

Skriptum Datenmodellierung

Dr. Ewald M. Jarz

© 2003

Inhalt

1 Datenmodellierung	3
1.1 Datenorientierung vs. Funktionsorientierung	4
1.2 Begriffe	5
1.3 Datenbankarchitektur	7
1.3.1 Hierarchisches Datenmodell	7
1.3.2 Netzwerkmodell	9
1.3.3 Relationale Datenbanken.....	10
1.3.4 Objektdatenbanken.....	11
1.4 Das 3-Schema-Modell	11
2 Entwurf relationaler Datenbanken	14
2.1 Identifikation der Funktionsträger und Funktionsbeschreibungen.....	14
2.2 Ableitung von Datenelementen	15
2.3 Zusammengesetzte und elementare Datenelemente	16
2.4 Identifikationsschlüsselvergabe	18
2.5 Beziehungen zwischen den Entitätstypen.....	19
2.6 Beziehungsarten	20
2.7 Datentypen und Normalisierung.....	26
2.8 Musterlösung für die Aufgabenstellung "Kursverwaltung":	32
2.9 Weiterführende Konzepte des relationalen Datenmodells	34
2.9.1 Generalisierung.....	34
2.9.2 Auflösen von Hierarchien.....	36
2.9.3 Indirekte Rekursion	41
2.9.4 4. und 5. Normalform.....	43
2.9.5 Integritätsbedingungen.....	45
2.9.6 Summenfelder	47
3 Zusammenfassung	48
3.1 Bedeutung von Datenmodellen.....	48
3.2 Vorgehensschritte der Datenmodellierung	49
3.3 Darstellungstechniken.....	49
4 Beispiele	50
4.1 Beispiel Begriffe:	50
4.2 Beispiel Kunden - Artikel	51

Datenmodellierung

4.3	Beispiel Stundenzettel.....	51
4.4	Beispiel Rechnung	52
4.5	Beispiel Liegenschaftsverwaltung.....	53
4.6	Beispiel PC-Komponenten.....	53
4.7	Beispiel Lager.....	54
4.8	Beispiel Fahrradkurier	54
4.9	Beispiel Sortiment Handelshaus	55
4.10	Beispiel Vinothek	55
4.11	Beispiel Inselverwaltung.....	56
5	Lösungsvorschläge.....	58
5.1	Lösungsvorschlag Stundenzettel.....	58
5.2	Lösungsvorschlag Rechnung	58
5.3	Lösungsvorschlag Liegenschaftsverwaltung.....	58
5.4	Lösungsvorschlag PC-Komponenten.....	59
5.5	Lösungsvorschlag Lager.....	59
5.6	Lösungsvorschlag Fahrradkurier	60
5.7	Lösungsvorschlag Sortiment Handelshaus:.....	60
5.8	Lösungsvorschlag Vinothek	61
5.9	Lösungsvorschlag Inselverwaltung.....	61
6	Literatur	62

1 Datenmodellierung

Ein hoher Prozentsatz aller betrieblichen EDV-Anwendungen verarbeitet strukturierte Daten, bildet Kartei- oder Formulare Systeme nach und stellt dabei die Verfügbarkeit der Daten für alle Benutzer sicher. Die programmtechnische Abbildung der Informationen und ihrer Zusammenhänge geschieht mit Hilfe so genannter Datenbanksysteme. Datenbanksysteme bilden die Grundlage für Anwendungen wie z.B. Finanzbuchhaltung, Kostenrechnung, Auftragsverwaltung oder PPS-Systeme. Auch einfache Anwendungen, wie eine Adressverwaltung für den persönlichen Bedarf, kann auf Basis eines Datenbanksystems (DBMS¹) entwickelt werden.

Nicht die Komplexität der Aufgabenstellung ist das Kriterium der Einsetzbarkeit von DBMS, sondern die Tatsache, dass strukturierte Informationen (= Daten) zu verwalten sind. Eine Textverarbeitung z.B. dient der Manipulation freien Textes, wobei der Inhalt und die Struktur des Textes unerheblich ist. Ein DBMS dient der Verwaltung von Daten z.B. Adressen, die eine klare Struktur aufweisen. Adressen beispielsweise lassen sich in "Straße", "Postleitzahl" und "Ort" strukturieren.

DBMS sind Programme, die auf verschiedensten Hardwareplattformen angeboten werden. Das Angebot reicht von PC-orientierten Produkten (z.B. MySQL, Interbase, dBase, Sybase, MS-Access, Access SQL, u.v.a.) bis zu Produkten die mittlere bis große EDV-Systeme voraussetzen (z.B. ORACLE, Ingres, Informix, DB2, MS-SQL Server u.s.w.). Die aufgabenbezogenen Datenstrukturen (das Datenmodell) sind jedoch unabhängig von den Eigenschaften der einzelnen Produkte. Es ist für das Datenmodell einer Adressverwaltung unerheblich ob die Umsetzung mit MS-Access oder ORACLE erfolgt, entscheidend ist zunächst einzig und allein die Aufgabenstellung.

DBMS und die damit in Zusammenhang stehenden Werkzeuge zur Programmentwicklung bilden die Grundlage für die Entwicklung aufgabenbezogener Programme. Im Gegensatz zu Programmen wie Textverarbeitung oder Grafikprogrammen, mit denen eine Aufgabenstellung direkt bearbeitet werden kann, müssen bei einem DBMS zunächst das Datenmodell und das Funktionsmodell aufgestellt und implementiert werden. Erst danach können Daten erfasst und ausgewertet werden.

Das **Datenmodell** beschreibt die Struktur der zu verarbeitenden Informationen (Welche Daten?, Wie hängen sie zusammen?), das **Funktionsmodell** beschreibt den Verarbeitungsprozess (Welche Auswertungen?, Wie werden Daten erfasst?).

Die folgenden Ausführungen beschreiben den Entwurfsprozess für ein Datenmodell, wobei dieser Prozess unabhängig von DBMS-Produkten gesehen wird.

¹ DBMS = **D**ata**B**ase**M**anagement**S**ystem, RDBMS = Relational DBMS

1.1 Datenorientierung vs. Funktionsorientierung

Jede betriebliche Problemstellung besitzt eine mehr oder weniger große Informationskomponente, von der meist ein Teil mittels eines Informations- und Kommunikationssystems (IKS) automatisiert oder zumindest unterstützt werden kann. Die Automatisierung oder Unterstützung erfolgt mittels Anwendungsprogrammen (z.B. Auftragsverwaltung, Finanzbuchhaltung, Lohn- und Gehaltsverrechnung usw.). Jedes Anwendungsprogramm benötigt Daten und Funktionen. Die Daten repräsentieren die Informationen, die zur Bearbeitung der Aufgabe benötigt werden. Die Struktur der Daten spiegelt sich im Datenmodell wieder. Funktionen beinhalten den dynamischen Aspekt der Bearbeitung. Die Beschreibung einer Funktion besteht aus dem Prozessdesign und dem Dialogdesign. Das Prozessdesign gibt den Ablauf der Arbeitsschritte wider aus denen die Funktion besteht², das Dialogdesign beschreibt den Kommunikationsprozess zwischen Mensch und Maschine, wobei besonders auf Ergonomie und Effizienz Wert gelegt wird. Dieser Zusammenhang wird in der Abbildung "Daten- und Funktionsaspekt" dargestellt.

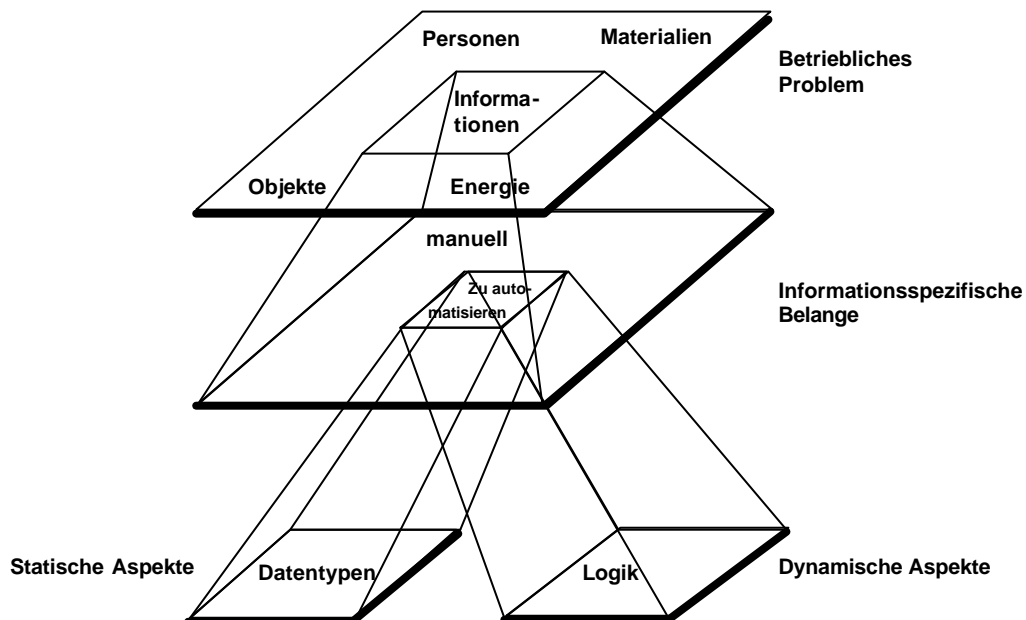


Abbildung 1: Daten- und Funktionsaspekt (nach Vetter M.)

Die Beschreibung eines Anwendungssystems beinhaltet demnach ein Daten- und ein Funktionsmodell. Das Funktionsmodell beschreibt dabei die Möglichkeiten eines Benutzers, Informationen in das System einzugeben oder abzufragen. Jeder Funktion kann somit ein externes Schema zugeordnet werden, das die Daten beschreibt, die diese Funktion bearbeitet.

² Das Prozessdesign der Funktion "Mitarbeiter erfassen" könnte (vereinfacht) wie folgt formuliert werden: Eingabe einer Personalnummer, Überprüfung der Eindeutigkeit, Vervollständigen der Bildschirmmaske, Abspeichern. Zur Darstellung des Prozessdesigns bedient man sich meist grafischer Darstellungsmethoden (Struktogramme, Flow Charts, SADT, UML)

1.2 Begriffe

Bevor Datenbankarchitekturen und der eigentliche Entwurfsprozess beschrieben werden, ist eine Klärung von Begriffen notwendig. Zu diesem Zweck kann man Datenbanken mit leistungsfähigen Karteikästen vergleichen. Das Grundmodul eines Karteikastens ist eine Karteikarte. Auf einer leeren Karteikarte befinden sich Beschriftungen und Bereiche, in denen die variablen Informationen eingetragen werden.

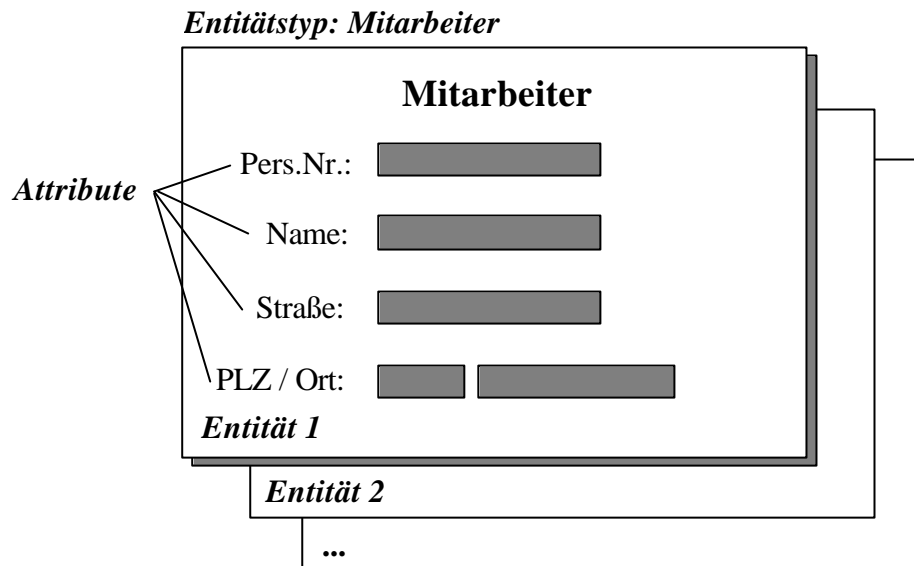


Abbildung 2: Begriffe der Datenmodellierung

Alle Mitarbeiter sind durch dieselben Eigenschaften (Personalnummer, Name, Adresse, usw.) zu beschreiben. Die Eigenschaften (nicht die konkreten Ausprägungen wie z.B. "Hans Maier") werden als **Attribute** bezeichnet. Die Menge aller Mitarbeiter (gekennzeichnet durch dieselben Eigenschaften) heißt **Entitätstyp**, die konkrete Ausprägung eines Mitarbeiters (z.B. der Mitarbeiter "Hans Maier, Dorfstr. 20,...") wird als **Entität** bezeichnet.

In dem Karteikasten werden somit die Informationen über den Entitätstyp "Mitarbeiter" verwaltet. Eine Karteikarte enthält die Daten einer Entität (z.B. Hans Maier), die Daten werden durch die vorgesehenen Attribute strukturiert.

Eine **Entität** ist ein individuelles und identifizierbares Exemplar von Dingen, Personen, Begriffen der realen oder der Vorstellungswelt. Ein **Entitätstyp**³ ist die Menge von Entitäten mit gleichen Attributen.

³ Der Begriff **Entitätstyp** wird oft auch als **Entitätsmenge** bezeichnet und entspricht dem englischen Begriff **Entity**, der mitunter mit "Entität" übersetzt wird. Es muss somit der Begriffsinhalt bei der Lektüre von Texten genau untersucht werden.

	Entitätstyp	Attribute	Entitäten
Individuen	Einwohner	Name Adresse	Stefan Müller Kirchplatz 2
	Mitarbeiter	Personalnummer Name Abteilung	423 Hans Maier Marketing
reale Objekte	Maschinen	Typ Funktion	PC 486 Personal Computer
	Gebäude	Adresse Baustil Architekt	Kirchplatz 1 Barock ...
abstrakte Objekte	Fachgebiete	Thema Definition	Informatik ...
	Krankheiten	lat. Name dt. Name Therapie	Influenza Grippe Aspirin
Ereignisse	Feiertage	Name Datum	Neujahr 1. Jänner
	Vorlesungen	Titel Ort Referent	Handelsrecht Hörsaal 12 Dr. H.G. Bauer

Abbildung 3: Beispiele für Entitätstypen mit Attributen

Aus der Aussage, dass es sich bei Entitäten um **eindeutig** identifizierbare Objekte handelt, lässt sich ableiten, dass es auch ein Attribut (oder zumindest eine Gruppe von Attributen) gibt, das die Entität tatsächlich identifiziert. Dieses Attribut wird **Identifikationsschlüssel** genannt. Für den Entitätstyp "Mitarbeiter" könnte beispielsweise die Personalnummer als Identifikationsschlüssel verwendet werden.

Identifikationsschlüssel müssen folgenden Kriterien genügen:

- **Eindeutigkeit:** Der Identifikationsschlüssel einer Entität darf nur einmal in diesem Entitätstyp aufscheinen. Es darf also jede Personalnummer nur einmal vorkommen, jeder Mitarbeiter hat seine eigene Personalnummer⁴.
- **Lebensdauer:** Der Identifikationsschlüssel darf sich während der gesamten Lebensdauer einer Entität nicht ändern. Ein Mitarbeiter behält seine Personalnummer solange er im Unternehmen beschäftigt ist. Wenn sie sich ändern würde, ergäben sich Probleme in den Akten, weil diese dann nicht mehr zuordenbar wären. Die alte Personalnummer wäre ja nicht mehr existent.

⁴ Der Name als Identifikationsschlüssel ist ungeeignet, weil dieselben Namen durchaus vorkommen können. Die Eindeutigkeit ist nicht gewährleistet.

- **Sofortige Vergabe:** Der Identifikationsschlüssel muss sofort beim Entstehen der Entität vergeben werden. Es darf keinen Mitarbeiter ohne Personalnummer geben, weil sonst eine Zuordnung von Akten nicht möglich wäre.

Die Begriffe Entitätstyp, Entität, Attribut und Identifikationsschlüssel sind die Grundbegriffe der Datenmodellierung. Die Aufgabenstellungen, die mit Datenbanksystemen gelöst werden, betreffen im allgemeinen mehr als einen Entitätstyp. Die Entitätstypen stehen miteinander in Beziehung. Eine Personaldatenbank verwaltet die Daten von **Mitarbeitern** und von **Abteilungen**. Die Beziehung zwischen diesen beiden Entitätstypen ergibt sich aus der Tatsache, dass alle Mitarbeiter einer Abteilung angehören.

Eine **Beziehung** stellt eine logische Verbindung zwischen zwei (oder mehreren) Entitätstypen dar und beschreibt einen konkreten Tatbestand.

Für einige der hier vorgestellten Begriffe gibt es eine Reihe von (fast) Synonymen, die sich durch ihre Abstraktionsebene unterscheiden. Ein **Entitätstyp** ist ein abstrakter Begriff, der bei der Realisierung der Datenbank in einer **Tabelle** oder **Relation**⁵ abgebildet wird. Diese wiederum wird im Computer in einer **Datei** (File) abgespeichert. Im Sprachgebrauch werden alle drei Begriffe praktisch synonym verwendet. Ebenso verhält es sich bei den Begriffen **Attribut - Spalte - Feld** und **Entität - Zeile - Datensatz** (Record oder Segment). In den folgenden Ausführungen werden diese Begriffe immer wieder auftauchen, wobei bei ihrer Verwendung auf die unterschiedlichen Abstraktionsgrade Rücksicht genommen wurde.

1.3 Datenbankarchitektur

Die unterschiedlichen Möglichkeiten der programmtechnischen Realisierung, insbesondere von Beziehungen, in DBMS führte zur Entwicklung verschiedener Datenbankarchitekturen. Die älteste Form ist das hierarchische Datenmodell. Aus ihr entwickelte sich das Netzwerkmodell⁶. Das Konzept der relationalen Datenbank ist zur Zeit weitgehend als Stand der Technik akzeptiert. Objektdatenbanken endlich stellen ein neues Konzept dar, dessen praktische Anwendungsbereiche noch sehr beschränkt sind.

1.3.1 Hierarchisches Datenmodell

Hierarchische Datenmodelle kennen Entitätstypen, Entitäten und Attribute wie bisher beschrieben. Die Möglichkeit Beziehungen darzustellen beschränkt sich jedoch auf streng hierarchische Strukturen. Einer Entität können ein oder mehrere Entitäten ("Söhne") zugeordnet werden, eine Entität der niedrigeren Hierarchiestufe (Sohn) ist genau von einer Entität (Vater) abhängig.

Man kann sich ein hierarchisches Datenmodell als Karteikasten vorstellen in dem für jede Abteilung eine Karteitasche angelegt ist, auf die über den Identifikationsschlüssel der Abteilung zugegriffen werden kann. In diesen Karteikasten sind die Karteikarten der Mitarbeiter abgelegt.

⁵ Der Begriff "Relation" wird oft auch verwendet, um Beziehungen zwischen Entitätstypen zu bezeichnen. Dieser Umstand kann zu Missverständnissen führen, deshalb sollte der Begriff mit Vorsicht verwendet und interpretiert werden.

⁶ Nicht zu verwechseln mit verteilten Datenbanken in Netzwerken.

Datenmodellierung

Auf die Mitarbeiter kann also nur über die Abteilung zugegriffen werden. Die Karteikarte des Mitarbeiters "Maier" kann nur gefunden werden, wenn bekannt ist, in welcher Abteilung er beschäftigt ist.

Die Daten einer Entität werden als **(Daten-)Satz, Segment oder Record** bezeichnet. Die Attribute einer Entität werden in **Feldern** abgespeichert. Der Datensatz für einen Mitarbeiter könnte z.B. wie folgt aussehen:

423 Hans Maier Marketing

Während das Karteikastenmodell nur eine Vater-Sohn Beziehung (Abteilung-Mitarbeiter) zulässt können in einem hierarchischen Datenmodell mehrere solche Beziehungen abgebildet werden (z.B. zusätzlich: Abteilung-Räume).

In einem hierarchischen Datenmodell kann der Entitätstyp "Abteilung" als Ausgangspunkt ("Wurzel", "Root" "Mastersegment") gesehen werden. Einer Abteilung gehören ein oder mehrere Mitarbeiter an, der Entitätstyp Mitarbeiter ist somit der Sohn. Der Zugriff auf die Daten einer bestimmten Entität "Mitarbeiter" kann nur über den Vater, also über die Abteilung erfolgen.

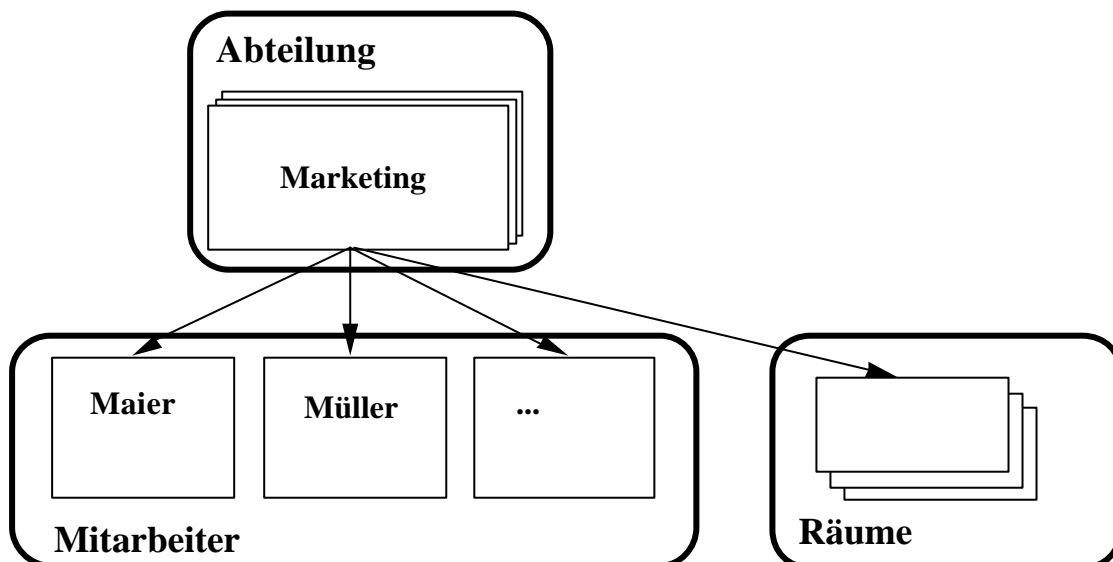


Abbildung 4: Hierarchisches Datenmodell

Es ist durch die strengen Restriktionen des Modells nicht möglich direkt auf die Mitarbeiter einer Abteilung zuzugreifen, es muss zuerst auf die Abteilung zugegriffen werden.

Eine weitere Schwäche des Modells liegt in der Tatsache, dass Entitäten nur einen Vater haben können. Somit können Mitarbeiter, wenn sie, wie oben dargestellt, von einer Abteilung abhängen, nicht gleichzeitig z.B. Projekten zugeordnet werden. Ebenso wenig lässt sich eine Beziehung zwischen Mitarbeiter und Raum herstellen. Um dies zu erreichen müssen die Mitarbeiter-

daten mehrfach, also **redundant**, geführt werden. Redundante (mehrfache) Datenhaltung steigert allerdings die Gefahr der **Inkonsistenz**⁷ und den Bedarf an Speicherkapazität.

Historisch ist das hierarchische Datenmodell weit verbreitet, weil es eine effiziente Weiterentwicklung herkömmlicher Dateisysteme darstellt. "Herkömmliche Dateisysteme" werden vom Betriebssystem zur Verfügung gestellt. Datensätze in den Dateien können nur sequentiell (von vorne nach hinten) gelesen werden oder es kann direkt auf einen Datensatz zugegriffen werden, z.B. auf den 5. Datensatz der Datei. Es besteht keine Unterstützung bei der Suche nach einem Datensatz mit einem bestimmten Inhalt. Eine Suche nach "Name" oder "Personalnummer" wird vom Betriebssystem nicht unterstützt⁸.

Ein Beispiel für ein hierarchisches DBMS-Produkt ist IMS (von IBM). Die erwähnten Schwachstellen führten jedoch zur Weiterentwicklung und somit zum Netzwerkmodell.

1.3.2 Netzwerkmodell

Das Netzwerkmodell hebt die Hierarchiebedingung in den Beziehungen auf. Es ist somit möglich die Entitätstypen beliebig miteinander in Beziehung zu setzen. Damit sind Beziehungen zwischen Entitätstypen auf den unteren Hierarchieebenen (z.B. Räume und Mitarbeiter) möglich und außerdem können Entitätstypen auch mehrere übergeordnete Entitätstypen (Väter) haben.

Im Netzwerkmodell kann man Mitarbeiter nun sowohl einer Abteilung, als auch einem Projekt zuordnen, außerdem ist eine Beziehung zwischen Entitätstypen derselben Hierarchiestufe möglich. So kann z.B. eine Beziehung zwischen Mitarbeiter und Raum abgebildet werden, womit festgestellt werden kann, wo Mitarbeiter ihr Büro haben.

Ein Beispiel für ein DBMS-Produkt, das das Netzwerkmodell unterstützt, ist CODASYL (Siemens).

Für das Netzwerkmodell besteht keine analoge Karteiorganisationsform. Das Netzwerkmodell ist in der Lage wesentlich komplexere Informationsstrukturen redundanzfrei (also ohne Mehrfachspeicherung von Informationen) darzustellen als das hierarchische Datenmodell. Dadurch ergeben sich gesteigerte Anforderungen an die Datenorganisation (... die Verwaltung der Daten im EDV-System). Der Aufwand zur Realisierung und zum Betrieb eines Netzwerk-DBMS ist erheblich, zudem ist eine Netzwerkdatenbank wegen der starren Abbildung von Beziehungen Änderungen der Datenstruktur nur schwer zugänglich.

⁷ Unter Inkonsistenz wird die fehlende Eindeutigkeit des Datenbestandes (Informationen) verstanden.. Es kann beispielsweise vorkommen, dass bei doppelter Datenhaltung Herr Maier einmal als ledig und einmal als verheiratet abgespeichert ist.

⁸ Diese Aussage gilt nicht unbeschränkt. Einige Betriebssysteme haben Mechanismen implementiert, die ebendies ermöglichen, andere wiederum werden mit einem integrierten Datenbanksystem vom Hersteller ausgeliefert

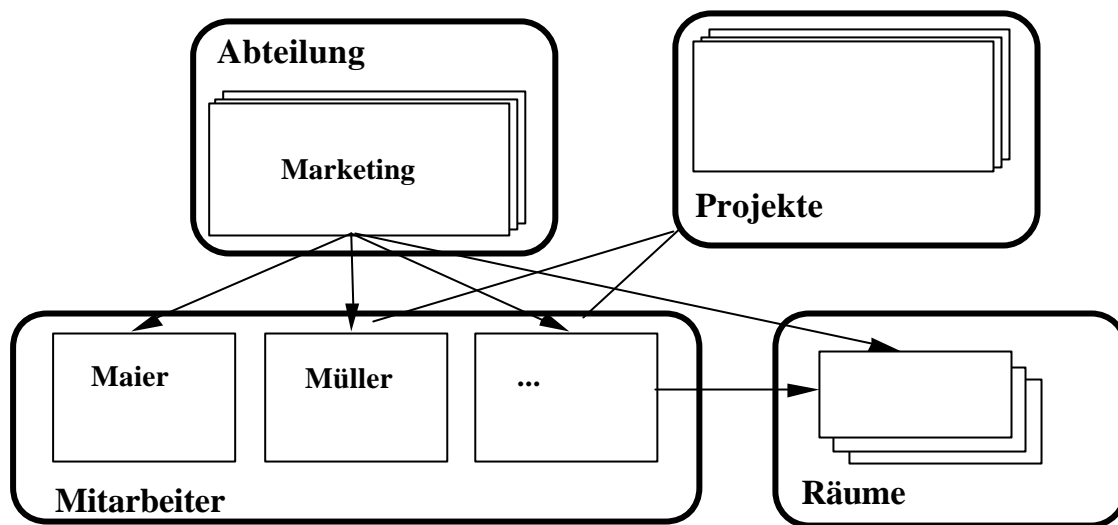


Abbildung 5: Datenbank im Netzwerkmodell

1.3.3 Relationale Datenbanken

Während im Netzwerk- und hierarchischen Modell die Beziehungen zwischen Entitätstypen fest im Datenmodell integriert sind, bildet das relationale Datenmodell Beziehungen erst über den Inhalt der Datensätze ab. Die Entitätstypen werden in Tabellen abgebildet, deren Spalten die Attribute und deren Zeilen die Entitäten darstellen. Diese Tabellen existieren zunächst unabhängig voneinander.

Entitätstyp: **Mitarbeiter**

Personalnr	Nachname	Vorname	Kontonummer	...
000123	Maier	Hans	8086291	...
000124	Müller	Peter	1721729	...
000125	Hager	Monika	2356487	

Eine Tabelle besteht aus einer Menge von Datensätzen (Zeilen) für die gilt, dass jeder Datensatz aus den gleichen Attributen (Spalten, Felder) besteht. Die eindeutige Identifikation eines Datensatzes erfolgt mittels eines Identifikationsschlüssels (Attribut, eindeutig, sofort vergeben, nicht veränderbar), der durch Unterstreichen gekennzeichnet ist. Im obigen Beispiel ist die Personalnummer der Identifikationsschlüssel.

Beziehungen werden abgebildet, indem Identifikationsschlüssel als Fremdschlüssel in andere Tabellen übernommen werden. Um z.B. eine Beziehung zwischen den Entitätstypen "Abteilung" und "Mitarbeiter" herzustellen sind folgende Tabellen erforderlich:

Datenmodellierung

Entitätstyp: **Mitarbeiter**

Personalnr	Nachname	Vorname	Kontonummer	Abteilungsnr.
000123	Maier	Hans	8086291	1
000124	Müller	Peter	1721729	1
000125	Hager	Monika	2356487	3

Entitätstyp: **Abteilung**

Abteilungsnr.	Bezeichnung	Kostenstelle	Budget	...
1	Marketing	2312	8086291	...
2	Verkauf	2313	1721729	...
3	Ausbildung	2314	2356487	

Die Abteilungsnummer in der Tabelle Mitarbeiter wird dort als Fremdschlüssel bezeichnet. Über diese Beziehung wird definiert, dass der Datensatz aus der Tabelle Mitarbeiter mit dem entsprechenden Datensatz in der Tabelle Abteilung verknüpft ist. (Hr. Maier ist Mitarbeiter der Abteilung 1 Marketing). Umgekehrt kann auch festgestellt werden, wer in einer Abteilung arbeitet (der Abteilung Marketing sind die Mitarbeiter Maier und Müller zugeordnet).

Das relationale Datenmodell stellt hohe Anforderungen an die Leistungsfähigkeit des Systems. Durch die Entwicklungen in diesem Bereich und die hohe Flexibilität des Modells bilden relationale DBMS (=RDBMS) heute die Grundlage für einen Großteil der datenbankgestützten Softwareentwicklung.

1.3.4 Objektdatenbanken

Objektdatenbanken sind eine Weiterentwicklung des relationalen Datenbankmodells. Objekte entstehen indem man Entitätstypen weitere Eigenschaften wie z.B. Vererbung von Attributen und Attributinhalt zuordnet. Diese Objekte beinhalten außerdem Informationen über Verarbeitungsmethoden und Funktionen, die auf sie anwendbar sind.

1.4 Das 3-Schema-Modell

Für die Beschreibung relationaler Datenbanken gibt es zunächst drei Sichtweisen:

Die **Benutzersicht**: Jeder Benutzer hat entsprechend seiner Aufgabenstellung unterschiedliche Anforderungen an das Datenbanksystem. So kann die Marketingabteilung für die Kunden "Branche" und "Vorjahresumsatz" als Attribute benötigen, während das Rechnungswesen am "Kreditlimit" als Attribut interessiert ist. Die Darstellung der Benutzersicht wird als **externes Schema** bezeichnet.

Die **Sicht des Datenbankdesigners**: Aus der Summe der Datenelemente (Entitätstypen und Attribute), die in den einzelnen Benutzersichten enthalten sind, ergibt sich eine Vereinigungs-

Datenmodellierung

menge von Entitätstypen und Attributen. Diese müssen im Zuge des Entwurfsprozesses normalisiert werden. Die Darstellung der Datenbank erfolgt mittels Attributlisten, Datenbankstrukturdiagrammen (Entity-Relationship - Diagramm) und Tabellen (zur Überprüfung der Funktionsfähigkeit - "Schreibtischtest"). Die Beschreibung der Datenbank auf dieser Ebene stellt das **konzeptionelle Schema** dar. Die Tabelle "Kunden" im konzeptionellen Schema besteht (ausgehend von obigem Beispiel) aus den Attributen "Branche", "Vorjahresumsatz" und "Kreditlimit". Dazu kommen natürlich der "Name" des Kunden und eine "Nummer" als Identifikationsschlüssel.

Die **Sicht des RDBMS**: Die Darstellung des konzeptionellen Schemas reicht noch nicht aus um das Datenmodell mittels eines RDBMS zu implementieren. Entsprechend den Möglichkeiten des RDBMS (Feldtypen, maximale Satzlängen usw.) muss das Datenmodell entsprechend modifiziert und/oder ergänzt werden. Dabei muss auch der Zugriffsalgorithmus des RDBMS berücksichtigt werden, um eine optimale Performance des Systems zu erlangen. Die an das RDBMS angepasste Darstellung des Datenmodells heißt **internes Schema**.

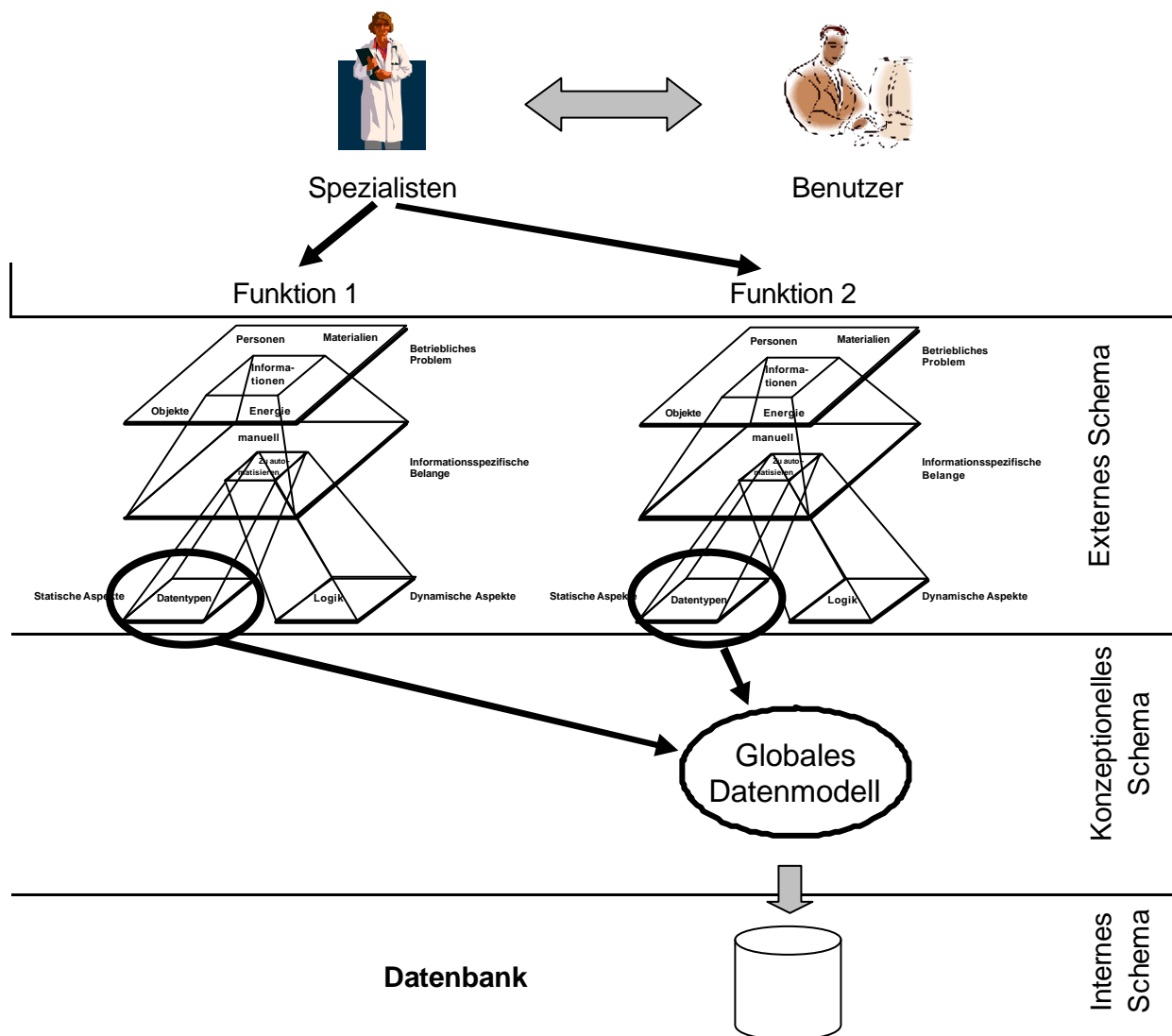


Abbildung 6: 3-Schemamodell

Datenmodellierung

Die drei Schemata stehen zueinander über Schnittstellen in Verbindung. Solange die Schnittstellen unverändert bleiben, können in jedem Schema unabhängig voneinander Änderungen durchgeführt werden. So kann z.B. aus den vorhandenen Attributen eine weitere Benutzersicht hinzugefügt werden, ohne dass sich am konzeptionellen Schema etwas ändert. Auch wenn neue Attribute oder Entitätstypen hinzukommen, bleiben die anderen Benutzersichten davon unberührt.

Das interne Schema wird zunächst aus dem konzeptionellen Schema abgeleitet, wobei eine Anpassung an das verwendete RDBMS erfolgt. Änderungen im konzeptionellen Schema haben somit immer eine Auswirkung auf das interne Schema. Umgekehrt kann aber das interne Schema angepasst werden (z.B. Änderung der Länge eines Textfeldes) ohne dass davon das konzeptionelle Schema berührt wird.

2 Entwurf relationaler Datenbanken

Die folgenden Ausführungen stellen ein Vorgehensmodell für den Entwurf relationaler Datenbanken vor. Zur praktischen Demonstration wird folgendes Beispiel verwendet:

Bei der Firma KNOWING werden laufend Kurse zur Mitarbeiterschulung organisiert. Zu jedem dieser Kurse, die von verschiedenen externen Referenten gehalten werden, können sich die Mitarbeiter der KNOWING anmelden. Aus Kostengründen wird jeder Kurs von genau einem Referenten gehalten, Co-Referate sind demnach ausgeschlossen. Die Administration der Kurse obliegt der Personalabteilung, die Anmeldung zu den Kursen soll über das EDV-System erfolgen. Die Teilnehmer müssen von ihren Vorgesetzten zu den Kursen freigestellt werden. Vor dem Kurs erhält jeder Teilnehmer eine Anmeldebestätigung. Der Erfolg der Teilnahme wird für Auswertungen durch Mitarbeiter der Personalabteilung erfasst.

Die Daten in einer Datenbank sind in einer bestimmten Struktur organisiert. Diese Struktur ist ein Abbild der interessierenden Realwelt. Bei der Datenmodellierung wird versucht, die Strukturen der Realwelt aus einer logischen Sicht darzustellen.

2.1 Identifikation der Funktionsträger und Funktionsbeschreibungen

Der Entwurfsprozess beginnt mit einer genauen Analyse der Problemstellung, insbesondere des Aspektes der Ablauforganisation. Wesentlich ist dabei die Beantwortung der Frage "Wer soll mit dem Datenbanksystem arbeiten?". Damit ist nicht gemeint "Wer entwirft das Datenmodell?", sondern wer erfasst Datensätze in den einzelnen Tabellen und wer benötigt diese Daten später. Es muss also streng zwischen dem Entwurf und der Implementierung einerseits und der Anwendung andererseits unterschieden werden. Bei diesem Schritt ist es wichtig, sich ein genaues Bild (Aufgaben, Ablauforganisation, Aufgabenträger, usw.) über die Realität zu machen, die in dem Datenmodell abgebildet wird und auch die Zielsetzung des Anwendungsprogramms zu definieren (z.B. Rationalisierung der Arbeitsabläufe im Zusammenhang mit der Kursadministration).

Benutzer des Datenmodells "Kursverwaltung" der Firma KNOWING sind:

Personalabteilung: Festlegen und Erfassen der Kurse, Erfassen der Teilnehmerbeurteilungen, Drucken und Versenden der Teilnehmerbestätigungen, Auskünfte über das Kursangebot erteilen usw.

Mitarbeiter: Online-Information und Anmeldung zu den Kursen.

Abteilungsleiter: Freistellung der Mitarbeiter

Vortragende: erhalten Teilnehmerlisten (werden in der Personalabteilung erstellt)

Nach der Identifikation der Benutzer und der Tätigkeiten, die von ihnen im Zusammenhang mit der Aufgabenstellung wahrgenommen werden, ist zu untersuchen, welche dieser Tätigkeiten als Funktionen durch das Informationssystem unterstützt bzw. automatisiert werden können.

Diese Funktionen sind nun detailliert zu beschreiben. Im Hinblick auf den Entwurf des Datenmodells ist dabei besonders auf die verarbeiteten Informationen Rücksicht zu nehmen. Der Entwurf des Prozess- bzw. des Dialogmodells betonen hingegen die dynamischen, ablauforientierten Aspekte der Funktion.

Die Beschreibung der Funktionen, welche den Mitarbeitern zur Verfügung stehen, kann beispielsweise wie folgt aussehen:

Mitarbeiter erhalten nicht mehr wie bisher ein gedrucktes Kursprogramm zur Verfügung gestellt, sondern können an allen EDV-Arbeitsplätzen im Unternehmen das aktuelle Kursangebot abfragen. Dabei erhalten Sie Informationen über Termin, Ort, Titel und Vortragenden und eventuelle Kosten, die vom Mitarbeiter zu tragen sind.

Die Anmeldung ersetzt das bisherige Anmeldeformular. Der Mitarbeiter identifiziert sich mit seiner Personalnummer und gibt den Kurs an, zu dem eine Anmeldung erfolgen soll. Diese Anmeldung ist vorläufig, sie wird endgültig, wenn der Vorgesetzte des Mitarbeiters die Freistellung genehmigt hat.

2.2 Ableitung von Datenelementen

Für jede Funktion gilt es nun die Datenelemente zu identifizieren. Datenelemente sind Begriffe, die Entitätstypen oder Attribute beschreiben. Der Informationsinhalt eines Datenelements kann in der Funktion entweder entstehen (Informationsquelle) oder benötigt werden (Informationssenke). Für jedes Datenelement in der Datenbank muss es zumindest eine Informationsquelle (Funktion, in der die Information entsteht) und eine Informationssenke geben.

Bei der Suche nach Datenelementen stehen uns folgende Hilfsmittel zur Verfügung:

Allgemeinwissen: Aus der Vorstellung um die Aufgabenstellung lassen sich im Sinne eines Brainstorming bereits Datenelemente anführen. z.B. "Kurstitel", "Kurs", "Mitarbeiter" usw.

Funktionsbeschreibungen: Insbesondere Begriffe (Hauptworte), die mit der Aufgabenstellung unmittelbar in Verbindung stehen. z.B. "Personalnummer", "Termin", "Ort", usw.

Formulare: Die Felder bestehender Formulare aus dem Aufgabenbereich können bei der Suche nach Datenelementen hilfreich sein (Formularanalyse). So kann z.B. aus einem Anmeldeformular, das bisher verwendet wurde ersichtlich werden, dass bei der Anmeldung die Angabe der Personal- und der Kursnummer erforderlich ist.

Gespeicherte Daten in bestehenden Informationssystemen: In Fällen, bei denen es um den Ersatz bestehender EDV-Lösungen geht, kann auf Grundlage der bereits verarbeiteten Daten (Bildschirmmasken, Listen, Datenmodelle) eine Sammlung von Datenelementen erstellt werden.

Das Ergebnis dieser Suche nach Datenelementen ist eine Komponentenliste: Welche Daten existieren in diesem Ausschnitt der Realwelt, und welche Daten benötigen wir tatsächlich?

Komponentenliste: Alle Datentypen, die in die Datenbank aufgenommen werden sollen, werden zunächst möglichst frei (im Sinne eines Brainstormings) gesammelt. Dabei hilft uns einerseits das allgemeine Wissen um die im Anwendungsbereich benötigten Datentypen, anderer-

Datenmodellierung

seits können wir aus den verwendeten Formularen sowie aus den bisher maschinell gespeicherten Daten den Bedarf ableiten.

Mitarbeitername	
Kurs	
Kursbezeichnung	
Kurszeit	
Kursort	
Referentename	
Personalnummer	
Haarfarbe des Mitarbeiters	<--- wird für die Kursverwaltung nicht benötigt!
Adresse des Referenten	
Kursbeurteilung	
...	

Die Aufstellung der Datenelemente für die Benutzer stellt das externe Schema dar. Zu den Datenelementen kann angegeben werden in welchen Funktionen sie verwendet werden und ob es sich bei der Funktion um eine Informationsquelle oder -senke handelt.

Das externe Schema für die Mitarbeiter kann wie folgt dargestellt werden:

Datenelement	Funktionen	
	Kursinformation	Anmeldung
Kurs	S	S
Kurstitel	S	S
Personalnummer		Q
Kursnummer	S	Q
Mitarbeiter		S

In der ersten Spalte werden die Datenelemente angeführt, in den weiteren Spalten wird die Verwendung der Datenelemente in den einzelnen Funktionen analysiert. "S" steht dabei für Informationssenkende, "Q" für Informationsquelle.

Nachdem für alle Benutzer ein externes Schema aufgestellt wurde erfolgt die Überprüfung, ob für jedes Datenelement jeweils zumindest eine Informationsquelle und eine -senke besteht. Am Ende dieses Vorgangs steht eine möglichst vollständige Sammlung von Datenelementen zur Verfügung.

2.3 Zusammengesetzte und elementare Datenelemente

Als nächstes stellt sich die Frage, wie diese Datentypen zusammengefasst bzw. strukturiert werden können.

Bei einer genauen Betrachtung der Datenelemente lassen sich zwei Gruppen unterscheiden:

Arten von Datentypen:

Datenmodellierung

Elementare Datenelemente sind Personalnummer, Kursort, etc. Sie sind nicht weiter zerlegbar; sie haben nur unterschiedliche Werte.(z.B. Für das Datenelement "Personalnummer" existieren die Datenwerte "12345", "67890", etc.)

Zusammengesetzte Datenelemente: Kurs ist ein zusammengesetztes Datenelement, weil es sich in die Datenelemente Kursname, Kursort, etc. zerlegen lässt.

Wenn man diese Erkenntnis auf die Sammlung von Datenelementen anwendet, erhält man eine Hierarchie von Datenelementen. Diese Hierarchie kann in Form einer Attributliste dargestellt werden. (Attributlisten lassen sich auch sehr einfach mit Hilfe von Textverarbeitungsprogrammen erstellen.)

Datentyphierarchie

MITARBEITER

- Mitarbeitername
- Personalnummer
- ...

KURS

- Kursbezeichnung
- Kurszeit
 - Beginnzeit
 - Kursende
- Kursort
- Kursbeurteilung
- ...

REFERENT

- Referentenname
- Adresse
 - Straße
 - Ort
 - Postleitzahl
 - Land
- Fachbereich
- ...

Für die Datenmodellierung benötigen wir eine ganz bestimmte Art von zusammengesetzten Datentypen. Diese heißen *Entitätstypen* (siehe auch oben). Eine Entität ist etwas Reales oder Abstraktes (Personen, Dinge oder Ereignisse), worüber wir Daten speichern wollen.

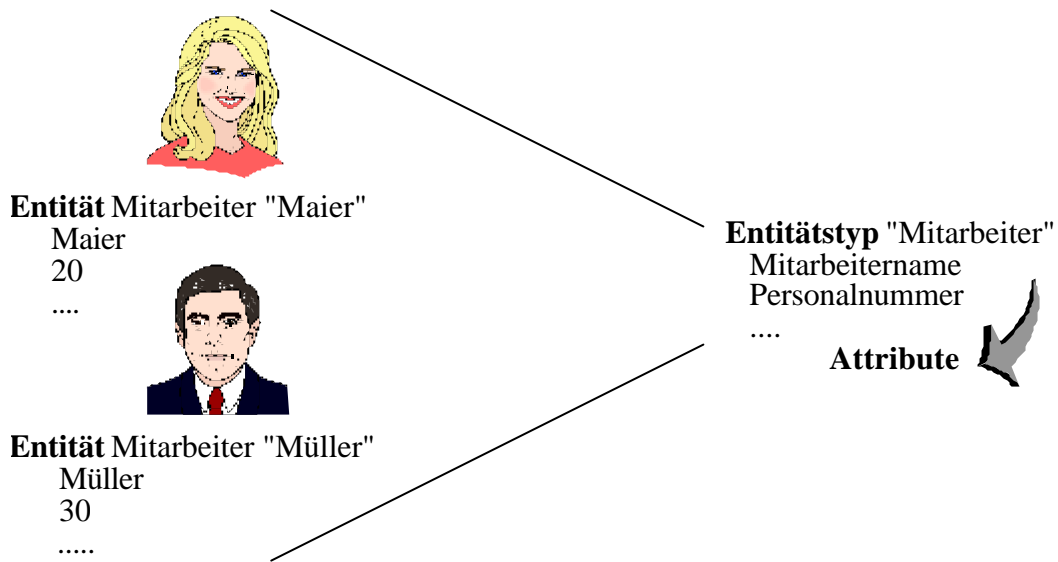
Wir haben in unserem Beispiel mit Hilfe der Datentyphierarchie zunächst die drei Entitätstypen "Mitarbeiter" (=Person), "Kurs" (=Ereignis) und "Referent" (=Person) identifizieren können.

Eine *Entität* ist dabei ein einzelnes Vorkommnis eines Entitätstyps: z.B. der Mitarbeiter Müller ist eine Entität des Entitätstyps Mitarbeiter, ebenso der Herr Maier, etc..

Datenmodellierung

Ein *Entitätstyp* ist eine Zusammenfassung aller Entitäten, welche die gleichen Eigenschaften haben: Alle Mitarbeiter haben einen Namen, eine Personalnummer, etc. und daher die gleichen Eigenschaften.

Der Entitätstyp ist ein zusammengesetzter Datentyp, der nur elementare Datentypen enthält. Diese elementaren Datentypen werden als *Attribute* bezeichnet



2.4 Identifikationsschlüsselvergabe

Jede Entität eines Entitätstyps muss über einen Identifikationsschlüssel eindeutig zu beschreiben sein. Es muss ein Attribut (oder eine Gruppe von Attributen) gefunden werden, die den Anforderungen an einen Identifikationsschlüssel genügt. Dabei sollte als Regel beachtet werden, dass der Identifikationsschlüssel möglichst einfach konstruiert wird. Wenn sich kein Attribut als Identifikationsschlüssel anbietet, wird ein weiteres, künstliches Attribut in den Entitätstyp aufgenommen.

Identifikationsschlüssel: Der Identifikationsschlüssel ist ein Attribut, das jeder Entitätstyp besitzen muss und folgende Eigenschaften besitzt:

Eindeutigkeit: Jeder Entität (Mitarbeiter, Kurs, Referent) ist ein Schlüsselwert (Personalnummer, Kursnummer, Referentennummer) zugeordnet, der anderweitig nie vorkommt. D.h. es darf beispielsweise keine zwei Mitarbeiter geben, welche dieselbe Personalnummer haben. Dies ist oft nur möglich, wenn der Identifikationsschlüssel künstlich ist, er also keine Aussagekraft (=sprechender Schlüssel) hat.

Gültigkeitszeitraum: Der Identifikationsschlüssel darf sich während der gesamten Lebenszeit einer Entität nicht ändern.

D.h. ein Mitarbeiter behält, solange er Mitarbeiter ist, immer dieselbe Personalnummer. Auch aus diesem Grund sollte ein Identifikationsschlüssel nicht sprechend sein, weil sich ja dann der Schlüsselwert ändern könnte.

Neue Entitäten: Jede neu auftretende Entität (Mitarbeiter, ...) bekommt sofort ihren eindeutigen Schlüsselwert.

Einführung eines Identifikationsschlüssels für jeden Entitätstyp: Wir weisen unseren drei Entitätstypen folgende Identifikationsschlüssel zu, die auch die oben genannten Eigenschaften besitzen:

- Mitarbeiter: Personalnummer
- Kurs: Kursnummer
- Referent: Referentennummer

Der Identifikationsschlüssel wird in den zwei bisher verwendeten Darstellungsformen - nämlich der Attributliste und der Tabelle - durch Unterstreichen gekennzeichnet.

Identifikationsschlüssel in der Attributliste:

MITARBEITER

Personalnummer
Mitarbeitername
...

KURS

Kursnummer
Kursbezeichnung
Beginnzeit
Kursende
Kursort
Kursbeurteilung
...

REFERENT

Referentennummer
Referentenname
Strasse
Ort
Postleitzahl
Land
Fachbereich
...

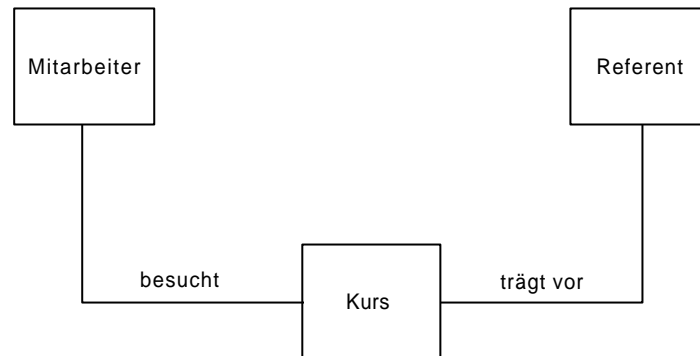
2.5 Beziehungen zwischen den Entitätstypen

Die bisher gebildeten Entitätstypen stehen in der Realität miteinander in Beziehung. Referenten tragen in den Kursen vor, die von den Mitarbeitern besucht werden. Dieser Sachverhalt wurde in den bisherigen Darstellungen nicht berücksichtigt.

Um diese Beziehungen darzustellen wird eine weitere Darstellungsform des Datenmodells eingeführt: Das Datenbank-Strukturdiagramm (Entity-Relationship Diagramm, ER-Diagramm). Die Darstellung der Entitätstypen erfolgt dabei mit grafischen Symbolen. Die Beziehungen zwischen den Entitätstypen werden durch Linien zwischen den Symbolen dargestellt. Die Aussage

die durch die Beziehungen repräsentiert werden, werden verbal in das Diagramm aufgenommen.⁹

Für die Kursverwaltung stellt sich das ER-Diagramm zunächst wie folgt dar:



Mitarbeiter besuchen Kurse, und Kurse werden von Mitarbeitern besucht

Kurse werden von Referenten vorgetragen, und Referenten tragen Kurse vor

Zwischen Mitarbeitern und Referenten besteht keine direkte Beziehung, sie haben nur mittelbar - nämlich über den Kurs - miteinander zu tun

2.6 Beziehungsarten

Es existieren grundsätzlich drei verschiedene Beziehungsarten.

1:n - Beziehung (eins zu keinem, einem oder mehrere):

Ein Mitarbeiter - kann er keinen, einen oder mehrere Kurse besuchen?

1:1 - Beziehung (eins zu eins):

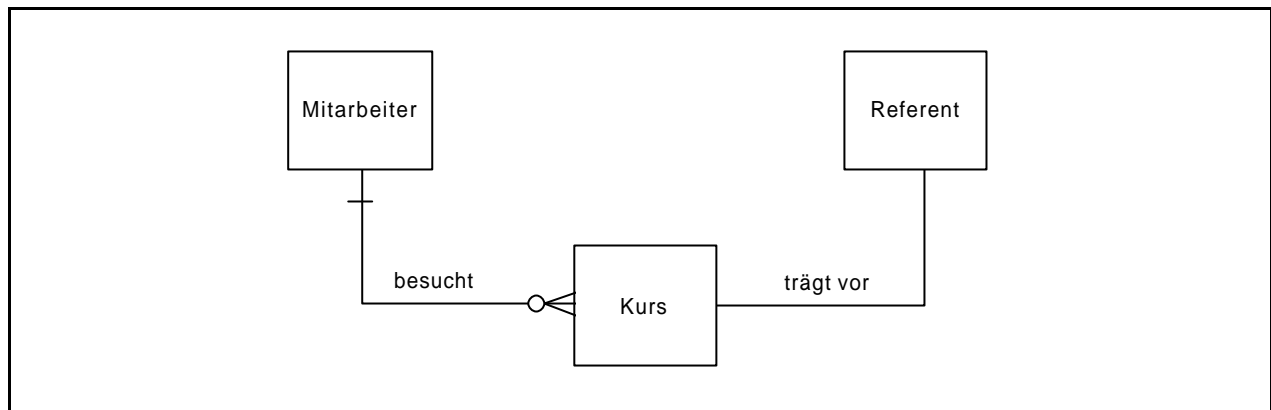
Ein Mitarbeiter - besucht er genau einen Kurs?

1:c - Beziehung (eins zu keinem oder einem):

Ein Mitarbeiter - kann er einen oder keinen Kurs (nicht aber mehrere Kurse) besuchen?

In unserem Beispiel ist es so, dass ein Mitarbeiter keinen, einen oder mehrere Kurse besuchen kann: Daher besteht zwischen dem Entitätstyp Mitarbeiter und dem Entitätstyp Kurs eine 1:n – Beziehung (n wird in der Darstellung z. B. durch einen „Krähenfuß“ dargestellt).

⁹ Die hier gewählte Darstellungsform geht konform mit üblichen Datenmodellierungswerkzeugen und orientiert sich an der CHEN – Notation. Es gibt jedoch noch eine Reihe anderer Notationsformen (z.B. Klassendiagramm in der UML-Notation), die aus pragmatischen Gründen hier nicht behandelt werden.



Beziehungen sind immer wechselseitig zu betrachten:

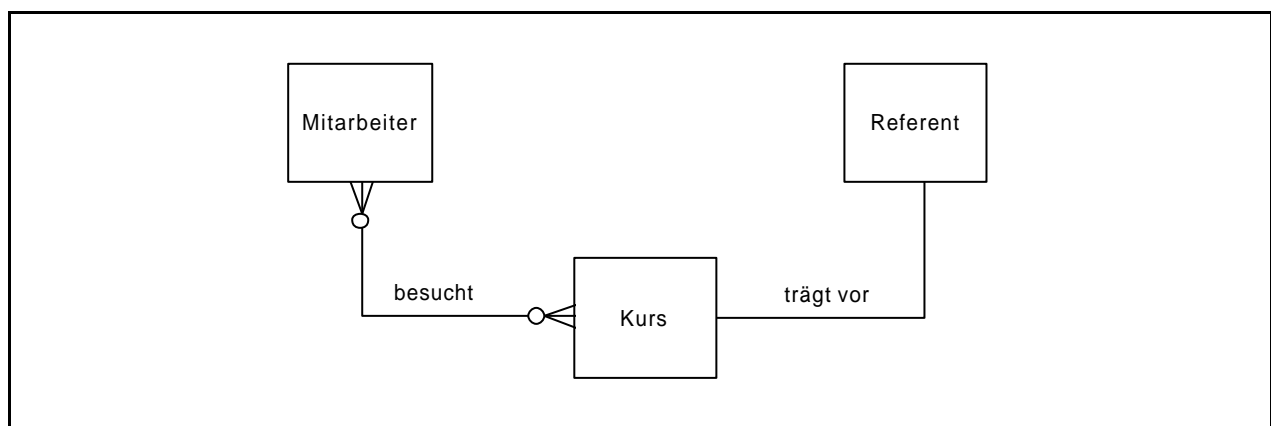
- Mitarbeiter hat einerseits zu Kurs eine Beziehung - nämlich eine 1:n - Beziehung,
- umgekehrt besteht aber auch zwischen Kurs und Mitarbeiter eine bestimmte Beziehungsart.

So sind jeweils die umgekehrt formulierten Fragen zu beantworten:

- Ein Kurs - kann er von keinem, einem oder mehreren Mitarbeitern besucht werden (1:n - Beziehung) ?
- Ein Kurs - wird er von genau einem Mitarbeiter besucht (1:1 - Beziehung) ?
- Ein Kurs - wird er von einem oder keinem Mitarbeiter besucht (1:c - Beziehung) ?

Beziehungen zwischen Entitätstypen sind immer wechselseitig: Daher sind die drei Fragen, mit deren Hilfe festgestellt werden kann, um welche Beziehungsart es sich handelt, immer aus der Sicht von beiden beteiligten Entitätstypen zu beantworten.

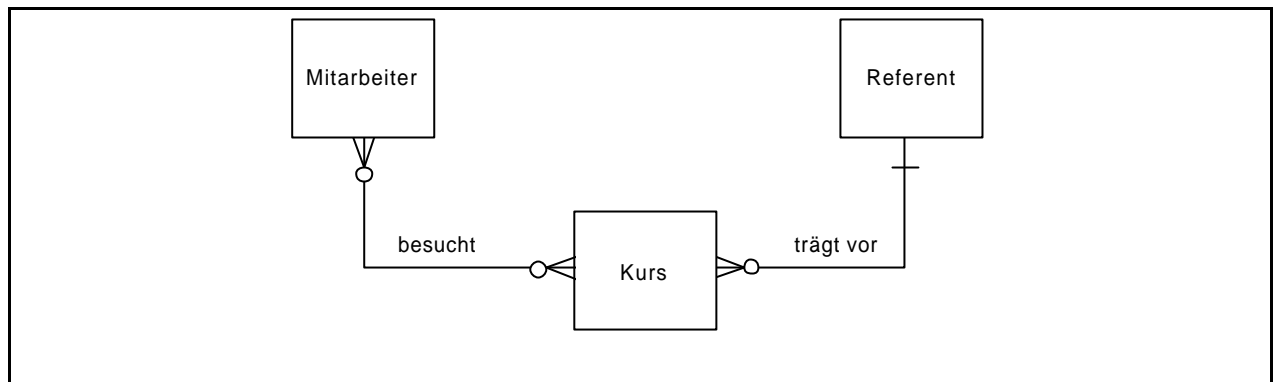
Insgesamt ergibt sich somit zwischen Mitarbeiter und Kurs eine n:n - Beziehung (richtiger eine n:m - Beziehung).



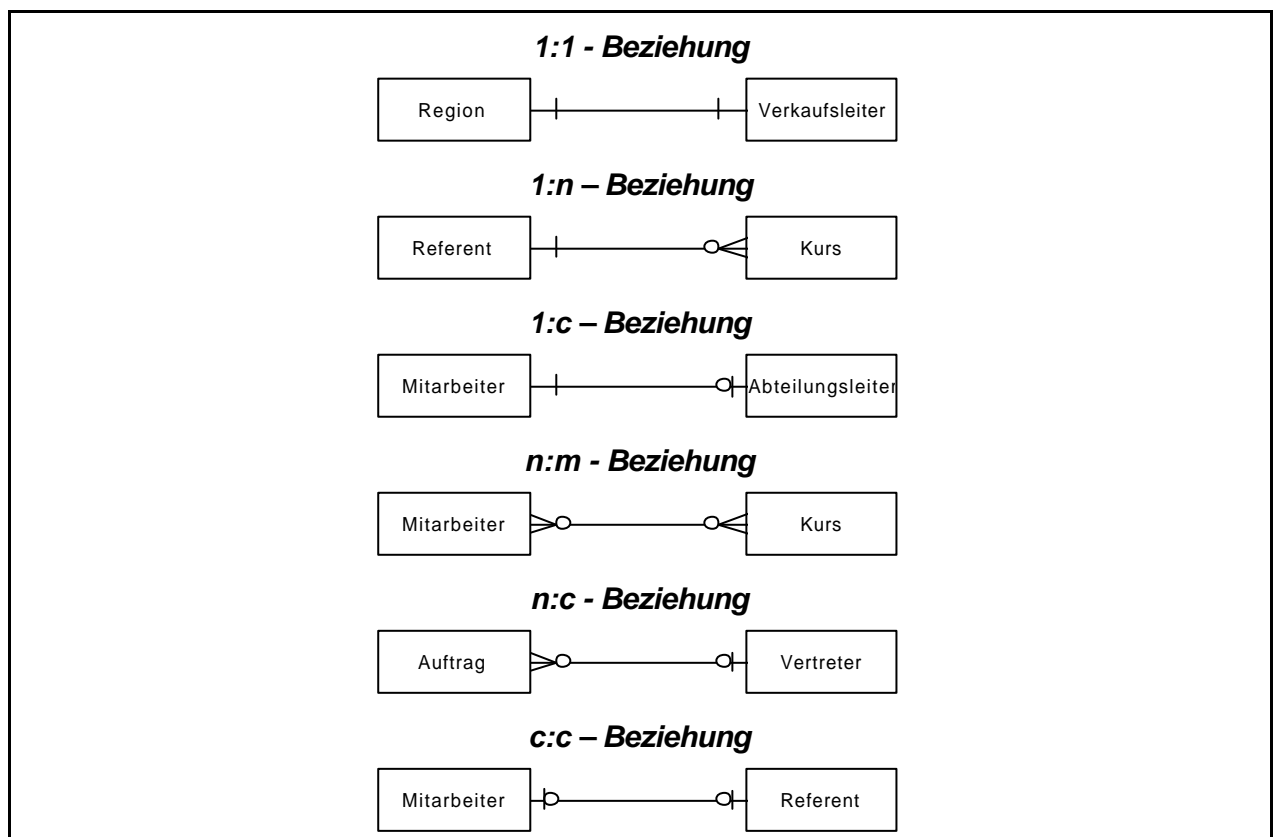
Datenmodellierung

Das selbe Prozedere ist nun für die beiden Entitätstypen Referent und Kurs durchzuführen, um die Beziehungsarten festzustellen. Als Antwort auf die Fragen ergibt sich, dass

- ein Referent keinen, einen oder mehrere Kurse abhalten kann,
- ein Kurs genau von einem Referenten geleitet wird.



Mögliche Kombinationen von Beziehungen: Insgesamt ergeben sich aus den drei Beziehungsarten sechs verschiedene Kombinationsmöglichkeiten.



Datenmodellierung

Beziehungsarten werden meist wie folgt im Datenbankstrukturdiagramm dargestellt:

n - Beziehung durch einen Krähfuß

1 - Beziehung durch einen einfachen kurzen Querstrich

c - Beziehung durch einen Kreis mit einfachem kurzen Querstrich

Damit sind in unserem Beispiel alle Beziehungen im Datenbankstrukturdiagramm identifiziert und klassifiziert worden. Diese Beziehungen lassen sich auch in den Darstellungsformen Attributliste und Tabelle darstellen.

Tabellen ohne Beziehungen: Bei den bisher entwickelten Tabellen wurden die Beziehungen zwischen einzelnen Entitätstypen nicht berücksichtigt. D.h. wir können aus den Tabellen zwar ablesen, welchen Namen der Referent mit der Referentennummer 12345 hat oder wo der Referent Schulz wohnt. Wir können aber nicht ablesen, welche Kurse der Referent Schulz leitet oder umgekehrt, welcher Referent den Kurs Datenmodellierung hält.

REFERENT	<u>Referentennummer</u>	Referentenname	Strasse	...
	12345	Schulz	Parkgasse 13	...
	67890	Bersten	Am Berg 11	...

KURS	<u>Kursnummer</u>	Kursbezeichnung	Beginnzeit	Kursende	...
	666	Datenmodellierung	08:30	17:45	...
	777	JAVA - Anfänger	08:00	16:00	...

Für das Ausdrücken von Beziehungen in Tabellen verwendet man die zuvor eingeführten Identifikationsschlüssel. Grundsätzlich ist es so, dass in Datenbanken jede Information nur einmal vorhanden sein darf, d.h. es darf nicht zweimal in den Tabellen die Information vorhanden sein, dass Schulz in der Parkgasse wohnt oder der Referent mit der Nummer 12345 Schulz heißt, denn diese Daten können sich ändern: Herr Schulz wechselt den Wohnort oder nimmt einen neuen Namen an, etc.

Der Identifikationsschlüssel aber hat die Eigenschaft, dass er sich niemals ändern kann und immer dieselbe Entität identifiziert. Daher kann der Identifikationsschlüssel mehrmals gespeichert werden.

Beziehungen in Tabellen: Beziehungen werden in Tabellen über die mehrmalige Nutzung des Identifikationsschlüssels ausgedrückt, wobei als Regel gilt: bei dem Entitätstyp, bei dem eine n- oder c-Beziehung vorliegt, wird der Identifikationsschlüssel des jeweils anderen Entitätstyps als zusätzliches Attribut aufgenommen.

Datenmodellierung

In unserem Beispiel ist zwischen den Entitätstypen Referent und Kurs eine 1:n - Beziehung, wobei die n - Beziehung bei Kurs ist. Deshalb wird der Identifikationsschlüssel von Referent (=Referentennummer) als zusätzliches Attribut beim Entitätstyp Kurs aufgenommen.

Beim Entitätstyp Referent liegt eine 1 - Beziehung vor und deshalb wird der Identifikationsschlüssel von Kurs (=Kursnummer) nicht als zusätzliches Attribut von Referent aufgenommen.

REFERENT	<u>Referentennummer</u>	Referentenname	Strasse	...
	12345	Schulz	Parkgasse 13	...
	67890	Bersten	Am Berg 11	...

KURS	<u>Kursnummer</u>	Referentennummer	Kursbezeichnung	...
	666	12345	Datenmodellierung	...
	777	67890	JAVA - Anfänger	...
	888	67890	JAVA - Fortgeschrittene	...

Lesen von Beziehungen in Tabellen: Nun können auch Informationen, die mehrere Entitätstypen betreffen, aus den Tabellen abgelesen werden.

Will man etwa wissen, welche Kurse der Referent Bersten leitet, ergibt sich folgender Ablauf:

- Unter der Tabellenspalte des Attributs Referentenname des Entitätstyps Referent sucht man den Namen "Bersten".
- Bersten hat die Referentennummer 67890.
- Man sucht diese Referentennummer beim Entitätstyp Kurs und findet sie dort zweimal.

Von dort aus lassen sich dann auch sehr leicht die Kursbezeichnungen der gewünschten Kurse ablesen.

REFERENT	<u>Referentennummer</u>	Referentenname	Strasse	...
	12345	Schulz	Parkgasse 13	...
	67890	Bersten	Am Berg 11	...

KURS	<u>Kursnummer</u>	Referentennummer	Kursbezeichnung	...
	666	12345	Datenmodellierung	...
	777	67890	Java - Anfänger	...
	888	67890	Java - Fortgeschrittene	...

Grundsätzlich gilt das gleiche Vorgehen für die Beschreibung der Beziehung zwischen den Entitätstypen Mitarbeiter und Kurs. Dort ergibt sich aber ein spezielles Problem, auf das etwas später bei der Normalisierung von Beziehungen eingegangen wird.

Die Darstellung von Beziehungen zwischen Entitätstypen bei Attributlisten erfolgt gleich wie bei den Tabellen: Bei dem Entitätstyp, bei dem eine n- oder c-Beziehung vorliegt, wird der Identifikationsschlüssel des jeweils anderen Entitätstyps als zusätzliches Attribut aufgenommen. Zur Kennzeichnung, dass es sich bei diesem Attribut um einen "Fremdschlüssel" bzw. um ein "globales Attribut" handelt, wird dies durch einen Pfeil und die Angabe des referenzierten Entitätstyps kommentiert.

Globales Attribut: Ein Attribut ist ein globales Attribut, wenn es an mehreren Orten im Datenmodell vorkommt. Es dürfen aber nur Identifikationsschlüssel an mehreren Orten eingesetzt werden. Globale Attribute sind daher immer Identifikationsschlüsselattribute.

Globale Attribute heißen Fremdschlüssel, wenn sie im Entitätstyp nicht für die Identifikation, sondern für die Referenz verwandt werden.

Lokales Attribut: Ein Attribut ist lokal, wenn es nirgendwo als Identifikationsschlüssel verwendet wird.

REFERENT

Referentennummer
Referentenname
Strasse
Ort
Postleitzahl
Land
Fachbereich
...

KURS

Kursnummer
Referentennummer --> Referent
Kursbezeichnung
Beginnzeit
Kursende
Kursort
Kursbeurteilung
...

2.7 Datentypen und Normalisierung

Ziel der Datentypenhierarchie ist eine klare Struktur der Daten. Diese Struktur soll vermeiden, dass Informationen in der Datenbank mehrfach gespeichert sind (**Redundanz**) und damit die Gefahr von Inkonsistenz der Datenbank besteht. Dieser Prozess wird auch als **Normalisierung** bezeichnet. Die Normalisierung hat das Ziel, die Redundanz innerhalb von Datenbeständen zu vermeiden. Redundanz kann zwei Ursachen haben:

Ein Attribut, das kein globales Attribut ist, wird mehrmals abgespeichert. Solche lokalen Attribute dürfen daher im gesamten Datenmodell nur einmal vorkommen (siehe 1. Und 3. Normalform)

Beziehungen werden mehrfach abgebildet (2. Normalform).

Theoretisch existieren 5 Normalformen, in der Regel werden nur die ersten drei Normalformen benötigt.

Eine Attributliste befindet sich in der ersten Normalform, wenn für jedes Attribut einer Entität genau ein Wert angegeben werden kann.

Ist das nicht der Fall, so müssen die betroffenen Attribute in einen eigenen Entitätstyp ausgegliedert werden. Folgendes Beispiel soll dies veranschaulichen:

Datenmodellierung

Der Entitätstyp REFERENT soll die Bezeichnung der KURSE dieses Referenten aufnehmen:

REFERENT

Referentenname

Adresse

.....

Kurse

Referentenname	Adresse	Kurse
Jarz	Innsbruck	Systemplanung 1, Systemplanung 2, Informationsmanagement

Für das Attribut Kurs sind in diesem Fall mehrere Werte möglich. Das Beispiel entspricht daher nicht der 1. Normalform.

Ein weiterer Grund, der eine Auslagerung in einen eigenen Entitätstyp erfordert, liegt vor, wenn auf den Zwischenebenen der Attributliste weitere Entitätstypen beschrieben werden. In diesem Fall wird von einem Verstoß gegen die 3. Normalform gesprochen.

KURS

Kursbezeichnung

Kurszeit

.....

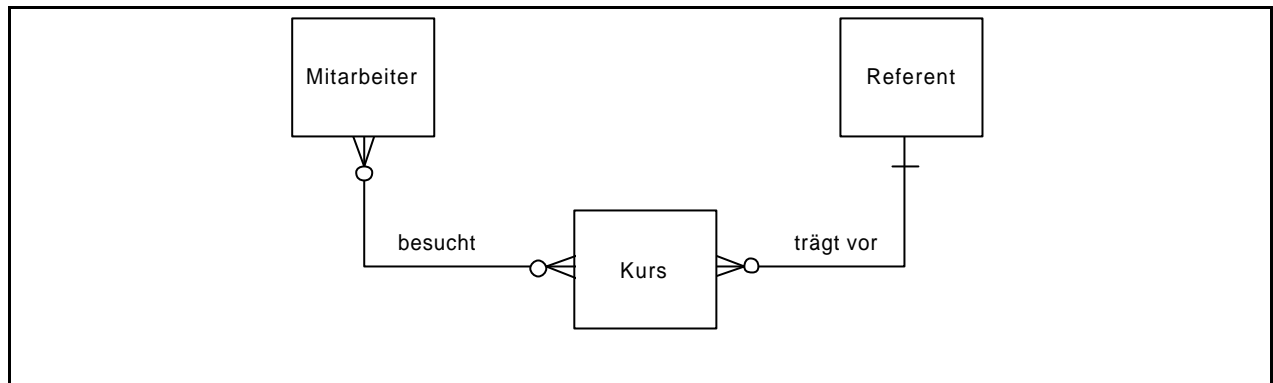
Referent

Referentenname

Adresse

Zunächst kann festgehalten werden, dass die erste Normalform nicht verletzt wird, bei jedem KURS kann eindeutig festgestellt werden, von welchem Referenten er gehalten wird (siehe Annahme in der Angabe, dass ein Kurs nur von einem Referenten gehalten wird!). Wenn der Entitätstyp KURS jedoch in dieser Form belassen wird, sind die gleichen Referentendaten bei unterschiedlichen Kursen zu erfassen (Redundanz). Zusätzlich ist es möglich, Einträge in die Tabelle vorzunehmen, bei denen die Kombination Referentenname - Adresse widersprüchlich ist. Für Referenten ist daher ein eigener Entitätstyp erforderlich.

In unserem Beispiel konnten wir zwischen den Entitätstypen Mitarbeiter und Kurse eine n:m-Beziehung feststellen.



Alle nicht 1: Beziehungen, also n:m, n:c und c:c, können nicht so einfach in den Tabellen abgebildet werden. Nach der Regel, dass der Identifikationsschlüssel beim anderen Entitätstyp als Fremdschlüssel einzufügen ist kann nicht vorgegangen werden, weil keine Seite der Beziehung eine "1" aufweist. Ein wechselseitiges Einfügen der Identifikationsschlüssel würde dazu führen, dass die Beziehung doppelt abgebildet wäre, was eine unzulässige Redundanz darstellen würde.

Wenn man auf diese Tabellen die im vorigen Kapitel aufgestellte Regel über die Abbildung einfacher Beziehungen anwenden würde ergäbe sich folgende Situation:

- Im Entitätstyp Mitarbeiter ist der Identifikationsschlüssel des Entitätstyps Kurs (=Kursnummer) als zusätzliches Attribut aufzunehmen, weil hier eine n-Beziehung vorliegt.
- Im Entitätstyp Kurs ist der Identifikationsschlüssel des Entitätstyps Mitarbeiter (=Personalnummer) als zusätzliches Attribut aufzunehmen, weil hier ebenfalls eine n-Beziehung vorliegt.

Beispiel:

- Mitarbeiter Müller besucht den Kurs Datenmodellierung und den Kurs JAVA-Anfänger.
- Mitarbeiter Maier besucht den Kurs JAVA-Anfänger.
- Für den Kurs JAVA-Fortgeschrittene hat sich bislang noch niemand angemeldet.

Datenmodellierung

MITARBEITER	<u>Personalnummer</u>	Kursnummer	Mitarbeitername	...
	20	12345	Müller	...
	20	67890	Müller	...
	30	67890	Maier	...

KURS	<u>Kursnummer</u>	Referentennummer	<u>Personalnummer</u>	Kursbezeichnung	...
	666	12345	20	Datenmodellierung	...
	777	67890	20	JAVA - Anfänger	...
	777	67890	30	JAVA - Anfänger	...
	888	67890	????????	JAVA - Fortgeschrittene	...

Es ist leicht zu erkennen, dass sich hier einige Anomalien (Ungereimtheiten) ergeben:

- Jede Beziehung wird zweimal abgespeichert: Dass Müller den Kurs Datenmodellierung besucht, wird zweimal abgespeichert - nämlich einmal in der Tabelle Mitarbeiter und einmal in der Tabelle Kurs.
- Die Daten über eine Entität müssen mehrmals abgespeichert werden, wenn mehrere Beziehungen existieren: Müller besucht zwei Kurse, daher müssen die Daten über seine Person zweimal abgespeichert werden. Dies führt dazu, dass Identifikationsschlüssel nicht mehr eindeutig sind. Wir müssen den Identifikationsschlüssel um das Attribut "Personalnummer" erweitern.
- Wir wissen nicht, wie man ausdrücken soll, dass eine Entität des einen Typs keine Beziehung zu einer Entität des anderen Typs hat: Der Kurs JAVA-Fortgeschr. wird nicht besucht. Da die Personalnummer jetzt Teil des Identifikationsschlüssels ist, muss für sie sofort ein Wert vergeben werden. Die Attribute des Entitätstyps Kurs sind nur von der Kursnummer nicht aber von der Personalnummer abhängig. Es liegt somit ein Verstoß gegen die **2. Normalform** vor.

Analoges gilt natürlich für den Entitätstyp Mitarbeiter.

Anomalien treten immer dann auf, wenn es sich um eine Beziehung handelt, bei der auf keiner der beiden Seiten eine 1-Beziehungsart vorliegt. Also bei den Beziehungen

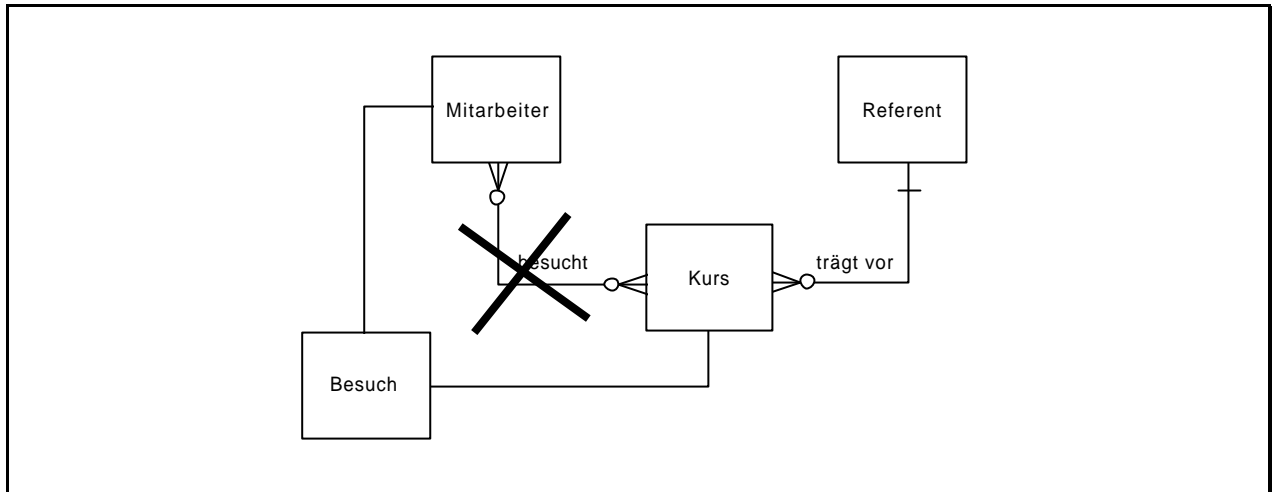
n:m

n:c und

c:c.

Gelöst wird dieses Problem durch die Einführung eines neuen Entitätstyps.

Normalisierung von Beziehungen: Beziehungen, die auf keiner Seite eine 1-Beziehung haben, müssen aufgelöst bzw. normalisiert werden. Dies erreicht man, indem man anstelle dieser Beziehung einen neuen Entitätstyp einführt, der mit beiden beteiligten Entitätstypen in Beziehung steht.

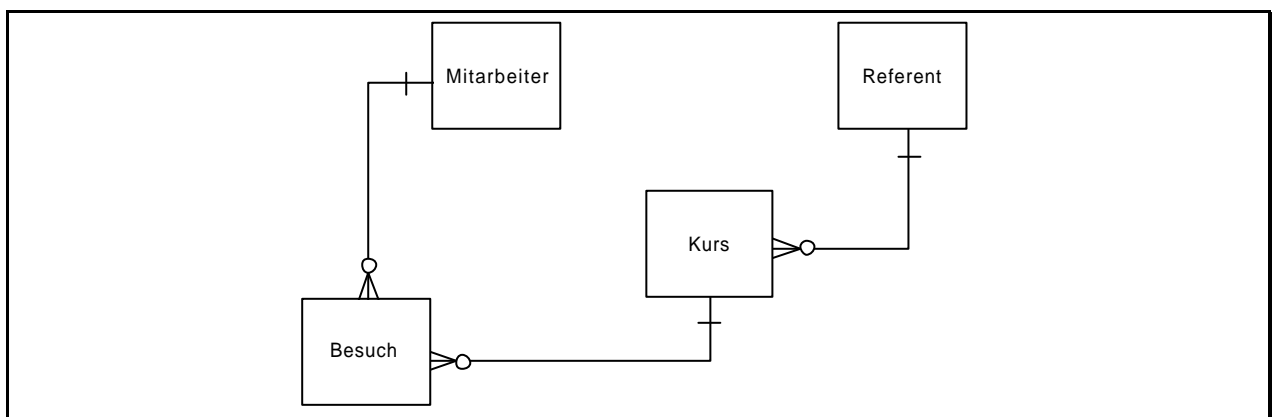


Die Arten der Beziehungen bleiben dabei grundsätzlich gleich:

Vom Entitätstyp Kurs aus betrachtet: Es besteht eine 1:n - Beziehung zu Mitarbeiter, daher besteht auch eine 1:n - Beziehung von Kurs zu dem neuen Entitätstyp (Besuch).

Vom Entitätstyp Mitarbeiter aus betrachtet: Es besteht eine 1:n - Beziehung zu Kurs, daher besteht auch eine 1:n - Beziehung von Mitarbeiter zu dem neuen Entitätstyp (Besuch).

Normalisiertes Datenbankstrukturdiagramm: Das Datenbankstrukturdiagramm ist dann normalisiert, wenn es nur noch 1: ... - Beziehungen enthält.



Der neue Entitätstyp ist so zu behandeln wie alle anderen:

- Es muss ein Name für ihn vergeben werden: BESUCH
- Es ist zu überlegen, welche Attribute dem Entitätstyp Besuch zuzuordnen sind: Beispielsweise könnte hier das Attribut Mitarbeiterbeurteilung aufgenommen werden:

Datenmodellierung

BESUCH

Mitarbeiterbeurteilung

- Es ist ein Identifikationsschlüssel zu vergeben.

BESUCH

Besuchnummer

Mitarbeiterbeurteilung

- Es sind die Beziehungen mit Hilfe der Identifikationsschlüssel darzustellen, wobei wiederum die Regel angewandt wird, dass bei α - oder n -Beziehungen der Identifikationsschlüssel des jeweils anderen Entitätstyps als zusätzliches Attribut aufzunehmen ist.

BESUCH

Besuchnummer

Mitarbeiterbeurteilung

Kursnummer

--> Kurs

Personalnummer

--> Mitarbeiter

- Änderung des Identifikationsschlüssels: Identifikationsschlüssel können sich auch aus mehreren Attributen zusammensetzen..

BESUCH

Kursnummer

--> Kurs

Personalnummer

--> Mitarbeiter

Mitarbeiterbeurteilung

Durch das bisher beschriebene Vorgehen wurden systematisch alle Verstöße gegen die 1., 2. und 3. Normalform beseitigt. Dieser Vorgang wird als **Normalisierung** des Datenmodells bezeichnet. Das Ergebnis ist ein Datenmodell in der 3. Normalform. Eine vollständige Normalisierung führt zu einem Datenmodell in der 5. Normalform. In der Praxis zeigt sich jedoch, dass die 3. Normalform in den meisten Anwendungen völlig ausreicht.

Folgende Bedingungen müssen erfüllt sein, damit ein Datenmodell in der 3. Normalform vorliegt:

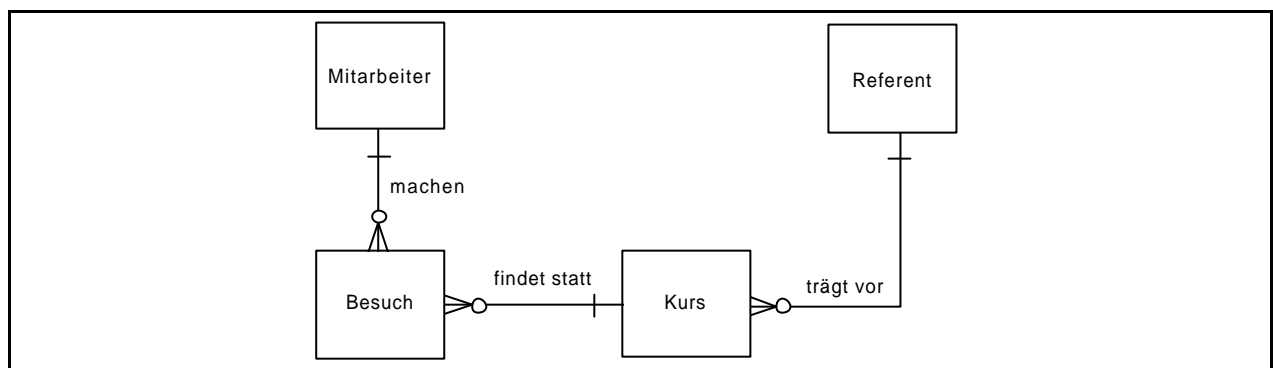
1. Normalform: Jedes Attribut eines Datensatzes enthält genau einen Wert. Wiederholungen müssen durch Auslagerung der Information in einen eigenen Entitätstyp beseitigt werden.
2. Normalform: Alle Attribute müssen vom gesamten Identifikationsschlüssel abhängig sein. Attribute, die nur von einem Teil des Identifikationsschlüssel abhängig sind, müssen in einen eigenen Entitätstyp zusammengefasst werden. Ein Verstoß gegen die 2. Normalform kann nur bei Entitätstypen vorkommen, die einen aus mehreren Attributen bestehenden Identifikationsschlüssel haben und mindestens ein weiteres Attribut aufweisen. Ein Verstoß gegen die 2. Normalform entsteht beim hier vorgestellten Vorgehen durch falsch abgebildete wechselseitige Beziehungen.
3. Normalform: Alle Attribute eines Entitätstyps müssen direkt vom Identifikationsschlüssel abhängen, sie dürfen von keinem anderen Attribut abhängig sein. Verstöße

gegen die 3. Normalform lassen sich in der Datenelementehierarchie erkennen, wenn ein Entitätstyp mehrere Hierarchieebenen enthält und die unteren Ebenen identifizierbare Entitätstypen repräsentieren.

Aus Performancegründen wird in manchen Fällen auf eine vollständige Normalisierung bis zur 3. Normalform verzichtet. Dies geschieht aber immer, nachdem ein Datenmodell zumindest in der 3. Normalform entwickelt wurde. Dieser Vorgang wird als **Denormalisierung** bezeichnet. Denormalisierung betrifft ausschließlich die 3. oder höhere Normalformen. Verstöße gegen die 1.- und 2. Normalform führen zu schwerwiegenden Anomalien sodass eine konsistente Datenehaltung nicht mehr möglich wäre.

2.8 Musterlösung für die Aufgabenstellung "Kursverwaltung":

ER-Diagramm: Im ER-Diagramm (Datenbankstrukturdiagramm) sind die Entitätstypen und die Beziehungen zwischen den Entitätstypen dargestellt.



Attributliste: In der Attributliste sind die Attribute, die Identifikations- und die Fremdschlüssel der einzelnen Entitätstypen dargestellt.

REFERENT

Referentennummer
Referentenname
Strasse
Ort
Postleitzahl
Land

KURS

Kursnummer

Referentennummer --> Referent

Kursbezeichnung

Beginnzeit

Kursende

Kursort

BESUCH

Kursnummer --> Fremdschlüssel zu Kurs

Personalnummer --> Fremdschlüssel zu Mitarbeiter

Mitarbeiterbeurteilung

Freigabe

MITARBEITER

Personalnummer

Mitarbeitername

Tabellen: In den Tabellen wird mit Beispielesdaten dargestellt, dass das Modell redundanzfrei und normalisiert ist, und dass die für die Aufgabenstellung benötigten Informationen abgebildet werden können.

Folgende Informationen sind als Beispielesdatensätze in den Tabellen dargestellt:

- Mitarbeiter Müller besucht den Kurs Datenmodellierung und den Kurs JAVA-Anfänger.
- Mitarbeiter Maier besucht den Kurs JAVA-Anfänger.
- Für den Kurs JAVA-Fortgeschrittene hat sich bislang noch niemand angemeldet.

MITARBEITER	<u>Personalnummer</u>	Mitarbeitername	Strasse	...
	20	Müller	Parkgasse 13	...
	30	Maier	Am Berg 11	...

BESUCH	Personalnummer	Kursnummer	Mitarbeiterbeurteilung	...
	20	666	Bestens	...
	20	777	Ganz gut	...
	30	777	Könnte besser sein	...

KURS	<u>Kursnummer</u>	Referentennummer	Kursbezeichnung	...
	666	12345	Datenmodellierung	...
	777	67890	JAVA - Anfänger	...
	888	67890	JAVA - Fortgeschrittene	...

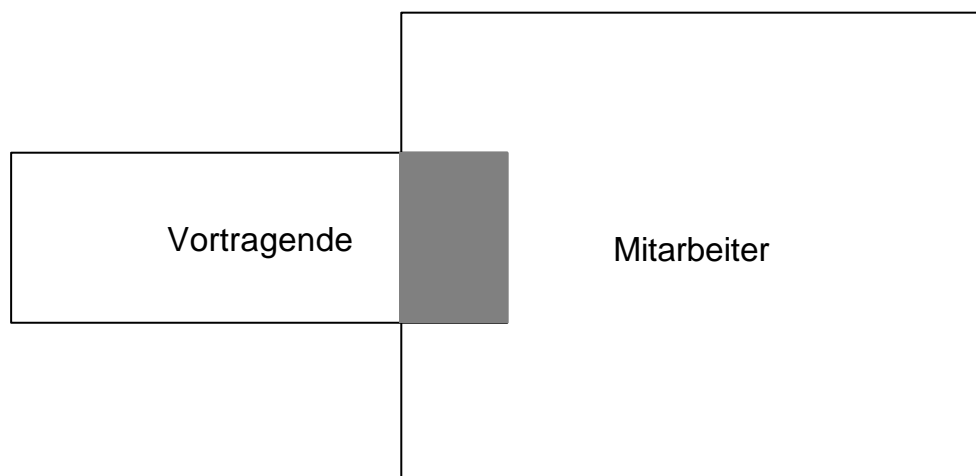
Auf der Basis von ER-Diagramm und Attributliste kann nun mit der Entwicklung des **internen Schemas** begonnen werden. Das interne Schema muss auf das aktuelle RDBMS-Produkt abgestimmt sein. Aus diesem Grund wird hier nicht weiter darauf eingegangen.

2.9 Weiterführende Konzepte des relationalen Datenmodells

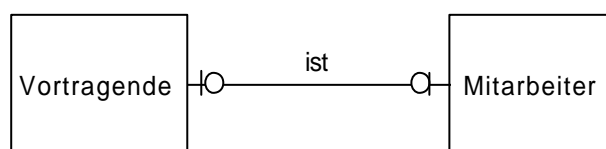
2.9.1 Generalisierung

Die Generalisierung berücksichtigt das Phänomen sich überlappender Mengen im Datenmodell. Im Datenmodell stellen sich solche Situationen in der Form von c:c Beziehungen mit der Aussage "... ist ein oder ist kein ..." dar.

Ein Mitarbeiter der Firma KNOWING kann durchaus selbst Kurse halten und somit auch Vortragender sein (überlappende Mengen).



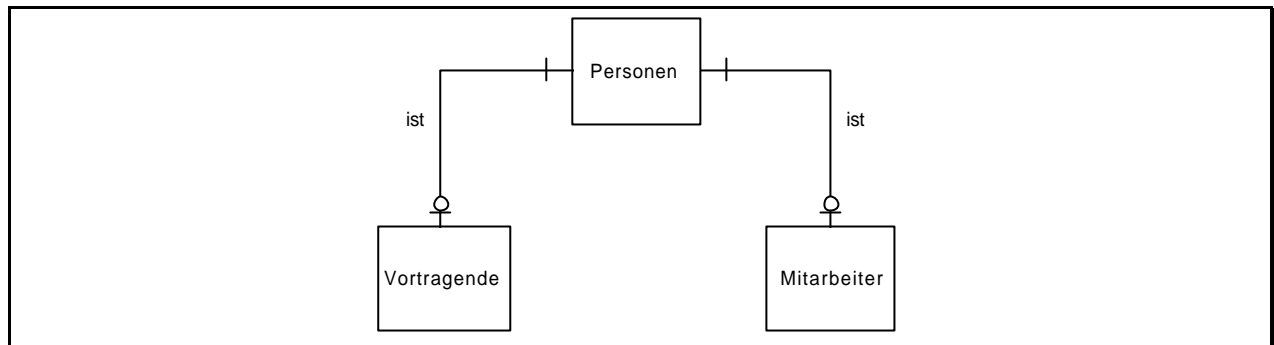
Im ER-Diagramm lässt sich dieser Sachverhalt als c:c - Beziehung darstellen:



- Ein Mitarbeiter ist 0,1 Vortragender
- Ein Vortragender ist 0,1 Mitarbeiter

Datenmodellierung

Nach der bekannten Regel: Einführung eines neuen Entitätstyps lässt sich auch diese Beziehung auflösen. Die Auflösung erfolgt mittels des Entitätstyps PERSONEN. Lediglich die Beziehungen zu diesem neuen Entitätstyp müssen genau untersucht werden.



Die in diesem ER-Diagramm dargestellten Beziehungen sagen aus:

Ein Mitarbeiter	ist	1	Person
Eine Person	ist	0,1	Mitarbeiter
Ein Vortragender	ist	1	Person
Eine Person	ist	0,1	Vortragender

In der Attributliste werden diese Beziehungen wie gewohnt abgebildet:

PERSONEN

Personennr

Name

Strasse

Ort

...

MITARBEITER

Personalnummer

Personennr --> Fremdschlüssel zu Personen

Gehalt

VORTRAGENDE

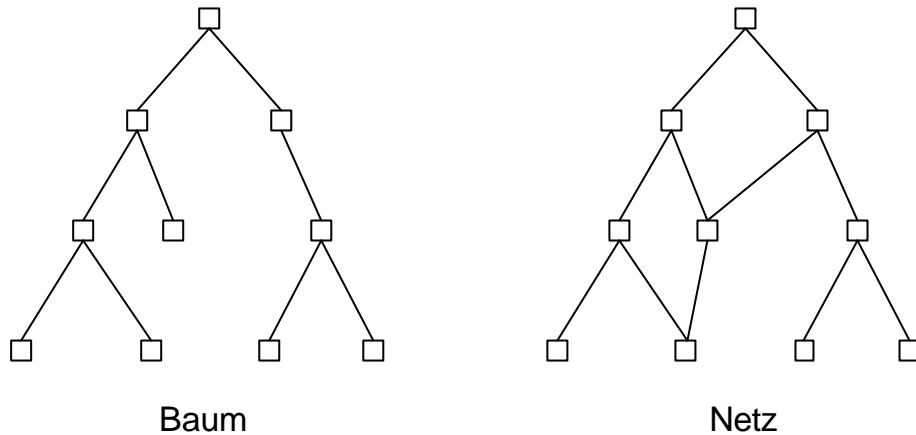
Vortragendennummer

Personennr --> Fremdschlüssel zu Personen

Bei der Generalisierung werden jene Attribute, die in beiden Entitätstypen vorkommen, in den neuen Entitätstyp übernommen.

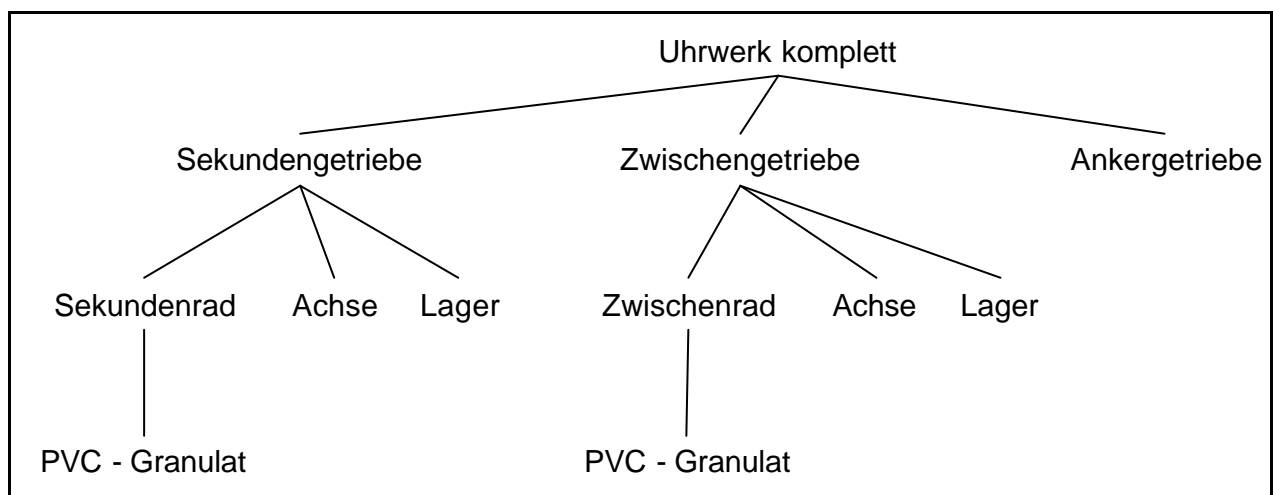
2.9.2 Auflösen von Hierarchien

Ein Problemfall beim Entwurf relationaler Datenmodelle besteht in der Abbildung hierarchischer Strukturen. Bei diesen Strukturen bestehen zwei verschiedene Typen:



Ein Baum ist dadurch gekennzeichnet, dass ein Element auf den unteren Hierarchieebenen (Sohn) nur ein übergeordnetes Element (Vater) haben kann, während bei einem Netz mehrere Väter zulässig sind. Beispiele für diese beiden Grundtypen einer Hierarchie sind das Organigramm eines Unternehmens (reine Linienorganisation) als Beispiel eines Baumes und die Strukturstückliste eines Produktes als Beispiel für eine Netz.

Die folgende Darstellung zeigt vereinfacht die Bestandteile eines Uhrwerks. Achse, Lager und PVC-Granulat werden in mehreren Bestandteilen verwendet, es handelt sich also um eine netzartige Hierarchie.

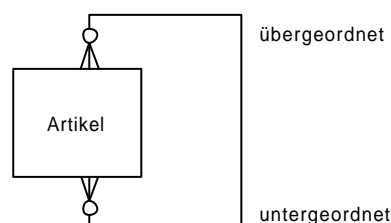


Als Stückliste dargestellt ergibt sich für diese Struktur folgende Aufstellung:

Struktur-Stückliste		Datum: 03.08.08	Seite: 1
Artikelnummer:	U1400		
Artikelbezeichnung:	Uhrwerk komplett		
Änderungsindex:	3		
Änderungsdatum:	03.08.01		
Zeichnung:	427/33		
Baustufe	Artikelnummer	Artikelbezeichnung	Menge
*1	U1410	Sekundengetriebe	1
**2	U1411	Sekundenrad	1
***3	U1555	PVC-Granulat	0,1
**2	U1888	Achse	1
**2	U1777	Lager	2
*1	U1430	Zwischengetriebe	1
**2	U1431	Zwischenrad	1
***3	U1555	PVC-Granulat	0,14
**2	U1888	Achse	1
**2	U1777	Lager	2
*1	U1420	Ankergetriebe	1
**2	U1421	Ankerrad	1
***3	U1555	PVC-Granulat	0,12
**2	U1888	Achse	1
*1	U1440	Minutengetriebe	1
**2	U1450	Stundengetriebe	1
***3	U1453	Stundenrad	1
***3	U1454	Minutenwelle	1
***3	U1451	Stundenzeiger	1
***3	U1452	Minutenzeiger	1
***3	U1999	Abdeckkappe	1
**2	U1441	Minutenrad	1
*1	U1460	Kleinbodengetriebe	1
**2	U1880	Achse	1
**2	U1461	Kleinbodenrad	1

In dieser Stückliste sind alle Baugruppen und deren Einzelteile angeführt. Es ist daraus ersichtlich, welche Artikel in welchen Mengen zunächst vormontiert werden. Die Artikel einer niedrigeren Baustufe gehen in einen Artikel höherer Baustufe ein.

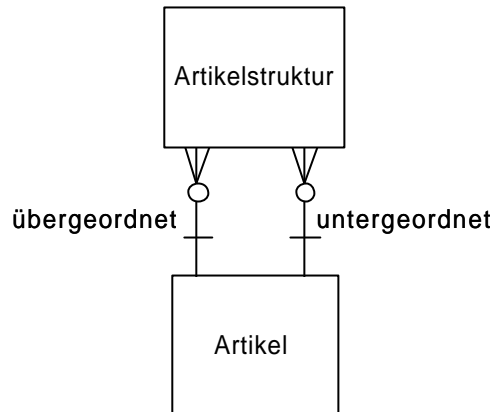
Aus der Stückliste lässt sich zunächst nur der Entitätstyp "Artikel" ableiten. Die Tatsache, dass ein Artikel aus anderen Artikeln besteht, bzw. in mehrere andere Artikel eingehen kann lässt sich als Beziehung darstellen.



Datenmodellierung

Ein Artikel ist übergeordnet 0,1,n Artikeln
Ein Artikel ist untergeordnet 0,1,m Artikeln

Es handelt sich also um eine n:m Beziehung von Artikel auf sich selbst. Diese n:m Beziehung kann ganz normal, durch Einfügen eines Entitätstyps aufgelöst werden:



Der neue Entitätstyp erhält den Namen Artikelstruktur. Es wird genau gleich verfahren, wie bei der Auflösung einer normalen n:m Beziehung. Es werden Attribute gesucht, die Fremdschlüssel übernommen und ein Identifikationsschlüssel definiert.

Als zusätzliches Attribut ist die Menge hinzuzufügen. Ein Datensatz in dieser Tabelle gibt an welcher Artikel in welchen Artikel eingeht, und welche Menge des untergeordneten Artikel eingesetzt werden muss.

Fremdschlüssel: Entsprechend der Regel, dass bei c- oder n-Beziehungen der Identifikationsschlüssel des jeweils anderen Entitätstyps als zusätzliches Attribut aufgenommen wird, wird die Artikelnummer zweimal als Fremdschlüssel übernommen - einmal für die Beziehung "übergeordnet", einmal für die Beziehung "untergeordnet".

Identifikationsschlüssel: Der Identifikationsschlüssel des Entitätstyps Artikelstruktur setzt sich aus beiden Fremdschlüsseln zusammen.

Als Einrückungsliste bzw. als Tabelle ergibt dies folgendes Ergebnis:

ARTIKEL

Artikelnr

Artikelbezeichnung

ARTIKEL-STRUKTUR

Artikelnr_(übergeordnet) --> Artikel

Artikelnr (untergeordnet) --> Artikel

Menge

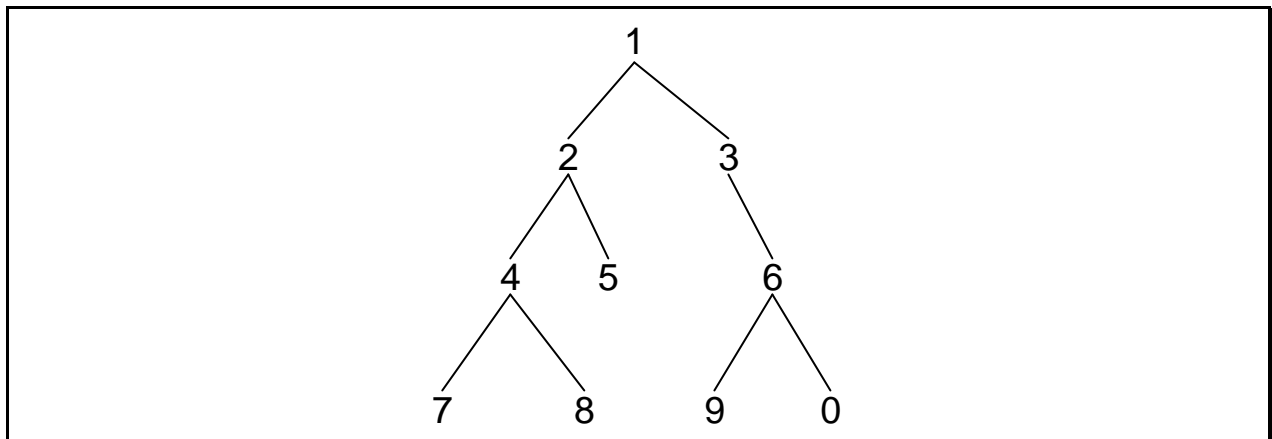
ARTIKEL	<u>Artikelnr</u>	Artikelbezeichnung
	U1400	Uhrwerk komplett
	U1410	Sekundengetriebe
	U1411	Sekundenrad
	U1430	Zwischengetriebe
	U1431	Zwischenrad
	U1555	PVC-Granulat
	U1777	Lager
	U1888	Achse

ARTIKELSTRUKTUR	<u>Artikelnr</u> (übergeord.)	<u>Artikelnr</u> (untergeord.)	Menge
	U1400	U1410	1
	U1400	U1430	1
	U1410	U1411	1
	U1410	U1777	2
	U1410	U1888	1
	U1411	U1555	0,1
	U1430	U1431	1
	U1430	U1777	1
	U1430	U1888	2
	U1431	U1555	0,14

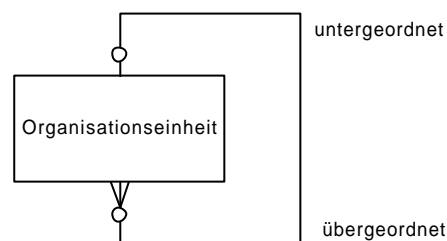
Aus diesem Datenmodell lassen sich alle Informationen der Strukturstückliste des Uhrwerkes ablesen.

Die bisherigen Ausführungen beziehen sich auf die Abbildung von **netzartigen Hierarchien**. Das folgende Beispiel erläutert die Darstellung **baumartiger Hierarchien**:

Es soll die Aufbauorganisation eines Unternehmens dargestellt werden. Die Abteilungen werden durch ihre Abteilungsnummer repräsentiert, die übrigen Attribute einer Abteilung bleiben für dieses Beispiel außer Betracht. Folgende Struktur liegt dem Beispiel zugrunde:



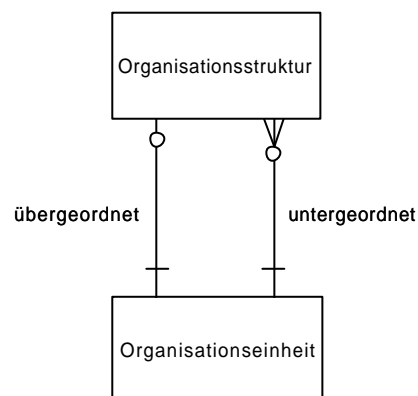
Analog zum Beispiel der Stückliste besteht die Auflösung der Struktur im Datenmodell in einer Beziehung des Entitätstyps "Organisationseinheit" auf sich selbst. Nur ist diese Beziehung bei baumartigen Strukturen eine n:c Beziehung.



Eine Organisationseinheit ist 0,1 Organisationseinheiten untergeordnet

Eine Organisationseinheit ist 0,1,n Organisationseinheiten übergeordnet

Die Auflösung dieser Beziehung erfolgt über den Entitätstyp "Organisationsstruktur":



ORGANISATIONSEINHEIT

Orgnr

Organisationsbezeichnung

ORGANISATIONSSTRUKTUR

Orgnr_(untergeordnet) --> Organisationseinheit

Orgnr (übergeordnet) --> Organisationseinheit

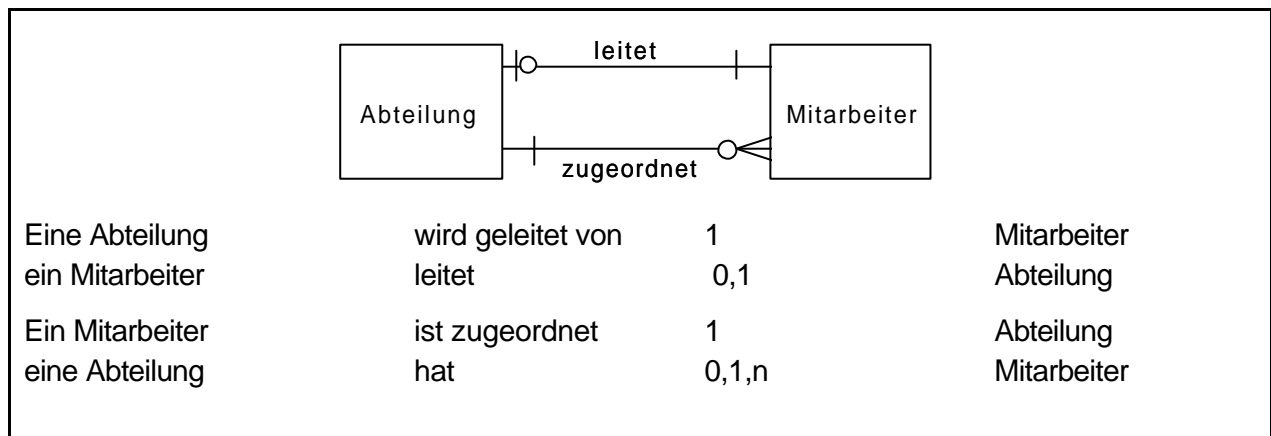
ORGANISATIONSTRUKTUR	Orgnr (übergeord.)	Orgnr (untergeord.)
	1	2
	1	3
	2	4
	2	5

Bei der Abbildung baumartiger Strukturen muss sehr sorgfältig auf die Arten der Beziehungen geachtet werden, da sich der Identifikationsschlüssel in der "- Struktur" Tabelle ändern könnte. Wenn der falsche Fremdschlüssel als Identifikationsschlüssel angenommen wird, führt dies zu schwerwiegenden Interpretationsproblemen. Das Problem kann gelöst werden, indem man mit der Tabellendarstellung und einigen Beispielen die Richtigkeit des Modells überprüft.

2.9.3 Indirekte Rekursion

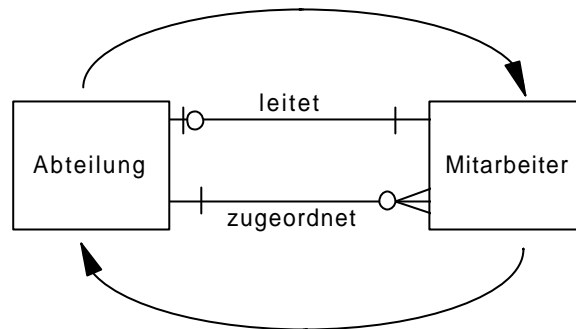
Wenn eine Entität in eine Datenbank aufgenommen werden soll, und der Entitätstyp Fremdschlüssel als Attribut enthält, so muss für dieses Attribut ein gültiger Wert angegeben werden, d.h. ein gültiger Identifikationsschlüssel des Entitätstyps, zu dem eine Beziehung besteht (= referenzielle Integrität).

Wenn allerdings zwischen den Entitätstypen mehrere Beziehungen mit unterschiedlicher Aussage bestehen so kann dies, wie im folgenden Beispiel zu Problemen führen.



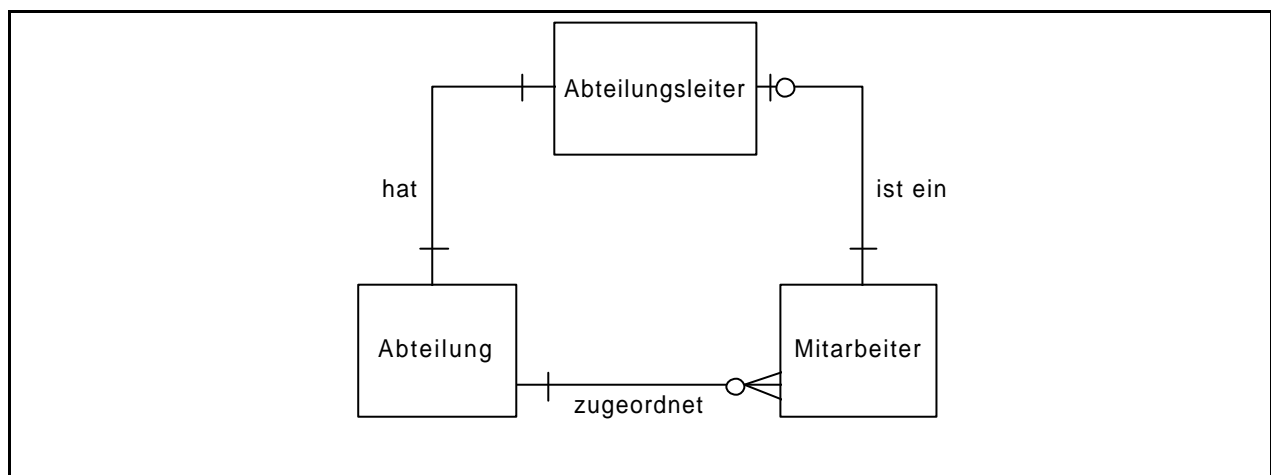
Das Problem bei indirekten Rekursionen besteht darin, dass jede Entität das Vorhandensein einer Entität des anderen Entitätstyps voraussetzt, ein Einfügen eines Datensatzes wird somit unmöglich.

Eine Abteilung kann nicht existieren, ohne dass sie von einem Mitarbeiter geleitet wird



Ein Mitarbeiter kann nicht existieren, ohne dass er einer Abteilung zugeordnet ist.

Das Problem kann durch Einfügen eines Entitätstyps gelöst werden:



ABTEILUNG

Abteilungsnummer

Abteilungsname

...

ABTEILUNGSLEITER

Personalnummer

--> Mitarbeiter

Abteilungsnummer

--> Abteilung

...

MITARBEITER

Personalnummer

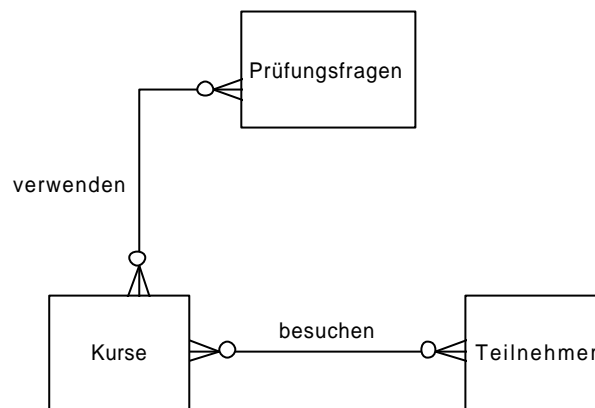
Name

...

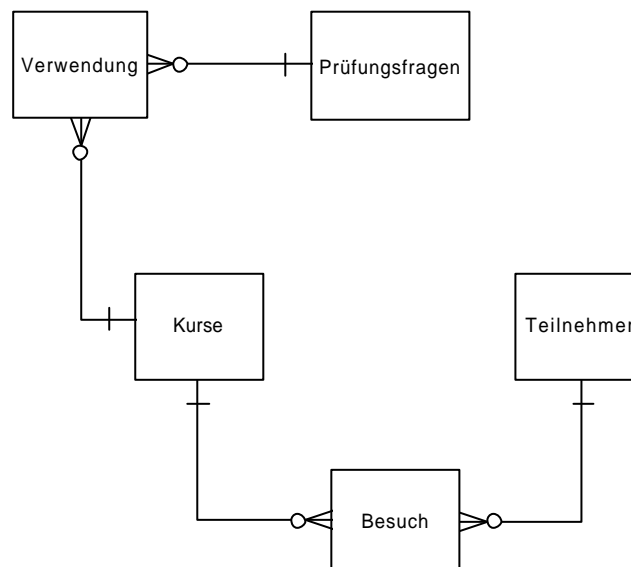
2.9.4 4. und 5. Normalform

Die 4. und 5. Normalform sind komplizierte theoretische Konstruktionen, deren praktische Relevanz umstritten ist¹⁰. Das Problem der 4. und 5. Normalform kann entstehen, wenn zwischen Entitätstypen mehrere wechselseitige (n:m) Beziehungen bestehen. Das folgende Beispiel soll die Problematik illustrieren, ohne auf den theoretischen Hintergrund im Detail einzugehen.

Ausgangspunkt sind drei Entitätstypen: Kurse, Teilnehmer und Prüfungsfragen. Kurse und Teilnehmer sind aus dem Einführungsbeispiel bekannt, die Prüfungsfragen sind ein Katalog aller Prüfungsbeispiele, die in den verschiedenen Kursen verwendet werden. Zwischen Teilnehmer/Kurse und Prüfungsfragen/Kurse besteht jeweils eine n:m Beziehung.



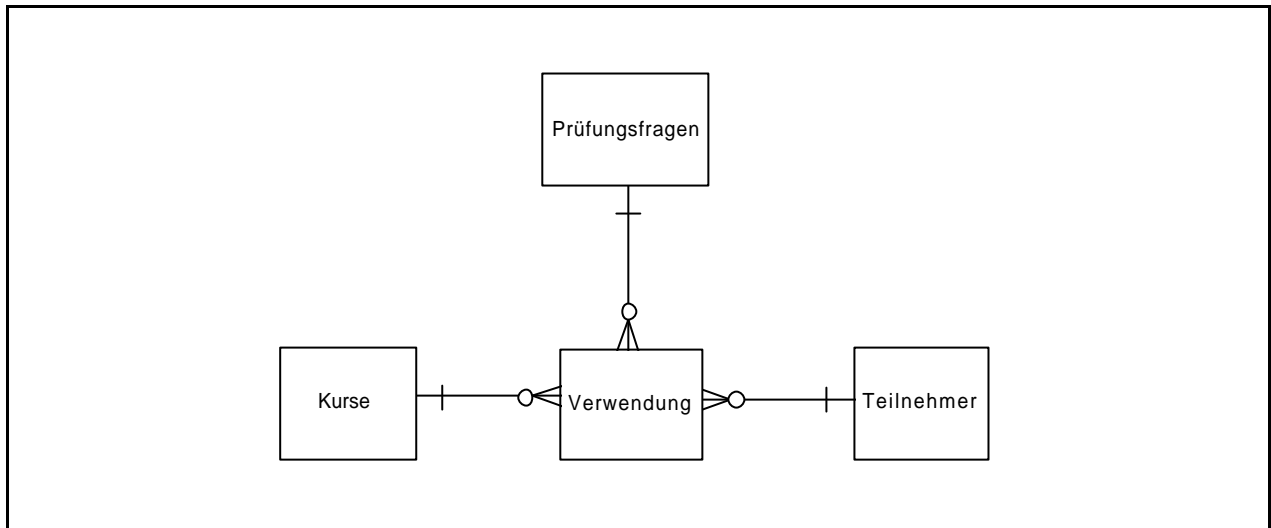
Nun ist es von der Realität, die im Datenmodell dargestellt werden soll abhängig, wie diese beiden Beziehungen "richtig" aufgelöst werden. Zunächst soll davon ausgegangen werden, dass die beiden Beziehungen voneinander unabhängig sind. Die Auflösung erfolgt, indem jede Beziehung für sich behandelt wird, also jeweils ein Entitätstyp eingefügt wird.



¹⁰ Vgl. Vetter, M.: Aufbau betrieblicher Informationssysteme. Mittels konzeptioneller Datenmodellierung, 8. Auflage.- Stuttgart: Teubner, 1998, S. 152ff

Datenmodellierung

Wenn die Realität nun leicht modifiziert wird, reicht die Darstellung in dieser Form nicht mehr aus. Geht man davon aus, dass sich die Verwendung einer Prüfungsfrage nicht auf den Kurs bezieht, sondern auch auf den Teilnehmer (z.B. um sicherzustellen, dass ein Teilnehmer nicht mehrmals dieselben Prüfungsfragen erhält) kann diese Information im obigen Datenmodell nicht dargestellt werden. Die Beziehungen zwischen den Entitätstypen sind nicht mehr voneinander unabhängig. Im vorliegenden Beispiel sagen die Beziehungen aus, welcher Teilnehmer in welchem Kurs welche Prüfungsfragen erhalten hat. Das ER-Diagramm des normalisierten Datenmodells unterscheidet sich wie folgt:



PRÜFUNGSFRAGEN

Fragennummer

Text

maximale Punkte

KURSE

Kursnummer

Titel

...

TEILNEHMER

Personalnummer

Name

...

VERWENDUNG

Fragennummer

--> Prüfungsfragen

Kursnummer

--> Kurse

Personalnummer

--> Teilnehmer

Beurteilung

Freigabe

...

2.9.5 Integritätsbedingungen

Datenintegrität bedeutet die Übereinstimmung des Inhalts einer Datenbank mit der Datenbeschreibung und die innere Widerspruchsfreiheit der Daten.

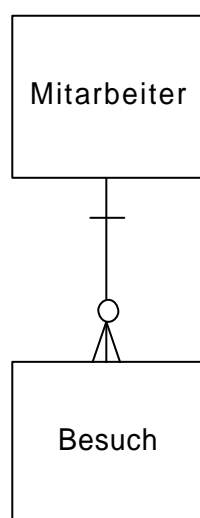
Arten von Integritätsbedingungen sind:

- Eindeutigkeit
- Referenzielle (existentielle) Integrität
- Value Constraints
- Entity Constraints

Datenintegrität - Eindeutigkeit: Es dürfen nicht zweimal dieselben Entitäten existieren. (wird durch Datenbankstruktur bereits ausgedrückt)

MITARBEITER	<u>Personalnr</u>	Mitarbeitername	...
	20	Müller	...
	30	Maier	
	40	Maier	...
	40	Maier	

Datenintegrität - Referenzielle (existentielle) Integrität: Bei Entitätstypen, die einen Fremdschlüssel enthalten, sind beim Fremdschlüssel nur solche Werte zugelassen, die in der referenzierten Entitätsmenge bereits existieren.
(wird durch Datenbankstruktur bereits ausgedrückt)



Datenmodellierung

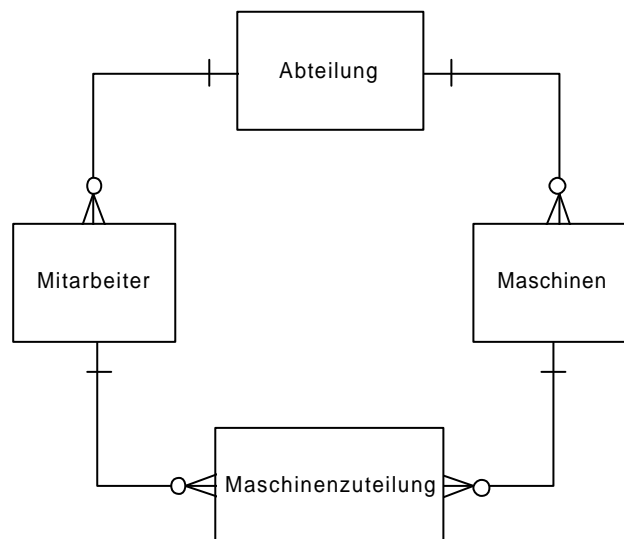
MITARBEITER	<u>Personalnr</u>	Mitarbeitername	...
	20	Müller	...
	30	Maier	...
	40	Hofer	...

BESUCH	<u>Personalnr</u>	<u>Kursnr</u>	Mitarbeiterbeurteilung	...
	20	666	bestens	...
	20	777		...
	30	777		...
	40	666		...

Datenintegrität - Value Constraints: Werte von lokalen Attributen, die bestimmten fachlichen Bedingungen unterliegen. (wird durch die Datenstruktur nicht ausgedrückt)

- 1) Vorgegebener Wertebereich:
 - Alter > 0; <= 150
 - Zivilstand: ledig, verheiratet, geschieden, verwitwet
- 2) Werte in Abhängigkeit von Werten anderer Attribute:
 - Auftragdatum <= Lieferdatum
 - Summe aller Gehälter kann nicht kleiner als das Budget einer Abteilung sein

Datenintegrität - Entity Constraints: Abhängigkeiten zwischen Entitätstypen, die in der Datenstruktur nicht zum Ausdruck kommen (wird durch die Datenstruktur nicht ausgedrückt).



Datenmodellierung

Aus diesem Datenbankstrukturdiagramm (auch aus der Datenstruktur) lässt sich nicht ablesen, dass ein Mitarbeiter nur den Maschinen zugewiesen werden kann, die derselben Abteilung angehören, wie er selbst.

Dies ist in Form einer Integritätsbedingung zu formulieren.

Darstellung von Integritätsbedingungen: Integritätsbedingungen sind neben dem Datenbankstrukturdiagramm und der Datenstruktur ein weiteres wesentliches Ergebnis des Datenentwurfs. Für die Darstellung der Integritätsbedingungen gibt es keine formalen Einschränkungen. Normalerweise werden sie am Ende der Datenstruktur aufgelistet.

1. Value Constraints (beziehen sich auf lokale Attribute):
 - Auftragsdatum \leq Lieferdatum
 - Lieferdatum: $> 1.1.2003; \leq 1.1.2050$
2. Entity Constraints (beziehen sich auf Entitätstypen):
 - Maschinenzuteilung: Ist nur möglich, wenn Abteilungsnummer des Mitarbeiters = Abteilungsnummer der Maschine

2.9.6 Summenfelder

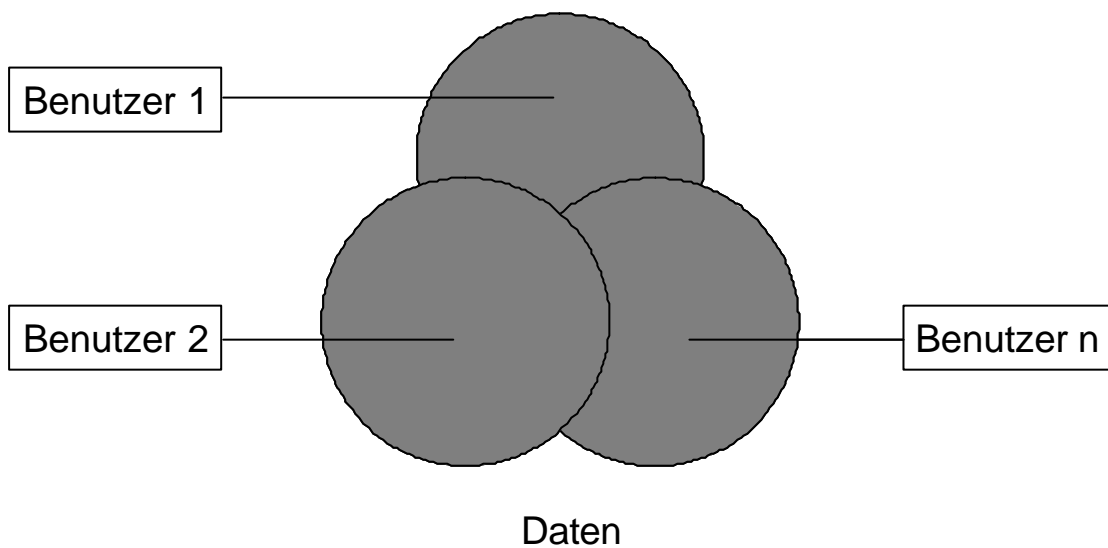
Manche Datentypen können aus anderen Datentypen berechnet werden. Insofern bedeutet deren Aufnahme in das Datenmodell Redundanz.

Für solche Attribute ist jedenfalls die Integritätsbedingung zu formulieren.

3 Zusammenfassung

3.1 Bedeutung von Datenmodellen

Verwendung von Daten: Ablegen, Lesen, Auswerten



- Vermeiden von Redundanz
 - Einmalige Datenerfassung
 - Aktualität der Daten
 - Rasche und individuelle Verfügbarkeit von Informationen
 - Entwicklungsfähigkeit der Datenstruktur
 - Datenschutz
- > *Voraussetzung ist korrekte Datenstruktur*

3.2 Vorgehensschritte der Datenmodellierung

1. Komponentenliste
 elementare Datentypen
 zusammengesetzte Datentypen
2. Datentyphierarchie
 Entitätstypen, Entitäten
 Attribute
3. Identifikationsschlüsselvergabe
 Eindeutigkeit
 gesamte Lebensdauer
 jede Entität
4. Beziehungen feststellen
 Beziehungsarten (1:1, 1:n, 1:c)
 wechselseitige Beziehungen
 Beziehung über Identifikationsschlüssel
5. Beziehungen normalisieren
 n:n, n:c und c:c - Beziehungen
 Einfügen neuer Entitätstypen
6. Datenstruktur vervollständigen

3.3 Darstellungstechniken

Datenbankstrukturdiagramm

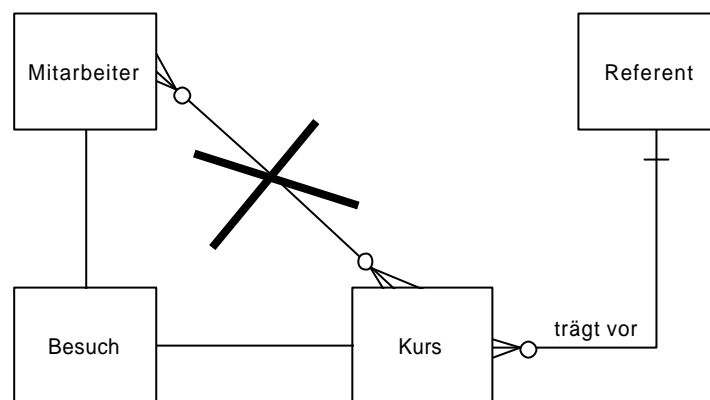
Attributliste

Tabellen

4 Beispiele

4.1 Beispiel Begriffe:

Bitte lesen Sie zuerst alle Fragen dieser Aufgabe, und beantworten Sie die Fragen anschließend



MITARBEITER

Personalnummer
Mitarbeitername
Funktion
Gehalt

KURS

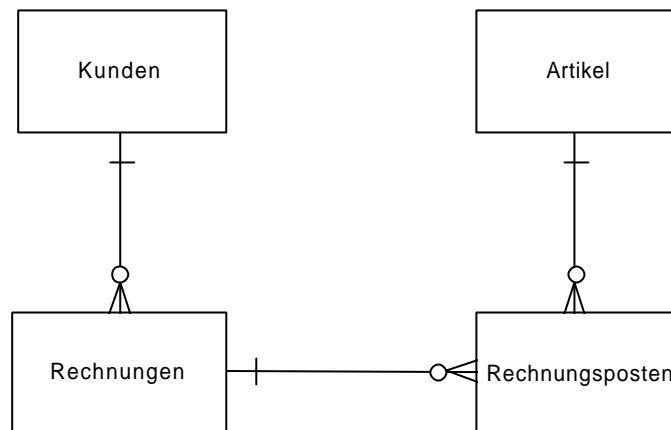
Kursnummer
Kursbezeichnung
Beginnzeit
Kursende
Kursort

REFERENT

Referentennummer
Referentenname
Strasse
Ort
Postleitzahl
Land
Fachbereich

1. Beschreiben Sie bitte, welche Aufgabenstellung mit einer Datenbank entsprechend dem oben dargestellten Modell erfüllt werden kann.
2. Wieso ist die Verbindung zwischen Mitarbeiter und Kurs durchgestrichen?
3. Tragen Sie im ER-Diagramm (Entity Relationship Diagramm) die korrekte Beziehungsart zwischen Mitarbeiter und Besuch sowie Kurs und Besuch ein.
4. Zeichnen Sie Tabellen für Mitarbeiter, Besuch und Kurs und demonstrieren Sie durch ein Beispiel, welche Funktion die Tabelle Besuch hat.

4.2 Beispiel Kunden - Artikel



Beantworten Sie bitte folgende Fragen:

- Wie wird die oben dargestellte Grafik mit einem Fachausdruck bezeichnet?
- Welche Darstellungsformen kennen Sie noch beim Datenbankentwurf?
- Interpretieren Sie den oben dargestellten Sachverhalt. Gehen Sie dabei genau auf die dargestellten Beziehungen zwischen den verschiedenen Entitätstypen (Kunden, Rechnungen, etc.) ein.
- Erklären Sie den Begriff Normalisierung?
- Warum kann in der oben dargestellten Datenbank die Verbindung nicht direkt von Rechnungen zu Artikel bestehen.
- Erläutern Sie die Funktion von Primärschlüsseln.
- Definieren Sie Attribute für die Entitätstypen Rechnungen, Rechnungsposten, Artikel (pro Entitätstyp ca. 3 Attribute). Stellen Sie die drei Entitätstypen und deren Attribute in Form von Tabellen dar.
- Über welche Attribute können die drei im Pt. g) dargestellten Entitätstypen in der Datenbank miteinander verbunden werden? (Zeichnen Sie diese Verbindungen eventuell in die Tabellen ein.)

4.3 Beispiel Stundenzettel

Entwerfen Sie ein normalisiertes, relationales Datenmodell für folgende Aufgabenstellung und stellen Sie es als ER-Diagramm (Datenbankstrukturdiagramm), als Attributliste und in Tabellenform dar. Tragen Sie in die Tabellen Beispieldatensätze ein, die den Zusammenhang der Tabellen erläutern. Max Bau, ein Unternehmen, das sich auf Holzkonstruktionen spezialisiert hat, will die Stundenzettel der Mitarbeiter mit einem PC erfassen und nach Baustellen sowie nach Mitarbeitern auswerten. Auf den Stundenzetteln steht die Personalnummer und der Name des Mitarbeiters, die Baustelle, das Datum und die Anzahl der dort gearbeiteten Stunden. Für jede Baustelle wird pro Mitarbeiter ein eigener Zettel abgegeben. Für die Mitarbeiterauswertung

Datenmodellierung

wird neben dem Namen noch die Sozialversicherungsnummer benötigt. Für die Baustellenauswertung ist die Adresse der Baustelle und der Name des Auftraggebers erforderlich.

Die Auswertungen sollen in etwa wie folgt aussehen:

Baustellenabrechnung	
Josef A.	
Marktstraße 3	
9999 Baudorf	
Dachbodenausbau	
Datum Arbeiter	Stunden
....

Mitarbeiterabrechnung	
Stefan M.	
SV: 9007 311293	
Datum Baustelle	Stunden
....

4.4 Beispiel Rechnung

Ihre Firma versendet Rechnungen wie unten abgebildet. Auf der Rechnung befinden sich Informationen über den Kunden, Artikel und Auftragsposten. Entwerfen Sie eine normalisierte Datenbank, die als Grundlage für eine solche Rechnung dienen kann. Stellen Sie Ihren Entwurf mit Hilfe eines ER-Diagramms und in Form von Tabellen dar. Zeichnen Sie in den Tabellen grafisch die Verbindungen ein, mit denen bestimmte Artikel einem Kunden zugewiesen werden.

Rabe Ges.mBH Innrain 10 6020 Innsbruck			Innsbruck, 21.1.2003
Rechnung			
Artikelnummer	Bezeichnung	Preis	
00213	DVDs	450,--	
23004	Festplatten	900,--	
<u>12345</u>	<u>Einbau pauschal</u>	<u>800,--</u>	
Summe		2150,--	
<u>+ 20% MWSt.</u>		<u>430,--</u>	
Rechnungssumme		2580,--	

4.5 Beispiel Liegenschaftsverwaltung

Sie sind Mitarbeiter einer Hausverwaltungsgesellschaft. Die Gesellschaft verwaltet Häuser (Liegenschaften) verschiedener Eigentümer. Eine Liegenschaft gehört nur jeweils einem Eigentümer. In der Regel enthält eine Liegenschaft mehrere Mietobjekte, über die mit einem Mieter ein Mietvertrag abgeschlossen wird. Es kann vorkommen, dass bei Mieterwechsel gleichzeitig mehrere Mietverträge für ein Mietobjekt existieren.

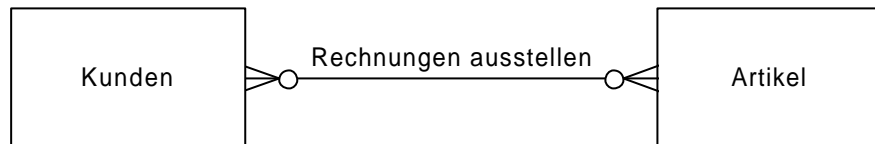
Aufgabe 1: Erstellen Sie ein vollständig normalisiertes Datenmodell. Dokumentieren Sie dieses Modell mit einem ER-Diagramm und einer Attributliste.

Aufgabe 2: Es kann vorkommen, dass ein Eigentümer gleichzeitig bei einer anderen Liegenschaft Mieter ist. Wie können Sie vermeiden, dass die Daten in diesem Fall doppelt gespeichert sind (Name und Adresse der Person einmal als Eigentümer und als Mieter). Ergänzen Sie das ER-Diagramm aus Aufgabe 1 um diese Lösung.

4.6 Beispiel PC-Komponenten

In einem Großhandelsunternehmen für PC-Komponenten liegt Ihnen aus dem Bereich der Fakturierung folgendes, unnormalisiertes ER-Diagramm vor:

Datenmodellierung



Führen Sie die Normalisierung bis zur 3. Normalform durch und ergänzen Sie die Entitätstypen mit sinnvollen Attributen. Stellen Sie das Ergebnis in einem ER-Diagramm und in Tabellen dar. Füllen Sie die Tabellen mit einigen Daten, um die Funktion Ihrer Lösung zu demonstrieren.

4.7 Beispiel Lager

Sie sind EDV Leiter des Handelsunternehmens Trash-Cash Inc. und stehen vor folgendem Problem:

- Die Artikel Ihres Unternehmens werden in verschiedenen, räumlich getrennten Lagern verwaltet.
- Jedes Lager hat eine eigene Adresse, Telefonnummer, usw. und einen eigenen Lagerverwalter.
- Es kann sein, dass ein Lagerverwalter für mehrere Lager verantwortlich ist.
- Die Artikel können in allen Lagern vorrätig sein (auch zur selben Zeit) oder nur in bestimmten Lagerorten.

Aufgabe:

- Erstellen Sie eine Datenbank, in der die oben angeführten Informationen gespeichert werden können. Der Datenentwurf besteht aus einem Datenbankstrukturdiagramm, Attributlisten und Tabellen.
- Wie werden die verschiedenen Entitätstypen miteinander verknüpft? Zeichnen Sie diese Verknüpfungen in den Tabellen ein.
- Erklären Sie den Begriff Normalisierung?
- Erläutern Sie die Funktion von Primärschlüsseln.

4.8 Beispiel Fahrradkurier

Sie haben eine Marktlücke entdeckt! In Ihrer Umgebung gibt es noch kein Fahrrad-Kurierservice. Die Idee besteht darin, dass Sie am Telefon sitzen und auf den Anruf eines Kunden warten, der ein Paket transportiert haben möchte. Sie geben diese Information dann an einen Ihrer Fahrer weiter (Es sind immer einige Fahrer in Bereitschaft.), der den Transport dann durchführt. Nach Erledigung informiert Sie der Fahrer über das Gewicht des Paketes und die gefahrene Strecke. Für die Bereitschaftszeit des Fahrers erhält er einen Stundenlohn, für die Fahrten ein Honorar.

Der Kunde zahlt für den Transport folgende Gebühr: $\text{kg} \cdot \text{km} \cdot 10,-$ (kg = Gewicht des Paketes, km = gefahrene Kilometer). Der Fahrer erhält folgendes Honorar: $\text{kg} \cdot \text{km} \cdot 0,4,-$ und außerdem einen Stundenlohn von 20,-/h.

Datenmodellierung

Die Daten eines Transportes und die Bereitschaftszeiten der Fahrer wollen Sie in einer relationalen Datenbank verwalten, damit Sie später mit einer einfachen SQL-Abfrage die angefallenen Gebühren, Honorare und Stundenlöhne berechnen können.

Entwerfen Sie das Datenmodell und dokumentieren Sie es mit einem Datenbankstrukturdiagramm (ER-Diagramm) und Attributliste. Stellen Sie das Modell in Tabellenform dar und zeigen Sie durch Eintragen von Musterdaten die Funktionsfähigkeit.

4.9 Beispiel Sortiment Handelshaus

Das Sortiment eines Handelsunternehmens ist entsprechend der unten dargestellten Struktur in Artikelgruppen gegliedert. Die Artikel sind den Artikelgruppen (in BLOCKSCHRIFT) zugeordnet.

SPORTARTIKEL

TENNIS

SCHLAEGER

Kopf GT

Jäger Carbon

BAELLE

Sphäroid Blau

Softbong Permanent

FRUEHJAHRSAKTION

FREIZEIT

YUPPY-SET

Jäger Carbon

Softbong Permanent

Tennishirt Pink Panther

Leibchen Alligator de Luxe

BAELLE

Sphäroid Blau

Softbong Permanent

BERUF

Aufgabe: Entwerfen Sie das Datenmodell (normalisiertes Datenbankstrukturdiagramm, Datenstruktur, Integritätsbedingungen) für das Handelshaus.

4.10 Beispiel Vinothek

Sie sind stolzer Besitzer einer Kartei von Interessenten. Die Kartei enthält 1500 Karteiblätter. Jedes Karteiblatt enthält Name, Adresse, Beruf und Einkommen einer Person. Ihre Freundin (Ihr Freund) besitzt eine Kartei mit erlesenen Weinen aus der ganzen Welt. Die Kartei enthält 500 Karteiblätter. Jedes Karteiblatt enthält Bezeichnung, Alter, Anbaugebiet und Preis pro Weinflasche. Noch ein Freund (eine Freundin) besitzt eine Vinothek. Er/Sie bietet Ihnen eine Reise in die schönsten Weinanbaugebiete der Welt im Tausch gegen ein datenbankgestütztes Bestellsystem für Weine.

Datenmodellierung

Das Bestellsystem erfüllt folgende Anforderungen: Ausgangsbasis ist die Kartei der Interessenten (Kunden) und die Kartei der Weine. Jeder Kunde kann beliebig viele Weine bestellen. Selbstverständlich kann auch der gleiche Wein von mehreren Kunden geordert werden. Für jedes Anbaugebiet sind Daten über Lage, Klima und Geschichte zu speichern um den Interessenten auch Hintergrundinformation anbieten zu können. (Ein Wein kann nur aus einem Anbaugebiet stammen).

Erstellen Sie ER-Diagramm und Tabellen für die Lösung der Aufgabe.

4.11 Beispiel Inselverwaltung

Die Insel Kuredu ist mit ihren 1200 Metern Länge und 300 Metern Breite eine der größten Inseln auf den Malediven. Sie ist ein Tauch- und Erholungsparadies. 250 kleine Bungalows stehen für die Gäste - die meist zwei Wochen bleiben - zur Verfügung, wobei es zwei Kategorien gibt: Die normale Kategorie mit Dusche/ WC/ Ventilator und die Superior - Kategorie mit zusätzlicher Klimaanlage und Kühlschrank. Auf der Insel gibt es verschiedene Einrichtungen: Die Tauchschule, zwei Restaurants, vier Bars (Babuna-Bar, Akiri-Bar und zwei Strandbars), einen Shop, einen Juwelier, usw. Etwa 50 einheimische Mitarbeiter kümmern sich um das Wohl der Gäste. Damit die Gäste nicht immer auf Geld und Wertsachen aufpassen müssen, gibt es einen Safe. Beahlt wird auf der ganzen Insel nur mit Angabe der Bungalow-Nummer und der Unterschrift auf einem Rechnungsbeleg wie beispielsweise folgendem:

Einrichtung: Akiri - Bar		
Mitarbeiter:	Amir (Nr. 24)	
Anzahl	Bezeichnung	Preis (MRF)
2	Cocktail 6 on the beach	120
1	Toast	20
Summe		<u>140</u>
15.8.1996	112	<i>Mayr</i>
Datum	Bungalow - Nummer	Unterschrift

Die Rechnungsbelege werden von den Kellnern bzw. sonstigen Mitarbeitern händisch ausgestellt und täglich gesammelt. Die Rechnungssumme (in maledivischen Rufias – MRF) nach Einrichtungen (z.B. Akiri-Bar) wird auf die Gesamtrechnung für einen Bungalow gesetzt. So können die Gäste jederzeit den Gesamtsaldo ihrer Ausgaben in der Rezeption einsehen.

Erstellen Sie dazu ein relationales Datenmodell in der dritten Normalform, sowie die Attributliste zur elektronischen Automatisierung der Rechnungsabwicklung für die Inselverwaltung. Be-

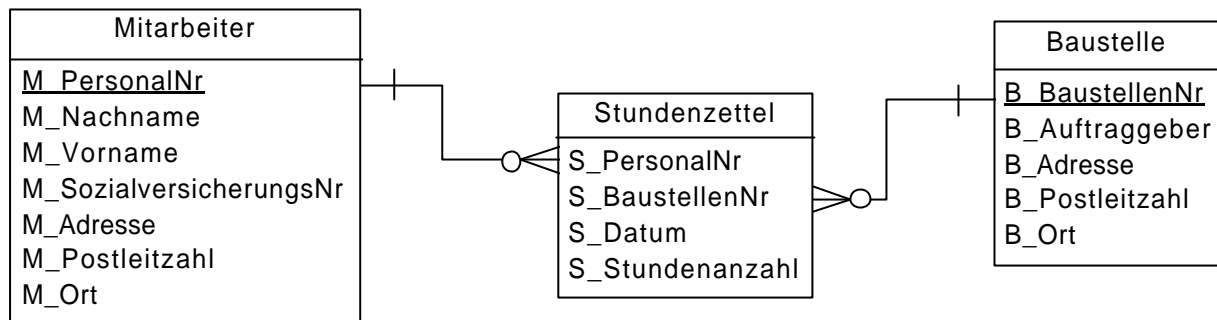
Datenmodellierung

denken Sie, dass damit auch der Umsatz der einzelnen Mitarbeiter der Insel erfasst werden soll und dass während des Jahres unterschiedliche Gäste in den Bungalows wohnen.

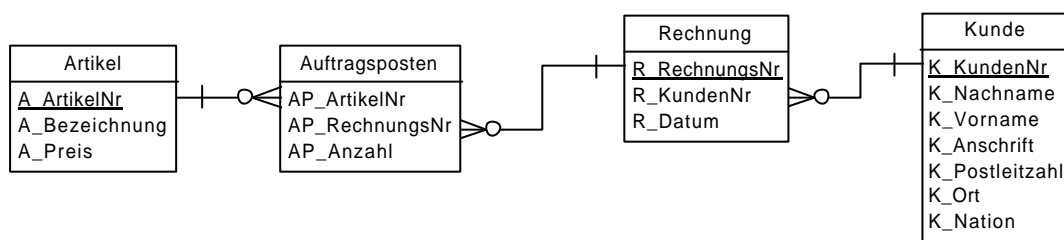
Zeichnen Sie die Tabellen und tragen Sie entsprechende Beispieldatensätze ein.

5 Lösungsvorschläge

5.1 Lösungsvorschlag Stundenzettel

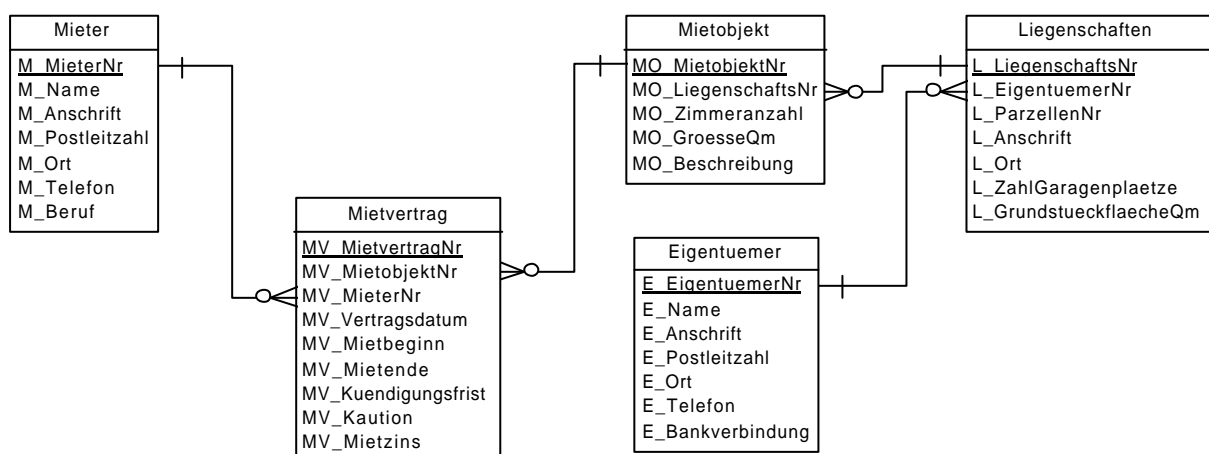


5.2 Lösungsvorschlag Rechnung

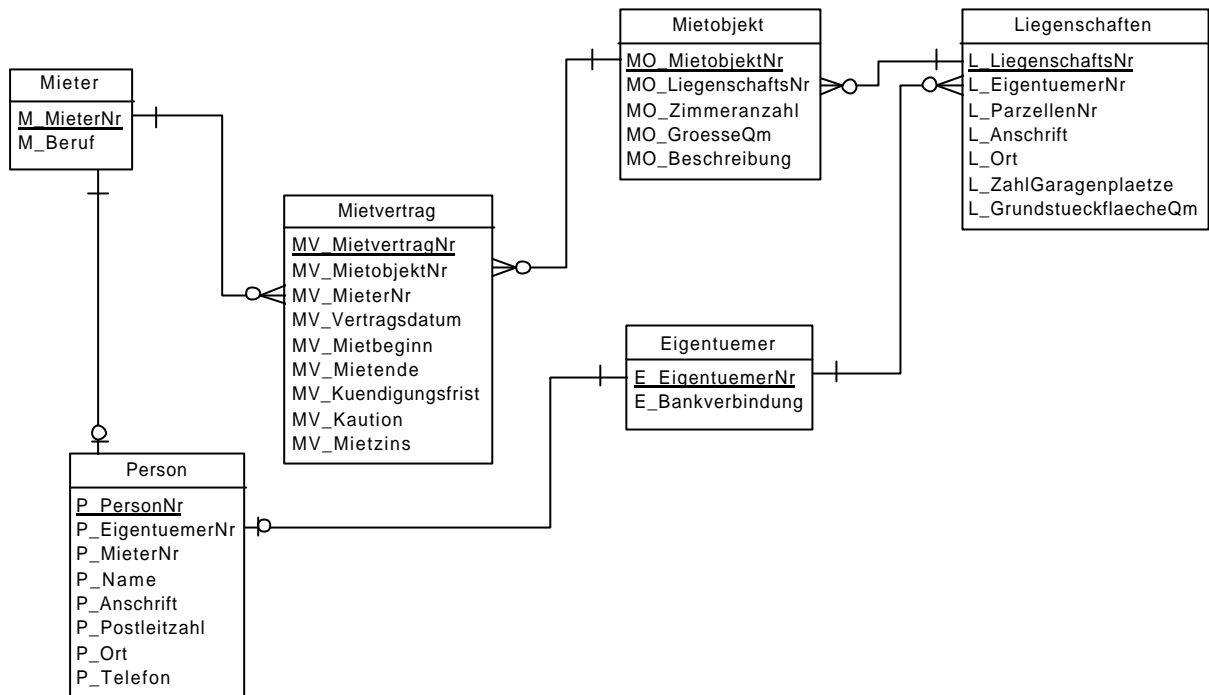


5.3 Lösungsvorschlag Liegenschaftsverwaltung

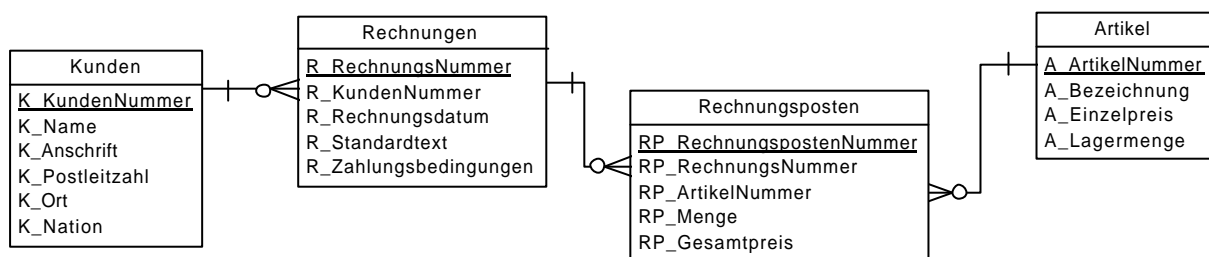
Aufgabe 1:



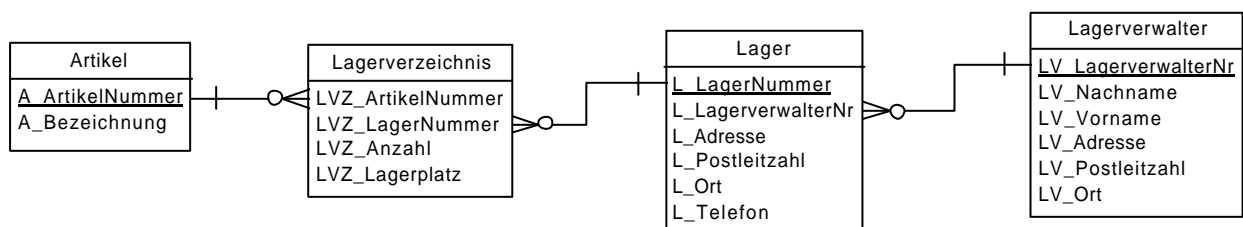
Aufgabe 2:



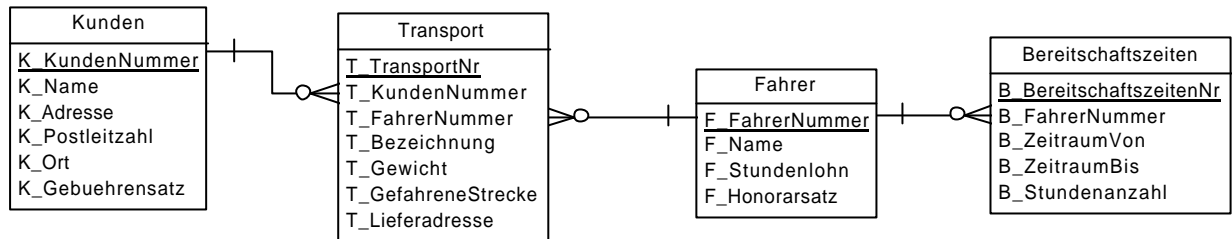
5.4 Lösungsvorschlag PC-Komponenten



5.5 Lösungsvorschlag Lager



5.6 Lösungsvorschlag Fahrradkurier



5.7 Lösungsvorschlag Sortiment Handelshaus:

1. Normalisierte Datenstruktur

ARTIKEL

ArtikelNr

Bezeichnung

ARTIKELGRUPPE

GruppenNr

Bezeichnung

STRUKTURELEMENT

GruppenNr (übergeordnet)

--> Artikelgruppe

GruppenNr (untergeordnet)

--> Artikelgruppe

ARTIKELZUORDNUNG

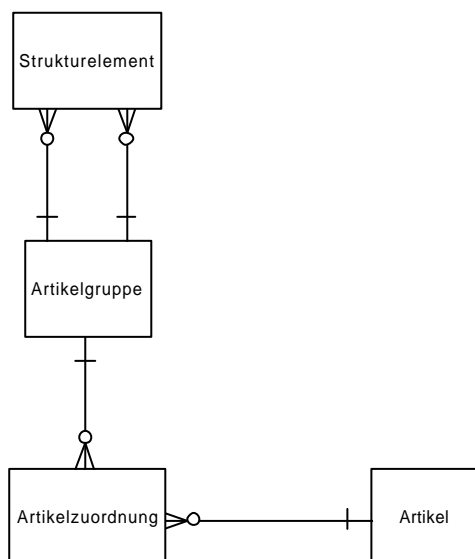
ArtikelNr

--> Artikel

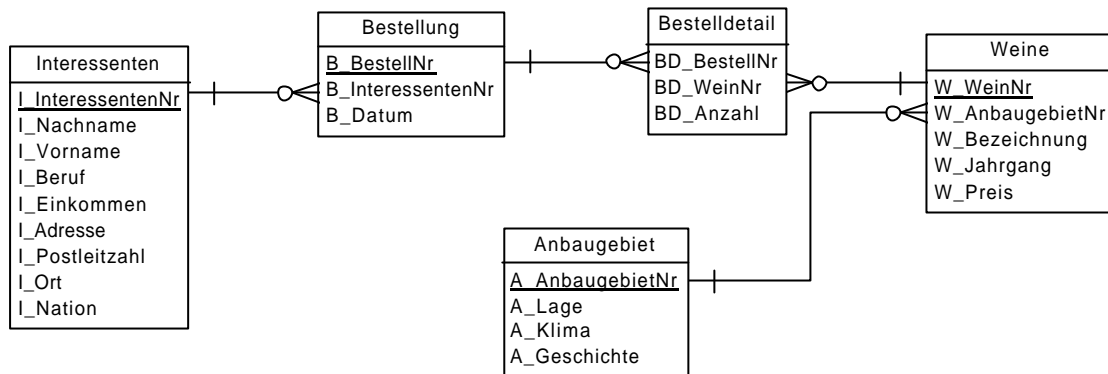
GruppenNr

--> Artikelgruppe

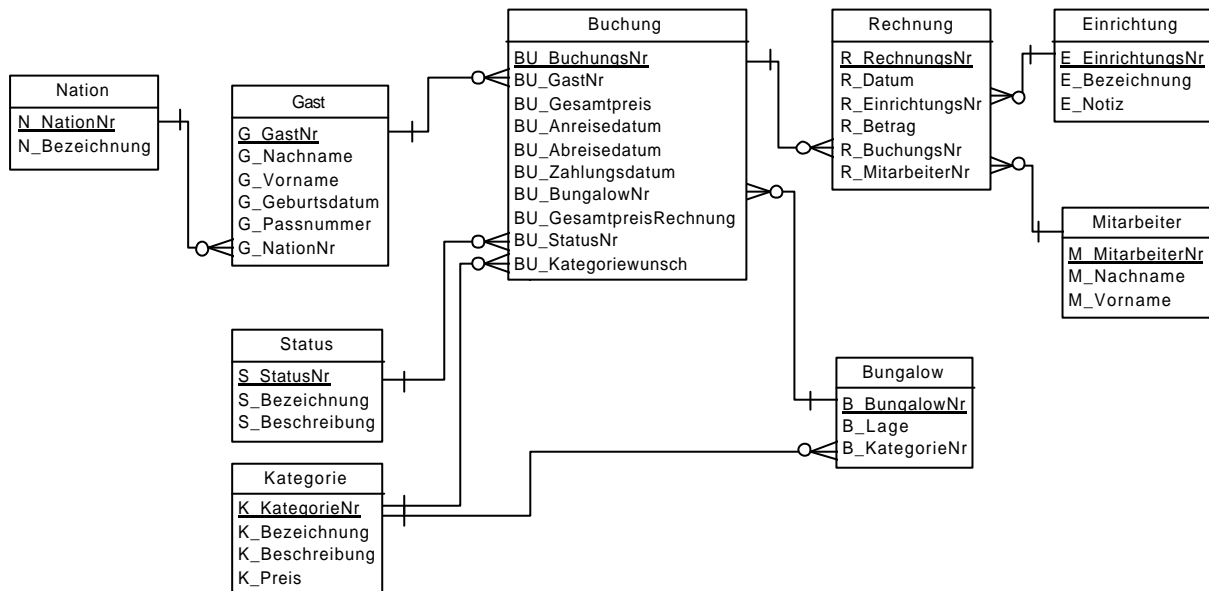
2. Normalisiertes Datenbankstrukturdiagramm:



5.8 Lösungsvorschlag Vinothek



5.9 Lösungsvorschlag Inselverwaltung



6 Literatur

- Vetter, M.: Aufbau betrieblicher Informationssysteme. Mittels konzeptioneller Datenmodellierung, 8. Auflage.- Stuttgart: Teubner, 1998
- Spielberger, J.: Datenmodellierung und relationale Datenbanktechnik. - Wollerau/Schweiz: Skript-Verl. Kühnel, 2001
- Steiner, R.: Grundkurs Relationale Datenbanken, 5. Auflage. - Wiesbaden: Vieweg, 2003
- Matthiessen, G.; Unterstein, M.: Relationale Datenbanken und SQL. - München et al.: Addison-Wesley, 2000