besteht aus vielen Waggons, ein Waggon ist (zu einem bestimmten Zeitpunkt) Teil von genau einem oder keinem Zug. Zeichne das Objektdiagramm für einen Zug aus drei Waggons und das Klassendiagramm der Klassen Waggon und Zug einschließlich der Kardinalitäten.

Zwischen den Klassen Schüler und Schulklassen gibt es eine weitere Beziehung: Genau ein Schüler ist (erstster) Klassensprecher.  
Ein Schüler ist Klassensprecher von genau einer oder keiner Schulklassen. Eine Schulklassen hat genau einen Klassensprecher.

Als Klassendiagramm:  
Schüler 1 ist Klassensprecher 0..1 Schulklassen

Dies nennt man eine **1:1-Beziehung**. Um anzudeuten, dass nicht jeder Schüler Klassensprecher ist, steht bei der Schulklassen 0..1.

**Üben und Vertiefen**  
Leichte und anspruchsvolle Aufgaben sowohl zum analogen als auch zum praktischen Arbeiten, dazu:

- ▶ weitere Übungsmöglichkeiten speziell zu jedem Unterkapitel
- ▶ vernetzende Aufgaben

### 30 1.9 Üben und Vertiefen

- Gegeben sind die SQL-Aufgabe `SELECT * FROM A, B WIESE A.16 = B.16` und die Funktion `Zeichne(A, B)`. A und B sind jeweils Tabellen mit den Primärschlüsseln ID.
  - Begründe, warum keine dieser Abfragen beschreibt.
  - Zeichne das gemeinsame Datenfeldmodell.
  - Gib die Datenfeldmodelle und die funktionale Schreibweise an, wenn von beiden Tabellen nur die Attribute `name` benötigt werden.
- Eine Beziehung kann auch zwischen mehr als zwei Tabellen bestehen, z. B. in diesem Fall sind mehrere Fremdschlüssel in der Beziehungstabelle nötig.
  - Gib die Primärschlüssel an, auf die sich die Fremdschlüssel in der Tabelle beziehen lassen.
  - Begründe für jede Spalte der Tabelle, warum sie in der Primärschlüssel aufgenommen werden muss. Beschreibe jeweils ein Beispiel, das verdeutlicht, warum die entsprechende Spalte nicht fehlen darf.
  - Begründe, warum die Attribute `name`, `betrag` und `akt` nicht in den Klassen Schüler, Lehrkraft oder Fach gespeichert werden können.
- Eine Online-Plattform speichert Daten in mehreren Tabellen.
  - Erkläre, welche Tabellen Klassen beschreiben und welche Beziehungen. Gib an, zwischen welchen Klassen es eine Beziehung beschreibt.
  - Stelle das zugehörige Datenmodell als Klassendiagramm dar.
  - Gib das zugehörige relationale Modell mithilfe der Tabellenstruktur aller Tabellen an. Kennzeichne Primär- und Fremdschlüssel.
  - Konvertiere in jeder eine Beziehung eine geeignete SQL-Aufgabe, sodass alle verknüpften Datensätze aufgelistet werden.
  - Bei den Tabellen wurde ein Kardinalitätsmodell angelegt. Nenne nun an, in welche Weise Vorname und Name der Primärschlüssel und in Produkt, Name und Typ. Gib an, wie denn die Beziehungstabelle zwischen beiden aufgebaut sein müsste und formuliere eine Abfrage, die alle verknüpften Datensätze auflistet. Beschreibe die Nachteile dieser Modellierung.

### Verknüpfen von Tabellen 31

Zwischen zwei Klassen kann auch mehr als eine Beziehung bestehen, z. B. Ein Schüler besucht genau eine Schulklassen. Ein Schüler ist (erstster) Klassensprecher von genau einer Schulklassen. Weiter gibt es Schulklassen in n genau einer Schule.

Erstelle das Datenmodell und stelle es grafisch dar. In allen erforderlichen Klassen gib eine 1:n-Beziehung an und die Attribute `name`.

- Alle Schüler, die Klassensprecher sind.
- Alle Schulklassen, die Klassenlehrer sind.
- Alle Klassenlehrer, die Klassenlehrer sind.

Als Klassendiagramm:  
Schüler 1 ist Klassensprecher 0..1 Schulklassen

Dies nennt man eine **1:1-Beziehung**. Um anzudeuten, dass nicht jeder Schüler Klassensprecher ist, steht bei der Schulklassen 0..1.

Wurde ein Begriff vergessen? Mithilfe von Verweisen kann schnell eine Erklärung zum Nachlesen im entsprechenden Unterkapitel oder im Glossar gefunden werden.

## Alles im Blick

Zum Nachlesen und Wiederholen:

- ▶ Alle Inhalte und Fachbegriffe werden übersichtlich zusammengefasst.
- ▶ Alle wichtigen Grundlagen sowie deren Umsetzung in der Praxis sind übersichtlich gegenübergestellt.

**1.8 Alles im Blick**

**SQL-Abfrage**  $\rightarrow$  1.1  
 Jede SQL-Abfrage kann als mehrstufige Funktion interpretiert werden. Die Funktion wählt Daten aus der Tabelle aus und bereinigt sie. Diese werden wieder als Tabelle ausgegeben. Welche Daten die Ausgabe enthält und wie sie dargestellt werden, hängt von weiteren Ergänzungen ab wie z. B. der Auswahl und Sortierung von Spalten oder Bedingungen.

**Kreuzprodukt**  $\rightarrow$  1.2  
 Unter dem Kreuzprodukt A x B zweier Mengen A und B versteht man die Menge aller geordneten Paare (a, b) mit a  $\in$  A und b  $\in$  B.

**Beispiel**  
 $A = \{1, 2\}$ ;  $B = \{1, 2\}$   
 $A \times B = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$   
 In relationalen Datenbanken entsprechen den Elementen der Datenmengen in Tabellen. Relationale Schreibweise:  
 Kreuzprodukt  $\langle$ Table1  $\times$  Table2  $\rangle$  als Datenlängengraben:  
 $\langle$ Table1  $\times$  Table2  $\rangle$   $\rightarrow$   $\langle$ Table1  $\times$  Table2  $\rangle$

**Foreign-Keys**  $\rightarrow$  1.3  
 Einmal eine Tabelle eine Spalte, in der ausschließlich Werte des Primärschlüssels einer meist anderen Tabelle stehen, so spricht man von einem Foreign-Keys. Damit kann ein Bezug zwischen den in den Tabellen gespeicherten Objekten hergestellt werden. Foreign-Keys werden in diesem Buch  $\rightarrow$  [https://www.ccbuchner.de](#)

**Foreign-Keys**  
 Foreign-Keys sind Spalten in einer Tabelle, die auf den Primärschlüssel einer anderen Tabelle zeigen. Sie stellen einen Zusammenhang zwischen den Datensätzen in zwei Tabellen her.

## Verknüpfen von Tabellen 15

### A2 1:1-Beziehungen analysieren

- Suche weitere Paare von Klassen z. B. aus dem Umfeld der Schule, zwischen denen eine 1:1-Beziehung besteht, und zeichne jeweils das Klassendiagramm mit Kardinalitäten.
- Zeichne für deine Schulklasse jeweils das Objektdiagramm für die Beziehungen **istKlassenlehrer** und **istKlassenleiter**.

Auch Beziehungen, bei denen an einer oder beiden Seiten 0, 1 steht, werden als 1:1-Beziehung bezeichnet.

### Die n:m-Beziehung

Zwischen **Lehrkraft** und **Schüler** gibt es die Beziehung **unterrichtet**: Eine Lehrkraft unterrichtet beliebig viele Schüler. Ein Schüler wird unterrichtet von beliebig vielen Lehrkräften.



Die Angabe n (oder m) schließt den Fall n=0 mit ein. Soll deutlich gemacht werden, dass dieser Fall nicht erlaubt ist, so schreibt man auch 1..n.

Das nennt man eine **n:m-Beziehung**. Die Bezeichnungen n und m sollen deutlich machen, dass die Anzahl bei **Lehrkraft** und **Schüler** unterschiedlich sein kann.

### A3 n:m-Beziehungen analysieren

- Suche weitere Paare von Klassen z. B. aus dem Umfeld der Schule, zwischen denen eine n:m-Beziehung besteht, und zeichne jeweils das Klassendiagramm.
- Lilly fährt gerne Fahrrad und Skateboard. Zum Fahrradfahren benötigt sie das Fahrrad und den Helm, zum Skateboardfahren den Helm, die Kniechoner und das Skateboard. Ordne die Daten und zeichne die Objektdiagramme und das Klassendiagramm für die Beziehung **wirdBenötigtFür**.

Hinweis:  
 Fasse Fahrradfahren und Skateboardfahren jeweils als ein Objekt auf.

Die **Kardinalität** gibt an, wie viele Objekte dieser Klasse mit einem Objekt der anderen Klasse in Beziehung stehen können, n bzw. m stehen für eine beliebige Anzahl von Objekten (auch keines) und die Zahl 1 für genau ein Objekt. Man unterscheidet die Beziehungstypen: 1:1-, 1:n- und n:m-Beziehung.

- Zeichne zu den folgenden Klassenpaaren ein Klassendiagramm mit entsprechenden Kardinalitäten. Benenne die Beziehungen in beide Richtungen sinnvoll und begründe deine Entscheidung. Formuliere die Beziehung in beide Richtungen jeweils in einem Satz, der mit „Ein(e) ...“ beginnt.
  - Schüler – Lehrkraft
  - Museum – Kunstwerk
  - Zimmer – Stuhl
  - Land – Hauptstadt
  - Land – Stadt
  - Land – Fluss

**MERKE**

## Merke

Hier steht das Wichtigste in Kürze.

**AUFGABEN**

## Übungsaufgaben

- ▶ Leichte Aufgaben zum Festigen der neu erarbeiteten Inhalte und Kompetenzen
- ▶ Symbole kennzeichnen Computer- und Gruppenarbeiten

Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ Kardinalitäten von Beziehungen und Beziehungstypen zu bestimmen.
- ▶ Beziehungen zwischen Klassen zu erkennen.

## Rückblick

Die im Unterkapitel erworbenen Kompetenzen im Überblick

## Am Ziel

- ▶ Sitzen alle neuen Begriffe und Basiskompetenzen?
- ▶ Lösungen im Anhang

## 32 1.10 Am Ziel

### DAS GROSSE INFO-QUIZ!

Hier sprechen Begriffe, die du in diesem Kapitel gelernt und wiederholt hast. Über dich: Die Zahlen in Klammern geben den Buchstaben in anderen Begriffen an, den du für die Lösung brauchst.

A. (M)ore Abstrakter sind welche aus C? (1)

B. (W)elche geben an, welche Art von Informationen in den Attributen gespeichert werden? (4)

C. (M)ich brauche das, damit jeder Datensatz eindeutig identifiziert ist? (1)

D. (L)ich erkläre, wenn keine (W)EISE mehr aus einer Tabelle ausgelesen wird? (1)

E. (D)er Begriff aus C wird so im Tabellenschema markiert? (1)

F. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

G. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

H. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

I. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

J. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

K. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

L. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

M. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

N. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

O. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

P. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

Q. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

R. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

S. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

T. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

U. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

V. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

W. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

X. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

Y. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

Z. (L)ich erkläre Objekte stehen nie in Beziehung? (1)

Um das digitale Material abzurufen, kann der QR-Code gescannt oder der Mediencode unter [www.ccbuchner.de/medien](http://www.ccbuchner.de/medien) eingegeben werden.



# 10 Informatik

Herausgegeben von Dieter Bergmann  
und André Greubel

Dieter Bergmann  
André Greubel  
Verena Haller  
Peter Mahns  
Thomas Rau  
Julia Reitberger  
Wolfgang Riffelmacher  
Ulrich Schneider  
Anja Stube  
Hilmar Vogel  
Barbara Wiczorek

C.C.Buchner

# Informatik

Gymnasium Bayern (NTG)

Herausgegeben von Dieter Bergmann und André Greubel

## Informatik 10

Bearbeitet von Dieter Bergmann, André Greubel, Verena Haller, Peter Mahns, Thomas Rau, Julia Reitberger, Wolfgang Riffelmacher, Ulrich Schneider, Anja Stube, Hilmar Vogel und Barbara Wieczorek

Zu diesem Lehrwerk sind erhältlich:

- ▶ Digitales Lehrmaterial **click & teach 10** Einzellizenz, Bestell-Nr. 380151
  - ▶ Digitales Lehrmaterial **click & teach 10** Box (Karte mit Freischaltcode), ISBN 978-3-661-38015-5
- Weitere Materialien finden Sie unter [www.ccbuchner.de](http://www.ccbuchner.de).

Dieser Titel ist auch als digitale Ausgabe **click & study** unter [www.ccbuchner.de](http://www.ccbuchner.de) erhältlich.

Bitte beachten: An keiner Stelle im Schülerbuch dürfen Eintragungen vorgenommen werden. Das gilt besonders für die Leerstellen in Aufgaben und Tabellen.

Die enthaltenen Links verweisen auf digitale Inhalte, die der Verlag bei verlagsseitigen Angeboten in eigener Verantwortung zur Verfügung stellt. Links auf Angebote Dritter wurden nach den gleichen Qualitätskriterien wie die verlagsseitigen Angebote ausgewählt und bei Erstellung des Lernmittels sorgfältig geprüft. Für spätere Änderungen der verknüpften Inhalte kann keine Verantwortung übernommen werden.

Teildruck 2022

Alle Drucke dieser Auflage sind, weil untereinander unverändert, nebeneinander benutzbar.

Dieses Werk folgt der reformierten Rechtschreibung und Zeichensetzung. Ausnahmen bilden Texte, bei denen künstlerische, philologische oder lizenzrechtliche Gründe einer Änderung entgegenstehen.

© 2022, C.C.Buchner Verlag, Bamberg

Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Nutzung in anderen als den gesetzlich zugelassenen Fällen bedarf der vorherigen schriftlichen Einwilligung des Verlags. Das gilt insbesondere auch für Vervielfältigungen, Übersetzungen und Mikroverfilmungen. Hinweis zu § 52 a UrhG: Weder das Werk noch seine Teile dürfen ohne eine solche Einwilligung eingescannt und in ein Netzwerk eingestellt werden.

Dies gilt auch für Intranets von Schulen und sonstigen Bildungseinrichtungen.

Layout und Satz: Wildner+Designer GmbH, Fürth  
Umschlag: Wildner+Designer GmbH, Fürth



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

ISBN 978-3-661-38010-0



<b>1</b>	<b>Verknüpfen von Tabellen</b>	<b>7</b>
1.1	Wiederholung: Daten in einer Tabelle	8
1.2	Daten in mehreren Tabellen, Kreuzprodukt	10
1.3	Fremdschlüssel und Joins	12
1.4	Beziehungen zwischen Klassen	14
1.5	Umsetzung von 1:n-Beziehungen in relationalen Datenbanken	16
1.6	Umsetzung von n:m-Beziehungen in relationalen Datenbanken	18
1.7	Datenmodellierung	20
1.8	Alles im Blick	22
1.9	Üben und vertiefen	26
1.10	Am Ziel	32



<b>2</b>	<b>Modellieren von Datenbanken</b>	<b>33</b>
2.1	Redundanz und Anomalien	34
2.2	Datenbankmodellierung zur Vermeidung von Redundanzen	36
2.3	Datenmodellierung	38
2.4	Automatische Analyse verknüpfter Datenbestände	40
2.5*	Thema: Normalformen	42
2.6	Alles im Blick	44
2.7	Üben und vertiefen	46
2.8	Am Ziel	50



<b>3</b>	<b>Wiederholung, Polymorphie</b>	<b>51</b>
3.1	Wiederholung: Programmierung in Java mit Greenfoot	52
3.2	Wiederholung: Modellierung von Algorithmen, Klassen und Objekten	54
3.3	Wiederholung: Vererben und Überschreiben	56
3.4	Klassen als Datentypen	58
3.5*	Thema: Typumwandlung	60
3.6	Polymorphismus	62
3.7	Alles im Blick	64
3.8	Üben und vertiefen	66
3.9	Am Ziel	70



<b>4</b>	<b>Objektbeziehungen</b>	<b>71</b>
4.1	Klassen in Java mit BlueJ implementieren	72
4.2	Kommunikation zwischen Objekten	74
4.3	Zugriffsrechte in Klassen	76
4.4	Das Hauptprogramm	78
4.5	Objektbeziehungen	80
4.6	Beziehungen in der Programmierung modellieren	82
4.7	Umsetzung von Beziehungen im Programmtext	84
4.8*	Thema: Exceptions und Null-Referenzen	86
4.9	Alles im Blick	88
4.10	Üben und vertiefen	92
4.11	Am Ziel	100



<b>5</b>	<b>Verwaltung gleichartiger Daten</b>	<b>101</b>
5.1	Verwaltung gleichartiger Objekte – Sammlungen	102
5.2	Die Datenstruktur Array	104
5.3	Anwendung der Datenstruktur Array	106
5.4	Durchlaufen eines Arrays	108
5.5	Arbeiten mit den Elementen eines Arrays	110
5.6	Arbeiten mit den Elementen eines Arrays 2	112
5.7	Sortieren in einem Array – der Bubblesort-Algorithmus	114
5.8*	Thema: Die Klasse ArrayList	116
5.9	Alles im Blick	118
5.10	Üben und vertiefen	122
5.11	Am Ziel	130



<b>6 Projektkapitel</b>	<b>131</b>
6.1 Das Projekt	132
6.2 Artikelmanagement in einem Onlineshop	134
6.3 Programmierung in BlueJ	138

### Anhang

Fit für die Praxis	144
Lösungen	166
Glossar	174
Stichwortverzeichnis	184
Bildnachweis	186



# Verknüpfen von Tabellen

# 1

## Einstieg

Im Projekt der Software-AG wurde eine Datenbank für die Verwaltung der Schule entwickelt. Die folgenden zwei Tabellen sind Teil der Datenbank.

- ▶ Erkläre den Aufbau der beiden Tabellen und wie sie zusammenhängen.
- ▶ Beschreibe das Problem, wenn gleichzeitig die Schulklasse, ihr Klassenzimmer und der vollständige Klassenleiternamen ausgegeben werden soll.

Schulklasse			Lehrkraft							
SKId	Zimmer	Klassenleiter	Kürzel	Name	Vorname	GebDatum	Ges	verbeamtet	Fach1	Fach2
...	...	...	...	...	...	...	...	...	...	...
9f	106	vog	bab	Bauer	Britta	1992-11-30	w	wahr	D	G
10a	210	bab	bax	Bauer	Xylko	1970-12-24	d	wahr	E	Ek
10b	211	wal	dri	Drimakis	Helena	1995-03-19	w	falsch	D	Sp
10c	212	dim	...	...	...	...	...	...	...	...
10d	213	dri	vog	Vogel	Hilmar	1973-02-09	m	wahr	Ph	Inf
11a	301	bax	wal	Walther	Andrea	1986-11-11	w	wahr	D	M
...	...	...	...	...	...	...	...	...	...	...

## Am Ende dieses Kapitels hast du gelernt, ...

- ▶ SQL-Abfragen über mehrere Tabellen zu erstellen.
- ▶ Beziehungen zwischen Klassen darzustellen und zu analysieren.

## EINSTIEG

Die Tabelle **Lehrkraft** der Schuldatenbank hat folgenden Aufbau:

Lehrkraft							
<u>Kürzel</u>	Name	Vorname	GebDatum	Ges	verbeamtet	Fach1	Fach2
bab	Bauer	Britta	1992-11-30	w	wahr	D	G

- ▶ Ordne **Lehrkraft**, **Kürzel** und **bab** den Begriffen **Attribut**, **Attributwert** und **Klassenname** zu.
- ▶ Zeichne die **Klassenkarte** der zugehörigen Klasse und die **Objektkarte** für Britta Bauer.
- ▶ Erkläre, warum es günstig ist, die Spalte **Kürzel** einzuführen.

## ERARBEITUNG

## Lehrkraft

Kürzel  
...  
Fach2

## Tabellen und ihr Schema

In relationalen Datenbanken werden die **Attributwerte** von Objekten in Tabellen gespeichert. Die Umsetzung erfolgt in drei Schritten:

1. Aus jeder Klasse wird eine Tabelle. 

Lehrkraft
-----------
2. Aus jedem Attribut wird eine Spalte. 

Kürzel	...	Fach2
--------	-----	-------
3. Aus jedem Objekt wird eine Zeile.  
Die Attributwerte ( **Datensatz**) stehen in den Zellen. 

Kürzel	...	Fach2
bab	...	G

Das **Tabellenschema** **Lehrkraft**(**Kürzel**: Text; ...; **Fach2**: Text) enthält neben den Spaltennamen zusätzlich die **Datentypen**.

Attribute, die einen Datensatz eindeutig identifizieren, nennt man **Primärschlüssel** (oder: **Schlüssel**). Sie werden im Tabellenschema und in der Tabelle unterstrichen.

Datentypen:  
Text; Zahl; Datum; Zeit;  
Wahrheitswert.  
Weitere Datentypen in  
Datenbanksystemen  
 Fit für die Praxis (III.)

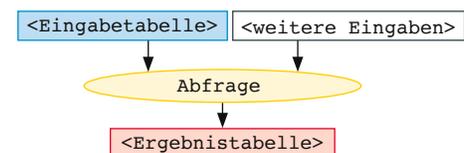
## A1 Tabellen erstellen

- Erstelle zu den Objektkarten eine Tabelle analog zu den drei Schritten oben.
- Die Klasse **Mond** hat die Attribute:  
**Name**, **Planet**, **Halbachse**, **Durchmesser**, **Umlaufzeit**, **Masse** und **entdeckt**. Gib das Tabellenschema an und kennzeichne ein geeignetes Primärschlüsselattribut.

M1: Mond		M2: Mond	
Name	= Titan	Name	= Phobos
...		...	
entdeckt	= 1655	entdeckt	= 1877

## Informationen aus einer Tabelle gewinnen

Mithilfe einer **SQL-Abfrage** können Informationen aus einer Tabelle gewonnen werden. Das Ergebnis ist wieder eine Tabelle.



```

SELECT      <Attribut1> AS <Name1>, <Attribut2> AS <Name2>, ...
FROM        <Tabellename>
WHERE       <Bedingung1> AND | OR <Bedingung2> ...
ORDER BY   <Attribut> [DESC | ASC]
  
```

## A2 Daten auswählen

Erstelle je eine SQL-Abfrage, deren Ergebnistabelle folgende Daten enthält:

- Namen und Vornamen aller weiblichen Lehrkräfte (w)
- Vornamen und Geburtsdaten aller Lehrkräfte nach dem Vornamen sortiert
- Namen aller Lehrkräfte, die Deutsch (D) und Englisch (E) unterrichten
- Namen aller Lehrkräfte, deren Nachname mit S anfängt

Hinter **SELECT** kann auch nur ein Stern (\*) stehen.



38010-01

benötigte Tabelle:  
**Lehrkraft**

Hinweis zu d):

**LIKE "S%"**

Fit für die Praxis (V.)

### Tabellen erweitern mit Aggregatfunktionen und Operatoren\*

↳ Aggregatfunktionen ermitteln aus den Attributwerten der angegebenen Spalte einen neuen Wert. GROUP BY <Attribut> fasst Datensätze, bei denen die Attributwerte von <Attribut> gleich sind, zu einer Gruppe zusammen. Dann wird für jede Gruppe ein Wert berechnet.

↳ Operatoren ermitteln aus unterschiedlichen Attributen eines Datensatzes einen neuen Wert.

Mit \* gekennzeichnete Inhalte sind fakultativ.

Aggregatfunktionen:

- COUNT(\*)
- MAX(<Attribut>)
- MIN(<Attribut>)
- SUM(<Attribut>)
- AVG(<Attribut>)

Operatoren:

+, -, \*, /; CONCAT

#### A3\* Aggregatfunktionen und Operatoren anwenden

Erstelle je eine SQL-Abfrage, deren Ergebnistabelle folgende Daten enthält:

- a) für jeden Planeten den Durchmesser seines größten Mondes
- b) für jeden Planeten die Summe der Massen all seiner Monde
- c) für Jupiter den Namen, die Halbachse in km und die Bahngeschwindigkeit v aller Monde in  $\frac{km}{h}$  (Die Umlaufzeit in der Tabelle ist in Tagen angegeben.)



38010-02

benötigte Tabelle:  
Mond

Werden Objekte derselben Klasse in einer Datenbank gespeichert, legt man für jede Klasse eine Tabelle an, deren Spaltenbezeichner den Attributen entsprechen. Die Attributwerte der Objekte werden zeilenweise eingetragen und bilden einen Datensatz. Im Tabellenschema werden alle Attribute mit ihren Datentypen angegeben und der Primärschlüssel gekennzeichnet. Eine SQL-Abfrage erstellt eine neue Tabelle, deren Spalten und Datensätze durch Angabe bestimmter Regeln bestimmt werden. \*Mit Aggregatfunktionen und Operatoren können neue Spalten erzeugt werden, die in der ursprünglichen Tabelle nicht vorhanden waren.

**MERKE**

 1 Untersuche diese Abfragen auf Fehler und beschreibe bei funktionierenden Abfragen, welchen Zweck sie erfüllen.

- a) SELECT Name, entdeckt  
FROM Mond  
WHERE entdeckt = '1610'
- b) SELECT Masse, Deimos  
FROM Mond
- c\*) SELECT Name,  
(Halbachse/2) AS R  
FROM Mond
- d\*) SELECT COUNT(\*), AVG(\*)  
FROM Mond  
GROUP BY Planet

**AUFGABEN**



38010-02

benötigte Tabelle:  
Mond

 2 Die Tabelle Basketball enthält alle Spieler der Schulmannschaft des Turing-Gymnasiums. Beantworte die Fragen mithilfe einer geeigneten SQL-Abfrage.

Basketball									
Name	Vorname	Klasse	Lebensalter	Größe	1Pkt	2Pkt	3Pkt	R	Block
Ehrlicher	David	10a	16	182.3	7	14	3	11	9
Daecher	Christian	10b	15	178.9	5	5	8	2	0

- a) Gibt es Spieler, die exakt gleich groß sind?
- b\*) Welche Spieler haben die meisten Punkte erzielt?
- c\*) Für welche Klasse ist der Durchschnitt der erzielten 3er am größten?
- d\*) Welches Durchschnittsalter haben die Spieler in jeder Klasse?



38010-03

benötigte Tabelle:  
Basketball

R steht für die Anzahl der Rebounds

Fortsetzung:

↳ 1.9 Aufgabe 2

**RÜCKBLICK**

### Am Ende dieses Unterkapitels hast du wiederholt, ...

- ▶ Tabellen zu erstellen und Daten in Tabellen zu speichern.
- ▶ SQL-Abfragen zu erstellen.

## EINSTIEG

Die Mensa der Schule hat eine Mittagskarte. Jeder Schüler oder jede Schülerin kann sich ein 3-Gänge-Menü aus Vorspeise, Hauptgang und Nachtisch zusammenstellen.

Vorspeise		Hauptgang		Nachtisch	
Name	Preis	Name	Preis	Name	Preis
Eis	2,50	Spaghetti Bolognese	3,50	Pudding	1,50
Lauchsuppe	1,00	Pizza Margherita	5,50	Eis	2,50
Salat	2,00	Pizza Tonno	6,50		

- ▶ Bestimme die Anzahl aller möglichen 3-Gänge-Menüs.
- ▶ Beschreibe das Problem bei der Berechnung des Gesamtpreises für ein Menü.

## ERARBEITUNG

## Abfragen über mehrere Tabellen

In einer SQL-Abfrage können nach FROM auch mehrere Tabellen angegeben werden:  
FROM <Tabellennamen1>, <Tabellennamen2>, ...



38010-04

benötigte Tabellen:  
Vorspeise, Hauptgang,  
Nachtisch

**A1 Abfragen über mehrere Tabellen erstellen**

- Erkläre den Aufbau der Datensätze der Ergebnistabellen.  

```
SELECT * FROM Vorspeise, Hauptgang
SELECT * FROM Hauptgang, Nachtisch
SELECT * FROM Vorspeise, Nachtisch
```
- Beschreibe, welches Problem bei folgender Abfrage auftritt:  

```
SELECT * FROM Vorspeise, Hauptgang, Nachtisch
WHERE Name = 'Eis'
```
- Führe folgende Abfrage durch und erkläre den Unterschied zu b).  

```
SELECT * FROM Vorspeise, Hauptgang, Nachtisch
WHERE Nachtisch.Name = 'Eis'
```

Statt Kreuzprodukt (wegen der Schreibweise mit einem Kreuz) wird oft auch der Name kartesisches Produkt verwendet.

Werden nach FROM bei einer SQL-Abfrage mehrere Tabellen angegeben, enthält die Ergebnistabelle das **Kreuzprodukt** der Tabellen. Schreibweise:

<Tabelle1> × <Tabelle2> × ...

Dabei wird jeder Datensatz einer Tabelle mit jedem der anderen kombiniert.

Enthalten die beteiligten Tabellen Spalten gleichen Namens, so können sie mithilfe der **Punktschreibweise** unterschieden werden (z. B. `Vorspeise.Name`).

Die Angabe von Spaltennamen nach SELECT ermöglicht zusätzlich eine **Projektion**. Die SQL-Abfrage

```
SELECT Vorspeise.Name
       Nachtisch.Name
FROM   Vorspeise, Nachtisch
```

liefert also die rechts abgebildete Tabelle.

Vorspeise.Name	Nachtisch.Name
Eis	Pudding
Eis	Eis
Lauchsuppe	Pudding
Lauchsuppe	Eis
Salat	Pudding
Salat	Eis

## Das Kreuzprodukt in funktionaler Schreibweise

Das Kreuzprodukt ist eine Funktion mit variabler **Stelligkeit**. Sowohl die Eingaben als auch die Ausgabe sind Tabellen. Schreibweise:

Ergebnistabelle = Kreuzprodukt(<Tabelle1>; <Tabelle2>; ...)

**A2 Das Kreuzprodukt funktional schreiben**

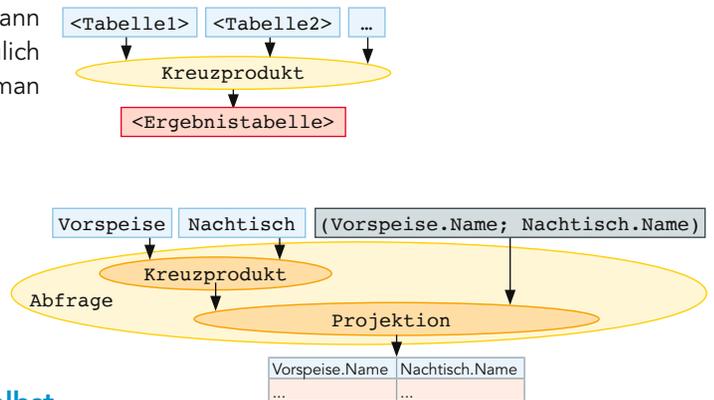
- Gib die SQL-Abfragen aus **A1a**) in funktionaler Schreibweise an.
- Schreibe die letzte Abfrage im Abschnitt „Abfragen über mehrere Tabellen“ als **Verkettung** der Funktionen **Kreuzprodukt** und **Projektion**.

### Das Kreuzprodukt als Datenflussdiagramm

Der Ablauf einer Abfrage mit dem Kreuzprodukt kann mithilfe eines Datenflussdiagramms anschaulich dargestellt werden. Aus mehreren Tabellen erhält man so eine Ergebnistabelle.

Die obige SQL-Abfrage

```
SELECT Vorspeise.Name
       Nachtisch.Name
FROM   Vorspeise, Nachtisch
wird durch das abgebildete Datenflussdiagramm veranschaulicht.
```



### Das Kreuzprodukt einer Tabelle mit sich selbst

#### A3 Das Kreuzprodukt einer Tabelle mit sich selbst erstellen

Fernanda möchte nur zwei Vorspeisen in beliebiger Reihenfolge essen.

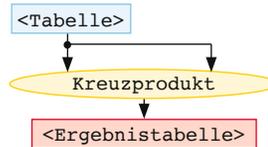
- a) Stelle fest, welche der beiden Abfragen zum Erfolg führt:
 

```
SELECT * FROM Vorspeise, Vorspeise
SELECT * FROM Vorspeise AS V1, Vorspeise AS V2
```
- b) Gib auch eine SQL-Abfrage an, die alle Kombinationen aus zwei Hauptspeisen (zwei Nachtischen) liefert. Lasse nur die Namen der Gerichte anzeigen.



38010-04  
benötigte Tabellen:  
Vorspeise, Hauptgang,  
Nachtisch

Möglich ist auch ein Kreuzprodukt einer Tabelle mit sich selbst. Dabei müssen der Tabelle mit AS zwei unterschiedliche Namen zugewiesen werden. Bei einer Projektion ist dann die Punktschreibweise zu verwenden, z. B.: V1.Name



Die SQL-Abfrage `SELECT * FROM <Tabelle1>, <Tabelle2>, ...` liefert das **Kreuzprodukt** der angegebenen Tabellen. Eventuell gleiche Spaltennamen können mithilfe der Punktschreibweise unterschieden werden.

**MERKE**

- 1 12 Mädchen und 12 Jungen nehmen an einem Tanzkurs teil. Die Kursleitung speichert sie in den Tabellen `Mädchen(Name: Text; Vorname: Text; Alter: Zahl; ...)` und `Junge(Name: Text; Vorname: Text; Alter: Zahl; ...)`. Während des Kurses soll es jedes Tanzpaar (Mädchen|Junge) einmal geben.
  - a) Gib eine entsprechende SQL-Abfrage an (nur mit Vornamen).
  - b) Schreibe die Abfrage funktional und zeichne das Datenflussdiagramm.

**AUFGABEN**

- 2 Bei Sportwettbewerben werden oft Gruppen (Tabellenschema: `GruppeA (ID: Zahl; Name: Text; ...)`) gebildet. Es spielt dann jede Mannschaft gegen jede.
  - a) Erstelle das Kreuzprodukt von `GruppeA` mit sich selbst. Begründe, welche Paarungen im Sachzusammenhang nicht sinnvoll sind.
  - b) Begründe, ob Hin- und Rückspiel enthalten sind. Gib Beispiele an, bei denen Rückspiele ausgetragen werden, und welche, wo das nicht der Fall ist.

Beispielgruppe eines Tennisturniers:

GruppeA		
ID	Name	...
1	TC Aufschlag	
2	TV Netzkante	
3	RW Auslinie	
4	TC Seitenaus	

**RÜCKBLICK**

### Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ das Kreuzprodukt von Tabellen zu erstellen.

## EINSTIEG

Am Elternabend soll eine Liste ausliegen, aus der für alle Schulklassen deren Klassenleiterin oder Klassenleiter einschließlich deren Fächerkombination ersichtlich ist. Die Daten sind in den folgenden Tabellen gespeichert.

Schulklasse			Lehrkraft					
Klasse	Zimmer	Klassenleiter	Kürzel	Name	Vorname	...	Fach1	Fach2
9f	106	vog	bab	Bauer	Britta	...	D	G
10a	210	bab	vog	Vogel	Hilmar	...	M	INF
...			...					

- ▶ Begründe, warum die folgende Abfrage nicht die gewünschte Tabelle liefert.  

```
SELECT Klasse, Name, Fach1, Fach2
FROM Schulklasse, Lehrkraft
```
- ▶ Beschreibe, woran man die im Sachzusammenhang richtigen Datensätze erkennt.

## ERARBEITUNG

Da der Begriff Klasse in der Schule und in der Informatik unterschiedliche Bedeutung hat, wird hier für Klassen in der Schule der Begriff Schulklasse verwendet.

## Fremdschlüssel

Um in der Tabelle `Schulklasse` deutlich zu machen, welche Lehrkraft Klassenleiterin oder Klassenleiter ist, wird das Attribut `Klassenleiter` eingefügt. Da dieses Attribut nur Werte des Primärschlüssels `Kürzel` enthält, nennt man `Klassenleiter` einen **Fremdschlüssel**. Fremdschlüssel werden gestrichelt überstrichen, z. B.:  
`Schulklasse(Klasse: Text; Zimmer: Zahl; Klassenleiter: Text)`

Schulklasse			Lehrkraft			
Klasse	Zimmer	<u>Klassenleiter</u>	Kürzel	Name	Vorname	...
9f	106	vog	bab	Bauer	Britta	...
10a	210	bab	vog	Vogel	Hilmar	...

**Beachte:** Die Namen von Fremd- und dem zugehörigen Primärschlüssel müssen nicht übereinstimmen, jedoch der Datentyp.

Hinweis:

Da ein Primärschlüssel mehrere Attribute enthalten kann, muss ein sich darauf beziehender Fremdschlüssel ebenso viele Attribute enthalten. Daher wird in den Beispielen wegen der besseren Übersicht meist ein künstlicher Primärschlüssel eingefügt.

In vielen Datenbanken kann man dem  $\rightarrow$  DBMS bekanntmachen, dass ein Attribut ein Fremdschlüssel ist und den zugehörigen Primärschlüssel angeben. Es wird dann automatisch sichergestellt, dass beim Fremdschlüssel kein Attributwert eingetragen werden kann, der nicht im zugehörigen Primärschlüssel vorkommt.

## A1 Fremdschlüssel erkennen

Kennzeichne bei den folgenden Tabellenschemata jeweils den Fremdschlüssel und gib an, auf welchen Primärschlüssel er sich bezieht.

- `Planet(Name: Text; Halbachse: Zahl; ...)`  
`Mond(Name: Text; Planet: Text; Halbachse: Zahl, ...)`
- `Stadt(Name: Text; Einwohner: Zahl; liegtIn: Text; ...)`  
`Land(Name: Text; Einwohner: Zahl; Staatsform: Text; ...)`  
`Fluss(Name: Text; Länge: Zahl; entspringtIn: Text; mündetIn: Text; ...)`
- `Person(Name: Text; Anschrift: Text; arbeitetIn: Text; Position: Text; ...)`  
`Firma(Name: Text; Anschrift: Text; Mitarbeiter: Zahl; ...)`  
`Position(Bezeichnung: Text; Gehalt: Zahl; ...)`

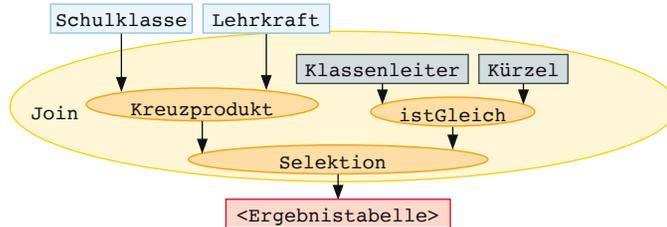
## Join

Bei der Zusammenstellung des Menüs waren alle Datensätze des Kreuzprodukts sinnvoll und erwünscht, das ist in der Liste für den Elternabend jedoch nicht so.

**A2 Objektbeziehungen erkennen**

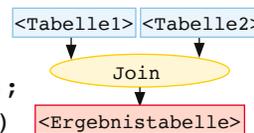
- a) Beschreibe, woran die im Sachzusammenhang richtigen Datensätze des Kreuzprodukts der Tabellen **Schulklasse** und **Lehrkraft** erkannt werden können.
- b) Formuliere eine Abfrage, welche die gewünschte Liste liefert.

Um nur die im Sachzusammenhang richtigen Datensätze des Kreuzprodukts zu erhalten, muss die Funktion Kreuzprodukt mit einer **Selektion** verkettet werden. In diesem Fall müssen die Attributwerte des Fremdschlüssels und die des Primärschlüssels übereinstimmen. Dies nennt man einen **Join**.



Join kann als neue Funktion angesehen werden:

```
Join(<Tabelle1>; <Tabelle2>) =
Selektion(Kreuzprodukt(<Tabelle1>; <Tabelle2>);
    <Primärschlüssel> = <Fremdschlüssel>)
```



Um die Anzahl der Datensätze, die ein Join geliefert hat, weiter einzuschränken, können zusätzliche Bedingungen angegeben werden, z. B.:  
**WHERE Klassenleiter = Kürzel AND Fach1 = 'D'**

Unter einem **Fremdschlüssel** versteht man ein Attribut, das Werte des Primärschlüssels einer (meist anderen) Tabelle enthält. Wird bei Abfragen, die ein Kreuzprodukt erzeugen, zusätzlich die Bedingung **<Tabelle1.Primärschlüssel> = <Tabelle2.Fremdschlüssel>** angegeben, spricht man von einem **Join**.



38010-01

benötigte Tabellen:  
Schulklasse, Lehrkraft

Hinweis:  
Es gibt unterschiedliche Arten von Joins. Da in diesem Buch nur eine Variante besprochen wird, wird dieser als Join ohne weitere Namenszusätze bezeichnet.

Gilt <Tabelle1> = <Tabelle2>, so spricht man von einem Self Join.

**MERKE**

**1** Gib jeweils den Fremdschlüssel an und formuliere einen Join.

- a)

Schüler					Schulklasse	
Id	Name	Vorname	Klasse	GebDatum	Klasse	Leiter
101	Meister	Bob	9b	2000-08-01	8a	vog
- b)

Stadt						Bundesland	
AGS	Name	BL	Vorwahl	Einwohner	HöheNN	Abk	Name
9761000	Augsburg	BY	0821	295000	494	HE	Hessen
- c)

Bestellung						Kunde	
BestNr	Name	Preis	ArtNr	Kunde	Datum	KNr	Name
9305	Toaster	21,00	6034	5023	2020-12-08	4515	Xylos

**AUFGABEN**

**AGS = Amtlicher  
Gemeindeschlüssel**

**2** Die Tabelle **Mond** enthält einige Monde, die Tabelle **Planet** alle bisher bekannten Planeten und Kleinplaneten des Sonnensystems.

- a) Erstelle eine Tabelle, welche die Namen aller (Klein)Planeten und ihrer Monde enthält sowie für beide deren Durchmesser.
- b\*) Ergänze die Tabelle aus a) um das Verhältnis Durchmesser Mond zu Durchmesser Planet und sortiere sie nach diesem Verhältnis absteigend.



38010-02

benötigte Tabellen:  
Planet, Mond

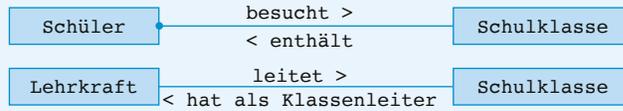
**RÜCKBLICK**

**Am Ende dieses Unterkapitels hast du gelernt, ...**

- ▶ Tabellen mit Hilfe von Fremdschlüsseln und Primärschlüsseln zu verknüpfen.

EINSTIEG

Niels ist in der Software-AG und hat sich folgende Notizen gemacht:



- ▶ Beschreibe mögliche Überlegungen, die Niels bei diesen Diagrammen hatte. Gib die Bedeutung der Zeichen < und > an.
- ▶ Erkläre die Funktion des Punkts im ersten Diagramm. Formuliere die untere Beziehung in beiden Richtungen in jeweils einem Satz, der mit „Ein(e) ...“ beginnt.

ERARBEITUNG

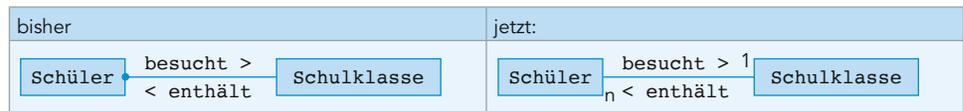
Beziehungen und Kardinalitäten

Bestehende Beziehungen zwischen Klassen können mit Sätzen beschrieben werden, die mit „Ein(e) ...“ beginnen. Beispiel für *Schulklasse* und *Schüler*:

Ein Schüler besucht *genau eine* Schulklasse.

Eine Schulklasse enthält *beliebig viele* Schüler.

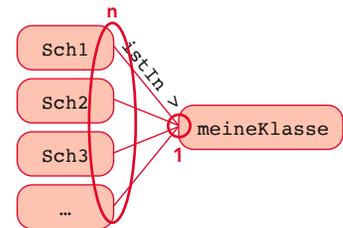
Bisher wurde im  $\hookrightarrow$  Klassendiagramm „beliebig viele“ mit einem dicken Punkt gekennzeichnet, dies geschieht nun durch die Variable *n*, „genau ein“ entsprechend mit 1.



„beliebig viele“ stimmt im Sachzusammenhang natürlich nicht, das ist aber für die Modellierung unerheblich.

- Bedeutungen:
- n: kein oder ein oder beliebig viele
  - 1: genau ein
  - 0..1: kein oder ein

Diese Angaben nennt man **Kardinalität**. *n* bedeutet, dass *n* Objekte der Klasse *Schüler* mit einem Objekt der Klasse *Schulklasse* in Beziehung stehen können. In diesem Fall spricht man von einer **1:n-Beziehung**.



A1 1:n-Beziehungen analysieren

- Suche weitere Paare von Klassen, z. B. aus dem Umfeld der Schule, zwischen denen eine 1:n-Beziehung besteht, und zeichne jeweils das Klassendiagramm mit den jeweiligen Kardinalitäten.
- Ein Zug besteht aus vielen Waggons, ein Waggon ist (zu einem bestimmten Zeitpunkt) Teil von genau einem oder keinem Zug. Zeichne das Objektdiagramm für einen Zug aus drei Waggons und das Klassendiagramm der Klassen *Waggon* und *Zug* einschließlich der Kardinalitäten.

„erster“ wird im Folgenden der Einfachheit halber weggelassen.

Zwischen den Klassen *Schüler* und *Schulklasse* gibt es eine weitere Beziehung: Genau ein Schüler ist (erster) Klassensprecher.

Ein Schüler ist Klassensprecher von *genau einer* oder *keiner* Schulklasse.

Eine Schulklasse hat *genau einen* Klassensprecher.

Als Klassendiagramm:



Dies nennt man eine **1:1-Beziehung**. Um anzudeuten, dass nicht jeder Schüler Klassensprecher ist, steht bei der Schulklasse 0..1.

Oft wird bei einer Beziehung nur die Bezeichnung für eine Richtung angegeben. Die für die andere Richtung ergibt sich dann aus dem Sachzusammenhang.

**A2 1:1-Beziehungen analysieren**

- a) Suche weitere Paare von Klassen z. B. aus dem Umfeld der Schule, zwischen denen eine 1:1-Beziehung besteht, und zeichne jeweils das Klassendiagramm mit Kardinalitäten.
- b) Zeichne für deine Schulklasse jeweils das Objektdiagramm für die Beziehungen `istKlassensprecher` und `istKlassenleiter`.

Auch Beziehungen, bei denen an einer oder beiden Seiten 0..1 steht, werden als 1:1-Beziehung bezeichnet.

**Die n:m-Beziehung**

Zwischen `Lehrkraft` und `Schüler` gibt es die Beziehung `unterrichtet`:

Eine Lehrkraft unterrichtet *beliebig viele* Schüler.  
 Ein Schüler wird unterrichtet von *beliebig vielen* Lehrkräften.



Die Angabe n (oder m) schließt den Fall n=0 mit ein. Soll deutlich gemacht werden, dass dieser Fall nicht erlaubt ist, so schreibt man auch 1..n.

Das nennt man eine **n:m-Beziehung**. Die Bezeichnungen n und m sollen deutlich machen, dass die Anzahl bei `Lehrkraft` und `Schüler` unterschiedlich sein kann.

**A3 n:m-Beziehungen analysieren**

- a) Suche weitere Paare von Klassen z. B. aus dem Umfeld der Schule, zwischen denen eine n:m-Beziehung besteht, und zeichne jeweils das Klassendiagramm.
- b) Lilly fährt gerne Fahrrad und Skateboard. Zum Fahrradfahren benötigt sie das Fahrrad und den Helm, zum Skateboardfahren den Helm, die Knieschoner und das Skateboard. Ordne die Daten und zeichne die Objektdiagramme und das Klassendiagramm für die Beziehung `wirdBenötigtFür`.

Hinweis:  
 Fasse Fahrradfahren und Skateboardfahren jeweils als ein Objekt auf.

Die **Kardinalität** gibt an, wie viele Objekte dieser Klasse mit einem Objekt der anderen Klasse in Beziehung stehen können. n bzw. m stehen für eine beliebige Anzahl von Objekten (auch keines) und die Zahl 1 für genau ein Objekt. Man unterscheidet die Beziehungstypen: **1:1-**, **1:n-** und **n:m-Beziehung**.

**MERKE**

- 1 Zeichne zu den folgenden Klassenpaaren ein Klassendiagramm mit entsprechenden Kardinalitäten. Benenne die Beziehungen in beide Richtungen sinnvoll. Formuliere die Beziehung in beide Richtungen jeweils in einem Satz, der mit „Ein(e) ...“ beginnt.

- a) Lehrkraft – Fach
- b) Museum – Kunstwerk
- c) Zimmer – Stuhl
- d) Land – Hauptstadt
- e) Land – Stadt
- f) Land – Fluss

**AUFGABEN**

- 2 Das Diagramm zeigt einen Teil der Datenbank einer Fluggesellschaft.

- a) Diskutiert in der Gruppe über mögliche logische Fehler.
- b) Zeichnet ein neues Diagramm ohne diese Fehler.



**RÜCKBLICK**

**Am Ende dieses Unterkapitels hast du gelernt, ...**

- ▶ Kardinalitäten von Beziehungen und Beziehungstypen zu bestimmen.
- ▶ Beziehungen zwischen Klassen zu erkennen.

EINSTIEG

Xenia betrachtet nun diese Beziehung zwischen den Klassen **Schüler** und **Schulklasse**.

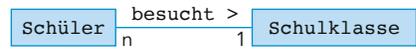
Schüler				Schulklasse		
SId	Name	Vorname	...	Klasse	Zimmer	...
101	Blackett	Patrick		9f	106	
102	Bohr	Aage		10a	210	
103	Boyle	Willard		10b	211	
104	Brattain	Walter		10c	212	
...						

- ▶ Beschreibe, welche Art von Beziehung Xenia mit den Linien umgesetzt hat.
- ▶ Erkläre, warum es nicht sinnvoll ist, auf die Tabelle **Schulklasse** zu verzichten und alle Angaben in die Tabelle **Schüler** zu schreiben.

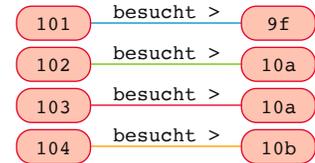
ERARBEITUNG

Objektbeziehungen und Fremdschlüssel

Klassendiagramme beschreiben, welche Beziehungen zwischen Klassen (Tabellen) bestehen (hier: dass Schüler in genau einer Schulklasse sind).



Damit ist aber nicht ersichtlich, welcher Schüler in welcher Schulklasse ist. Dies wird durch Objektdiagramme dargestellt und muss mithilfe von Fremdschlüsseln (→ 1.3) in den entsprechenden Tabellen deutlich gemacht werden.



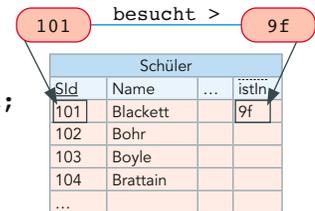
Umsetzung einer 1:n-Beziehung

1. Die Klasse auf der „n-Seite“ (hier: **Schüler**) bekommt eine neue Spalte als Fremdschlüssel (hier: **istIn**).

Schüler				Schulklasse	
SId	Name	...	istIn	Klasse	Zimmer
101	Blackett			9f	106
102	Bohr			10a	210
103	Boyle			10b	211
104	Brattain			10c	212
...					

2. In diese Spalte werden Werte des Primärschlüssels der „1-Seite“ (hier **Klasse**) eingetragen.

**Tabellenschema:** **Schüler**(SId: Zahl; Name: Text; ...; istIn: Text)



Hinweis:  
Der Fremdschlüssel muss nicht denselben Namen haben wie der zugehörige Primärschlüssel, jedoch denselben Datentyp.

A1 1:n-Beziehungen umsetzen

- Übertrage die Tabelle **Schüler** in dein Heft und ergänze die restlichen Einträge des Fremdschlüssels unter Beachtung der Objektdiagramme im obigen Abschnitt **Objektbeziehungen und Fremdschlüssel**.
- Beschreibe die auftretenden Probleme, wenn der Fremdschlüssel in die Tabelle auf der „1-Seite“ (hier: **Schulklasse**) eingetragen wird.

Implementierung einer 1:1-Beziehung

Bei 1:1-Beziehungen kann der Fremdschlüssel in jeder der beiden Tabellen eingefügt werden. Ist eine der Kardinalitäten wie in diesem Beispiel 0..1, ist es besser den Fremdschlüssel auf der 0..1-Seite einzufügen:

Jede Schulklasse hat einen (ersten) Klassensprecher. Ein Fremdschlüssel in der Tabelle **Schüler** hätte bis auf einen Schüler pro Klasse keinen Eintrag.



Viele fehlende Attributwerte sind oft ein Hinweis auf eine schlechte Struktur der Datenbank.

## Abfragen bei einer 1:n-Beziehung

SQL-Abfragen über mehrere Tabellen, die miteinander in Beziehung stehen, müssen so strukturiert sein, dass sie die richtigen Objektbeziehungen beachten. Oft geschieht das mithilfe von Joins (→ 1.3).

### A2 Abfragen über zwei Tabellen einer 1:n-Beziehung erstellen

- a) Beschreibe, welche Information mit der SQL-Abfrage gewonnen werden sollen:
- ```
SELECT  Schüler.Name, Schulklasse.Klassenleiter
FROM    Schüler, Schulklasse
WHERE   Schüler.istIn = Schulklasse.Klasse
```
- b) Erstelle eine Abfrage, die jedem Schülernamen die Zimmernummer seiner Schulklasse zuordnet.

Befinden sich die gesuchten Informationen in beiden Tabellen einer 1:n-Beziehung, so wird der Join durch die **Bedingung <Fremdschlüssel auf der n-Seite> = <Primärschlüssel der 1-Seite>** hergestellt.

Beziehungen zwischen Objekten werden mithilfe von Fremdschlüsseln eingefügt. Bei einer 1:n-Beziehungen wird der Fremdschlüssel in die Tabelle mit der Kardinalität n eingetragen. Bei einer 1:1-Beziehung ist die Wahl der Tabelle beliebig, sollte jedoch im Sachzusammenhang sinnvoll sein.

Bei Abfragen über Tabellen, die über eine 1:n- oder eine 1:1-Beziehung miteinander verknüpft sind, muss ein Join erstellt werden:

```
WHERE <Tabelle1.Primärschlüssel> =
      <Tabelle2.FremdschlüsselZuTabelle1>
```

- 1 Eine Datenbank über Deutschland enthält Tabellen mit diesen Tabellenschemata:

Bundesland(Name: Text; Fläche: Zahl; Einwohner: Zahl)

Stadt(Name: Text; Einwohner: Zahl)

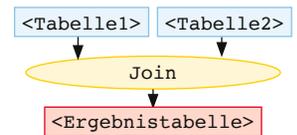
Folgende Beziehungen sollen berücksichtigt werden:

- ▶ Eine Stadt liegt in genau einem Bundesland.
- ▶ Ein Bundesland hat genau eine Hauptstadt.
- a) Gib zu jeder Klassenbeziehung vier tatsächliche Objektbeziehungen an.
- b) Erstelle zu jeder Beziehung ein Klassendiagramm mit Kardinalitäten.
- c) Gib für beide Klassen das geänderte Tabellenschema an, nachdem die Beziehungen implementiert wurden.



- d) Formuliere jeweils SQL-Abfragen, die folgende Tabellen liefern:
- ▶ alle Bundesländer (jeweils Name) mit ihren Hauptstädten (jeweils Name)
  - ▶ alle Städte, deren Einwohnerzahl mit der Einwohnerzahl des entsprechenden Bundeslandes übereinstimmt
- Erkläre die Ergebnisse im Sachzusammenhang. Begründe, warum Bremen in dieser Liste fehlt.

- e\*) Ermittle die Fläche der Bundesrepublik Deutschland.



38010-01

benötigte Tabellen:  
Schüler, Schulklassen

### MERKE

### AUFGABEN



38010-05

benötigte Tabellen:  
Stadt, Bundesland

Die Einwohnerzahlen sind in der Einheit 1000 Einwohner angegeben.

Fortsetzung: → 1.6 Aufgabe 2

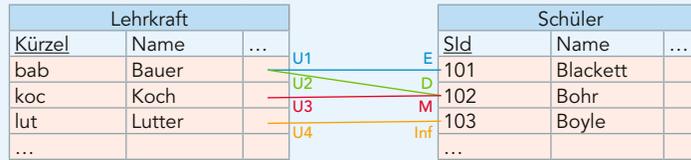
### RÜCKBLICK

### Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ 1:1- und 1:n-Beziehungen in relationalen Datenbanken umzusetzen und geeignete Abfragen zu erstellen.

EINSTIEG

Betrachte die Beziehung *unterrichtet* zwischen *Lehrkraft* und *Schüler*.



- ▶ Beschreibe, wie man anhand der Linien erkennen kann, dass es sich um eine n:m-Beziehung handelt.
- ▶ Erläutere die entstehenden Probleme, wenn ein Fremdschlüssel zu *Kürzel* in die Tabelle *Schüler* eingefügt wird.
- ▶ Erläutere die entstehenden Probleme, wenn ein Fremdschlüssel zu *Sid* in die Tabelle *Lehrkraft* eingefügt wird.

ERARBEITUNG

n:m-Beziehungen

Die Beziehung *unterrichtet* zwischen *Lehrkraft* und *Schüler* ist eine n:m- Beziehung: *Eine* Lehrkraft unterrichtet *beliebig viele* Schüler.

*Ein* Schüler wird von *beliebig vielen* Lehrkräften unterrichtet.



38010-01

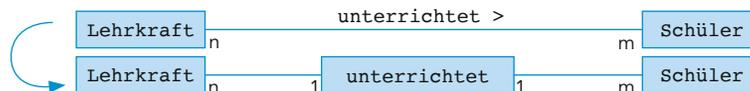
benötigte Tabellen:  
Lehrkraft, Schüler,  
unterrichtet

A1 Beziehungen zwischen Tabellen untersuchen

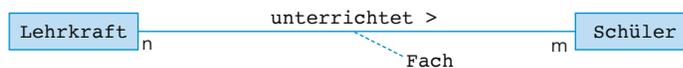
Untersuche die Tabelle *unterrichtet*.

- Gib ihr Tabellenschema schriftlich an. Kennzeichne darin Fremdschlüssel und gib die zugehörigen Primärschlüssel an.
- Beschreibe die Objektbeziehungen *U3* und *U4* aus dem Einstieg in Worten.
- Gib den Datensatz in der Tabelle *unterrichtet* an, der zu folgenden Objektbeziehungen gehört:  
Lehrkraft *bab* unterrichtet Schüler 101 im Fach G.  
Lehrkraft *lut* unterrichtet Schüler 103 im Fach M.

n:m-Beziehungen lassen sich in relationalen Datenbanken nicht direkt umsetzen, sie müssen mithilfe einer **Beziehungstabelle** in zwei 1:n-Beziehungen aufgelöst werden:



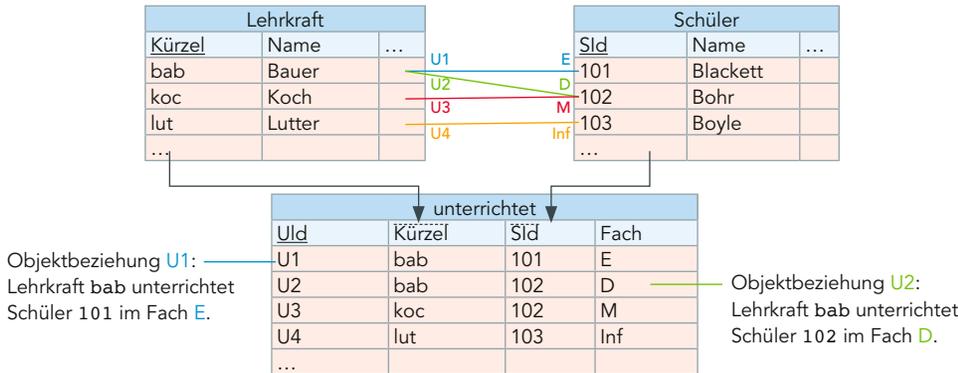
Soll die Beziehungstabelle außer den Fremdschlüsseln weitere Spalten enthalten, hier z. B. *Fach*, so wird dies oft durch eine gestrichelte Linie angedeutet:



Die Beziehungstabelle *unterrichtet* wird wie folgt aufgebaut:

- ▶ Sie enthält die Primärschlüssel von *Lehrkraft* *und* von *Schüler* als Fremdschlüssel.  
`unterrichtet(Kürzel: Text; Sid: Zahl)`
- ▶ Der Primärschlüssel ist entweder die Kombination aus den beiden Fremdschlüsseln (*Kürzel* und *Sid*) oder es wird ein neuer Primärschlüssel (z. B. *UId*) vereinbart.  
`unterrichtet(Kürzel: Text; Sid: Zahl)` oder:  
`unterrichtet(UId: Text; Kürzel: Text; Sid: Zahl)`
- ▶ Falls erforderlich werden weitere Spalten eingefügt, z. B. für das unterrichtete *Fach*.  
`unterrichtet(UId: Text; Kürzel: Text; Sid: Zahl Fach: Text)`

Jede Objektbeziehung (im folgenden Diagramm dargestellt als Verbindungslinie) wird als Datensatz in die Beziehungstabelle eingetragen.



Alternative ohne eine neue Primärschlüsselspalte:

| unterrichtet |     |      |
|--------------|-----|------|
| Kürzel       | Sid | Fach |
| bab          | 101 | E    |
| bab          | 102 | D    |
| koc          | 102 | M    |
| lut          | 103 | Inf  |
| ...          |     |      |

### Abfragen bei einer n:m-Beziehung

Zwischen jeder Tabelle, in der sich eine Information befindet und der Beziehungstabelle muss ein Join erstellt werden. Die Bedingungen sind mit AND zu verknüpfen.

#### A2 Abfragen über zwei Tabellen einer n:m-Beziehung erstellen

- Beschreibe, welche Informationen mit der SQL-Abfrage gewonnen werden sollen:  

```
SELECT Lehrkraft.Name, Schüler.Name
FROM Lehrkraft, Schüler, unterrichtet
WHERE Lehrkraft.Kürzel = unterrichtet.Kürzel
      AND unterrichtet.Sid = Schüler.Sid
```
- Erstelle mithilfe einer Abfrage eine Liste aller Lehrkräfte des Schülers Zwerger.

AND siehe ↪ logische Funktionen



38010-01

Verwende die Tabellen Lehrkraft, Schüler und unterrichtet

Bei der Implementierung einer n:m-Beziehung wird diese durch eine zusätzliche **Beziehungstabelle** in zwei 1:n-Beziehungen aufgelöst. Die beiden Bedingungen der Joins sind mit der logischen Funktion AND zu verknüpfen:

```
WHERE <Tabelle1.Primärschlüssel> =
      <Beziehungstabelle.FremdschlüsselZuTabelle1>
      AND <Tabelle2.Primärschlüssel> =
      <Beziehungstabelle.FremdschlüsselZuTabelle2>
```

**MERKE**

- Gib zu jeder Beziehung ein mögliches, zusätzliches Attribut an, das in der Beziehungstabelle gespeichert werden könnte, und erstelle deren Tabellenschema.



**AUFGABEN**

- Fortsetzung von ↪ 1.5 Aufgabe 1  
 In der Datenbank über Deutschland gibt es zusätzlich die Tabelle Fluss (Name: Text; Länge: Zahl) mit der Beziehung fließtDurch zwischen Fluss und Stadt. Erstelle jeweils eine geeignete Anfrage um folgende Fragen beantworten zu können.

- Welche Flüsse mit welcher Länge fließen durch welche Städte?
- Welche Flüsse fließen durch welche Bundesländer?



38010-05

benötigte Tabelle: Fluss, Stadt, Bundesland, fließtDurch

Fortsetzung: ↪ 1.9 Aufgabe 11

**RÜCKBLICK**

### Am Ende dieses Unterkapitels hast du gelernt, ...

- n:m-Beziehungen in relationalen Datenbanken umzusetzen und geeignete Abfragen zu erstellen.

## EINSTIEG

Sandra und Pedro von der SMV helfen mit, die Projektwoche am Dijkstra-Gymnasium zu organisieren. Sie überlegen sich, ob dafür nicht der Einsatz einer Datenbank sinnvoll wäre.

- ▶ Beschreibe einige Vorteile, die eine Datenbank gegenüber Listen bietet.
- ▶ Beschreibe einige Listen (z. B. Klassenlisten, Teilnehmerlisten der Projekte), die für die Durchführung der Projektwoche hilfreich sind.



## ERARBEITUNG

## Datenbestand erstellen

Sandra und Pedro bitten ihren Informatiklehrer, Herrn Lutter, um Rat. Er schlägt vor, zunächst einige typische Daten und Anforderungen zu notieren:

- ▶ Aage Bohr aus der 10a hat sich für die Projekte Handball und Chor gemeldet.
- ▶ Die Gitarrengruppe, an der höchstens 10 Schülerinnen oder Schüler teilnehmen können, wird von Frau Bayer im Musiksaal geleitet.
- ▶ In jedem Klassenzimmer soll eine Liste aufgehängt werden, auf der jeder Schüler nachlesen kann, an welchen Projekten er teilnimmt und wo diese stattfinden.
- ▶ Jede Projektleiterin und jeder Projektleiter bekommt eine Liste mit den Namen aller Teilnehmer seines Projekts ins Fach gelegt.
- ▶ Auf dem Sportplatz finden die Projekte Fußball, Weitwurf und Sportgymnastik statt.
- ▶ Zum Schwimmbad sind 20 Minuten Gehzeit einzuplanen, zur Turnhalle keine.

## Datenbestand analysieren und strukturieren

Dieser ungeordnete Datenbestand ist in einen objektorientiert geordneten Datenbestand zu überführen. Dazu wird er **analysiert** und **strukturiert**.

## A1 Objekte und Klassen sowie deren Beziehungen erkennen

- a) Identifiziere aus den Daten einige Objekte.
- b) Gib an, welche Klassen benötigt werden und ordne ihnen die Objekte zu.

## Objektorientiertes Datenmodell entwickeln und darstellen

Herr Lutter schlägt nun vor, aus dem Datenbestand das objektorientierte **Datenmodell** zu **entwickeln**. Dieses Modell beschreibt die Struktur der Daten und deren Beziehungen zueinander. Es kann im Klassendiagramm übersichtlich **dargestellt** werden.

## A2 Objektorientiertes Datenmodell darstellen

- a) Zeichne alle Klassenkarten mit einigen wichtigen Attributen.
- b) Zeichne das Klassendiagramm.

## Objektorientiertes Datenmodell in ein relationales Modell überführen

Im nächsten Schritt muss das objektorientierte Datenmodell in ein **relationales Modell überführt** werden. Dieses legt fest:

- ▶ die Spalten der Tabellen und deren Datentypen
  - ▶ die Primärschlüsselattribute, evtl. neu einzuführende künstliche Schlüssel
  - ▶ zur Realisierung der Beziehungen nötige Fremdschlüssel und Beziehungstabellen
- Dazu sollen Sandra und Pedro die Tabellenschemata aller Tabellen angeben.

Alle Arbeitsaufträge in diesem Kapitel bauen aufeinander auf, sollten also vollständig und in der angegebenen Reihenfolge bearbeitet werden.

Beispiel:

| Schüler     |
|-------------|
| Name        |
| Vorname     |
| Schulklasse |
| ...         |

**A3 Relationales Modell erstellen**

- Begründe die im relationalen Modell zusätzlich benötigten Klassen.
- Lege geeignete Attribute als Primärschlüssel fest oder führe evtl. künstliche ein.
- Gib die Fremdschlüssel an und auf welche Primärschlüssel sie sich beziehen.
- Gib alle Tabellenschemata an.

Beispiel:  
Schüler(SId: Zahl; ...)

**Relationales Modell in einem relationalen DBS umsetzen**

Jetzt können Sandra und Pedro in einem DBS Tabellen anlegen und Daten eingeben. Dieser Vorgang wird als **Umsetzung** oder **Implementierung** bezeichnet.

DBS

➔ Datenbanksystem

**SQL-Abfragen konzipieren**

Sandra und Pedro **konzipieren** benötigte SQL-Abfragen, um Listen durch SQL-Abfragen zu erstellen. So gewinnen sie aus vorhandenen Daten zielgerichtet **neue Informationen**. Schlägt das ganz oder teilweise fehl, müssen sie das Modell nachbessern.

**A4 SQL-Abfragen konzipieren**

- Gib einige SQL-Abfragen an, die sich aus den obigen Anforderungen ergeben.
- Beschreibe, welche neuen Informationen man aus der Datenbank gewinnen könnte und konzipiere jeweils eine geeignete SQL-Abfrage.

Der Aufbau einer Datenbank erfolgt in diesen Schritten:

- ▶ Datenbestand **analysieren** und **strukturieren**
- ▶ **objektorientiertes Datenmodell entwickeln** und grafisch **darstellen**
- ▶ objektorientiertes Datenmodell in ein **relationales Modell überführen**
- ▶ relationales Modell in einem Datenbanksystem **umsetzen**

Anschließend können geeignete SQL-Abfragen **konzipiert** werden, um **neue Informationen** zu gewinnen.

**MERKE**

- 1 Beschreibe jeweils, wo und welche Änderungen Sandra und Pedro vornehmen müssen, wenn sich nachträglich folgende Änderungen ergeben.

- Es soll ersichtlich sein, ob ein Projekt draußen oder drinnen stattfindet.
- Für einige Projekte werden mehrere Projektleiter eingeteilt.
- Für jedes Projekt wird eine Mindestteilnehmerzahl festgelegt. Gibt es weniger Interessenten, wird es abgesagt und die Personen müssen umwählen.



- 2 Findet euch in Gruppen zusammen und bearbeitet analog dem oben beschriebenen Vorgehen das Problem: Organisation einer Klassenfeier.

Berücksichtigt folgende Punkte und ergänzt sie um eigene Ideen.

- ▶ Jeder arbeitet in einer Gruppe an einer Aufgabe, z. B. Getränke besorgen, Raum schmücken, Geschirr bereitstellen, Musik, usw.
- ▶ Lebensmittel und Gegenstände, die nicht geliehen werden können, müssen gekauft werden. Zum Schluss bezahlen alle einen Anteil.

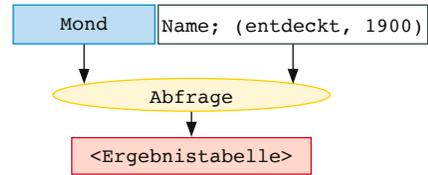
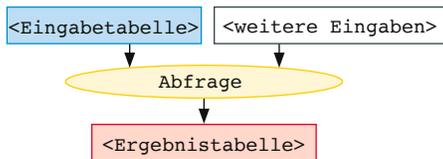
**AUFGABEN****RÜCKBLICK****Am Ende dieses Unterkapitels hast du gelernt, ...**

- ▶ einen Datenbestand mithilfe der Datenmodellierung in eine relationale Datenbank umzusetzen.
- ▶ mithilfe von SQL-Abfragen neue Informationen zu gewinnen.



**SQL-Abfrage** ↪ 1.1

Jede SQL-Abfrage kann als mehrstellige Funktion interpretiert werden. Die Funktion wählt Daten aus der Tabelle aus und berechnet gegebenenfalls neue Daten. Diese werden wieder als Tabelle ausgegeben. Welche Daten die Ergebnistabelle enthält und wie sie dargestellt werden, hängt von weiteren Eingaben ab wie z. B. der Auswahl und Sortierung von Spalten oder Bedingungen.



**Kreuzprodukt** ↪ 1.2

Unter dem Kreuzprodukt  $A \times B$  zweier Mengen A und B versteht man die Menge aller geordneten Paare  $(a|b)$  mit  $a \in A$  und  $b \in B$ .

Beispiel:

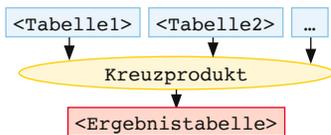
$A = \{x; y; z\}; B = \{1; 2\}$   
 $\Rightarrow A \times B = \{(x|1); (y|1); (z|1); (x|2); (y|2); (z|2)\}$

In relationalen Datenbanken entsprechen die Datensätze in Tabellen den Elementen.

funktionale Schreibweise:

Kreuzprodukt (<Tabelle1>;  
 <Tabelle2>; ...)

als Datenflussdiagramm:



| Vorspeise  |       | Nachtsch |       |
|------------|-------|----------|-------|
| Name       | Preis | Name     | Preis |
| Eis        | 2,50  | Pudding  | 1,50  |
| Lauchsuppe | 1,00  | Pudding  | 1,50  |
| Salat      | 2,00  | Eis      | 2,50  |

| Vorspeise x Nachtsch |                 |                 |                  |
|----------------------|-----------------|-----------------|------------------|
| Vorspeise.Name       | Vorspeise.Preis | Nachspeise.Name | Nachspeise.Preis |
| Eis                  | 2,50            | Pudding         | 1,50             |
| Lauchsuppe           | 1,00            | Pudding         | 1,50             |
| Salat                | 2,00            | Pudding         | 1,50             |
| Eis                  | 2,50            | Eis             | 2,50             |
| Lauchsuppe           | 1,00            | Eis             | 2,50             |
| Salat                | 2,00            | Eis             | 2,50             |

**Fremdschlüssel** ↪ 1.3

Enthält eine Tabelle eine Spalte, in der ausschließlich Werte des Primärschlüssels einer meist anderen Tabelle stehen, so spricht man von einem Fremdschlüssel. Damit kann ein Bezug zwischen den in den Tabellen gespeicherten Objekten hergestellt werden.

Fremdschlüssel werden in diesem Buch gestrichelt überstrichen.

| Schulklasse |        |               | Lehrkraft |       |         |     |
|-------------|--------|---------------|-----------|-------|---------|-----|
| Klasse      | Zimmer | Klassenleiter | Kürzel    | Name  | Vorname | ... |
| 9f          | 106    | vog           | bab       | Bauer | Britta  | ... |
| 10a         | 210    | bab           | vog       | Vogel | Hilmar  | ... |



**SQL-Abfragen** ↪ 1.1

Eine SQL-Abfrage beginnt mit **SELECT** und hat folgenden Aufbau:

```
SELECT    <Attribut1>,
          <Attribut2>, ...
FROM      <Tabellename>
WHERE     <Bedingung>
GROUP BY  <Attribut>
ORDER BY  <Attribut> [ASC|DESC]
```

Es müssen nicht immer alle Schlüsselwörter vorhanden sein, die Reihenfolge darf jedoch nicht verändert werden. In den Zeilen können noch weitere Angaben dazukommen.

Eine SQL-Abfrage kann als eine mehrstellige Funktion interpretiert werden, ihre Ausgabe ist eine Tabelle. Diese Tabelle wird jedoch nicht dauerhaft gespeichert!

```
SELECT    Name, entdeckt
FROM      Mond
WHERE     entdeckt > 1900
ORDER BY  Durchmesser
```

| Name   | entdeckt |
|--------|----------|
| Styx   | 2012     |
| Mab    | 2003     |
| Emapus | 2000     |
| Cupid  | 2003     |
| ...    |          |

```
Abfrage(Mond;
        (Name, entdeckt, 1900))
```

**Abfragen über mehrere Tabellen, Kreuzprodukt** ↪ 1.2

Bei einer SQL-Abfrage können hinter **FROM** mehrere Tabellennamen angegeben werden:

```
SELECT *
FROM    <Tabelle1>, <Tabelle2>, ...
```

Die Ergebnistabelle enthält dann das Kreuzprodukt der angegebenen Tabellen.

```
SELECT *
FROM    Vorspeise, Nachtisch
liefert die Tabelle:
```

| Vorspeise. Name | Vorspeise. Preis | Nachspeise. Name | Nachspeise. Preis |
|-----------------|------------------|------------------|-------------------|
| Eis             | 2,50             | Pudding          | 1,50              |
| Lauchsuppe      | 1,00             | Pudding          | 1,50              |
| Salat           | 2,00             | Pudding          | 1,50              |
| Eis             | 2,50             | Eis              | 2,50              |
| Lauchsuppe      | 1,00             | Eis              | 2,50              |
| Salat           | 2,00             | Eis              | 2,50              |

**Fremdschlüssel, Join** ↪ 1.3

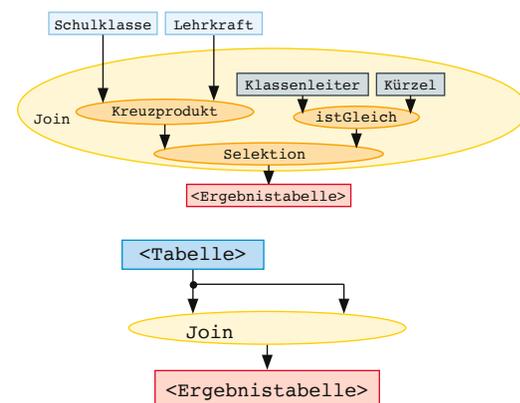
Wird in einem DBMS eine Spalte als Fremdschlüssel deklariert, so prüft es automatisch, ob jeder beim Fremdschlüssel eingetragene Wert auch im zugehörigen Primärschlüssel existiert.

Ein Join ist eine Verkettung des Kreuzprodukts mit einer Selektion mit der Bedingung

```
<Tabelle1.Primärschlüssel> =
= <Tabelle2.Fremdschlüssel>
```

Die Ergebnistabelle enthält dann nur noch die Datensätze, die den Objektbeziehungen entsprechen.

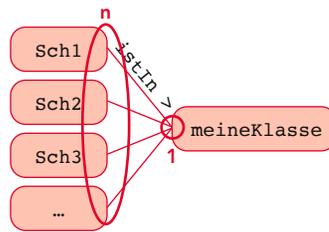
Gilt  $Tabelle1 = Tabelle2$ , so spricht man von einem Self-Join.



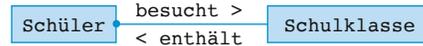


**Kardinalität** ↪ 1.4

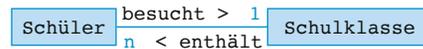
Die Kardinalität gibt an, wie viele Objekte einer Klasse mit *einem* Objekt der anderen Klasse in Beziehung stehen können. n bzw. m steht dabei für eine beliebige Anzahl von Objekten ( $n, m \in \mathbb{N}_0$ ), 1 für genau ein Objekt, 0..1 für kein oder ein Objekt. Die bisher übliche Kennzeichnung (mit oder ohne Punkt), wird nun durch Angaben wie n und 1 ersetzt.



bisher:



jetzt:



**Beziehungstypen** ↪ 1.4

Je nach Kardinalität der beteiligten Klassen unterscheidet man bei Klassenbeziehungen zwischen 1:1-, 1:n- und n:m-Beziehungen. Hinweis: Statt 1 kann jeweils auch 0..1 stehen.

1:1-Beziehung:

Jeder Datensatz einer Tabelle steht mit genau einem Datensatz einer anderen Tabelle in Beziehung.

1:n-Beziehung:

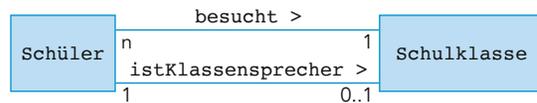
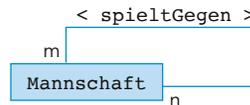
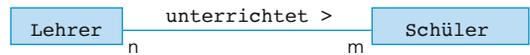
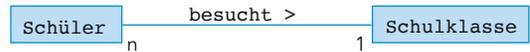
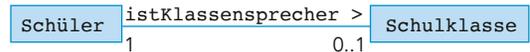
Beliebig viele Datensätze einer Tabelle stehen mit genau einem Datensatz einer anderen Tabelle in Beziehung.

n:m-Beziehung:

Beliebig viele Datensätze einer Tabelle stehen mit beliebig vielen Datensätzen einer anderen Tabelle in Beziehung.

Die beiden Tabellen müssen nicht notwendigerweise verschieden sein.

Zwischen Klassen können mehrere Beziehungen unterschiedlicher Kardinalität bestehen.



**Datenmodellierung** ↪ 1.7

So geht man vor, um aus einem ungeordneten Datenbestand eine Datenbank zu erhalten:

- ▶ Datenbestand erstellen
- ▶ Datenbestand analysieren und strukturieren
- ▶ objektorientiertes Datenmodell entwickeln und grafisch darstellen
- ▶ objektorientiertes Datenmodell in relationales Modell überführen
- ▶ relationales Modell in einem DBS umsetzen
- ▶ SQL-Abfragen konzipieren

Zuordnung zu Klassen, Beziehungen erkennen  
Klassenkarten, Klassendiagramm

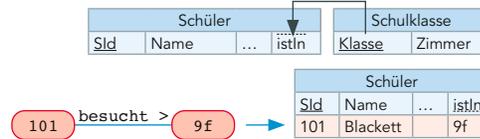
Primär- und Fremdschlüssel festlegen,  
Tabellenschemata angeben



**Umsetzung von 1:n-Beziehungen und Abfragen** ↪ 1.5

Auf der „n-Seite“ wird ein Fremdschlüssel eingefügt, der sich auf den Primärschlüssel der „1-Seite“ bezieht. In diese Spalte werden Werte des Primärschlüssels der „1-Seite“ eingetragen.

Bei Abfragen muss ein Join beider Tabellen erstellt werden.



```
SELECT Schüler.Name, Schulklasse.Name
FROM Schüler, Schulklasse
WHERE Schüler.istIn
      = Schulklasse.Klasse
```

**1:1-Beziehungen** ↪ 1.5

1:1-Beziehungen werden wie 1:n-Beziehungen implementiert und abgefragt. Meist ist es unerheblich, auf welcher Seite der Fremdschlüssel eingefügt wird.

Ist eine der Kardinalitäten 0..1, so sollte der Fremdschlüssel auf dieser Seite eingefügt werden. Sonst würden viele Attributwerte leer bleiben.

| Schulklasse |        |                 |
|-------------|--------|-----------------|
| Klasse      | Zimmer | Klassensprecher |
| 9f          | 106    | 101             |
| 10a         | 210    | 314             |
| 10b         | 211    | 512             |
| 10c         | 212    | 297             |
| ...         |        |                 |

**Umsetzung von n:m-Beziehungen und Abfragen** ↪ 1.6

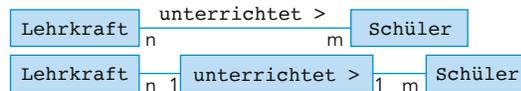
n:m-Beziehungen müssen mithilfe einer Beziehungstabelle in zwei 1:n-Beziehungen aufgelöst werden.

Die Beziehungstabelle

- ▶ enthält die Primärschlüssel der beiden beteiligten Tabellen als Fremdschlüssel.
- ▶ enthält evtl. weitere Attribute.

Der Primärschlüssel der Beziehungstabelle ist entweder die Kombination aus den beiden Fremdschlüsseln oder es wird ein neuer Primärschlüssel vereinbart.

Bei Abfragen muss ein Join sowohl zwischen der einen Tabelle und der Beziehungstabelle als auch zwischen der anderen Tabelle und der Beziehungstabelle formuliert werden. Die beiden Bedingungen sind mit AND zu verknüpfen.



```
SELECT Lehrkraft.Name, Schüler.Name
FROM Lehrkraft, Schüler
WHERE Lehrkraft.Kürzel
      = unterrichtet.Kürzel
      AND unterrichtet.Sid
      = Schüler.Sid
```

↪ 1.1



38010-06

benötigte Tabellen:  
Kunde, Buch

Fortsetzung: ↪ Aufgabe 4



1 Eine Bücherei speichert die Bücherdaten in dieser Tabelle.

Erstelle je eine SQL Abfrage, die Folgendes leistet (geforderte Attribute in Klammern).

- Alle Bücher (Titel, Jahr), die nach 1980 erschienen sind.
- Alle Bücher (vollständige Daten) von B. D. Decker.
- Die ID des Autors George von Brehm.

| Buch |                  |              |      |
|------|------------------|--------------|------|
| Id   | Titel            | Autor        | Jahr |
| 1    | Zu den Sternen   | B. D. Decker | 1982 |
| 2    | Sternestaub      | J. Isamov    | 1960 |
| 3    | Planlos im All   | B. D. Decker | 1985 |
| 4    | Ich, der Android | J. Isamov    | 1950 |
| 5    | 400jährige       | J. Isamov    | 1978 |
| 6    | Aufpasser        | B. D. Decker | 2009 |
| 7    | Sudon            | B. D. Decker | 1996 |
| ...  |                  |              |      |

↪ 1.1



38010-03

benötigte Tabelle:  
Basketball

2 Fortsetzung von ↪ 1.1, Aufgabe 2.

Erstelle geeignete SQL-Abfragen, um folgende Fragen beantworten zu können.

- Welche Spieler sind am größten?
- Welchen Spielern aus der 10a gelangen die meisten Rebounds?
- \*) Wie viele Punkte haben die einzelnen Spieler erzielt?
- \*) Wie hoch ist der Durchschnittswert der Rebounds in den einzelnen Klassen?
- Wie vielen Spielern sind mehr 3-Punktwürfe als 1-Punktwürfe gelungen?

↪ 1.2

3 Begründe, welche Datensätze durch das Kreuzprodukt der angegebenen Tabellen entstehen. Gib jeweils eine entsprechenden SQL-Abfrage an und zeichne jeweils das Datenflussdiagramm.

Diskutiere, ob erhaltenen Datensätze im Sachzusammenhang sinnvoll sind und welche evtl. fehlen.

- In Tabellenkalkulationsprogrammen werden Zelladressen als Kombination des Spaltennamens (`Spalte(Name: Text)`) und der Zeilennummer angegeben (`Zeile(Nummer: Zahl)`).
- Die Tabelle `Farbe(Name: Text)` enthält einige Farben. Mit dem Kreuzprodukt `Farbe × Farbe × Farbe` sollen alle Möglichkeiten bestimmt werden, eine Trikolore zu entwerfen.
- Die Tabelle `Zeichen(z: Zeichen)` enthält alle Buchstaben und Ziffern sowie einige Sonderzeichen. Marius möchte sich damit Vorschläge für ein achtstelliges Passwort erzeugen lassen.

Eine Trikolore ist eine Flagge aus drei verschiedenfarbigen Streifen, z. B.:



↪ 1.3

In der Aufgabe wird vorausgesetzt, dass von jedem Buch nur ein Exemplar vorhanden ist.

4 Fortsetzung von ↪ Aufgabe 1

Die Tabelle `Kunde` speichert die Kundendaten der Bücherei. In die Tabelle `Buch` wird die Spalte `ausgeliehenVon` als Fremdschlüssel eingefügt.

- Gib das geänderte Tabellenschema der Tabelle `Buch` an.
- Begründe, warum es besser ist, den Fremdschlüssel in die Tabelle `Buch` einzufügen und nicht in `Kunde`.
- Diskutiere verschiedene Möglichkeiten, wie kenntlich gemacht werden kann, dass ein Buch zur Zeit nicht ausgeliehen ist.
- Es soll eine Liste aller Kunden (Vorname und Name) erstellt werden, die gerade ein Buch ausgeliehen haben. Gib an:
  - ▶ die Funktion `Join` als Verkettung von Kreuzprodukt und Selektion
  - ▶ das Datenflussdiagramm
  - ▶ die Abfrage in SQL

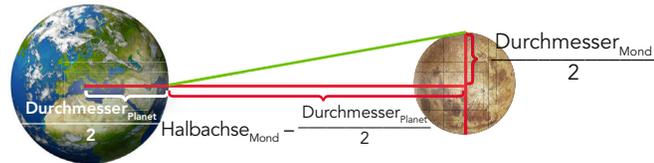
| Kunde |           |         |
|-------|-----------|---------|
| Id    | Name      | Vorname |
| 1     | Tuman     | Ibrahim |
| 2     | Liu       | Nadja   |
| 3     | Olovski   | Ingo    |
| 4     | Freund    | Hanna   |
| 5     | Offra     | Adam    |
| 6     | Von Brehm | George  |
| 7     | Mahr      | Paula   |
| ...   |           |         |

5 Beantworte jeweils mithilfe einer SQL-Abfrage die folgenden Fragen.

- a) Gibt es Monde, deren Entdeckungsjahr mit dem Planeten, den sie umkreisen, übereinstimmt?
- b) Bei welchen Monden ist die Halbachse seiner Bahn mehr als hundertmal so groß wie der Durchmesser des Planeten, den sie umkreisen?
- c) Damit ein Mond von der Planetenoberfläche aus größer erscheint als der

Erdmond von der Erde aus, muss gelten: 
$$\frac{\frac{\text{Durchmesser}_{\text{Mond}}}{2}}{\text{Halbachse}_{\text{Mond}} - \frac{\text{Durchmesser}_{\text{Planet}}}{2}} > 0,0046$$

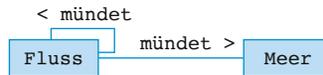
Den Wert 0,0046 erhält man, wenn die Daten der Erde und des Erdmondes in die Formel eingesetzt werden. Stelle fest, ob es Monde mit der beschriebenen Eigenschaft im Sonnensystem gibt.



38010-02  
benötigte Tabellen:  
Planet, Mond

6 In der Tabelle Stadt sind die Daten einiger Städte gespeichert, analoges gilt für die Tabellen Land, Fluss und Meer.

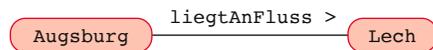
- a) Flüsse können in Meere münden, aber auch in andere Flüsse. Beispiel: Ein Fluss mündet in genau einen Fluss.



Der Fall, dass ein Fluss in einen abflusslosen See mündet, z. B. die Wolga ins Kaspische Meer, soll hier nicht berücksichtigt werden.

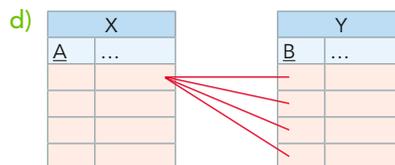
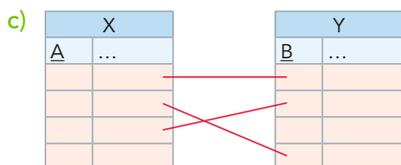
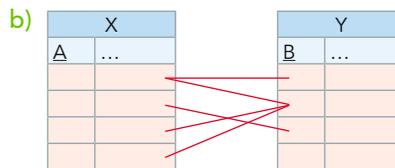
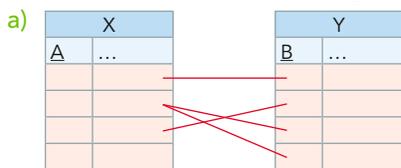
Übertrage das Klassendiagramm in dein Heft und gib alle Kardinalitäten an. Formuliere die Beziehungen (mit der Umkehrbeziehung hatAlsZufluss) jeweils als Satz, der mit „Ein Fluss ...“ bzw. „Ein Meer ...“ beginnt.

- b) Ergänze das Diagramm um die Klassen Stadt und Land sowie die Beziehungen: liegtIn (Stadt in Land); istHauptstadtVon (Stadt von Land); fließtDurch (Fluss durch Stadt); liegtAnMeer (Stadt an Meer).
- c) Gib für jede Beziehung drei Objektdiagramme an, z. B.:



- d) Ergänze im Diagramm die Kardinalitäten und gib in einer Liste alle 1:1-, 1:n- und n:m-Beziehungen an.
- e) Begründe, an welchen Stellen die Kardinalität 1 durch 0..1 ersetzt werden kann. Nenne jeweils ein Beispiel.

7 Gib an, ob in den Grafiken 1:1-, 1:n- oder n:m-Beziehungen dargestellt werden. Zeichne jeweils das Klassendiagramm mit den Kardinalitäten.



↪ 1.5

Alle Angaben sind eindeutig,  
das heißt z. B. bei a):  
Die Vornamen und Straßennamen  
sind Primärschlüssel.

- 8 Führe anhand der genannten Objektbeziehungen Folgendes durch:
- ▶ Zeichne die Objektkarten aller genannten Objekte. Stelle fest, welche Klassen benötigt werden, und ordne die Objekte den passenden Klassen zu.
  - ▶ Zeichne das Klassendiagramm.
  - ▶ Entwerfe jeweils die Tabellenschemata aller benötigten Tabellen, so dass die Objektbeziehungen umgesetzt werden können.
  - ▶ Gib die Tabellen mit so vielen Datensätzen an, dass die Objektbeziehungen erkennbar werden.
    - a) Anton wohnt in der Badestraße. Beate wohnt in der Hauptstraße. Carmen wohnt in der Badestraße. Daniel wohnt in der Veilchenstraße.
    - b) Erwin ist Mannschaftsführer vom FC Schienbein. Frieda ist keine Mannschaftsführerin. Gina ist Mannschaftsführerin der Meniskus Kickers. Hannes ist kein Mannschaftsführer.
    - c) Der Stern Sirius ist im Sternbild Großer Hund. Der Stern Rigel ist im Sternbild Orion. Der Stern Wega ist im Sternbild Leier. Der Stern Beteigeuze ist im Sternbild Orion.

↪ 1.5



38010-07  
benötigte Tabelle:  
Person



9 Eine Firma speichert Daten über ihre Angestellten in dieser Tabelle.

- a) Gib an, auf welches Attribut sich das Attribut `Vorgesetzter` bezieht.
- b) Zeichne das Klassendiagramm der Beziehung `istVorgesetzterVon`.
- c) Begründe, warum ein Attribut `Untergebener` statt `Vorgesetzter` als Fremdschlüssel ungünstig ist.
- d) Stelle mithilfe einer SQL-Abfrage die Achim Barone direkt unterstellten Mitarbeiter fest.
- e) Stelle die Hierarchie der Firma als Objektdiagramm (nur Nachnamen) dar. Gewinne die nötigen Informationen mithilfe passender SQL-Abfragen.
- f) Gib an, welche Struktur das Objektdiagramm aus e) hat. Beschreibe im Sachzusammenhang die Bedeutung des Fehlens eines Attributwerts bei `Vorgesetzter` im Datensatz von Achim Barone.

| Person |           |        |        |              |
|--------|-----------|--------|--------|--------------|
| Id     | Vorname   | Name   | Gehalt | Vorgesetzter |
| 1003   | James     | Khang  | 45000  | 1008         |
| 1006   | Christine | Herzog | 38000  | 1008         |
| 1007   | Erik      | Györfi | 36000  | 1008         |
| 1008   | Sos       | Hunor  | 60000  | 1009         |
| 1009   | Achim     | Barone | 150000 |              |
| 1010   | Jasmin    | Kaiser | 60000  | 1009         |
| ...    |           |        |        |              |

↪ 1.6



38010-01  
benötigte Tabellen:  
Lehrkraft, Schüler,  
unterrichtet



10 Untersuche die Tabellen `Schüler`, `Lehrkraft` und `unterrichtet`.

- a) Führe die SQL-Abfrage aus, beschreibe den Aufbau und die Funktionsweise.
 

```
SELECT Schüler.Name, Lehrkraft.Name, unterrichtet.Fach
FROM Schüler, Lehrkraft, unterrichtet
WHERE Schüler.SId = unterrichtet.SId
AND unterrichtet.Kürzel = Lehrkraft.Kürzel
AND unterrichtet.Fach = 'D'
```
- b) Zeichne zur SQL-Abfrage aus a) das Datenflussdiagramm und gib die funktionale Schreibweise an. Verwende jeweils die Funktionen Kreuzprodukt, Selektion und Projektion.
- c) Erstelle eine Abfrage, die folgender Funktion entspricht:
 

```
Projektion(Selektion(Kreuzprodukt(Schüler; unterrichtet);
Schüler.SId = unterrichtet.SId
AND Fach = 'Inf'));
Schüler.Name)
```

- 11 In der Datenbank über Deutschland gibt es auch die Tabelle `fließtDurchBL`, die eine Beziehung zwischen `FLuss` und `Bundesland` realisiert.
- Gib mindestens vier Objektbeziehungen an, die verdeutlichen, dass es sich um eine n:m-Beziehung handelt.
  - Konzipiere eine SQL-Abfrage, sodass in der Ergebnistabelle für jeden Fluss (alle Attribute) ersichtlich ist, durch welche Bundesländer (alle Attribute) er fließt.
  - Konzipiere eine SQL-Abfrage, die für jeden Fluss, an dem eine Millionenstadt liegt, alle Bundesländer, durch die er fließt, angegeben werden.

Fortsetzung von:  
 ↪ 1.6 Aufgabe 2



38010-05  
 benötigte Tabellen:  
 Fluss, Stadt, Bundesland,  
 fließtDurchBL

↪ 1.6

- 12 Begründe die folgenden Aussagen und gib jeweils ein Beispiel an.
- Die Anzahl der Klassenkarten muss nicht notwendigerweise mit der Anzahl der Tabellen übereinstimmen.
  - In einer Tabelle kann es mehr als einen Fremdschlüssel geben.
  - Ein Tabellenschema enthält mehr Informationen als eine Klassenkarte.
  - Es gibt auch 0..1:0..1-Beziehungen.
  - Schon zu Beginn der Modellierung muss berücksichtigt werden, welche SQL-Abfragen man durchführen möchte.

↪ 1.7

- 13 Familie Geiger ist musikalisch, fast alle Familienmitglieder spielen ein oder mehrere Instrumente. Im Haushalt gibt es vier Musikinstrumente: eine Flöte (25€), eine Geige (450€), ein Klavier (1200€) und eine Gitarre (380€). Tochter Eva (13 Jahre) spielt Flöte, Geige und Klavier, Sohn Max (16) Klavier und Gitarre, Mutter Erika (38) spielt Flöte, Geige und Gitarre, Vater Alois (41) Klavier und Geige. Tochter Claire ist erst sechs Jahre alt und spielt noch kein Instrument.
- Analysiere und strukturiere den Datenbestand.
  - Zeichne das Objektdiagramm für alle angegebenen Objekte und Objektbeziehungen. Gib auch an, wie lange jemand ein Instrument spielt und wie gut (in Schulnoten). Denke dir sinnvolle Attributwerte aus.
  - Entwickle das Datenmodell und stelle es grafisch dar.
  - Überführe das Datenmodell in ein relationales Modell.
  - Setze das relationale Modell in einem DBS um.
  - Führe jeweils eine SQL-Abfrage aus, um folgende Informationen zu erhalten. Kontrolliere die Ergebnisse jeweils anhand der Tabellen.
    - ▶ alle Personen (mit Alter), die Klavier spielen
    - ▶ alle Personen, die länger als fünf Jahre ein bestimmtes Instrument spielen
    - ▶ alle Personen (mit Alter), die ein mehr als 400 € teures Instrument spielen

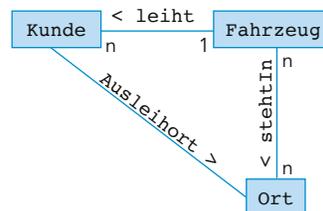


Eva spielt seit 5 Jahren sehr gut Klavier.



↪ 1.7

- 14 Die Autovermietung MietCar möchte ihre Fahrzeuge in einer Datenbank speichern. Folgende, teilweise unkorrekte Klassenstruktur wurde erstellt:

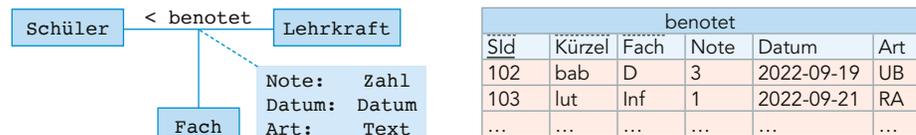


Beschränke dich jeweils auf die wichtigsten Attribute.

- Analysiere das Diagramm. Finde die Fehler und begründe die nötigen Änderungen.
- Entwickle ein korrektes Datenmodell und stelle es grafisch dar.
- Überführe das Datenmodell in ein relationales Modell.
- Als Service wird nun eingeführt, dass ein Auto nicht dort zurückgegeben werden muss, wo es übernommen wurde. Beschreibe nötige Änderungen.

- 15 Gegeben sind die Funktion  $\text{Join}(A; B)$  und die SQL-Abfrage  $\text{SELECT } * \text{ FROM } A, B \text{ WHERE } A.Pid = B.Fid$ .  $A$  und  $B$  sind jeweils Tabellennamen,  $Pid$  der Primärschlüssel von  $A$ ,  $Fid$  ein Fremdschlüssel in  $B$ .
- Begründe, unter welchen Voraussetzungen beide gleichwertig sind.
  - Zeichne für diesen Fall das gemeinsame Datenflussdiagramm.
  - Gib das Datenflussdiagramm und die funktionale Schreibweise an, wenn von beiden Tabellen nur das Attribut  $\text{Name}$  angezeigt werden soll.

- 16 Eine Beziehung kann auch zwischen mehr als zwei Tabellen bestehen, z. B.: In diesem Fall sind mehrere Fremdschlüssel in der Beziehungstabelle nötig.



UB: Unterrichtsbeitrag  
RA: Rechenschaftsablage

*Beachte:  
Bisher war Fach ein Attribut  
der Beziehungstabelle unter-  
richtet, jetzt ist es eine Klasse.*

- Gib die Primärschlüssel an, auf die sich die Fremdschlüssel in der Tabelle  $\text{benotet}$  beziehen.
- Begründe für jede Spalte der Tabelle  $\text{benotet}$ , ob sie in den Primärschlüssel mit aufgenommen werden muss. Beschreibe jeweils ein Beispiel, das verdeutlicht, warum die entsprechende Spalte nicht fehlen darf.
- Begründe, warum die Attribute  $\text{Note}$ ,  $\text{Datum}$  und  $\text{Art}$  nicht in den Klassen  $\text{Schüler}$ ,  $\text{Lehrkraft}$  oder  $\text{Fach}$  gespeichert werden können

- 17 Eine Online-Plattform speichert Daten in diesen Tabellen.

| Kunde |         |          |       | Abo |         |
|-------|---------|----------|-------|-----|---------|
| KId   | Vorname | Name     | AboNr | AId | Typ     |
| 46234 | Heinz   | Senfer   | 1     | 1   | Free    |
| 46235 | Miriam  | Missgard | 3     | 2   | Premium |
| ...   |         |          |       | ... |         |

| hatGekauft |        | Produkt |                     |       |
|------------|--------|---------|---------------------|-------|
| KId        | PId    | PId     | Name                | Typ   |
| 46232      | F60343 | F56563  | Tages des Lichts    | Film  |
| 34343      | S35030 | S65725  | Small Smash Theorem | Serie |
| ...        |        | ...     |                     |       |

| Genre |         | favorisiert |     | gehörtZu |     |
|-------|---------|-------------|-----|----------|-----|
| GId   | Name    | KId         | GId | PId      | GId |
| 1     | Komödie | 46200       | 2   | F56563   | 2   |
| 2     | Action  | 46200       | 4   | S65725   | 1   |
| ...   |         | ...         |     | ...      |     |

- Erkläre, welche Tabellen Klassen beschreiben und welche Beziehungen. Gib an, zwischen welchen Klassen sie eine Beziehung herstellen.
- Stelle das zugehörige objektorientierte Datenmodell als Klassendiagramm dar.
- Gib das zugehörige objektorientierte relationale Modell mithilfe der Tabellenschemata aller Tabellen an. Kennzeichne Primär- und Fremdschlüssel.
- Konzipiere zu jeder  $n:m$ -Beziehung eine geeignete SQL-Abfrage, sodass alle verknüpften Datensätze aufgelistet werden.
- Bei den Tabellen wurde ein künstlicher Primärschlüssel eingefügt. Nimm nun an, in  $\text{Kunde}$  wäre  $\text{Vorname}$  und  $\text{Name}$  der Primärschlüssel und in  $\text{Produkt}$ ,  $\text{Name}$  und  $\text{Typ}$ . Gib an, wie dann die Beziehungstabelle zwischen beiden aufgebaut sein müsste und formuliere eine Abfrage, die alle verknüpften Datensätze auflistet. Beschreibe die Nachteile dieser Modellierung.

18 Zwischen zwei Klassen kann auch mehr als eine Beziehung bestehen: z. B.:

- ▶ Ein Schüler besucht genau eine Schulklasse.
- ▶ Ein Schüler ist (erster) Klassensprecher von genau einer Schulklasse.

Weiter gilt: Eine Schulklasse ist in genau einer Schule.

- a) Entwirf das objektorientierte Datenmodell und stelle es grafisch dar. In allen Klassen genügt eine `Id` als Primärschlüssel und das Attribut `Name`.
- b) Beschreibe die Umsetzung der Beziehungen im relationalen Modell und überführe das objektorientierte Datenmodell in ein relationales Modell.
- c) Das relationale Modell wurde in einem DBS umgesetzt. Konzipiere geeignete SQL-Abfragen, um folgende Informationen gewinnen zu können (es sollen jeweils die Namen angezeigt werden):
  - ▶ Alle Schüler, die Klassensprecher sind.
  - ▶ Alle Schüler, welche die Klasse 10c in der Schule mit der Schulnummer 0153 besuchen.
  - ▶ Alle Klassensprecher und die Schule, die sie besuchen.

In diesem Fall spricht man auch von unterschiedlichen Rollen, die eine Schülerin oder ein Schüler in einer Schulklasse spielt.

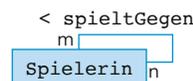
19 In Lenas Tischtennisverein sind die sechs Jugendspielerinnen nach Spielstärke in einer Rangliste geordnet. Um die Jugendmeisterin zu ermitteln, spielt jedes Mädchen gegen jede andere, für jeden Sieg (Unentschieden gibt es nicht) erhält die Siegerin einen Punkt.

Die sechs Spielerinnen sind in dieser Tabelle gespeichert (R steht für den Ranglistenplatz).

| Spielerin |          |         |       |   |
|-----------|----------|---------|-------|---|
| Nr        | Nachname | Vorname | Alter | R |

- a) Begründe, warum sich der Ranglistenplatz nicht als Primärschlüssel eignet, obwohl er eindeutig ist.
- b) Konzipiere eine SQL-Abfrage, die alle Paarungen (nur die Vornamen) auflistet. Beachte das Datenflussdiagramm in 1.2 und, dass der Tabelle `Spielerin` nach `FROM` mithilfe von `AS` zwei neue Namen gegeben werden müssen.
- c) In der Ergebnistabelle sind auch Paarungen einer Spielerin gegen sich selbst enthalten. Ergänze die SQL-Abfrage aus b) durch Hinzufügen einer Selektion so, dass dies verhindert wird. Schreibe die Abfrage auch funktional.
- d) In der Ergebnistabelle ist jede Paarung doppelt vorhanden (es gibt aber keine Hin- und Rückspiele). Ändere die SQL-Abfrage aus c) so ab, dass jede Paarung nur einmal erscheint. Tipp: `Nr` oder `R` in der Selektion verwenden.

e) Ein Spiel ist eine Beziehung dieser Tabelle mit sich selbst. Begründe, dass es sich um eine n:m-Beziehung handelt und zeichne das Klassendiagramm rechts in der Form, dass die n:m-Beziehung in zwei 1:n-Beziehungen aufgelöst wird.



- f) In der Beziehungstabelle können nun auch die Ergebnisse gespeichert werden. Da das Spiel nach drei gewonnenen Sätzen abgebrochen wird, sind die Ergebnisse 3:0, 3:1 und 3:2 möglich. Gib das Tabellenschema der Tabelle `spieltGegen` an.
- g) Setze das relationale Modell in einem DBS um. Gib sinnvolle Werte in die Tabelle `spieltGegen` ein. Solltest du bisher mit der Online-Version gearbeitet haben, übertrage die Daten der Tabelle `Spielerin`.
- h) Konzipiere SQL-Abfragen zur Gewinnung folgender Informationen:
  - ▶ Ergebnisse aller Spiele (mit Vor- und Nachnamen)
  - ▶ Ergebnisse aller Spiele Lenas (mit Vornamen)
  - ▶ Spiele, die "zu null" ausgegangen sind (mit Ranglistenplatz)

Informiere dich, wann ein Satz im Tischtennis als gewonnen gewertet wird.



38010-08  
benötigte Tabelle:  
Spielerin

## DAS GROSSE INFO-QUIZ!

Hier sprechen Begriffe, die du in diesem Kapitel gelernt und wiederholt hast, über sich. Die Zahlen in Klammern geben den Buchstaben im erratenen Begriff an, den du für das Lösungswort brauchst.

- A** „Meine Attributwerte sind welche aus C.“ .....(6)  
**B** „Wir geben an, welche Art von Informationen in den Attributen gespeichert werden.“ .....(4)  
**C** „Mich brauchst du, damit jeder Datensatz eindeutig festgelegt ist.“ ..... (10)  
**D** „Ich entstehe, wenn hinter FROM mehr als eine Tabelle angegeben wird.“ .....(3)  
**E** „Der Begriff aus C wird so im Tabellenschema markiert.“ ..... (10)  
**F** „So viele Objekte stehen mit mir in Beziehung.“ ..... (12)



38010-09

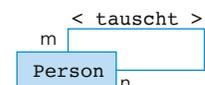
Hast du das Ziel des Kapitels erreicht? Schätze dich zunächst mithilfe des hinterlegten Bogens selbst ein. Bearbeite dann die Aufgaben und vergleiche mit deiner Selbsteinschätzung.

- 1** Nimm Stellung zu dieser Behauptung: „Bei einer 1:n-Beziehung, wie z. B. zwischen **Schüler** und deren **Schulklasse**, ist es unerheblich, in welcher der beiden Tabellen der Fremdschlüssel eingefügt wird.“
- 2** Die Tischtennisgruppe der Schule führt ein Turnier für gemischte Doppel (Fachbezeichnung: Mixed) durch. Die Schülerinnen und Schüler sind in jeweils einer Tabelle pro Geschlecht gespeichert.
  - a) Erkläre, wie man alle möglichen Paare aus den beiden Tabellen erhält, mit einem Begriff, den du in diesem Kapitel gelernt hast.
  - b) Begründe, warum man in einer Datenbank keine zwei „Listen“ speichern würde.
- 3** In einer Socialmedia Plattform sind die Nutzer und ihre Hobbies gespeichert.

| Nutzer |          |         |          | Hobby |      |
|--------|----------|---------|----------|-------|------|
| ID     | Nachname | Vorname | GebDatum | HID   | Name |

- a) Benenne die Beziehung zwischen den beiden Klassen.
- b) Begründe, warum diese zwei Tabellen nicht genügen, die Hobbies den Nutzern zuzuordnen.
- c) Gib das relationale Modell mithilfe der Tabellenschemata an.
- d) Konzipiere eine passende SQL-Abfrage, um folgende Information zu gewinnen: alle Nutzer mit dem Hobby Handball
- e) Erkläre an diesem Beispiel die Begriffe Kreuzprodukt, Fremd- und Primärschlüssel.

- 4** Alle Tauschvorgänge einer Online-Tauschbörse für gebrauchte Gegenstände sollen in einer Datenbank erfasst werden. Zunächst zeichnet Khuyen dieses Klassendiagramm.



- a) Gib an, welche Angabe in diesem Datenmodell noch fehlt.
- b) Überführe das Datenmodell in ein relationales Modell. Dabei soll die fehlende Angabe aus Teilaufgabe a) berücksichtigt werden. Zusätzlich können beide Tauschpartner den Tausch jeweils mit einer Schulnote bewerten, diese sollen auch erfasst werden.
- c) Jede teilnehmende Person soll auf Wunsch eine Liste erhalten können, aus der die Anzahl der Tauschvorgänge mit den Bewertungen 1; ...; 6 ersichtlich ist. Beschreibe, wie Khuyen dies am besten durchführt.
- d) Das Datum, an dem die Tauschaktion erledigt wurde, will Khuyen in der Tabelle **Person** speichern. Erkläre, warum du Khuyen davon abraten würdest.

# Modellieren von Datenbanken

# 2

## Einstieg

Am Dijkstra-Gymnasium wurde der Pausenverkauf digitalisiert. In der Tabelle sind die rot markierten Änderungen nachträglich durchgeführt worden.

- ▶ Erkläre, warum diese Änderungen nötig gewesen sein könnten.
- ▶ Beschreibe, warum die Tabelle unnötige und widersprüchliche Informationen enthält.

| Pausenverkauf |          |           |                  |               |               |                      |
|---------------|----------|-----------|------------------|---------------|---------------|----------------------|
| Datum         | Name     | Art       | ArtNr            | StückVerkauft | Verkaufspreis | Kaufpreis            |
| 2022-05-09    | Wasser   | Getränk   | 1                | 23            | 2,00          | 1,50                 |
| 2022-05-09    | Brötchen | Essen     | 11               | 70            | 1,00          | 0,60                 |
| 2022-05-09    | Sticker  | Sonstiges | 31               | 5             | 1,00          | 0,50                 |
| 2022-05-09    | O-Saft   | Getränk   | 4                | 8             | 3,50          | 2,80                 |
| 2022-05-09    | Brezeln  | Essen     | 10               | 80            | 0,70          | 0,40                 |
| 2022-05-10    | Wasser   | Getränk   | 1                | 16            | 2,00          | 1,50                 |
| 2022-05-10    | Brötchen | Essen     | <del>11</del> 12 | 74            | 1,00          | 0,60                 |
| 2022-05-10    | Breze    | Essen     | 10               | 67            | 0,70          | <del>0,30</del> 0,40 |
|               | Waffleis | Essen     | 13               |               |               | 0,40                 |

## Am Ende dieses Kapitels hast du gelernt, ...

- ▶ Redundanzen und Anomalien zu erkennen.
- ▶ die Konsistenz von Datenbeständen zu beurteilen.
- ▶ die automatische Analyse verknüpfter Datenbestände zu bewerten.

## EINSTIEG

Der Hausmeister hat die Bestellungen für den Pausenverkauf, die bei verschiedenen Läden gekauft und abgeholt werden, bisher in eine Tabelle eines Tabellenkalkulationsprogramms eingetragen. Nils von der Software AG soll diese in ein DBS übertragen, um die mehrfache Eingabe von Daten zu vermeiden.

| Bestellung |            |                              |        |        |             |           |
|------------|------------|------------------------------|--------|--------|-------------|-----------|
| Händler    | Datum      | Adresse                      | TelNr  | Anzahl | Artikel     | Kaufpreis |
| Brehm      | 2022-09-19 | 98000 Neustadt, Hafenstr. 14 | 58812  | 80     | Wasser      | 1,50      |
| Popp       | 2022-09-19 | 98000 Neustadt, Baumweg 5    | 11126  | 50     | Sticker     | 0,50      |
| Kolb       | 2022-09-20 | 98001 Altdorf, Am Berg 7     | 22325  | 100    | Brezeln     | 0,40      |
| Brehm      | 2022-09-20 | 98000 Neustadt, Hafenstr. 1  | 58812  | 30     | O-Saft      | 2,80      |
| Kolb       | 2022-09-20 | 98001 Altdorf Am Berg 7      | 223250 | 100    | Brezel      | 0,40      |
| Ruben      | 2022-09-22 | 98002 Erweiler, Hügelpfad 10 | 44455  | 50     | Orangensaft | 2,80      |
| Haller     | 2022-09-23 | 98000 Neustadt, Hafenstr. 1  | 12789  | 100    | Wasser      | 1,50      |
| ...        |            |                              |        |        |             |           |

- ▶ Gib einige Informationen an, die mehrfach gespeichert wurden.
- ▶ Finde einige Fehler, die sich dadurch eingeschlichen haben.
- ▶ Nenne Voraussetzungen für einen Primärschlüssel aus **Händler** und **Datum**.

## ERARBEITUNG

Hinweis:

In Sicherheitsfragen sind redundante Systeme nützlich.  
Beispiele:  
zweites Bremssystem im Auto,  
mehrere Datenbackups zum  
Schutz vor Datenverlust, usw.

## Redundanz und Inkonsistenz

Werden dieselben Daten in einem Datenbestand mehrfach gespeichert, so spricht man von **Redundanz**.

## A1 Redundanzen beurteilen

- a) Beschreibe einen praktischen Nachteil beim Ausfüllen einer Tabelle mit redundanten Daten.
- a) Begründe, warum es ungünstig wäre, z. B. die Adresse nur bei der ersten Bestellung anzugeben.

Die mehrfache Eingabe gleicher Informationen ist nicht nur zeitraubend, es besteht auch die Gefahr, dass die Daten widersprüchlich sind. Beim Händler Kolb sind zum Beispiel unterschiedliche Telefonnummern eingetragen. Diese Widersprüche im Datenbestand werden als **Inkonsistenzen** bezeichnet.

## Anomalien

Enthält ein Datenbestand widersprüchliche Informationen, so liefern Abfragen falsche oder unvollständige Ergebnisse. Dies bezeichnet man als **Anomalie**. Anomalien können durch eine sinnvolle Modellierung (↔ 2.2) im Vorfeld verhindert werden.

## A2 Anomalien finden

Beschreibe Probleme, die bei Abfragen in der Tabelle **Bestellung** (Primärschlüssel: **Händler**, **Datum**), aufgrund der genannten Änderungen auftreten können.

- a) Bei Händler Brehm wird viel bestellt. Damit gleich der richtige Sachbearbeiter am Telefon ist, soll statt der Basisrufnummer der Firma die Durchwahl zum Sachbearbeiter eingetragen werden.
- b) Bei Bäcker Humboldt soll nach den nächsten Ferien erstmals bestellt werden. Der Hausmeister möchte seine Kontaktdaten aber jetzt schon erfassen.
- c) Bei Händler Müller wurde bisher erst einmal bestellt. Da er nicht liefern konnte, wurde der entsprechende Datensatz wieder gelöscht.

Arbeitet nur ein Benutzer mit einer Datenbank, sind folgende Anomalien möglich:

| Anomalie                                                                                                                                                                | Beispiel                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Änderungs-Anomalie (Update-Anomalie):</b><br>Bei redundanten Daten werden nicht alle Attributwerte gleichzeitig geändert.                                            | Die bisher eingetragene Artikelbezeichnung Breze soll in Brezel geändert werden, dies wurde aber bei einigen Bestellungen vergessen. Dadurch ist der Name nicht mehr eindeutig und bei entsprechenden Abfragen werden nicht alle Datensätze gefunden.                                             |
| <b>Lösch-Anomalie (Delete-Anomalie):</b><br>Es werden mehr Informationen gelöscht als beabsichtigt.                                                                     | Aus Datenschutzgründen werden alle Bestellungen gelöscht, die länger als ein halbes Jahr zurückliegen. Wurde in diesem Zeitraum bei einem Händler keine Bestellung aufgegeben, so verschwinden auch seine Kontaktdaten. Wird eine Liste der Telefonnummern erstellt, taucht diese nicht mehr auf. |
| <b>Einfüge-Anomalie (Insertion-Anomalie):</b><br>Ein neuer Datensatz kann nur schwer oder gar nicht eingetragen werden, wenn Werte der Primärschlüsselattribute fehlen. | Solange keine Bestellung vorliegt, kann das neue Produkt Waffeleis nicht eingetragen werden, da die Primärschlüsselattribute Händler und Datum nicht bekannt sind.                                                                                                                                |

Unter **Redundanz** versteht man die mehrfache, unnötige Speicherung von gleichen Informationen. **Inkonsistent** ist ein Datenbestand, der widersprüchliche Informationen enthält. Daraus können bei Abfragen falsche oder unvollständige Ergebnisse resultieren, was als **Anomalie** bezeichnet wird. Bei Einbenutzerdatenbanken unterscheidet man **Änderungs-**, **Lösch-** und **Einfüge-**Anomalie.

MERKE

1 Der Handwerkerservice Moll speichert alle Aufträge in dieser Tabelle:

| Auftrag |              |            |           |            |                           |
|---------|--------------|------------|-----------|------------|---------------------------|
| Nr      | Arbeiter     | Beruf      | Arbeiten  | Datum      | Kunde                     |
| 1       | Hubert Hobel | Schreiner  | Einbau    | 2022-06-01 | Martin Glotz, Am Pfahl 13 |
| 2       | Ibrahim Chem | Schlosser  | Einbau    | 2022-06-01 | Olivia Nimm, Kohlenweg 2  |
| 3       | Hubert Hobel | Schreiner  | Reparatur | 2022-06-08 | Martin Glotz, Am Pfahl 13 |
| 4       | Ibrahim Chem | Schlossern | Reparatur | 2022-06-08 | Chan Jan, Marktplatz 5    |
| 5       | Ibrahim Chem | Schlosser  | Kontrolle | 2022-06-08 | Olivia Nimm, Kohlenweg 2  |

AUFGABEN



- a) Nenne die in der Tabelle redundant gespeicherten Daten.
- b) Führe zwei Änderungen an der Tabelle durch und gib je eine Abfrage an, sodass je eine Änderungs- und eine Lösch-Anomalie auftritt.

2 Eine Kundenberatung hat die kontaktierten Personen in dieser Tabelle gespeichert. Es wurden vier Datensätze nacheinander geändert (farbige Markierungen).

| Kontaktprotokoll |                   |           |            |                  |
|------------------|-------------------|-----------|------------|------------------|
| Datum            | Name              | TelNr     | GebDatum   | KontaktArt       |
| 2022-05-05       | Giom Larke        | +4999001  | 2002-10-03 | Beratung Service |
| 2022-05-05       | Paula Kalliomäki  | +4988008  | 1996-11-30 | Service          |
| 2022-05-06       | Giom Larke Möbius | +49177222 | 2002-10-03 | Verkauf          |
|                  | Katrin Leng       | +49111111 |            |                  |

- a) Beschreibe je einen möglichen Grund für diese Änderungen.
- b) Begründe, warum eine Einfüge-Anomalie nicht erst bei Abfragen auftritt. Gib zwei Abfragen an, die jeweils eine der beiden anderen Anomalien bewirken.

RÜCKBLICK

Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ redundante Informationen zu erkennen.
- ▶ wie Anomalien in Datenbanken entstehen können.

## EINSTIEG

Analysiere die Tabelle `Bestellung` aus ↪ 2.1.

| Bestellung |            |                              |       |        |         |           |
|------------|------------|------------------------------|-------|--------|---------|-----------|
| Händler    | Datum      | Adresse                      | TelNr | Anzahl | Artikel | Kaufpreis |
| Brehm      | 2022-09-19 | 98000 Neustadt, Hafenstr. 14 | 58812 | 80     | Wasser  | 1,50      |

Erkläre jeweils:

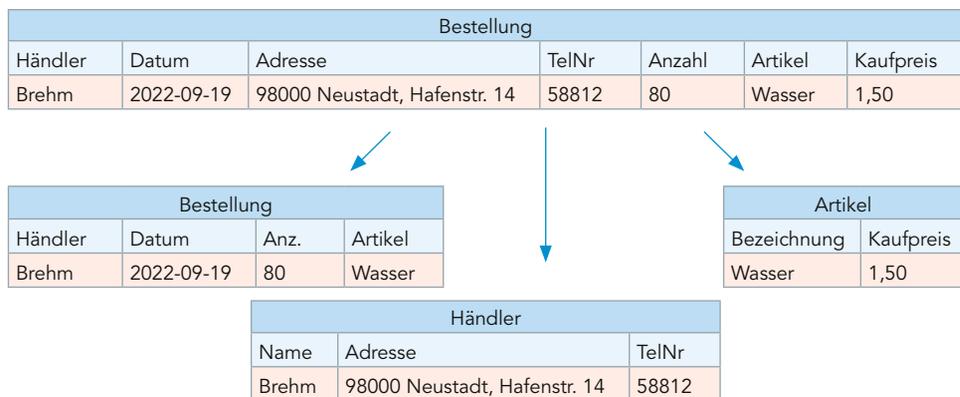
- ▶ Welche Spalten enthalten Informationen über den Händler?
- ▶ Welche Spalten enthalten Informationen über den bestellten Artikel?
- ▶ Welche Spalten enthalten Informationen über die Bestellung?

## ERARBEITUNG

### Aufteilung von Informationen in mehrere Tabellen

Die Tabelle `Bestellung` enthält Informationen, die inhaltlich nichts miteinander zu tun haben. In einer Klasse – und damit in einer Tabelle – sollten jedoch nur inhaltlich zusammenhängende (kohärente) Informationen gespeichert werden. Nils von der Software AG macht daher folgenden Vorschlag:

Die Informationen in der Tabelle werden aufgeteilt:



#### A1 Modellierung analysieren

- a) Diskutiert die Vorteile dieser Aufteilung, besonders im Hinblick auf Redundanz und Konsistenz.
- b) Begründet, dass damit alle drei Anomalien (↪ 2.1) vermieden werden können.
- c) Beschreibt Nachteile, die durch die Verteilung der Informationen auf mehrere Tabellen entstehen.
- d) Stellt das objektorientierte Datenmodell (↪ 1.7) grafisch dar.

Durch diese Aufteilung in mehrere Tabellen wird erreicht, dass in einer Tabelle nur inhaltlich zusammenhängende Informationen gespeichert sind. Redundanzen können damit weitgehend vermieden werden, z. B. muss die Adresse eines Händlers nicht länger mehrfach eingegeben werden. Somit ist auch die Eingabe inkonsistenter Daten (z. B. unterschiedliche Telefonnummern beim gleichen Händler) nicht mehr möglich.

#### Merksatz: Speichere jedes Datum nur einmal!

Sachlich falsche Einträge kann es allerdings weiterhin geben. Sie lassen sich nicht automatisch verhindern, sondern müssen durch andere Methoden erkannt werden, z. B. durch mehrfaches Kontrollieren der eingegebenen Daten.

Beispiel:

Die erste Bestellung bei Händler Brehm war eigentlich O-Saft, in der Tabelle wurde jedoch irrtümlich Wasser eingetragen.

### Aufteilung von Informationen auf mehrere Spalten

Zur Speicherung der Adressen macht die Software AG diese Vorschläge:

| Version 1                    |  | Version 2 |          |           |      |
|------------------------------|--|-----------|----------|-----------|------|
| Adresse                      |  | Adresse   |          |           |      |
| Anschrift                    |  | PLZ       | Ort      | Straße    | HsNr |
| 98000 Neustadt, Hafenstr. 14 |  | 98000     | Neustadt | Hafenstr. | 14   |

| Version 3 |          | Adresse |           |      |
|-----------|----------|---------|-----------|------|
| Ort       |          | PLZ     | Straße    | HsNr |
| ID        | Name     |         |           |      |
| 98000     | Neustadt | 98000   | Hafenstr. | 14   |

#### A2 Modellierung analysieren

- Diskutiert die Vor- und Nachteile der drei Versionen.
- Gebt mögliche Primärschlüssel für alle Tabellen an.
- Begründet, mit welcher Version sich Abfragen wie z. B. „Welche Händler sind in der Hafenstr.“ besonders einfach durchführen lassen.

Enthält eine Spalte zu viele Informationen, werden Abfragen schwierig (z. B. „Welche Händler sind in Altdorf?“). Sind solche Abfragen geplant, sollte es eine Spalte mit nur dieser Information geben. Lässt sich die Information in einer Spalte nicht weiter zerlegen, nennt man sie atomar (siehe [↪ 2.5 Normalformen](#)).

Neustadt ist z. B. atomar, Hafenstr. 14 besteht jedoch aus den Bestandteilen Straße und Hausnummer.

Redundanzen und somit auch ein inkonsistenter Datenbestand können durch die Aufspaltung in mehrere, inhaltlich zusammenhängende Tabellen vermieden werden. Damit lassen sich auch die drei bekannten Anomalien vermeiden. Die Informationen in einer Spalte sollten atomar (nicht weiter zerlegbar) sein.

**MERKE**

- Sportverein SV Vitalis hat für seine Mitglieder und die ausgeübten Sportarten diese Tabelle angelegt:

| Mitglied    |                |            |                    |
|-------------|----------------|------------|--------------------|
| Sportart    | Name           | Ort        | Wohnort            |
| Turnen, Mo  | Luca Schießler | Sporthalle | Ballheim           |
| Fußball, Mi | Harry Ecker    | Arena      | Schnellingen       |
| Fußball, Mi | Felix Läufer   | Arena      | Lauf a. d. Pegnitz |
| Fußball, Mi | Luca Schießler | Arena      | Ballheim           |
| ...         |                |            |                    |

Außerdem kann man noch am Donnerstag Tennis auf dem Tennisplatz spielen und am Dienstag Gymnastik in der Sporthalle betreiben.

- Beschreibe alle Fehler in der Tabelle `Mitglied`, die in diesem Kapitel angesprochen wurden.
- Gib eine bessere Aufteilung an, die diese Fehler vermeidet.
- Beschreibe drei Beispiele, die in der angegebenen Tabelle je eine der in [↪ 2.1](#) angesprochenen Anomalien verursachen, und begründe, warum diese in der neuen Version nicht auftreten.

**AUFGABEN**

#### Am Ende dieses Unterkapitels hast du gelernt, ...

- bei der Modellierung von Datenbanken Redundanzen und Anomalien zu vermeiden.

**RÜCKBLICK**

## EINSTIEG

Die Tabelle *Bestellung* des Hausmeisters aus ↪ 2.1 wurde in diese drei Tabellen aufgeteilt:

| Bestellung |            |      |         | Artikel     |           |
|------------|------------|------|---------|-------------|-----------|
| Händler    | Datum      | Anz. | Artikel | Bezeichnung | Kaufpreis |
| Brehm      | 2022-05-09 | 615  | Wasser  | Wasser      | 1,50      |

| Händler |                             |       |
|---------|-----------------------------|-------|
| Name    | Adresse                     | TelNr |
| Brehm   | 98000 Neustadt Hafenstr. 14 | 58812 |

Beschreibe jeweils:

- ▶ Welche Verbesserungen sollten noch durchgeführt werden?
- ▶ Welche Schritte sind nötig, um daraus ein relationales Modell zu erhalten?

## ERARBEITUNG

### Zusätzliche Anforderungen an eine gute Datenmodellierung

Zusätzlich zu den in ↪ 1.7 dargestellten Überlegungen ist zu beachten:

- ▶ Der Datenbestand sollte immer konsistent sein, um Anomalien zu verhindern.
- ▶ Schon im Datenmodell sind geplante SQL-Abfragen und absehbare Änderungen zu berücksichtigen.

### Sicherung der Konsistenz

#### A1 Modellierung analysieren: Inkonsistenzen vermeiden

Beurteile folgende Vorschläge unter Beachtung des bisherigen Modells.

- a) Jede Bestellung soll auch den Bestellwert enthalten.
- b) Jede Bestellung soll auch den Ansprechpartner des Händlers enthalten.
- c) Der Hausmeister erwägt, bei einem neuen Händler zu bestellen.

Um den Datenbestand konsistent zu halten, ist zu beachten:

- ▶ Ein Fremdschlüssel (↪ 1.3) darf nur Werte enthalten, die im entsprechenden Primärschlüssel vorhanden sind. Dies kann von vielen DBS automatisch geprüft werden, die solche fehlerhafte Einträge dann nicht zulassen.
- ▶ Daten, die übereinstimmen müssen, dürfen immer nur einmal gespeichert werden. Dies wird im Beispiel durch die in ↪ 2.2 beschriebene Aufteilung in drei Tabellen erreicht.
- ▶ Es dürfen keine Daten gespeichert werden, die aus anderen berechenbar sind.  
*Beispiel:* Wird bei einem Quadrat Seitenlänge *und* der Flächeninhalt gespeichert, so ist nach einer Änderung der Seitenlänge der Flächeninhalt falsch. Dies kann vermieden werden, indem nur die Seitenlänge gespeichert wird, der Flächeninhalt wird nur bei Bedarf berechnet und ist dann immer richtig.

### Abfragen berücksichtigen

#### A2 Modellierung analysieren: Abfragen berücksichtigen

Gib die nötigen Änderungen am obigen Modell an, die sich aus folgenden Anforderungen des Hausmeisters ergeben.

- a) Er benötigt eine Liste der Händler nach Ort und Straße geordnet.
- b) Am Ultimo benötigt er eine Liste aller Bestellungen dieses Monats.

Bei der Analyse und Strukturierung der Daten ist zu beachten, welche Abfragen geplant sind. Entsprechende Informationen sollten atomar (↪ 2.2) gespeichert sein.

siehe ↪ 1.1:  
Tabellen erweitern mit  
Operatoren (fakultativer Inhalt)

di ultimo (ital): am letzten Tag  
Im Bankwesen wird mit Ultimo  
der letzte Bankarbeitstag des  
Monats bezeichnet.

## Änderungen berücksichtigen

### A3 Modellierung analysieren: Änderungen berücksichtigen

Gib die nötigen Anpassungen an, die sich aus diesen Änderungen ergeben.

- Der Händler Brehm bietet jetzt Wasser sowohl in großen als auch in kleinen Flaschen an.
- Der Händler Ruben erhöht den Preis für den O-Saft um 5 Cent.

Absehbare Änderungen des Datenbestands sollten keine Änderungen am Modell nötig machen.

## Überführung in ein relationales Modell

Durch die Aufteilung in mehrere Tabellen sind die Objektbeziehungen verloren gegangen (z.B.: Wo muss angerufen werden, um Wasser zu bestellen?). Sie müssen durch Joins (→ 1.3) wiederhergestellt werden. Dabei ist es hilfreich, wenn die Primär- und Fremdschlüssel möglichst einfach sind (nur ein Attribut umfassen).

### A4 Modellierung analysieren: Primär- und Fremdschlüssel festlegen

- Begründe, unter welchen Voraussetzungen bei den Tabellen `Artikel` und `Händler` jeweils ein vorhandenes Attribut als Primärschlüssel geeignet ist.
- Begründe, warum das bei der Tabelle `Bestellung` nicht möglich ist und finde eine geeignete Lösung.

Die Datenmodellierung sollte so erfolgen, dass

- ▶ der Datenbestand immer konsistent bleibt.
- ▶ berücksichtigt wird, welche SQL-Abfragen nötig sein werden.
- ▶ absehbare Änderungen keine Änderung am objektorientierten Datenmodell und/oder relationalen Modell nach sich ziehen.

MERKE

- Gib unter Berücksichtigung der obigen Punkte für die Bestellliste an:
  - das objektorientierte Datenmodell (Klassenkarten und Klassendiagramm)
  - ein dazu passendes relationales Modell

AUFGABEN



- Analysiert in der Gruppe folgende Situation und entwickelt unter Berücksichtigung der in → 1.7 und in diesem Kapitel angesprochenen Themen schrittweise ein relationales Modell.  
Sabrina und ihre Freundinnen sind begeisterte Reiterinnen; Sabrina und Mandy kümmern sich auch im Reitverein um die Tiere.
  - ▶ Die Mädchen und die Pferde werden in je einer Tabelle gespeichert.
  - ▶ Je ein Mädchen reitet zu einem bestimmten Zeitpunkt ein Pferd, Sabrina oder Mandy versorgen zu einem bestimmten Zeitpunkt alle Pferde.
  - ▶ Neue Mädchen kommen dazu oder springen ab, auch die Pferde wechseln.



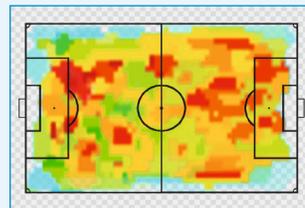
RÜCKBLICK

## Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ bei der Datenmodellierung Redundanzen und Anomalien zu vermeiden.
- ▶ bei der Datenmodellierung geplante Abfragen und absehbare Änderungen zu berücksichtigen.

## EINSTIEG

Bei Spielen der Fußball-Bundesliga wird die Position der Spieler erfasst und daraus z. B. eine sogenannte Heatmap erstellt. Sie zeigt, mit welcher Häufigkeit sich die Spieler wo aufgehalten haben.



- ▶ Begründe, warum die Erfassung automatisch erfolgen muss.
- ▶ Begründe, warum die Auswertung automatisch erfolgen muss.
- ▶ Begründe, warum die Daten für die Trainer schon während des Spiels interessant sein können.

## ERARBEITUNG

## Datenanalysen

Mit **Datenanalyse** bezeichnet man den Vorgang, aus meist großen und oft unstrukturierten Datenbeständen neue Erkenntnisse zu gewinnen.

Dabei interessieren z. B. diese Fragen: Was ist bisher passiert? Gibt es dafür Gründe? Wie wird es weitergehen? Welche Entscheidungen muss ich treffen, um den weiteren Verlauf zu meinen Gunsten entscheiden zu können?



## A1 Datenanalyse beschreiben und bewerten

Bearbeitet in einer Gruppe jeweils eine der untenstehenden Teilaufgaben. Eine Person trägt die Ergebnisse dann der Klasse vor. Beachtet folgende Punkte:

Wie wird die Datenanalyse durchgeführt (automatisch/per Hand, permanent/episodisch, ...)? Welche Datenbestände werden genutzt? Welche Vorteile verspricht man sich davon? Bewertet die Vor- und Nachteile für die Kunden (bzw. Bürger).

- a) Datenanalyse durch Online-Shops
- b) Datenanalyse durch Regierungen
- c) Datenanalyse durch die Betreiber sozialer Netzwerke
- d) Datenanalyse der Bilder von Kameras an öffentlichen Plätzen

## Verknüpfte Datenbestände

## A2 Datenbestände verknüpfen

Sarah nimmt an einem Feriencamp teil. Die Teilnehmer mussten sich mit ihren Kontaktdaten anmelden und werden mit dem Bus täglich abgeholt und heimgefahren. Da es nach einigen Tagen zu Beschwerden wegen des Transports kommt, wird eine anonyme Umfrage durchgeführt.

## Fragebogen

Geschlecht: \_\_\_\_\_

Entfernung zum Wohnort: \_\_\_\_\_

0–5 km  5–10 km  mehr als 5 km

Bewerte jeweils (1 = sehr gut; ...; 6 = sehr schlecht)

allgemeine Sicherheit: \_\_\_\_\_

Verhalten des Fahrers: \_\_\_\_\_

- a) Begründe, warum im Sachzusammenhang das Geschlecht wichtig sein kann.
- b) Sarah hat eine besonders schlechte Bewertung abgegeben. Beschreibe, unter welchen Bedingungen es unter Zuhilfenahme der Kontaktdaten möglich sein kann, ihr den Fragebogen zuzuordnen.

Werden mehrere Datenbestände miteinander **verknüpft**, können evtl. Zusammenhänge erkannt werden, die aus nur einem Datenbestand nicht herauslesbar wären.

## Automatisierte Analyse

Die automatische Speicherung von Daten führt zu großen Datenmengen (Big Data). Für herkömmliche Analyse-Methoden sind sie oft zu komplex, zu schnelllebig oder zu schwach strukturiert. Besonders wenn ständig neue Daten angeliefert werden, verwendet man häufig **automatisierte**, vorher festgelegte Verfahren (Data Mining, teilweise mithilfe von Methoden aus der KI). Der Vorteil ist, dass die Ergebnisse sofort vorliegen und unmittelbar darauf reagiert werden kann.

### A3 Automatisierte Analysen beschreiben und bewerten

Navigationsgeräte im Auto melden oft: „Die Route wurde aufgrund der aktuellen Verkehrslage geändert.“

- Beschreibe Gründe, die zu einer solchen Änderung führen können.
- Beschreibe, welche Daten dafür erhoben werden müssen, wie diese Erhebung durchgeführt wird und warum die Ergebnisse schnell verfügbar sein müssen.
- Diese Meldung geht an alle Navigationsgeräte (zumindest dieses Herstellers). Beschreibe, welche Folgen das für den Verkehrsfluss haben könnte.

KI (Künstliche Intelligenz) versucht menschliches Verhalten mithilfe von Algorithmen nachzubilden.



## Chancen und Risiken automatisierter Analysen

### A4 Automatisierte Analysen beschreiben und bewerten

Bewertet die Chancen und Risiken der beschriebenen Datenanalysen.

- Mithilfe von Predictive Policing wird versucht, aufgrund von bisherigen Straftaten vorherzusagen, wo in Zukunft welche begangen werden.
- In der Volksrepublik China wurde ein Sozialkreditsystem eingeführt, bei dem man bei (aus Sicht der Regierung) wünschenswertem Verhalten Punkte erhält. Ist der Punktestand zu niedrig, drohen Einschränkungen.

Predictive Policing (engl.):  
vorausschauende Polizeiarbeit

Automatische Analysen können schnell Zusammenhänge in großen und sich schnell ändernden Datenbeständen entdecken. Werden daraufhin auch automatisiert Entscheidungen getroffen, ist eine sehr schnelle Reaktion möglich. Dabei besteht das Risiko, dass falsche Entscheidungen getroffen werden.

Mithilfe der **Datenanalyse** können Zusammenhänge erkannt und für Entscheidungen genutzt werden. Dies gilt vor allem für **verknüpfte** Datenbestände. Vor allem sich schnell ändernde Datenbestände werden permanent **automatisch** analysiert, und oft auch automatisiert Entscheidungen gefällt. Dies beinhaltet Chancen und Risiken.

**MERKE**

- Sofern vom Benutzer nicht abgeschaltet, ermitteln Mobiltelefone ständig ihren Standort und geben ihn weiter.
  - Bewerte Vor- und Nachteile, die sich daraus für
    - ▶ dich privat ergeben.
    - ▶ deine Freundesgruppe ergeben.
    - ▶ die Gesamtbevölkerung ergeben.
  - Begründe, warum du diese Funktion an- bzw. abschalten würdest.

**AUFGABEN**

## Am Ende dieses Unterkapitels hast du gelernt, ...

- ▶ Chancen und Risiken der automatisierten Analyse verknüpfter Datenbestände, auch im Hinblick auf gesellschaftliche Auswirkungen, zu bewerten.

**RÜCKBLICK**

## EINSTIEG

Die Softwarefirma Turing hat den Auftrag erhalten, eine Datenbank zur Notenverwaltung einer Schule zu erstellen. Als Anhaltspunkt wurde ihnen folgende Tabelle als Beispiel durch die Schulleitung übermittelt:

| Noten |             |      |        |        |        |             |         |
|-------|-------------|------|--------|--------|--------|-------------|---------|
| Std   | Name        | Fach | Klasse | Kürzel | Lehrer | Art         | Note    |
| 101   | Bob Meister | M    | 10a    | vog    | Vogel  | Ref, UB, SA | 1, 4, 3 |
| 102   | Clara Hoch  | Inf  | 10a    | vog    | Vogel  | UB          | 2       |
| 102   | Clara Hoch  | Phy  | 10a    | fin    | Fink   | UB, UB, SA  | 2, 1, 3 |

Attributwerte für Art:

Ref Referat

UB Unterrichtsbeitrag

SA Schulaufgabe

- ▶ Erkläre, warum Noten nicht wie in der Beispieltabelle gespeichert werden sollten.
- ▶ Schlage eine bessere Struktur vor.

## ERARBEITUNG

## Normalisierung

Schlecht modellierte Datenbanken können mithilfe von **Normalformen** vermieden werden. Der Vorgang der Überführung in eine den Normalformen entsprechende Struktur, wird als Normalisierung bezeichnet.

## 1. Normalform (1NF)

Eine Datenbank liegt in der 1. Normalform vor, wenn alle Attributwerte **atomar** sind. Dann steht in jeder Spalte ein nicht weiter zerlegbarer Wert.

In der Tabelle **Noten** sind die Werte der Spalten **Name**, **Art** und **Note** nicht atomar. Es gibt zwei Möglichkeiten, dies zu erreichen:

- ▶ Ein nicht atomares Attribut wird in mehrere Attribute aufgespalten, z. B. **Name** in **Vorname** und **Nachname**.
- ▶ Enthalten Attribute Listen (z. B. **Art** und **Note**), so wird für jeden Wert ein eigener Datensatz angelegt.

Notentabelle in der 1. Normalform:

| Noten |          |         |      |        |        |        |     |     |      |
|-------|----------|---------|------|--------|--------|--------|-----|-----|------|
| Std   | Nachname | Vorname | Fach | Klasse | Kürzel | Lehrer | PNr | Art | Note |
| 101   | Meister  | Bob     | M    | 10a    | vog    | Vogel  | 1   | Ref | 1    |
| 101   | Meister  | Bob     | M    | 10a    | vog    | Vogel  | 2   | UB  | 4    |
| 101   | Meister  | Bob     | M    | 10a    | vog    | Vogel  | 3   | SA  | 3    |
| 102   | Hoch     | Clara   | Inf  | 10a    | vog    | Vogel  | 1   | UB  | 2    |
| 102   | Hoch     | Clara   | Ph   | 10a    | fin    | Fink   | 1   | UB  | 2    |
| 102   | Hoch     | Clara   | Ph   | 10a    | fin    | Fink   | 2   | UB  | 1    |
| 102   | Hoch     | Clara   | Ph   | 10a    | fin    | Fink   | 3   | SA  | 3    |

Damit sind z. B. die beiden UB des Lehrers Fink für Clara Hoch unterscheidbar.

Zu den vorhandenen Primärschlüsselattributen **Std** und **Fach** muss jetzt zusätzlich **PNr** (Prüfungsnummer) hinzugefügt werden.

## A1 1. Normalform untersuchen

- Beschreibe, welche Anomalien in der 1. Normalform auftreten können.
- Erkläre nötige Änderungen, um die Anomalien aus a) zu vermeiden.

## 2. Normalform (2NF)

Eine Datenbank liegt in der 2. Normalform vor, wenn gilt:

- ▶ Alle Tabellen müssen in der 1. Normalform vorliegen.
- ▶ Alle Attribute, die nicht Teil des Primärschlüssels sind, müssen von allen Primärschlüsselattributen abhängen.

Vor- und Nachname sowie Klasse hängen nur von **Std** ab, nicht jedoch von **Fach** und **PNr**

Das kann durch eine Aufteilung in mehrere Tabellen erreicht werden.

Für die Tabelle **Noten** ergibt sich:

- ▶ **Nachname**, **Vorname** und **Klasse** kommen in die Tabelle **Schüler** mit dem Primärschlüsselattribut **SIID**.
- ▶ **Kürzel** und **Lehrer** kommen in die Tabelle **unterrichtet** mit den Primärschlüsselattributen **SIID** und **Fach**.
- ▶ **Art** und **Note** kommen in die Tabelle **Prüfung** mit den Primärschlüsselattributen **SIID**, **Fach** und **PNr**.

|          | SIID | Fach | PNr |
|----------|------|------|-----|
| Nachname | X    |      |     |
| Vorname  | X    |      |     |
| Klasse   | X    |      |     |
| Kürzel   | X    | X    |     |
| Lehrer   | X    | X    |     |
| Art      | X    | X    | X   |
| Note     | X    | X    | X   |

Notentabellen in der 2. Normalform:

| Schüler |          |         |        | unterrichtet |      |        |        |
|---------|----------|---------|--------|--------------|------|--------|--------|
| SIID    | Nachname | Vorname | Klasse | SIID         | Fach | Kürzel | Lehrer |
| 101     | Meister  | Bob     | 10a    | 101          | M    | vog    | Vogel  |
| 102     | Hoch     | Clara   | 10a    | 102          | Inf  | vog    | Vogel  |
|         |          |         |        | 102          | Ph   | fin    | Fink   |

### A2 2. Normalform erstellen und untersuchen

- a) Gib die Tabelle **Prüfung** an.
- b) Beschreibe mögliche Anomalien in der Tabelle **unterrichtet**.

### 3. Normalform (3NF)

Eine Datenbank liegt in der 3. Normalform vor, wenn gilt:

- ▶ Alle Tabellen müssen in der 2. Normalform vorliegen.
- ▶ Alle Attribute, die nicht Teil des Primärschlüssels sind, dürfen *nur* von Primärschlüsselattributen abhängen, nicht von einem anderen Nichtschlüsselattribut.

Im Beispiel hängt in der Tabelle **unterrichtet** das Attribut **Lehrer** vom Nichtschlüsselattribut **Kürzel** ab (oder umgekehrt).

Dies kann durch eine Aufteilung dieser Tabelle behoben werden:

| Prüfer |      |        | Lehrer |       |
|--------|------|--------|--------|-------|
| SIID   | Fach | Kürzel | Kürzel | Name  |
| 101    | M    | vog    | vog    | Vogel |
| 102    | Inf  | vog    | fin    | Fink  |
| 102    | Phy  | fin    |        |       |

Es gibt noch weitere Normalformen, die jedoch selten eine Rolle spielen.

Die Modellierung einer redundanzfreien und konsistenten Datenbank wird durch Anwendung der **drei Normalformen** in aufsteigender Reihenfolge gewährleistet.

**MERKE**

**1** Erstelle das durch die Normalisierung der Tabelle **Noten** entstandene Klassendiagramm (mit Kardinalitäten) und gib die Fremdschlüssel an.

**AUFGABEN**

**2** Ein Kino speichert seine Kartenreservierungen in dieser Tabelle:

| Buchung |        |     |        |       |       |       |            |       |
|---------|--------|-----|--------|-------|-------|-------|------------|-------|
| BNr     | Film   | KNr | Name   | Reihe | Platz | Preis | Datum      | Zeit  |
| 8023    | Hilde  | 335 | Grell  | 8     | 4     | 10    | 2022-09-08 | 20.00 |
| 8024    | Toys V | 337 | Ortega | 9     | 6     | 12    | 2022-09-08 | 20.00 |
| ...     | ...    | ... | ...    | ...   | ...   | ...   | ...        | ...   |

- a) Begründe, in welcher Normalform die Tabelle vorliegt.
- b) Normalisiere das Modell bis zur 3. Normalform.

**Redundanz** ↪ 2.1

Werden dieselben Daten in einem Datenbestand mehrfach gespeichert, so spricht man von Redundanz.

| Bestellung |            |                              |       |      |         |           |
|------------|------------|------------------------------|-------|------|---------|-----------|
| Händler    | Datum      | Adresse                      | TelNr | Anz. | Artikel | Kaufpreis |
| Brehm      | 2022-09-19 | 98000 Neustadt, Hafenstr. 14 | 58812 | 80   | Wasser  | 1,50      |
| Brehm      | 2022-09-20 | 98000 Neustadt, Hafenstr. 14 | 58812 | 30   | O-Saft  | 2,80      |
| ...        |            |                              |       |      |         |           |

redundant  
gespeichert:  
Händler,  
Adresse, TelNr

**Inkonsistenz** ↪ 2.1

Redundante Daten können widersprüchlich sein. Das bezeichnet man als Inkonsistenz.

| Bestellung |            |                          |        |      |         |           |
|------------|------------|--------------------------|--------|------|---------|-----------|
| Händler    | Datum      | Adresse                  | TelNr  | Anz. | Artikel | Kaufpreis |
| Kolb       | 2022-09-20 | 98001 Altdorf, Am Berg 7 | 22325  | 100  | Brezeln | 0,40      |
| Kolb       | 2022-09-20 | 98001 Altdorf Am Berg 7  | 223250 | 100  | Brezel  | 0,40      |
| ...        |            |                          |        |      |         |           |

widersprüchliche  
Informationen:  
Adresse,  
TelNr, Artikel

**Anomalie** ↪ 2.1

Enthält ein Datenbestand widersprüchliche Informationen, so liefern Abfragen falsche oder unvollständige Ergebnisse. Dies bezeichnet man als Anomalie. Sie können durch eine sinnvolle Modellierung (↪ 2.2) im Vorfeld verhindert werden. Bei Einbenutzerdatenbanken unterscheidet man die folgenden drei Anomalien.

**Änderungs-Anomalie (Update-Anomalie)**

↪ 2.1

Bei der Änderung von Daten werden nicht alle der redundant gespeicherten Informationen geändert.

Wird in einer SQL-Abfrage eine Selektion nach diesem Datum durchgeführt, so ist die Ergebnistabelle unvollständig.

Der Artikel **Breze** soll in **Brezel** geändert werden. Dies wurde jedoch bei einigen Bestellungen vergessen.

| Bestellung |     |         |       |
|------------|-----|---------|-------|
| Händler    | ... | Artikel | Preis |
| Kolb       |     | Breze   | 0,40  |
| Kolb       |     | Brezel  | 0,40  |
| ...        |     |         |       |

**Lösch-Anomalie (Delete-Anomalie)** ↪ 2.1

Beim gewollten Löschen von Datensätzen werden weitere Informationen unbeabsichtigt gelöscht, die an keiner anderen Stelle gespeichert sind.

Wird der einzige Datensatz von Händler **Ruben** gelöscht, gehen die Werte für **Adresse** und **TelNr** verloren.

| Bestellung |            |                             |       |      |         |       |
|------------|------------|-----------------------------|-------|------|---------|-------|
| Händler    | Datum      | Adresse                     | TelNr | Anz. | Artikel | Preis |
| Ruben      | 2022-05-16 | 98002 Erweiler Hügelpfad 10 | 44455 | 56   | O-Saft  | 0,50  |
| ...        |            |                             |       |      |         |       |

**Einfüge-Anomalie (Insertion-Anomalie)** ↪ 2.1

Neue Daten können nicht eingegeben werden, wenn beim Datensatz Teile des Primärschlüssels fehlen.

Das neue Produkt **waffeleis** und der zugehörige Preis können nicht eingetragen werden, da noch keine Bestellung vorliegt.

**Datenbankmodellierung: Aufteilung in mehrere Tabellen** ↪ 2.2

Redundanzen können vermieden werden, wenn der Inhalt einer Tabelle auf mehrere aufgeteilt wird, die dann jeweils inhaltlich zusammenhängende Daten enthalten.

| Bestellung |            |                              |       |      |         |           |
|------------|------------|------------------------------|-------|------|---------|-----------|
| Händler    | Datum      | Adresse                      | TelNr | Anz. | Artikel | Kaufpreis |
| Brehm      | 2022-09-09 | 98000 Neustadt, Hafenstr. 14 | 58812 | 80   | Wasser  | 1,50      |

| Bestellung |            |      |         |
|------------|------------|------|---------|
| Händler    | Datum      | Anz. | Artikel |
| Brehm      | 2022-09-09 | 80   | Wasser  |

| Artikel     |           |  |
|-------------|-----------|--|
| Bezeichnung | Kaufpreis |  |
| Wasser      | 1,50      |  |

| Händler |                              |       |
|---------|------------------------------|-------|
| Name    | Adresse                      | TelNr |
| Brehm   | 98000 Neustadt, Hafenstr. 14 | 58812 |

**Datenbankmodellierung: Aufteilung in mehrere Spalten** ↪ 2.2

Die Werte innerhalb einer Spalte sollten atomar vorliegen, d. h. keine zusammengesetzten Informationen enthalten. Dies erleichtert Abfragen, die nach dieser Information selektieren.

| Anschrift                    |  |  |  |
|------------------------------|--|--|--|
| 98000 Neustadt, Hafenstr. 14 |  |  |  |

↓

| PLZ   | Ort      | Straße    | HsNr |
|-------|----------|-----------|------|
| 98000 | Neustadt | Hafenstr. | 14   |

zusammengesetzte Daten

atomare Daten

**Datenmodellierung** ↪ 2.3

Zusätzlich zum in 1.7 beschriebenen Vorgehen ist zu berücksichtigen:

Sicherung der Konsistenz:

- ▶ Übernimmt bei Fremdschlüsseln meist das DBS.
- ▶ Daten, die gleich sein müssen, dürfen nur einmal gespeichert werden.
- ▶ Es dürfen keine Daten gespeichert werden, die aus anderen berechenbar sind.

Abfragen berücksichtigen:

- ▶ Atomare Speicherung von Daten, nach denen dann selektiert werden soll.

Änderungen berücksichtigen:

- ▶ Absehbare Änderungen sollten keine Änderung der Struktur nötig machen.

**Automatische Analyse verknüpfter****Datenbestände** ↪ 2.4

Mithilfe der Datenanalyse werden aus Datenbeständen neue Erkenntnisse gewonnen. Werden dabei unterschiedliche Datenbestände verknüpft, sind Erkenntnisse möglich, die aus einem einzelnen Datenbestand nicht gewonnen werden können. Werden Daten ständig automatisch erhoben, entstehen riesige, sich schnell ändernde Datenmengen. Diese werden automatisch mit vorher festgelegten Verfahren analysiert. Dabei müssen Chancen und Risiken gegeneinander abgewogen werden.

Standortdaten von Mobiltelefonen:

- ▶ Wartezeiten z. B. beim Arzt

Kontaktdaten mit anonymem Fragebogen:

- ▶ Zuordnung der Bewertung

Standortdaten und Geschwindigkeit vieler Fahrzeuge:

- ▶ Alternativstrecke berechnen
- ▶ automatische Geschwindigkeitsbegrenzung

Standortdaten von Mobiltelefonen:

- ▶ Chance: Kontaktverfolgung bei ansteckenden Krankheiten
- ▶ Risiko: Bewegungsprofil erstellen

## ↪ 2.1

Ein mögliche Objektkarte:

| W153: Wohnung |                  |
|---------------|------------------|
| Haus          | = C              |
| Stockwerk     | = 5              |
| WNr           | = 504            |
| Typ           | = K7             |
| Größe         | = 85             |
| Miete         | = 635 €          |
| Mieter        | = Fa. Cx24R, ... |
| mitKeller     | = nein           |

Fortsetzung: ↪ Aufgabe 2

- 1 Die Firma WohnGut vermietet eine große Zahl von Wohnungen. Die Mieter sind meist Firmen, die mehrere Wohnungen gemietet haben, um dort zeitweise Angestellte unterzubringen. Das Tabellenschema der Tabelle sieht so aus:  
 Wohnung(Haus: Zeichen; Stockwerk: Zahl; WNr: Zahl;  
 Typ: Text; Größe: Zahl; Miete: Währung; Mieter: Text;  
 mitKeller: Wahrheitswert)

Beachte:

Die Häuser werden mit A, B, C, ... bezeichnet, die Stockwerke mit 0; 1; 2; ... und die Wohnungen in einem Haus mit einer dreistelligen Zahl, wobei die erste Ziffer für das Stockwerk steht. Die Mieter werden mit allen Kontaktdaten eingetragen. Alle Wohnungen des gleichen Typs haben dieselbe Größe und kosten dieselbe Miete. Ob ein Kellerraum dabei ist, spielt für den Typ keine Rolle.

- Gib einen geeigneten Primärschlüssel an.
- Das Stockwerk ist sowohl einzeln gespeichert als auch in der Wohnungsnummer (WNr) enthalten. Begründe, warum diese doppelte Speicherung trotzdem sinnvoll sein kann.
- Gib weitere redundant gespeicherte Informationen an.
- Beschreibe, wie dadurch Inkonsistenzen entstehen können.
- Beschreibe, wie durch folgende Aktionen Anomalien entstehen können und um welche es sich handelt.
  - ▶ Die Fa. VR7 kündigt mangels Bedarf alle aktuellen Mietverträge, möchte jedoch später wieder Wohnungen mieten.
  - ▶ Die Miete für die Wohnungen vom Typ R9 wird um 10% erhöht.
  - ▶ Ein neuer Kunde interessiert sich für Wohnungen, im Moment sind aber keine frei.
  - ▶ Die Fa. Cx24R hat eine neue Anschrift.

## ↪ 2.2

Fortsetzung: ↪ Aufgabe 3

- 2 Fortsetzung von ↪ Aufgabe 1  
 Die Datenstruktur soll so verbessert werden, dass keine Anomalien mehr auftreten.
- Beschreibe in Worten, wie du dabei vorgehen musst.
  - Entwickle ein objektorientiertes Datenmodell und stelle es grafisch dar (↪ 1.7).
  - Begründe, warum nun die in Aufgabe 1e) beschriebenen Anomalien nicht mehr auftreten können.

## ↪ 2.3

- 3 Fortsetzung von ↪ Aufgabe 2
- Analysiere das Datenmodell aus Aufgabe 2b) und überführe es in ein relationales Modell (↪ 1.7).
  - Gib die nötigen Änderungen an, wenn sich nachträglich folgende Anforderungen ergeben. Diskutiere, ob es verschiedene Möglichkeiten gibt, diese umzusetzen. Die Mieter möchten wissen,
    - ▶ ob sich im Bad eine Dusche oder eine Badewanne befindet.
    - ▶ ob im Haus ein Aufzug vorhanden ist.
    - ▶ wie groß der Kellerraum ist.
  - Mit einer SQL-Abfrage sollen alle Wohnungen im Haus B ermittelt werden, deren Mietpreis weniger als 500 € beträgt. Konzipiere eine Abfrage und zeichne das zugehörigen Datenflussdiagramm.

4 Die Tabelle **Medizin** enthält medizinische Daten.

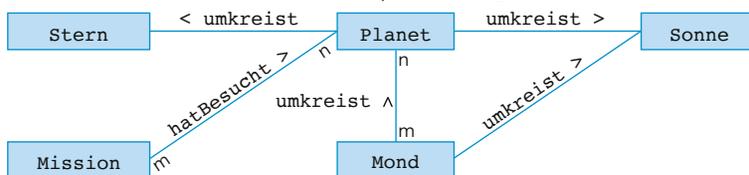
- a) Beschreibe die gespeicherten Informationen.
- b) Untersuche mit Hilfe passender SQL-Abfragen, ob es einen Zusammenhang zwischen folgenden Eigenschaften gibt (Tipp: Vierfeldertafel):  
Geschlecht – Raucher; Raucher – Krankenhauskosten; Alter – Raucher
- c) Begründe, wer ein Interesse an solchen Zusammenhängen haben könnte.
- d) Beschreibe jeweils die daraus folgenden Konsequenzen für dich.
- e) Beurteile die Konsequenzen für die Bürger, wenn diese Zusammenhänge aufgrund falscher oder manipulierter Datensätze gefunden wurden.



38010-10  
benötigte Tabelle:  
Medizin

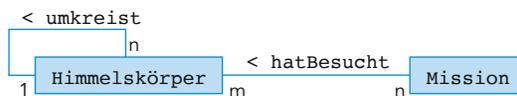
↪ 2.4

5 Die Mitglieder der Software AG haben ein erstes, noch fehlerhaftes Klassendiagramm einer Datenbank für die Speicherung astronomischer Objekte erstellt.



- a) Beschreibe alle Fehler im Diagramm.
- b) Entwickle ein verbessertes Datenmodell (↪ 1.7) und stelle es grafisch in Form eines Klassendiagramms dar.
- c) Begründe, warum in allen hier auftretenden Klassen **Name** als Primärschlüssel möglich ist, bei Personen z. B. in einem Verein jedoch nicht.
- d) Begründe, warum es nicht sinnvoll ist sowohl die Attribute **Halbachse** (der Bahn), **Umlaufzeit** als auch **Bahngeschwindigkeit** von Planeten und Monden zu speichern.

e) Die Leiterin der Software AG, Frau Beimer, schlägt dieses Klassendiagramm vor. Es soll weiterhin möglich sein, z. B. eine Liste aller Sterne (oder Planeten) in der Datenbank zu erstellen.



Beschreibe, welche Voraussetzungen die Tabelle **Himmelskörper** dazu erfüllen muss.

Begründe, dass es nun auch möglich ist, weitere Himmelskörper wie z. B. Kleinplaneten und Asteroiden mit in der Tabelle zu speichern.

- f) 1803 entdeckte Wilhelm Herschel den ersten Doppelstern. Auch solche Systeme können Planeten haben, die dann um mehr als einen Stern kreisen. Beschreibe die nötigen Änderungen am Datenmodell und am relationalen Modell, um auch diese Fälle berücksichtigen zu können.
- g) Vergleiche für die beiden relationalen Modelle, wie die Daten von Missionen gespeichert werden, die neben Planeten auch Monde und Asteroiden besuchen.  
Gib für die Mission New Horizons (siehe rechts) alle nötigen Datensätze im Modell von Teilaufgabe e) an.
- h) Gib für das relationale Modell, das aus Teilaufgabe e) folgt, jeweils eine SQL-Abfrage zur Beantwortung folgender Fragen an:
  - ▶ Welche Planeten sind Exoplaneten?
  - ▶ Welche Monde umkreisen den Planeten Jupiter?
  - ▶ Welche Sterne gehören zu einem Mehrfachsternsystem?



Unter Mission, genauer Raumfahrtmission, versteht man das Entsenden einer Raumsonde zu Forschungszwecken.

Asteroiden sind kleinere Körper, die um die Sonne kreisen.

Als Doppelstern bezeichnet man zwei Sterne, die sich gegenseitig umkreisen. Inzwischen kennt man auch Systeme mit mehr als zwei Sternen.

Während der Mission New Horizons hat die Raumsonde u. a. den Asteroiden (132524) APL, die Planeten Jupiter und Pluto sowie die Monde des Pluto erforscht.

Exoplaneten sind Planeten, die sich außerhalb unseres Sonnensystems befinden.



38010-11

benötigte Tabelle:  
Tab1, Tab2, Tab3

38010-12

benötigte Tabelle:  
Kredit

6 Untersuche die Tabellen Tab1, Tab2 und Tab3 mithilfe von SQL-Abfragen.

- Beschreibe die Art der gespeicherten Informationen.
- Gib für jede Tabelle eine sinnvolle Bezeichnung an.
- Entwickle aufgrund der Ergebnisse deiner Untersuchungen das Datenmodell und überführe es in ein relationales Modell.



7 Untersuche die Tabelle Kredit.

- Beschreibe die Art der gespeicherten Informationen.
- Erkläre, inwiefern ein Kreditinstitut aus diesen Informationen Nutzen ziehen und sie als Grundlage für zukünftige Entscheidungen verwenden kann.
- Beschreibe negative Auswirkungen für die Kunden, wenn die Daten in falsche Hände geraten.

8 Herr Bader, der Leiter der Musikkapelle Altneustadt, hat die Daten über die vielen Mitglieder und das vorhandene Inventar bisher in diesen beiden Tabellen in einem Tabellenkalkulationsprogramm verwaltet:

| Mitglied |        |         |                    |                 |
|----------|--------|---------|--------------------|-----------------|
| MNr      | Name   | Vorname | ... spielt1        | spielt2         |
| 1        | Bader  | Hans    | Trompete, gut      |                 |
| 2        | Danner | Luise   | Trompete, sehr gut | Klarinette, gut |
| 3        | Kasch  | Werner  | Tuba, mittelmäßig  | Posaune, gut    |
| ...      |        |         |                    |                 |

| Inventar |            |            |     |
|----------|------------|------------|-----|
| InvNr    | Art        | gekauftAm  | ... |
| 1        | Trompete   | 2015-06-14 |     |
| 2        | Trompete   | 2014-10-22 |     |
| 3        | Klarinette | 2021-02-03 |     |
| ...      |            |            |     |

Die Mitglieder spielen ihre eigenen Instrumente, können sich aber auch welche aus dem Inventar ausleihen. Herr Bader möchte die Daten lieber in einem DBS speichern, da er aus folgenden Gründen nicht zufrieden ist:

- ▶ Es gibt auch Mitglieder, die mehr als zwei Instrumente spielen.
  - ▶ Es ist nicht ersichtlich, wer welches Instrument ausgeliehen hat.
  - ▶ Er muss oft für einen Auftritt schnell Ersatzleute finden, die ein bestimmtes Instrument mindestens mittelmäßig spielen können.
- Begründe, inwiefern die zusätzliche Tabelle Instrument (siehe links) hilfreich sein kann. Gehe auch darauf ein, wie weitere Angaben (z. B. wie gut spielt ein Mitglied ein Instrument) dabei berücksichtigt werden können.
  - Entwickle unter Berücksichtigung aller Anforderungen ein objektorientiertes Datenmodell und stelle es grafisch dar (↪ 1.7 und ↪ 2.3).
  - Überführe das objektorientierte Datenmodell in ein relationales (↪ 1.7).
  - Frau Danner macht Herrn Bader darauf aufmerksam, dass es auch zwei Sänger gibt, die kein Instrument spielen. Beschreibe, wie sich dies durch Einführung eines „Instruments“ ohne Änderung des Modells lösen lässt.
  - Herr Bader sucht nach Mitgliedern, die Trompete mindestens mittelmäßig spielen können. Gib dazu eine SQL-Abfrage an.
  - Herr Bader empfindet die Abfrage aus e) ziemlich kompliziert. Da er solche Daten häufig braucht, überlegt er, ob das nicht einfacher ginge. Gib Änderungen am relationalen Modell an, die einfachere Abfragen möglich machen. Formuliere dann die Abfrage aus Teilaufgabe e) neu.
  - Da sich die Daten immer wieder ändern, möchte Herr Bader aktuelle Listen, welches Mitglied sich welches Instrument von welcher Art ausgeliehen hat. Formuliere eine entsprechende SQL-Abfrage.
  - Herr Kasch schlägt vor, auch das Alter der Leihinstrumente zu speichern. Diskutiere diesen Vorschlag.

MNr = Mitgliedsnummer  
InvNr = Inventarnummer



| Instrument |
|------------|
| Art        |
| Trompete   |
| Posaune    |
| Klarinette |
| ...        |

- 9 Der neue Paketlieferdienst „Speedy“ in Dacity beauftragt die Softwarefirma „Klein und Weich“ eine Datenbank zur Verwaltung der Pakete, Fahrzeuge und Paketboten zu erstellen. Folgende Daten sollen gespeichert werden:
- ▶ Fahrzeuge: Typ, Kilometerstand, Baujahr, Kennzeichen, nächste Wartung, maximale Zuladung
  - ▶ Paketboten: Name, Vorname, Geburtsdatum, Adresse, Telefonnummer, aktuelles Gehalt, derzeitige Überstunden
  - ▶ Pakete: Zieladresse, Absenderadresse, Masse, Auslieferungsfahrzeug, zerbrechlich oder nicht, Datum und Uhrzeit der Auslieferung
- a) Analysiere und strukturiere den Datenbestand (Klassen, Beziehungen).
  - b) Entwickle aus dem Datenbestand ein objektorientiertes Datenmodell. Beachte dabei, dass
    - ▶ Redundanzen möglichst vermieden werden.
    - ▶ in einer Klasse nur zusammengehörige Daten gespeichert werden.
    - ▶ die Daten atomar sind.
    - ▶ keine Daten gespeichert werden, die aus anderen berechenbar sind.
 Stelle das Datenmodell grafisch dar (Klassenkarten, Klassendiagramm).
  - c) Überführe das objektorientierte Datenmodell in ein relationales Modell. Mache deutlich, welcher Fremdschlüssel sich auf welchen Primärschlüssel bezieht (Tabellenschema, Datentypen, Primärschlüssel, Fremdschlüssel).
  - d) Gib zu jedem Tabellenschema zwei plausible Datensätze an.
  - e) Die Fahrzeuge dürfen nicht überladen werden. Diskutiere, ob es sinnvoll ist, ein Attribut **Gesamtmasse** (aller Pakete) einzuführen.
  - f) Erstelle zur Beantwortung der folgenden Fragen passende SQL-Abfragen:
    - ▶ „Welche Fahrzeuge müssen nächstes Jahr zum Kundendienst?“
    - ▶ „Welcher Bote hat mehr als zehn Überstunden?“
    - ▶ „Wer fährt im Moment das Fahrzeug mit dem Kennzeichen DC-AC 42 und welche Pakete sind dort geladen?“



Orientiere dich an den Kapiteln  
 ↳ 1.7 und ↳ 2.3.

- 10 Der Cateringdienst „Good Chef“ digitalisiert seinen Betrieb. Beim Auftrag für die Erstellung einer Datenbank macht er folgende Vorgaben:



- ▶ Es müssen Informationen über Personen gespeichert werden (z. B. Name, Adresse). Für die Angestellten zusätzlich der Verdienst, bei den Kunden zusätzlich die (evtl. abweichende) Lieferadresse.
  - ▶ Jeder Auftrag wird mit den bestellten Speisen und dem Lieferdatum gespeichert.
  - ▶ Jeder Auftrag hat einen verantwortlichen Angestellten und mehrere weitere, die diesen Auftrag und parallel dazu evtl. auch andere bearbeiten.
  - ▶ Kunden können mehrere Aufträge erteilen, jeder Auftrag wird einem Kunden zugeordnet.
- a) Ermittelt in der Gruppe welche Attribute für die Tabellen **Arbeiter**, **Kunde**, **Speise** und **Auftrag** nötig sind.
  - b) Entwickelt ein objektorientiertes Datenmodell und stellt es grafisch dar.
  - c) Überführt das objektorientierte Datenmodell in ein relationales Modell.
  - d) Implementiert die Datenbank in einer Datenbanksoftware und trägt in jeder Tabelle mindestens fünf Beispieldatensätze ein.
  - e) Überlegt euch mögliche Fragen, die mit Hilfe der Datenbank beantwortet werden könnten und erstellt passende SQL-Abfragen dazu, z. B.: „Welche Aufträge hat Hans Müller erteilt?“

Orientiert euch an den  
 Kapiteln ↳ 1.7 und ↳ 2.3.

## DAS GROSSE INFO-QUIZ!

Hier sprechen Begriffe, die du in diesem Kapitel gelernt hast, über sich. Die Zahlen in Klammern geben den Buchstaben im erratenen Begriff an, den du für das Lösungswort brauchst.

- A** „Ich bin der Vorgang, mit dem du neue Erkenntnisse gewinnst.“ .....(1)  
**B** „Ich bin ein Spezialfall von D und trete auf, wenn ein Datum nicht in allen Datensätzen geändert wird.“ (engl. Bezeichnung) .....(5)  
**C** „Werden zu viele Daten in sehr kurzer Zeit angeliefert, muss man sie so eintragen.“ .....(4)  
**D** „Werden Tabellen schlecht modelliert, trete ich bei Abfragen auf.“ .....(4)  
**E** „Das musst du mit einer Tabelle machen, wenn sie Informationen enthält, die nicht zusammengehören.“ .....(1)  
**F** „Ich bin gut bei der Datensicherung, aber nicht bei der Eingabe von Daten.“ .....(1)



38010-13

Hast du das Ziel des Kapitels erreicht? Schätze dich zunächst mithilfe des hinterlegten Bogens selbst ein. Bearbeite dann die Aufgaben und vergleiche mit deiner Selbsteinschätzung.

- 1** Helene möchte schnell von ihren CDs diejenigen herausuchen können, die bestimmte Songs oder Songs von bestimmten Interpreten enthalten. Analysiere das Problem und entwickle ein passendes objektorientiertes Datenmodell. Dabei soll die Konsistenz gewahrt werden können. Berücksichtige auch, dass ein Song mehr als einen Interpreten haben und sich auf mehr als einer CD befinden kann. Gib dann das zugehörige relationale Modell an.
- 2** Das Sekretariat des Dijkstra-Gymnasiums führt in einer Tabelle Buch, welche Kommunikation zwischen den Eltern und der Schule stattgefunden hat. Nachträglich werden die rot markierten Änderungen durchgeführt.

| Kontakt |        |            |                            |                           |                                           |             |
|---------|--------|------------|----------------------------|---------------------------|-------------------------------------------|-------------|
| Id      | Eltern | Datum      | Art                        | TelNr                     | E-Mail                                    | beantwortet |
| 1       | Kohl   | 2022-10-11 | Telefon                    | 55676                     | kohl@mail.de                              | ja          |
| 2       | Baum   | 2022-10-11 | Mail                       | 47526                     | b129@bmail.de                             | ja          |
| 3       | Nguyen | 2022-10-11 | persönlich                 | <del>12900</del><br>32943 | g.nguyen@mail.de                          | ja          |
| 4       | Nguyen | 2022-10-12 | Mail                       | 12900                     | g.nguyen@mail.de                          | ja nein     |
| 5       | Baum   | 2022-10-12 | <del>Mail</del><br>Telefon | 47526                     | b129@bmail.de                             | nein ja     |
| 6       | Yilmaz | 2022-10-12 | Telefon                    | 85774                     | <del>y44@tmail.de</del><br>mail@yilmaz.de | ja          |

- a) Gib eine Erklärung für die durchgeführten Änderungen an!  
 b) Begründe Anomalien, die entstehen können. Benenne diese mit passendem Fachausdruck.  
 c) Beschreibe, wie sich diese Anomalien vermeiden lassen.
- 3** Soziale Netzwerke erhalten durch die gespeicherten Profile sehr viele Kundeninformationen.
- a) Nenne einige Informationen, die gespeichert werden könnten.  
 b) Bei manchen sozialen Netzwerken wird bei den Bildern eine automatische Gesichtserkennung durchgeführt. Recherchiere, wozu das genutzt wird.  
 c) Begründe das Interesse von z. B. Versicherungen oder Online-Händlern an diesen Daten.  
 d) Vor einiger Zeit wurde bekannt, dass amerikanische Firmen von Bewerbern ihre Zugangsdaten zu sozialen Netzwerken verlangt haben. Lege die Gründe dafür dar.  
 e) Alena verlässt das soziale Netzwerk, in dem sie bisher aktiv war, die meisten ihrer Freunde und Freundinnen nicht. Überlege, welche Gründe sie für diesen Schritt haben könnte und welche Nachteile ihr dadurch entstehen könnten.

AdobeStock / bildergala – S. 94; - / dlyastokiv – S. 136; - / Ilya\_kovshik – S. 40; - / Inm – S. 3, 51; - / Robert Kneschke – Cover; - / Dusan Kostic – S. 124; - / SecondSide – S. 58; - / VectorMine – S. 134; - / VIAR PRO Studio – S. 52; Fotolia / adiruch na chiangmai – S. 135; - / freshidea – S. 27; - / Kaliste A – S. 54; - / meryll – S. 4, 71; - / pixelunikat – S. 73; - / Stefan Richter – S. 83; - / Dmitry Vereshchagin – S. 4, 71; Getty Images Plus / Digital-Vision, Image Source – S. 114; - / Dorling Kindersley – S. 27; - / iStock Editorial, Real Life! – S. 48; - / iStock Editorial, Seremin – S. 79; - / iStockphoto – S. 103; - / iStockphoto, ahturner – S. 39; - / iStockphoto, Daniel Albach – S. 137; - / iStockphoto, AlexMaster – S. 29; - / iStockphoto, ArisSu – S. 80; - / iStockphoto, artorn – S. 104; - / iStockphoto, Michael Blann – S. 96; - / iStockphoto, BradCalkins – S. 104; - / iStockphoto, Michael Burrell – S. 35; - / iStockphoto, Cameris – S. 99; - / iStockphoto, Holly Cromer – S. 96; - / iStockphoto, ctermit – S. 76 (2); - / iStockphoto, dobok – S. 72; - / iStockphoto, Edafoto – S. 76; - / iStockphoto, eriksvoboda – S. 95; - / iStockphoto, fermate – S. 29; - / iStockphoto, gorodenkoff – S. 126, 134; - / iStockphoto, Highwaystarz-Photography – S. 74; - / iStockphoto, Oleksandr Hruts – S. 138; - / iStockphoto, ipopba – S. 86, 93, 97; - / iStockphoto, ismagilov – S. 73, 92; - / iStockphoto, JaySi – S. 99; - / iStockphoto, Jeevan GB – S. 143; - / iStockphoto, Jurgute – S. 92; - / iStockphoto, klikk – S. 29; - / iStockphoto, koksikoks – S. 139; - / iStockphoto, Sanjin Kusan – S. 3, 7; - / iStockphoto, LightFieldStudios – S. 143; - / iStockphoto, Lightstar59 – S. 96; - / iStockphoto, LiliGraphie – S. 136; - / iStockphoto, losw – S. 21; - / iStockphoto, Macrovector – S. 127; - / iStockphoto, matimix – S. 102; - / iStockphoto, MaYcaL – S. 94; - / iStockphoto, Melpomenem – S. 137; - / iStockphoto, monkeybusinessimages – S. 4, 101, 108, 116; - / iStockphoto, Jordi Mora Igual – S. 117; - / iStockphoto, Nerthuz – S. 49; - / iStockphoto, opico – S. 106; - / iStockphoto, Pheelings Media – S. 82; - / iStockphoto, pixelliebe – S. 140 (5); - / iStockphoto, Pkpix – S. 73; - / iStockphoto, Piotr Polaczyk – S. 97; - / iStockphoto, Prostock-Studio – S. 85; - / iStockphoto, **Михаил Руденко** – S. 47; - / iStockphoto, Rassco – S. 142; - / iStockphoto, Rawpixel – S. 20; - / iStockphoto, Larisa Rudenko – S. 95; - / iStockphoto, Jochen Sand – S. 117; - / iStockphoto, sanjagrujic – S. 95; - / iStockphoto, SementsovaLesia – S. 78; - / iStockphoto, TimArbaev – S. 5, 131; - / iStockphoto, turk\_stock\_photographer – S. 93; - / iStockphoto, upixa – S. 99; - / iStockphoto, urbazon – S. 82; - / iStockphoto, vchal – S. 112; - / iStockphoto, Igor Vershinsky – S. 100; - / iStockphoto, Robert Way – S. 4, 101; - / iStockphoto, wragg – S. 29; - / iStockphoto, XXLPhoto – S. 41; - / OJO Images, Paul Bradbury – S. 122; iStockphoto / DrAfter123 – Cover; picture-alliance / Süddeutsche Zeitung Photo, Rumpf, Stephan – S. 3, 33; Pixabay / Jazella – S. 112; - / Peggy und Marco Lachmann-Anke – S. 138.



T38010