

**C1TT**

No & Low Code

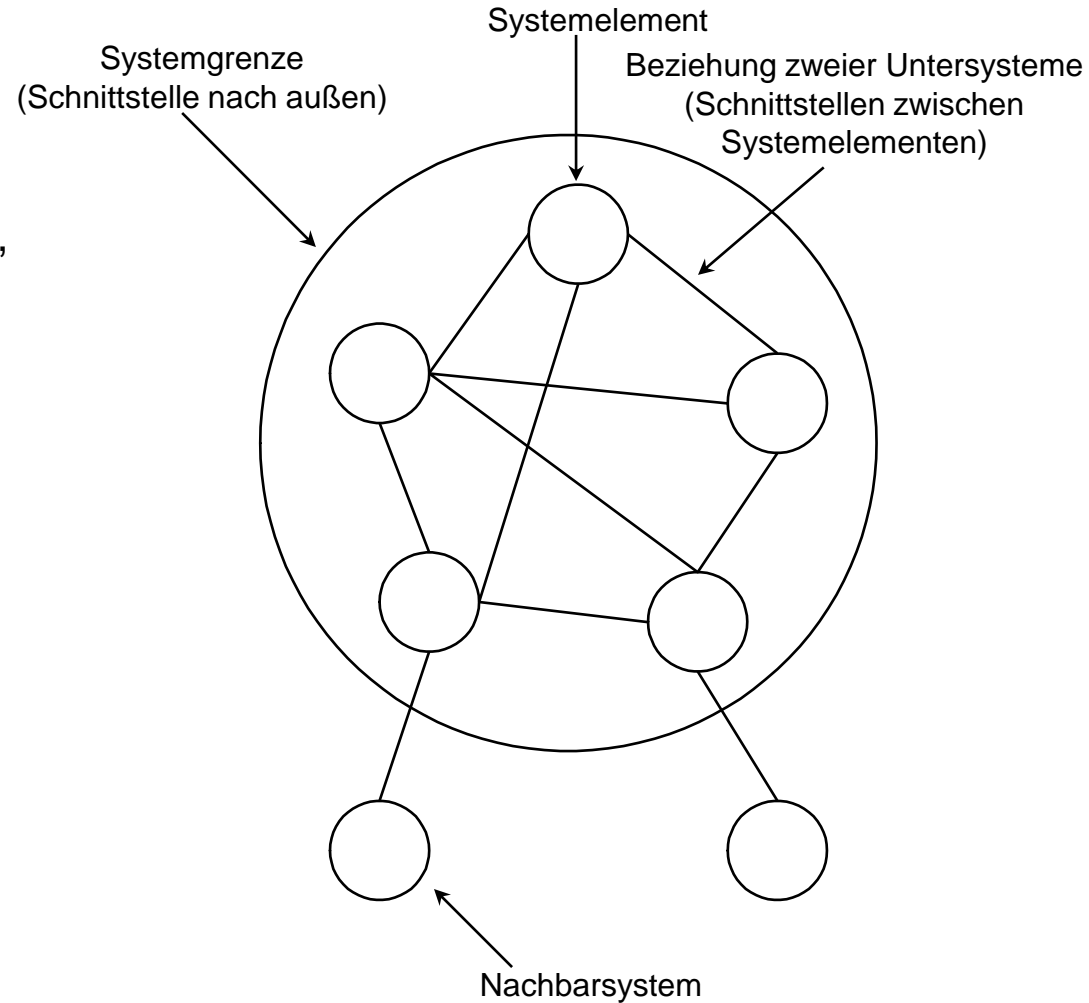
**C1TT** Continental Institut  
für Technologie  
und Transformation

# Datenbanken

## Kapitel 1

# Einführung in die Datenmodellierung

Ein **System** besteht aus einer Anzahl von miteinander in Beziehung stehenden Elementen, die bestimmte Eigenschaften aufweisen, in einem abgegrenzten Bereich.



- **Systeme**
  - können offen oder abgeschlossen sein
  - haben Grenzen
  - lassen sich unterteilen (in Subsysteme oder Systemkomponenten)
- **Ein IT-System umfasst:**
  - Daten (Informationen)
  - Funktionen (Aktivitäten, Prozesse),  
in denen diese Daten bearbeitet oder verarbeitet werden
  - Abläufe, anhand derer diese Daten verarbeitet werden
- **Um ein System aussagekräftig zu beschreiben, müssen also modelliert werden:**
  - die Daten des Systems
  - die Funktionen des Systems
  - die Abläufe im System

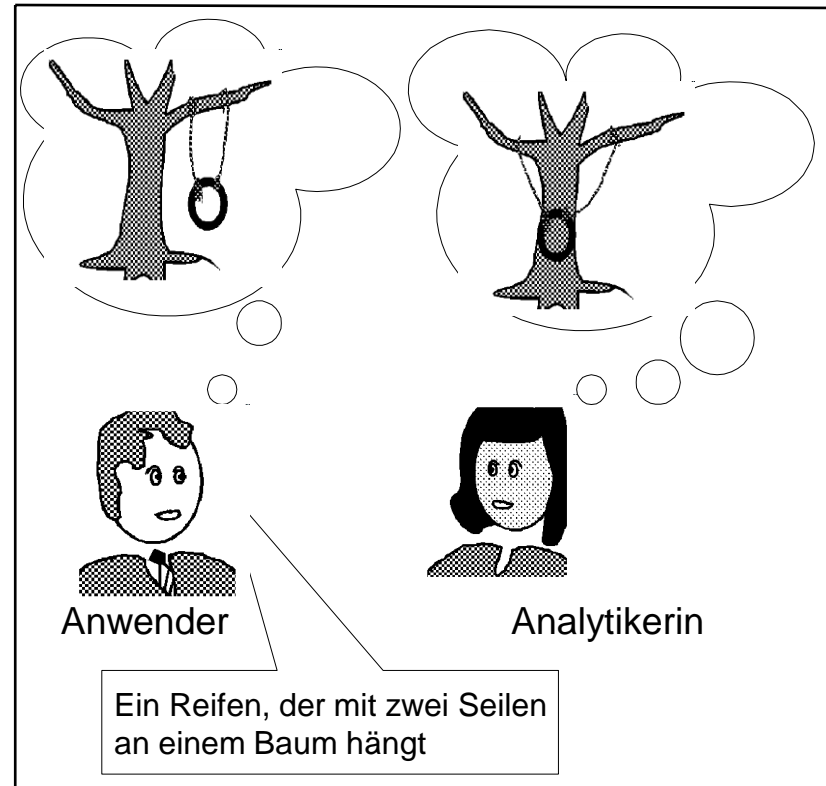
Eine Kunst – aber auch eine Wissenschaft

Die Kunst:            Struktur in ein unstrukturiertes  
Problem zu bringen

Die Wissenschaft:            Die Struktur in eine  
verwertbare  
Form umzusetzen

## Erste Phase: Kommunikation

Den Anwenderwünschen **zuhören!!**





## Kommunikation

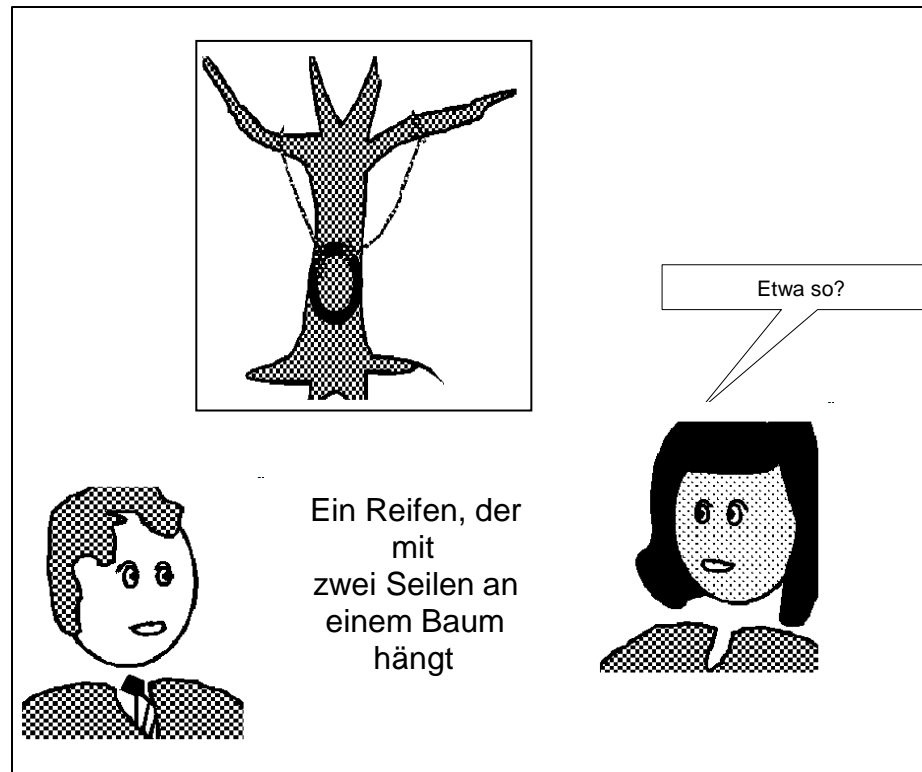
Anforderungen mit eigenen Worten formulieren –  
Habe ich sie richtig verstanden?





## Zweite Phase: Repräsentation

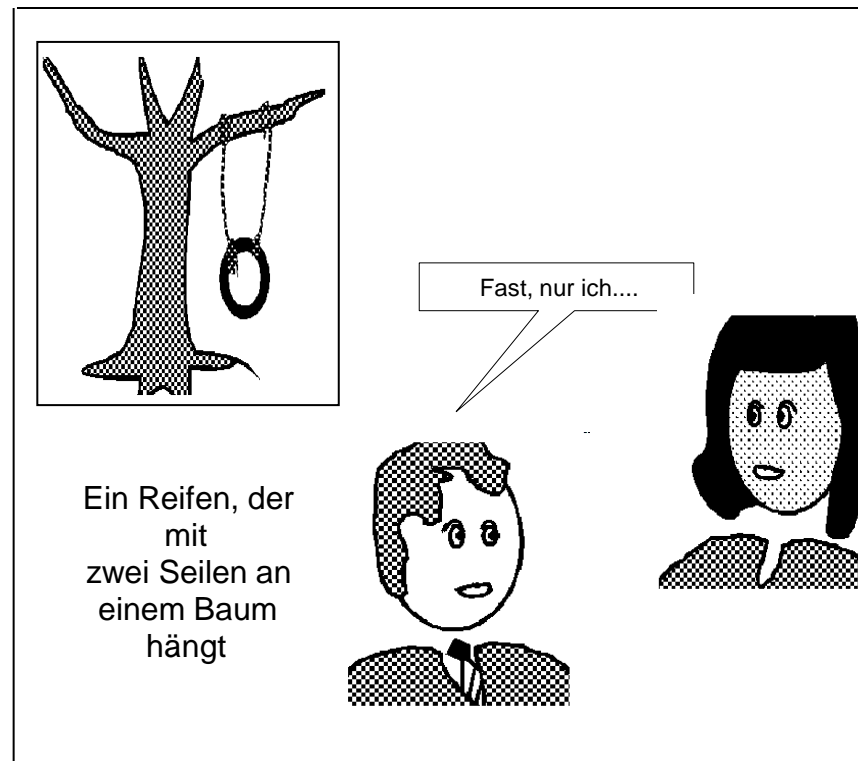
Darstellung erfolgt am besten grafisch.



und dann: Verifikation

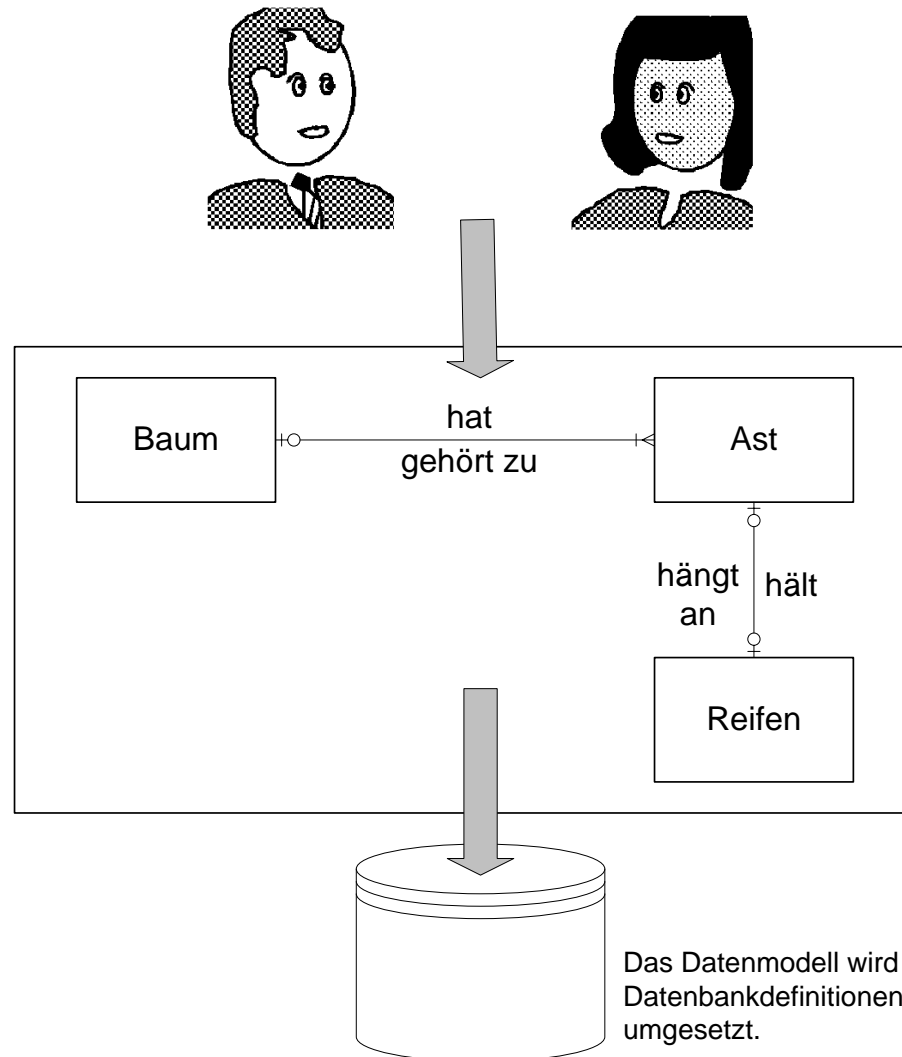
Hat der Anwender unsere Darstellung verstanden?

Ist der Anwender mit unserer Darstellung einverstanden?



## ■ Die Datenmodellierung

- beschreibt die Daten (Entities) und deren Beziehungen (Relationships) innerhalb eines Systems. Es war eines der ersten konzeptionellen Modelle.
- kann Ist-Modelle und Soll-Modelle beschreiben.
- kann ein Modell für ein komplettes Unternehmen (Unternehmensdatenmodell), einen Bereich (Bereichsdatenmodell) oder ein Projekt erstellen.
- ist eine graphische Darstellungstechnik, eine "graphische Sprache" (ein Bild sagt mehr als 1000 Worte!).
- basiert auf wenigen graphischen Elementen. Diese Sprache ist rasch zu erlernen. Die Dokumente sind leicht lesbar und verständlich und das ohne weitere Hilfsmittel.



## Ziel:

- eine stabile Grundlage für Informationssysteme schaffen
- Grundlage schaffen für Hard- und Software Unabhängigkeit
- Daten unabhängig von der Anwendung speichern

## Wesentliche Problemfelder:

- mangelnde Transparenz des Istzustandes
- mangelhafte Planung und Koordination
- Kommunikationsprobleme / Schnittstellenprobleme
- unkontrollierte Datenredundanzen

## Lösungsansatz:

- Implementierungsunabhängige Datenanalyse
- Orientierung an fachlichen Zusammenhängen
- Aufbau der Daten für alle Systeme unabhängig von den jeweiligen Anwendungen
- Verwendung eines gemeinsamen Kommunikationsmediums von Anwendungs- und IT-Fachleuten
- Vermeidung von Redundanzen

```
DATA DIVISION.  
FILE SECTION.  
  
FD KUNDE LABEL RECORDS ARE OMITTED  
01 KUNDEN-SATZ.  
   05 NAME.  
       10 VORNAME      PIC X(20).  
       10 NACHNAME     PIC X(25).  
       10 TITEL        PIC X(10).  
   05 ADRESSE.  
       10 STRASSE      PIC X(20).  
       10 HAUSNR       PIC X(5).  
       10 PLZ          PIC S9(6) COMP.  
       10 ORT          PIC X(20).
```

```
DATA DIVISION.  
FILE SECTION.  
  
FD KUNDEN LABEL RECORDS ARE OMITTED  
01 KU-REC.  
   05 KU-NAM.  
       10 KU-VOR       PIC X(25).  
       10 KU-NAM       PIC X(30).  
       10 KU-GEB       PIC 99.99.9999.  
       10 KU-GES       PIC X.  
   05 KU-ADR.  
       10 KU-STR       PIC X(30).  
       10 KU-PLZ       PIC X(6).  
       10 KU-ORT       PIC X(25).  
       10 KU-POSTF     PIC 9(8).
```

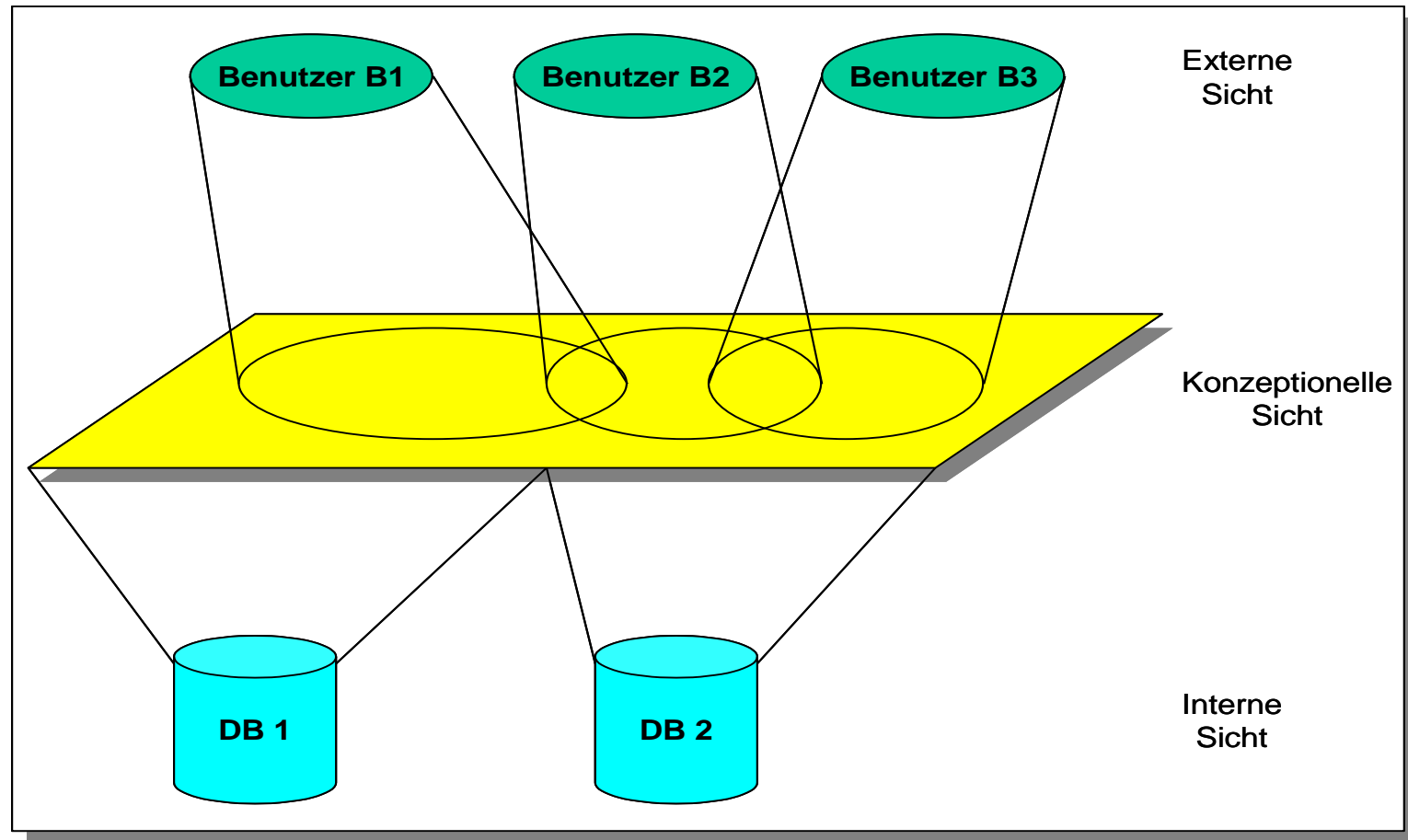
## Kapitel 2

# Das Drei-Sichten-Modell

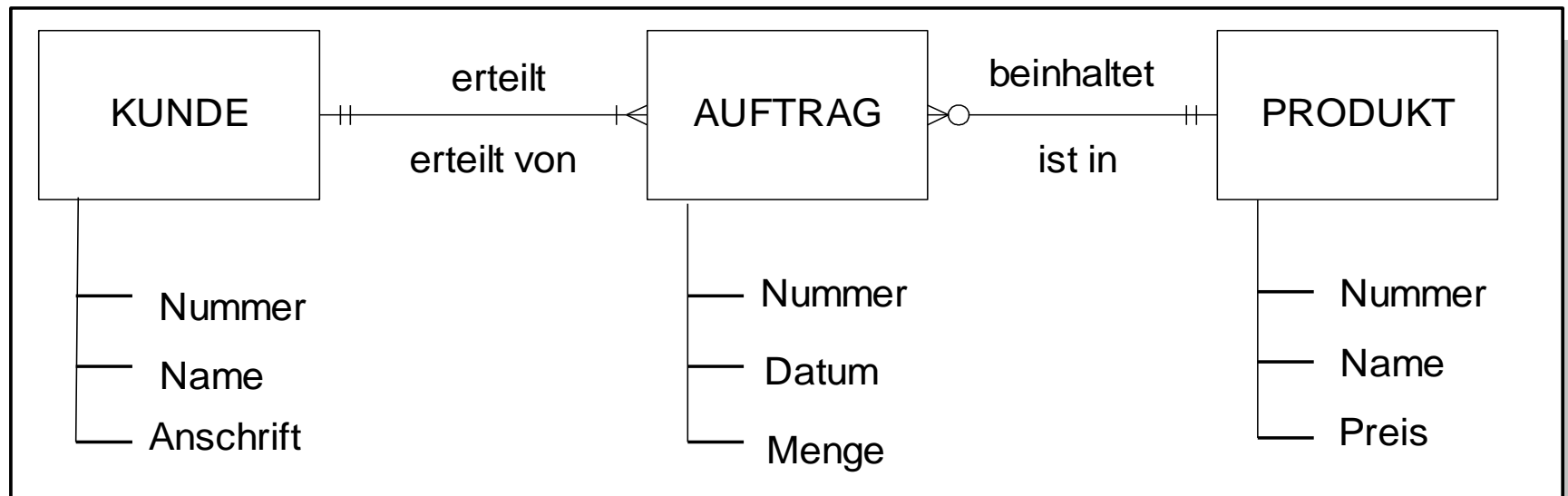


Ein Modell zeigt - abstrahiert –  
den für einen bestimmten Zweck wichtigen Teil der Welt.

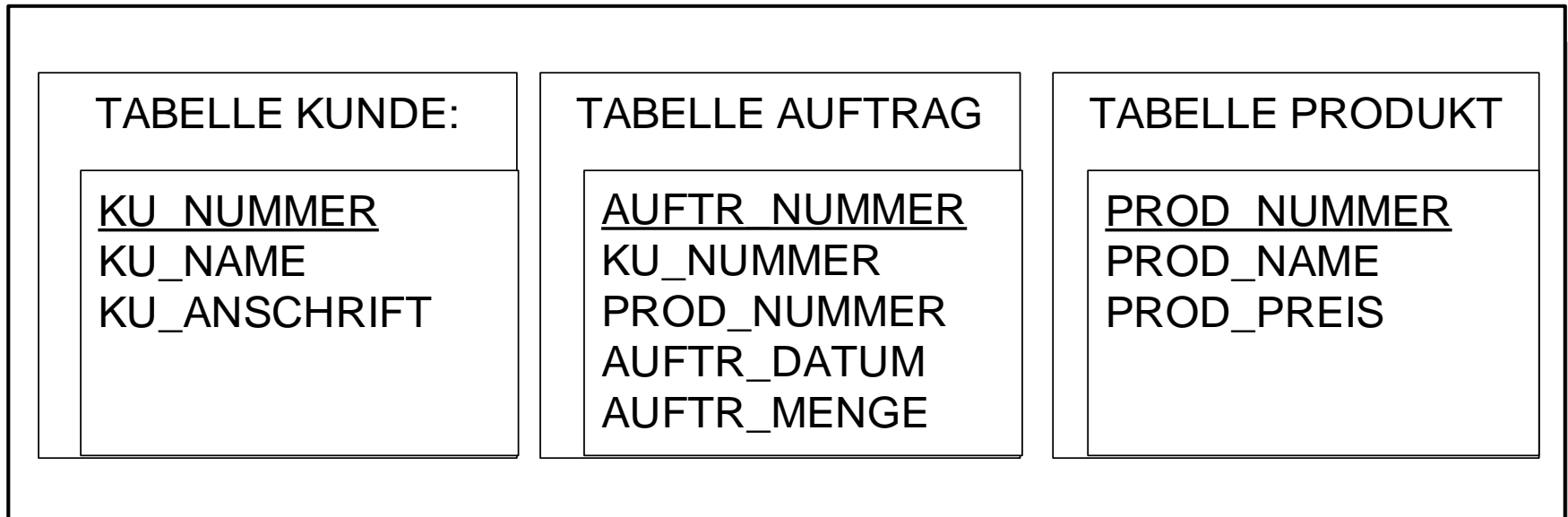
- Ein gutes Beispiel für Modelle sind Landkarten:
  - Wir könnten theoretisch ein Satellitenbild verwenden.
  - Eine Alternative ist eine Straßenkarte.
  - Die dritte Möglichkeit wäre eine Straßenskizze.



# Abbildung des konzeptionellen Modells durch das ER-Modell



# Abbildung des internen Modells durch das relationale Modell



## KUNDENLISTE WEIHNACHTS-DIREKT-MAIL:

Alle Kunden, die über 50.000 AUFTR\_MENGE im letzten Monat hatten,  
sortiert nach KU\_NUMMER:

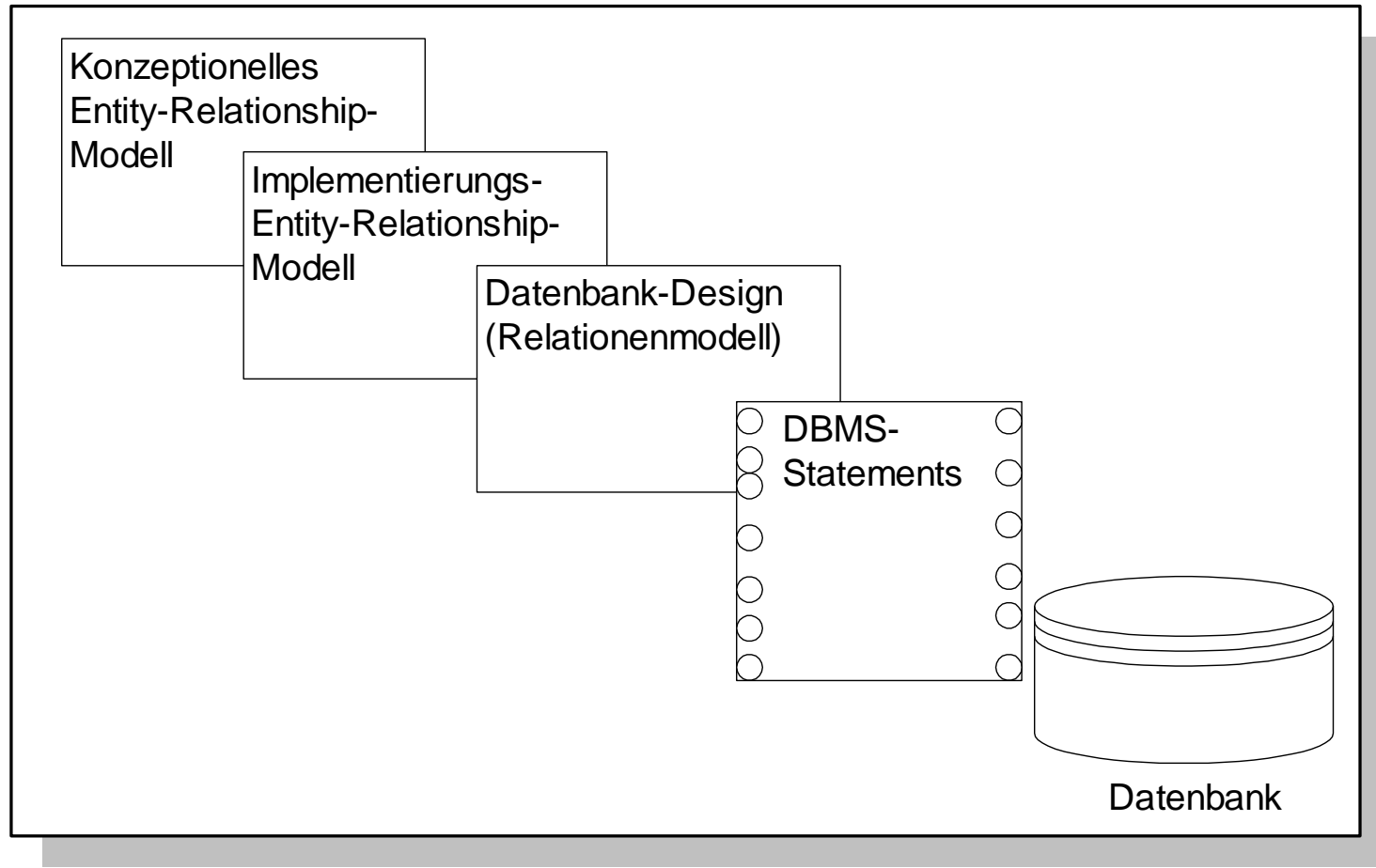
KU\_NUMMER

KU\_NAME

PROD\_NAME (mit dem größten Auftragsumfang)

davon: AUFTR\_MENGE

GESAMT\_PREIS

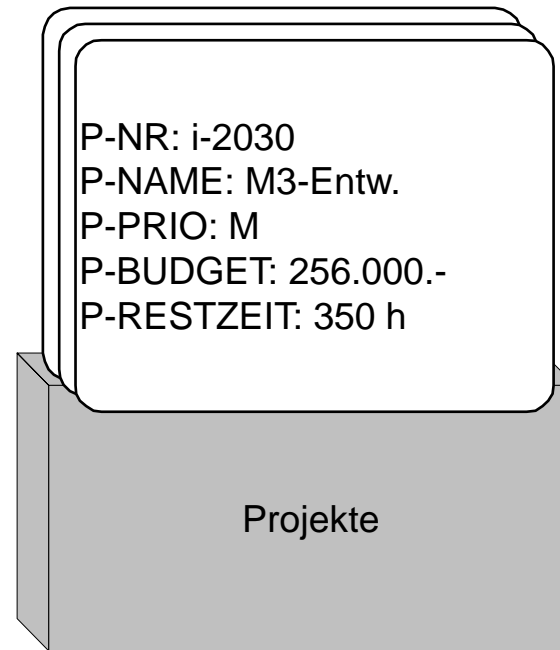
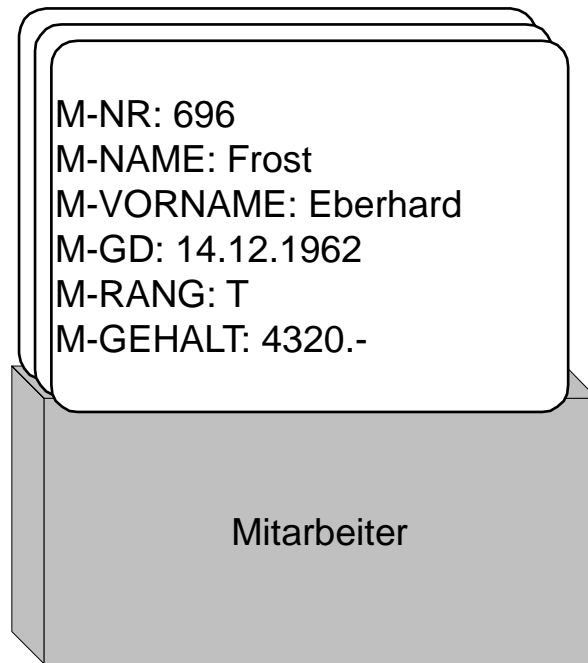


## Kapitel 3

### Grundlagen ERM: Entitätstypen



- Entität und Entitätstyp
- Beziehung und Beziehungstyp
- Attribut und Attributtyp

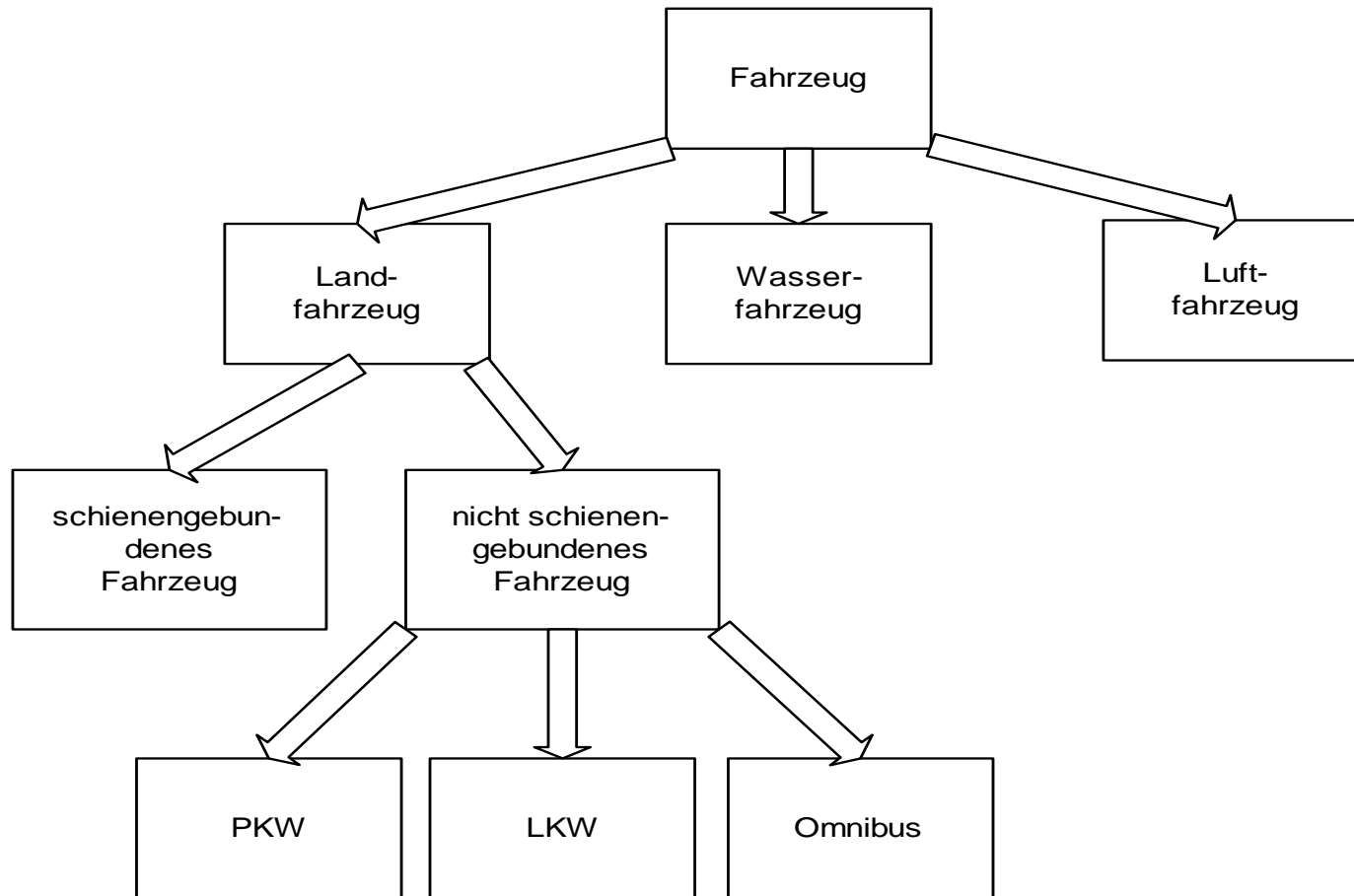


Der Mitarbeiter Herr Frost ist ein konkretes, individuell identifizierbares Objekt,

Er gehört zu einer Klasse, der Klasse der Mitarbeiter.

Entitäten:	Mitarbeiter Frost Eberhard,	M-Nr.696
	Mitarbeiter Bonello Leonardo,	M-Nr.737
	Mitarbeiter Meier Hugo,	M-Nr.314
	Mitarbeiter Meier Karl,	M-Nr.425

# Entitäten können selbst Klassen sein



## 1. Frage an die Fachabteilung

An die Fachabteilung!

Bitte geben Sie bis Freitag, den 13.,  
Ihre Entitäten in der Abteilung Organisation und IT ab.

Vielen Dank und  
mit freundlichen Grüßen

## 2. Re-Design und Wartungsprojekte

- Formulare (Erfassung, Antrag, Änderung, ...)
- Listen (interne Listen, externe Listen)
- Masken (Eingabe, Ausgabe, Änderung, Anzeige)
- Datenbeschreibungen (Makros, Tabellen, ...)
- fachliche Beschreibungen

## 3. Neuentwicklung von Systemen




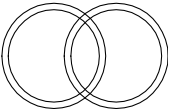
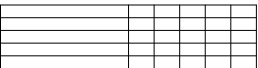
- Analyse der Fachanforderung
- Analyse von Besprechungsprotokollen
- Durchführen von Interviews
- Workshop mit der Fachabteilung

### Identifizieren von Objekten

- **Jedes Hauptwort** auf einem Formular und im Interview ist ein Kandidat für einen Entitätstyp.

Beispiel:	"Mitarbeiter arbeiten praktisch immer in Projekten, wobei diese verschiedene Prioritäten haben können."	
	Entitätstypen:	Mitarbeiter, Projekt
	Attributtypen:	Priorität (von Projekt)

# In Anlehnung an Shlaer/Mellor gibt es zur Identifizierung von Objekten folgende Kategorien:

Kategorie	Beschreibung	Beispiele
<b>Reale Dinge</b> 	Körperliche Dinge, über die das Unternehmen Informationen speichert.	Flugzeug Buch Gebäude Fahrzeug Maschine
<b>Rollen</b> 	Personen sind Objekte, die in unterschiedlichen Rollen im Unternehmen auftreten.	Dozent Kunde Mitarbeiter Vertreter Eigentümer Patient
<b>Ereignisse</b> 	Geschäftsprozesse, die sich beim Unternehmen ereignen.	Zeitpunkt Vertragsabschluß Unfall Anruf Kündigung Eingang Gutschrift
<b>Interaktionen</b> 	Interaktionsobjekte beziehen sich auf zwei Objekte (der Kauf z.B. auf den Käufer und das gekaufte Produkt)	Heirat Kauf Projektmitarbeit
<b>Instanzen</b> 	Instanzen repräsentieren einen Standard oder eine Verallgemeinerung von Objekten.	Vertragstypen Nutzfahrzeuge Säugetiere



- Der Name eines Entitätstyps ist, entsprechend seinem Charakter ein Hauptwort oder ein zusammengesetztes Hauptwort in der Einzahl, aber kein Tätigkeitswort
- Beispiele:
  - Kunde
  - potentieller Kunde
  - Käufer
  - Kaufvertrag
  - **aber nicht:** Vertrag abschließen (das ist ein Prozess).

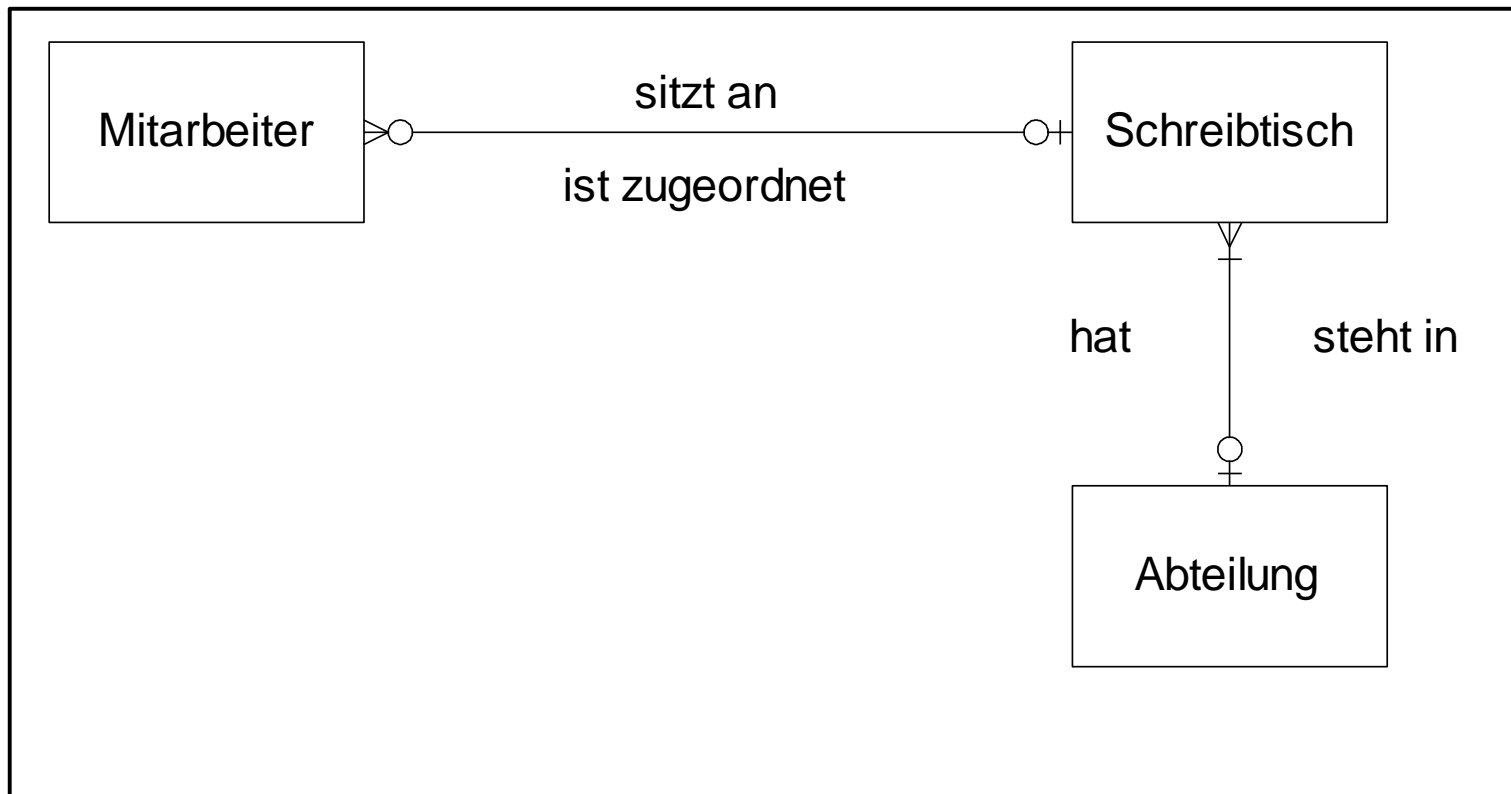
Es ist darauf zu achten, dass die Namen

- verständlich und eindeutig sind
- den Namenskonventionen der Fachabteilung entsprechen
- keine Punkte, Schrägstriche oder Sonderzeichen enthalten
- möglichst kurz sind, denn sie müssen in die Kästchen passen.

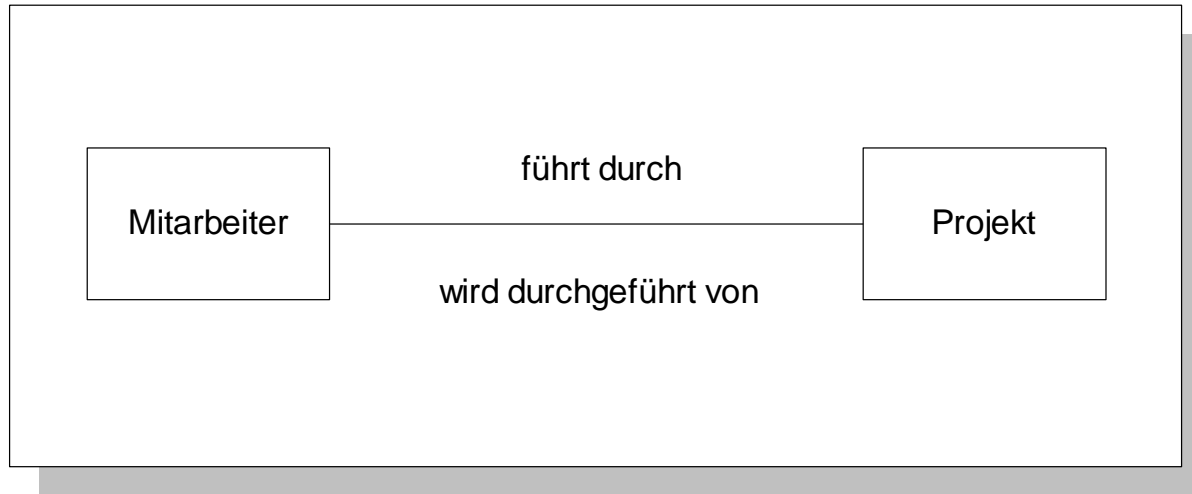
- Beispiel:
  - Kunde: Person, die Produkte bei uns gekauft hat
  - Interessent: Person, die unseren Katalog angefordert hat (und nicht Kunde ist)
  - Klare Grenzen: Jede natürliche oder juristische Person, die von uns Produkte gekauft hat, ist Kunde.  
Umgekehrt gilt dann, dass jede Person, die noch keine Produkte von uns gekauft hat, auch kein Kunde ist.  
  
Jede natürliche oder juristische Person, die unseren Katalog angefordert hat, ist Interessent.  
Umgekehrt gilt dann, dass jede Person, die noch keinen Katalog von uns angefordert hat, auch kein Interessent ist.
  - Unklare Grenzen: Jede Person, die an unseren Produkten interessiert ist, ist ein Interessent.  
Jede Person, die an unseren Produkten nicht interessiert ist, ist nicht Interessent.

## Kapitel 4

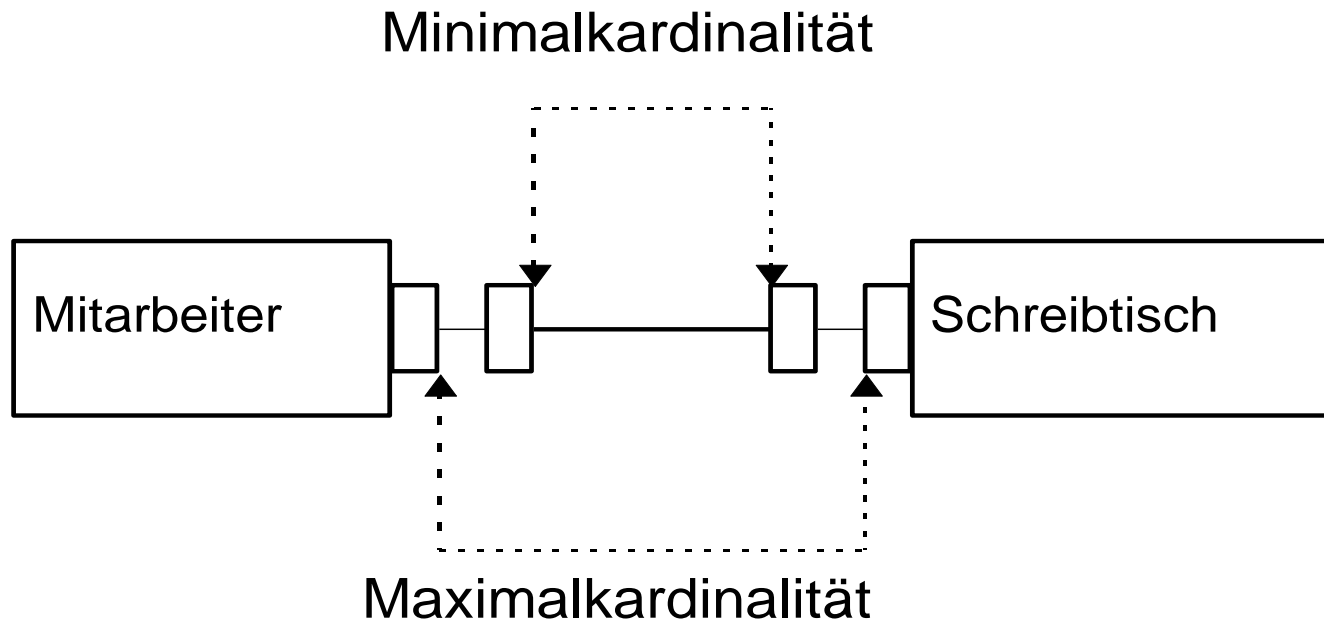
### Grundlagen ERM: Beziehungstypen



Beispiel:

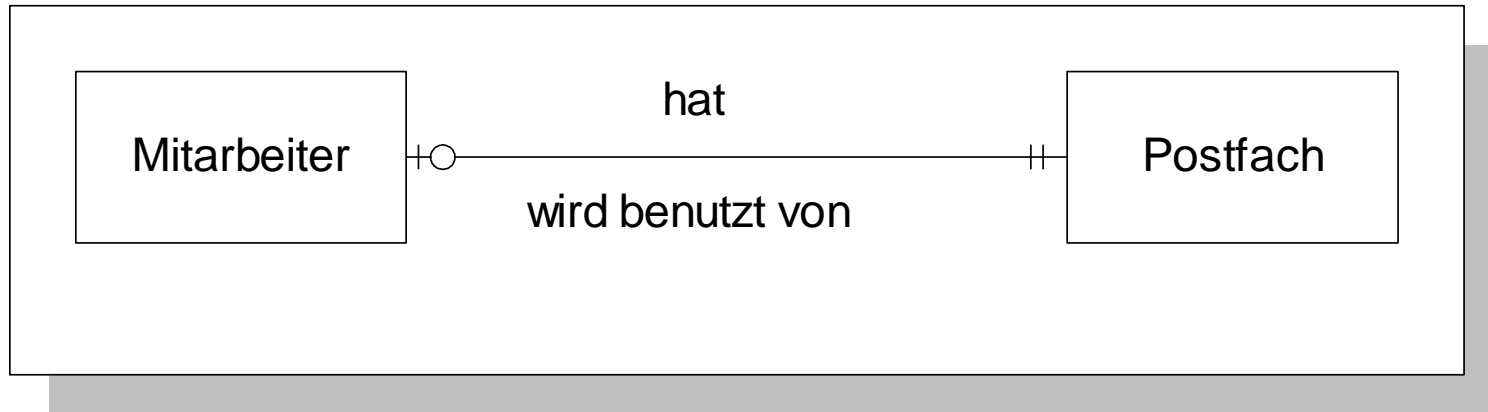


Mitarbeiter	Projekt
694	742
695	
696	743
697	744
...	...





Beziehungs-kardinalität	Symbol	Interpretation
Minimal-kardinalität	<b>0</b>	Der Minimalwert <b>0</b> (Null) bedeutet, dass die konkrete Beziehung zwischen zwei Entitäten <b>nicht existieren muss</b> , also optional ist.
	<b>1</b>	Der Minimalwert <b>1</b> (Eins) bedeutet, dass die konkrete Beziehung zwischen zwei Entitäten <b>existieren muss</b> .
Maximal-kardinalität	<b>1</b>	Der Maximalwert <b>1</b> (Eins) bedeutet, dass die konkrete Beziehung zwischen zwei Entitäten <b>höchstens einmal</b> existieren darf.
	<b>&lt;</b>	Der Maximalwert <b>&lt;</b> (Viele) bedeutet, dass die konkrete Beziehung zwischen zwei Entitäten <b>mehrmals</b> existieren darf.

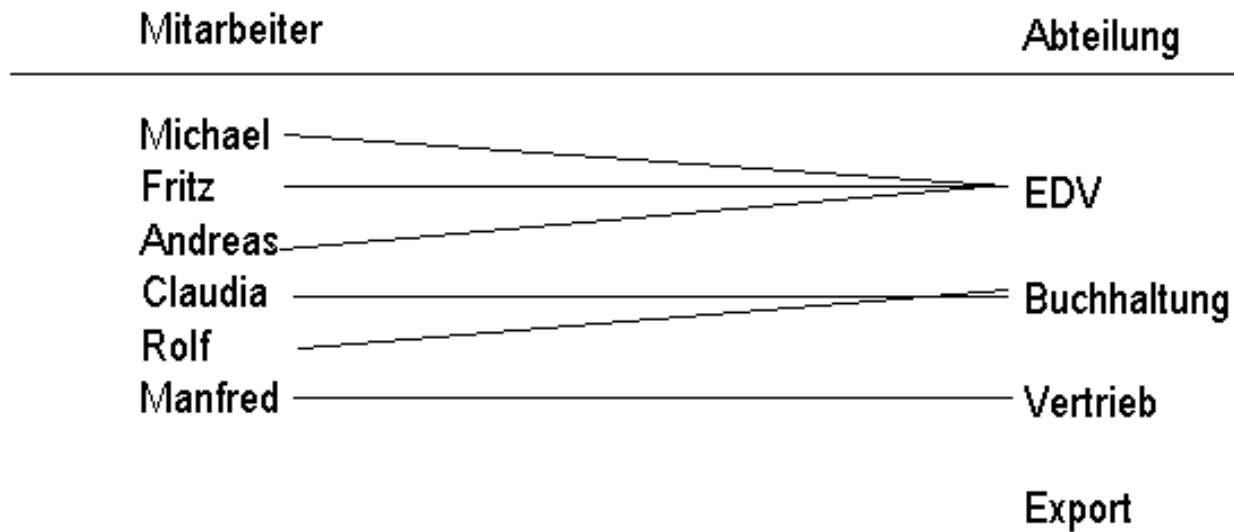


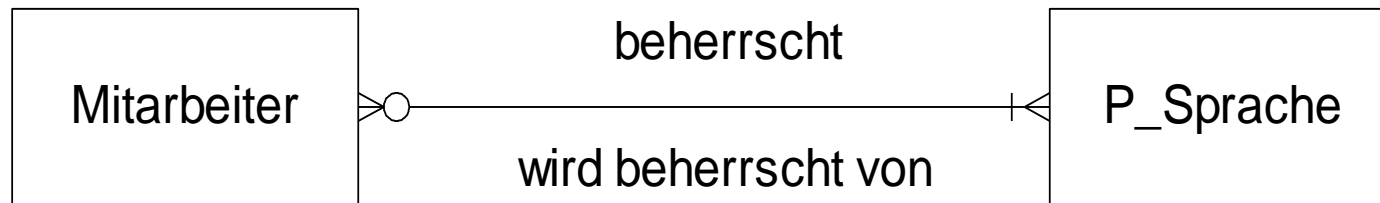
- Jeder Mitarbeiter hat minimal und maximal ein Postfach - genau ein Postfach.
- Jedes Postfach wird benutzt von minimal null und maximal einem Mitarbeiter.

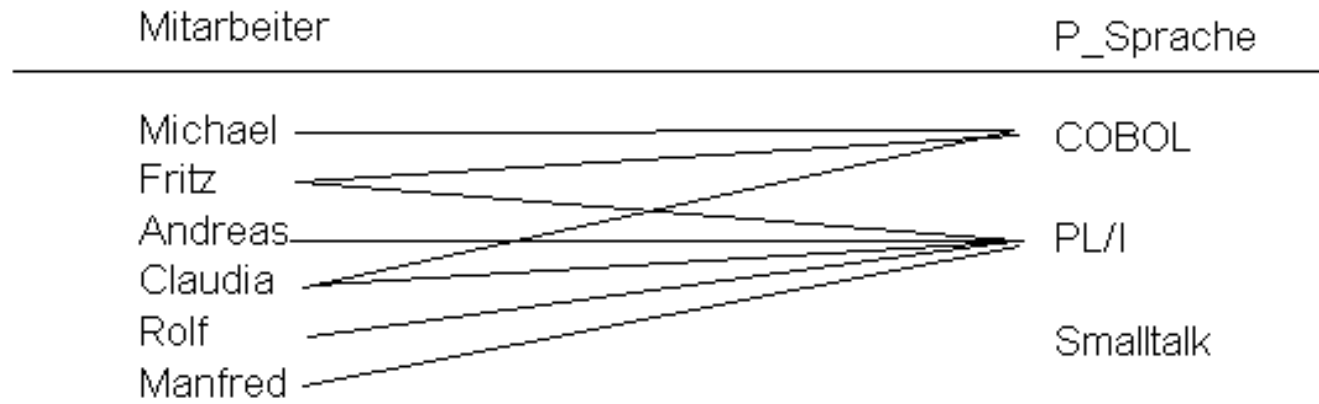
Mitarbeiter	Postfach
Michael	2701
Fritz	2702
Andreas	2703
	2704
Claudia	2705
Rolf	2706



# Die 1:N-Beziehung







## Kapitel 5

# Grundlagen ERM: Attributtypen



Beispiel:

## Entitätstyp Mitarbeiter

Attributtyp:

Entität 1:

Entität 2 :

Entität 3:

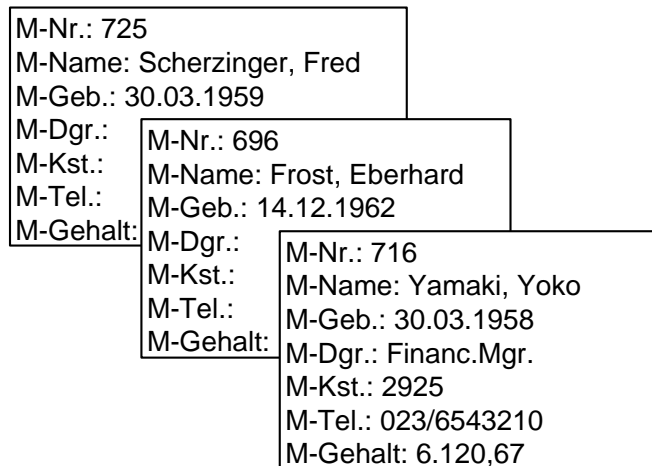
Entität 4:

Name	# Mitarbeiternr.	Gehalt
Frost	696	8.760
Bonello	737	4.800
Meier	314	11.520
Meier	784	6.590

Attributtypen sind

- die (abstrakten) Eigenschaften der Entitätstypen
- die kleinste Informationseinheit und unteilbar

Attribute sind konkrete Ausprägungen der Attributtypen



Der Name eines Attributtyps sollte seine Eigenschaft mit einem Hauptwort in Einzahl beschreiben

Beispiele:

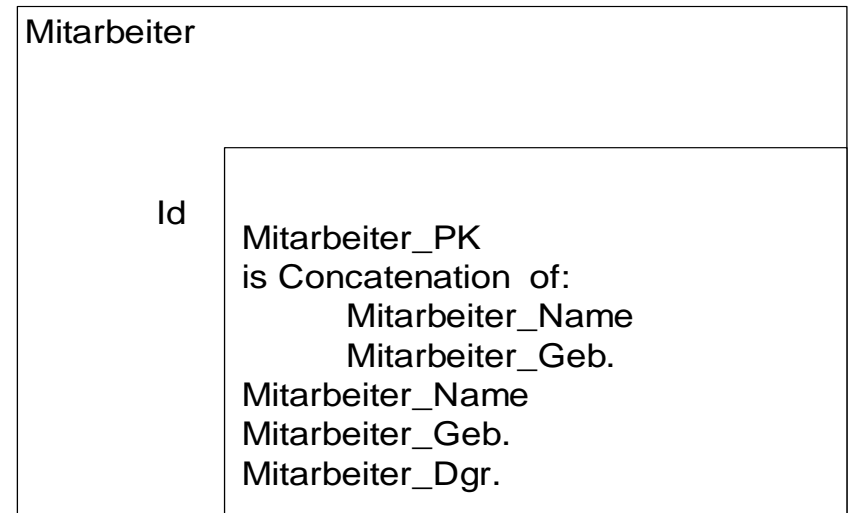
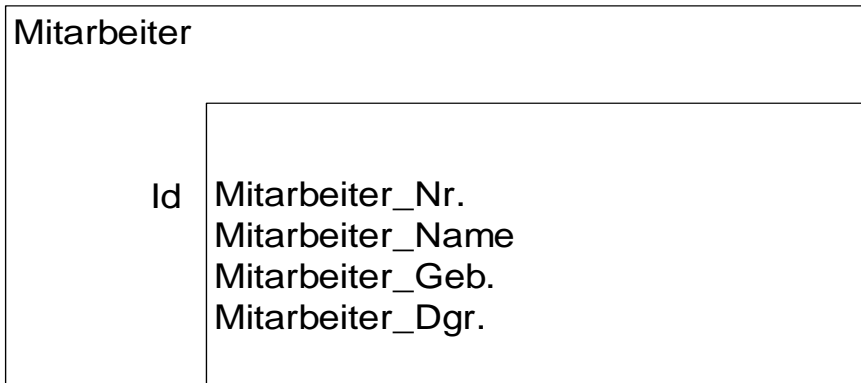
- Personal\_Nr
- Kunde\_Nummer
- Mitarbeiter\_Name

Die Bedeutung des Attributtyps muss präzise und verständlich definiert sein

Beispiele:

- Kunde\_Vorname = Rufname (Vorname) und evtl. weitere Vornamen, evtl. abgekürzt
- Kunde\_Name = Nachname, ohne Titel und Akademische Grade

- Kandidaten für den Entitätsschlüssel müssen
  - eindeutig sein
  - für jede Entität jederzeit bekannt sein und
  - stabil sein

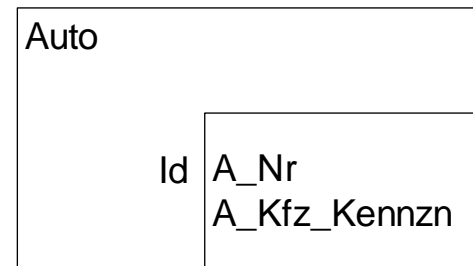
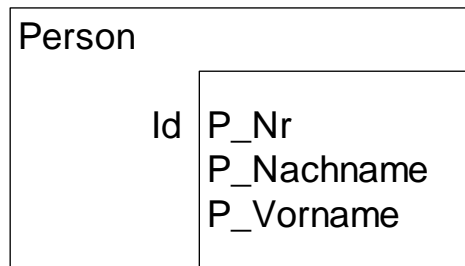


# Zuordnung von Attributtypen: Funktionale Abhängigkeit

Attribut = Funktion (Entitätstyp, Attributtyp, Schlüsselwert)

- Beispiel:

- Gehört die Autokennzeichen-Nummer zur Person, der das Auto gehört?  
Oder sollte man einen neuen Entitätstypen Auto schaffen, der zu Person eine Beziehung hat?

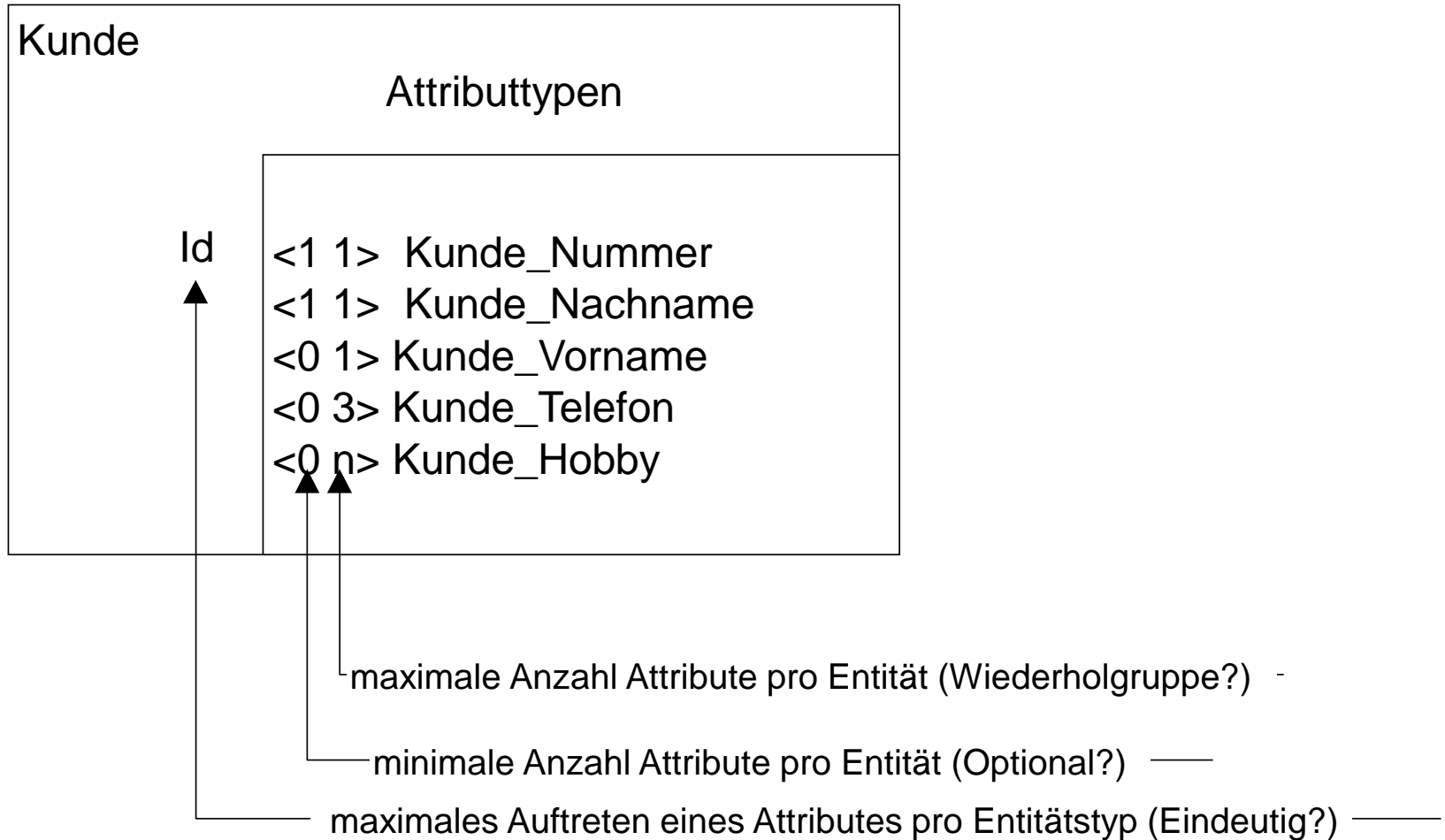


Produkt	
Id	Prod_Nr Prod_Name Prod_Preis Prod_Preis_incl_MwSt

Produkt (Implementierungsmodell)	
Id	Prod_Nr Prod_Name Prod_Preis Prod_Jahresumsatz

- Minimal: Wie viel verschiedene Ausprägungen dieses Attributtyps muss eine Entität mindestens haben?
- Maximal: Wie viel verschiedene Ausprägungen dieses Attributtyps kann eine Entität höchstens haben? Kann es nur ein oder mehrere Attribute pro Entität geben?
- Eindeutig: Kann ein Attribut (konkreter Wert) nur einmal im Entitätstyp auftauchen, dann eignet es sich als Schlüsselkandidat zur Identifikation?





Kunde		
	Attributtypen	Format
Id	<1 1> Kunde_Nummer	N(9)
	<1 1> Kunde_Name	X(30)
	<0 1> Kunde_Kreditlimit	N(6,2)
	<0 1> Kunde_Gesch_Umsatz	N(6,2)
	<0 1> Kunde_Telefon	X(16)

Attributtyp	Globaler Datentyp
Kunde_Nummer	ID_NR
Kunde_Name	NAME_LANG
Kunde_Kreditlimit	BETRAG
Kunde_Telefon	TEL_NR
Lieferant_Name	NAME_LANG
Mitarbeiter_Gehalt	BETRAG
Mitarbeiter_Telefon	TEL_NR
Produkt_Preis	BETRAG
Auftrag_Datum	DATUM

- Dadurch erhalten wir:

Kunde		
	Attributtypen	Datentyp
Id	<1 1> Kunde_Nummer	ID_NR
	<1 1> Kunde_Name	NAME_LANG
	<0 1> Kunde_Kreditlimit	BETRAG
	<0 1> Kunde_Gesch_Umsatz	BETRAG
	<0 1> Kunde_Telefon	TEL_NR

- Der Datentyp wird dann an einer Stelle zentral definiert:

Datentyp	SQL-Format	COBOL-Format
ID_NR	LONGINT	S9(9) COMP
NAME_LANG	CHAR(50)	X(50)
BETRAG	DECIMAL 8.2	9(6)V99
TEL_NR	CHAR(16)	X(16)

Wir kennen drei verschiedene Arten von Wertebereichen:

1. Einen endlichen Wertebereich, durch Aufzählung definierbar, möglichst klein.

Beispiel: Wochentage (Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag)

Beispiel: Postleitzahlen (... , 72070, 72071, 72072,...)

2. Einen durch Unter- und Obergrenze als Bereich definierten.

Beispiel: Gewicht eines Päckchens (stufenlos von 10 g bis 4000 g)

3. Einen praktisch unendlichen oder doch zumindest sehr großen Wertebereich, eventuell durch bestimmte Algorithmen definiert.

Beispiel: Name (alle möglichen Namen)

Beispiel: Datum (alle möglichen Tagesdaten)

# Wie definiert man einen Entitätstyp präzise?

Was ist ein PRODUKT? Ist es

- Ein besonderer Gegenstand, den ein KUNDE mit nach Hause nehmen kann? oder
- Irgend ein Gegenstand, den wir verkaufen? und
- Kann mehr als ein Kunde ein bestimmtes PRODUKT kaufen?
- Meinen wir nicht PRODUKT-TYP? und
- Ist alles, was wir an KUNDEN verkaufen, ein PRODUKT –
- also auch ein a

## PRODUKT

Güter und Dienstleistungen, die unser Unternehmen planmäßig und regelmäßig an seine Kunden verkauft

Id	<1 1>	PROD_NUMMER	lfd.Nummer
	<1 1>	PROD_NAME	der Name
	<0 1>	PROD_BESCHR	Beschreibung
	<0 1>	PROD_PREIS	Preis in €

Mengengerüst:  
2007: 64000  
MAX: 150000

## Kapitel 6

# Fortgeschrittenes Modellieren – spezielle Entitätstypen



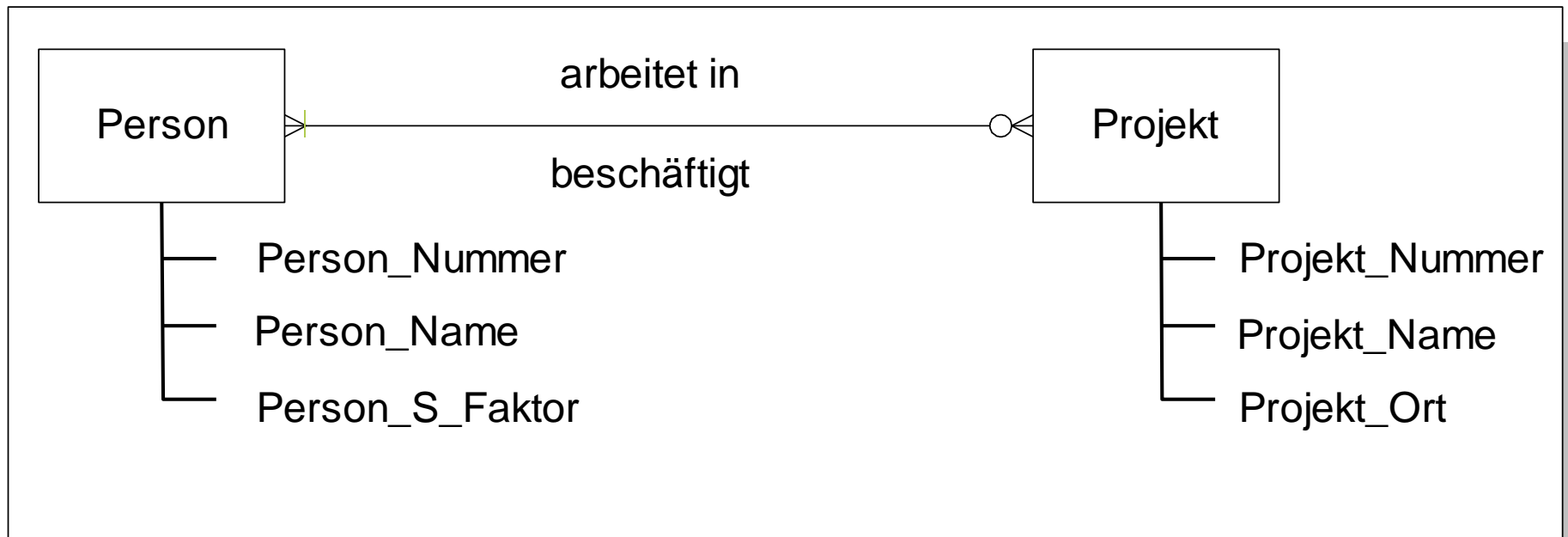
Um komplexe Zusammenhänge mit ERM korrekt darstellen zu können, bedarf es bestimmter Konstruktionen:

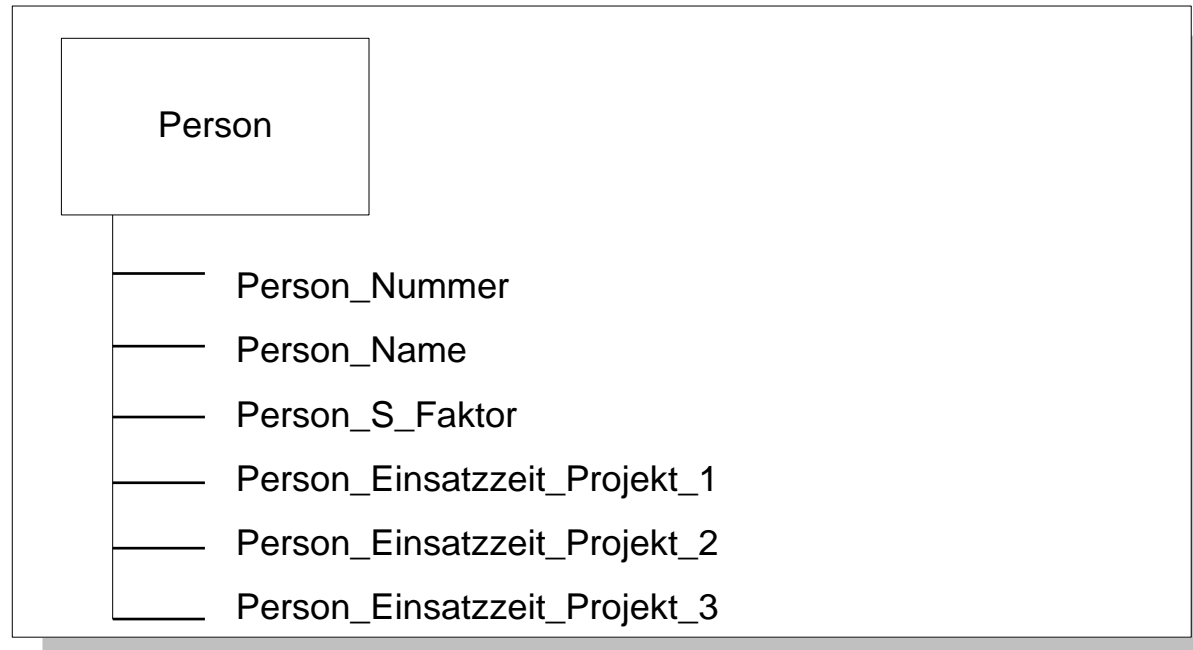
1. Beziehungsattribute durch assoziative Entitätstypen
2. Subtyp / Supertyp
3. Hierarchien und rekursive Beziehungen
4. Status / Historie
5. in das Datenmodell integrierte Wertebereiche.

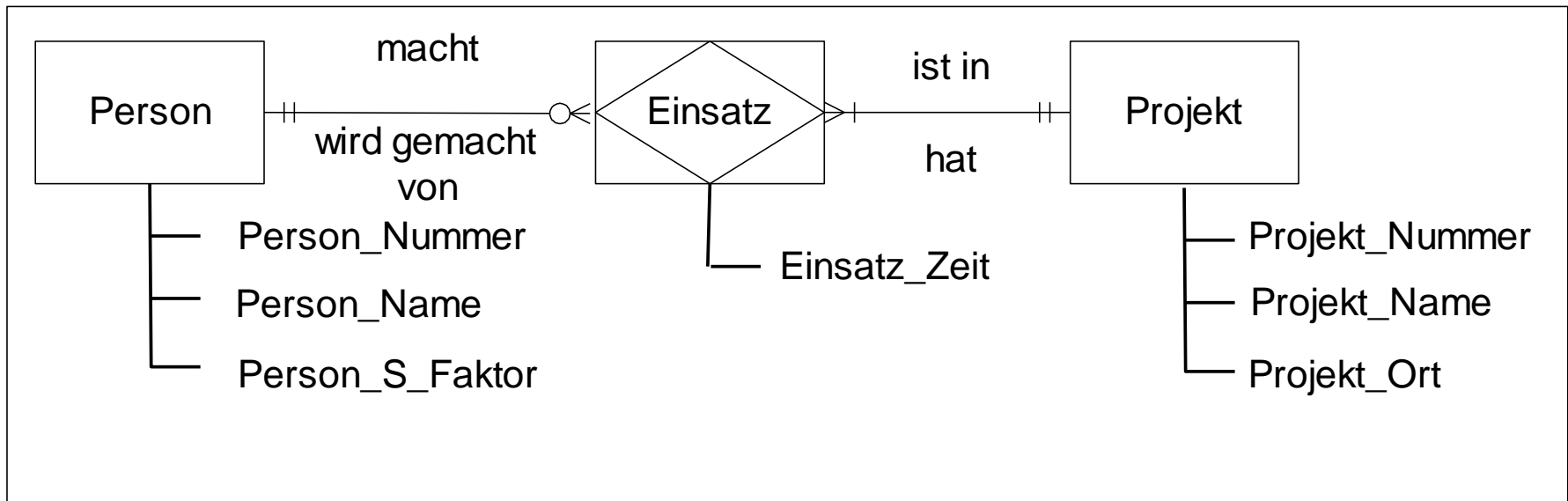
## Beispiel:

Für ein Abrechnungssystem, das den Einsatz von Personen in Projekten nach Stunden abrechnet, brauchen wir folgende Datenelemente:

- Person\_Name
- Person\_Nummer
- Person\_S\_Faktor
- Projekt\_Name
- Projekt\_Nummer
- Projekt\_Ort
- Projekt\_Einsatzzeit.







# Entitätsschlüssel bei Assoziativen Entitätstypen

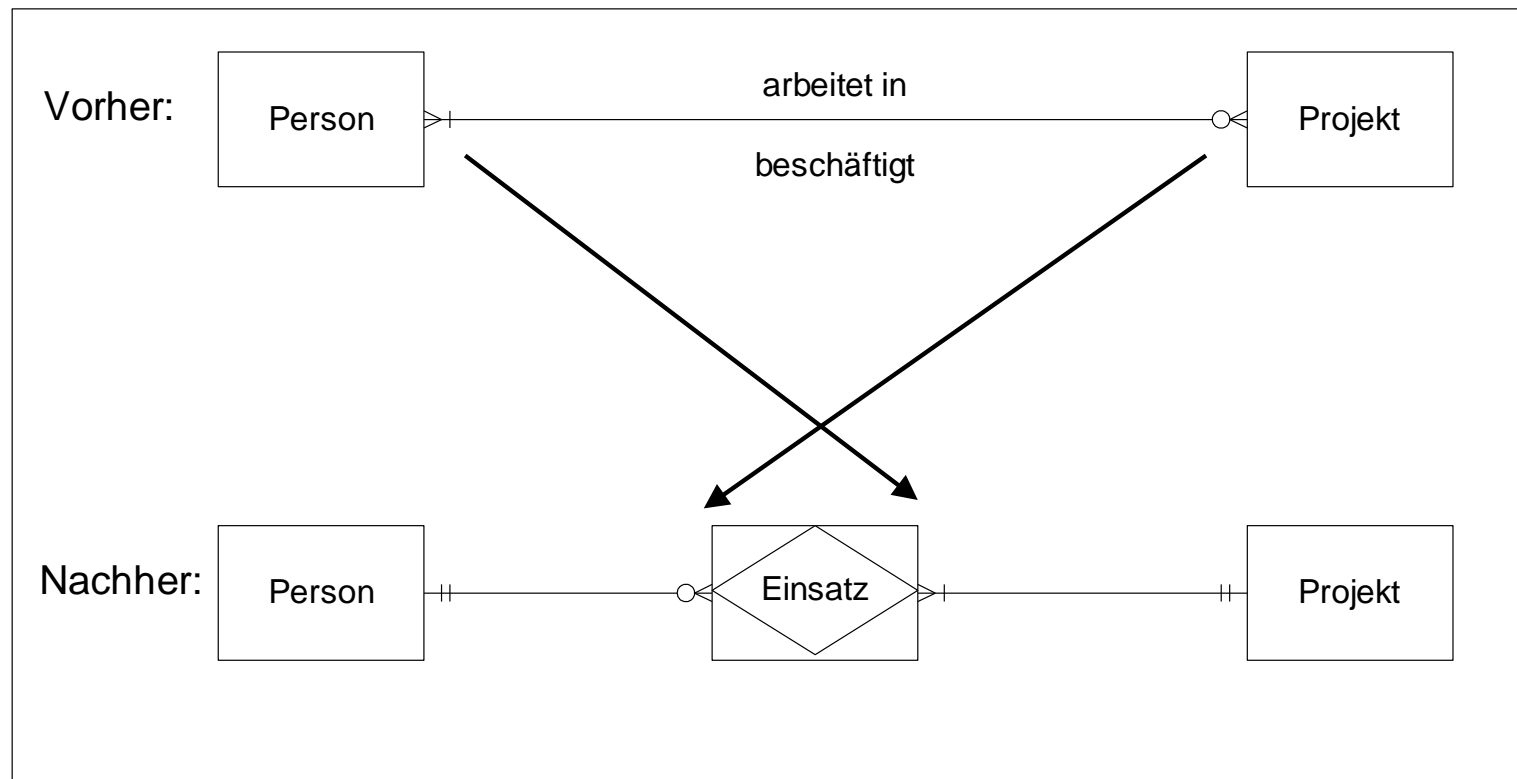
## Einsatz

Id    <1 1>    Einsatz\_PK  
         is Concatenation of:  
              Einsatz.ist in.Projekt  
              Einsatz.wird gemacht von.Person

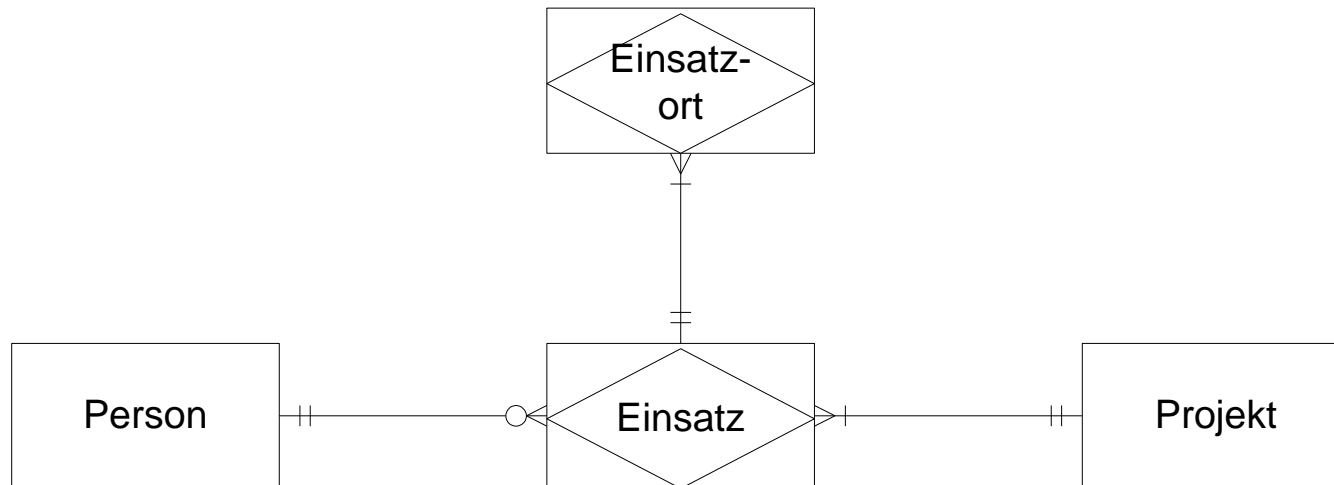
<1 1>    Einsatz\_Zeit

Einsatz.1(ist in)1.Projekt  
      Projekt.1(hat)N.Einsatz  
Einsatz.1(wird gemacht von)1.Person  
      Person.0(macht)N.Einsatz

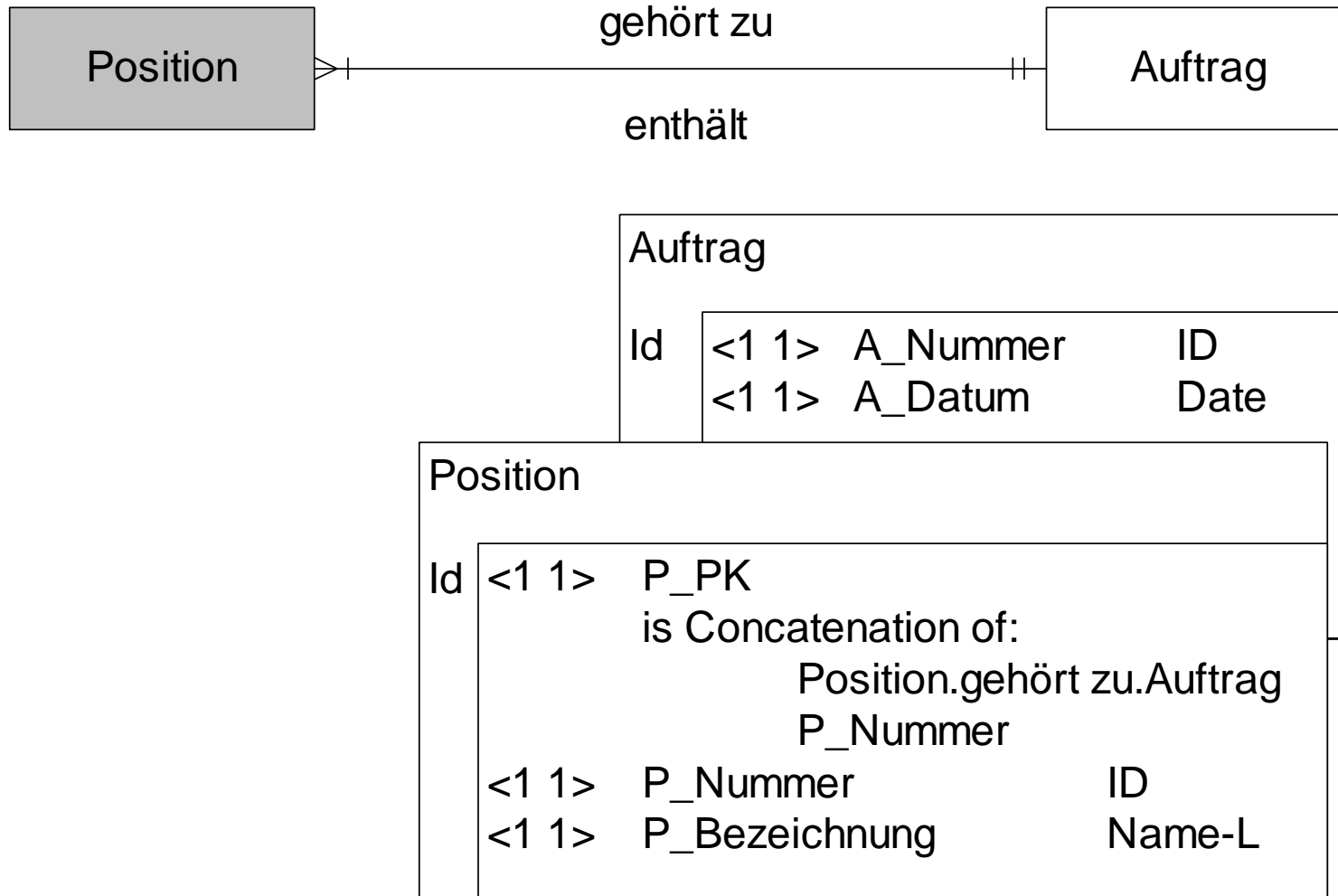
# Auflösen von N:M -Beziehungen und Kardinalitäten

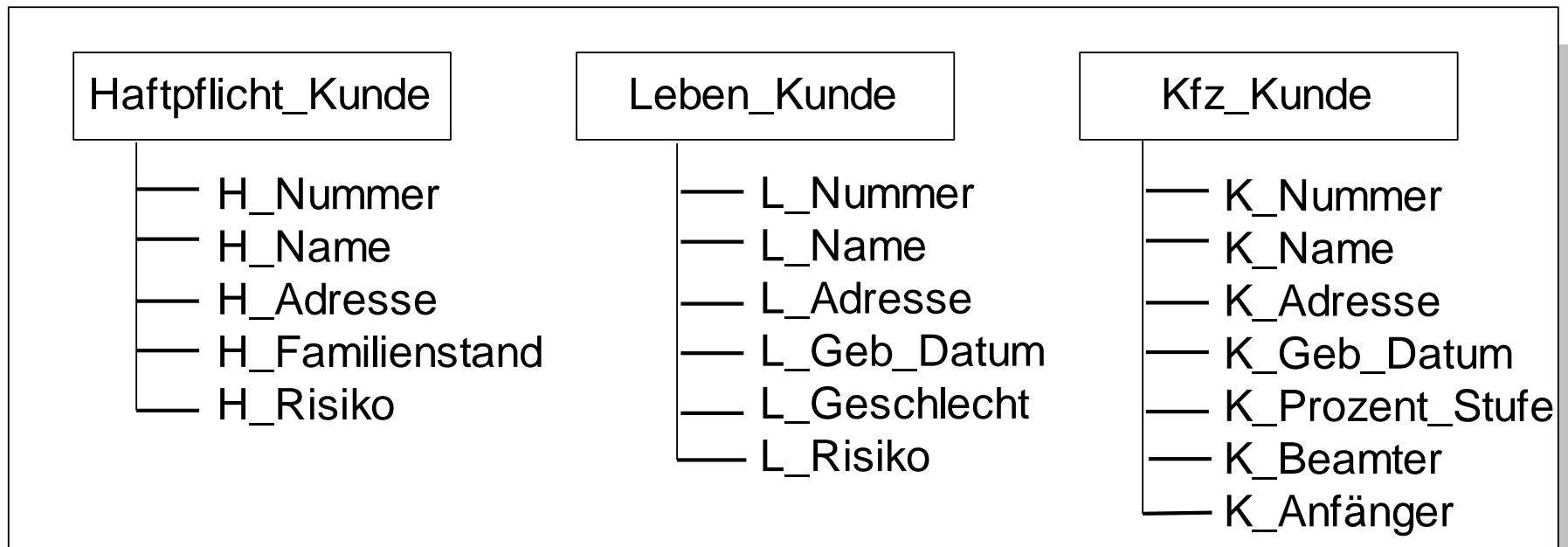


# Müssen N:M -Beziehungen grundsätzlich aufgelöst werden?



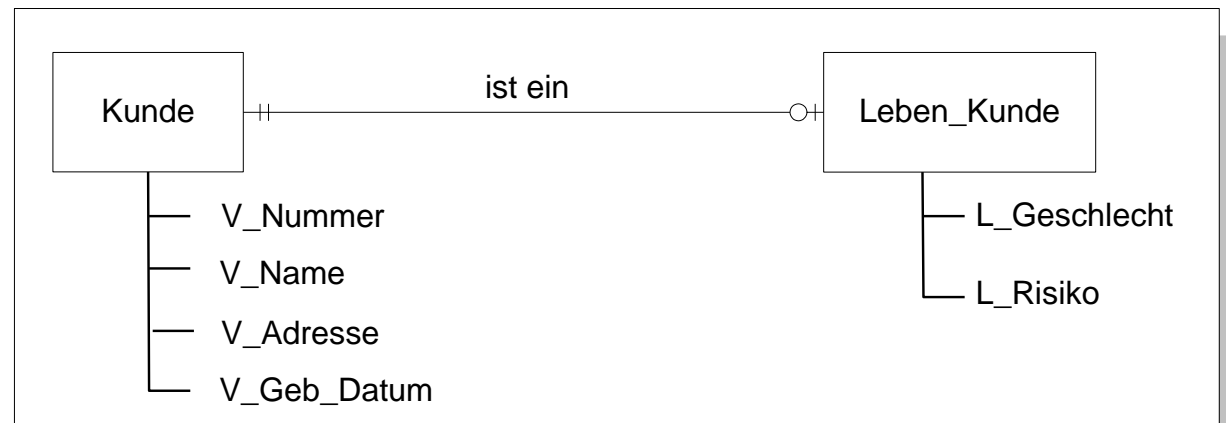
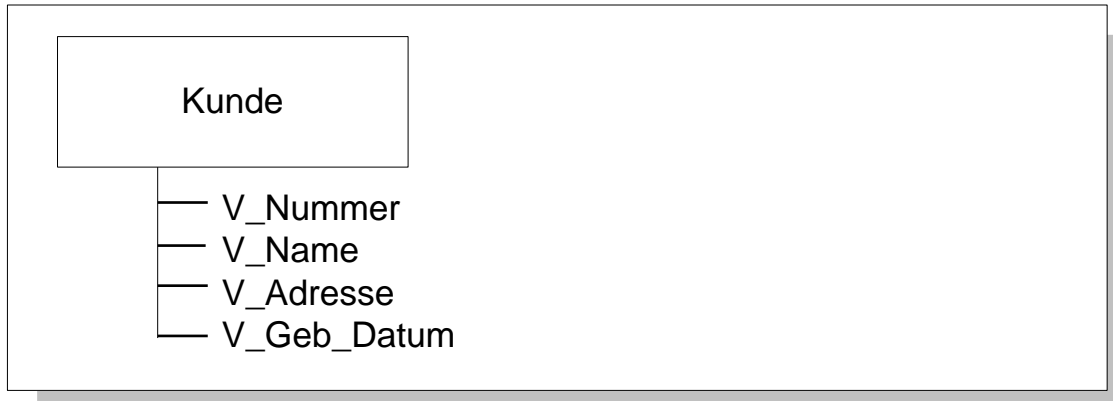


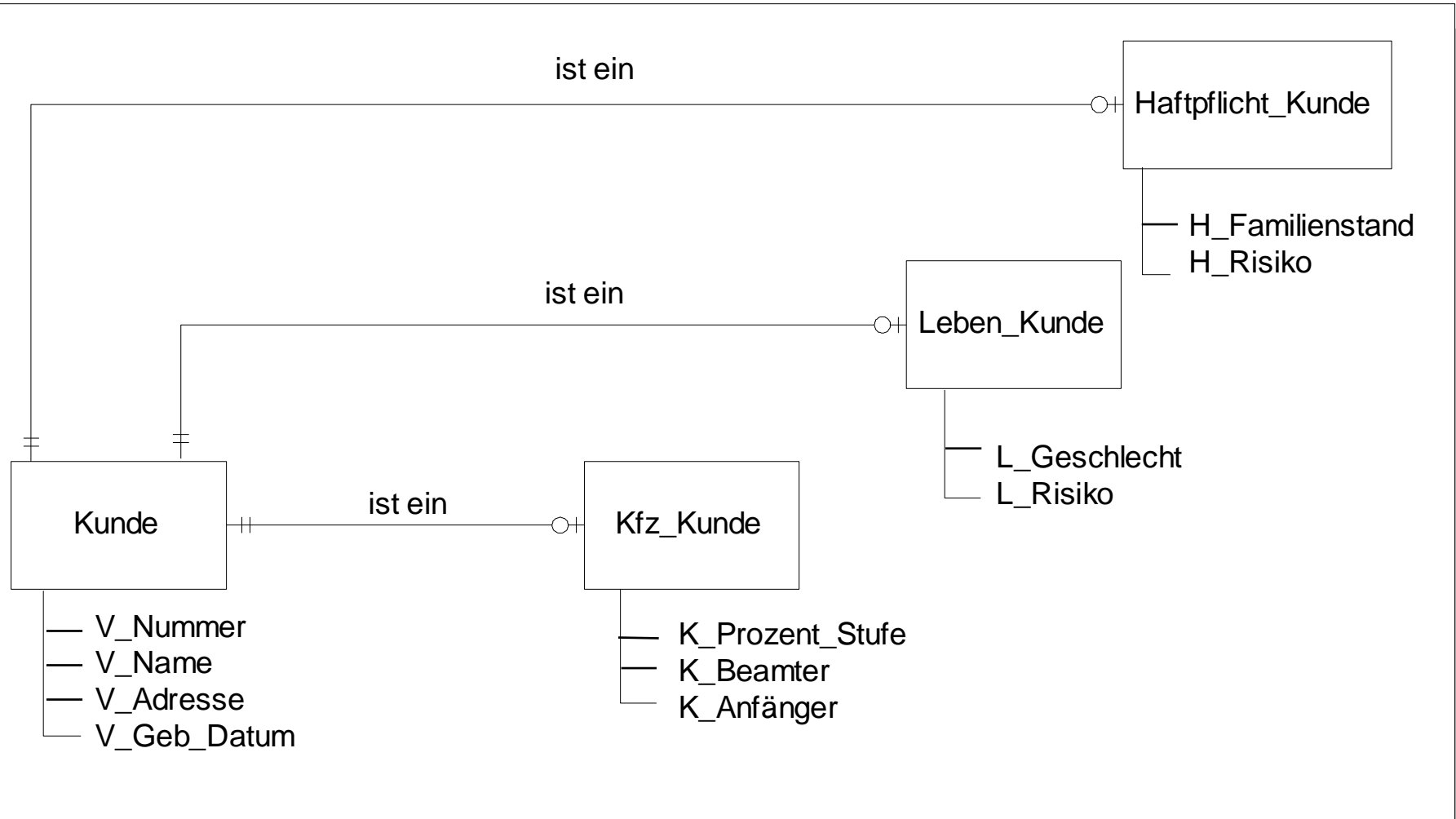




Kunde

- V\_Nummer
- V\_Name
- V\_Adresse
- V\_Geb\_Datum
- H\_Familienstand
- H\_Risiko
- L\_Geschlecht
- L\_Risiko
- K\_Prozent\_Stufe
- K\_Beamter
- K\_Anfänger





Leben\_Kunde

Id

Eigene Attributtypen:

<1 1> L\_Geschlecht

<1 1> L\_Risiko

Geerbte Attributtypen:

<1 1> V\_Nummer

<1 1> V\_Name

<1 1> V\_Adresse

<1 1> V\_Geb\_Datum

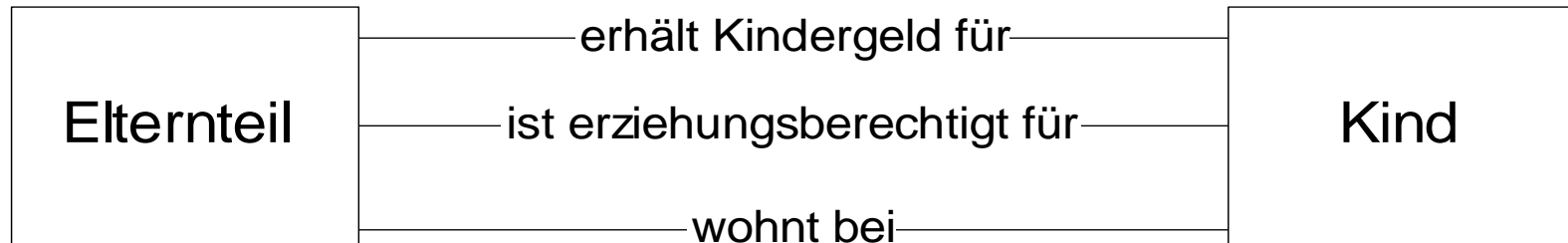
Eigene Beziehungen:

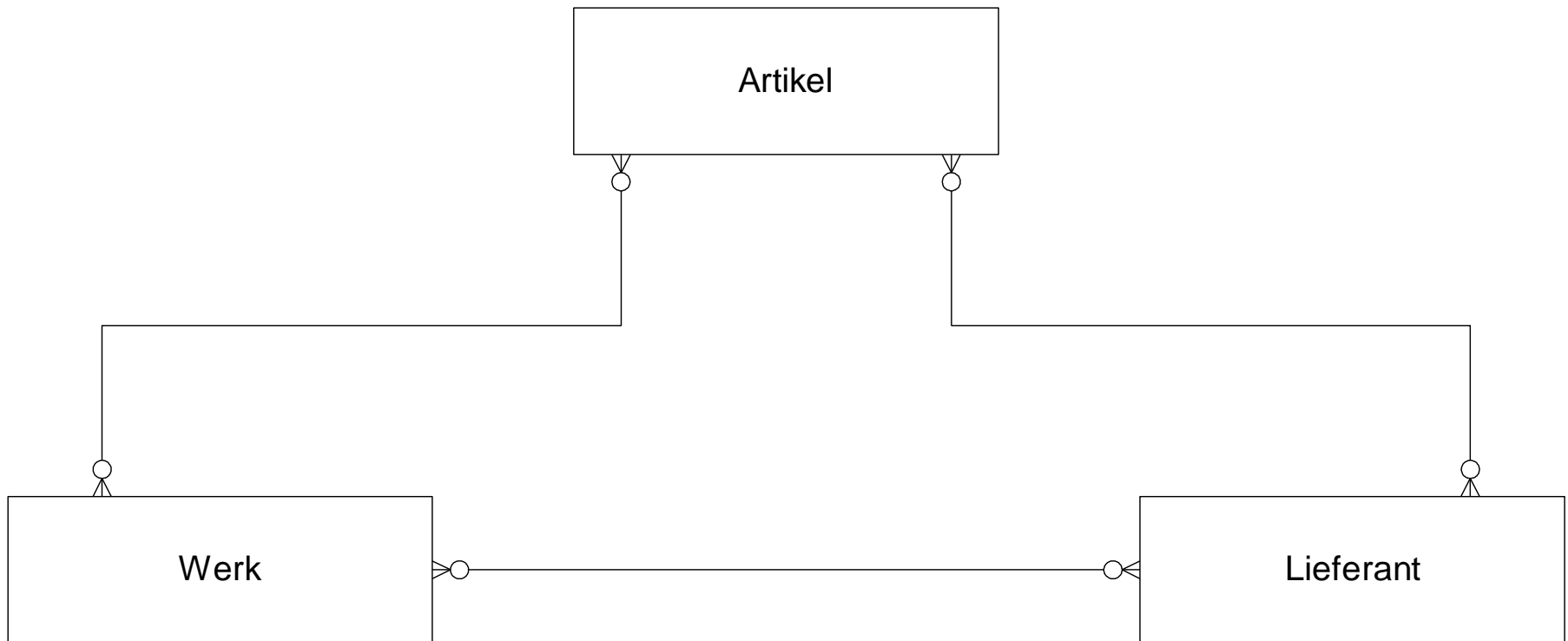
=== keine eigenen Beziehungen definiert ===

Geerbte Beziehungen:

=== keine geerbten Beziehungen definiert ===

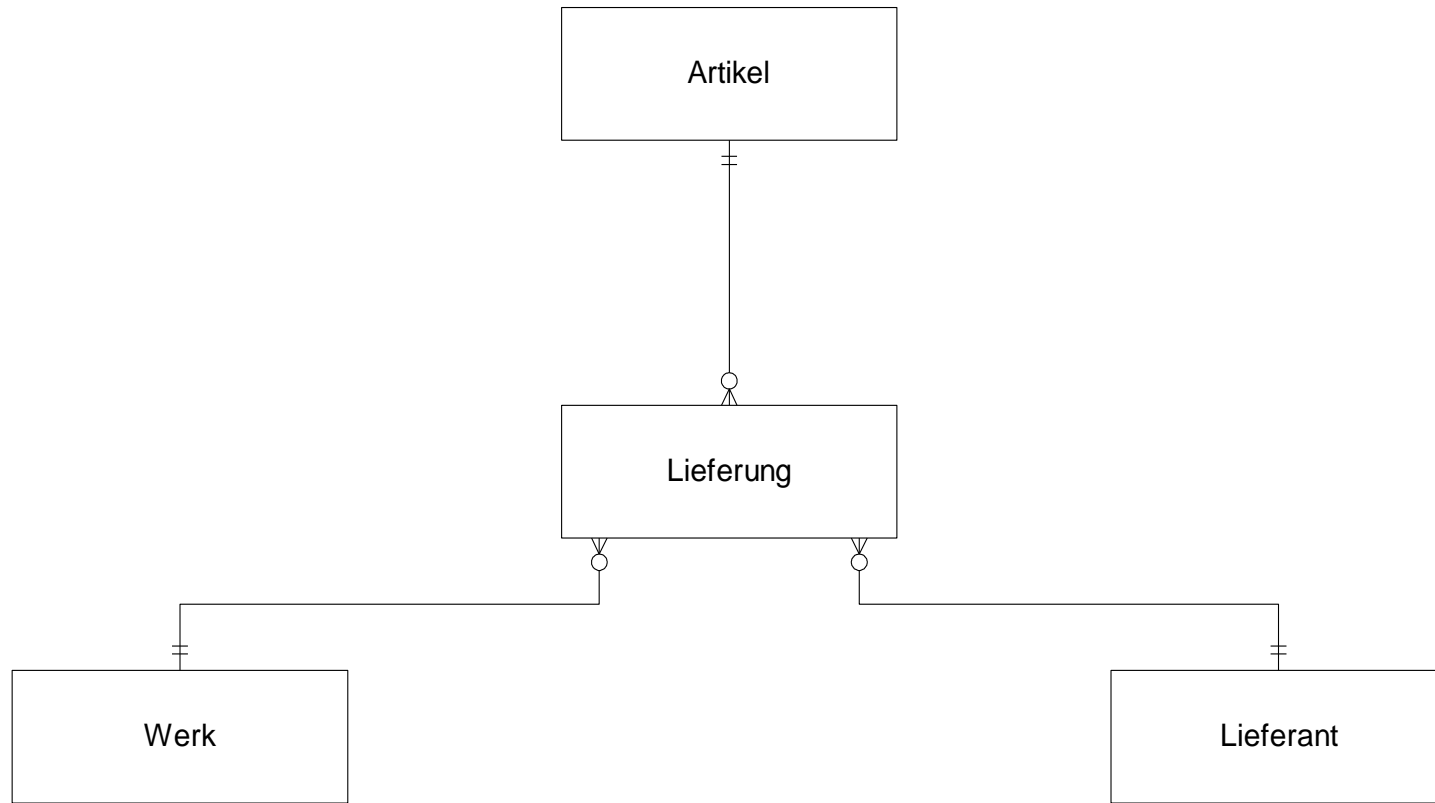
- Mehrere Beziehungstypen





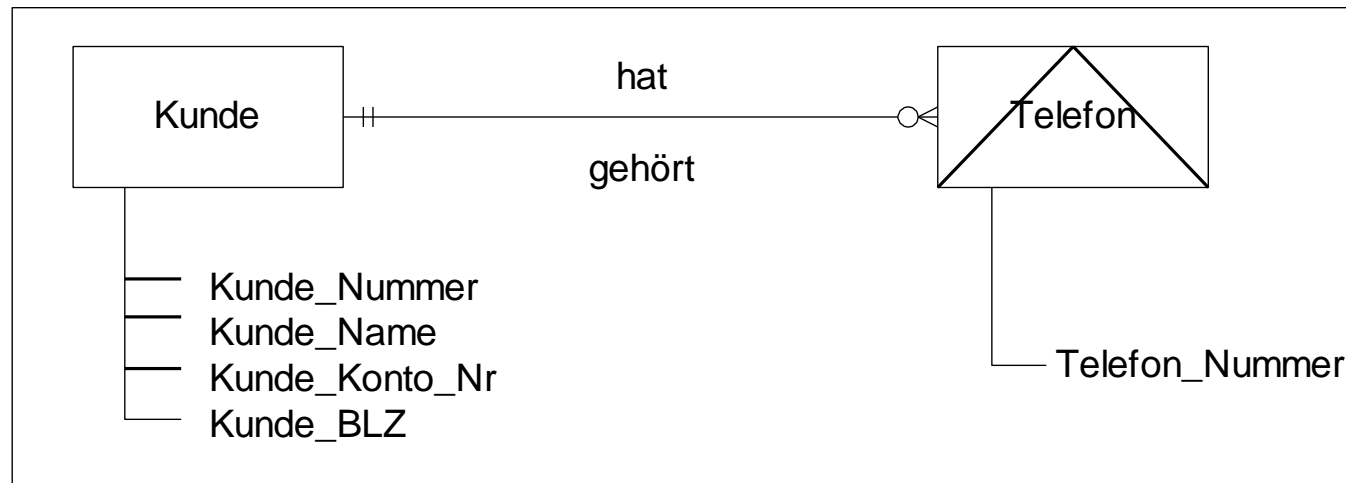
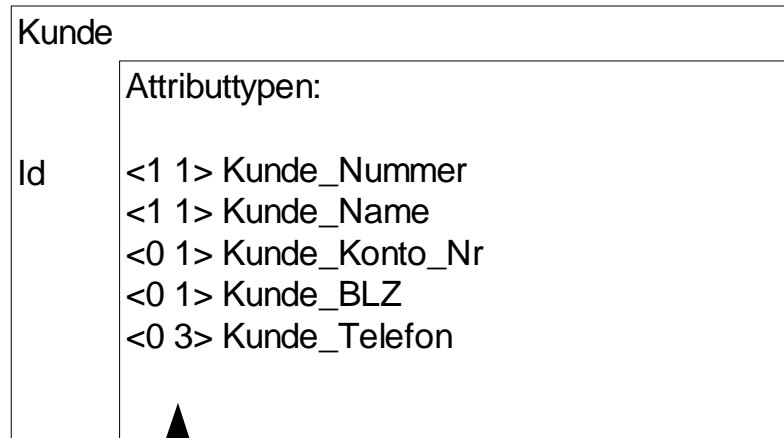
Darstellung eines N-stelligen Beziehungstyps (vorher)

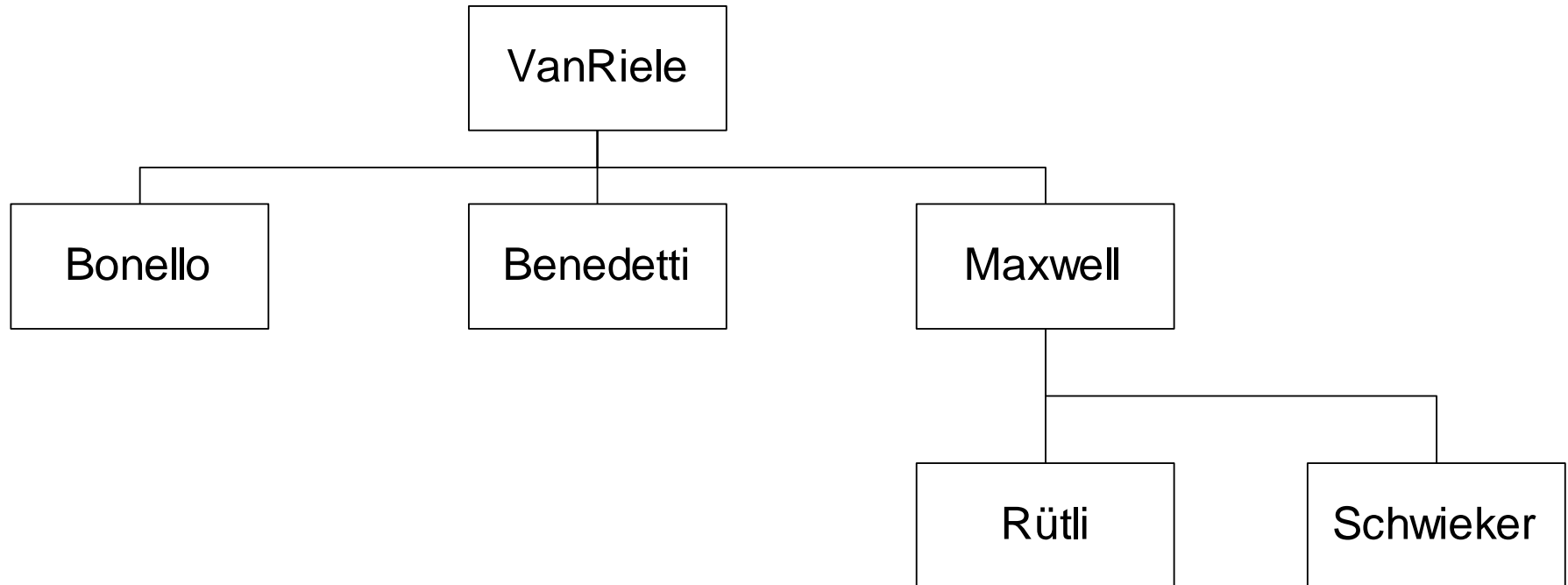




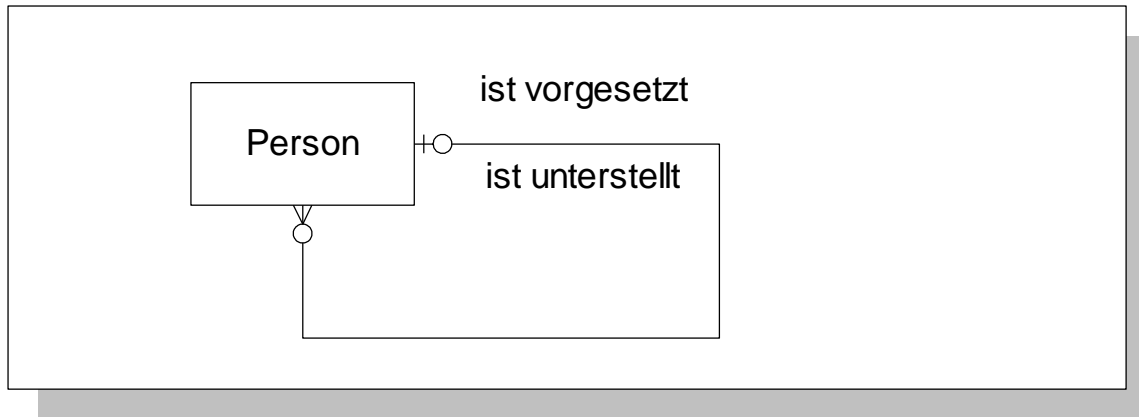
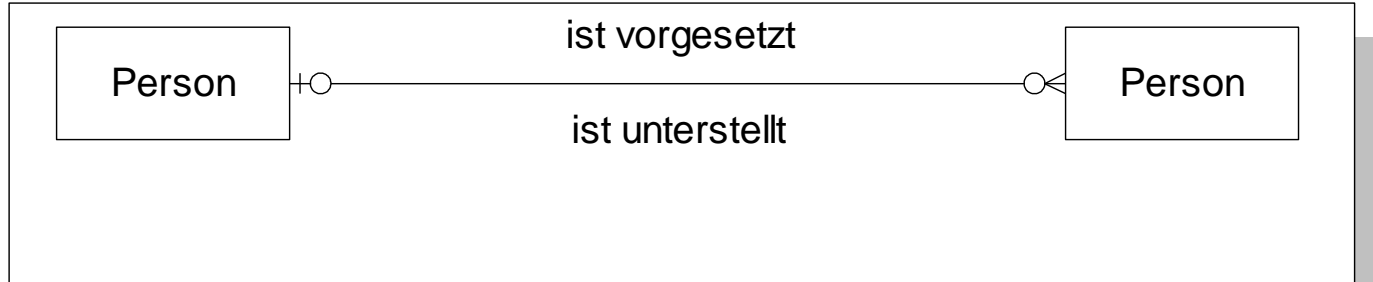
Darstellung eines N-stelligen Beziehungstyps  
(nachher)

# Entitätstypen aus mehrwertigen Attributtypen

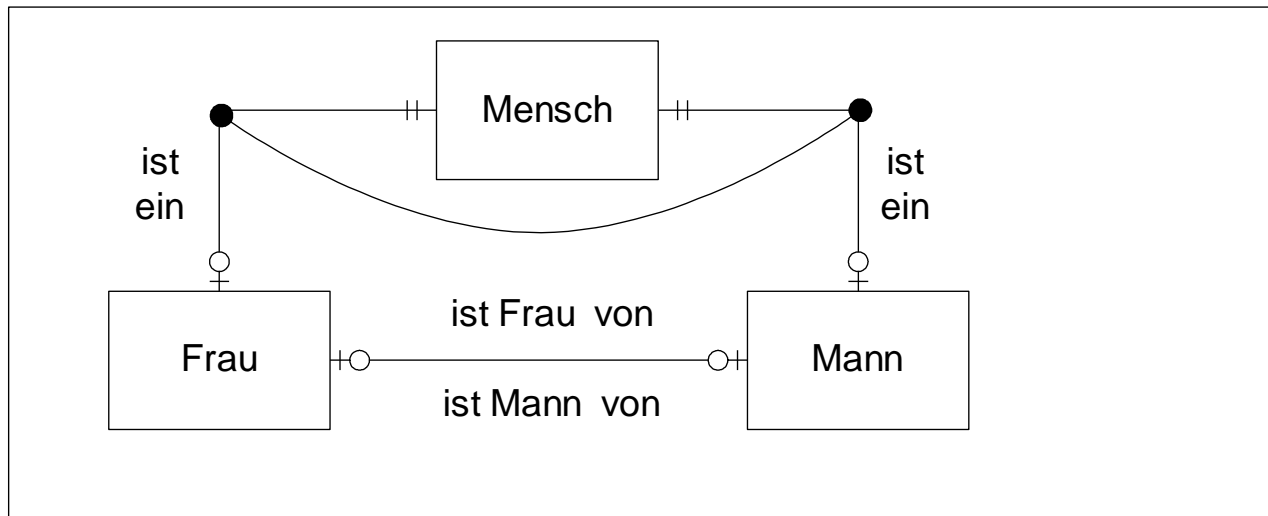
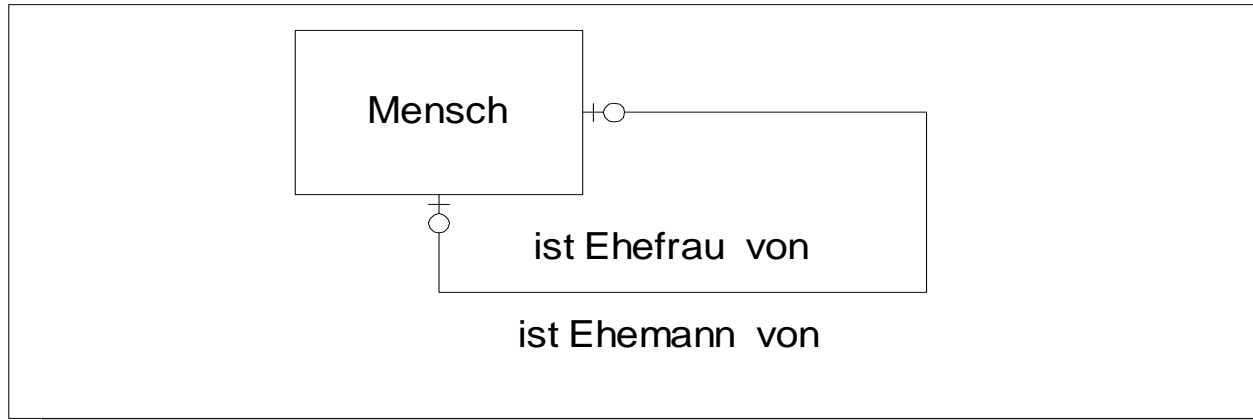




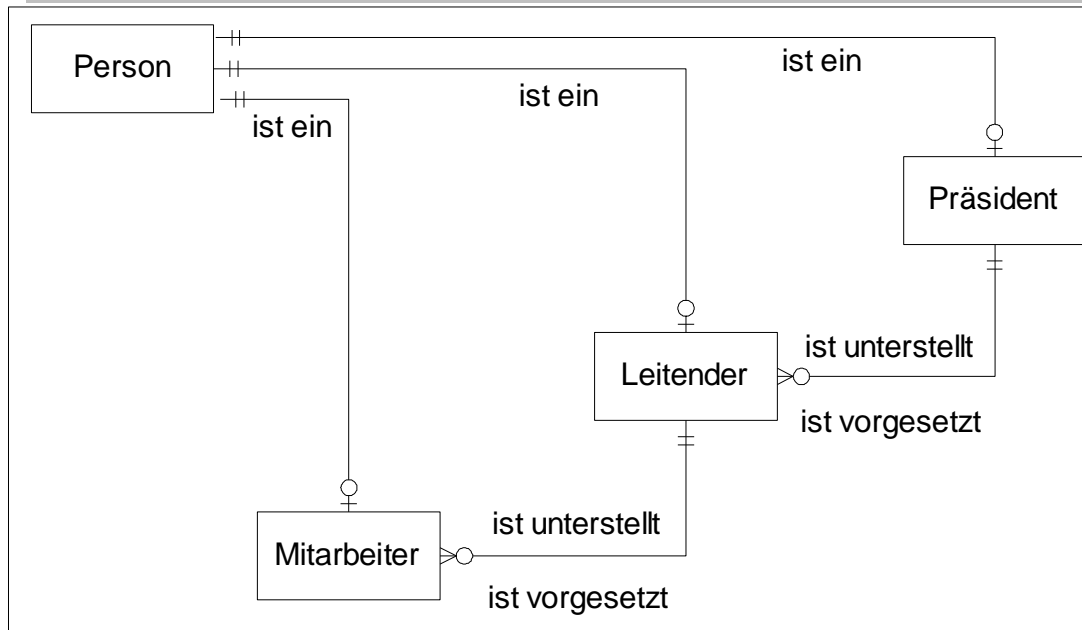
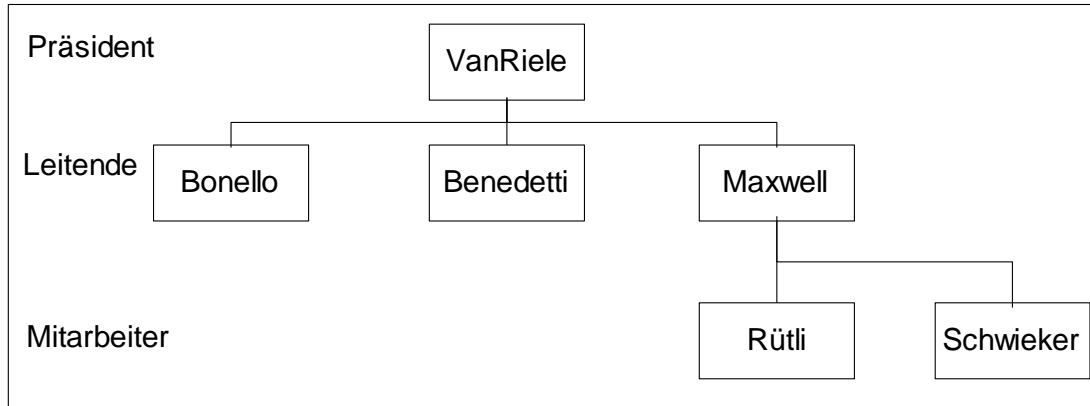
# Das ließe sich so darstellen:



# Einige Rekursionsprobleme lassen sich durch Subtypen elegant lösen



# Einige Rekursionsprobleme lassen sich durch Subtypen elegant lösen

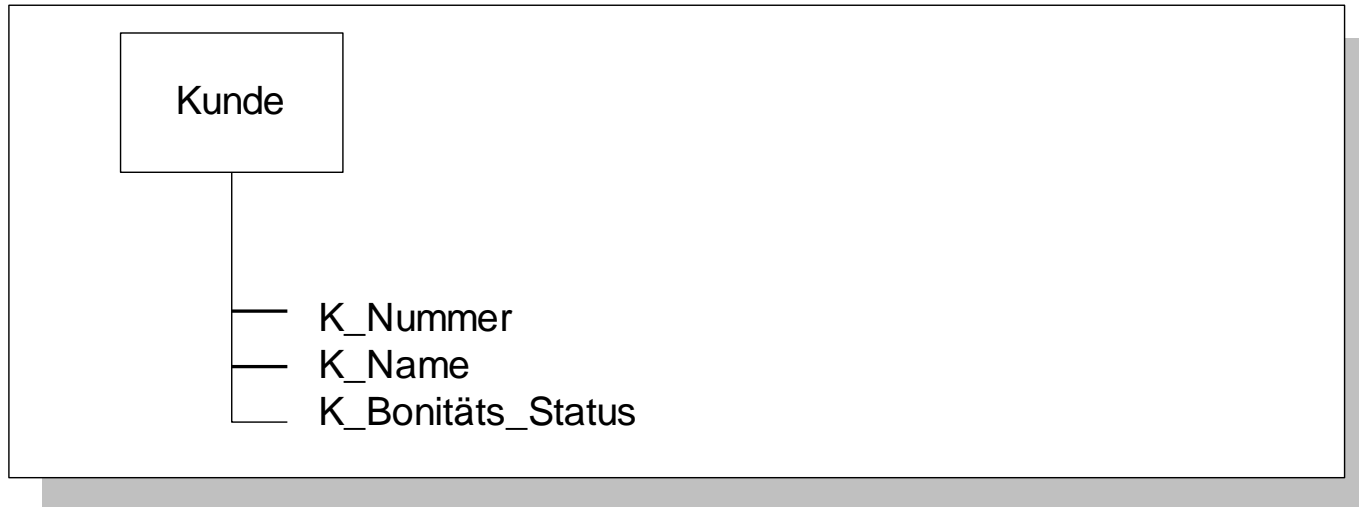


Mit Status bezeichnen wir eine Eigenschaft einer Entität, nämlich die Zugehörigkeit zu einer Gruppe oder Klasse.

Beispiele:

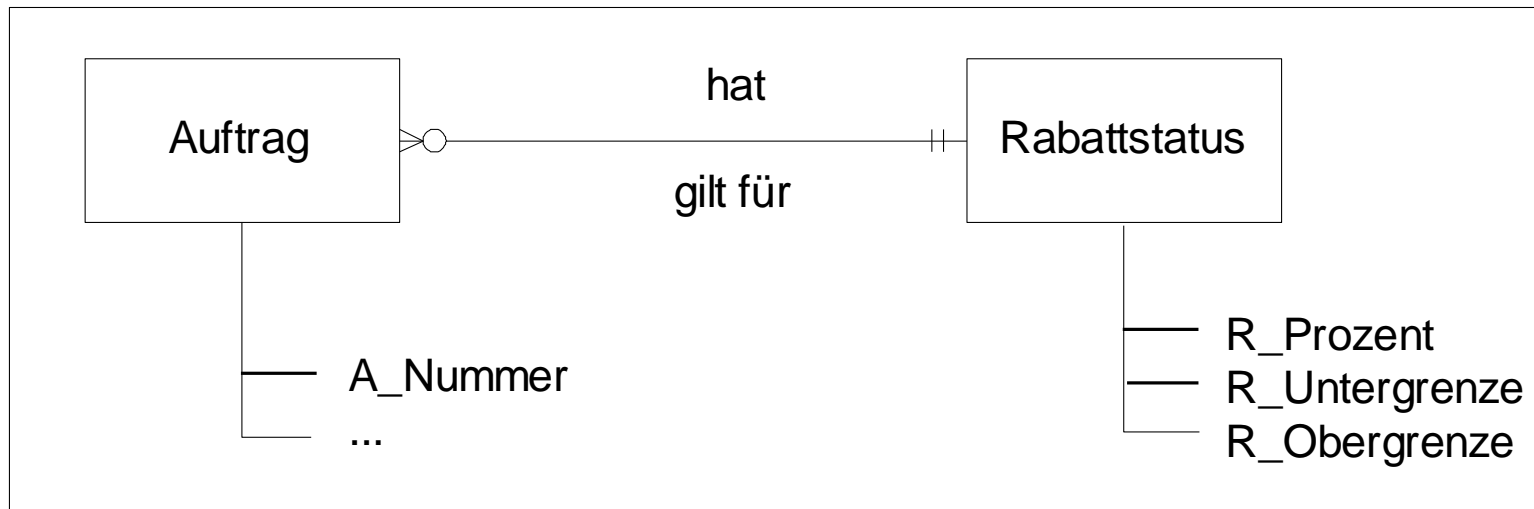
- Ein Kunde hat einen Bonitäts-Status (gut, mittel, schlecht).
- Ein Vertrag hat einen bestimmten Mahnstatus (neutral, gemahnt, 2. Mahnung, gekündigt).
- Eine Lieferung befindet sich in einem bestimmten Lieferstatus.
- Ein Kunde hat einen Rabattstatus.

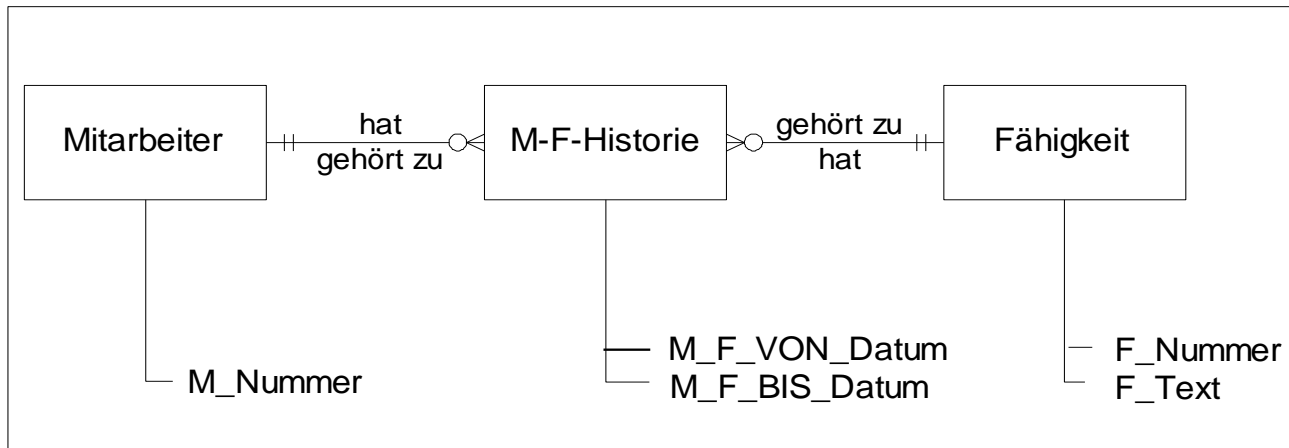
- Er kann ein Attributtyp des Entitätstyps sein, weil es sich um eine Eigenschaft der Entität handelt.





- Man kann diesen Status auch als Zugehörigkeit einer Entität zu einer Klasse, zum Beispiel des Kunden zu einer Rabattklasse oder die eines Vertrages zu einer Mahnstufe, betrachten. Er kann durch eine Beziehung des Entitätstyps zur Statusklasse und im konkreten Fall einer bestimmten Entität zur konkreten Status-Entität beschrieben werden.





## M-F-Historie

Id	<p>&lt;1 1&gt; M_F_Hist_PK is Concatenation of:                          Historie. gehört zu. Mitarbeiter                          Historie. gehört zu .Fähigkeit                          M_F_VON_Datum</p> <p>&lt;1 1&gt; M_F_VON_Datum                          &lt;0 1&gt; M_F_BIS_Datum</p>
	<p>M-F-Historie.gehört zu. Mitarbeiter                          Mitarbeiter. hat. M-F-Historie                      M-F-Historie.gehört zu. Fähigkeit                          Fähigkeit. hat. M-F-Historie</p>

Wir kennen drei verschiedene Arten von Wertebereichen

1. Einen endlichen Wertebereich, durch Aufzählung definierbar, möglichst klein.

Beispiel: Wochentage

(Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag)

Beispiel: Postleitzahlen (..., 72070, 72071, 72072,...)

2. Einen durch Unter- und Obergrenze als Bereich definierten.

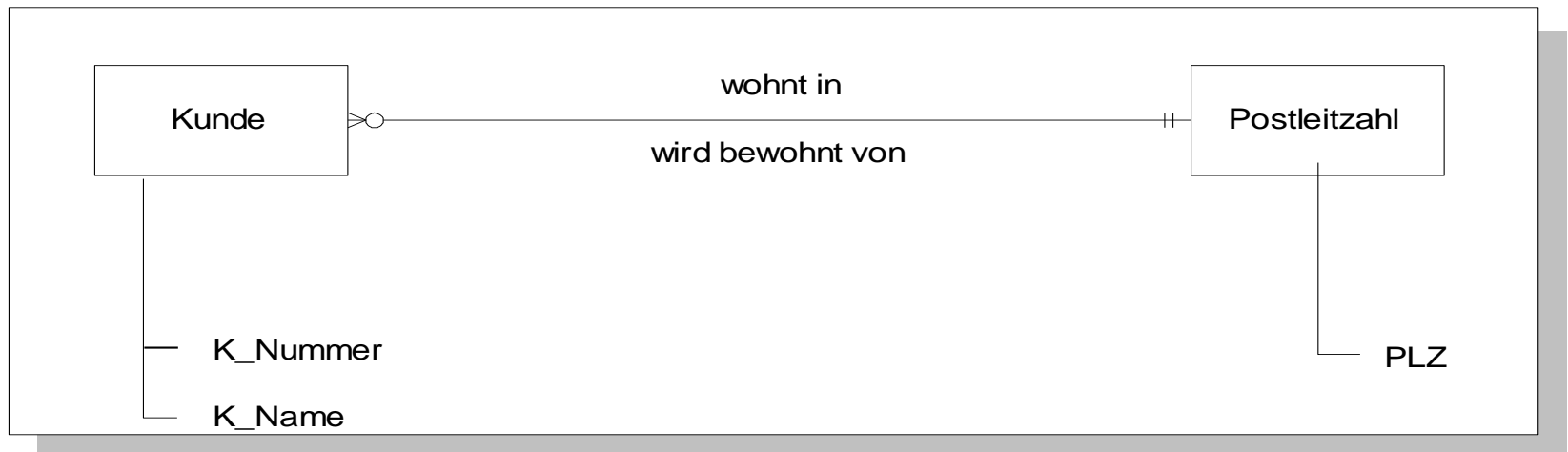
Beispiel: Gewicht eines Päckchens (stufenlos von 10 g bis 4000 g)

3. Einen praktisch unendlichen oder doch zumindest sehr großen Wertebereich, i. d. R. durch bestimmte Algorithmen definiert.

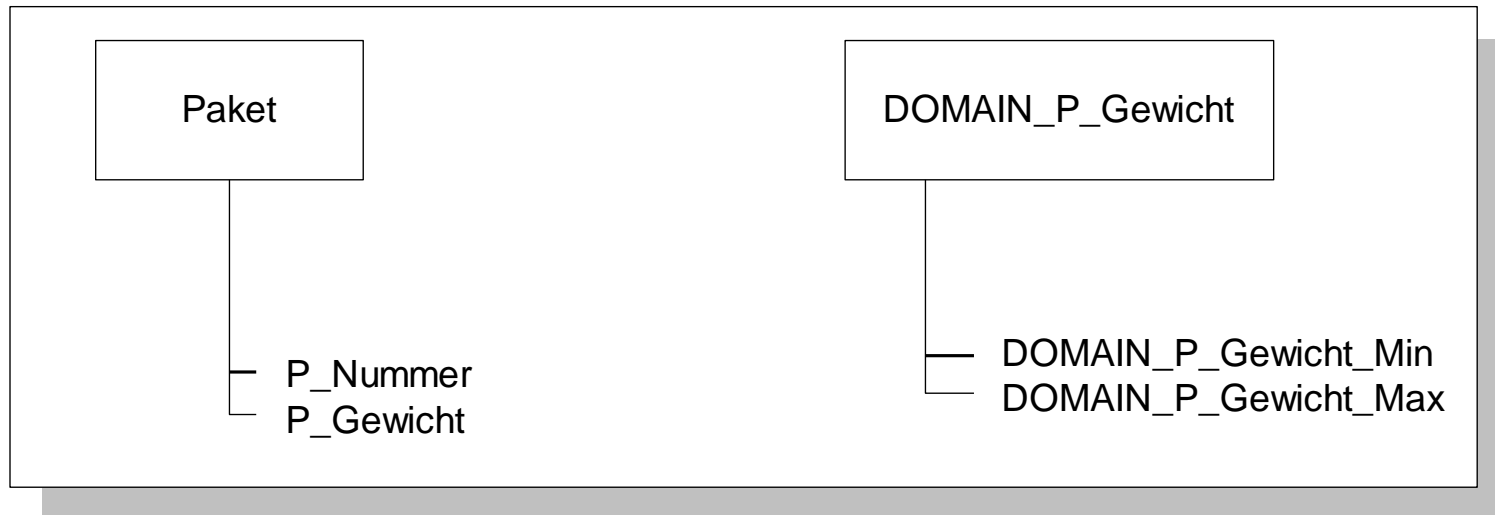
Beispiel: Name (alle möglichen Namen)

Beispiel: Datum (alle gültigen Tagesdaten)

## ■ Lösung Fall 1: wie Statusentität



- Lösung Fall 2: Wertebereiche mit Grenzen (Domänenentität)



- Lösung Fall 3: Wertebereiche „ohne“ Grenzen
  - Im dritten Fall gibt es i. a. keine Lösung im Datenmodell oder der Datenbank.

## Kapitel 7

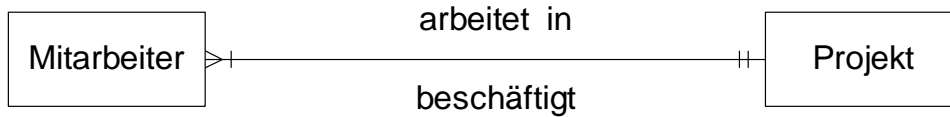
# Zusammenhang zwischen ER- und relationalem Modell

Eine wichtige Eigenschaft der ER-Modelle ist ihre leichte Überführbarkeit in relationales Design. Hierbei geht es vor allem um die Fragen:

- Was wird aus Entitätstypen?
- Was wird aus Attributtypen?
- Was wird aus Beziehungstypen?
- Wie wirken sich Beziehungskardinalitäten aus?



- Im Entity-Relationship-Modell gibt es
  - Entitätstypen
  - Beziehungstypen
  - Attributtypen



- Im relationalen Design werden
  - Entitätstypen zu Tabellen
  - Beziehungstypen zu Fremdschlüsseln in Tabellen
  - Attributtypen zu Spalten von Tabellen.

Mitarbeiter			

Projekt		

# Was wird im relationalen Design aus Entitätstypen und Attributtypen?

Produkt		
	Attributtypen	Datentyp
Id	<1 1> P_Nummer	9(9)
	<1 1> P_Name	X(10)
	<0 1> P_Preis	9(6),99

# Was wird im relationalen Design aus Entitätstypen und Attributtypen?

- Aus dem Entitätstyp Produkt wird die Tabelle Produkt mit folgendem SQL-Befehl:

```
CREATE TABLE PRODUKT
```

```
( P_NUMMER INTEGER      NOT NULL,  
  P_NAME     CHAR(10)    NOT NULL,  
  P_PREIS    DECIMAL (8,2)  
).
```

```
CREATE UNIQUE INDEX PRODUKT_IDX  
ON PRODUKT (P_NUMMER).
```

# Was wird im relationalen Design aus Entitätstypen und Attributtypen?

- Als Tabelle sieht PRODUKT wie folgt aus (hier mit Inhalt gefüllt):

<b><i>P_Nummer</i></b>	<b><i>P_Name</i></b>	<b><i>P_Preis</i></b>
000000001	Deckel.....	000003.64
000000023	Pinzel.....	000007.23
000020045	Schraube	000000.39

- Entitätstypen werden zu Tabellen.
- Attributtypen werden zu Spalten der Tabelle.
- Entitätsschlüssel werden zu Indizes.

# Was wird im relationalen Design aus den Kardinalitäten der Attributtypen?

Produkt		
	Attributtypen	Datentyp
Id	<1 1> P_Nummer	9(9)
	<1 1> P_Name	X(10)
	<0 1> P_Preis	9(6),99

CREATE TABLE PRODUKT

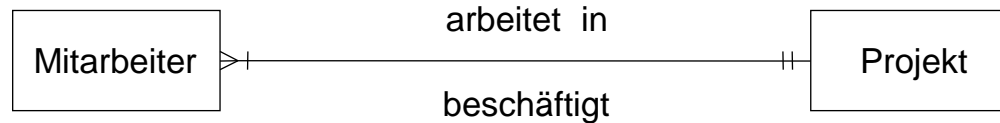
( P\_NUMMER INTEGER **NOT NULL,**  
P\_NAME CHAR(10) **NOT NULL,**  
P\_PREIS DECIMAL (8,2)  
).

# Was wird im relationalen Design aus den Kardinalitäten der Attributtypen?

## Fazit:

- Optionalität bei den Attribut-Kardinalitäten wird in NULL,
- Nicht-Optionalität bei den Attribut-Kardinalitäten wird in NOT NULL umgesetzt.

# Was wird im relationalen Design aus den Beziehungstypen?



Mitarbeiter			
		Attributtypen	Datentyp
Id	<1 1>	M_Nummer	9(4)
	<1 1>	M_Name	X(30)
	<0 1>	M_Verr_Satz	9(3),99

Projekt			
		Attributtypen	Datentyp
Id	<1 1>	P_Nummer	9(4)
	<1 1>	P_Name	X(30)
	<0 1>	P_End	Datum



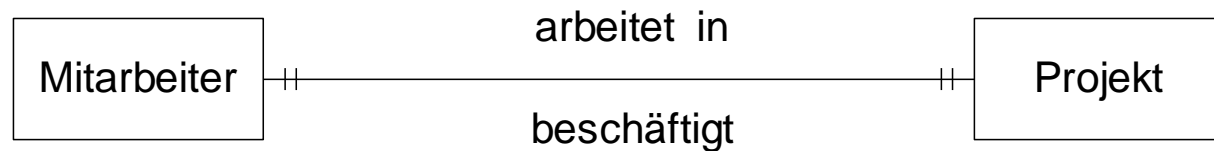
# Was wird im relationalen Design aus den Beziehungstypen?

<b><i>M_Nummer</i></b>	<b>M_Name</b>	<b>M_Verr_Satz</b>	<b>P_Nummer</b>
0012	Müller	123,45	4711
0013	Nagel	234,56	4710
0014	Puvogel	345,67	4711
0015	Rassel	456,78	4710
0016	Pflümli	012,34	4710

<b><i>P_Nummer</i></b>	<b>P_Name</b>	<b>P_Ende</b>
4710	Dat_Ana	2008.12.31
4711	Proz_Ana	2008.05.31

Fazit: Beziehungstypen werden im relationalen Design als Fremdschlüssel in Tabellen umgesetzt

# Wie wirkt sich eine 1:1 -Beziehung am Beispiel aus?



<b><i>M_Nummer</i></b>	<b><i>M_Name</i></b>	<b><i>M_Verr_Satz</i></b>	<b><i>P_Nummer</i></b>
0012	Müller	123,45	4710
0013	Nagel	234,56	4711

<b><i>P_Nummer</i></b>	<b><i>P_Name</i></b>	<b><i>P_Ende</i></b>
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31

# Wie wirkt sich eine 1:1 -Beziehung am Beispiel aus?

- oder

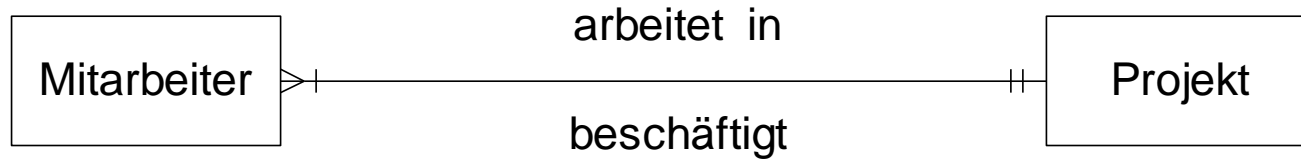
<b><i>P_Nummer</i></b>	<b><i>P_Name</i></b>	<b><i>P_Ende</i></b>	<b><i>M_Nummer</i></b>
4710	Dat-Ana	2008.12.31	0012
4711	Proz-Ana	2008.05.31	0013

<b><i>M_Nummer</i></b>	<b><i>M_Name</i></b>	<b><i>M_Verr_Satz</i></b>
0012	Müller	123,45
0013	Nagel	234,56

## Fazit:

- Bei 1:1 -Beziehungen (ohne Optionalität) ist es unerheblich, auf welcher Seite der Fremdschlüssel und auf welcher Seite der Primärschlüssel steht. Daher gibt es zwei gleichwertige Möglichkeiten.
- Letztlich bedeutet eine solche 1:1 -Beziehung, dass sich beide Tabellen ohne Informationsverlust zu einer Tabelle zusammenfassen lassen.

# Was wird im relationalen Design aus 1:N -Beziehungen?

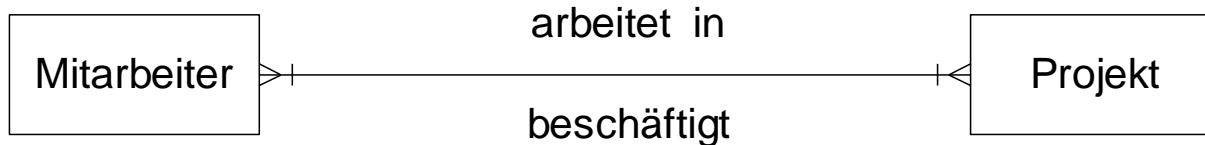


<i><b>M_Nummer</b></i>	<b>M_Name</b>	<b>M_Verr_Satz</b>	<b>P_Nummer</b>
0012	Müller	123,45	4711
0013	Nagel	234,56	4710
0014	Puvogel	345,67	4711
0015	Rassel	456,78	4710
0016	Pflümli	012,34	4710

<i><b>P_Nummer</b></i>	<b>P_Name</b>	<b>P_End</b>
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31

Fazit: Beziehungen werden bei 1:N als Fremdschlüssel in der Tabelle der "vielen" umgesetzt.

# Was wird im relationalen Design aus N:M -Beziehungen?



<b><i>M_Nummer</i></b>	<b><i>M_Name</i></b>	<b><i>M_Verr_Satz</i></b>
0012	Müller	123,45
0013	Nagel	234,56
0014	Puvogel	345,67
0015	Rassel	456,78
0016	Pflümli	012,34

<b><i>P_Nummer</i></b>	<b><i>P_Name</i></b>	<b><i>P_Ende</i></b>
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31

<b><i>M_Nummer</i></b>	<b><i>P_Nummer</i></b>
0012	4710
0013	4710
0013	4711
0014	4710
0015	4710
0015	4711
0016	4711

Fazit: Beziehungen werden im relationalen Design bei N:M grundsätzlich als Schlüsseltabelle umgesetzt

# Was wird im relationalen Design aus N:M -Beziehungen mit assoziativen Entitätstypen?



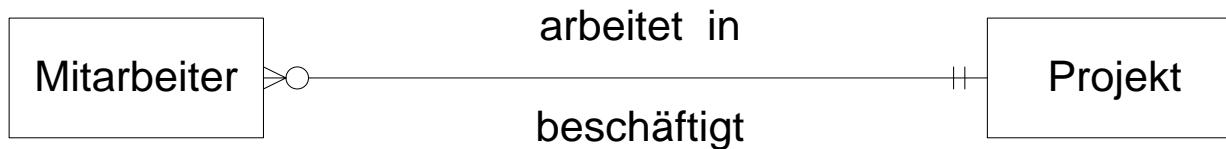
<i>M_Nummer</i>	M_Name	M_Verr_Satz
0012	Müller	123,45
0013	Nagel	234,56
0014	Puvogel	345,67
0015	Rassel	456,78
0016	Pflümli	012,34

<i>P_Nummer</i>	P_Name	P_Ende
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31

<i>M_Nummer</i>	P_Nummer	P_Eintritts_Datum
0012	4710	2007.01.10
0013	4710	2008.01.01
0013	4711	2007.05.15
0014	4710	2007.02.01
0015	4710	2007.10.01
0015	4711	2007.03.01
0016	4711	2006.12.01

Fazit: Auch assoziative Entitätstypen werden im relationalen Design grundsätzlich als Schlüsseltabelle umgesetzt. Hinzu kommt lediglich eine Anzahl weiterer Spalten.

# Was wird aus den Minimal-Kardinalitäten der Beziehungstypen?

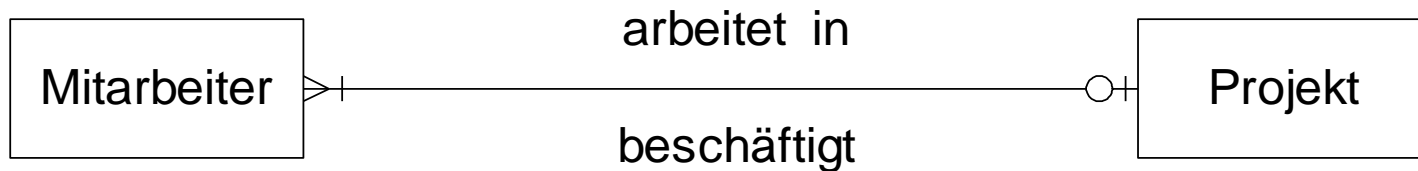


... ergibt im relationalen Design:

<i><b>M_Nummer</b></i>	<i><b>M_Name</b></i>	<i><b>M_Verr_Satz</b></i>	<i><b>P_Nummer</b></i>
0012	Müller	123,45	4711
0013	Nagel	234,56	4710
0014	Puvogel	345,67	4711
0015	Rassel	456,78	4710
0016	Pflümli	012,34	4710

<i><b>P_Nummer</b></i>	<i><b>P_Name</b></i>	<i><b>P_Ende</b></i>
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31
4713	Progr	2008.07.31

# Was wird aus den Minimal-Kardinalitäten der Beziehungstypen?



... ergibt im relationalen Design:

<i><b>M_Nummer</b></i>	<i><b>M_Name</b></i>	<i><b>M_Verr_Satz</b></i>	<i><b>P_Nummer</b></i>
0012	Müller	123,45	4711
0013	Nagel	234,56	4710
0014	Puvogel	345,67	4711
0015	Rassel	456,78	4710
0016	Pflümli	012,34	NULL

<i><b>P_Nummer</b></i>	<i><b>P_Name</b></i>	<i><b>P_Ende</b></i>
4710	Dat-Ana	2008.12.31
4711	Proz-Ana	2008.05.31



# Was wird aus den Minimal-Kardinalitäten der Beziehungstypen?

## Fazit:

- Je nachdem, auf welcher Seite die Null steht (bei den “vielen” oder den “wenigen”) hat die Optionalität unterschiedliche Auswirkungen.
- Im ersten Fall steht die Null bei den „Vielen“. Es gibt Projekte, (z. B. das Projekt 4713), die nicht als Fremdschlüssel vorkommen.
- Im zweiten Fall steht die Null bei den „Wenigen“. Es gibt Mitarbeiter, (z. B. der Mitarbeiter 0016), die keinen Fremdschlüssel haben. Dies wird durch das Symbol „NULL“ dargestellt.

## Kapitel 8

# Integritätsregeln

Der ER-Ansatz erlaubt durch Kombination seiner Elemente die Modellierung auch sehr komplexer Zusammenhänge zwischen mehreren Entitätstypen

Es ergeben sich jedoch noch Probleme:

- Die Zusammenhänge des konzeptionellen Modells sind statisch.
- ER-Modelle beschreiben nur die Beziehung zwischen Entitäts-Typen und nicht direkt die Beziehung zwischen Attributtypen verschiedener Entitätstypen.

- Bedeutung der Integritätsregeln

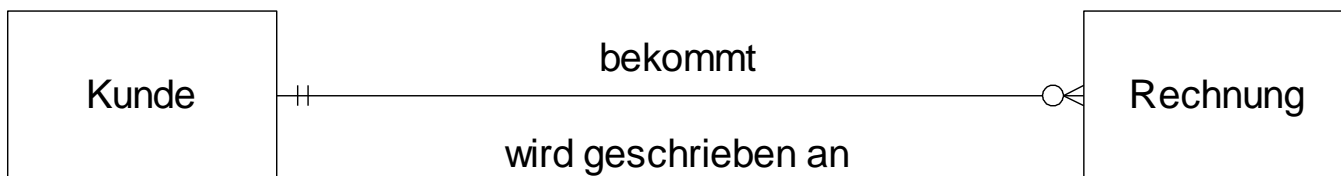
- Betrachten wir die Integritätsregeln (englisch: Integrity Constraints) an einem einfachen Beispiel:

Es sollen in einem System Rechnungen, die an Kunden geschrieben werden, erfasst und gespeichert werden. Dabei erscheinen folgende Regeln sinnvoll:

1. Eine Rechnung darf nur geschrieben werden, wenn der Kunde dazu bereits erfasst worden ist - keine Rechnung ohne Kunden!
2. Eine Rechnung darf immer nur an genau einen Kunden gehen.
3. Ein Kunde darf mehrere Rechnungen bekommen, kann aber auch keine Rechnung bekommen.
4. Der Kunde darf nicht gelöscht werden, solange noch Rechnungen ausstehen.
5. Bei Änderung der Kundendaten (etwa der Bonitätsangabe oder des Kreditlimits) muss ein Hinweis an den Sachbearbeiter erfolgen, dass noch eine Rechnung vorhanden ist.

# Ein mögliches ER-Modell sähe zum Beispiel so aus:

- Jeder Kunde bekommt keine bis mehrere Rechnungen.
  - Jede Rechnung wird geschrieben an genau einen Kunden.
  - Damit wären Regel 1, Regel 2 und Regel 3 abgedeckt.
- Prinzipiell ist auch die Regel 4 mit der Minimal-Kardinalität der Beziehung “wird geschrieben an” ausgedrückt, jedoch gibt es verschiedene Möglichkeiten diese Forderungen in der Datenbank umzusetzen: Wenn automatisch beim Löschen des Entitätstyps Kunde auch alle abhängigen Entitätstypen, also zum Beispiel Rechnung, gelöscht werden, ist das ER-Modell auch korrekt.
- Regel 5 (Wechselwirkung von Attributtypen in zwei verschiedenen Entitätstypen) lässt sich mit ERM nicht ausdrücken.



# Weitere Beispiele für (vernünftige) Forderungen an eine Datenbasis könnten sein:

- Alle Seminarnummern der Integrata müssen mit “INT” beginnen.
- Ein Seminarteilnehmer muss entweder ein Mann oder eine Frau sein.
- Eine Person kann nur dann den Status “geschieden” erhalten, wenn sie vorher verheiratet war.
- Eine Person ist nur dann in die Wählerliste aufzunehmen, wenn sie über 18 Jahre alt ist.

Solche Möglichkeiten sind in der ER-Modellierung nicht vorgesehen.

- Der Inhalt der Datenbank muss zu jeder Zeit ein korrektes Abbild der Wirklichkeit bleiben.
- Insbesondere muss bei INSERT, UPDATE, DELETE verhindert werden, dass Inkonsistenzen in Datenbeständen entstehen können.
- Wo immer möglich, sollten diese Regeln direkt an den Daten, die sie betreffen, gespeichert werden.

**Rechnung**

<i>R_Nr</i>	<i>R_Datum</i>	<i>K_Nr</i>
00012	12.07.1996	4711
00013	15.07.1996	4712
00014	16.07.1996	4711
00015	17.07.1996	4711
00016	17.07.1996	4712

**Kunde**

<i>K_Nr</i>	<i>K_Name</i>	<i>K_etc.</i>
4711	Willie	etc...
4712	Andrea	etc...
4713	Siemens	etc...

- Konflikte kann es jetzt z. B. bei folgenden Vorgängen geben:
  1. Einfügen einer neuen Rechnung 00017 ohne *K\_Nr*.
  2. Der Kunde Willie soll gelöscht werden.  
Was soll mit den Rechnungen 00012, 00014 und 00015 geschehen?
  3. Update der *K\_Nr* von Andrea.



- Zu 1.: Es ist möglich,
  - Rechnung 00017 einzufügen und unter K\_Nr einen NULL-Wert (das heißt nicht definierter Wert) zu setzen, wenn das Feld Rechnung\_K\_Nr diese Möglichkeit vorsieht (keine NOT NULL - Klausel im CREATE TABLE - Statement).
  - Die Eingabe vom System solange ablehnen zu lassen, bis eine korrekte K\_Nr eingegeben ist.
- Zu 2.: Es ist möglich,
  - Willie zu löschen und die “abhängigen” Rechnungen sofort mit zu löschen.
  - Willie nicht zu löschen und eine Meldung abzugeben (“Rechnungen stehen offen”).
  - Willie zu löschen und die Fremdschlüssel (K\_Nr) der Rechnungen auf NULL-Wert zu setzen.

- Zu 3.: Es ist möglich,
  - Andrea's K\_Nr zu ändern und alle abhängigen Rechnung\_K\_Nr gleich mit zu ändern.
  - Andrea's K\_Nr nicht zu ändern und eine Meldung abzugeben ("Rechnungen bestehen").
  - Andrea's K\_Nr zu ändern und die Rechnung\_K\_Nr der ehemals abhängigen Rechnungen auf NULL-Wert zu setzen.

- Im relationalen Design (z. B. DB2 SQL-DDL) gibt es die Möglichkeit, diese Beziehungsintegritäten zu definieren. Dabei verwendet man:
  - CASCADE
    - Mit CASCADE werden die abhängigen Fremdschlüssel-Tabellen mit der Primärschlüssel-Tabelle gelöscht.
  - RESTRICTED
    - Mit RESTRICTED wird das Löschen abgelehnt, bis alle abhängigen Tabellen gelöscht sind.
  - NULLIFIES
    - Mit NULLIFIES werden die Fremdschlüssel auf NULL-Werte gesetzt.

```
CREATE TABLE KUNDE

(      K_NR          INTEGER          NOT NULL,
        K_NAME CHAR (24)              NOT NULL,
        K_ETC        CHAR (50)        ,

        PRIMARY-KEY (K_NR)

).

CREATE TABLE RECHNUNG

(      R_NR          INTEGER          NOT NULL,
        R_DATUM      DATE              NOT NULL,
        K_NR          INTEGER          NOT NULL,

        PRIMARY-KEY          (R_NR)
        FOREIGN-KEY          (K_NR)
        REFERENCES KUNDE
        ON DELETE CASCADE

).
```

	Insert/Update FS	Delete PS	Update PS
<b>CASCADE</b>		FS wird gelöscht, wenn PS gelöscht.	Relation FS wird geändert, wenn PS geändert wird.
<b>RESTRICTED</b>	FS kann nur eingefügt werden, wenn PS existiert.	PS darf nur gelöscht werden, wenn kein FS mehr existiert.	PS darf nur geändert werden, wenn FS nicht existiert.
<b>NULLIFIES</b>	FS ohne PS wird als NULL-Wert eingefügt.	PS wird gelöscht, FS wird NULL-Wert.	PS wird geändert, FS wird NULL-Wert.

## Kapitel 9

# Qualitätssicherung und Datenmodellierung

Provokation oder bittere Realität?

Warum Projekte immer wieder scheitern?

Titel der Untersuchung: 'Chaos Report'

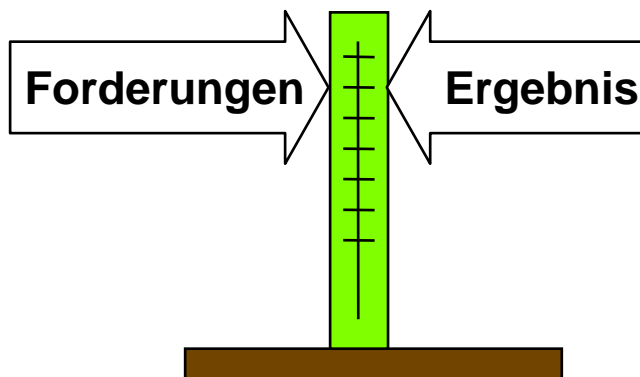
- 53 Prozent der Projekte dauern länger und kosten viel mehr als geplant
- 18 Prozent der IT-Projekte scheitern gänzlich
- Bessere Planung, Kommunikation und Qualitätssicherung könnten Abhilfe schaffen

"Warum haben wir eigentlich nie Zeit, eine Sache richtig zu machen, aber immer Zeit, sie noch mal zu machen?" (Gerald Weinberg)

## ■ Qualität

Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt

Diese Definition ist sehr abstrakt und nicht sofort zu verstehen. Die folgende Grafik soll den Qualitätsbegriff verdeutlichen:



### 1. Optimale Qualität

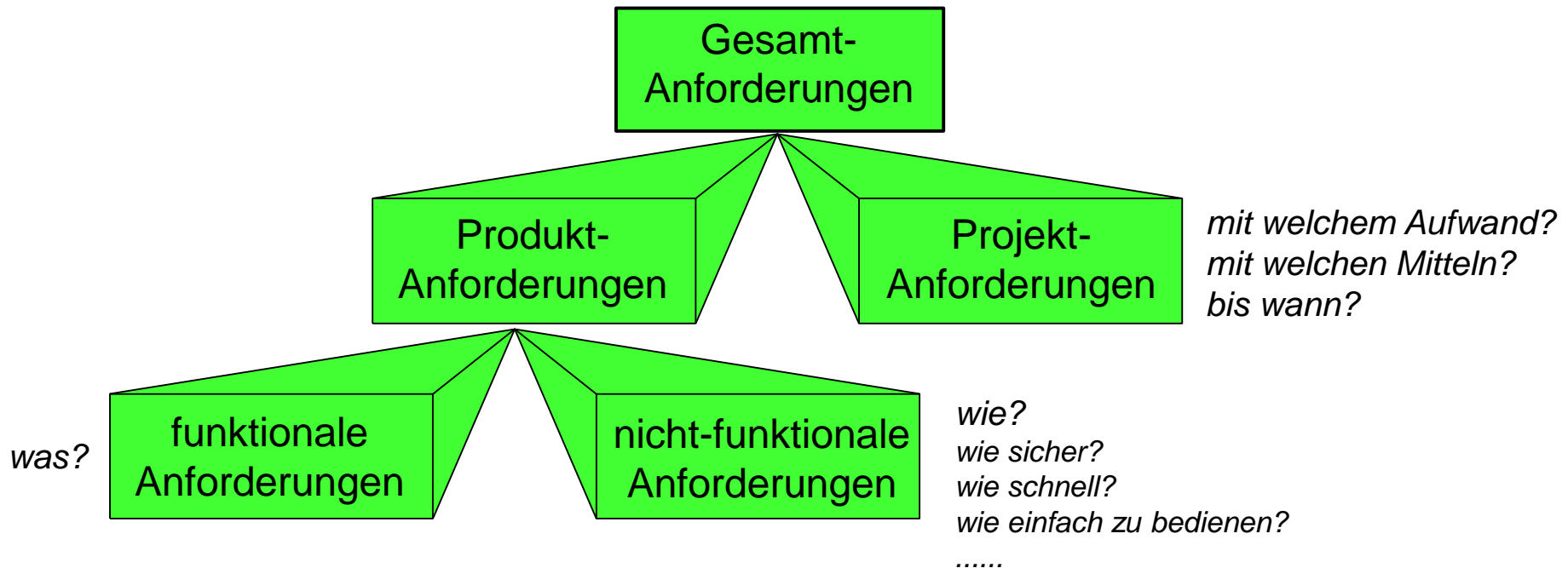
Von optimaler Qualität sprechen wir dann, wenn **das Ergebnis die Forderungen genau erfüllt**.

Ein besseres Ergebnis hat also eine "schlechtere Qualität" als das optimale. Man spricht von Übererfüllung (VW bestellt und Porsche geliefert). Sie ist unerwünscht, weil das bessere Ergebnis in der Regel mehr kostet ohne dass dafür mehr verlangt werden kann.



## Anforderungen

Anforderungsbaum in Anlehnung an Frühauf



## Funktionale Anforderungen

Die Funktionen (und Abläufe), die zur Problemlösung notwendig sind

Beispiel Auto: Beschleunigen (0-100 in ... s), bremsen (100-0 in ... s),  
Anzahl Sitzplätze, Kofferraumgröße,

***Funktionale Anforderungen können mit  
Use Cases beschrieben werden***

## **Nichtfunktionale Anforderungen**

(auch häufig als Qualitätsanforderungen bezeichnet) sind essentiell für erfolgreiche Projekte und Produkte.

- Qualitätsanforderungen,
- technische Anforderungen,
- organisatorische Anforderungen und
- Anforderungen an sonstige Lieferbestandteile.

## Nichtfunktionale Anforderungen

Aussagen, wie (genau, schnell, sicher,...) die Funktionen auszuführen sind

### **Beispiel Auto:**

**Anwender:** freundlicher Service, rückenfreundliche Sitze,  
☞ Instrumente übersichtlich und blendfrei, leicht zu reparieren,  
☞ Image des Herstellers, Preis, Lieferzeit,...

**Werkstatt:** leicht zu reparieren, wenig Schulung,  
☞ wenig neue Werkzeuge,...

**Hersteller:** Wiederverwendung von Baugruppen, zuverlässige Lieferanten,  
☞ Auslastung der Fabrik,...

## Akzeptanzkriterien für Anforderungen

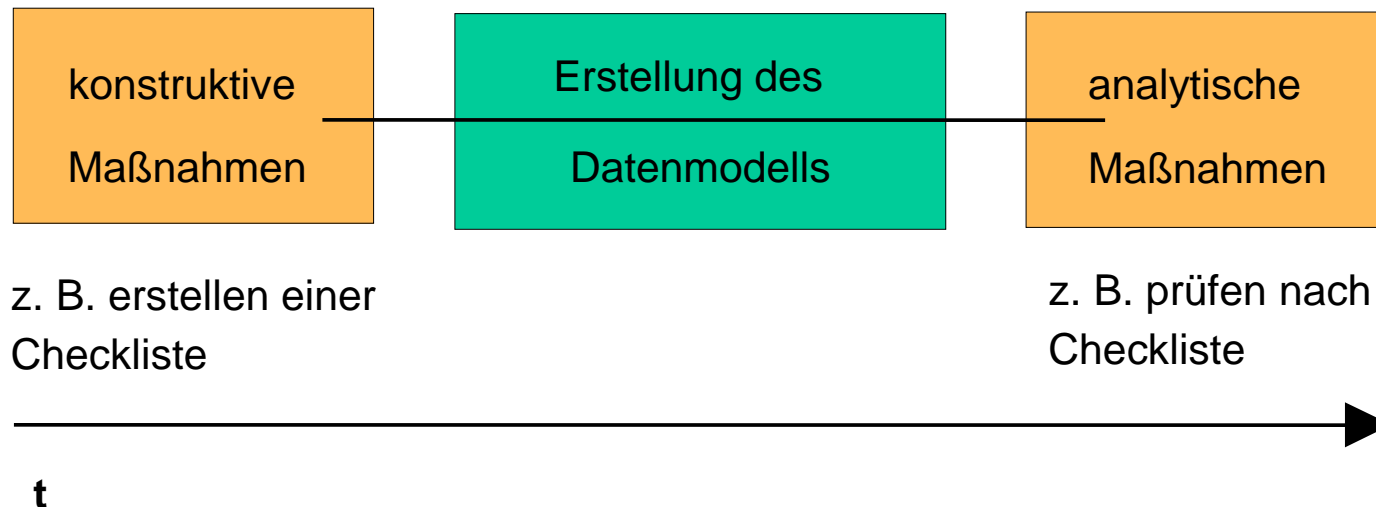
Anforderungen werden präzisiert durch **Akzeptanzkriterien**

Diese geben an, welche Bedingungen erfüllt sein müssen, damit das Produkt (System, Teilsysteme) **abgenommen** wird

Die Akzeptanzkriterien sollen nach Möglichkeit als messbare Größen formuliert sein

Messbar heißt: als Kategorie, als Zahlenwert, als Vergleichswert, als Rang, usw.

## Beispiel: Konstruktive und analytische Maßnahmen für ein Datenmodell



1. Vollständige Definitionen von Entitätstypen, Attributtypen und Beziehungstypen
2. Vollständigkeit der relevanten Attributtypen
3. Es dürfen keine Homonyme / Synonyme auftreten
4. Eindeutige Attributtypzuordnung
5. Überflüssige Beziehungstypen müssen entfernt werden
6. Isolierte Entitätstypen müssen überprüft werden
7. Konsistenzprüfung Datenmodell und Funktionsmodell
8. Das Datenmodell muss in der 3. Normalform vorliegen

Def.Entitätstyp Kunde:  
Person, die unsere Produkte kauft.

Kunde Entitätstypbeschreibung  
Entitätsschlüssel: Kunde\_Nummer

Attributtypen:

Id	<1 1> Kunde_Nummer	9(8)
	<1 1> Kunde_Name	X(30)
	<0 1> Kunde_IBAN	IBAN
	<0 1> Kunde_BIC	BIC
	<0 1> Kunde_Telefon	Telnr





Def. Attributtyp Kunde\_Nummer:  
Identifiziert den Kunden,  
fortlaufend maschinell vergeben

- Homonyme sind unterschiedliche Dinge mit gleichem Namen. Sie entstehen, wenn Definitionen nicht präzise sind oder überhaupt nicht vorliegen.
- Beispiele:
- “Lieferung“ kann sein:
  - Lieferung von und an Kunden
  - Lieferung unseres Lieferanten an uns
- “Kunde“ kann sein:
  - ein Käufer
  - ein Interessent
  - Besteller einer Ware

- Synonyme sind gleiche Dinge mit unterschiedlichen Namen (Aliasnamen). Sie entstehen oft, wenn getrennte Teams an der Definition arbeiten, zum Beispiel an der Definition von Attributtypen für Entitätstypen. Als Aliasnamen können sie sinnvoll sein.

- Beispiele:

statt	besser
“Adresse“	Kunde_Anschrift Kunde_Adresse Kunde_Postanschrift Kunde_Wohnung
“Name“	Vorname Geburtsname Nachname Akad. Titel
“Geschäftspartner“	Kunde Interessent Lieferant



- Das Datenmodell muss in dritter Normalform vorliegen. Dadurch können wir es problemlos in ein vernünftiges relationales Design übertragen (beziehungsweise durch ein CASE-Tool übertragen lassen).

Der Ausdruck 3. Normalform stammt aus der Welt des relationalen Datenmodells. Normalisierung ist die von E.F.Codd beschriebene schrittweise Vorgehensweise, mit der wir sicherstellen, dass nur solche Daten zu Tabellen zusammengefasst werden, die logisch zusammengehören. Die einzelnen Schritte gehen von der völlig unnormalisierten Form bis hin zur 3. Normalform.

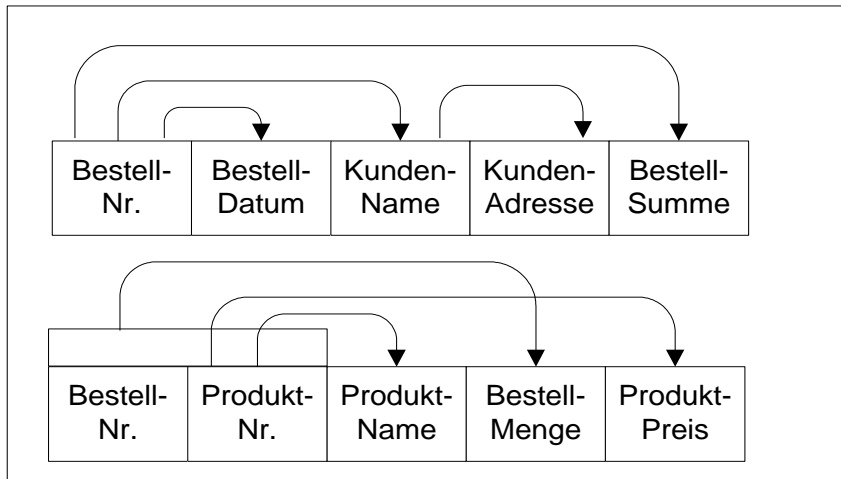
## 1.: Erste Normalform bilden:

Wiederholungsgruppen aus dem unnormalisierten Satz entfernen.

Bestell-Nr.	Bestell-Datum	Kunden-Name	Kunden-Adresse	Produkt-Nr.	Produkt-Name	Bestell-Menge	Produkt-Preis	Bestell-Summe
-------------	---------------	-------------	----------------	-------------	--------------	---------------	---------------	---------------



1.NF:



## Beispiel: Unnormalisierte Datenstruktur

### 01-01000-N0-Angebot

EL Kundennummer (PS)

EL Angebotsnummer (PS)

EL Angebotsdatum

IFL Hardware-Produkt (1-N)

EL Produktnummer

EL Menge

EL Angebot.Preis

EL Angebotsrabatt

### 02-01000-N0-Produktinformationen

EL Produktnummer (PS)

EL Preis

IFL Lieferantendaten

EL Lieferantenummer

EL Lieferantenbezeichnung

EL Lieferantenort

### 02-02000-N0-Produktbezeichnung

EL Schlagwortkurzbezeichnung (PS)

EL Produktnummer (PS)

EL Schlagwort

EL Produktbezeichnung

## Beispiel: Datenstruktur in der 1. Normalform

### 01-01100-N1-Angebot

EL Kundennummer (PS)

EL Angebotsnummer (PS)

EL Angebotsdatum

EL Angebotsrabatt

### 01-01200-N1- Hardware-Produkt

EL Kundennummer (PS)

EL Angebotsnummer (PS)

EL Produktnummer (PS)

EL Menge

EL Angebot.Preis

### 02-01000-N1-Produktinformationen

EL Produktnummer (PS)

EL Preis

IFL Lieferantendaten

EL Lieferantenummer

EL Lieferantenbezeichnung

EL Lieferantenort

### 02-02000-N1-Produktbezeichnung

EL Schlagwortkurzbezeichnung (PS)

EL Produktnummer (PS)

EL Schlagwort

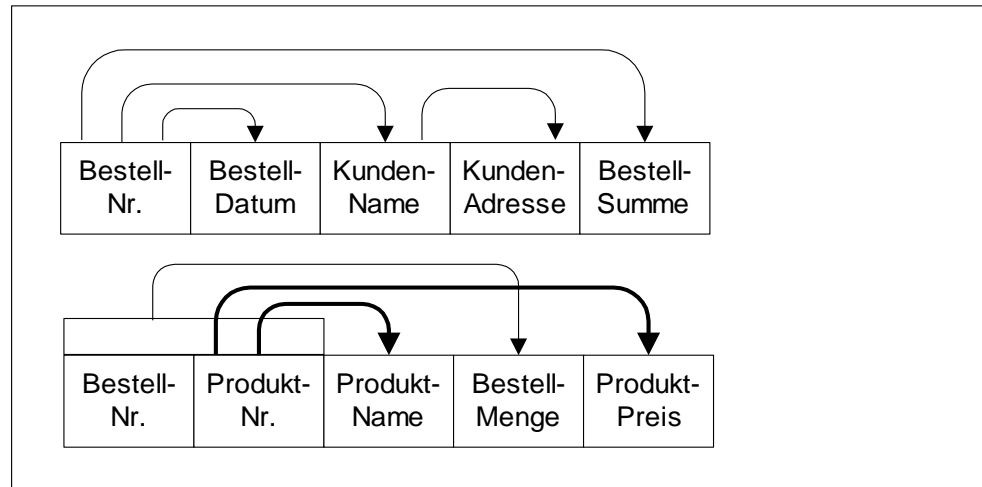
EL Produktbezeichnung



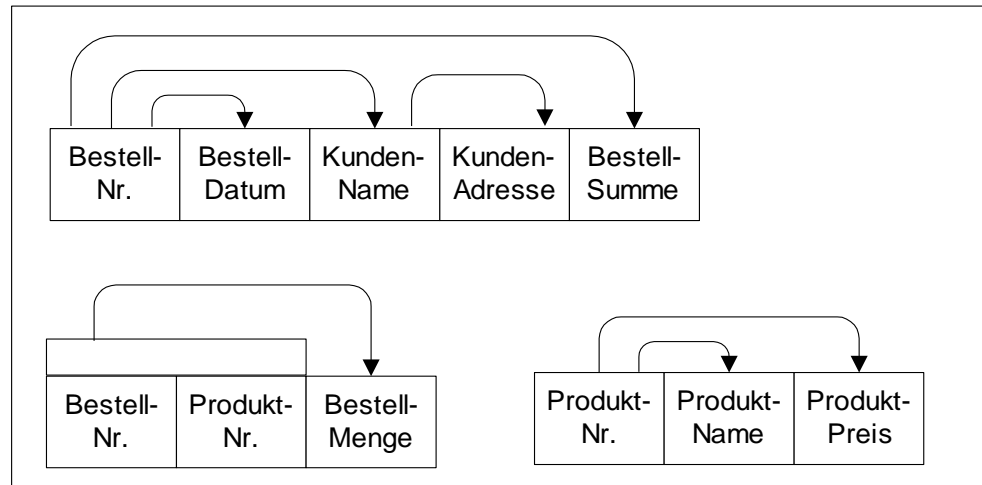
## 2.: Zweite Normalform:

Entfernen von Attributtypen, die nur von Schlüsselteilen abhängig sind.

1. NF:



2. NF:



## Von der 1. Normalform

02-02000-N1-Produktbezeichnung

EL Schlagwortkurzbezeichnung (PS)

EL Produktnummer (PS)

EL Schlagwort

EL Produktbezeichnung

## In die 2. Normalform

### **02-02100-N2-Produktbezeichnung**

EL Schlagwortkurzbezeichnung (PS)

EL Produktnummer (PS)

### **02-02200-N2- Schlagwort**

EL Schlagwortkurzbezeichnung (PS)

EL Schlagwort

### **02-02300-N2-Produkt**

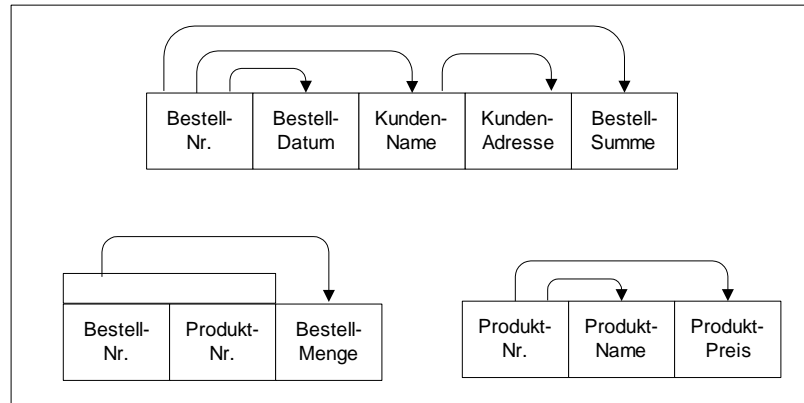
EL Produktnummer (PS)

EL Produktbezeichnung

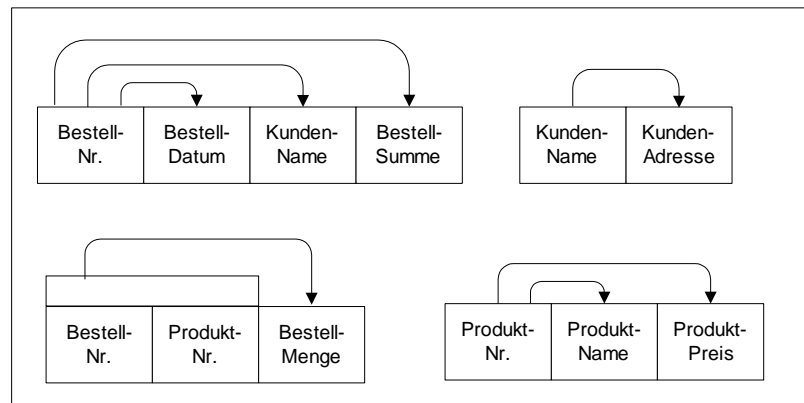
3.: Dritte Normalform:

Sämtliche Attributtypen, die von anderen Attributtypen abhängig sind, werden aus den Strukturen ausgelagert.

2. NF:



3. NF:



# Datenstrukturen in die 3. Normalform übertragen

## 01-01100-N3-Angebot

EL Kundennummer (PS)

EL Angebotsnummer (PS)

EL Angebotsdatum

EL Angebotsrabatt

## 01-01200-N3- Hardware-Produkt

EL Kundennummer (PS)

EL Angebotsnummer (PS)

EL Produktnummer (PS)

EL Menge

EL Angebot.Preis

## 02-01100-N3-Produktinformationen

EL Produktnummer (PS)

EL Preis

EL Lieferantenummer (FS)

## 02-01200-N3-Lieferant

EL Lieferantenummer (PS)

EL Lieferantenbezeichnung

EL Lieferantenort

## 02-02100-N3-Produktbezeichnung

EL Schlagwortkurzbezeichnung (PS)

EL Produktnummer (PS)

## 02-02200-N3- Schlagwort

EL Schlagwortkurzbezeichnung (PS)

EL Schlagwort

## 02-02300-N3-Produkt

EL Produktnummer (PS)

EL Produktbezeichnung

