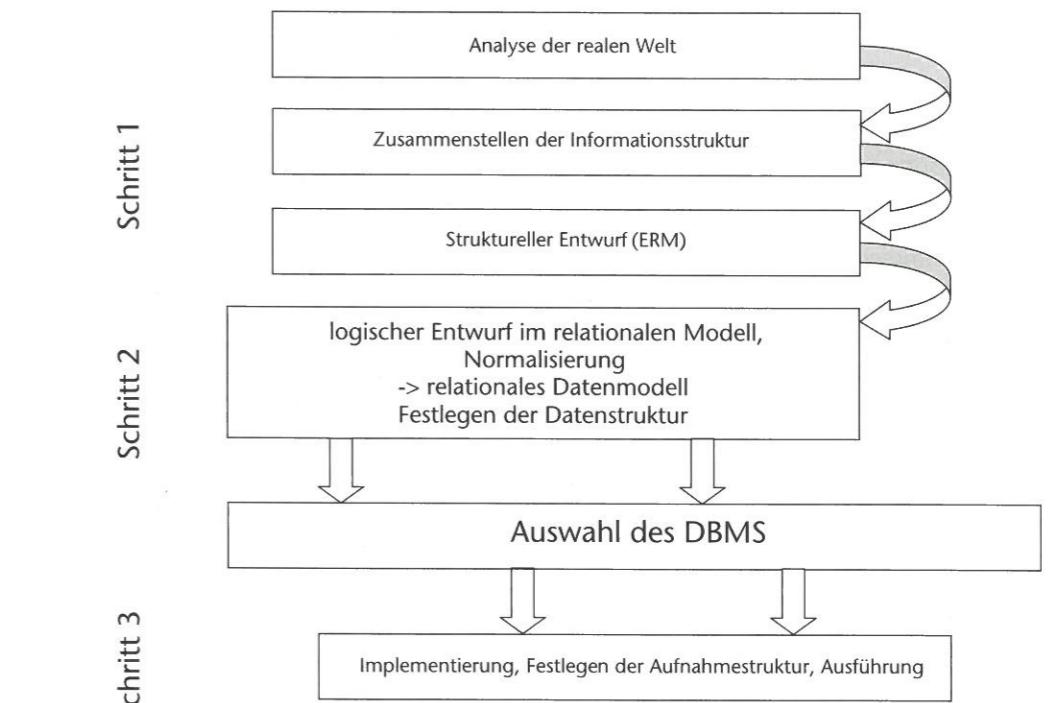


# 3 Entwicklung einer Datenbank und Normalisierung

## 3.1 Datenbankentwicklung

Die Entwicklung einer Datenbank wird in mehreren Schritten vollzogen.



### Schritt 1: Festlegen der Informationsstruktur

Zuerst erfolgt eine Analyse der realen Welt mit daraus resultierenden Informationsanforderungen. Welche Informationen erwartet der Anwender vom Datenbanksystem? Daraus wird ein ERM als grafische Darstellung der Zusammenhänge erstellt.

### Schritt 2: Festlegen der Datenstruktur

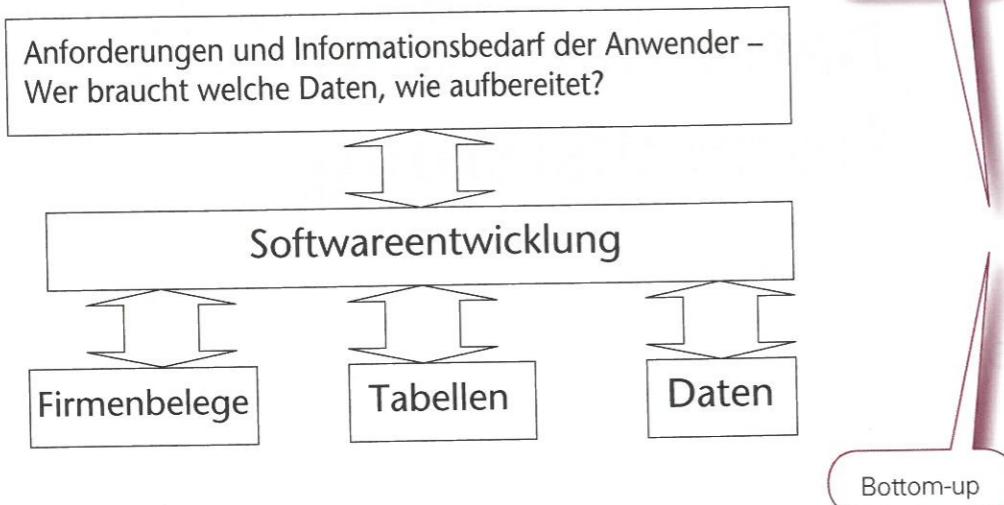
Beantwortet wird die Frage, welche Beziehungen die Daten und Tabellen untereinander besitzen. Der Entwurf der Objekte (Entitäten) und deren Beziehungen zueinander erfolgt. Das Ergebnis von Schritt 1 und Schritt 2 ist das relationale Datenmodell, das in der dritten Normalform vorliegen sollte (siehe folgendes Kapitel).

### Schritt 3: Festlegen der Aufnahmestruktur

Nun werden die Tabellen und die einzelnen Datenfelder mit ihren Datentypen festgelegt. Nach der Auswahl eines geeigneten DBMS wird die zukünftige Datenbank mithilfe einer Software angelegt.

Die Informationsstruktur der Datenbank gemäß Schritt 1 kann durch das Bottom-up-Verfahren oder das Top-down-Verfahren ermittelt werden. In der Praxis werden meist beide Verfahren kombiniert.

## 3.1.1 Verfahren der Software-Entwicklung

**Bottom-up-Verfahren:**

Beim Bottom-up-Verfahren (= von Grund aufwärts) müssen sämtliche im Unternehmen vorhandene Schriftstücke und Daten untersucht werden, um die Informationen abzuleiten, die das Datenbanksystem liefern soll. Diese Entwurfsmethode kann zu einer Vielzahl sehr unterschiedlicher Lösungen führen. Es besteht die Gefahr, dass sich die Lösungen verschiedener Aufgabenstellungen überschneiden und gleiche Daten mehrfach in verschiedenen Tabellen gespeichert werden (Redundanzen). Redundanzen müssen vermieden werden. Bei der Änderung eines Wertes, z. B. einer Adresse, muss die Bearbeitung eines einzigen Datensatzes die entsprechende Änderung in der Datenbank sicherstellen.

**Hinweis:**

Das Vorhandensein überflüssiger, nicht notwendiger Datensätze nennt man **Redundanz**. Redundanzen dürfen in relationalen Datenbanken nicht vorkommen.

**Top-down-Verfahren**

Hier bestimmt nicht die einzelne Anwendung das Datenmodell, sondern beim Top-down-Verfahren (= von der Spitze abwärts) werden zunächst die Informationsanforderungen aller späteren Datenbankbenutzer festgestellt. Aus diesen unterschiedlichen Anforderungen wird anschließend abgeleitet, welche Informationen z. B. über die Kunden in der Datenbank enthalten sein müssen. Somit gibt es für alle Datenbankanwendungen nur eine einzige Tabelle Kunden.

## 3.2 Normalisierung

Die durch die Informationsstruktur festgelegten Datenfelder können nicht ohne weitere Vorarbeit in eine Tabelle übernommen werden. Es muss gewährleistet sein, dass keine Mehrfachspeicherungen durchgeführt werden (Redundanzen). Der Vorgang, bei dem die Daten stufenweise in verschiedene Tabellen aufgegliedert werden, bezeichnet man als **Normalisierung**. Dabei werden unterschiedliche Normalformen beschrieben.

**Hinweis:**

Durch die Normalisierung werden Datenfelder schrittweise nach logischen Regeln auf verschiedene Tabellen verteilt.

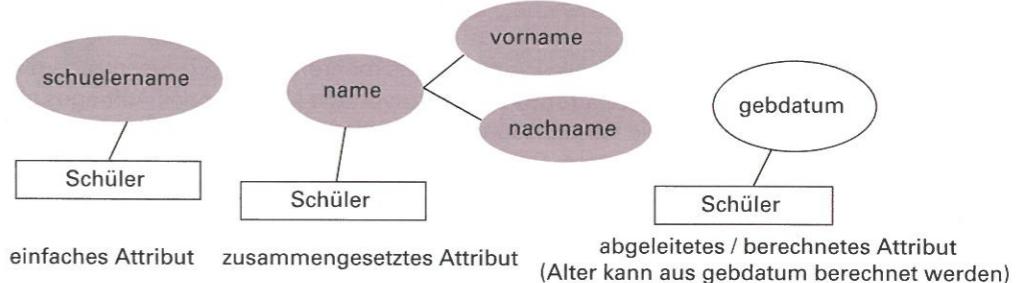
Befindet sich eine Tabelle nicht in der dritten Normalform, können Fehler auftreten. Diese Fehler werden auch als Anomalien bezeichnet.

INFO: Anomalien in Datenbanken treten bei einer nicht existierenden oder fehlerhaften Normalisierung auf

Anomalien innerhalb einer Tabelle können beim Einfügen, Löschen oder Ändern von Datensätzen auftreten. Man unterscheidet drei Anomalien:

**Einfüge-Anomalie**

Wird ein neuer Schüler eingefügt, der keinen Primärschlüsselwert „schuelernr“ eingetragen hat, dann kann dieser Schüler nicht an einem Kurs teilnehmen. In einigen DBMS ist es gar nicht möglich, einen solchen Datensatz zu speichern.

**Lösch-Anomalie**

Wird ein Schüler in der Tabelle **Schüler** gelöscht, werden auch seine Noten in der Tabelle **Belegung** gelöscht.

**Änderungs-Anomalie**

Wird die Kursnummer „kursnr“ geändert, muss diese auch in allen verbundenen Datensätzen in der Tabelle **Belegung** mit Fremdschlüssel „kursnr“ geändert werden.

## 3.2.1 Normalformen

**Erste Normalform**

Werden in einer Tabelle **Personal** im Feld **wohnort** sowohl die Postleitzahl als auch der Name des Ortes gespeichert, so ergeben sich mehrere Probleme. Zum einen sind größeren Orten, z. B. Ulm, mehrere Postleitzahlen zugeordnet, zum anderen ist es dadurch schwierig, nach den Ortsnamen zu sortieren und zu suchen.

Deshalb müssen zunächst zusammengesetzte Informationen in verschiedenen Feldern als einfache, nicht mehr aufteilbare Werte gespeichert werden. Man nennt diese Werte auch atomare Werte.

**Tabelle: Personal**

pn	name	vorname	wohnort
1	Palmer	Carlo	89348 Jettingen
2	Müller	Herta	86416 Krumbach
3	Mai	Hans	89077 Ulm
4	Schulz	Anna	88471 Laupheim
5	Winter	Susanne	89312 Günzburg

Felder enthalten  
2 Informationen

**Tabelle: Personal**

pn	name	vorname	wohnort	pl
1	Palmer	Carlo	Jettingen	89348
2	Müller	Herta	Krumbach	86416
3	Mai	Hans	Ulm	89077
4	Schulz	Anna	Laupheim	88471
5	Winter	Susanne	Günzburg	89312

Feldinhalte sind  
aufgeteilt

Tabelle **Personal** unnormalisiert und in der ersten Normalform

**Hinweis:**

Eine Tabelle befindet sich in der ersten Normalform, wenn alle Datenfelder nicht mehr aufteilbare Werte enthalten.

Kunden, Lagerteile, Bestellungen, andere Objekte und Vorgänge sind über die Angabe des Primärschlüsselfeldes eindeutig festlegbar. Sucht man nach einer Personalnummer, kann man alle abgespeicherten Mitarbeiterdaten, z. B. den entsprechenden Namen und die Adresse, erhalten. Die Attribute name, vorname, wohnort sind von der pnr funktional abhängig.

#### Zweite Normalform

Der Primärschlüssel einer Tabelle kann auch aus mehreren Feldern zusammengesetzt sein. So enthält die Tabelle Auftragsposition sowohl die Nummer des Auftrages als auch die Nummer des Artikels. Die Kombination dieser beiden Nummern kann hier als Primärschlüssel vereinbart werden. Weitere Angaben zum Artikel selbst sind nicht mehr nötig, da der Fremdschlüssel artikelnr in der Tabelle Artikel als Primärschlüssel einen eindeutigen Artikeldatensatz kennzeichnet.

**Tabelle: Auftragsposition**

Feldname	Felddate	Zusammengesetzter Primärschlüssel : auftragsnummer und artikelnr
auftragsnummer	Text	
artikelnr	Text	
auftragsmenge	Integer	
auftragspreis	Währung	

Tabelle Auftragsposition mit zusammengesetztem Primärschlüssel, erkennbar am Schlüsselsymbol.

#### Hinweis:

Eine Tabelle befindet sich in der zweiten Normalform, wenn

- sie in der ersten Normalform ist und
- jedes Attribut nicht nur von einem Teil des Primärschlüssels, sondern vom gesamten Primärschlüssel identifiziert wird.

Diese Abhängigkeit vom gesamten Schlüssel bezeichnet man als voll funktionale Abhängigkeit.

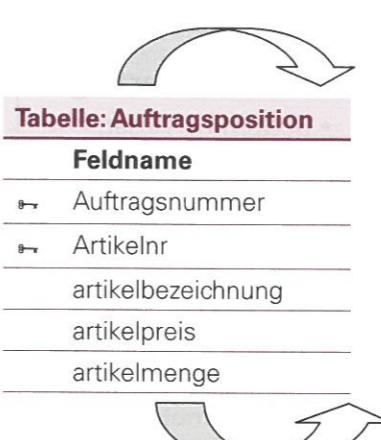
Tabellen mit zusammengesetztem Primärschlüssel werden beim Entwickeln der 2. Normalform betrachtet.

#### Hinweis:

Die 2. Normalform ist bei Tabellen mit zusammengesetzten Primärschlüsseln wichtig.

Die Tabelle Auftragsposition soll in die zweite Normalform gebracht werden. Hierzu lagert man die Felder, die nicht vom gesamten Primärschlüssel abhängen, in eine weitere Tabelle aus.

Am Beispiel der Tabelle Auftragsposition hängt nur das Attribut Menge von beiden Schlüsselfeldern ab. Alle Daten zum Artikel selbst werden schon von einem Teil des Schlüssels, nämlich der artikelnr, festgelegt. Sie werden in die Tabelle Artikel ausgelagert.



Aufteilung der Tabelle Auftragsposition in 2 Tabellen z. B. Position und Artikel. Beide neuen Tabellen sind in der 2. NF.

2. Normalform am Beispiel Auftragsposition.

#### Dritte Normalform

Befindet sich eine Tabelle in der zweiten NF, dann erfolgt die Überprüfung auf die 3. Normalform.

#### Hinweis:

Eine Tabelle befindet sich in der dritten Normalform, wenn sie in der zweiten Normalform ist und wenn zwischen den Feldern, die nicht den Primärschlüssel bilden, keine Abhängigkeiten bestehen.

Man nennt dies auch transitive Abhängigkeit, kein Nichtschlüssel-Attribut ist von einem anderen Nichtschlüssel-Attribut funktional abhängig.

Diese Forderung bezieht sich auf Tabellen mit einfacherem Primärschlüssel. Datenfelder, die von anderen Nichtschlüsselfeldern abhängen, werden in eine eigene Tabelle ausgegliedert.

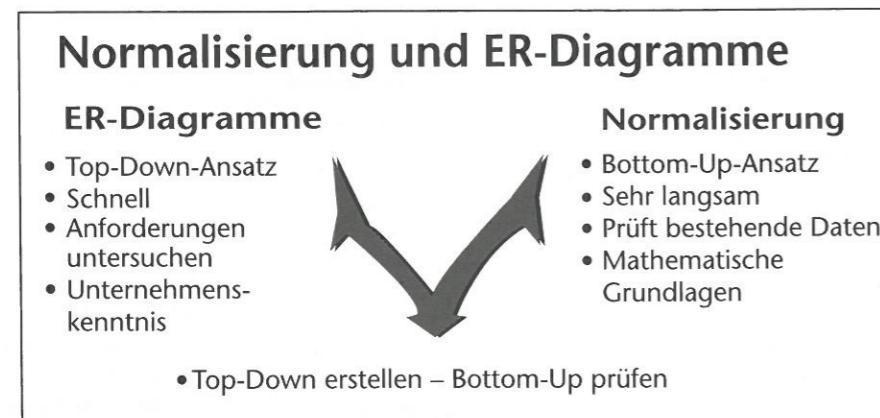
Dies ist bei der Tabelle Personal durchführbar. Dort scheint die Postleitzahl abhängig vom Ort zu sein. Für kleinere Orte ist dies richtig, bei größeren Orten wie Hamburg oder Stuttgart gilt dies jedoch nicht. Dort hängt die Postleitzahl auch von der Straße oder dem Postfach des Kunden ab. Es gilt jedoch eine umgekehrte Abhängigkeit: Der Ort ist abhängig von der Postleitzahl, jeder Postleitzahl ist genau ein Ortsname zugeordnet. Sind mehrere Orte unter einem Ortsverband zusammengefasst, ist dies der Name der Verbandsgemeinde. Somit kann das Feld ort aus dieser Tabelle ausgegliedert und in einer eigenen Tabelle Ort gespeichert werden.

**Tabelle: Ort**

Feldname	Felddatentyp
plz	Text
ort	Text

Tabelle Ort in der dritten Normalform. Jeder plz ist eindeutig ein ort zugeordnet.

Werden alle Tabellen nach den Regeln der Normalisierung erstellt und bearbeitet, sind Redundanzen sicher vermeidbar. Die Normalisierung stellt sicher, dass die Datenbank logisch richtig ist.



Normalformen NF	
Bedeutung	Anmerkung
1. NF	alle Werte atomar (nicht mehr teilbar)
2. NF	jedes Nichtschlüsselmerkmal ist abhängig vom gesamten Primärschlüssel
3. NF	jedes Nichtschlüsselattribut ist direkt abhängig vom Primärschlüssel

Die Aufgliederung der einzelnen Daten in viele Tabellen hat den Nachteil, dass Suchvorgänge durch das DBMS, die sich über mehrere Tabellen erstrecken, viel Zeit benötigen. Deshalb kann es notwendig sein, bestimmte Ausgliederungen, z. B. Postleitzahlen, wieder zurückzunehmen, um die Leistungsfähigkeit (= Performance) des Datenbanksystems zu erhöhen. Diesen Vorgang bezeichnet man als **Denormalisierung**.

### 3.2.2 Beispiel zur Normalisierung: Versandhandel

Ein Versandhandelsunternehmen verwaltet die Rechnungsdaten mit einer Tabellenkalkulations-Software, z. B. Excel. Das so erstellte Rechnungsformular enthält alle wichtigen Angaben. Diese Software wird durch eine Datenbank ersetzt. Es wird hier die Normalisierung bis zur 3. Normalform angewendet.

## FOODVERSAND

Postfach 1234, 88471 Laupheim, Tel +49 739233333, Fax +49 739233334

Ralf Bär An der Heide 7 83262 X-Stadt	<b>Rechnung</b> Kunden-Nr 110	Datum: 20.12.2015 Rechnungs-Nr 342
		Menge Packung Artikel-Nr Bezeichnung Lager Preis Gesamt-preis 2 0,75 ltr G2 Rotwein 3 3,99 7,98 € 3 6 x 0,5 ltr G1 Bier 2 6,59 19,77 €

Rechnungsformular

### Unnormalisierte Tabelle

Zuerst werden alle Daten aus dem Rechnungsformular in eine unnormalisierte Tabelle übernommen.

Rechnungs-		Kunden		Artikel				Menge				
Nr	Datum	Nr	Name	Straße	PLZ	Ort	Nr	Packung	Bezeichnung	La-ger	Preis	MWSt
342	20.12.2015	110	Bär, Ralf	An der Heide 7	83262	X-Stadt	G2, G1	0,75 ltr, 6 x 0,5 ltr	Rotwein, Bier	3, 2	3,99 6,59	0,19 0,19
...	...	...	...	...	...	...	...	...	...	...	...	...

### Erste Normalform

Um nun die unnormalisierte Tabelle in die 1. Normalform zu überführen, werden die Spalten mit nicht-atomaren Daten in eine neue Tabelle Positionsdaten ausgelagert. Das Ergebnis sind die Tabellen Bestelldaten und Positionsdaten in der 1. NF.

Rechnungs-Nr	Rechnungs-Datum	Kunden-Nr	Kunden-Name	Kunden-Straße	Kunden-PLZ	Kunden-Ort
342	20.12.2015	110	Bär, Ralf	An der Heide 7	83262	X-Stadt
...	...	...	...	...	...	...

Tabelle Bestelldaten und Positionsdaten in der 1. NF

Rechnungs-Nr	Artikel-Nr	Artikel-Packung	Artikel-Bezeichnung	Artikel-Lager	Artikel-Preis	Artikel-MwSt	Menge
342	G2	0,75 l	Rotwein	3	3,99	0,19	2
342	G2	6 x 0,5 l	Bier	2	6,59	0,19	3
...	...	...	...	...	...	...	...

In der Tabelle Bestelldaten bildet die Spalte rechnungsnr den Primärschlüssel. In der Tabelle Positionsdaten ist der Primärschlüssel in der Spalte rechnungsnr und artikelnr gemeinsam enthalten, dies wird zusammengesetzter Primärschlüssel genannt.

### Zweite Normalform

In der Tabelle Positionsdaten hängen die Artikelattribute von einem Teil des zusammengesetzten Schlüssels ab, nämlich von artikelnr. Somit befindet sich diese Tabelle nicht in der zweiten Normalform. Zur Normalisierung in die 2. NF werden die artikelnr sowie alle davon abhängigen Attribute zu einer eigenen Tabelle mit dem Schlüssel artikelnr zusammengefasst. Es werden so die Tabellen Artikel und Position gebildet.

Artikel-Nr	Artikel-Packung	Artikel-Bezeichnung	Artikel-Lager	Artikel-Preis	Artikel-MwSt
G2	0,75 l	Rotwein	3	3,99	0,19
G3	6 x 0,5 l	Bier	72	6,59	0,19
...	...	...	...	...	...

Tabellen Artikel und Position in der 2. NF

Rechnungs-Nr	Artikel-Nr	Menge
342	G2	2
342	G3	3
...	...	...

**Hinweis:**

Die 2. NF ist nur für Tabellen mit zusammengesetztem Primärschlüssel wichtig.

**Dritte Normalform**

Die Tabelle Bestelldaten ist nicht in der 3. Normalform, da eine Reihe von Attributen von der kunden\_nr abhängen. Eine Zusammenfassung der von kunden\_nr abhängigen Attribute mit kunden\_nr als Schlüssel in eine Tabelle Kunden löst dieses Problem. In der 3. NF bleiben in der Tabelle Bestelldaten die Attribute rechnungsnr, rechnungsdatum und kunden\_nr übrig.

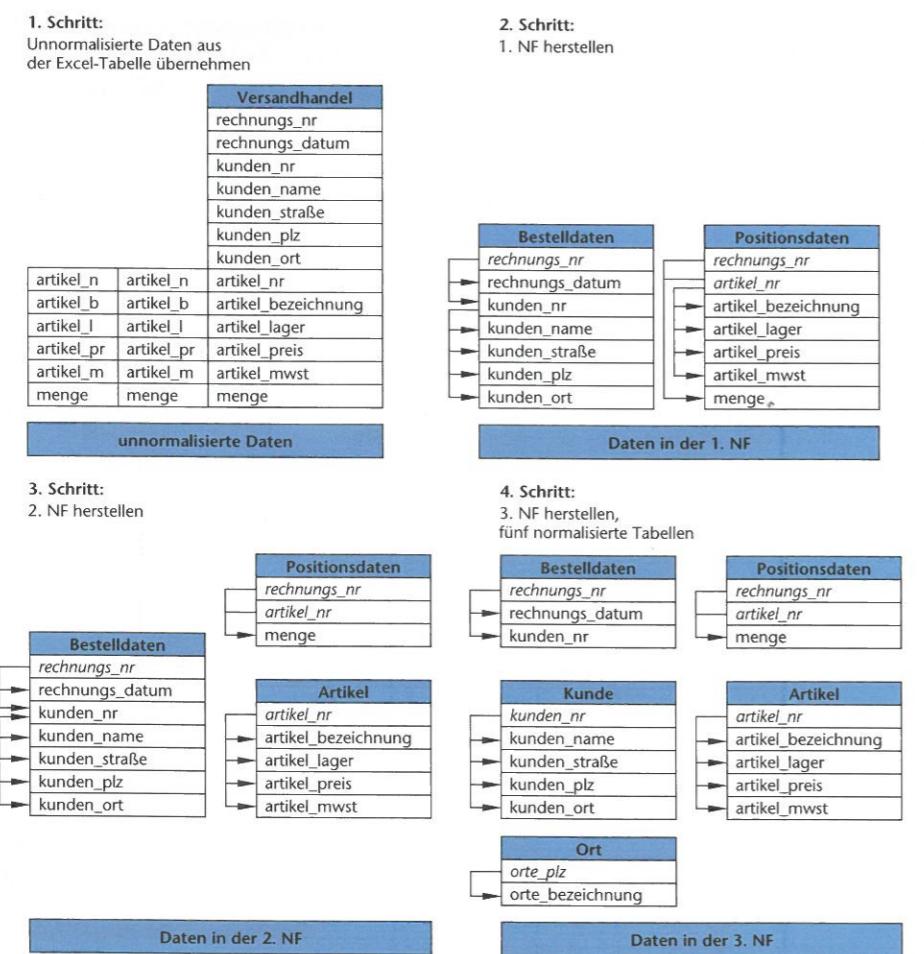
Kunden-Nr	Kunden-Name	Kunden-Straße	Kunden-PLZ	Kunden-Ort
110	Bär, Ralf	An der Heide 7	83262	X-Stadt
...	...	...	...	...

Tabelle Kunden in der 3. NF, alle Nichtschlüssel-Attribute hängen vom Schlüsselattribut kunden\_nr ab.

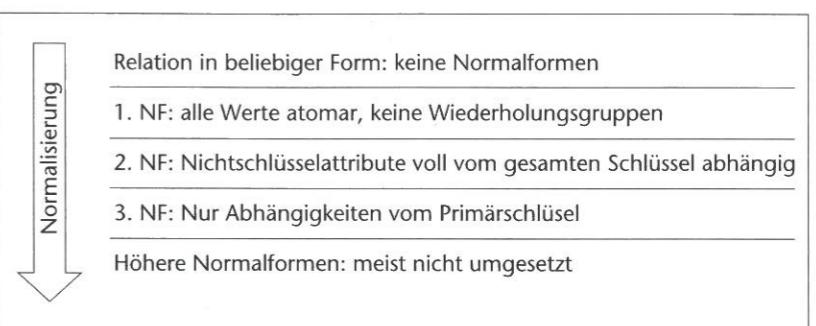
Als Ergebnis des Normalisierungsprozesses ergeben sich im Beispiel Versandhandel 5 Tabellen, die sich alle in der 3. NF befinden. Die Nicht-Schlüsselattribute sind in jeder Tabelle voll funktional abhängig vom gesamten Primärschlüssel. Die jeweiligen funktionalen Abhängigkeiten sind durch Pfeile dargestellt. Die Attribute sind kleingeschrieben. Es gibt auch noch höhere Normalformen. Diese finden praktisch aber kaum Anwendung.

**Hinweis:**

Bei den Tabellen in der 3. NF sind alle Abhängigkeiten auf Abhängigkeiten von Schlüsseln bzw. auf Verknüpfungen über Fremdschlüssel reduziert.



Schematischer Normalisierungsprozess



### 3.2.3 Weitere Normalformen

Die vierte und fünfte Normalform sind wohl deswegen nicht so populär, weil die Situationen unwahrscheinlicher sind, in denen eine Verletzung der Normalformenbedingungen zu erwarten ist. Dies vor allen Dingen deshalb, weil man bereits intuitiv solche Verletzungen vermeidet. Insbesondere wenn man den Weg wählt, zunächst ein ER-Modell aufzustellen, das dann in einen relationalen Entwurf umgesetzt wird, reicht die Prüfung auf dritte Normalform. Verletzungen der zweiten oder dritten Normalform sind zwar nicht zu erwarten, aber immerhin denkbar und sollten durch Nachprüfen ausgeschlossen werden. In allen praktisch relevanten Fällen kann davon ausgegangen werden, dass dann auch die Bedingungen der vierten und fünften Normalform nicht verletzt werden.

### 3.2.4 Integritätsbedingungen

Die Forderung nach **Integrität** einer Datenbank zielt darauf, dass das gespeicherte Datenbild tatsächlich auch der Wirklichkeit entsprechen muss. Fehlerhafte Eingaben, unterlassene Pflege (Updates) stellen schnell den Nutzen einer Datenbank infrage.

- Datentypen und Wertebereiche sind auf der Ebene einzelner Attribute ein wichtiges Instrument, um Datenfehler zu entdecken und auszuschließen.
- Eine weitere Klasse möglicher Fehler wird durch strukturelle Eigenschaften der Datenbank in Verbindung mit der Überprüfung von Schlüsseleigenschaft und von Fremdschlüsseln (Referentielle Integrität) vermieden.

**Referentielle Integrität**

Referentielle Integrität ist ein Regelsystem, mit dessen Hilfe ein Datenbankmanagementsystem, z. B. MS Access, sicherstellt, dass Beziehungen zwischen Datensätzen in Detailtabellen gültig sind und dass verknüpfte Daten nicht versehentlich gelöscht oder geändert werden. Referentielle Integrität kann festgelegt werden, wenn alle der folgenden Bedingungen erfüllt sind:

- Das übereinstimmende Feld aus der Mastertabelle ist ein Primärschlüssel oder hat einen eindeutigen Index.
- Beide Tabellen gehören zu derselben Datenbank. Um referentielle Integrität festzulegen, wird die Datenbank, in der die Tabellen gespeichert sind, geöffnet. Referentielle Integrität kann nicht für verknüpfte Tabellen aus Datenbanken anderer Formate durchgesetzt werden.

Beim Durchsetzen referentieller Integrität gelten die folgenden Regeln:

- Man kann in das Fremdschlüsselfeld der Detailtabelle keinen Wert eingeben, der nicht im Primärschlüsselfeld der Mastertabelle enthalten ist. Man kann jedoch in das Fremdschlüsselfeld einen Nullwert eingeben und damit angeben, dass die Datensätze nicht miteinander verknüpft sind. So ist es z. B. nicht möglich, eine Bestellung zu haben, die einem nichtvorhandenen Kunden zugeordnet ist. Man kann jedoch eine Bestellung haben, die niemandem zugeordnet ist, indem das Feld Kunden-Nr einen Nullwert enthält.
- Ein Datensatz aus der Mastertabelle kann nicht gelöscht werden, wenn übereinstimmende Datensätze in einer Detailtabelle enthalten sind. Beispielsweise kann man einen Datensatz eines Mitarbeiters aus der Tabelle Personal nicht löschen, wenn diesem Mitarbeiter in der Tabelle Bestelldaten ein Datensatz zugeordnet ist.
- Ein Primärschlüsselwert in der Mastertabelle kann nicht geändert werden, wenn es zu diesem Datensatz Detaildatensätze gibt. Beispielsweise kann man die Personalnummer eines Mitarbeiters in der Tabelle Personal nicht ändern, wenn diesem Mitarbeiter in der Tabelle Bestellungen Datensätze zugeordnet sind.

### 3.3 Aufgaben zu Kapitel 3

1. Betrachten Sie die Tabelle 1:

- a) Welche Normalformen sind verletzt?
- b) Erzeugen Sie ein gleichwertiges System in normalisierter Form.

Tabelle 1

ISBN-Nr	Autoren	Titel	Jahr	Seiten
0-201-14192-2	Date, Ch.	The Relational Model for Database Management: Version 2	1990	538
3-89319-117-8	Finkenzeller, H. Kracke, U. Unterstein, M	Systematischer Einsatz von SQL-Oracle	1989	494
1-55860-245-3	Melton, J. Simon, A.	Understanding the new SQL	1993	536

2. Betrachten Sie die Tabelle 2:

- a) Welche Normalformen sind hier verletzt?
- b) Erzeugen Sie ebenfalls ein gleichwertiges System in normalisierter Form.

Tabelle 2

Matrikel	Student	Kurs-Nr	Kurs-Titel	Note
30321	Meyer, J.	706S6	Datenbanksysteme	1,0
30321	Meyer, J.	715S4	Software-Engineering	1,7
30346	Ahrens, H.	715S4	Software-Engineering	3,0
30346	Ahrens, H.	706S6	Datenbanksysteme	2,0
30346	Ahrens, H.	713S5	relationale u. unfunktionale Programmierung	1,7
30378	Knudsen, K.	706S6	Datenbanksysteme	2,0

### 3. Inventurdaten Hardware

Die folgende Darstellung zeigt Hardware Inventardaten (z. B. einer Schule). Die Daten sollen in einem relationalen DBS in normalisierter Form gespeichert werden. (Die Portangaben haben folgende Bedeutung: M1 = IDE Port 1 Master, S1 IDE Port1 Slave, ...)

Rechner		Verantwortlicher		HD-Laufwerke			IDE Port	
Invent Nr	Standort	Pers Nr	Name	Telefon	Hersteller	Produkt Nr	Kapazität	
L1001	B246	F100	M. Mayer	8975	Western Digital	102BA	1TB	M1
					Western Digital	102BA	1TB	S1
					Fujitsu	MPF3204 AT	2TB	M2
					Fujitsu	MPF3204 AT	2TB	S2
					IBM	DTLA-305020	2TB	M1
L1003	B251	F101	F. Binder	5635	IBM	DTLA-307045	4TB	M2
					Western Digital	102BA	1TB	S1

Für die dargestellte Datensammlung sollen die normalisierten Tabellen für eine relationale Datenbank entworfen werden. Gehen sie dabei schrittweise vor:

**1. Schritt:** Stellen Sie die 1. NF her, indem Sie die Datensammlung in 2 Relationen aufteilen. Geben Sie als Ergebnis die Relationen mit ihren Attributen an und kennzeichnen Sie den Primärschlüssel und die Fremdschlüssele.

**2. Schritt:** Stellen Sie, sofern notwendig, für die Relationen aus Schritt 1 die 2. NF her. Geben Sie als Ergebnis wieder die Relationen mit ihren Attributen an und kennzeichnen Sie den Primärschlüssel und die Fremdschlüssele.

**3. Schritt:** Stellen Sie, sofern notwendig, für die Relationen aus Schritt 2 die 3. NF her und geben Sie als Gesamtergebnis alle Relationen mit ihren Attributen an, die jetzt der Ausgangsdatensammlung in normalisierter Form entsprechen.

### Fragen:

Welche Redundanzen konnten durch die Normalisierung beseitigt werden?

Für die Tabellen (nach Schritt 1), die zwar der 1. NF, nicht aber der 2. NF entsprechen, ergeben sich beim Einfügen, Ändern und Löschen Probleme. Geben Sie anhand dieser Tabellen für jede Form ein Beispiel an.

### 4. Gepäckliste einer Fluggesellschaft

Gegeben ist die Gepäckliste einer Fluggesellschaft. Führen Sie eine schrittweise Normalisierung bis zur 3. NF durch.

Flugnummer:	WA876	Flugzeugtyp:	B 747
von:	München	Flugkapitän:	Ebeling
nach:	Honolulu		
am:	1.4.2015		
<b>Passagier</b>			
Personalausweisnummer	Name	Vorname	Straße
89021238	Brezenhuber	Ralph	Bäckerstr. 9
			81669 München
			1 Koffer 20,5
			2 Surfboard 6,5
			3 Rucksack 7
74921209	Hippthalter	Hans	Bräustr. 6
			85049 Ingolstadt
			4 Mountainbike 35
96554367	Maja	Matthias	Brainstr. 12
			85049 Ingolstadt
			5 Koffer 12
			6 Angelrute 1,2

### Unnormalisierte Datensammlung

FNr	von	nach	am	FT	FK	PNr	N	VN	Str	PLZ	Ort	GNr	A	G
WA876	MUC	HON	1.4.15	B747	Ebeling	89...	Brezenh.	Ralph	Bäckerstr. 9	81669	München	1	Koffer	20,5
												2	Surfboard	6,5
												3	Rucksack	7
												4	Mountain-bike	35
												5	Koffer	12
												6	Angelrute	1,2
74...														
96...														

**5. Gegeben ist eine unnormalisierte Tabelle.** Führen Sie eine schrittweise Normalisierung bis zur 3. NF durch.

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse
3	Wilschrein	Theo	1	Personal	1,2,3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse
4	Richter	Hans	3	Verkauf	2	Verkaufspromotion
5	Wiesenland	Werner	2	Einkauf	1	Kundenumfrage

**6. Gegeben ist folgende unnormalisierte Relation, die Daten über Mitarbeiter und deren Abteilungszugehörigkeit enthält. Weiterhin sind die Beteiligung(en) der einzelnen Mitarbeiter an unternehmensweiten Projekten vermerkt sowie welche Abteilung die Verantwortung für die Projekte übernommen hat.**

- a) Führen Sie eine schrittweise Normalisierung bis zur 3. Normalform durch.

Hinweis: Beschränken Sie sich bei der Darstellung der abgeleiteten Relationen auf die jeweiligen Attributnamen.

- b) Fügen Sie sinnvolle Primärschlüssel ein.  
c) Erläutern Sie anhand der Regeln zur Bildung der Normalformen Ihre Teilschritte

NACHNAME	VORNAME	GEBURTSTAG	EINSTELLUNGS-DATUM	ABTEILUNGS-NAME	ABTEILUNGSLEITER	PROJEKT-NAME	PROJEKT-START	PROJEKTVERANTWORTLICHE ABTEILUNG
Soll	Siggi	04.05.65	01.06.90	Controlling	Plan	Konzernweite BSC Aufbau MIS	01.00 04.01	Controlling Informatik
Return	Rudi	21.12.60	01.06.87	Informatik	Maus	Aufbau MIS	04.01	Informatik
Plan	Peter	09.03.61	01.03.84	Controlling	Plan	Vorber. Börsengang	12.99	Controlling
Exakt	Erika	22.07.69	01.09.98	Controlling	Plan	Konzernweite BSC Vorber. Börsengang	01.00 12.99	Controlling Controlling
Korrekt	Kurt	30.01.55	01.01.72	Finanzen	Korrekt	Vorber. Börsengang	12.99	Controlling
Maus	Michael	09.10.50	01.01.90	Informatik	Maus	Vorber. Börsengang	12.99	Controlling
Excel	Emil	16.04.68	01.09.97	Controlling	Plan	Vorber. Börsengang Aufbau MIS Konzernweite BSC	12.99 04.01 01.00	Controlling Informatik Controlling
Giro	Gerd	03.09.73	01.09.99	Finanzen	Korrekt	Vorber. Börsengang	12.99	Controlling

**7. Gegeben ist die unnormalisierte Relation Lernangebot, die Daten über Schüler, Klassen, Klassenlehrer und Lernzeiten enthält.**

Führen Sie eine schrittweise Normalisierung bis zur 3. Normalform durch.

Relation: Lernangebot

SchülerNr	Name	Vorname	Klasse	Klassenlehrer	LernangebotsNr	Beschreibung	Zeit in h
1	Jürgens	Ina	11a	Lempel	2	Tanz	12
2	Schmidt	Tom	12a	Breier	3	Chor	22
3	Jäger	Franz	11a	Lempel	1, 2, 3	Elektronik, Tanz, Chor	15, 12, 2
4	Olsen	Ina	11b	Sommer	2	Tanz	5
5	Jürgens	Paula	12a	Breier	1	Elektronik	23

### 3.4 Digitale Inhalte zu Kapitel 3

Hinweis: Um die Aufgaben online zu bearbeiten, bitte den QR-Code scannen oder den Link eingeben.

#### Aufgabe 1

<https://vel.plus/VPl2>



#### Aufgabe 2

<https://vel.plus/G5QG>



#### Aufgabe 3

Kahoot-App Suchbegriff 36087 oder Kahoot.it

Questions (9)

- 1 - Quiz Welche Stufen der Normalform NF gibt es?
- 2 - Quiz Durch die Normalisierung ...
- 3 - Quiz Primary Keys oder Primärschlüssel....
- 4 - Quiz Worauf verweist ein Fremdschlüssel (FS)?
- 5 - Quiz Was ist das Kennzeichen der 1. Normalform?
- 6 - Quiz Kennzeichne die 3. Normalform (3NF).

Datenbanken 36087 Kapitel 3 Normalisierung Normalformen