it informatik

Ramez A. Elmasri
Shamkant B. Navathe

Grundlagen von Datenbanksystemen

Bachelorausgabe

3., aktualisierte Auflage

ADDISON-WESLEY

PEARSON Studium

KAPITEL 3

Datenmodellierung mit Hilfe des Entity-Relationship-Modells

Konzeptuelle Modellierung ist eine wichtige Phase beim erfolgreichen Entwurf einer **Datenbankanwendung**. Im Allgemeinen bezieht sich der Begriff »Datenbankanwendung« auf eine konkrete Datenbank, z.B. eine Datenbank, die Kundenkonten verwaltet, und die damit zusammenhängenden *Programme*. Diese implementieren beispielsweise Datenbankanfragen und -aktualisierungen und verändern beispielsweise Datenbanken, sobald Kunden Einzahlungen oder Abhebungen tätigen. Diese Programme weisen meist eine grafische Benutzeroberfläche (GUI) mit Formularen und Menüs auf. Folglich setzt ein Teil der Datenbankanwendung voraus, dass man diese **Anwendungsprogramme** entwirft, implementiert und testet. Traditionell fällt der Entwurf und das Testen von Anwendungsprogrammen eher in den Bereich des Software-Engineering. Andererseits wächst die Erkenntnis, dass es zwischen den Methoden des Datenbankentwurfs und des Software-Engineering Gemeinsamkeiten gibt. Diese Gemeinsamkeiten dürften aus zweierlei Gründen zunehmen: Einerseits wird versucht, in Methoden des Datenbankentwurfs vermehrt Konzepte für die Spezifikation von Operationen auf Datenbankobjekte einzubeziehen; andererseits ist man bestrebt, bei Methoden für das Software-Engineering die Struktur der Datenbanken, die Programme nutzen, ausführlicher zu spezifizieren. Kapitel 4 beschreibt einige Konzepte für die Spezifikation von Datenbankoperationen; in Kapitel 8 werden Objektdatenbanken behandelt.

In diesem Kapitel gehen wir nach dem herkömmlichen Ansatz vor und konzentrieren uns auf den Entwurf der Struktur der Datenbank und darauf definierter Einschränkungen. Wir stellen die Modellierungskonzepte des **Entity-Relationship-Modells** (kurz **ER-Modell**) vor. Dieses Modell ist ein konzeptuelles Datenmodell auf der logischen Ebene. Seine Variationen werden häufig für den konzeptuelle Entwurf von Datenbankanwendungen benutzt. Die Konzepte des ER-Modells werden in Datenbankentwurfswerkzeugen verwendet. Wir beschreiben die grundlegenden Datenstrukturierungskonzepte und Einschränkungen des ER-Modells und befassen uns anschließend mit der Anwendung des Modells für den Entwurf konzeptueller Schemas für Datenbankanwendungen.

Dieses Kapitel ist wie folgt organisiert: Abschnitt 3.1 beschreibt die Rolle logischer konzeptueller Datenmodelle im Datenbankentwurf. In Abschnitt 3.2 werden die Anforderungen für eine Beispieldatenbank aufgeführt, um die Verwendung der Konzepte des ER-Modells zu verdeutlichen. Diese Beispieldatenbank wird auch in den

nächsten Kapiteln verwendet. In Abschnitt 3.3 werden die Konzepte von Entitäten und Attributen beschrieben. Außerdem führen wir stufenweise die Diagrammtechnik für die Darstellung eines ER-Schemas ein. In Abschnitt 3.4 werden die Konzepte binärer Beziehungen sowie ihre Rollen und strukturellen Einschränkungen vorgestellt. Abschnitt 3.5 beschreibt schwache Entitätstypen. Abschnitt 3.6 zeigt, wie ein Schemaentwurf um Beziehungen erweitert bzw. verfeinert werden kann. Abschnitt 3.7 enthält eine Übersicht über die Notation für ER-Diagramme und Fragen, die beim Schemaentwurf entstehen. Außerdem wird die Benennung von Konstrukten des Datenbankschemas behandelt. Abschnitt 3.8 enthält eine Zusammenfassung dieses Kapitels.

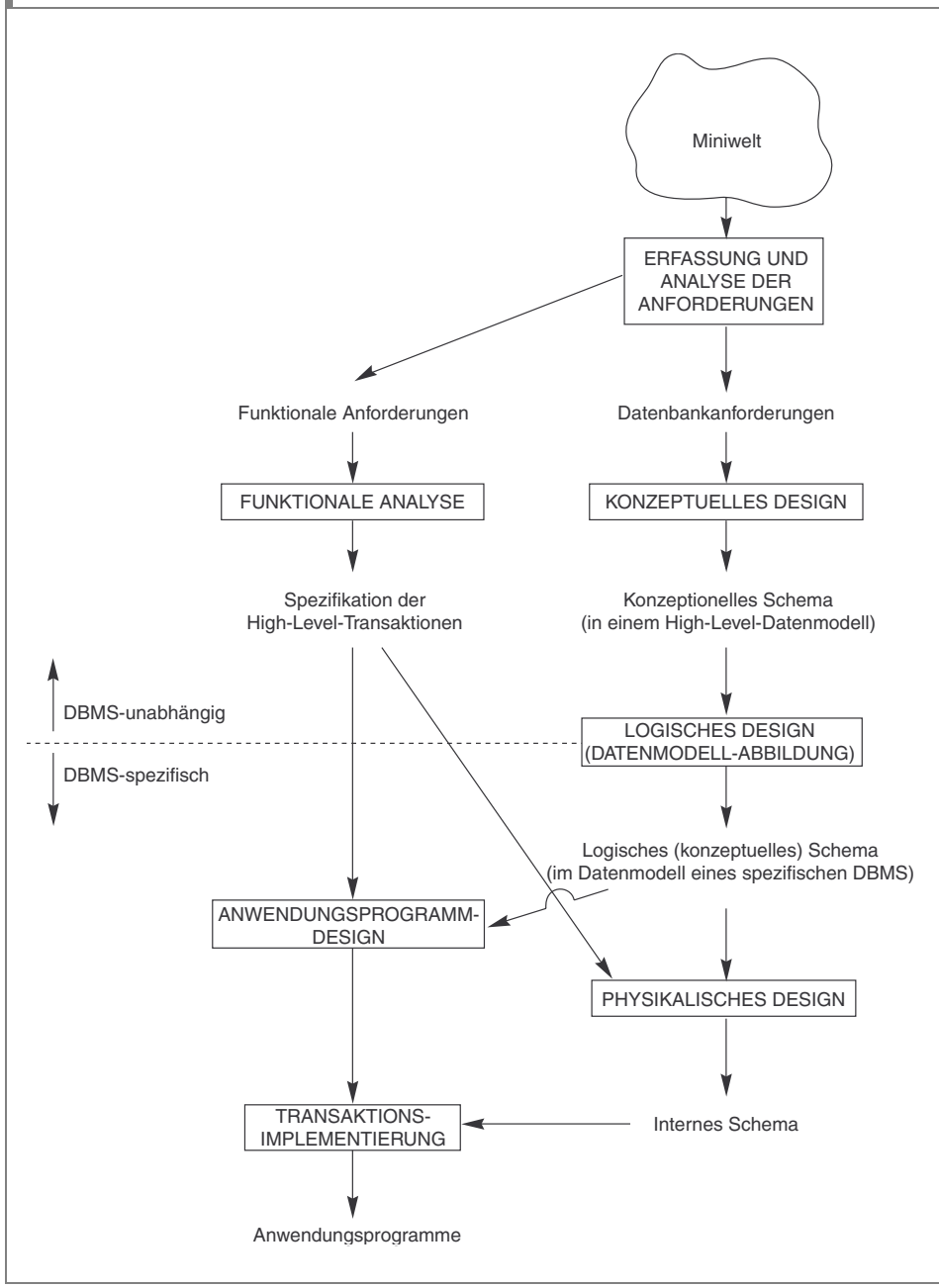
Die Beschreibung des Materials in den Abschnitten 3.3 und 3.4 ist relativ ausführlich und kann für einen Einführungskurs übersprungen werden. Sollen andererseits Datenmodellierungskonzepte und der konzeptuelle Datenbankentwurf ausführlich behandelt werden, sollte der Leser Kapitel 3 und anschließend Kapitel 4 vollständig durcharbeiten. In Kapitel 4 werden Erweiterungen des ER-Modells beschrieben, die zum so genannten Enhanced-ER- bzw. EER-Modell führen, das Konzepte wie Spezialisierung, Generalisierung, Vererbung und Unionstypen (Kategorien) beinhaltet. Außerdem enthält Kapitel 4 eine Einführung in Objektmodellierung und in die UML-Notation (Universal Modeling Language), die als Standard für Objektmodellierung vorgeschlagen wurde.

3.1 Verwendung konzeptueller Datenmodelle für den Datenbankentwurf

Abbildung 3.1 zeigt eine vereinfachte Darstellung des Datenbankentwurfsprozesses. Der erste hier dargestellte Schritt ist die **Erfassung und Analyse der Anforderungen**. In diesem Schritt befragen die Datenbankdesigner künftige Datenbanknutzer nach ihren Anforderungen an die Datenbank und den Umfang der zu speichernden Daten. Das Ergebnis dieses Schritts ist ein meist umfangreicher Katalog mit Benutzeranforderungen. Diese Anforderungen sollten so ausführlich und vollständig wie möglich spezifiziert werden. Zusammen mit der Spezifikation der Datenanforderungen sollten die bekannten **funktionalen Anforderungen** der Anwendung spezifiziert werden. Diese Anforderungen setzen sich aus den benutzerdefinierten **Operationen** (oder **Transaktionen**) zusammen, die auf die Datenbank ausgeführt werden sollen. Im Softwareentwurf ist es üblich, *Datenflussdiagramme*, *Ablaufdiagramme*, *Szenarien* und weitere Techniken anzuwenden, um funktionale Anforderungen zu spezifizieren. Eine ausführliche Diskussion dieser Techniken würde über Zweck und Umfang dieses Buchs hinausgehen; entsprechende Beschreibungen findet man in Fachbüchern aus dem Bereich Software-Engineering.

Nach der Erfassung und Analyse aller Anforderungen wird im nächsten Schritt mit Hilfe eines konzeptuellen Datenmodells ein **konzeptuelles Schema** für die Datenbank erstellt. Dieser Schritt wird als **konzeptueller Entwurf** bezeichnet. Das konzeptuelle Schema ist eine kompakte Beschreibung der Datenanforderungen der Benutzer und beinhaltet ausführliche Beschreibungen der Entitätstypen, Beziehungen und Einschränkungen. Diese Angaben werden mit Hilfe der vom High-Level-Datenmodell bereitgestellten Konzepte ausgedrückt. Da diese Konzepte keine Implementierungsdetails beinhalten, sind sie normalerweise leichter zu verstehen und können benutzt werden, um mit rechentechnisch nicht versierten Benutzern zu kommunizieren. Das konzeptuelle Schema kann auch als Referenz dienen, um sicherzustellen, dass alle Datenanforderungen der Benutzer erfüllt werden und keine widersprüchlichen oder

Abbildung 3.1: Vereinfachtes Diagramm mit den wichtigsten Phasen im Datenbankentwurf.



gegenläufigen Anforderungen enthalten sind. Dieser Ansatz ermöglicht es den Datenbankdesignern, sich auf die Spezifikation der Eigenschaften der Daten zu konzentrieren, ohne sich mit Speicherdetails befassen zu müssen. Somit ist es für sie leichter, einen guten konzeptuellen Datenbankentwurf sicherzustellen.

Beim Entwurf des konzeptuellen Schemas können die grundlegenden Datenmodelloperationen herangezogen werden, um die höherschichtigen Benutzeroperationen zu spezifizieren, die in der funktionalen Analyse identifiziert wurden. Dies dient auch als Bestätigung, dass das konzeptuelle Schema alle identifizierten funktionalen Anforderungen erfüllt. Modifikationen des konzeptuellen Schemas lassen sich einbinden, wenn eine funktionale Anforderung nicht im anfänglichen Schema spezifiziert werden kann.

Der nächste Schritt im Datenbankentwurf ist die eigentliche Implementierung der Datenbank unter Verwendung eines kommerziellen DBMS. Die meisten heute angebotenen kommerziellen DBMS verwenden ein Implementierungsdatenmodell, z.B. das relationale oder Objekt-Datenbankmodell, so dass das konzeptuelle Schema vom konzeptuellen Datenmodell in das Implementierungsdatenmodell transformiert wird. Dieser Schritt wird als **logischer Entwurf** oder auch **Datenmodellabbildung** bezeichnet und hat als Ergebnis ein Datenbankschema im Implementierungsdatenmodell des DBMS.

Schließlich folgt im letzten Schritt die Phase des **physischen Entwurfs**, in der die internen Speicherstrukturen, Zugriffspfade und Dateiorganisationen festgelegt werden. Parallel zu diesen Aktivitäten werden Anwendungsprogramme entworfen und implementiert, d.h., es werden Datenbanktransaktionen realisiert, die den logischen Transaktionsspezifikationen entsprechen. Der Datenbankentwurfsprozess wird in Kapitel 16 der ungekürzten Ausgabe auf der CWS mit einer Übersicht des physischen Datenbankentwurfs wieder aufgegriffen.

Wir präsentieren in diesem Kapitel nur die ER-Modellkonzepte für den Entwurf des konzeptuellen Schemas. Die Realisierung der benutzerdefinierten Operationen wird in Zusammenhang mit Objektmodellierung in Kapitel 4 behandelt.

3.2 Beispiel für eine Datenbankanwendung

In diesem Abschnitt wird eine Beispieldatenbankanwendung namens `FIRMA` beschrieben, die uns dazu dienen soll, die ER-Modellkonzepte und ihre Verwendung im Schemaentwurf zu erläutern. Wir führen die Datenanforderungen für die Datenbank hier auf und erstellen dann schrittweise das konzeptuelle Schema im Zuge der Einführung der Modellierungskonzepte des ER-Modells. Die Datenbank `FIRMA` verwaltet die Angestellten, Abteilungen und Projekte einer Firma. Angenommen, die Phase der Erfassung und Analyse der Anforderungen ist abgeschlossen und die Designer wenden sich dem in der Datenbank darzustellenden Teil der Firma zu, d.h., sie haben folgende Beschreibung der »Miniwelt« erstellt:

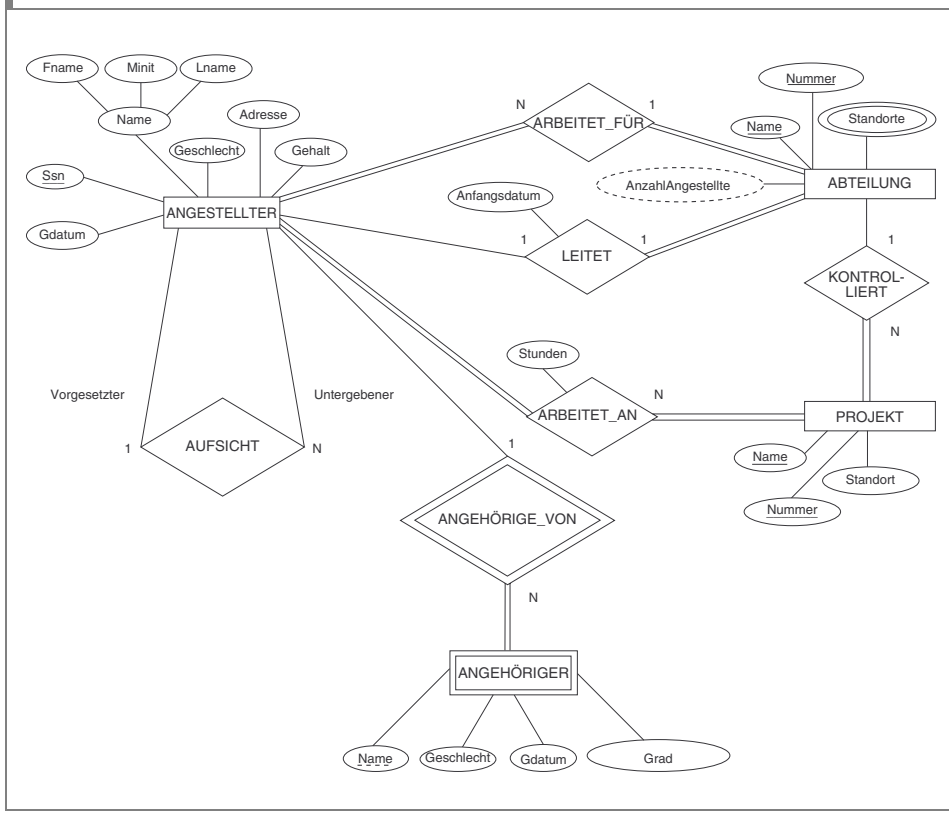
1. Die Firma ist in Abteilungen organisiert. Jede Abteilung hat eine eindeutige Bezeichnung, eine eindeutige Nummer und einen bestimmten Angestellten, der die Abteilung leitet. Wir verfolgen das Anfangsdatum, ab dem dieser Angestellte die Leitung der Abteilung übernommen hat. Eine Abteilung verfügt über mehrere Standorte.

2. Eine Abteilung kontrolliert eine Reihe von Projekten, die jeweils einen eindeutigen Namen, eine eindeutige Nummer und einen einzigen Standort haben.

3. Wir speichern zu jedem Angestellten den Namen, die Sozialversicherungsnummer[1], die Adresse, das Gehalt, das Geschlecht und das Geburtsdatum. Ein Angestellter wird einer Abteilung zugewiesen, kann aber an mehreren Projekten arbeiten, die nicht unbedingt alle von der gleichen Abteilung kontrolliert werden. Wir verfolgen die Stundenzahl pro Woche, die ein Angestellter an jedem Projekt arbeitet, und den unmittelbaren Vorgesetzten jedes Angestellten.
4. Zu Versicherungszwecken möchten wir die Familienangehörigen jedes Mitarbeiters verfolgen. Wir führen also jeden Angehörigen mit Vorname, Geschlecht, Geburtsdatum und Verwandtschaftsgrad zum jeweiligen Angestellten.

Abbildung 3.2 zeigt, wie das Schema für diese Datenbankanwendung als so genanntes **ER-Diagramm** dargestellt werden kann. Wir beschreiben, wie dieses Schema von den festgehaltenen Anforderungen abgeleitet werden kann, und erklären die Notation des ER-Diagramms im Zuge unserer Einführung der ER-Modellkonzepte im folgenden Abschnitt.

Abbildung 3.2: ER-Schemadiagramm für die Datenbank FIRMA.



1. Die Sozialversicherungsnummer (SSN) in den USA ist eine eindeutige 9-stellige Personenidentifizierung, die jedem US-Bürger zugeteilt wird, um Beschäftigtenverhältnis, Verdienst und Steuern zu verfolgen; etwa vergleichbar mit Lohnsteuerkarten in Deutschland.

3.3 Entitätstypen, Entitätsmengen, Attribute und Schlüssel

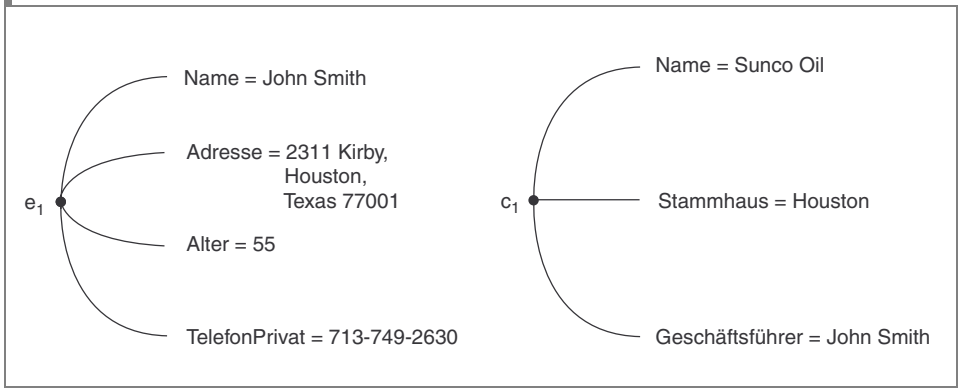
Das ER-Modell beschreibt Daten als Entitäten, Beziehungen und Attribute. In Abschnitt 3.3.1 werden die Konzepte von Entitäten und ihrer Attribute eingeführt. Entitätstypen und Schlüsselattribute werden in Abschnitt 3.3.2 beschrieben. Abschnitt 3.3.3 spezifiziert den konzeptuellen Grobentwurf der Entitätstypen für die Datenbank `FIRMA` und die Beziehungen werden in Abschnitt 3.4 beschrieben.

3.3.1 Entitäten und Attribute

Entitäten und ihre Attribute Das vom ER-Modell dargestellte Basisobjekt ist eine **Entität**, also »etwas« aus der realen Welt mit einer unabhängigen Existenz. Eine Entität kann ein Objekt sein, das physisch existiert, z.B. eine Person, ein Auto, ein Haus oder ein Angestellter, oder das konzeptuell existiert, z.B. eine Firma, eine Arbeitsstelle oder ein Universitätskurs. Jede Entität hat **Attribute**, d.h. bestimmte Eigenschaften, die sie beschreiben. Eine Angestelltenentität kann beispielsweise durch Name, Alter, Adresse, Gehalt und Arbeitsstelle des Mitarbeiters beschrieben werden. Eine bestimmte Entität hat für jedes ihrer Attribute einen **Wert**. Die Attributwerte, die jede Entität beschreiben, bilden einen wichtigen Teil der in der Datenbank gespeicherten Daten.

Abbildung 3.3 zeigt zwei Entitäten mit Attributwerten. Die Angestelltenentität e_1 hat vier Attribute: Name, Adresse, Alter und TelefonPrivat; die Werte lauten »John Smith«, 2311 Kirby, Houston, Texas 77001«, »55« bzw. »713-749-2630«. Die Firmenentität c_1 hat drei Attribute: Name, Stammhaus und Geschäftsführer; die entsprechenden Werte sind »Sunco Oil«, »Houston« und »John Smith«.

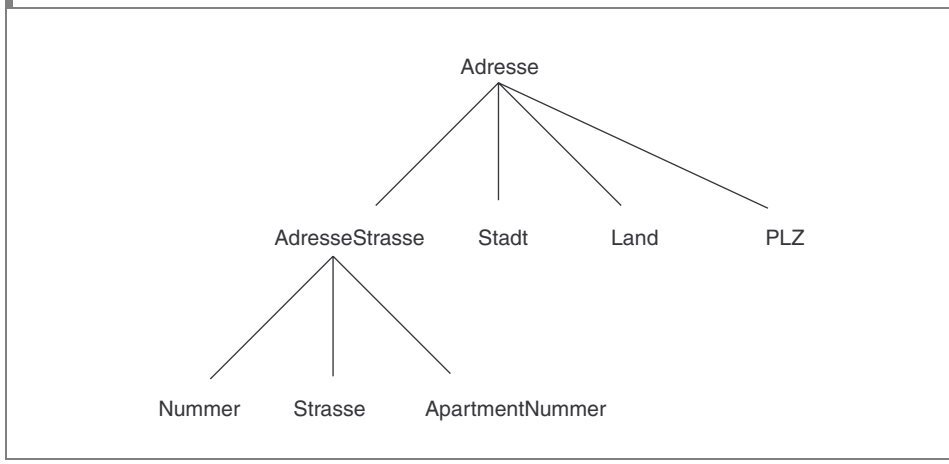
Abbildung 3.3: Die Entitäten eines Angestellten e_1 und einer Firma c_1 mit den entsprechenden Attributwerten.



Im ER-Modell kommen mehrere Attributtypen vor: *einfach* bzw. *zusammengesetzt*, *einwertig* bzw. *mehrwertig* und *gespeichert* bzw. *abgeleitet*. Wir definieren zuerst diese Attributtypen und zeigen dann die jeweilige Verwendung anhand von Beispielen auf. Anschließend beschreiben wir das Konzept eines *Nullwerts* für ein Attribut.

Zusammengesetzte oder einfache (atomare) Attribute Zusammengesetzte Attribute lassen sich in kleinere Teile zerlegen, die grundlegendere Attribute mit unabhängigen Bedeutungen darstellen. Das Attribut Adresse der Entität Angestellter in Abbildung 3.3 kann man beispielsweise in AdresseStrasse, Stadt, Land und PLZ[2] mit den Werten »2311 Kirby«, »Houston«, »Texas« und »77001« unterteilen. Nicht teilbare Attribute nennt man **einfache** oder **atomare Attribute**. Zusammengesetzte Attribute können eine Hierarchie bilden; beispielsweise lässt sich AdresseStrasse in drei einfache Attribute, nämlich Nummer, Strasse und ApartmentNummer, weiter aufteilen (siehe Abbildung 3.4). Der Wert eines zusammengesetzten Attributs ist die Verkettung der Werte der einfachen Attribute, aus denen es sich zusammensetzt.

Abbildung 3.4: Eine Hierarchie aus zusammengesetzten Attributen; die Komponente AdresseStrasse einer Adresse besteht aus den weiteren Komponenten Nummer, Strasse und ApartmentNummer.



Zusammengesetzte Attribute sind in Modellsituationen nützlich, in denen ein Benutzer sich manchmal auf das zusammengesetzte Attribut insgesamt, andere Male jedoch spezifisch auf seine Komponenten bezieht. Wird auf das zusammengesetzte Attribut nur als Ganzes zugegriffen, besteht keine Notwendigkeit, es in Komponentenattribute aufzuteilen. Wenn es beispielsweise nicht nötig ist, auf die einzelnen Komponenten einer Adresse (PLZ, Strasse usw.) zuzugreifen, gilt die ganze Adresse als einfaches Attribut.

Einwertige oder mehrwertige Attribute Die meisten Attribute haben nur einen einzigen Wert für eine bestimmte Entität zu einem bestimmten Zeitpunkt; sie werden deshalb **einwertige** Attribute genannt. Alter ist z.B. ein einwertiges Attribut von Personen. In manchen Fällen kann ein Attribut mehrere Werte für die gleiche Entität zum gleichen Zeitpunkt annehmen, z.B. ein Attribut Farben für ein Auto oder Universitätsabschluss für eine Person. Autos mit einer Farbe haben einen einzigen Wert, während solche mit zwei Farbtönen zwei Werte für das Attribut Farben haben. Ebenso hat eine Person vielleicht keinen akademischen Titel, wieder eine andere hat einen und eine dritte Person hat vielleicht zwei oder mehr Titel. Unterschiedliche Personen können

2. Postleitzahl; wird in den USA als *ZIP* bezeichnet.

also verschiedene Werte für das Attribut Universitätsabschluss haben. Dies sind die so genannten **mehrwertigen Attribute**. Ein mehrwertiges Attribut kann eine Unter- und Obergrenze für die Anzahl der zulässigen Werte für jede einzelne Entität haben. Das Attribut Farben eines Autos kann z.B. zwischen einem und drei Werten haben, wenn wir davon ausgehen, dass das Auto höchstens drei Farben hat.

Gespeicherte oder abgeleitete Attribute In manchen Fällen stehen zwei (oder mehr) Attributwerte in einer Beziehung, z.B. die Attribute Alter und Geburtsdatum einer Person. Für eine bestimmte Personenentität lässt sich der Wert für Alter aus dem aktuellen Tagesdatum und dem Wert von Geburtstag der Person ermitteln. Das Attribut Alter ist demzufolge ein **abgeleitetes Attribut**, d.h., es kann vom Attribut Geburtsdatum, das als **gespeichertes Attribut** bezeichnet wird, abgeleitet werden. Einige Attributwerte können von *zusammenhängenden Entitäten* abgeleitet werden. Beispielsweise kann ein Attribut AnzahlAngestellte einer Entität Abteilung durch Zählung der Angestellten, die für die betreffende Abteilung arbeiten, abgeleitet werden.

Nullwerte In manchen Fällen trifft auf ein Attribut einer bestimmten Entität vielleicht kein Wert zu. Das Attribut ApartmentNummer einer Adresse ist z.B. nur für Wohnhäuser, nicht aber für Einfamilienhäuser relevant. Ebenso trifft das Attribut CollegeGrad nur auf Personen mit einem akademischen Grad zu. Für solche Fälle wird ein spezieller Wert erzeugt, der als **Nullwert** bezeichnet wird. Die Adresse eines Einfamilienhauses hätte für das Attribut ApartmentNummer und eine Person ohne akademischen Grad für das Attribut Universitätsabschluss einen Nullwert. Ein Nullwert kann aber auch benutzt werden, wenn der Wert eines Attributs für eine bestimmte Entität nicht bekannt ist, z.B. wenn die private Telefonnummer von »John Smith« in Abbildung 3.3 nicht bekannt ist. Im ersten Fall hat der Nullwert die Bedeutung *nicht zutreffend* und im zweiten *unbekannt*. Die Kategorie »unbekannt« von Nullwerten lässt sich weiter in zwei Fälle klassifizieren. Im ersten Fall ist bekannt, dass der Attributwert existiert, aber ein Wert *fehlt*, z.B. wenn für das Attribut Größe einer Person ein Nullwert angegeben wird, wobei klar ist, dass jeder Mensch eine bestimmte Größe hat. Im zweiten Fall ist *nicht bekannt*, ob der Attributwert existiert, z.B. wenn das Attribut TelefonPrivat einen Nullwert hat.

Komplexe Attribute Zusammengesetzte und mehrwertige Attribute können beliebig verschachtelt werden. Wir können dieses Verschachteln dadurch darstellen, dass wir Komponenten eines zusammengesetzten Attributs zwischen Klammern () setzen, die Komponenten durch Kommas trennen und mehrwertige Attribute zwischen geschweiften Klammer {} setzen. Das sind die so genannten **komplexen Attribute**. Wenn eine Person z.B. mehr als eine Adresse und in jeder Wohnung mehrere Telefonanschlüsse hat, kann das Attribut AdresseTelefon einer Entität PERSON wie in Abbildung 3.5 spezifiziert werden.

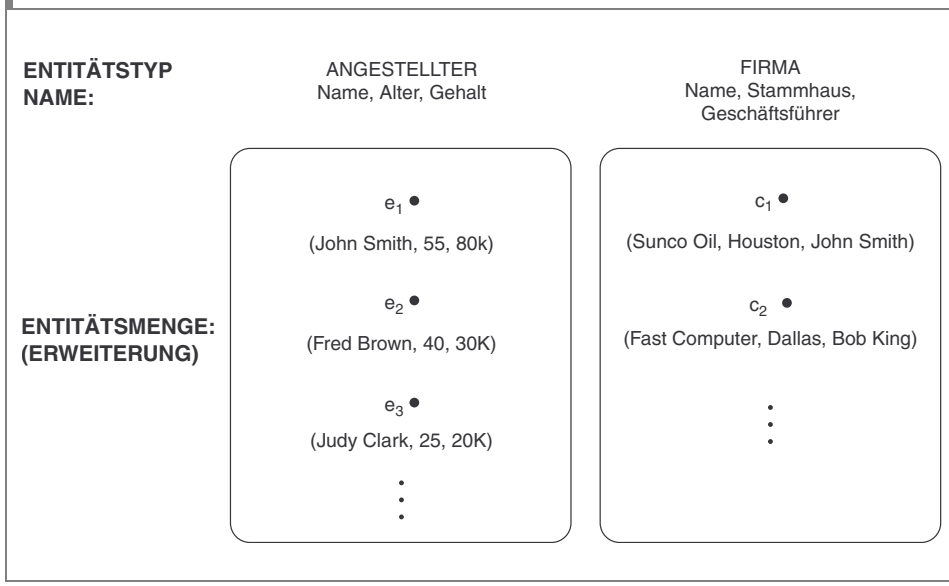
Abbildung 3.5: Das komplexe Attribut AdresseTelefon mit mehrwertigen und zusammengesetzten Komponenten.

{AdresseTelefon({Telefon(Vorwahl, Telefonnummer)})},
Adresse(AdresseStrasse(Nummer, Strasse, ApartmentNummer), Stadt, Land, PLZ)

3.3.2 Entitätstypen, Entitätsmengen, Schlüssel und Wertemengen

Entitätstypen und -mengen Eine Datenbank enthält normalerweise Gruppen gleicher Entitäten. Eine Firma, die Hunderte von Angestellten beschäftigt, möchte z.B. die gleichen Informationen über jeden Angestellten speichern. Diese Angestelltenentitäten weisen gemeinsame Attribute auf, jede Entität hat aber *eigene Werte* für jedes Attribut. Ein **Entitätstyp** definiert eine *Sammlung* (oder *Menge*) von Entitäten mit den gleichen Attributen. Jeder Entitätstyp in der Datenbank wird nach Name und Attributen beschrieben. Abbildung 3.6 zeigt zwei Entitätstypen mit den Namen `ANGESTELLTER` und `FIRMA` und jeweils einer Liste von Attributen. In der Abbildung sind außerdem ein paar einzelne Entitäten jedes Typs mit Attributwerten dargestellt. Die Sammlung aller Entitäten eines bestimmten Entitätstyps in der Datenbank zu einem bestimmten Zeitpunkt wird als **Entitätsmenge** bezeichnet. Auf eine Entitätsmenge wird normalerweise mit dem gleichen Namen wie der Entitätstyp zugegriffen, wenn sich `ANGESTELLTER` z.B. auf einen *Entitätstyp* und die aktuelle *Menge aller Angestelltenentitäten* in der Datenbank bezieht.

Abbildung 3.6: Zwei Entitätstypen, `ANGESTELLTER` und `FIRMA`, und einige Mitgliederentitäten aus der Entitätssammlung (bzw. Entitätsmenge) jedes Typs.



Ein Entitätstyp wird in ER-Diagrammen[3] (siehe Abbildung 3.2) als Rechteck dargestellt, in dem der Name des Entitätstyps steht. Attributnamen sind durch Ovale symbolisiert und durch gerade Linien mit ihrem Entitätstyp verbunden. Zusammenge-

3. Die von uns verwendete Notation für ER-Diagramme lehnt sich eng an die ursprünglich vorgeschlagene Notation an (Chen, 1976). Leider sind viele andere Notationen in Gebrauch, von denen einige in Anhang A und später in diesem Kapitel dargestellt werden.

setzte Attribute werden mit ihren Komponentenattributen durch gerade Linien verbunden. Mehrwertige Attribute werden in doppelten Ovalen dargestellt.

Ein Entitätstyp beschreibt das **Schema** bzw. die **Intension** einer *Entitätsmenge*, deren einzelne Entitäten allesamt die gleiche Struktur aufweisen. Entitäten eines bestimmten Entitätstyps werden zu jeweils einer Entitätsmenge gruppiert, die man auch als die **Extension** des Entitätstyps bezeichnet.

Schlüsselattribute eines Entitätstyps Eine wichtige Einschränkung für die Entitäten eines Entitätstyps ist der **Schlüssel** bzw. eine **Eindeutigkeitseinschränkung** für Attribute. Ein Entitätstyp hat normalerweise ein Attribut, dessen Werte sich für jede einzelne Entität der Sammlung unterscheiden. Ein solches Attribut nennt man **Schlüsselattribut** und anhand seiner Werte lässt sich jede Entität eindeutig identifizieren. In Abbildung 3.6 ist z.B. das Attribut Name ein Schlüssel für den Entitätstyp `FIRMA`, weil keine zwei Firmen den gleichen Namen haben dürfen. Für den Entitätstyp `PERSON` ist Sozialversicherungsnummer ein typisches Schlüsselattribut. Manchmal bilden mehrere Attribute zusammengenommen einen Schlüssel, was bedeutet, dass sich die *Kombination* der Attributwerte jeder Entität unterscheiden muss. Weisen mehrere Attribute diese Eigenschaft auf, können wir ein *zusammengesetztes Attribut* definieren, das dann als Schlüsselattribut für den Entitätstyp dient. Man beachte aber, dass ein zusammengesetzter Schlüssel *minimal* sein sollte; d.h., alle im zusammengesetzten Attribut enthaltenen Komponentenattribute müssen die Eindeutigkeitseigenschaft[4] aufweisen. In ER-Diagrammen wird jedes Schlüsselattribut als Oval mit darin befindlichem **unterstrichenem** Namen dargestellt (siehe Abbildung 3.2).

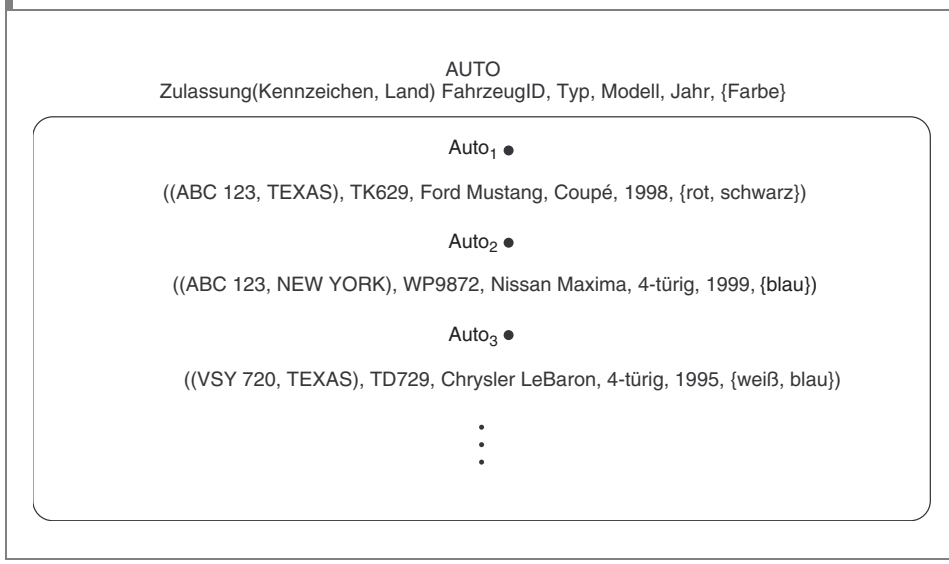
Die Spezifikation eines Attributs als Schlüssel für einen Entitätstyp bedeutet, dass die vorausgehende Eindeutigkeitseigenschaft auf *jede Erweiterung* des Entitätstyps zutreffen muss. Folglich liegt die Einschränkung vor, dass keine zwei Entitäten zum gleichen Zeitpunkt den gleichen Wert für das Schlüsselattribut besitzen dürfen. Dies ist nicht die Eigenschaft einer bestimmten Erweiterung, sondern vielmehr eine Einschränkung für *alle Erweiterungen* des Entitätstyps. Diese Schlüsseleinschränkung (und weitere Einschränkungen, die später beschrieben werden) wird aus den Einschränkungen der Miniwelt abgeleitet, die von der Datenbank dargestellt wird.

Einige Entitätstypen haben *mehr als ein* Schlüsselattribut. Beispielsweise sind die beiden Attribute FahrzeugKZ und Zulassung des Entitätstyps `AUTO` in Abbildung 3.7 eigenständige Schlüssel. Das Attribut Zulassung ist ein Beispiel eines zusammengesetzten Schlüssels, der aus den beiden einfachen Komponentenattributen Zulassungsnummer und Land gebildet wurde, die beide keine eigenständigen Schlüssel sind. Ein Entitätstyp kann also auch *keinen Schlüssel* haben; in diesem Fall wird er als *schwacher Entitätstyp* bezeichnet (siehe Abschnitt 3.5).

Wertemengen (Domänen oder Wertebereiche) von Attributen Jedes einfache Attribut eines Entitätstyps ist mit einer **Wertemenge** assoziiert, die festlegt, welche Werte diesem Attribut für jede einzelne Entität zugewiesen werden können. Wenn der für Angestellte zulässige Altersbereich auf 16 bis 70 festgesetzt wurde, können wir die Wertemenge des Attributs Alter von `ANGESTELLTER` als Menge von ganzzahligen Zahlen zwischen 16 und 70 spezifizieren. Ebenso können wir die Wertemenge für das Attribut Name auf die Menge von Zeichenketten spezifizieren, die aus alphabetischen Zei-

4. Überflüssige Attribute dürfen nicht in einen Schlüssel einbezogen werden, während ein **Superschlüssel** auch überflüssige Attribute beinhalten kann, wie in Kapitel 5 erklärt wird.

Abbildung 3.7: Der Entitätstyp AUTO mit zwei Schlüsselattributen, Zulassung und FahrzeugKZ; mehrwertige Attribute stehen zwischen geschweiften Klammern {} und Komponenten eines zusammengesetzten Attributs zwischen runden Klammern ().



chen, getrennt durch Leerzeichen, bestehen usw. Wertemengen werden in ER-Diagrammen nicht dargestellt.

Mathematisch kann ein Attribut *A* vom Entitätstyp *E*, dessen Wertemenge *V* ist, als **Funktion** von *E* zur Potenzmenge[5] *P(V)* von *V* definiert werden:

$$A : E \rightarrow P(V)$$

Wir beziehen uns auf den Wert von Attribut *A* für Entität *e* als *A(e)*. Die vorherige Definition gilt für einwertige und mehrwertige Attribute sowie Nullwerte. Ein Nullwert wird durch eine leere Menge dargestellt. Für einwertige Attribute wird *A(e)* als Singleton[6] für jede Entität *e* in *E* eingeschränkt, während für mehrwertige Attribute keine Einschränkung besteht. Für ein zusammengesetztes Attribut *A* ist die Wertemenge *V* das kartesische Produkt von $P(V_1)$, $P(V_2)$, ..., $P(V_n)$, wobei V_1, V_2, ..., V_n die Wertemengen der einfachen Komponentenattribute sind, die *A* bilden:

$$V = P(V_1) \times P(V_2) \times \ldots \times P(V_n)$$

3.3.3 Konzeptueller Grobentwurf der Datenbank FIRMA

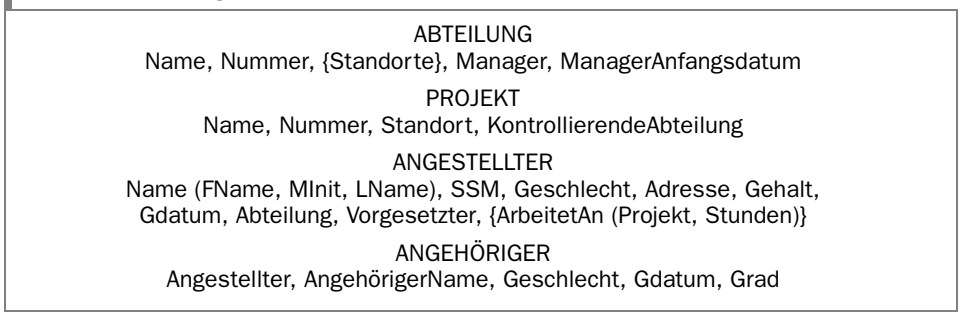
Wir können jetzt die Entitätstypen für die Datenbank FIRMA auf der Grundlage der in Abschnitt 3.2 beschriebenen Anforderungen definieren. Nach der Definition mehrerer Entitätstypen und ihrer Attribute *verfeinern* wir unseren Entwurf in Abschnitt 3.4 (nach der Einführung des Konzepts einer Beziehung). Entsprechend den in Abschnitt

5. Die **Potenzmenge** *P(V)* einer Menge *V* ist die Menge aller Teilmengen von *V*.
6. Ein **Singleton** ist eine Menge mit nur einem Element (Wert).

3.2 aufgeführten Anforderungen können wir vier Entitätstypen identifizieren, von denen je einer den vier Exemplaren aus der Spezifikation entspricht (siehe Abbildung 3.8):

1. Ein Entitätstyp ABTEILUNG mit den Attributen Name, Nummer, Standort, Manager und ManagerAnfangsdatum, wobei Standort das einzige mehrwertige Attribut ist. Wir können Name und Nummer als (getrennte) Schlüsselattribute spezifizieren, weil beide eindeutig sind.
2. Ein Entitätstyp PROJEKT mit den Attributen Name, Nummer, Standort und KontrollierendeAbteilung, wobei Name und Nummer (getrennte) Schlüsselattribute sind.
3. Ein Entitätstyp ANGESTELLTER mit den Attributen Name, SSN (Sozialversicherungsnummer), Geschlecht, Adresse, Gehalt, Geburtsdatum, Abteilung und Vorgesetzter, wobei Name und Adresse zusammengesetzte Attribute sein können. Dies wurde in den Anforderungen allerdings nicht spezifiziert. Wir müssen uns wieder an die Benutzer wenden, um festzustellen, ob Zugriffe auf die einzelnen Komponenten von Name, d.h. Vorname, Initial und Nachname, oder von Adresse erforderlich sind.
4. Ein Entitätstyp ANGEHÖRIGER mit den Attributen Angestellter, AngehörigerName, Geschlecht, Geburtsdatum und Grad (Verwandtschaftsgrad zum Angestellten).

Abbildung 3.8: Vorläufiger Entwurf von Entitätstypen für die Datenbank FIRMA, deren Anforderungen in Abschnitt 3.2 beschrieben wurden.



Bisher haben wir die Tatsache nicht berücksichtigt, dass ein Angestellter an mehreren Projekten arbeiten kann. Außerdem haben wir die Anzahl der Stunden pro Woche, die ein Angestellter an jedem Projekt arbeitet, nicht dargestellt. Diese Eigenschaft ist Teil von Anforderung 3 in Abschnitt 3.2 und kann durch ein mehrwertiges zusammengesetztes Attribut von ANGESTELLTER unter der Bezeichnung ArbeitetAn mit einfachen Komponenten (Projekt, Stunden) dargestellt werden. Alternativ kann man sie als mehrwertiges zusammengesetztes Attribut von PROJEKT mit der Bezeichnung Arbeiter und einfachen Komponenten (Angestellter, Stunden) darstellen. Wir haben uns in Abbildung 3.8, in der die oben beschriebenen Entitätstypen dargestellt sind, für die erste Alternative entschieden. Das Attribut Name von ANGESTELLTER ist als zusammengesetztes Attribut dargestellt, möglicherweise nach erneuter Rücksprache mit den Benutzern.

3.4 Beziehungen, Beziehungstypen, Rollen und strukturelle Einschränkungen

In Abbildung 3.8 bestehen mehrere *implizite Beziehungen* zwischen den verschiedenen Entitätstypen. Das heißt, wenn sich ein Attribut eines Entitätstyps auf einen anderen Entitätstyp bezieht, wird hiermit eine Beziehung wiedergegeben. Beispielsweise bezieht sich das Attribut Manager von ABTEILUNG auf einen Angestellten, der die Abteilung leitet; das Attribut KontrollierendeAbteilung von PROJEKT bezieht sich auf die Abteilung, die für das Projekt verantwortlich zeichnet; das Attribut Vorgesetzter von ANGESTELLTER nimmt Bezug auf einen anderen Angestellten (denjenigen, der diesen Angestellten beaufsichtigt); das Attribut Abteilung von ANGESTELLTER bezieht sich auf die Abteilung, für die der Angestellte arbeitet; und so weiter. Im ER-Modell sollten diese Zusammenhang nicht als Attribute, sondern als **Beziehungen** dargestellt werden, die in diesem Abschnitt erklärt werden. Das Datenbankschema FIRMA wird in Abschnitt 3.6 auf die ausdrückliche Darstellung der Beziehungen verfeinert. Im Grobentwurf der Entitätstypen werden Beziehungen normalerweise in Form von Attributen erfasst. Während der Verfeinerung des Entwurfs erfolgt eine Umwandlung dieser Attribute in Beziehungen zwischen Entitätstypen.

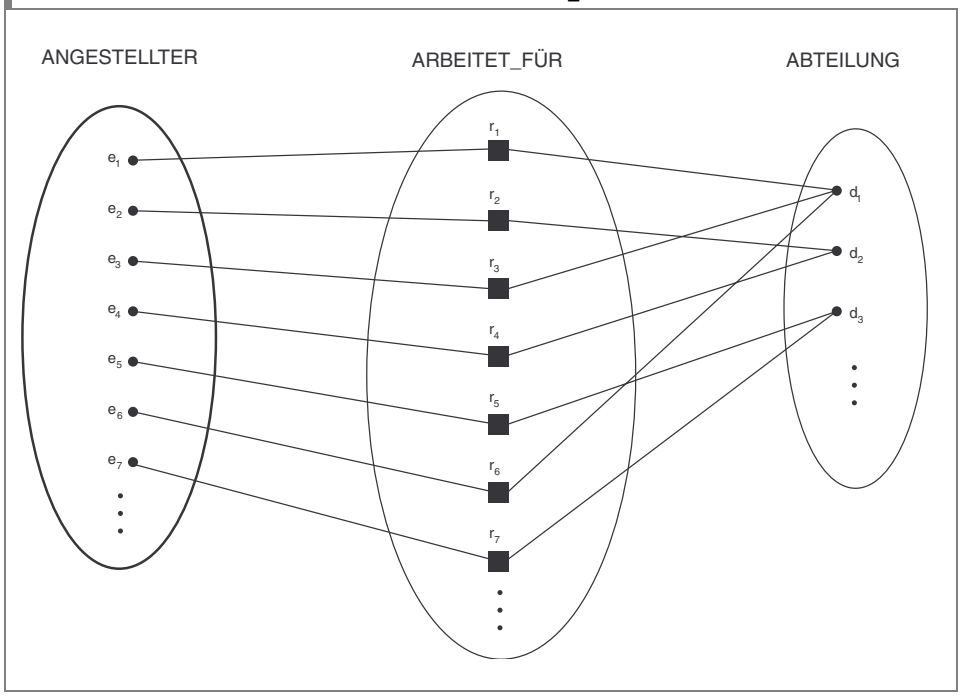
Dieser Abschnitt ist wie folgt organisiert: Abschnitt 3.4.1 enthält eine Einführung in die Konzepte von Beziehungstypen, Mengen und Instanzen. In Abschnitt 3.4.2 werden die Konzepte von Beziehungsgraden, Rollennamen und rekursiven Beziehungen definiert. Abschnitt 3.4.3 beschreibt strukturelle Einschränkungen in Bezug auf Beziehungen, z.B. Kardinalitätsverhältnisse (1:1, 1:N, M:N) und Existenzabhängigkeiten. Abschnitt 3.4.4 zeigt, dass Beziehungstypen auch Attribute haben können.

3.4.1 Beziehungstypen, Mengen und Instanzen

Ein **Beziehungstyp** R zwischen n Entitätstypen $E_1, E_2, ..., E_n$ definiert eine Reihe von Assoziationen, also eine **Beziehungsmenge** zwischen Entitäten dieser Typen. Wie bei Entitätstypen und Entitätsmengen erhalten ein Beziehungstyp und seine Beziehungsmenge üblicherweise den *gleichen Namen R*. Mathematisch ist die Beziehungsmenge R eine Menge von **Beziehungsinstanzen** r_i, wobei jede r_i n einzelne Entitäten $(e_1, e_2, ..., e_n)$ assoziiert und jede Entität e_j in r_i ein Mitglied des Entitätstyps E_j, $1 \le j \le n$, ist. Ein Beziehungstyp ist folglich eine mathematische Relation zu $E_1, E_2, ..., E_n$, oder kann alternativ als Teilmenge des kartesischen Produkts $E_1 \times E_2 \times ... \times E_n$ definiert werden. Jeder der Entitätstypen $E_1, E_2, ..., E_n$ ist **Teilnehmer** im Beziehungstyp R, ebenso wie jede der einzelnen Entitäten $e_1, e_2, ..., e_n$ an der Beziehungsinstanz $r_i = (e_1, e_2, ..., e_n)$ teilnimmt.

Informell entspricht jede Beziehungsinstanz r_i in R einer Assoziation von Entitäten, wobei die Assoziation genau eine Entität von jedem teilnehmenden Entitätstyp beinhaltet. Jede Beziehungsinstanz r_i stellt die Tatsache dar, dass die Entitäten, die an r_i teilnehmen, auf die eine oder andere Art einen Bezug zur entsprechenden Situation in der Miniwelt haben. Man betrachte beispielsweise einen Beziehungstyp ARBEITET_FÜR zwischen den beiden Entitätstypen ANGESTELLTER und ABTEILUNG, der jeden Angestellten mit der Abteilung assoziiert, für die er arbeitet. Jede Beziehungsinstanz in der Beziehungsmenge ARBEITET_FÜR assoziiert eine Angestellten- und eine Abteilungsentität. Abbildung 3.9 zeigt dieses Beispiel, wobei jede Beziehungsinstanz r_i mit den Angestellten- und Abteilungsentitäten verbunden ist, die an r_i teilnehmen. In der in Abbildung 3.9 dargestellten Miniwelt arbeiten die Angestellten e_1, e_3 und e_6 für die Abteilung d_1; e_2 und e_4 arbeiten für d_2; und e_5 und e_7 arbeiten für d_3.

Abbildung 3.9: Einige Instanzen der Beziehung ARBEITET_FÜR zwischen ANGESTELLTER und ABTEILUNG.



In ER-Diagrammen werden Beziehungstypen als Rauten dargestellt, die durch gerade Linien mit den Rechtecken verbunden sind, die die teilnehmenden Entitätstypen darstellen. Der Beziehungsname wird in der Raute angegeben (siehe Abbildung 3.2).

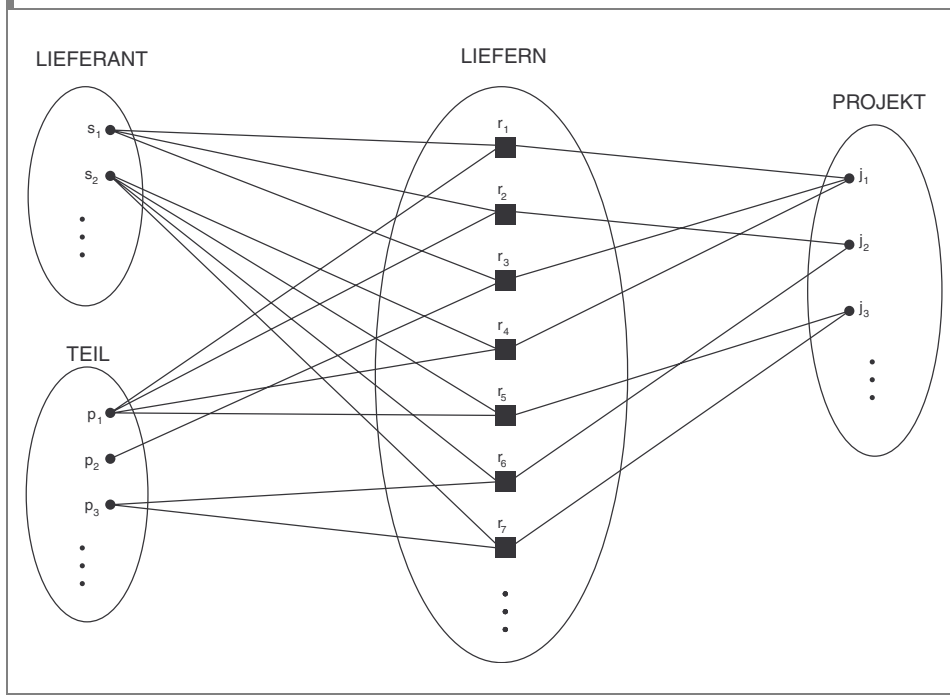
3.4.2 Beziehungsgrad, Rollennamen und rekursive Beziehungen

Grad eines Beziehungstyps Mit **Grad** eines Beziehungstyps ist die Anzahl der teilnehmenden Entitätstypen gemeint. Folglich hat die Beziehung ARBEITET_FÜR einen Grad von Zwei. Ein Beziehungstyp mit Grad Zwei wird als **binär** und einer mit Grad drei als **ternär** bezeichnet.

Abbildung 3.10 zeigt ein Beispiel für eine ternäre Beziehung, LIEFERN, wobei jede Beziehungsinstanz r_i jedes Mal drei Entitäten – Lieferant s, Teil p und Projekt j – assoziiert, wenn s ein Teil p an Projekt j liefert. Beziehungen können im Allgemeinen jeden beliebigen Grad haben, die häufigsten sind aber die binären. Höhergradige Beziehungen sind generell komplexer als binäre; sie werden in Kapitel 4 wieder aufgegriffen.

Beziehungen als Attribute Manchmal ist es praktisch, sich einen Beziehungstyp in Bezug auf Attribute vorzustellen, wie in Abschnitt 3.3.3 beschrieben wurde. Man betrachte den Beziehungstyp ARBEITET_FÜR in Abbildung 3.9. Hier kann man sich ein Attribut namens Abteilung des Entitätstyps ANGESTELLTER vorstellen, dessen Wert für jede Angestelltenentität (eine Referenz auf) die *Abteilungsentität* ist, für die der Angestellte arbeitet. Folglich ist die Wertemenge für dieses Attribut Abteilung die Menge *aller Entitäten* ABTEILUNG. Genau das haben wir in Abbildung 3.8 realisiert, als wir den anfänglichen Entwurf des Entitätstyps ANGESTELLTER für die Datenbank FIRMA spezifizier-

Abbildung 3.10: Beziehungsinstanzen der ternären Beziehung LIEFERN.



ten. Wenn wir uns aber eine binäre Beziehung als Attribut vorstellen, haben wir immer zwei Möglichkeiten. In diesem Beispiel ist die Alternative ein mehrwertiges Attribut Angestellter des Entitätstyps ABTEILUNG, dessen Werte für jede Abteilungsentität die *Menge der Angestelltenentitäten* bilden, die für diese Abteilung arbeiten. Die Wertemenge des Attributs Angestellter ist die Entitätsmenge ANGESTELLTER. Jedes dieser beiden Attribute, d.h. Abteilung von ANGESTELLTER oder Angestellter von ABTEILUNG, kann den Beziehungstyp ARBEITET_FÜR darstellen. Wenn beide dargestellt werden, muss man sie dahingehend einschränken, dass sie jeweils die Umkehr des anderen sind.[7]

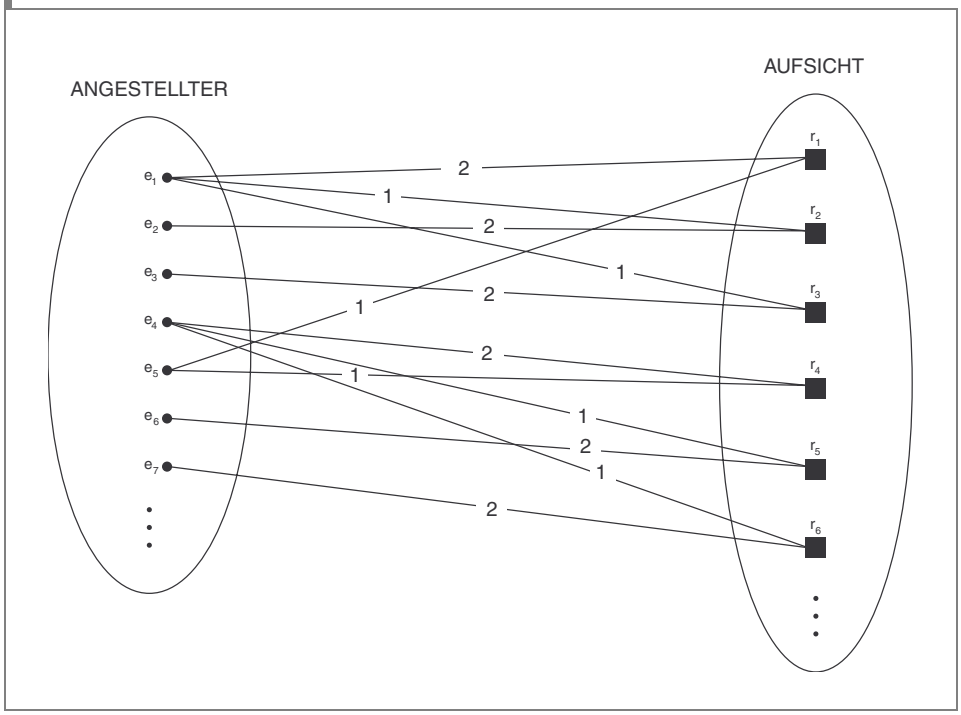
Rollennamen und rekursive Beziehungen Jeder Entitätstyp, der an einem Beziehungstyp teilnimmt, spielt in der Beziehung eine bestimmte **Rolle**. Der **Rollenname** bezeichnet diese Rolle, die eine teilnehmende Entität des Entitätstyps in jeder Beziehungsinstanz spielt, und vereinfacht die Erklärung, was die Beziehung bedeutet. Beispielsweise spielt ANGESTELLTER im Beziehungstyp ARBEITET_FÜR die Rolle eines *Angestellten* oder *Arbeitnehmers*, während ABTEILUNG die Rolle einer *Abteilung* oder eines *Arbeitgebers* übernimmt.

Rollennamen sind technisch in Beziehungstypen mit teilnehmenden Entitätstypen, die sich alle unterscheiden, nicht erforderlich, weil in diesem Fall jeder Entitäts-

7. Dieses Konzept der Darstellung von Beziehungstypen als Attribute wird in einer Datenmodellklasse angewandt, die als **funktionale Datenmodelle** bezeichnet werden. In Objektdatenbanken (siehe Kapitel 11 und Kapitel 12 der ungekürzten Ausgabe auf der CWS) können Beziehungen durch Referenzattribute – entweder in einer oder in beiden Richtungen – als Umkehr dargestellt werden. In relationalen Datenbanken (siehe Kapitel 5 und 6) werden Sekundärschlüssel als eine Art Referenzattribut für die Darstellung von Beziehungen verwendet.

typname als Rollenname verwendet werden kann. In manchen Fällen übt aber der *gleiche* Entitätstyp in einem Beziehungstyp mehr als einmal *unterschiedliche Rollen* aus. In solchen Fällen ist der Rollenname für die Unterscheidung der Bedeutung jeder Teilnahme notwendig. Solche Beziehungstypen werden als **rekursive Beziehungen** bezeichnet. Abbildung 3.11 zeigt ein Beispiel. Der Beziehungstyp AUFSICHT stellt einen Angestellten mit einem Manager in Beziehung, wobei sowohl die Angestellten- als auch die Managerentität Mitglieder des gleichen Entitätstyps ANGESTELLTER sind. Folglich nimmt der Entitätstyp ANGESTELLTER zweimal an AUFSICHT teil: einmal in der Rolle des Vorgesetzten und ein zweites Mal in der Rolle des Untergebenen. Jede Beziehungsinstanz r_i in AUFSICHT assoziiert zwei Angestelltenentitäten e_j und e_k, von denen eine die Rolle des Vorgesetzten und die andere die des Untergebenen übernimmt. In Abbildung 3.11 stellen die mit »1« gekennzeichneten Linien die Vorgesetztenrolle und die mit »2« gekennzeichneten die Untergebenenrolle dar. Hier gilt also: e_1 beaufsichtigt e_2 und e_3; e_4 beaufsichtigt e_6 und e_7; und e_5 beaufsichtigt e_1 und e_4.

Abbildung 3.11: Die rekursive Beziehung AUFSICHT, wobei der Entitätstyp ANGESTELLTER die beiden Rollen Vorgesetzter (1) und Untergebener (2) spielt.



3.4.3 Einschränkungen für Beziehungstypen

Für Beziehungstypen gelten normalerweise bestimmte Einschränkungen, mit denen die möglichen Kombinationen von Entitäten, die an der entsprechenden Beziehungsmenge teilnehmen dürfen, begrenzt werden. Diese Einschränkungen werden durch die Miniwelt-Situation vorgegeben, die von den Beziehungen dargestellt wird. In Abbildung 3.9 kann für die Firma z.B. eine Regel gelten, die besagt, dass jeder Ange-

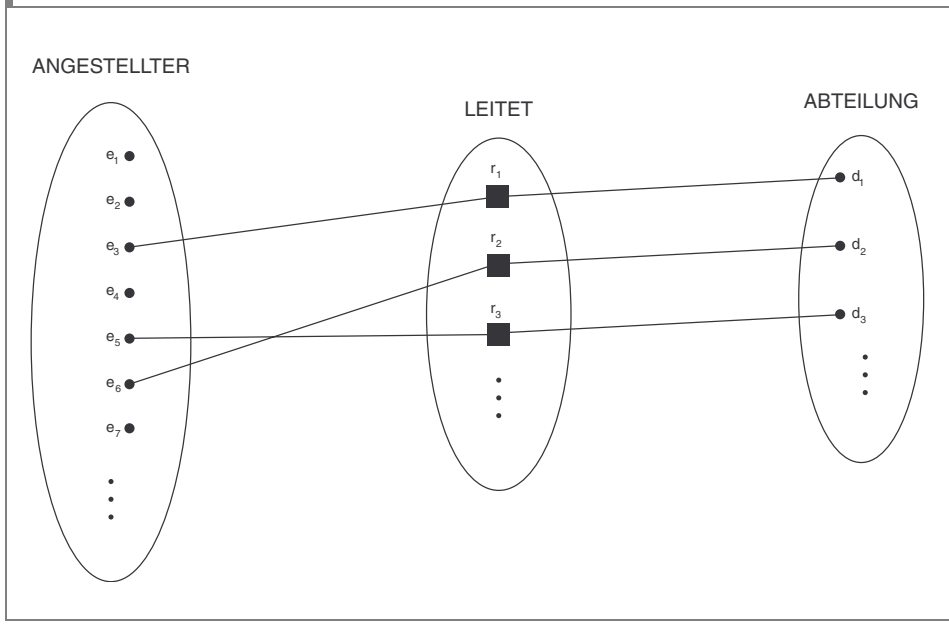
stellte für genau eine Abteilung arbeiten muss; dann müsste man diese Einschränkung im Schema beschreiben. Wir können Einschränkungen für Beziehungen in zwei Haupttypen unterteilen: Angaben zu *Kardinalitätsverhältnissen* und *Teilnahmebedingungen*.

Kardinalitätsverhältnisse für binäre Beziehungen Das Kardinalitätsverhältnis für eine binäre Beziehung spezifiziert die Anzahl der Beziehungsinstanzen, an denen eine Entität teilnehmen kann. Bei dem binären Beziehungstyp ARBEITET_FÜR hat z.B. ABTEILUNG:ANGESTELLTER ein Kardinalitätsverhältnis von 1:N. Dies bedeutet, dass sich jede Abteilung auf zahlreiche Angestellten beziehen (d.h. diese beschäftigen) kann, ein Angestellter aber nur für eine Abteilung arbeiten kann.[8] Die möglichen Kardinalitätsverhältnisse für binäre Beziehungstypen sind 1:1, 1:N, N:1 und M:N.

Ein Beispiel einer binären 1:1-Beziehung ist LEITET (Abbildung 3.12) zwischen einer Abteilungsentität und dem Angestellten, der diese Abteilung leitet. Dies entspricht den Miniwelt-Einschränkungen, dass ein Angestellter nur eine Abteilung leiten und eine Abteilung nur einen Abteilungsleiter haben kann. Der Beziehungstyp ARBEITET_AN (Abbildung 3.13) hat ein Kardinalitätsverhältnis von M:N, weil die Miniwelt-Regel lautet, dass ein Angestellter an mehreren Projekten arbeiten und ein Projekt von mehreren Angestellten bearbeitet werden kann.

Kardinalitätsverhältnisse für binäre Beziehungen werden in ER-Diagrammen durch 1, M und N auf Rauten dargestellt (siehe Abbildung 3.2).

Abbildung 3.12: Die 1:1-Beziehung LEITET, mit partieller Teilnahme von ANGESTELLTER und totaler Teilnahme von ABTEILUNG.



8. N bedeutet eine *beliebige Zahl* zusammenhängender Entitäten (keine oder mehrere).

Teilnahmeeinschränkungen und Existenzabhängigkeiten Eine **Teilnahmeeinschränkung** spezifiziert, ob eine Entität über einen gegebenen Beziehungstyp mit einer anderen Entität in Bezug stehen muss oder nicht. Entsprechend gibt es zwei Arten von Teilnahmeeinschränkungen: totale und partielle; wir verdeutlichen dies an einem Beispiel. Wenn die Regeln einer Firma besagen, dass *jeder* Angestellte für eine Abteilung arbeiten muss, dann kann eine Angestelltenentität nur bestehen, wenn sie an einer Beziehungsinstanz ARBEITET_FÜR teilnimmt (Abbildung 3.9). Folglich ist die Teilnahme von ANGESTELLTER an ARBEITET_FÜR eine **totale Teilnahme**, was bedeutet, dass jede Entität der »gesamten Menge« von Angestelltenentitäten über ARBEITET_FÜR mit einer Abteilungsentität in Bezug stehen muss. Eine totale Teilnahme nennt man auch **Existenzabhängigkeit**. In Abbildung 3.12 erwarten wir nicht, dass jeder Angestellte eine Abteilung leitet, so dass ANGESTELLTER im Beziehungstyp LEITET nur partiell teilnimmt, was bedeutet, dass ein »Teil der Menge« der Angestelltenentitäten, aber nicht unbedingt alle, über LEITET mit einer Abteilungsentität in Bezug stehen. Das Kardinalitätsverhältnis und die Teilnahmeeinschränkungen werden insgesamt als **strukturelle Einschränkungen** eines Beziehungstyps bezeichnet.

In ER-Diagrammen wird eine totale Teilnahme als *doppelte Linie*, die den teilnehmenden Entitätstyp mit der Beziehung verbindet, und die partielle Teilnahme durch eine *einfache Linie* dargestellt (siehe Abbildung 3.2).

3.4.4 Attribute von Beziehungstypen

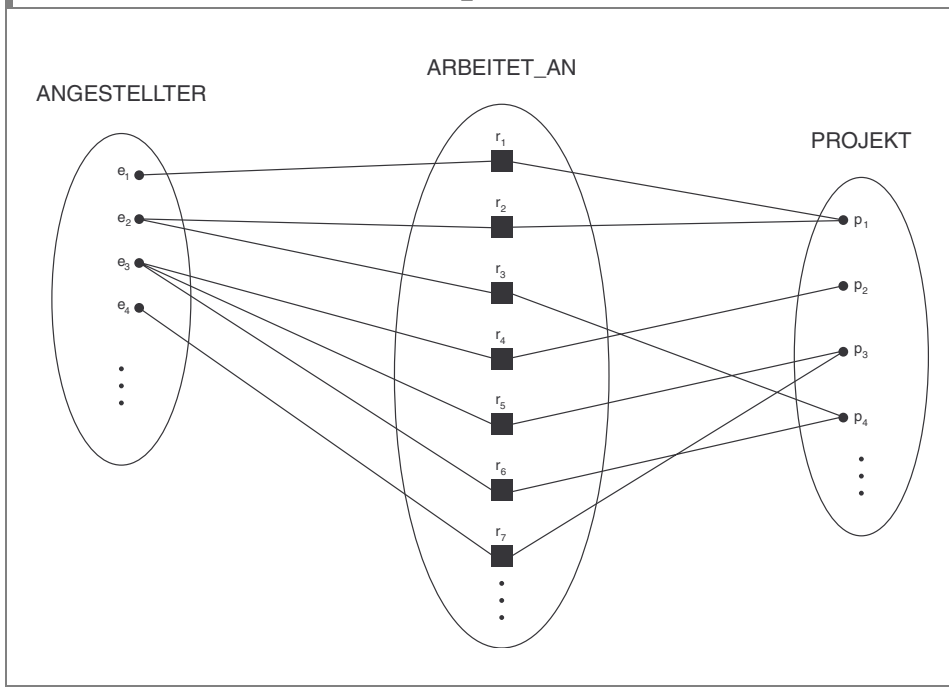
Genauso wie Entitätstypen können auch Beziehungstypen durch Attribute charakterisiert werden. Um beispielsweise die Stundenzahl pro Woche zu erfassen, die ein Angestellter an einem bestimmten Projekt arbeitet, können wir das Attribut Stunden für den Beziehungstyp ARBEITET_AN von Abbildung 3.13 einbeziehen. Ein weiteres Beispiel wäre die Verwendung des Datums, an dem ein Manager zum Abteilungsleiter wurde, durch ein Attribut namens Anfangsdatum für den Beziehungstyp LEITET von Abbildung 3.12.

Man beachte, dass Attribute von 1:1- oder 1:N-Beziehungstypen auf einen der teilnehmenden Entitätstypen übertragen werden können. Beispielsweise kann das Attribut Anfangsdatum für die Beziehung LEITET ein Attribut von ANGESTELLTER oder ABTEILUNG sein, obwohl es konzeptuell zu LEITET gehört. Der Grund ist, dass es sich bei LEITET um eine 1:1-Beziehung handelt, so dass jede Abteilungs- oder Angestelltenentität an *höchstens* einer Beziehungsinstanz teilnimmt. Folglich kann der Wert des Attributs Anfangsdatum getrennt durch die teilnehmende Abteilungsentität *oder* die teilnehmende Angestelltenentität (Manager) bestimmt werden.

Bei einem 1:N-Beziehungstyp lässt sich ein Beziehungsattribut *nur* auf den Entitätstyp auf der N-Seite der Beziehung übertragen. Wenn die Beziehung ARBEITET_FÜR in Abbildung 3.9 z.B. auch ein Attribut Anfangsdatum hat, also das Datum, an dem ein Angestellter seine Arbeit für eine Abteilung begonnen hat, dann kann dieses Attribut als Attribut von ANGESTELLTER berücksichtigt werden. Der Grund ist, dass jede Angestelltenentität an höchstens einer Beziehungsinstanz in ARBEITET_FÜR beteiligt ist. Bei den beiden Beziehungstypen 1:1 und 1:N wird die Entscheidung, wie man ein Beziehungsattribut platzieren soll – als Attribut für einen Beziehungstyp oder für einen teilnehmenden Entitätstyp –, subjektiv vom Datenbankdesigner getroffen.

Bei M:N-Beziehungstypen werden Attribute der Beziehung durch die *Kombination teilnehmender Entitäten* an einer Beziehungsinstanz statt durch eine einzelne Entität bestimmt. Solche Attribute *müssen als Beziehungsattribute spezifiziert werden*. Ein Bei-

Abbildung 3.13: Die M:N-Beziehung ARBEITET_AN zwischen ANGESTELLTER und PROJEKT.



spiel ist das Attribut Stunden der M:N-Beziehung ARBEITET_AN (Abbildung 3.13); die Anzahl der Stunden, die ein Angestellter an einem Projekt arbeitet, wird durch eine Angestellter/Projekt-Kombination und nicht getrennt durch eine der beiden Entitäten bestimmt.

3.5 Schwache Entitätstypen

Entitätstypen, die noch keine eigenen Schlüsselattribute haben, werden als **schwache Entitätstypen** bezeichnet. Entsprechend werden normale Entitätstypen, die ein Schlüsselattribut besitzen, als **starke Entitätstypen** bezeichnet. Entitäten, die zu einem schwachen Entitätstyp gehören, werden dadurch identifiziert, dass sie mit spezifischen Entitäten von einem anderen Entitätstyp in Kombination mit einem ihrer Attributwerte in Beziehung gesetzt werden. Wir nennen diesen anderen Entitätstyp den **identifizierenden** oder **Eigentümer-Entitätstyp**[9] **und der Beziehungstyp, der mit einem schwachen Entitätstyp in Bezug steht, ist der Eigentümer der identifizierenden Beziehung** des schwachen Entitätstyps.[10] Ein schwacher Entitätstyp hat immer eine *totale Teilnahmeeinschränkung* (Existenzabhängigkeit) hinsichtlich seiner identifizierenden Beziehung, weil eine schwache Entität nicht ohne Eigentümerentität identifiziert werden kann. Allerdings führt nicht jede Existenzabhängigkeit zu einem schwachen Entitätstyp. Beispielsweise kann eine Entität FÜHRERSCHEIN nicht ohne Bezie-

9. Der identifizierende Entitätstyp wird auch **Vater-Entitätstyp** oder **dominanter Entitätstyp** genannt.
10. Der schwache Entitätstyp wird auch als **Sohn-Entitätstyp** oder **untergeordneter Entitätstyp** bezeichnet.

hung zu einer Entität PERSON existieren, obwohl sie einen eigenen Schlüssel (FahrerlaubnisNummer) hat und somit keine schwache Entität ist.

Man betrachte den Entitätstyp ANGEHÖRIGER mit einer Beziehung zu ANGESTELLTER, mit der die Familienangehörigen jedes Angestellten über eine 1:N-Beziehung verwaltet werden (Abbildung 3.2). Die Attribute von ANGEHÖRIGER sind Name (Vorname des Familienmitglieds), Geburtsdatum, Geschlecht und Grad (Verwandtschaftsgrad zum Angestellten). Zwei Angehörige *zweier verschiedener Angestellter* können zufällig die gleichen Werte für Name, Geburtsdatum, Geschlecht und Grad haben, dennoch aber unterschiedliche Entitäten darstellen. Sie werden erst als unterschiedliche Entitäten identifiziert, nachdem die *jeweilige Angestelltenentität*, zu der die beiden Angehörigen jeweils gehören, festgestellt wurde. Jede Angestelltenentität **besitzt** also die Angehörigenentitäten, zu denen eine Beziehung besteht.

Ein schwacher Entitätstyp hat normalerweise einen **partiellen Schlüssel**, der die Menge der Attribute darstellt, die schwache Entitäten eindeutig identifizieren können, die mit der *gleichen Eigentümer-Entität in Bezug stehen*.[11] Wenn wir in unserem Beispiel annehmen, dass nie zwei Angehörige des gleichen Angestellten den gleichen Vornamen haben, wäre das Attribut Name von ANGEHÖRIGER der partielle Schlüssel. Im schlechtesten Fall ist ein zusammengesetztes Attribut *aller Attribute der schwachen Entität* der partielle Schlüssel.

In ER-Diagrammen unterscheidet man einen schwachen Entitätstyp und seine identifizierende Beziehung durch *doppelte Linien* um die Kästen und Rauten (siehe Abbildung 3.2). Das partielle Schlüsselattribut wird mit einer gestrichelten oder gepunkteten Linie unterstrichen.

Schwache Entitätstypen können manchmal als komplexe (zusammengesetzte mehrwertige) Attribute dargestellt werden. Im vorherigen Beispiel könnten wir ein mehrwertiges Attribut Angehörige für ANGESTELLTER spezifizieren, bei dem es sich um ein zusammengesetztes Attribut mit den Komponentenattributen Name, Geburtsdatum, Geschlecht und Grad handelt. Die Wahl der Darstellung wird vom Datenbankdesigner getroffen. Ein Kriterium, das für diese Wahl herangezogen werden kann, ist die Darstellung des schwachen Entitätstyps, wenn es viele Attribute gibt. Nimmt die schwache Entität unabhängig an anderen Beziehungstypen als an ihrem identifizierenden Beziehungstyp teil, sollte sie *nicht* als komplexes Attribut modelliert werden.

Im Allgemeinen können schwache Entitätstypen in jeder beliebigen Anzahl von Ebenen definiert werden. Ein Eigentümer-Entitätstyp kann selbst ein schwacher Entitätstyp sein. Zusätzlich kann ein schwacher Entitätstyp mehr als einen identifizierenden Entitätstyp und ein identifizierender Beziehungstyp einen Grad von höher als Zwei haben, wie in Kapitel 4 beschrieben wird.

3.6 Verfeinerung des ER-Entwurfs der Datenbank FIRMA

Wir können jetzt den Datenbankentwurf von Abbildung 3.8 dadurch verfeinern, dass wir die Attribute, die Beziehungen darstellen, in Beziehungstypen abändern. Das Kardinalitätsverhältnis und die Teilnahmeeinschränkung jedes Beziehungstyps werden aus den in Abschnitt 3.2 aufgeführten Anforderungen ermittelt. Lässt sich ein Kardinalitätsverhältnis oder eine Abhängigkeit nicht aus den Anforderungen bestimmen, müssen die Benutzer nochmals befragt werden, um diese strukturellen Einschränkun-

11. Der partielle Schlüssel wird auch als **Diskriminator** bezeichnet.

gen festzulegen. In unserem Beispiel können wir folgende Beziehungstypen spezifizieren:

1. LEITET – ein 1:1-Beziehungstyp zwischen ANGESTELLTER und ABTEILUNG. Die Teilnahme ANGESTELLTER ist partiell. Die Teilnahme an ABTEILUNG geht nicht klar aus den Anforderungen hervor. Wir befragen die Benutzer, von denen wir erfahren, dass eine Abteilung ständig einen Abteilungsleiter braucht, was auf eine totale Teilnahme hindeutet.[12] Dieser Beziehungstyp wird durch das Attribut Anfangsdatum genauer beschrieben.
2. ARBEITET_FÜR – ein 1:N-Beziehungstyp zwischen ABTEILUNG und ANGESTELLTER. Beide Teilnahmen sind total.
3. KONTROLLIERT – ein 1:N-Beziehungstyp zwischen ABTEILUNG und PROJEKT. Die Teilnahme von PROJEKT ist total, während diejenige von ABTEILUNG partiell ist, wie sich nach Rücksprache mit den Benutzern herausgestellt hat.
4. AUFSICHT – ein 1:N-Beziehungstyp zwischen ANGESTELLTER (in der Vorgesetztenrolle) und ANGESTELLTER (in der Untergebenenrolle). Beide Teilnahmen sind partiell, da die Benutzer darauf hingewiesen haben, dass nicht jeder Angestellte ein Vorgesetzter ist und nicht jeder Angestellte einen Vorgesetzten hat.
5. ARBEITET_AN – wurde als M:N-Beziehungstyp mit dem Attribut Stunden festgelegt, nachdem die Benutzer darauf hingewiesen haben, dass mehrere Angestellte an einem Projekt arbeiten können. Beide Teilnahmen sind total.
6. ANGEHÖRIGER_VON – ein 1:N-Beziehungstyp zwischen ANGESTELLTER und ANGEHÖRIGER, der auch die identifizierende Beziehung für den schwachen Entitätstyp ANGEHÖRIGER ist. Die Teilnahme von ANGESTELLTER ist partiell, während die von ANGEHÖRIGER total ist.

Nach der Spezifikation der obigen sechs Beziehungstypen entfernen wir von den Entitätstypen in Abbildung 3.8 alle Attribute, die in Beziehungen verfeinert wurden. Dazu zählen Manager und ManagerAnfangsdatum aus ABTEILUNG; KontrollierendeAbteilung aus PROJEKT; Abteilung, Manager und ArbeitetAn aus ANGESTELLTER; und Angestellter aus ANGEHÖRIGER. Wichtig ist die Sicherstellung, dass das konzeptuelle Schema einer Datenbank die geringst mögliche Redundanz aufweist. Ist eine gewisse Redundanz auf der Speicherebene oder auf der Ebene der Benutzersichten (Views) erwünscht, kann sie später eingeführt werden, wie in Abschnitt 1.6.1 beschrieben wurde.

3.7 ER-Diagramme, Namenskonventionen und Entwurfsfragen

3.7.1 Übersicht der Notation für ER-Diagramme

Abbildungen 3.9 bis 3.13 illustrieren die Entitäts- und Beziehungstypen durch Darstellung der jeweiligen Erweiterungen – die einzelnen Entitäten und Beziehungsinstanzen. In ER-Diagrammen liegt der Schwerpunkt ausschließlich auf der Darstellung

12. Die Regeln in der Miniwelt, die die Einschränkungen bestimmen, werden auch als *Geschäftsregeln* bezeichnet, weil sie vom »Geschäft« oder Unternehmen bestimmt werden, das die Datenbank nutzt.

der Schemas und nicht auf der von Instanzen. Dies ist nützlicher, weil sich ein Datenbankschema selten ändert, während sich die Instanz häufig ändert.

Abbildung 3.2 stellt das **ER-Datenbankschema** FIRMA als ER-Diagramm dar. Wir gehen im Folgenden die gesamte Notation von ER-Diagrammen durch. Entitätstypen wie ANGESTELLTER, ABTEILUNG und PROJEKT werden in Rechtecken dargestellt. Beziehungstypen wie ARBEITET_FÜR, LEITET, KONTROLLIERT und ARBEITET_AN werden in Rauten dargestellt, die durch gerade Linien mit den teilnehmenden Entitätstypen verbunden sind. Attribute werden in Ovalen dargestellt und jedes Attribut ist durch eine gerade Linie mit seinem Entitäts- oder Beziehungstyp verbunden. Die Komponentenattribute eines zusammengesetzten Attributs werden mit dem Oval verbunden, das das zusammengesetzte Attribut darstellt, z.B. das Attribut Name von ANGESTELLTER. Mehrwertige Attribute werden als doppelte Ovale dargestellt, wie z.B. das Attribut Standorte von ABTEILUNG. Bei Schlüsselattributen ist der Name unterstrichen. Abgeleitete Attribute werden in gepunkteten Ovalen dargestellt, z.B. das Attribut AnzahlAngestellte von ABTEILUNG.

Schwache Entitätstypen werden dadurch unterschieden, dass man sie in doppelte Rechtecke und ihre identifizierende Beziehung in doppelte Rauten setzt, wie z.B. der Entitätstyp ANGEHÖRIGER und der identifizierende Beziehungstyp ANGEHÖRIGE_VON. Der partielle Schlüssel des schwachen Entitätstyps wird mit einer gepunkteten Linie unterstrichen.

In Abbildung 3.2 wird das Kardinalitätsverhältnis jedes *binären* Beziehungstyps durch Anfügen von 1, M oder N auf jeder teilnehmenden Kante spezifiziert. Das Kardinalitätsverhältnis von ABTEILUNG:ANGESTELLTER in LEITET ist 1:1, während das von ABTEILUNG:ANGESTELLTER in ARBEITET_FÜR 1:N und für ARBEITET_AN M:N ist. Die Teilnahmeeinschränkung wird durch eine einzelne Linie für die partielle Teilnahme bzw. durch eine doppelte Linie für die totale Teilnahme (Existenzabhängigkeit) spezifiziert.

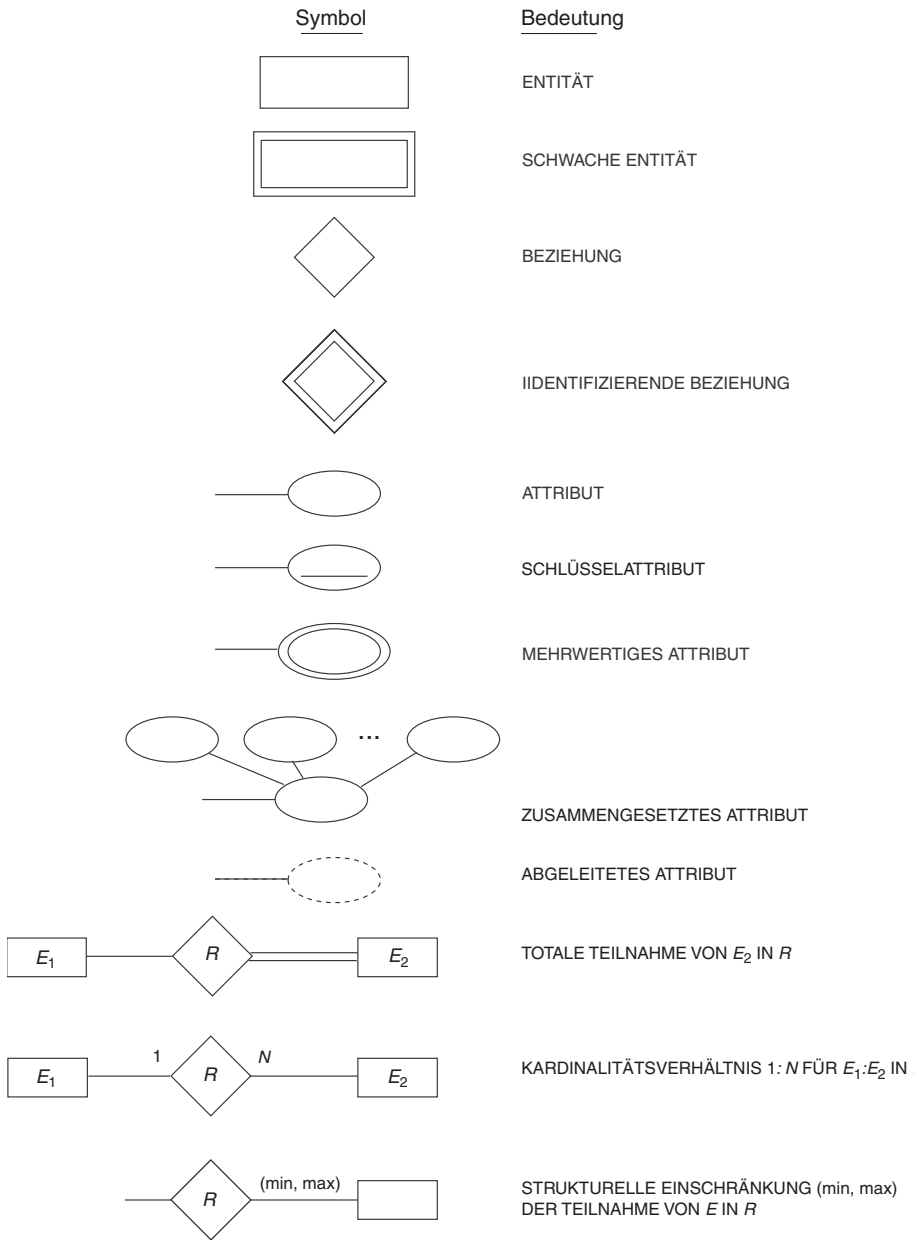
In Abbildung 3.2 sind die Rollennamen für den Beziehungstyp AUFSICHT enthalten, weil der Entitätstyp ANGESTELLTER in dieser Beziehung beide Rollen spielt. Man beachte, dass die Kardinalität vom Vorgesetzten zum Untergebenen 1:N ist, weil einerseits jeder Angestellte in der Rolle des Untergebenen höchstens einen direkten Vorgesetzten, in der Rolle des Vorgesetzten aber keinen oder mehrere Angestellte unter sich haben kann. Abbildung 3.14 zeigt eine Übersicht der Konventionen für ER-Diagramme.

3.7.2 Korrekte Benennung von Schema-Konstrukten

Die Wahl der Namen für Entitätstypen, Attribute, Beziehungstypen und (insbesondere) Rollen ist nicht immer leicht. Man sollte auf jeden Fall Namen wählen, die möglichst die Bedeutung der verschiedenen Konstrukte im Schema vermitteln. Wir können für Entitätstypen Namen im *Singular* statt im Plural wählen, weil der Name des Entitätstyps für jede ihm zugehörige einzelne Entität gilt. In unseren ER-Diagrammen verwenden wir die Konvention, dass Namen von Entitätstypen und Beziehungstypen in Großbuchstaben, Attributnamen mit großem Anfangsbuchstaben und Rollennamen klein geschrieben werden. Diese Konvention kam bereits in Abbildung 3.2 zum Einsatz.

In einer textlichen Beschreibung der Datenbankanforderungen ist es üblich, aus den in der Beschreibung enthaltenen *Substantiven* Entitätstypnamen und aus den *Verben* Namen für Beziehungstypen herzuleiten. Attributnamen werden im Allgemeinen aus zusätzlichen Substantiven gebildet, die die Substantive in Bezug auf Entitätstypen beschreiben.

Abbildung 3.14: Übersicht über die Notation von ER-Diagrammen.



Ein weiterer Benennungsfaktor ist die Auswahl von Beziehungsnamen in Bezug auf Lesensart, d.h., ob sich das ER-Diagramm des Schemas von links nach rechts und von oben nach unten lesen lässt. Wir sind dieser Richtlinie in Abbildung 3.2 im Allgemeinen gefolgt. Eine Ausnahme bildet der Beziehungstyp ANGEHÖRIGE_VON, der sich von unten nach oben liest. Das hat folgenden Grund: Wir sagen, dass die Entitäten ANGEHÖRIGER (unterer Entitätstyp) ANGEHÖRIGE_VON (Beziehungsname) eines ANGESTELLTEN (oberer Entitätstyp) sind. Um dies auf die Lesart von oben nach unten zu ändern, könnten wir den Beziehungstyp in HAT_ANGEHÖRIGE ändern, was wie folgt zu lesen wäre: Eine Entität ANGESTELLTER (oberer Entitätstyp) HAT_ANGEHÖRIGE (Beziehungsname) vom Typ ANGEHÖRIGER (unterer Entitätstyp).

3.7.3 Entwurfsalternativen für den konzeptuellen ER-Entwurf

Gelegentlich ist die Feststellung schwierig, ob ein bestimmtes Konzept in der Miniwelt als Entitätstyp, Attribut oder Beziehungstyp modelliert werden soll. Dieser Abschnitt enthält kurze Richtlinien darüber, welches Konstrukt in bestimmten Situationen gewählt werden sollte.

Im Allgemeinen sollte der Schemaentwurfsprozess als iterativer Verfeinerungsprozess betrachtet werden, bei dem zuerst ein grober Entwurf erstellt und dann schrittweise verfeinert wird, bis ein zufriedenstellender Entwurf erreicht wird. Häufige Verfeinerungen können beispielsweise wie folgt charakterisiert werden.

1. Ein Konzept kann zunächst als Attribut modelliert und dann auf eine Beziehung verfeinert werden, wenn man feststellt, dass es sich bei dem Attribut um eine Referenz auf einen anderen Entitätstyp handelt. Oftmals stellt man fest, dass ein Paar solcher Attribute, die die Umkehr des jeweils anderen sind, auf eine binäre Beziehung verfeinert wird. Diese Art der Verfeinerung wurde ausführlich in Abschnitt 3.6 behandelt.

2. Ein Attribut, das in mehreren Entitätstypen vorkommt, kann auf seinen eigenen unabhängigen Entitätstyp verfeinert werden. Wenn wir beispielsweise annehmen, dass mehrere Entitätstypen der Beispieldatenbank UNIVERSITY, wie z.B. STUDENT, INSTRUCTOR und COURSE, im Grobentwurf jeweils ein Attribut Department haben, kann der Designer einen Entitätstyp DEPARTMENT mit einem einzigen Attribut DeptName erstellen und es mit den drei Entitätstypen (STUDENT, INSTRUCTOR und COURSE) über entsprechende Beziehungen verknüpfen. Möglicherweise werden später weitere Attribute/Beziehungen für DEPARTMENT festgestellt.

3. Alternativ ist eine umgekehrte Verfeinerung des vorherigen Falls möglich. Wenn ein Entitätstyp DEPARTMENT im Grobentwurf beispielsweise ein einziges Attribut DeptName und nur eine Beziehung zu einem Entitätstyp STUDENT hat, kann man DEPARTMENT auf ein Attribut von STUDENT verfeinern.

4. In Kapitel 4 werden weitere Verfeinerungen in Zusammenhang mit Spezialisierung, Generalisierung und Beziehungen eines höheren Grads beschrieben.

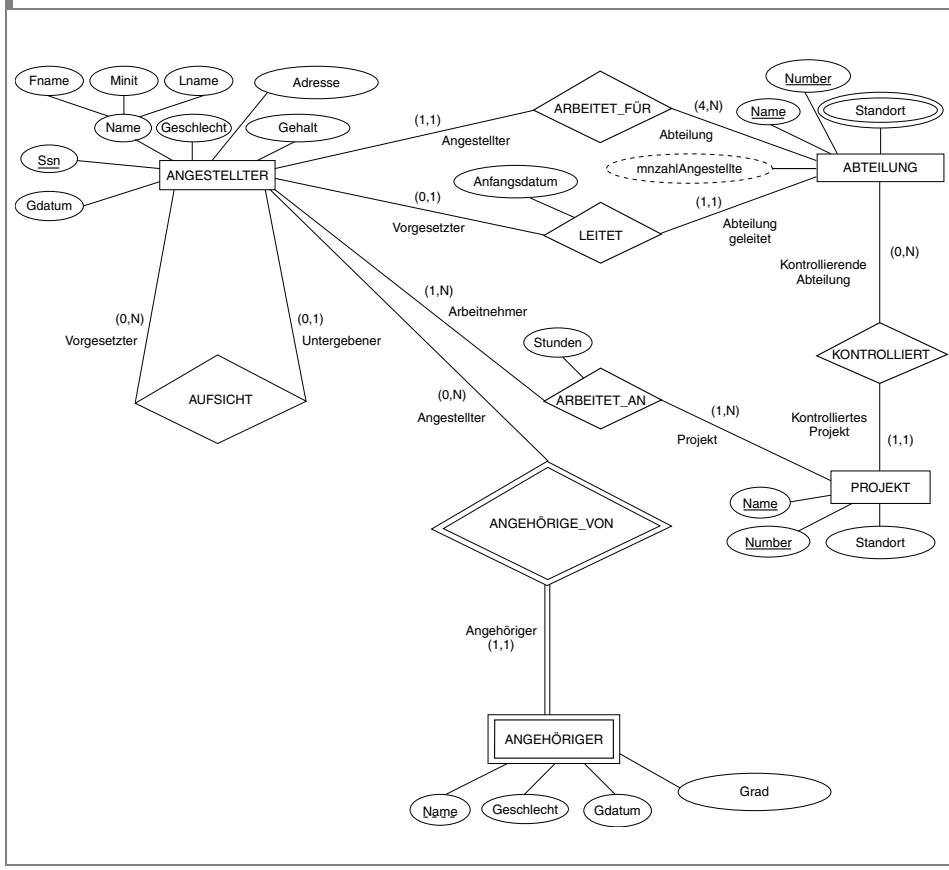
3.7.4 Weitere Notationen für ER-Diagramme

Für die Darstellung von ER-Diagrammen stehen viele alternative Notationen zur Verfügung, von denen die beliebtesten in Anhang A aufgeführt werden. In Kapitel 4 wird außerdem die UML-Notation (Universal Modeling Language) eingeführt, die als Standard für die konzeptuelle Objektmodellierung vorgeschlagen wurde.

In diesem Abschnitt wird eine alternative ER-Notation für die Spezifikation struktureller Einschränkungen für Beziehungen beschrieben. Bei dieser Notation wird ein Ganzzahlenpaar (min, max) mit jeder Teilnahme eines Entitätstyps *E* in einem Beziehungstyp *R* assoziiert, wobei $0 \leq min \leq max$ und $max \geq 1$ ist. Die Zahlen bedeuten, dass *e zu einem Zeitpunkt* für jede Entität *e* in *E* mindestens in *min* und höchstens in *max* Beziehungsinstanzen in *R* teilnehmen muss. Diese Notation bedeutet, dass $min = 0$ eine partielle und $min > 0$ eine totale Teilnahme darstellt.

Abbildung 3.15 zeigt das Datenbankschema FIRMA in der min, max-Notation.[13] Normalerweise benutzt man entweder die Notation Kardinalitätsverhältnis/einfache Linie/doppelte Linie *oder* die min/max-Notation. Die min/max-Notation ist präziser; sie eignet sich gut für die Spezifikation struktureller Einschränkungen für Beziehungstypen eines *beliebigen Grads*.

Abbildung 3.15: ER-Diagramm für das Schema FIRMA mit allen Rollennamen und strukturellen Einschränkungen für Beziehungen in der alternativen Notation (min, max).



13. In manchen Notationen, insbesondere denjenigen für Objektmodellierung, werden (min, max) auf den *entgegengesetzten Seiten* wie in unserem Beispiel dargestellt. Für die Beziehung ARBEITET_FÜR in Abbildung 3.15 befände sich (1,1) z.B. auf der Seite ABTEILUNG und (4,N) auf der Seite ANGESTELLTER. Wir verwenden die Originalnotation von Abrial (1974).

Andererseits reicht sie für die Spezifikation einiger Schlüsseleinschränkungen für Beziehungen mit höherem Grad nicht aus, wie in Kapitel 4 noch beschrieben wird. In Abbildung 3.15 sind auch alle Rollennamen für das Datenbankschema FIRMA dargestellt.

3.8 Zusammenfassung

In diesem Kapitel wurden die Modellierungskonzepte eines hohen konzeptuellen Datenmodells – des ER-Modells (Entity-Relationship Model) – dargestellt. Zuerst wurde die Rolle beschrieben, die ein logisches Datenmodell im Datenbankentwurfsprozess spielt. Anschließend wurde ein Beispiel von Datenbankanforderungen für die Datenbank FIRMA aufgeführt. Dieses Beispiel wird in allen Kapiteln immer wieder herangezogen, um grundlegende Konzepte des ER-Modells für Entitäten und Attribute zu beschreiben. Ferner wurden Nullwerte und die verschiedenen Attributtypen vorgestellt, die beliebig verschachtelt werden können, um komplexe Attribute zu bilden:

- Einfach oder atomar
- Zusammengesetzt
- Mehrwertig

Außerdem wurden gespeicherte gegenüber abgeleiteten Attributen beschrieben. Anschließend wurden die ER-Modellkonzepte auf der Schema- bzw. Intensionsebene behandelt:

- Entitätstypen und ihre entsprechenden Entitätsmengen
- Schlüsselattribute von Entitätstypen
- Wertemengen (Bereiche) von Attributen
- Beziehungstypen und ihre entsprechenden Beziehungsmengen
- Rollen von Entitätstypen an der Teilnahme in Beziehungstypen

Schließlich wurden zwei Methoden für die Spezifikation der strukturellen Einschränkungen für Beziehungstypen vorgestellt. Die erste Methode ließ sich in zwei Arten struktureller Einschränkungen unterteilen:

- Kardinalitätsverhältnisse (1:1, 1:N, M:N für binäre Beziehungen)
- Teilnahmeeinschränkungen (total, partiell)

Wir stellen fest, dass bei einer alternativen Methode für die Spezifikation struktureller Einschränkungen die Minimum- und Maximumzahlen (min, max) der Teilnahme jedes Entitätstyps an einem Beziehungstyp spezifiziert werden. Ferner wurden schwache Entitätstypen und die damit verbundenen Konzepte von Eigentümer-Entitätstypen, identifizierenden Beziehungstypen und partiellen Schlüsselattributen beschrieben.

ER-Schemas können in Form von ER-Diagrammen dargestellt werden. Wir haben gezeigt, dass für den Entwurf eines ER-Schemas für die Datenbank FIRMA zuerst die Entitätstypen und ihre Attribute definiert werden und der Entwurf anschließend unter Einbeziehung von Beziehungstypen verfeinert wird. Wir haben als Beispiel das ER-Diagramm für das Datenbankschema FIRMA dargestellt.

Die bisher behandelten ER-Modellierungskonzepte, d.h. Entitätstypen, Beziehungstypen, Attribute, Schlüssel und strukturelle Einschränkungen, können traditionelle geschäftsorientierte Datenbankanwendungen modellieren. Viele neuere, komplexere Anwendungen, wie z.B. Konstruktionsentwürfe, medizinische Informationssysteme oder Telekommunikation, setzen zusätzliche Konzepte voraus, um sie genauer modellieren zu können. Diese modernen Modellierungskonzepte werden in Kapitel 4 beschrieben. Außerdem werden in Kapitel 4 ternäre und höhere Beziehungstypen ausführlicher behandelt, ebenso wie die Umstände, unter denen sie von binären Beziehungen unterschieden werden.

WIEDERHOLUNGSFRAGEN

1. Diskutieren Sie die Rolle eines logischen Datenmodells im Datenbankentwurfsprozess.
2. Führen Sie die verschiedenen Fälle auf, in denen sich die Verwendung eines Nullwerts eignen würde.
3. Definieren Sie die folgenden Begriffe: *Entität*, *Attribut*, *Attributwert*, *Beziehungsinstanz*, *zusammengesetztes Attribut*, *mehrwertiges Attribut*, *abgeleitetes Attribut*, *komplexes Attribut*, *Schlüsselattribut*, *Wertemenge* (*Bereich*).
4. Was ist ein Entitätstyp? Was ist eine Entitätsmenge? Erklären Sie die Unterschiede zwischen einer Entität, einem Entitätstyp und einer Entitätsmenge.
5. Erklären Sie den Unterschied zwischen einem Attribut und einer Wertemenge.
6. Was ist ein Beziehungstyp? Erklären Sie die Unterschiede zwischen einer Beziehungsinstanz, einem Beziehungstyp und einer Beziehungsmenge.
7. Was ist eine Teilnahmerolle? Wann ist es nötig, Rollennamen in der Beschreibung von Beziehungstypen zu verwenden?
8. Beschreiben Sie die beiden Alternativen für die Spezifikation struktureller Einschränkungen für Beziehungstypen. Welche Vor- und Nachteile weist jede Alternative auf?
9. Unter welchen Bedingungen kann ein Attribut eines binären Beziehungstyps migrieren, um zu einem Attribut eines der teilnehmenden Entitätstypen zu werden?
10. Wenn wir uns Beziehungen als Attribute vorstellen, was sind dann die Wertemengen dieser Attribute? Welche Klasse von Datenmodellen basiert auf diesem Konzept?
11. Was ist mit einem rekursiven Beziehungstyp gemeint? Führen Sie einige Beispiele rekursiver Beziehungstypen auf.
12. Wann wird das Konzept einer schwachen Entität in der Datenmodellierung angewandt? Definieren Sie die Begriffe *Eigentümer-Entitätstyp*, *schwacher Entitätstyp*, *identifizierender Beziehungstyp* und *partieller Schlüssel*.
13. Kann eine identifizierende Beziehung eines schwachen Entitätstyps einen Grad größer als Zwei sein? Unterlegen Sie Ihre Antwort mit ein paar Beispielen.

14. Diskutieren Sie die Konventionen für die Darstellung eines ER-Schemas als ER-Diagramm.
15. Diskutieren Sie die Namenskonventionen für ER-Diagramme.

ÜBUNGEN

16. Betrachten Sie die folgenden Anforderungen für eine Universitätsdatenbank, die benutzt wird, um die Studentendatensätze zu verwalten. Dies ist zwar nicht identisch, aber vergleichbar mit der Datenbank aus Abbildung 1.2.

    a. Die Universität verwaltet Name, Matrikel (StudentNumber), Sozialversicherungsnummer (SSN), vorübergehende Adresse und Telefonnummer, ständige Adresse und Telefonnummer, Geburtsdatum, Geschlecht, Klasse (Freshman, Sophomore, ..., Graduate), Major Department, Minor Department (falls zutreffend) und Degree Program (BA, BS, ..., Ph.D.). Einige Benutzeranwendungen müssen auf Stadt, Land und PLZ-Code der ständigen Adresse und auf den Nachnamen von Studenten zugreifen. Die Sozialversicherungs- und Matrikelnummer sind je ein eindeutiger Wert für einen Studenten.

    b. Jedes Department wird nach Name, Code, Büronummer (Office Number), Bürotelefon und College beschrieben. Der Name und der Code sind jeweils eindeutige Werte für ein Department.

    c. Jeder Course wird nach Name, Beschreibung, CourseNumber, Number of Semester Hours, Level und Department geführt. Der Wert von CourseNumber ist ein eindeutiger Wert für jeden Course.

    d. Jede Section wird mit Instructor, Semester, Jahr, Course und Section Number erfasst. Die Section Number unterscheidet Sections des gleichen Course, der im gleichen Semester/Jahr unterrichtet werden. Die Werte sind 1, 2, 3, ..., bis zur Anzahl der in jedem Semester unterrichteten Sections.

    e. Ein Grade Report wird mit Student, Section, Letter Grade und Numeric Grade (0, 1, 2, 3 oder 4) erfasst.

    Entwerfen Sie ein ER-Schema für diese Anwendung und zeichnen Sie es als ER-Diagramm. Spezifizieren Sie Schlüsselattribute für jeden Entitätstyp und strukturelle Einschränkungen zu jedem Beziehungstyp. Beachten Sie eventuelle nicht spezifizierte Anforderungen und treffen Sie die entsprechenden Annahmen, um die Spezifikation zu vervollständigen.

17. Zusammengesetzte und mehrwertige Attribute können in eine beliebige Anzahl von Ebenen verschachtelt werden. Angenommen, wir möchten ein Attribut für einen Entitätstyp STUDENT entwerfen, um die frühere College-Ausbildung zu erfassen.

Ein solches Attribut hat einen Eintrag für jedes früher besuchte College und jeder dieser Einträge setzt sich aus College-Name, Beginn- und Enddatum, Grad (an diesem College erworbene Grade, soweit zutreffend) und Transcript-Einträge (an diesem College abgeschlossene Kurse, soweit zutreffend). Jeder Gradeintrag enthält den Gradnamen und Monat und Jahr, wann der Grad erworben wurde, und jeder Transcript-Eintrag enthält den Kursnamen, Semester, Jahr und Grad. Entwerfen Sie ein Attribut, um diese Informationen aufzunehmen. Verwenden Sie die Konventionen aus Abbildung 3.5.

18. Entwickeln Sie einen alternativen Entwurf für das in Übung 17 beschriebene Attribut, das nur Entitätstypen (einschließlich schwache Entitätstypen, soweit erforderlich) und Beziehungstypen nutzt.

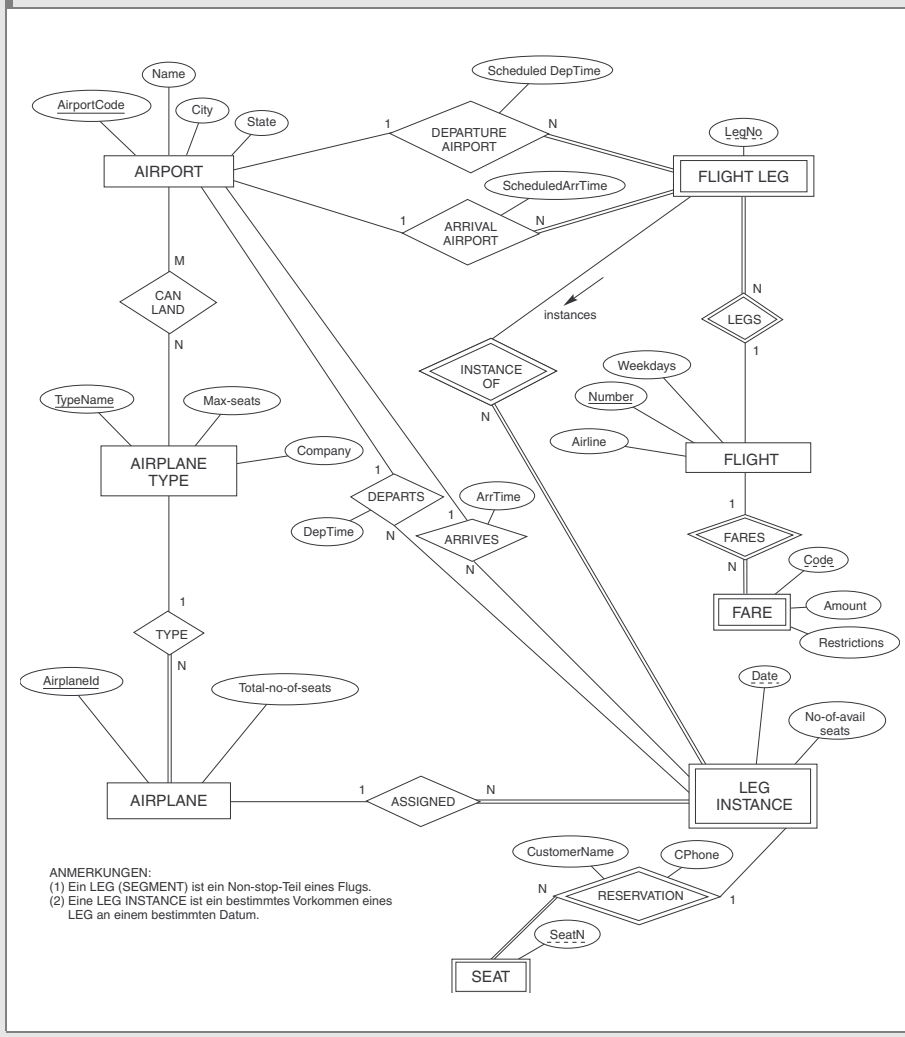
19. Betrachten Sie das ER-Diagramm in Abbildung 3.16, das ein vereinfachtes Schema für ein Flugreservierungssystem darstellt. Extrahieren Sie aus dem ER-Diagramm die Anforderungen und Einschränkungen, die zu diesem Schema geführt haben. Versuchen Sie, die Anforderungen und Einschränkungen so genau wie möglich zu definieren.

20. In Kapitel 1 und 2 wurden die Datenbankumgebung und Datenbanknutzer behandelt. Wir können viele Entitätstypen in Erwägung ziehen, um eine solche Umgebung zu beschreiben, z.B. DBMS, gespeicherte Datenbank, DBA und Katalog/Datenwörterbuch. Versuchen Sie, alle Entitätstypen zu spezifizieren, die ein Datenbanksystem und seine Umgebung voll beschreiben können. Anschließend spezifizieren Sie die zwischen ihnen bestehenden Beziehungstypen und zeichnen Sie ein ER-Diagramm, um eine solche allgemeine Datenbankumgebung zu beschreiben.

21. Entwerfen Sie ein ER-Schema für die Verwaltung von Informationen über Abstimmungen (Votes) im US House of Representatives während der laufenden zweijährigen Kongresssitzung. Die Datenbank muss jeden US-Bundesstaat nach Namen (z.B. Texas, New York, California) erfassen und beinhaltet die Region des Staats (mit Wertebereich {Northeast, Midwest, Southeast, Southwest, West}). Jede CONGRESSPERSON im House of Representatives wird nach Name beschrieben und beinhaltet den von ihr vertretenen District, das Anfangsdatum (StartDate), wann sie erstmals gewählt wurde, und die politische Partei, der die Person angehört (mit Domänen in {Republican, Democrat, Independent, Other}). Die Datenbank verfolgt jeden BILL (Gesetzesvorschlag) und beinhaltet BillName, DateOfVote, PassedOrFailed (mit Wertebereich {YES, NO}) und Sponsor (die Kongressperson(en), die den Gesetzesvorschlag unterbreitet hat bzw. haben). Die Datenbank verfolgt, wie jede Kongressperson zu jedem Gesetzesvorschlag abstimmt (mit Wertebereich {Yes, No, Abstain, Absent}). Zeichnen Sie ein ER-Schemadiagramm für diese Anwendung. Beschreiben Sie detailliert Ihre Annahmen.

22. Es soll eine Datenbank konstruiert werden, um die Teams und Spiele einer Sportliga zu verwalten. Ein Team umfasst eine Reihe von Spielern, die nicht alle an jedem Spiel teilnehmen. Es wird gewünscht, die Spieler, die in jedem Spiel für jedes Team teilnehmen, die im jeweiligen Spiel gespielten Positionen und die Spielergebnisse zu erfassen. Versuchen Sie, ein ER-Schemadiagramm

Abbildung 3.16: ER-Diagramm einer Datenbank AIRLINE.



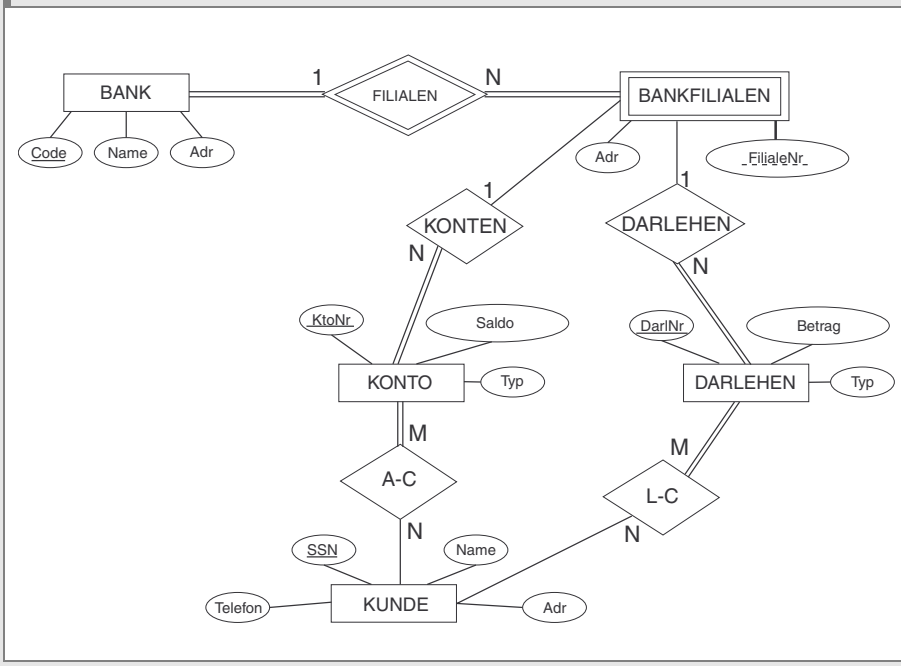
für diese Anwendung zu entwerfen, und nennen Sie eventuell von Ihnen getroffene Annahmen an. Wählen Sie Ihr bevorzugtes Spiel (Fußball, Baseball, Football, ...).

23. Betrachten Sie das ER-Diagramm in Abbildung 3.17 als Teil einer Datenbank BANK. Jede Bank kann mehrere Filialen und jede Filiale mehrere Konten und Darlehen haben.

    a. Führen Sie die (nicht schwachen) Entitätstypen im ER-Diagramm auf.

    b. Enthält das ER-Diagramm einen schwachen Entitätstyp? Falls ja, nennen Sie seinen Namen, den partiellen Schlüssel und die identifizierende Beziehung.

Abbildung 3.17: Ein ER-Diagramm für eine Datenbank BANK.



c. Welche Einschränkungen werden vom partiellen Schlüssel und von der identifizierenden Beziehung des schwachen Entitätstyps in diesem Diagramm spezifiziert?

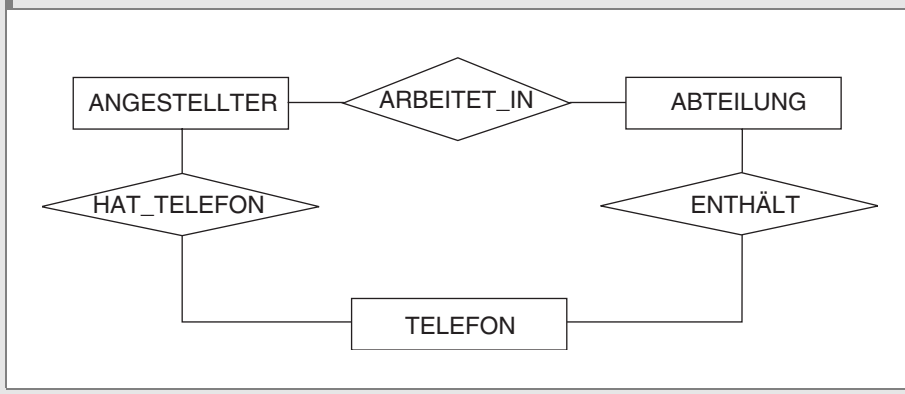
d. Führen Sie die Namen aller Beziehungstypen auf und spezifizieren Sie die (min, max) Einschränkung jeder Teilnahme eines Entitätstyps an einem Beziehungstyp. Belegen Sie Ihre Angaben.

e. Führen Sie möglichst genau die Benutzeranforderungen auf, die zu diesem ER-Schemaentwurf geführt haben.

f. Nehmen wir an, dass jeder Kunde mindestens ein Konto haben muss, jedoch auf höchstens zwei Darlehen begrenzt ist, und dass eine Bankfiliale nicht mehr als 1000 Darlehen haben kann. Wie zeigt sich dies bei den (min, max) Einschränkungen?

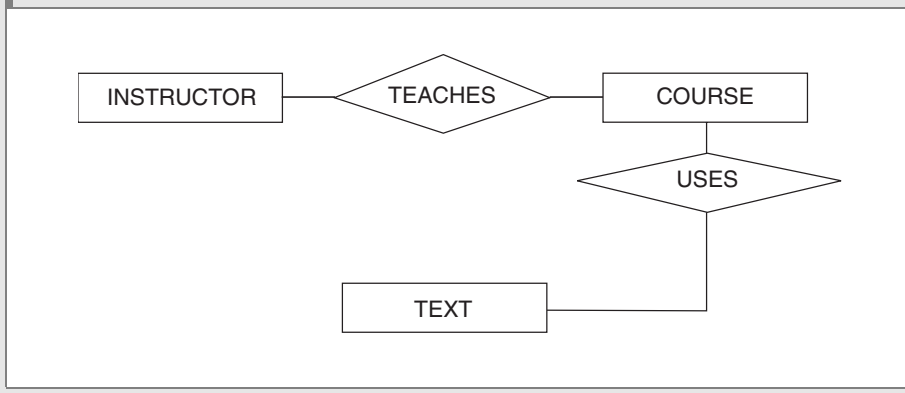
24. Betrachten Sie das ER-Diagramm in Abbildung 3.18. Angenommen, ein Angestellter arbeitet in bis zu zwei Abteilungen, kann aber auch überhaupt keiner Abteilung zugeteilt sein. Weiter wird angenommen, dass jede Abteilung mindestens eine und bis zu drei Telefonnummern haben kann. Geben Sie die (min, max) Einschränkungen zu diesem Diagramm an. *Nennen Sie eventuelle weitere, von Ihnen getroffene Annahmen*. Unter welchen Bedingungen wäre die Beziehung HAT_TELEFON im obigen Beispiel redundant?

Abbildung 3.18: Ein ER-Diagramm für eine Datenbank, die Firmen- und Angestelltentelefone verwaltet.



25. Betrachten Sie das ER-Diagramm in Abbildung 3.19. Angenommen, ein Kurs (Course) kann, muss aber kein Fachbuch (Text) verwenden, während ein Fachbuch der Definition zufolge ein Buch ist, das in irgendeinem Kurs benutzt wird. Ein Kurs darf nicht mehr als fünf Bücher verwenden. Die Dozenten (Instructor) unterrichten zwei bis vier Kurse. Geben Sie die (min, max) Einschränkungen zu diesem Diagramm an. *Nennen Sie eventuelle weitere, von Ihnen getroffene Annahmen*. Wenn wir die Beziehung `ADOPTS` zwischen `INSTRUCTOR` und `TEXT` hinzufügen, welche (min, max) Einschränkungen würden Sie dann anfügen? Warum?

Abbildung 3.19: Ein ER-Diagramm für eine Datenbank, die Fachbücher verwaltet, die in Kursen verwendet werden.



26. Betrachten Sie einen Entitätstyp `SECTION` in einer Datenbank `UNIVERSITY`, der die Sections beschreibt, die Courses anbieten. Die Attribute von `SECTION` sind: SectionNumber, Semester, Year, CourseNumber, Instructor, RoomNo (der Raum, in dem die Section unterrichtet wird), Building (Gebäude, in dem die Section unterrichtet wird), Weekdays (der gültige Bereich ist die möglichen Kombina-

tion von Wochentagen, in denen eine Section angeboten werden kann {MWF, MW, TT usw.}) und Hours (der Bereich umfasst alle möglichen Zeitperioden, in denen Sections angeboten werden {9–9.50, 10–10.50, ..., 15.30–16.50, 17.30–18.20 usw.}). Es wird angenommen, dass SectionNumber für jeden Course in einer bestimmten Semester/Year-Kombination eindeutig ist (d.h., wenn ein Course mehrmals in einem bestimmten Semester angeboten wird, werden die Section-Angebote mit 1, 2, 3 usw. durchnummeriert). Es gibt mehrere zusammengesetzte Schlüssel für SECTION und einige Attribute sind Komponenten von mehr als einem Schlüssel. Identifizieren Sie drei zusammengesetzte Schlüssel und zeigen Sie, wie sie in einem ER-Schemadiagramm dargestellt werden können.

AUSGEWÄHLTE LITERATUR

Das Entity-Relationship-Modell (ER-Modell) wurde von Chen (1976) eingeführt; damit zusammenhängende Arbeiten erscheinen in Schmidt und Swenson (1975, Wiederhold und Elmasri (1979) sowie Senko (1975). Seither wurden zahlreiche Modifikationen für das ER-Modell vorgeschlagen. Wir haben einige davon in unserer Präsentation berücksichtigt. Strukturelle Einschränkungen für Beziehungen werden in Abrial (1974), Elmasri und Wiederhold (1980) sowie Lenzerini und Santucci (1983) diskutiert. Mehrwertige und zusammengesetzte Attribute im ER-Modell werden in Elmasri u.a. (1985) behandelt. Obwohl wir keine Sprachen für das Entity-Relationship-Modell und seine Erweiterungen behandelt haben, wurden für solche Sprachen mehrere Vorschläge unterbreitet. Elmasri und Wiederhold (1981) schlagen die Anfragesprache GORDAS für das ER-Modell vor. Eine weitere ER-Anfragesprache wurde von Markowitz und Raz (1983) vorgeschlagen. Senko (1980) präsentiert eine Anfragesprache für sein DIAM-Modell. Eine formelle Reihe von Operationen mit der Bezeichnung ER-Algebra wurde von Parent und Spaccapietra (1985) nennen. Gogolla und Hohenstein (1991) präsentieren eine weitere formale Sprache für das ER-Modell. Campbell u.a. (1985) präsentieren eine Reihe von ER-Operationen und zeigen, dass sie relational komplett sind. Eine Konferenz für die Verbreitung von Forschungsergebnissen über das ER-Modell wird regelmäßig seit 1979 abgehalten. Die inzwischen als »International Conference on Conceptual Modeling« bezeichnete Konferenz fand in Los Angeles (ER 1979, ER 1983, ER 1997), Washington (ER 1981), Chicago (ER 1985), Dijon (ER 1986), New York (ER 1987), Rom (ER 1988), Toronto (ER 1989), Lausanne (ER 1990), San Mateo (ER 1991), Karlsruhe (ER 1992), Arlington (ER 1993), Manchester (ER 1994), Brisbane (ER 1995), Cottbus (ER 1996) und Singapur (ER 1998) statt.