

## Inhaltsverzeichnis: Kapitel 2

- ◆ **2.1 Entity-Relationship-Modell**
  - 2.1.1-3 Entitäten, Beziehungen, Diagramme
- ◆ **2.2 Objektorientiertes Modell**
  - 2.2.1-2 Objekte, Beziehungstypen,
  - 2.2.3-4 Statisches u. Dynamisches Modell, Diagramm
- ◆ **2.3 Relationales Modell**
  - 2.3.1-3 Relationen, Normalisierung, Diagramme
- ◆ **2.4 Datenbankentwurf**
  - 2.4.1-3 Semantische Datenmodellierung, Datenbankentwurf

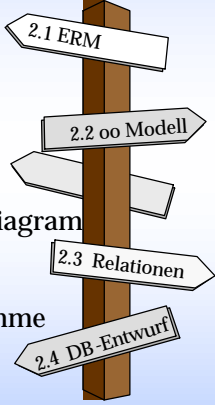
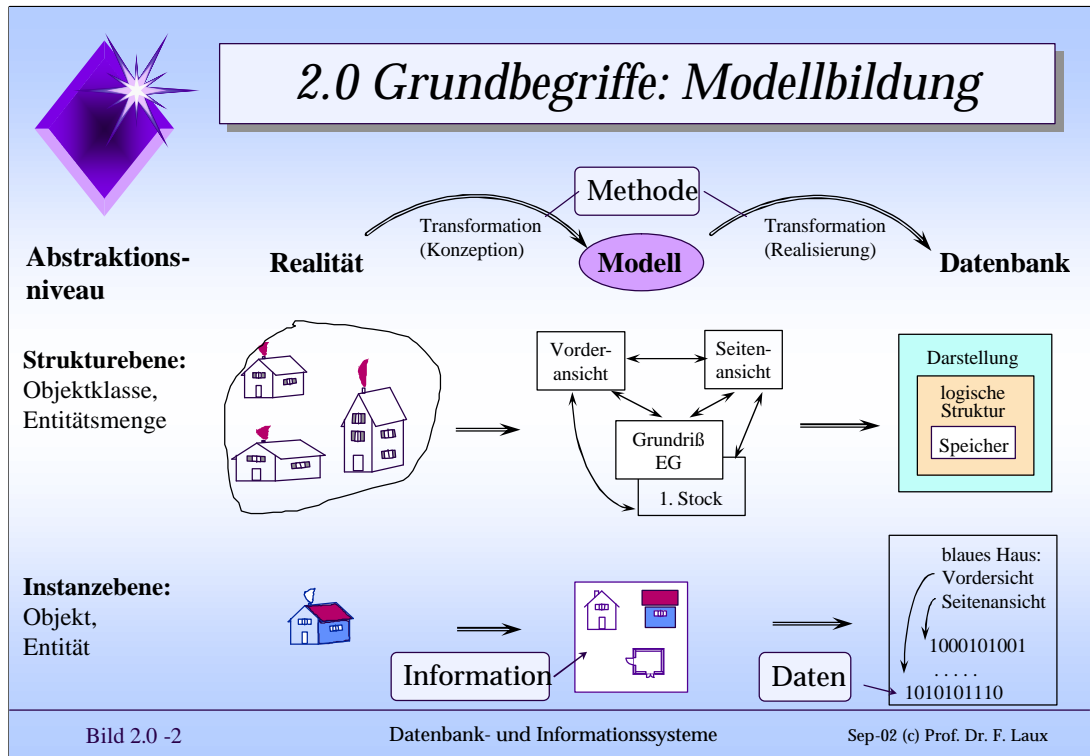


Bild 2.0 -1      Datenbank- und Informationssysteme      Sep-02 (c) Prof. Dr. F. Laux

Kapitel 2 befasst sich mit den Datenmodellen, die zur formalen Beschreibung und Strukturierung eines Sachverhalts (Modellierung) dienen. Wir lernen verschiedene Modelle zur konzeptionellen (fachlichen) Beschreibung von betrieblichen Aufgaben kennen. Entity-Relationship und relationales Modell sind datenorientiert, d.h. sie betrachten vor allem die Daten. Die Betonung beim objektorientierten Modell liegt auf den sogenannten Objekten (zusammengehörigen Daten *und* die darauf erlaubten Operation). Dadurch wird eine 'gesamtheitliche' Beschreibung möglich.

In Kapitel 2.4 präsentieren wir ein einfaches Vorgehensmodell für den Datenbankentwurf.

Zunächst jedoch werden einige Grundbegriffe der Modellbildung wiederholt.



**Objekt: (Entität)** Im allgemeinen ein *Exemplar der realen oder irrealen Welt*, z.B. ein Gegenstand, eine Person, eine Methode, ein Auftrag, mit dem etwas geschieht, geschehen kann/soll oder auf den sich unsere Wahrnehmung richtet. In der Informatik werden solche Exemplare durch ihre Eigenschaften (*Attribute*) und die darauf anwendbaren Manipulationsmöglichkeiten (*Methoden*) beschrieben. Die Attribute werden in Form von Datenstrukturen realisiert. Die aktuellen Werte dieser Struktur bezeichnet man als Status des Objektes. Die Methoden verändern den Status in Bezug auf die Realität in sinnvoller Weise. Eine Statusänderung wird durch eine *Nachricht* (message) an das O. ausgelöst, indem die zuständige Methode die Werte der Datenstruktur verändert. Die Methoden verbergen (kapseln) somit die Implementierung der Datenstrukturen wie bei einem abstrakten Datentyp.

**Beispiele:** Weißes Haus in Washington, Albert Einstein, Kundenauftr. Nr. 12345

**Klasse: (Typ eines Objekts)** Eine Klasse definiert eine *Menge von gleichartigen Objekten*. Objekte sind somit die Instanzen (Materialisierungen) einer Klasse. Alle Objekte einer Klasse besitzen den gleichen *Typ*, d.h. die gleiche Schnittstelle (Methodensignatur) zu ihrer Manipulation und die gleiche Datenstruktur.

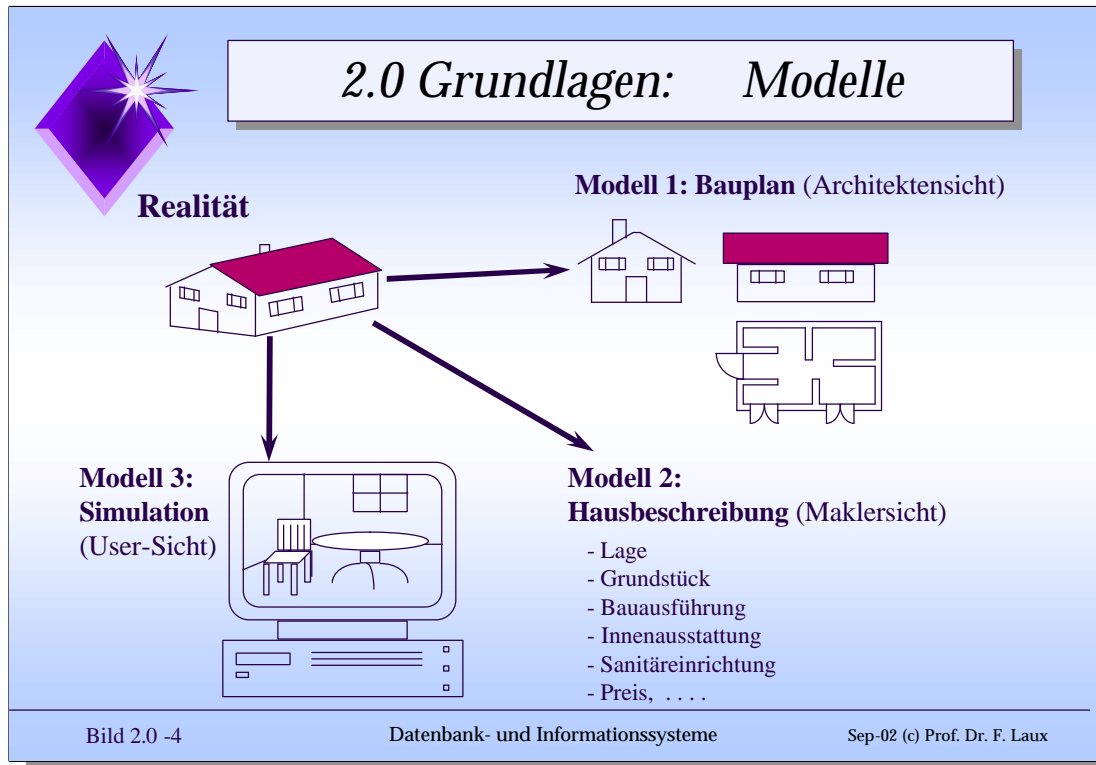
**Beispiele:** Häuser mit Giebeldach, Menschen, Kundenaufträge

NB: Eine Klasse kann ihrerseits wieder als Objekt aufgefaßt werden (Metaebene)

**Information: im informationstechnischen Sinn** verstehen wir als ein *Objekt* und seine Bedeutung (*Semantik*), d.h. Daten und ihre Interpretation.

**Beispiele:** "40000 DM" nicht der String, sondern ein Betrag in DM (Währung)

- Daten:** Eine Darstellung *maschinell verarbeitbarer Information* nennt man in der Informatik Daten. Die für die Informationsverarbeitung geeigneten D. bestehen aus Zeichenfolgen, die digital codiert sind.  
**Beispiele** für digitale Zeichen sind Buchstaben und Ziffern (z.B. ASCII). *Formatierte Daten* ergeben sich, wenn für Mengen gleichartiger D. Typen definiert werden. Z.B. Fließkommazahlen (Typ float) für numerische Werte oder den Typ PERSON mit den Attributen Personalnr, Name, Adresse, die wiederum bestimmte (einfachere) Datentypen darstellen.
- Semantik: (von Daten)** Lehre von der Bedeutung sprachlicher *Zeichen* und Zeichenfolgen (Daten). Die S. legt die *Bedeutung* (Interpretation) von Daten fest.  
**Beispiele:** ASCII-Code, BCD-Code, 5 Äpfel, 5 FF.
- Struktur:** *Objekte*, Klassen und/oder Attribute mit ihren *statischen Beziehungen*.  
**Beispiele:** Kunde - Aufträge, Haus - Mieter, Person - Adresse
- Informationssystem: (technisch)** Eine *Informationssammlung*, *Beziehungen* und *Regeln* zur *Bearbeitung* (Algorithmen, Programme) durch einen Computer. Sein Zweck ist, durch ein Informationsangebot eine -nachfrage zu decken.  
**Beispiel:** Absatzprognosesystem
- Modell:** Eine vereinfachende und *abstrahierende Darstellung* eines Realitätsausschnitts anhand dessen die wichtigsten Aspekte verstanden und untersucht werden können. In der Informatik besteht ein M. aus *Konzepten*, *Regeln* und einen *Formalismus zur Darstellung* (oft graphisch) der Realität.  
**Beispiele:** Atommodell von N. Bohr, Wirtschaftsmodell des Club of Rome, ANSI/SPARC-Modell
- Modellierung:** Tätigkeit im Rahmen der Software-Entwicklung bei der ein Modell des zu entwickelnden Systems erstellt wird.
- Datenmodell:** Ein Modell zur Darstellung der Realität in Form einer *Datenstruktur* und seiner *Manipulationsmöglichkeiten*.  
**Beispiele:** Relationales Modell, objektorientiertes Modell
- Transformation: (im Sinne d. Informatik)** Die *Umsetzung eines Objekts* (z.B. System, Struktur, Element) in eine andere Darstellung. Eine T., welche die Strukturinformation erhält, heißt Isomorphismus.  
**Beispiel:** Umsetzung des konzeptionellen Schemas in das interne
- Methode:** Detaillierte und systematische *Handlungsvorschrift* (Vorgangsmodell), wie im Rahmen eines bestimmten *Modells* ein vorgegebenes Ziel erreicht werden kann.  
**Beispiele:** The Rational Process (RP) nach Rumbaugh, Booch, Jacobson; Object Oriented Software Engineering (OOSE) nach I. Jacobson.




Die Realität oder eine Zielsetzung als Information formal so darzustellen, dass sie durch einen Computer ausgewertet werden kann, ist Aufgabe einer Methode. Die Darstellung erfolgt im Rahmen eines Modells. Ein Modell liefert uns die Konzepte, Abstraktionen und Organisationshilfen, um wichtige Aspekte der Realität zu erfassen. D.h., die Realität wird anhand des Modells in eine vereinfachte Struktur gebracht.

Diese Vorgehensweise trifft man auch ausserhalb der EDV-Welt an. Sie dient dazu, von der Komplexität der Realität zu abstrahieren und ihre Wesenszüge leichter zu verstehen. Natürlich gibt es verschiedene Modelle, die für bestimmte Sichtweisen optimiert sind, um sich auf bestimmte Aspekte zu konzentrieren.

Dies lässt sich am Beispiel eines Hauses illustrieren:

Der Architekt verwendet Modell 1, um das Haus zu zeichnen. Der Schwerpunkt liegt auf der stereometrisch korrekten Darstellung der Hauses. Ein Makler erstellt ein Exposé nach Modell 2, während der zukünftige Bewohner vor allem einen Eindruck von der Innengestaltung gewinnen will; dazu ist Modell 3 geeignet. In Wirklichkeit geht es um das gleiche Objekt, es wird nur in unterschiedlichen Modellen (Abstraktionen) dargestellt.

Wir stellen im folgenden Modelle der Informatik zur Datenstrukturierung vor.



## 2.1 Entity-Relationship-Diagramm nach P.P. Chen (Beispiel mit Attributen)

◆ Das Entity-Relationship Modell (ERM) basiert auf der Anschauung, daß Realität und Vorstellungswelt durch Objekte (*Entitäten*) und ihre Beziehungen (*Relationships*) untereinander dargestellt werden kann.

**Beispiel:** Ein ERM zur personenbezogenen Projektabrechnung, besteht aus:

- 2 Entitätsmengen (MITARBEITER, PROJEKTE)
- 5 Attribute (MA-Nr , Name , Adresse, Projektname, Budget)
- 1 Beziehung (arbeitet) mit einem Attribut (Stunden) vom Typ *m : cn*
- 8 Verbindungen

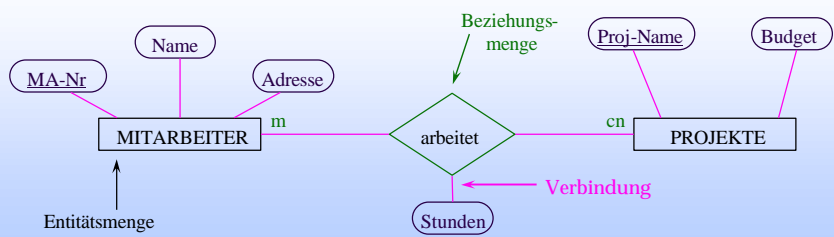


Bild 2.1 -5
Datenbank- und Informationssysteme
Sep-02 (c) Prof. Dr. F. Laux

Wir wollen zuerst das klassische Entity-Relationship Modell (ERM) nach P.P. Chen (1976) kennenlernen. Es geht davon aus, dass sich die Welt durch genau abgrenzbare Exemplare von Objekten, Personen, Begriffe, Ideen, etc. (**Entitäten**) zusammensetzt und dass zwischen diesen Beziehungen (**Relationships**) bestehen. Es wird angenommen, daß sich die zu modellierenden Systeme durch diese beiden Begriffe hinreichend beschreiben lassen. Zu den Entitäten und Beziehungen gehören noch charakterisierende Größen, Attribute genannt. Jede Entität muß durch ein Attribut oder eine Attributkombination eindeutig identifizierbar sein.

Das ERM macht keine Voraussetzungen über den Computer oder die verwendete Software, d.h. es ist ein computerunabhängiges Modell. Entitäten und Beziehungen des ERM können klassifiziert werden, so dass eine Menge von gleichartigen Elementen unter einem Typ zusammengefasst werden kann.


Die Darstellung von Informationen im ER-Modell erfolgt auf zwei Detaillierungsebenen:

- der *obere Begriffsbereich* mit folgenden Elementen:  
Entitätstypen, Beziehungsmengen
- der *untere Begriffsbereich* mit folgenden Elementen:  
Entitätsattribute, Beziehungsattribute

Vorgehensweise:

Zur Modellierung müssen zunächst die Entitätstypen/-mengen definiert werden; anschließend werden die Zusammenhänge zwischen ihnen festgelegt. Die Attribute (Elemente des unteren Begriffsbereichs) können anstatt im Diagramm auch in Form einer Liste mit Datentyp und Wertebereich erfasst werden.

Die Darstellung erfolgt in der nachfolgend dargestellten graphischen Notation.



### 2.1.1 Syntax des klassischen ER-Modells






Symbol	Elementtyp	Beschreibung
	<b>Entitäts-Typ/Menge</b> (entity type/set)	Eine Menge von unterscheidbaren Objekten (eines Typs) der realen oder Gedankenwelt. <b>Beispiele:</b> Personen, Dinge, Aufträge, Idee
	Schwache Entitäten (weak entity)	= abhängige Entitäten; <b>Beispiele:</b> Adressen, Auftr.Position
	<b>Beziehungsmenge</b> (relationship set)	Beziehungen/Relation zwischen 2 oder mehr Entitätsmengen. <b>Beispiele:</b> Person-Wohnort-Beziehung (wohnt in)
	<b>Attribut</b> (attribute)	Attribut (Eigenschaft) eines Entitätstyps. Dient der Beschreibung und Identifikation von Entitäten. <b>Beispiele:</b> Name, Preis, Farbe, Ident-Nr.
k 	<b>Verbindung</b> (association) existentielle Abhängigkeit	Verbindung zwischen Entitäts- und Beziehungs- bzw. Attributmenge. k := Anzahl Verbindungen/Entität <b>Beispiel:</b> Student - Matrikel-Nr (1 - 1) <b>Beispiel:</b> Auftragskopf -Positionen (1-n)

Bild 2.1 -6
Datenbank- und Informationssysteme
Sep-02 (c) Prof. Dr. F. Laux

**Beispiele von Entitätstypen und (Entitäten)**

- *Mensch* (Albert Einstein),
- *Student* (Eva-Maria Fleissig),
- *Firma* (Hugo Boss)
- *Produkt* (Schokolade),
- *Gebäude* (Gebäude 9 der FH Reutlingen)
- *Konzept* (ERM),
- *Fachgebiet* (Datenbanken),
- *Veröffentlichung* (P.P. Chen: "The Entity-Relationship Model - Toward a Unified View of Data" ACM TODS Vol. 1, No 1, March 1976, pp. 9 - 36)

**Beispiele von Beziehungsmengen**

- "Ist beschäftigt bei" ist eine 1:n Beziehung zwischen *Firma* und *Mitarbeiter*
- "besucht" ist eine n:m Beziehung zwischen *Student* und *Vorlesung*

**Beispiele von Attributen**

- Name, Geburtstag, etc. sind Attribute des Entitätstyps *Mensch*
- Albert, 14. März 1879 sind Attribute der Entität *Albert Einstein*
- Matrikelnr, Semester, etc. sind Attribute zum Entitätstyp *Student*
- Keywords, Zeitschrift, etc. sind Attribute eines Entitätstyps *Veröffentlichung*
- {Datenmodell, semantisch, Entität, ...}, Transactions on Database Systems sind Attribute der 1. Veröffentlichung über das ERM von P.P. Chen

## 2.1.2 Kardinalität der Beziehungen

- ◆ Seien E1 und E2 Entitätsmengen,  $k_1$  bezeichne die Anzahl von Verbindungen von  $e_1 \in E1$  in die Menge E2. Wir unterscheiden
  - ◆  $k_1 = c$  (höchstens eine, d. h. 0 oder 1)
  - ◆  $k_1 = 1$  (genau eine)
  - ◆  $k_1 = m$  (mindestens eine, d.h. 1 bis m)
  - ◆  $k_1 = cn$  (keine, eine oder mehrere, d. h. 0 bis n)

Daraus ergeben sich 16 Kombinationen für einen Beziehungstyp zwischen E1 und E2.

- ◆ Beispiele:
 

1 - cn

1 - 1

cn - cm

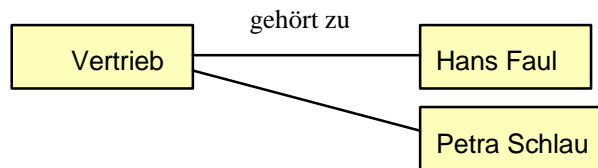
Bild 2.1 -7
Datenbank- und Informationssysteme
Sep-02 (c) Prof. Dr. F. Laux

### Beziehungen

Eine Beziehung (oder Relationship) ist eine wechselseitige Zuordnung von zwei oder mehr Entitäten. Beziehungen können in verschiedene Beziehungstypen klassifiziert werden.

Beispiel für eine Beziehung:

Zu einem bestimmten Zeitpunkt sind die Mitarbeiter 'Hans Faul' und 'Petra Schlau' der Abteilung 'Vertrieb' zugeordnet. Ein Beispiel der *Instanzebene* könnte so aussehen:



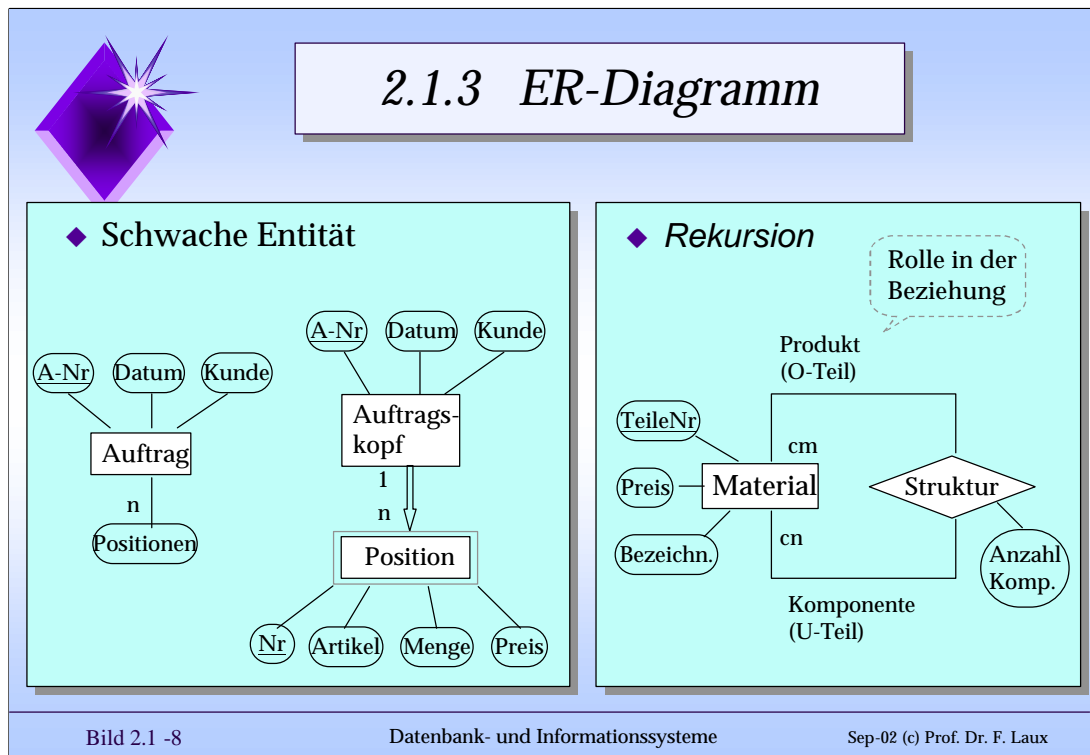
### Beziehungstypen

Für andere Mitarbeiter existieren ebenfalls Zuordnungen zu Abteilungen.

Alle diese Zuordnungen haben folgendes gemeinsam: ein Mitarbeiter gehört nur zu einer Abteilung, umgekehrt können viele Mitarbeiter zu ein und derselben Abteilung gehören. Damit ist die Beziehung zu einem Typ 1:n klassifiziert. (Beziehungstyp)

Diese Sachverhalt wird graphisch im folgenden Bild dargestellt (*Strukturebene*):





### Schwache Entität

Gelegentlich ist es schwierig, Attributgruppen oder mengenwertige Attribute von Entitäten zu unterscheiden. In solchen Fällen bedient man sich einer 'schwachen Entität'. Wir verstehen darunter eine im Kontext einer anderen Entität (*starke Entität*) eindeutig identifizierbare Einheit, die ohne diese nicht existieren kann (*abhängige Entität*).

In unserem Beispiel ist die *Position* eine schwache Entität des *Auftrags*. Die Positionsnummer (*Nr*) identifiziert die *Position* innerhalb eines *Auftrags*. Eine *Position* ist ohne Bezug zum *Auftrag* bedeutungslos. Häufig ist eine Beurteilung, ob eine Entität schwach oder stark sein soll, vom fachlichen Kontext bestimmt. In einer Personaldatei sind die Adressen und Kinder der Mitarbeiter abhängige Entitäten oder Attribute, da beim Ausscheiden aus dem Betrieb diese Information nicht mehr benötigt wird. Im Katasteramt werden sicherlich die Adressen als eigenständige Entitäten betrachtet; ebenso führt das Einwohnermeldeamt die Kinder als (*starke*) Entitäten.

### Rekursion

Entitäten können Beziehungen zu Entitäten des eigenen Typs besitzen. Damit lassen sich Netzwerke *eines* Entitätstyps aufbauen.

Die *Produkt/Komponenten-Beziehung* ist ein Beispiel für ein Netzwerk, das zwei gerichtete Beziehungen (*Stückliste* und *Verwendungsnachweis*) enthält. Das *Material* spielt je nach Beziehungsrichtung unterschiedliche Rollen (*Produkt/Oberteil* oder *Komponente/Unterteil*). Da das ERM nur ungerichtete Beziehungen kennt, ist diese Differenzierung nur durch Rollennamen (*Kommentare*) deutlich zu machen. Deshalb hat auch das Attribut der Beziehung *Struktur* einen erklärenden Namen erhalten.

Eine bessere Präzisierung des Beziehungstyps wäre nicht nur bei Rekursionen wünschenswert. Dieser Mangel wird durch mehrere Erweiterungen beseitigt, die das ERM erfahren hat.



### 2.1.4 Syntax des erweiterten ER Modells


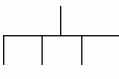


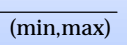
Symbol	Elementtyp	Beschreibung
	<b>Generalisierung/ Spezialisierung</b> (is a)	Generalisierung oder Spezialisierung eines Entitätstyps. <b>Beispiel:</b> Mitarbeiter - Angestellte, Arbeiter, Abteilungsleiter...
	<b>Aggregat/ Komponenten</b> (part of)	Aggregat/Komponentenbildung eines Entitätstyps. <b>Beispiel:</b> Auftrag - Auftragskopf, Auftragspositionen
	<b>abhängige Entität</b>	Zur schwachen Entität erhobene Beziehung (erbt die Identifikation von benachbarten Entitäten)
	<b>Gruppierung</b>	Teilmenge/Klassifizierung einer Entitätsmenge <b>Beispiele:</b> verheiratete Studenten, weibliche Angestellte
	<b>Kardinalität</b> einer binären Beziehung	Bezeichnet die Anzahl (Minimal- und Maximalwert) der möglichen Verbindungen, die eine Entität eingehen kann. <b>Beispiele:</b> cn := (0,n), d.h. 0 bis n Verbindungen; (5,12), d.h. 5 bis 12 Verbindungen

Bild 2.1 -9
Datenbank- und Informationssysteme
Sep-02 (c) Prof. Dr. F. Laux

Vor allem die Beziehung (relationship) des ERM hat Erweiterungen erfahren:

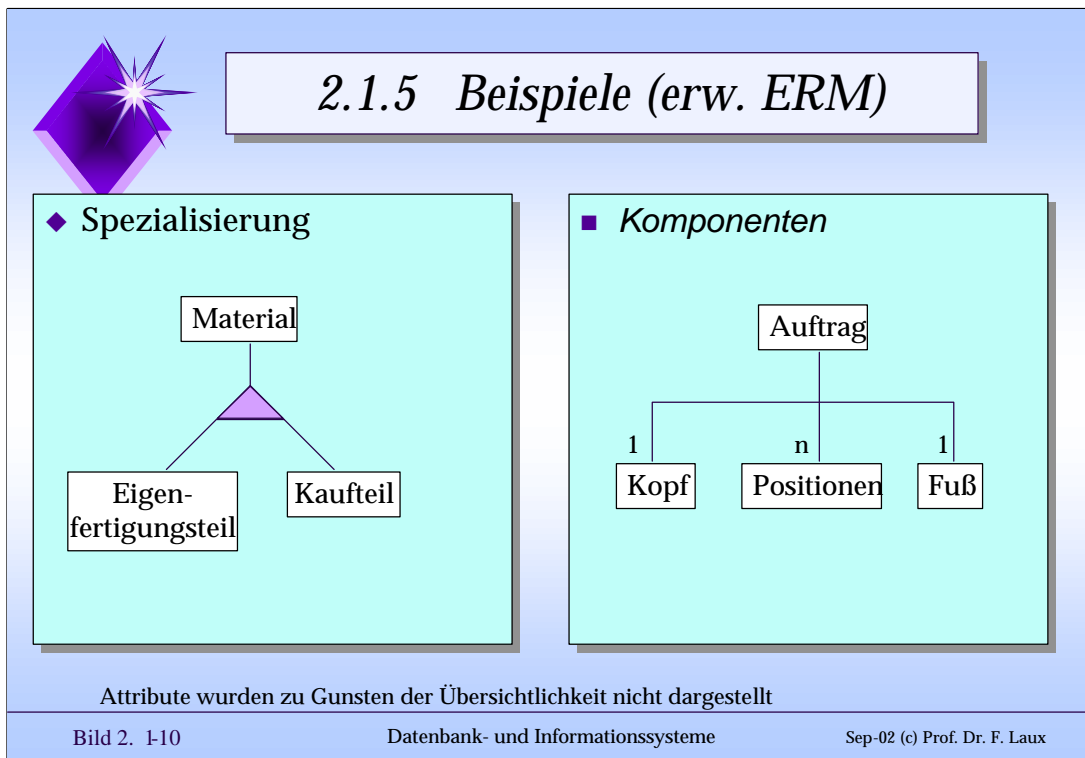
Wenn ein Entitätstyp eine **Spezialisierung** eines anderen darstellt, wird dies durch ein Dreieck anstelle der Raute angezeigt. Dabei zeigt die Spitze des Dreiecks zum allgemeineren Entitätstyp (Generalisierung). Spezialisierungen eines Entitätstyps (z.B. Mitarbeiter) können *disjunkt* (Angestellte, Arbeiter) oder *überlappend* (Angestellter, Programmierer) sein. Jede Entität einer Spezialisierung gehört auch zu einer allgemeineren (übergeordneten) Entitätsmenge (Generalisierung).

Entitäten, die zu einer anderen gehören und nur zusammen mit dieser existieren, bezeichnet man als **Komponenten**. D.h. die Komponenten bilden eine neue Entität, das sogenannte **Aggregat**. Diese Beziehung ist dadurch charakterisiert, dass die Komponenten (Teile) nur zu einem Aggregat gehören und von diesem abhängig sind. Aggregat/Komponentenbeziehungen sind daher 1:n-Beziehungen. Beispiel: Aggregat Auftrag, Komponenten: Auftragspositionen

Gelegentlich können Aggregate auch aus unabhängigen Komponenten bestehen, die eventuell an der Bildung mehreren Aggregate teilhaben. Dies wird dadurch ausgedrückt, dass die *Beziehung*, welche die Teile *zu einem Aggregat zusammenfügt*, als eine (**abhängige**) **Entität** dargestellt wird. Die Schlüssel aller Komponenten bilden die Identifikation der schwachen Entität. Damit ist klar, daß diese Entität nur existiert, wenn auch ihre Komponenten existieren. Als Beispiel hierfür können wir ein Vorlesungsverzeichnis betrachten, das aus den Komponenten Veranstaltung, Dozent und Raum besteht. Ohne diese Teile ist das Vorlesungsverzeichnis nicht existent. Eine Komponente (z.B. Raum) kann auch anderweitig verwendet werden.

Innerhalb einer Entitätsmenge kann ein Attribut zur **Gruppierung** der Menge herangezogen werden. Der Unterschied zur Spezialisierung ist der, dass die Entitäten der Gruppe die gleichen Attribute besitzen wie die Entitätsmenge selbst. Bei Spezialisierungen hingegen können weitere Attribute oder sonstige Einschränkungen hinzukommen.

Zur Präzisierung von binären Beziehungen wurde eine min/max.-Notation für die **Kardinalität** eingeführt.



**Spezialisierung**

Hier handelt es sich um eine Spezialisierung die nicht disjunkt sein muss. Ein Material kann sowohl Eigenfertigungs- als auch Kaufteil sein. Die Kardinalität der Beziehungen muss nicht angegeben werden, da sie jeweils 1 ist.

ÜA: finden Sie Attribute für die drei Entitäten.

**Komponenten**

In diesem Beispiel ist ein Auftrag als Aggregat der Komponenten Kopf, Positionen und Fuß dargestellt. Die Kardinalitäten geben an, dass zu einem Auftrag mehrere Positionen und genau ein (Auftrags)kopf und -fuß gehören. Wie die Positionen strukturiert sind, ist in unserem Modell nicht angegeben.

Hinweis: Bei der Implementierung werden häufig die Attribute der Komponenten mit der Kardinalität 1 zusammen mit den Attributen des Aggregats gespeichert. Am Beispiel der Auftragsstruktur würde dies bedeuten, dass die Attribute von Auftrag, Kopf und Fuß zusammen gespeichert werden, während die Positionen getrennt davon abgelegt werden.

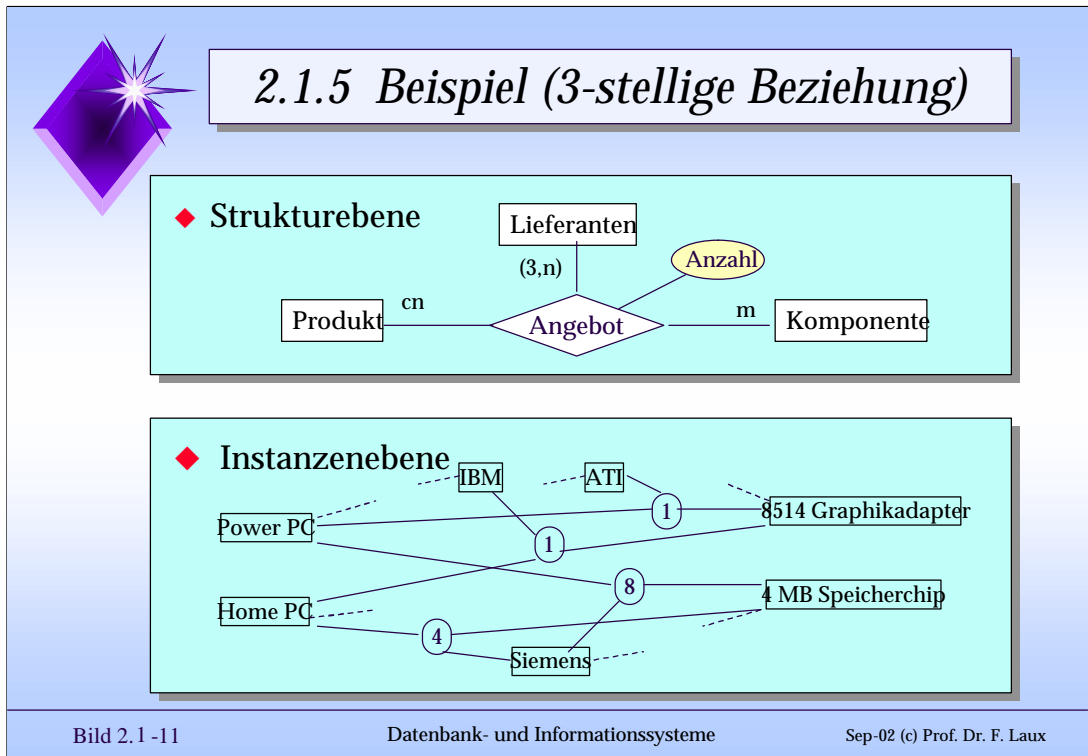


Bild 2.1-11

Das Angebot wird als 3-stellige Beziehung zwischen *Lieferant*, *Produkt* und *Komponente* aufgefasst. Die Kardinalität bei jeder Entität gibt an, an wievielen Verbindungen (Angeboten) diese teilnehmen kann. Der Grad der Beziehung (Kardinalität) bei Lieferant ist also so zu lesen, daß für jede Produkt/Komponentenkombination mindestens 3 Lieferanten anzugeben sind. Man beachte, dass nicht für jedes Produkt ein Angebot vorhanden sein muss. Allerdings muss ein Angebot sich wenigstens auf eine Komponente beziehen. Das Attribut *Anzahl* gibt an, wieviele Komponenten ein Angebot zu einem Produkt enthält.

ÜA: Erstellen Sie analog zu o.g. Beispiel ein ER-Diagramm für eine Bibliotheksausleihe mit den Entitätstypen Buch, Exemplar, Ausleiher, Leihfrist und Standort.



## 2.1.6 Methodik der ER-Modellierung

Nach P.P. Chen

- ◆ Festlegen der Entitäts- und Beziehungsmengen
- ◆ Identitätsschlüssel für Entitäten bestimmen
- ◆ Festlegen der Beziehungskardinalitäten
- ◆ Attribute und ihre Wertebereiche ermitteln
  
- ◆ Entity-Relationship-Diagramme (ERD) für Entitäts- und Beziehungsmengen zeichnen
- ◆ Attribute und ihre Wertebereiche in Tabellen eintragen

---

Bild 2.1 -12
Datenbank- und Informationssysteme
Sep-02 (c) Prof. Dr. F. Laux

Dieses Vorgangsmodell entspricht dem Top-Down Prinzip. Zuerst werden die Elemente des oberen Begriffsbereichs ermittelt. Dann sind die Identitätsschlüssel festzulegen, damit die Beziehungskardinalitäten und Attribute gewählt werden können.

Die Elemente der oberen Begriffswelt (Entitäten, Relationships) werden in ER-Diagrammen und die der unteren (Attribute) in Tabellenform dargestellt, um die graphische Darstellung nicht zu überfrachten.

**Beispiel:** (siehe auch 2.1-8)

