



**it**  
informatik

Heide Faeskorn-Woyke  
Birgit Bertelsmeier  
Petra Riemer  
Elena Bauer

# Datenbanksysteme

Theorie und Praxis mit SQL2003  
Oracle und MySQL

Heide Faeskorn-Woyke  
Birgit Bertelsmeier  
Petra Riemer  
Elena Bauer

# Datenbanksysteme

Theorie und Praxis mit  
SQL2003, Oracle und MySQL

## eBook

Die nicht autorisierte Weitergabe dieses eBooks  
an Dritte ist eine Verletzung des Urheberrechts!

PEARSON  
Studium

---

ein Imprint von Pearson Education  
München • Boston • San Francisco • Harlow, England  
Don Mills, Ontario • Sydney • Mexico City  
Madrid • Amsterdam

(Fortsetzung)

<u>TNr</u>	<u>Bezeichnung</u>	<u>Artikel_Typ</u>	<u>VP</u>	<u>Jahresumsatz</u>	<u>LiefNr</u>	<u>Zeitstempel</u>
56	HERCULES NEPA	Trekkingrad	1700	80	1	06.10.06
57	Steppenwolf TAO	Mountainbike	1900	80	2	06.10.06
58	SWITCHBACK AGENT	Jugendrad	899	80	2, 3	06.10.06
59	STEVENS R.P.R.2 RX100 8FACH	Rennmaschine	1800	80	1,3	06.10.06
60	Scott ATACAMA TOUR	Crossrad	2399	80	1	06.10.06
61	ROTWILD RCC-03	Mountainbike	3499	80	2	06.10.06

In unserem Beispiel ist VP die Abkürzung für Verkaufspreis. Diese Relation ist nicht in der ersten Normalform, da Wiederholungsgruppen in der Lief\_Nr. auftreten. Bei einer Überführung in die erste Normalform werden alle nichteinfachen Attribute in neue Tupel aufgelöst und es entstehen teilweise redundante Tupel, wie die folgende Tabelle zeigt:

<u>TNr</u>	<u>Bezeichnung</u>	<u>Artikel_Typ</u>	<u>VP</u>	<u>Jahresumsatz</u>	<u>Lief_Nr</u>	<u>Zeitstempel</u>
1	Rocky Mountain Element Rac	Mountainbike	3500	200	1	06.10.06
1	Rocky Mountain Element Rac	Mountainbike	3500	200	2	06.10.06
31	Herrenrad GT-LTS 18	Rennrad	3500	400	1	06.10.06
31	Herrenrad GT-LTS 18	Rennrad	3500	400	2	06.10.06
31	Herrenrad GT-LTS 18	Rennrad	3500	400	3	06.10.06
54	Klapprad Prompton P3	Klapprad	1600	180	2	06.10.06
55	CANNONDALE FSL	Mountainbike	3700	80	3	06.10.06
56	HERCULES NEPA	Trekkingrad	1700	80	1	06.10.06
57	Steppenwolf TAO	Mountainbike	1900	80	2	06.10.06
58	SWITCHBACK AGENT	Jugendrad	899	80	2	06.10.06
58	SWITCHBACK AGENT	Jugendrad	899	80	3	06.10.06
59	STEVENS R.P.R.2 RX100 8FACH	Rennmaschine	1800	80	1	06.10.06
59	STEVENS R.P.R.2 RX100 8FACH	Rennmaschine	1800	80	3	06.10.06
60	Scott ATACAMA TOUR	Crossrad	2399	80	1	06.10.06
61	ROTWILD RCC-03	Mountainbike	3499	80	2	06.10.06

Die Verletzung der ersten Normalform tritt in unserem Datenbankschema der Firma Byce & Co. (im Abschnitt 4.4.3) nicht auf, da die Lief\_Nr in den Tabellen „Lieferprogramm“ bzw. „Lieferungen“ aufgenommen wurde. Das entspricht schon der ersten Normalform.

Bei objektrelationalen Datenmodellen ist die Verletzung der ersten Normalform gewollt und wird daher in Kauf genommen. Dazu wird ein eigener Typkonstruktor, der LIST-Konstruktor, eingeführt (vgl. Kapitel 6).

### 4.3.2 Die zweite Normalform

Bei der zweiten Normalform geht es um die Beseitigung unerwünschter Abhängigkeiten und die Vermeidung von Redundanzen. Dabei spielen funktionale Abhängigkeiten (vgl. Abschnitt 4.2) von Teilbereichen eines zusammengesetzten Schlüssels die zentrale Rolle.

#### Zweite Normalform

Eine Relation  $R$  mit Primärschlüssel  $S$  befindet sich in der zweiten Normalform (2NF), wenn sie (1NF) ist und jedes Nichtschlüsselattribut voll funktional abhängig vom Primärschlüssel  $S$  ist.

Die volle funktionale Abhängigkeit bedeutet, dass keine funktionale Abhängigkeit einer Attributmenge der Nichtschlüsselattribute von einer Teilmenge der Schlüsselattribute besteht. Eine partielle funktionale Abhängigkeit bedeutet dagegen, dass Nicht-Schlüsselattribute von einem Teil des Primärschlüssels funktional abhängig sind. Die 2NF beseitigt daher alle partiellen funktionalen Abhängigkeiten.

Bei der Überführung einer Relation in die 2NF geht man folgendermaßen vor: Wenn der Primärschlüssel nur aus einem Attribut besteht, ist die Relation immer in der zweiten Normalform. Hat der Primärschlüssel mehr als ein Attribut, spaltet man alle Attribute als neue Relationen ab, die nur von einer Teilmenge<sup>15</sup> des Primärschlüssels voll funktional abhängen. Dieses Vorgehen wird in Abbildung 4.11 veranschaulicht, wobei die dünnen schwarzen Pfeile auf partielle abhängige Attribute weisen. Für die beiden partiellen Abhängigkeiten entstehen neue Relationen und in der ursprünglichen Relation verbleibt der vollständige (ursprüngliche) Primärschlüssel mit allen von ihm voll funktional abhängigen Nichtschlüsselattributen.

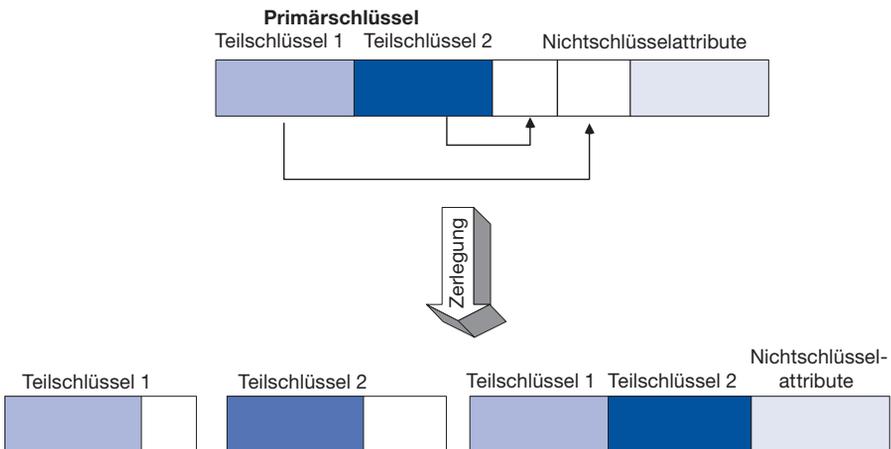


Abbildung 4.11: Normalform

<sup>15</sup> Die Anzahl der Teilmengen einer Menge mit  $n$  Elementen berechnet sich zu  $2^n$ , wobei die leere Menge mitzählt. Eine leere Menge kann nicht aus Schlüsselattributen bestehen. Daher kann eine Zerlegung eines Schlüssels mit zwei Attributen höchstens  $3 = (2^2) - 1$  Relationen ergeben, eine Relation mit drei Schlüsselattributen  $7 = (2^3 - 1)$  Relationen etc. vgl. <http://de.wikipedia.org/wiki/Teilmenge>, 6.10.2006

## Beispiel

Produktion							
Ang_Nr	Name	Ort	Abt_Nr	Abt_Name	TNr	Bezeichnung	Zeit
101	Paul	Bonn	1	Chemie	11	Kali	60
101	Paul	Bonn	1	Chemie	12	Farbe	40
102	Hugo	Köln	2	Kunststoff	13	PVC	20

Die Relation „Produktion“ mit dem Primärschlüssel Ang\_Nr und TNr soll ein stark vereinfachtes Produktionsgeschehen abbilden. Personen, identifiziert über die Personalnummer (Ang\_Nr), gehören zu Abteilungen, wohnen in einem Ort und produzieren Produkte, die durch die TNr als Primärschlüssel identifiziert werden. Die Zeit, die eine bestimmte Person zur Fertigung eines Produkts benötigt, ist in der Spalte „Zeit“ (in Minuten) abgespeichert. Diese Relation ist 1NF, aber nicht 2NF. Redundanzen treten auf, da für eine Person der Ort und der Abteilungsname mehrmals gespeichert werden.

**Normalisierung zur (2NF)** Da in unserem Beispiel der Schlüssel der Relation zusammengesetzt ist, kann es grundsätzlich erst einmal sein, dass partielle Abhängigkeiten auftreten. In der obigen Relation lassen sich folgende partielle funktionale Abhängigkeiten vom Schlüssel (Ang\_Nr, TNr) finden:

- Ang\_Nr → Name, Ort, Abt\_Nr, Abt\_Name
- TNr → Bezeichnung

sowie die volle funktionale Abhängigkeit vom gesamten Schlüssel:

- Ang\_Nr, TNr → Menge.

Bei der Normalisierung zur 2NF werden in unserem Beispiel die Attribute in drei Relationen aufgeteilt. In der ersten Relation (Person) stehen alle Attribute, die nur von einer Person bestimmt sind, also vom Teilschlüssel 1, der „Ang\_Nr“, funktional abhängig sind. In der zweiten Relation „Produkte“ stehen entsprechend alle Attribute, die vom Teilschlüssel 2, also der „TNr“, abhängig sind. In der dritten Relation „Produktion“ verbleiben alle Attribute, die von beiden Schlüsselattributen abhängig sind. Bei dieser Art der Normalisierung ist darauf zu achten, dass alle partiellen funktionalen Abhängigkeiten eliminiert werden, ohne dass Informationen, die in der ursprünglichen Tabelle enthalten sind, verloren gehen.

Dieser Hinweis ist besonders wichtig, wenn es keine Attribute gibt, die voll funktional abhängig sind vom gesamten Schlüssel. Denn dann wird schnell vergessen, für den gesamten Schlüssel noch eigene Relationen anzulegen. Damit gehen dann wichtige Informationen der Ursprungsrelation verloren und ein Informationsverlust bei der Normalisierung ist unbedingt zu vermeiden. Für unser Beispiel hieße das, wenn es die volle funktionale Abhängigkeit des Attributs „Menge“ nicht geben würde, so müsste trotzdem eine dritte Relation bestehend nur aus den Schlüsselattributen (Anr\_Nr, TNr) erstellt werden, um nicht die zentrale Information „Wer hat welches Teil gefertigt“ zu verlieren.

Person				
<u>Ang_Nr</u>	Name	Ort	Abt_Nr	Abt_Name
101	Paul	Bonn	1	Chemie
102	Hugo	Köln	2	Kunststoff

Produktion		
<u>Ang_Nr</u>	<u>TNr</u>	Zeit
101	11	60
101	12	40
102	13	20

Produkt	
<u>TNr</u>	Bezeichnung
11	Kali
12	Farbe

Es ist klar, dass bei einem aus zwei Attributen zusammengesetzten Schlüssel nicht immer eine Aufteilung in drei Relationen erfolgen muss. Dies ist dann der Fall, wenn es nur ein oder gar kein Attribut gibt, welches von einem Teilschlüssel funktional abhängig ist. Bei Schlüsseln, die aus mehr als zwei Attributen bestehen, funktioniert das Verfahren entsprechend. Alle nicht leeren Teilmengen des Schlüssels sind Kandidaten für Primärschlüssel in den aufgeteilten Relationen. Wenn Sie sich die partiellen funktionalen Abhängigkeiten herausschreiben, dann wird offensichtlich, welche neuen Relationen für die 2NF gebraucht werden. Es kann auch sein, dass es gar keine partiellen Abhängigkeiten und damit keine 2NF gibt. In diesem Fall entsteht die Aufteilung der Tabellen erst bei der Überführung in die dritte Normalform oder die Tabelle ist schon in der dritten Normalform.

### 4.3.3 Die dritte Normalform

Die dritte Normalform bezieht auch Nichtschlüsselattribute, die nicht direkt vom Schlüssel abhängen, in ein ähnliches Verfahren mit ein, wie die Überführung von der 1NF zur 2NF. Grundlage ist der Begriff der transitiven Abhängigkeit.

#### Transitive Abhängigkeit

Mit  $X$ ,  $Y$  und  $Z$  seien paarweise verschiedene Attributkombinationen einer Relation  $R = R(A_1, A_2, \dots, A_n)$  bezeichnet.

$Z$  heißt **transitiv abhängig** von  $X$ , wenn  $Y$  voll funktional abhängig von  $X$  und  $Z$  voll funktional abhängig von  $Y$  ist, aber  $X$  nicht voll funktional abhängig von  $Y$  ist.

Also  $X \rightarrow Y \rightarrow Z$ , aber **nicht**  $Y \rightarrow X$ .

### Beispiel

Die Relation „Person“ aus der 2NF

Person				
Ang_Nr	Name	Ort	Abt_Nr	Abt_Name
101	Paul	Bonn	1	Chemie
102	Hugo	Köln	2	Kunststoff

enthält noch die transitive Abhängigkeit des Abteilungsnamens von der Ang\_Nr:  $Ang\_Nr \rightarrow Abt\_Nr \rightarrow Abt\_Name$ . In diesem Fall ist es naheliegend, noch eine zusätzliche Tabelle „Abteilung“ mit einem Schlüssel Abt\_Nr anzulegen.

Die zweite Bedingung in der obigen Definition der transitiven Abhängigkeit kann nicht wegfallen, wenn man funktionale Abhängigkeiten betrachtet, die rekursiv sind. So können z.B. in einer Relation „Person“ mehrere Attribute definiert sein, die eine Person identifizieren:

Person (Schlüssel1, Schlüssel2, Schlüssel3, Nachname, Vorname ...)<sup>16</sup>

Dann gilt:  $Schlüssel1 \rightarrow Schlüssel2 \rightarrow Schlüssel3$  und  $Schlüssel3 \rightarrow Schlüssel2 \rightarrow Schlüssel1$ .

Eine Aufteilung auf unterschiedliche Relationen macht hier keinen Sinn.

## Dritte Normalform

Eine Relation R ist in der dritten Normalform (3NF), wenn sie sich in der ersten und der zweiten Normalform befindet und kein Nichtschlüsselattribut transitiv abhängig von einem Schlüsselattribut ist.

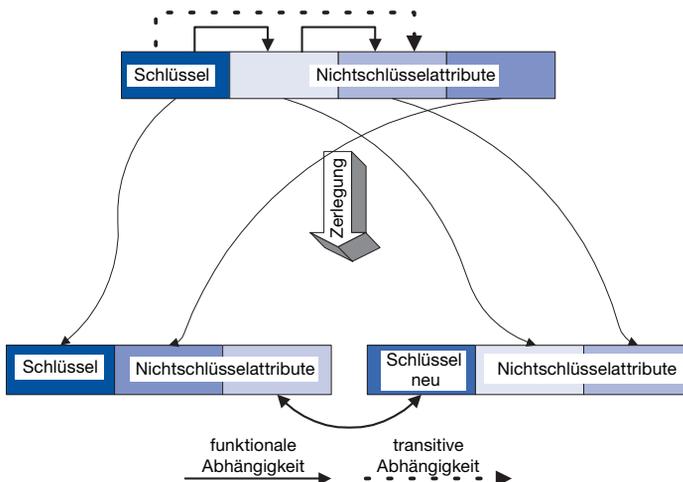


Abbildung 4.12: Dritte Normalform

16 z.B. Schlüssel1 = Personalausweisnummer, Schlüssel2 = Sozialversicherungsnummer, Schlüssel3 = Matrikelnummer

Bei der Überführung in die dritte Normalform wird aus den an der transitiven Abhängigkeit beteiligten Attributen Y und Z (vgl. Definition der transitiven Abhängigkeit) eine neue Relation gebildet.

Das Verfahren wird am nachfolgenden Beispiel erläutert.

### Überführung in die dritte Normalform

Person				Abteilung	
Ang_Nr	Name	Ort	Abt_Nr	Abt_Nr	Abt_Name
101	Paul	Bonn	1	1	Chemie
102	Hugo	Köln	2	2	Kunststoff

Zusätzlich zur bestehenden Aufteilung auf die Relationen „Person“, „Produktion“ und „Produkt“ wird aus der Relation „Person“ noch die Abteilung herausgelöst. Dabei ist auch wieder darauf zu achten, dass dies ohne Informationsverlust geschieht. Eine wichtige Information der 2NF-Relation „Person“ ist „In welcher Abteilung arbeitet der Mitarbeiter“, die erhalten bleibt, wenn der Primärschlüssel der neuen Relation „Abteilung“ als Fremdschlüssel in der Relation „Person“ auftaucht. Kennzeichen der dritten Normalform ist, dass ihr Informationsgehalt der gleiche ist wie bei der 1NF und dass alle Redundanzen – außer den Schlüsselredundanzen bei Fremdschlüsselbeziehungen – verschwunden sind. Diese Schlüsselredundanzen müssen in Kauf genommen werden, weil sonst ein Informationsverlust entstehen würde.

Der Prozess der Normalisierung und Zerlegung einer Relation in die 1NF, 2NF und 3NF muss die Wiederherstellbarkeit und die Abhängigkeitswahrung der ursprünglichen Relation erhalten.

## Wiederherstellbarkeit und Abhängigkeitswahrung

**Wiederherstellbarkeit:** Eine Zerlegung einer Relation ist **verlustfrei**, wenn sich alle Tupel der ursprünglichen Tabelle durch einen Join aus den abgeleiteten Relationen wiederherstellen lassen. Eine verlustfreie Zerlegung stellt damit die Wiederherstellbarkeit der ursprünglichen Relation sicher.

**Abhängigkeitswahrung:** Die Zerlegung einer Relation ist **abhängigkeitstreu**, wenn jede funktionale Abhängigkeit der Ausgangstabelle in einer der resultierenden Tabellen erhalten bleibt. Eine abhängigkeitstreue Zerlegung stellt damit die Wiederherstellbarkeit der ursprünglichen funktionalen Abhängigkeiten sicher.

Kemper beschreibt in [Kemper et al. 2004, S. 180], wie der Synthesearchivus aus einer gegebenen Menge von Relationen eine verlustfreie, abhängigkeitstreu Zerlegung in Relationen der dritten Normalform liefert. Dieser Algorithmus entspricht unserer intuitiven Vorgehensweise, die wir in diesem Kapitel dargelegt haben, um eine 3NF zu erzeugen.

Die Verletzung der dritten Normalform hat die gleichen Nachteile, wie wir sie schon bei der zweiten Normalform gefunden haben. In der Praxis ergeben sich jedoch auch einige Nachteile bei der vollständigen Normalisierung in die dritte Normalform.

### Nutzen und Grenzen des Normalisierungsverfahrens

- 1** Die Normalisierung geht von einem Datenvolumen aus, bei dem die in den Relationen enthaltenen Daten schon vorhanden sind und sich nicht verändern. Auf diesen Relationen werden aufgrund funktionaler Abhängigkeiten Maßnahmen zur besseren Strukturierung durchgeführt. In der Praxis sind natürlich die Relationen beim Datenbankentwurf noch nicht in allen Fällen mit Daten gefüllt; Bewegungsdaten verändern sich immer im laufenden Betrieb. Der Entwickler muss also einen gewissen Spürsinn haben, wo in Zukunft unerwünschte funktionale Abhängigkeiten und Redundanzen auftreten könnten. Außerdem wird beim konzeptionellen Modell als ER-Modell oft schon intuitiv die dritte Normalform gewählt.
- 2** Anders sieht das Ganze bei der Reorganisation alter Datenbestände aus. In diesen Fällen ist die Normalformenlehre ein nützliches Instrument, um Redundanzen zu beseitigen und eine gute Datenstruktur zu erreichen.
- 3** Durch die Normalisierung ergibt sich bei größeren Projekten eine Vielzahl von mit Fremdschlüsselbeziehungen verbundenen Tabellen. Informationen, die inhaltlich im Sinne der Objektorientierung zusammengehören, werden unter Umständen auf viele Tabellen verteilt. Bei Abfragen kann es zu erheblichen Performanceverlusten kommen, da oft viele Relationen aufwändig miteinander verknüpft werden müssen.
- 4** Aus den angeführten Gründen ist eine Verletzung der dritten Normalform (Denormalisierung) in der Praxis manchmal gewollt. Die Normalisierung ist nicht als Dogma zu verstehen, aber nach wie vor ein analytisches Instrument, um Relationenstrukturen auf ihre Qualität hin zu überprüfen. Im Abschnitt 7.3 werden wir mit den Datenbanktriggern dann auch ein Konzept vorstellen, mit dem das Problem der Redundanzen handhabbar wird.

## 4.4 Von der Analyse zum Entwurf im relationalen Datenmodell

Nach dem Entwurf des Lastenhefts und des konzeptionellen Modells als ER-Modell besteht der nächste Schritt unserer Datenbankentwurfsmethode aus der Transformation eines konzeptionellen Modells auf ein relationales Datenbankschema. Ein Datenbankschema ist die konkrete Ausprägung eines Datenmodells mit Metadaten, die einen Ausschnitt aus der Wirklichkeit beschreiben.<sup>17</sup> In unserem Fall ist das Datenmodell relational und entspricht der relationalen Algebra.

---

17 vgl. Kapitel 1

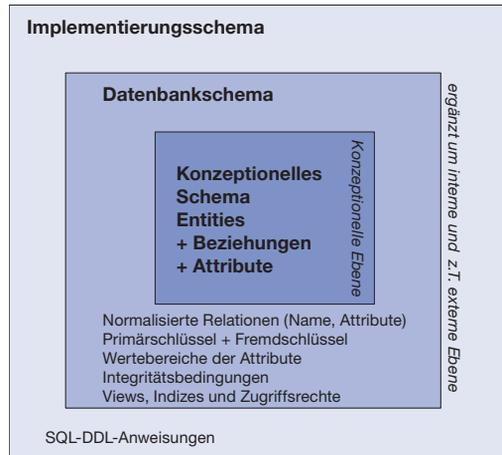


Abbildung 4.13: Datenbankentwurf

In diesem Kapitel werden wir uns zunächst damit beschäftigen, wie man ein konzeptionelles ER-Modell oder ein EERM auf ein Datenbankschema abbildet. Anschließend wird noch eine Überführung des Datenbankschemas in die dritte Normalform durchgeführt und das entstandene Datenbankschema verfeinert. Dazu gehören:

- 1** Primärschlüssel + Fremdschlüssel: Jede Relation muss einen Primärschlüssel haben. Fremdschlüssel werden verwendet, um die referentielle Integrität zu sichern, und müssen daher nicht in jeder Relation existieren.
- 2** Wertebereiche der Attribute: Neben den Basisdatentypen sind endliche Wertebereiche und Domänen (z.B. weiblich, männlich) festzulegen.
- 3** Semantische Integritätsbedingungen (vgl. Abschnitt 4.1) werden in Textform beschrieben.
- 4** Views stellen Daten aus einer oder auch verschiedenen Relationen für bestimmte Benutzersichten, die für spezielle Funktionen verwendet werden, zur Verfügung.
- 5** Indizes beschreiben Zweitschlüssel.
- 6** Zugriffsrechte werden auf den Relationen oder Sichten für einzelne Benutzer oder Gruppen von Benutzern erklärt.

Die Vorgehensweise entspricht Schritt 3 aus unserer Methode der Entwicklung einer Datenbankanwendung aus Abschnitt 2.2. Der nächste und letzte Schritt ist die Implementierungsphase, die wir in Kapitel 5 genauer betrachten werden.

### 4.4.1 Transformation des ER-Modells auf ein relationales Datenmodell

Bei der Transformation, die auch maschinell erfolgen kann, gibt es zwei Hauptregeln<sup>18</sup>:

- Regel 1: Die Entity-Mengen des ER-Modells werden auf Relationen abgebildet.
- Regel 2: Beziehungen werden auf Fremdschlüsselattribute oder Relationen abgebildet.

Die folgende Tabelle fasst die Vorgehensweise bei den Beziehungen zusammen.

Tabelle 4.3		
Vorgehensweise bei den Beziehungen		
Verfahren	Beziehungsart	Vorgehensweise
Typ 1	1:1-Beziehung	Beide Entity-Mengen werden entweder in zwei Tabellen mit der verbindenden Beziehung überführt oder zu einer Tabelle zusammengefasst.
Typ 2	1:n-Beziehung 1:cn-Beziehung 1:c-Beziehung c:c-Beziehung	Für beide Entity-Mengen wird je eine Tabelle erzeugt. Der Primärschlüssel der Master-Tabelle wird Fremdschlüssel der cn-Detail-Tabelle und dort als Attribut neu erzeugt, falls es noch nicht existiert.
Typ 3	n:m-Beziehung cn:m-Beziehung cn:cm-Beziehung ternäre Beziehung	Für beide (bzw. drei bei einer ternären Beziehung) Entity-Mengen und die Beziehung werden je eine Tabelle mit entsprechenden verbindenden Beziehungen erzeugt.

Entity-Mengen werden generell auf Relationen abgebildet. Bei den Beziehungen gibt es unterschiedliche Verfahrensweisen. Beim Verfahren Typ 2, also 1:(c)n-Beziehungen, wird der Fremdschlüssel als neues Attribut in der 1-Relation aufgenommen.

**Auflösung einer 1:1-Beziehung** Diese Auflösung ist ein Verfahren vom Typ 1. Ein Beispiel einer 1:1-Beziehung findet sich in unserem Beispiel Byce & Co. nicht. Denkbar wären Personendaten im Einwohnermeldeamt und in der Sozialversicherung, die zusammengefasst werden:



Abbildung 4.14: ER-Modell 4

In diesem Beispiel werden die Attribute der beiden Relation zu einer gemeinsamen Personenrelation zusammengeführt. 1:1-Beziehungen kommen häufig vor, wenn Daten aus verschiedenen Datenbanken zu einem gemeinsamen Modell vereinigt werden oder die Attributlisten beider Entity-Mengen zusammen zu umfangreich sind. In solchen Fällen wird auch schon mal aus Performancegründen vom Standard abgewichen,

<sup>18</sup> vgl. [Elmasri et al. 2002, S. 321]

indem man die Entity-Mengen auf zwei unterschiedliche Relationen abbildet, obwohl eine 1:1-Beziehung vorliegt.

**Auflösung einer 1:n-Beziehung** Diese Auflösung ist ein Verfahren vom Typ 2. Aus dem ER-Diagramm der Firma Byce & Co. übernehmen wir als Beispiel einen Auszug des ER-Modells:

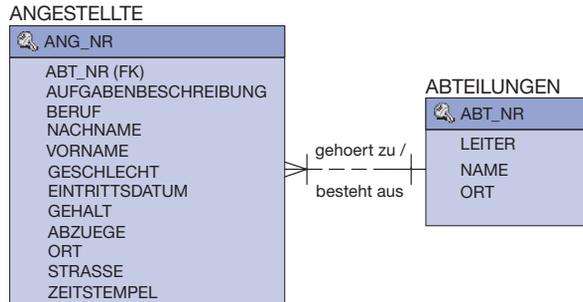


Abbildung 4.15: ER-Modell mit 1:n-Beziehung

Das Datenbankschema enthält die Relationen „Angestellte“ und „Abteilungen“, wobei die *Abt\_Nr* als Fremdschlüssel in der Angestellten-Relation aufgenommen wurde. In ERWIN wird der Fremdschlüssel *Abt\_Nr* schon im konzeptionellen Modell dargestellt, so dass konzeptionelles ER-Modell und Datenbankschema in den Attributen übereinstimmen.

In sämtlichen Datenbankschemata sind die Primärschlüssel unterstrichen, die Fremdschlüssel kursiv dargestellt. Primärschlüsselattribute, die Fremdschlüsselattribute beinhalten, sind sowohl unterstrichen als auch kursiv dargestellt.

#### Das Datenbankschema sieht wie folgt aus:

**Abteilungen** (*Abt\_Nr*, *Leiter*, *Name*, *Ort*)

**Angestellte** (*Ang\_Nr*, *Abt\_Nr*, *Aufgabenbeschreibung*, *Beruf*, *Nachname*, *Vorname*, *Geschlecht*, *Eintrittsdatum*, *Gehalt*, *Abzüge*, *Ort*, *Strasse*, *Zeitstempel*)

*Ang\_Nr* und *Abt\_Nr* sind Primärschlüssel der beiden Relationen, das Attribut *Abt\_Nr* ist Fremdschlüssel in der Relation „Angestellte“.

#### Wertebereiche bestimmter Attribute:

Hier sollen nur diejenigen Attribute zusammengetragen werden, die keinen Standardwertebereich (z.B. Zahl, Text oder Datum) haben.

Angestellte.Geschlecht kann die Werte „w“ oder „m“ annehmen.

Angestellte.Gehalt muss größer oder gleich 0 sein.

**Auflösung von n:m-Beziehungen** Diese Auflösungen sind Verfahren vom Typ 3. Als Beispiele sollen die n:m-Beziehungen „Lagerbestand“ zwischen Teilen und Lagern sowie die rekursive Beziehung „besteht aus“ von Teilen auf sich selbst dienen:



### **Copyright**

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als persönliche Einzelplatz-Lizenz zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschliesslich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs
- und der Veröffentlichung

bedarf der schriftlichen Genehmigung des Verlags.

Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

### **Zusatzdaten**

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. Der Rechtsweg ist ausgeschlossen.

### **Hinweis**

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website



herunterladen