

# Datenbanksysteme

## Ein Glossar

von:

**Christian Bräunlich**

Martikel-Nr.: xxxxxx

Studiengang: Wirtschaftsinformatik

3. Semester

Straße

PLZ Flensburg

Tel.:

Textfelder: STRG + F9

Ansicht: ALT + F9

Hochschule Flensburg

Fakultät

Fachbereich 4: Wirtschaft

Bachelorstudiengang: Wirtschaftsinformatik

Gutachter: Prof. Dr. Vorname Nachname

Zweitgutachter: Prof. Dr. Vorname Nachname

Abgabetermin:

# Inhaltsverzeichnis

<b>1</b>	<b>Datenbanksysteme</b>	<b>1</b>
1.1	Einführung	1
1.2	Aufgaben von Datenbanksystemen	1
<b>2</b>	<b>Grundlagen von Datenbanksystemen</b>	<b>3</b>
2.1	Definition Datenbanksystemen	3
2.1.1	Datenbanksystem	3
2.1.2	Datenbank	3
2.1.3	Datenbasis	3
2.1.4	Data Dictionary	4
2.1.5	Datenbankmanagementsystem	4
2.2	Arten von Datenbanksystemen	4
2.2.1	Key-Value-Store	4
2.2.2	Dokumenten-Datenbanken	5
2.2.3	Graph-Datenbanken	5
2.3	Architektur von Datenbanksystemen	5
2.3.1	Physische Datenunabhängigkeit	5
2.3.2	Logische Datenunabhängigkeit	5
2.3.3	ANSI-SPARC-3-Ebenen-Modell	6
2.4	Phasenmodell	6
<b>3</b>	<b>Grundlagen des konzeptuellen Entwurfs</b>	<b>8</b>
3.1	Überblick	8
3.2	Datenmodell	8
3.3	Geschäftsobjekte	8
3.4	Attribute	9
3.5	Wertebereiche	9
3.6	Schlüssel	10

3.7	Beziehungen -----	10
3.8	Schwache Entitäten -----	11
3.9	Sub- und Supertypen -----	12
<b>4</b>	<b>Entity-Relationship-Modell -----</b>	<b>13</b>
4.1	Chen -----	13
4.1.1	Einführung -----	13
4.1.2	Beispiele -----	14
4.2	Barker -----	16
4.2.1	Übung-----	17
<b>5</b>	<b>Relationsmodell -----</b>	<b>18</b>
5.1	Grundlagen-----	18
5.1.1	Integritätsbedingungen -----	18
<b>6</b>	<b>Logischer Entwurf -----</b>	<b>19</b>
6.1.1	Überblick -----	21
6.1.2	Übung-----	21
<b>7</b>	<b>Relationale Entwurfstheorie -----</b>	<b>23</b>
7.1	Normalformen -----	23
7.2	Schlüssel-----	25
7.2.1	Eindeutiger Schlüssel-----	26
7.3	Zerlegung-----	26
7.3.1	1. Normalform-----	27
7.3.2	2. Normalform-----	27
7.3.3	3. Normalform-----	29



# 1 Datenbanksysteme

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem

## 1.1 Einführung

## 1.2 Aufgaben von Datenbanksystemen

Datenbankmanagementsysteme umfassen die folgenden Aufgaben:

Redundanz: Mehrfachspeicherung von Daten

Konsistenz: Widerspruchsfreie und semantisch korrekte Speicherung von Daten

Datensicherheit: Bewahrung der Daten vor Verfälschung, Vernichtung und unberechtigten Zugriff

Datenschutz: Verhinderung der unberechtigten Verwendung von Daten, z.B. von personenbezogenen Daten.

Redundanzabbau	Daten sollten nicht, wenn nicht gewollt, mehrfach gespeichert werden. Dies betrifft einzelne Datensätze, wie auch einzelne Einträge in den Datensätzen.
Datenunabhängigkeit	Daten und Anwendungsprogramme sind weitestgehend voneinander getrennt. Änderungen an den Daten haben keinen Einfluss auf die Anwendungsprogramme. Es wird unterschieden zwischen logischer und physischer Datenunabhängigkeit. Die logische Datenunabhängigkeit fordert Unabhängigkeit zwischen konzeptuellen Schema und Anwendungsprogramm bzw. den Benutzer. Die physische Datenunabhängigkeit fordert die Unabhängigkeit zwischen Schema und der physischen Datenorganisation.
Datenintegrität /- konsistenz	Unter Datenbankintegrität / -konsistenz versteht man die Widerspruchsfreiheit einer Datenbank. Die Daten müssen hierzu fehlerfrei erfasst sein.

## 1. Datenbanksysteme

	Integritätsbedingungen sollen bei Einfüge- oder Änderungsoperationen gewährleisten, dass die Konsistenz sichergestellt ist.
Datensicherheit	Daten werden vor Verfälschung, Vernichtung und vor unberechtigtem Zugriff geschützt.
Datenschutz (auch: Zugriffsschutz)	Es wird sichergestellt, dass Daten nicht unberechtigt verwendet werden.
Benutzersichten	Es werden, beispielsweise für Abteilungen, unterschiedliche Sichten auf die Daten zur Verfügung gestellt.
Mehrbenutzerzugriff	Auch bei gleichzeitigem Zugriff durch mehrere Personen ist sichergestellt, dass die Daten fehlerfrei bleiben (auch operative Integrität).

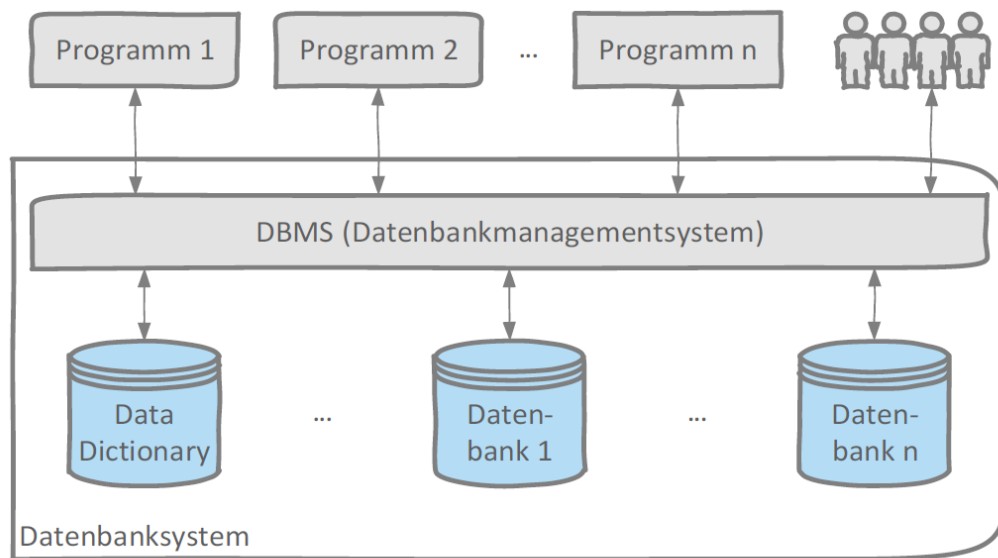
## 2 Grundlagen von Datenbanksystemen

Ut wisi ?

### 2.1 Definition Datenbanksystemen

#### 2.1.1 Datenbanksystem

„Ein Datenbanksystem (DBS) ist eine Ansammlung von Daten, die allen Benutzern bzw. Anwendungen zur Verfügung steht und in der die Daten nach einheitlichen Regeln abgespeichert werden. Ein Datenbanksystem besteht aus einer Datenbasis und einem Datenbankmanagementsystem. Der Begriff der Datenbank wird synonym verwendet.“ – Faeskorn-Woyke (2007), S. 21 f.



Quelle: Faeskorn-Woyke et al. (2007), S. 22, mit eigenen Anpassungen

#### 2.1.2 Datenbank

„Eine Datenbank ist eine Sammlung von Daten, die untereinander in einer logischen Beziehung stehen [...]“ (Schicker, 2017, Kap. 1.1).

Neben den eigentlichen Datenbanken werden durch ein Datenbankmanagementsystem auch Metadaten gespeichert. Diese Datenbasis wird als Data Dictionary oder Repository bezeichnet.

#### 2.1.3 Datenbasis

"Unter Datenbasis versteht man die eigentlichen Daten der Datenbank, die im Dateisystem gespeichert werden, also eine gewisse Anzahl von physikalischen

## 2. Grundlagen von Datenbanksystemen

Dateien, in denen die Anwendungsdaten und das Data Dictionary[..] gespeichert werden." (Faeskorn-Woyke(2007), S. 22)

### 2.1.4 Data Dictionary

"Das DataDictionary enthält Daten (sogenannte Metadaten), die die Datenbasis, z. B. Tabellenstrukturen, definieren. Es umfasst außerdem Daten über die Verwendung[..], die Beziehungen der Daten untereinander und Integritätsbedingungen." – Quelle: Faeskorn-Woyke(2007), S. 22 f

### 2.1.5 Datenbankmanagementsystem

Sammlung von Programmen zum Erstellen und Verwalten einer Datenbank [oder auch mehrerer Datenbanken], die es mehreren Anwendungen gleichzeitig ermöglicht, die von ihnen benötigten Daten zu speichern, zu extrahieren und zu manipulieren, ohne jeweils eigene Dateien erstellen zu müssen.

## 2.2 Arten von Datenbanksystemen

### 2.2.1 Key-Value-Store

Eigenschaften:

- Es gibt eine Menge von identifizierenden Datenobjekten, die Schlüssel.
- Zu jedem Schlüssel gibt es genau ein assoziiertes deskriptives Datenobjekt, welches den Wert zum zugehörigen Schlüssel darstellt.
- Mit der Angabe des Schlüssels kann der zugehörige Wert aus der Datenbank abgefragt werden.



### 2.2.2 Dokumenten-Datenbanken

Speichern strukturierte Daten in Datensätzen

"Filmname" : "The Kid",

"Regisseur": "Turteltaub",

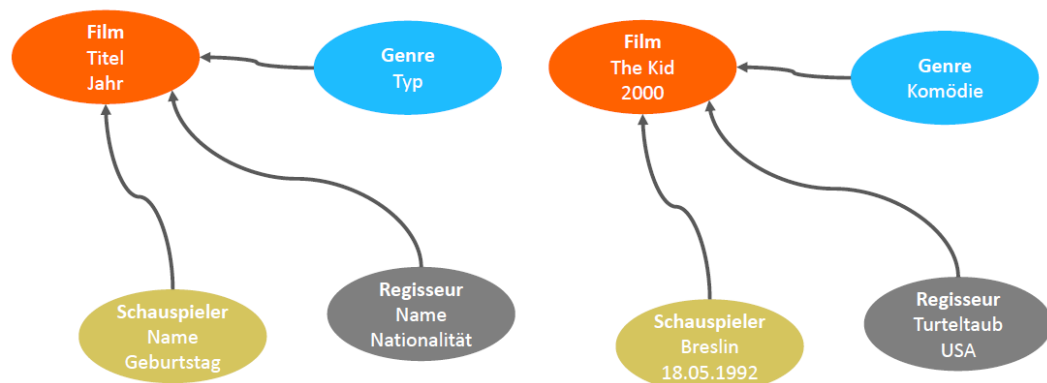
"Year": "2000",

"Genre": ["Comedy", "Family", "Fantasy"],

"Actor": ["Bruce Willis", "Spencer Breslin", "Emily Mortimer", "Lily Tomlin"],

"Plot": "An unhappy and disliked image consultant gets a second shot at life when he is mysteriously confronted by an eight-year-old version of himself."

### 2.2.3 Graph-Datenbanken



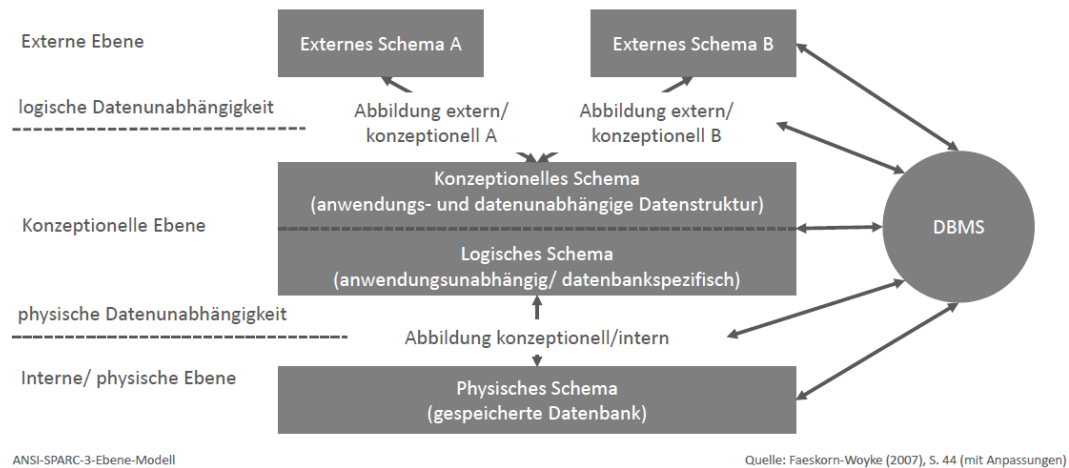
## 2.3 Architektur von Datenbanksystemen

### 2.3.1 Physische Datenunabhängigkeit

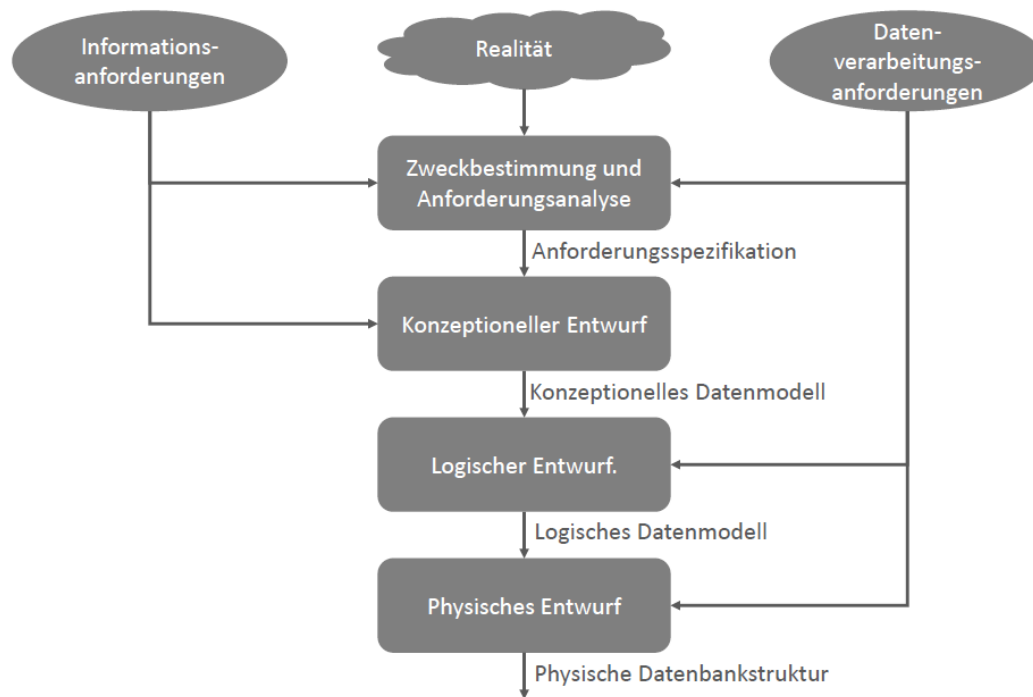
### 2.3.2 Logische Datenunabhängigkeit

## 2. Grundlagen von Datenbanksystemen

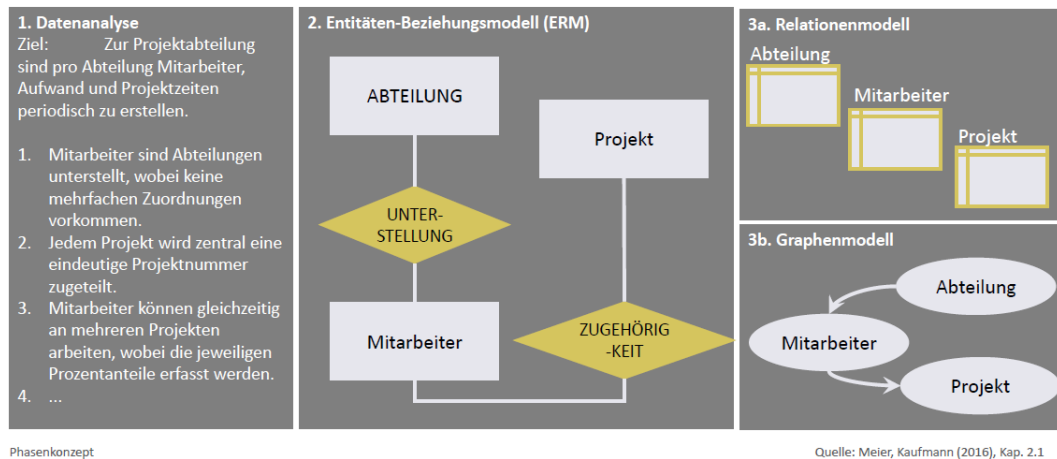
### 2.3.3 ANSI-SPARC-3-Ebenen-Modell



### 2.4 Phasenmodell



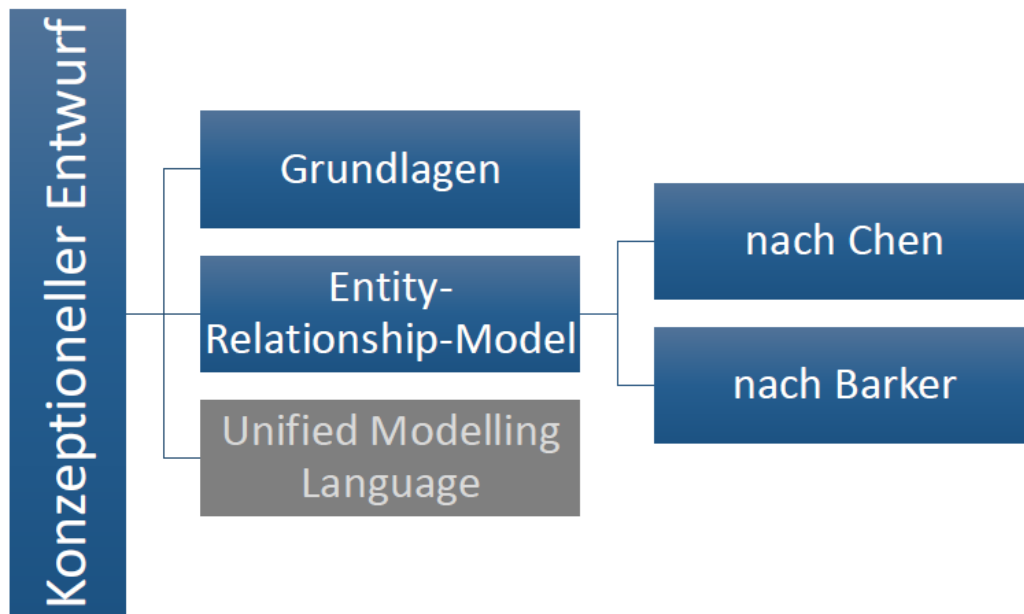
## 2. Grundlagen von Datenbanksystemen



1. Identifikation von Organisationseinheiten
2. Identifikation der zu unterstützenden Aufgaben
3. Anforderungssammelplan: Ermittlung der zu befragenden Personen
4. Anforderungssammlung
5. Filterung: Gesammelte Information auf Verständlichkeit und Eindeutigkeit überprüfen
6. Satzklassifikation: Information wird Objekten, Beziehungen zwischen Objekten, Operationen und Ergebnissen zugeordnet
7. Formalisierung bzw. Systematisierung: Übertragung auf Verzeichnisse, die in ihrer

## 3 Grundlagen des konzeptuellen Entwurfs

### 3.1 Überblick



### 3.2 Datenmodell

"Ein Datenmodell stellt einen allgemeinen Begriffsapparat zur Verfügung, der es gestattet, Realitätsausschnitte ohne Eingrenzung auf ein bestimmtes Sachgebiet zu modellieren..." (Faeskorn-Woyke(2007), S. 30)

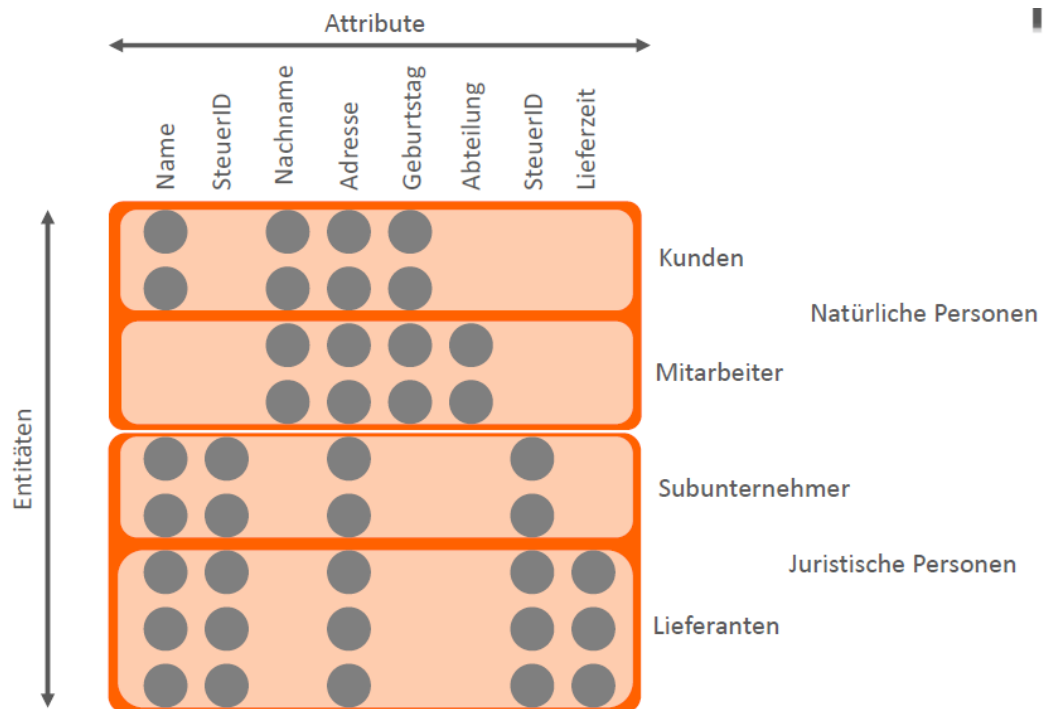
"Ein Datenmodell beschreibt auf strukturierte und formale Weise die für ein Informationssystem notwendigen Daten und Datenbeziehungen." (Meier, Kaufmann (2016), Kap. 2.1)

### 3.3 Geschäftsobjekte

Ein Geschäftsobjekt ist ein eindeutig zu identifizierendes Objekt, das von anderen Objekten durch seine Eigenschaften unterschieden werden kann.

- Etwas real Existierendes
- Ein Ereignis
- Eine Rolle oder Person
- Eine Organisation
- Ein Konzept
- Eine Transaktion

### 3. Grundlagen des konzeptuellen Entwurfs



### 3.4 Attribute

Attribut: Entitäten (bzw. Geschäftsobjekte) werden durch Attribute beschrieben. Alle Entitäten (Geschäftsobjekte) einer Entitätsmenge haben dieselben Attribute.

Attributwerte: Attributwerte beschreiben die Entität (das Geschäftsobjekt).

atomar	vs.	zusammengesetzt
einwertig ("single-valued")	vs.	mehrwertig (multivalued")
unabhängig	vs.	abhängig
optional (NULL-Werte zulässig)	vs.	obligatorisch (NULL-Werte unzulässig)

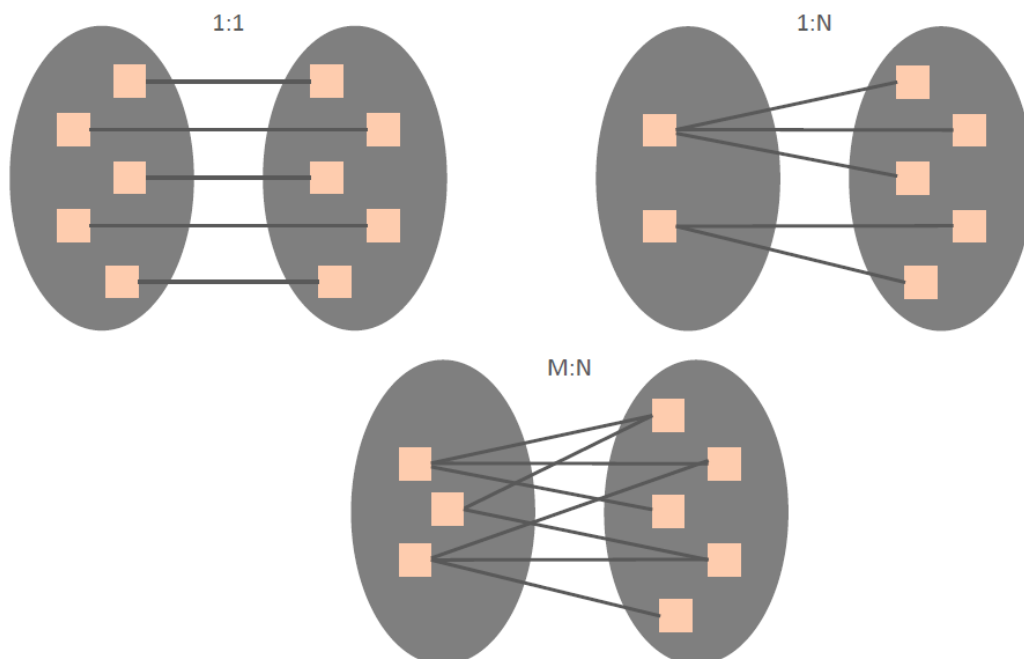
### 3.5 Wertebereiche

## 3.6 Schlüssel

- **Schlüssel:** "Minimale Menge von Attributen, die die zugeordnete Entitäten eindeutig identifizieren. Diese Menge kann sich aus einem oder mehreren Schlüsseln zusammensetzen." (Faeskorn-Woyke 2007, S. 81)
- **Eindeutigkeitsbedingung (-einschränkung):** Eine Entität lässt sich eindeutig identifizieren (Elmasri, Navathe (2009), S. 66).
- **Zusammengesetzter Schlüssel:** Schlüssel, der aus einer Kombination von Attributen resultiert (basierend auf Elmasri, Navathe (2009), S. 66).
- **Künstliche Schlüssel (auch: Surrogatschlüssel):** Bei einem künstlichen Schlüssel handelt es sich um ein zusätzliches Attribut, dass keine Entsprechung in der realen Welt hat. Dabei handelt es sich in der Regel um einen einfachen Zähler (z.B. Kundennummer, Personalnummer).
- **Forderung nach Minimalität:** Ein Primärschlüssel soll eine minimale Anzahl von Attributen enthalten. Kein Attribut darf wegfallen, um die Eindeutigkeit aller Entitäten zu gewährleisten.

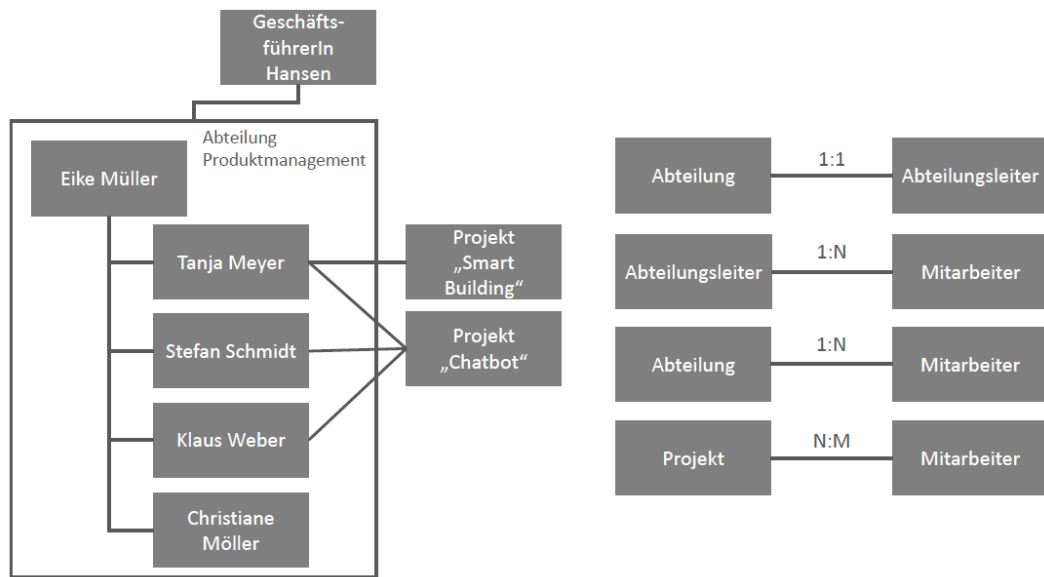
## 3.7 Beziehungen

Kardinalität: Die Kardinalität beschreibt, wie viele Entitäten eines Entitätstyps in Beziehungen zu Entitäten eines zweiten Entitätstyps stehen und umgekehrt (Maximum).



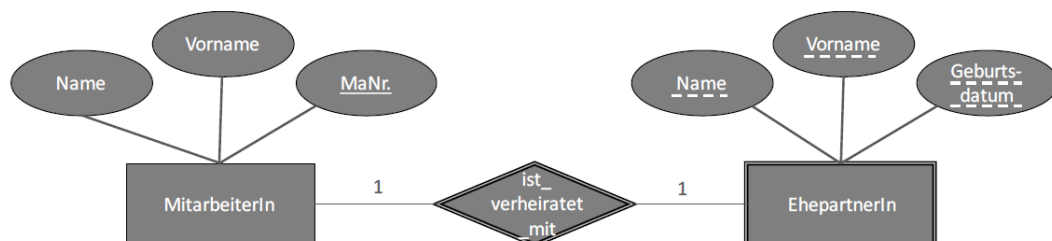
Optionalität: Die Optionalität legt fest, ob eine Entität eines Entitätstyps immer in Beziehung zu einer Entität eines anderen Entitätstyps stehen muss oder lediglich in Beziehung stehen kann (Minimum).

### 3. Grundlagen des konzeptuellen Entwurfs



	1 genau ein		1 optional Ein oder kein		N ein oder mehreren		N optional kein, ein oder mehreren	
1	1	1	1	0...1	1	N	1	0...N
1 optional			0...1	0...1	0...1	N	0...1	0...N
M					M	N	M	0...N
M optional							0...M	0...N

### 3.8 Schwache Entitäten

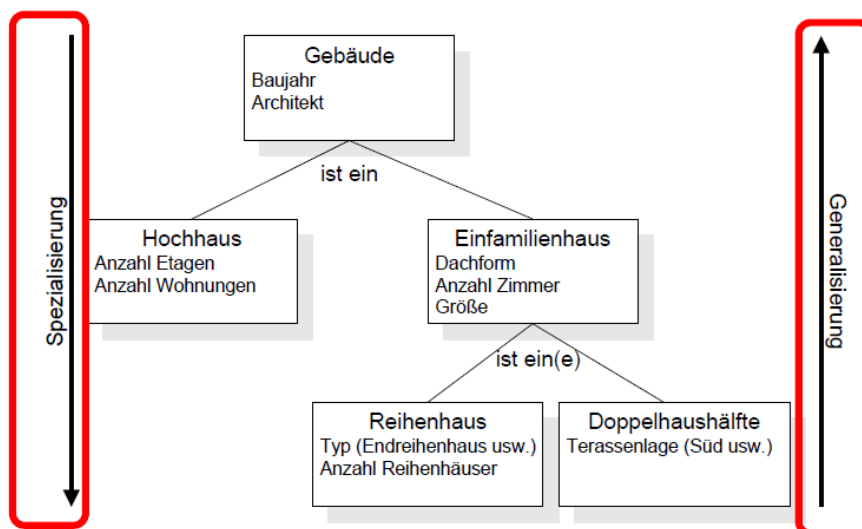


## 3.9 Sub- und Supertypen

Spezialisierung: "Unter einer Spezialisierung versteht man den Prozess der Gewinnung von Subtypen aus einem gegebenen Supertyp. Der abgeleiteten Typen haben dann neben den vom Supertyp ererbten Attributen eigene Attribute, die nur den Subtyp beschreiben."

Generalisierung: "Unter Generalisierung versteht man den Prozess der Gewinnung eines Supertyps aus mehreren ähnlichen Subtypen. Der neue Supertyp wird dann durch diejenigen Attribute beschrieben, die den ähnlichen Subtypen gemeinsam sind."

Hierarchisierung: Die Hierarchisierung beschreibt die Entstehung der gesamten Struktur.



Quelle: siehe auch Elmasri, Navathe (2009), S. 92 ff.

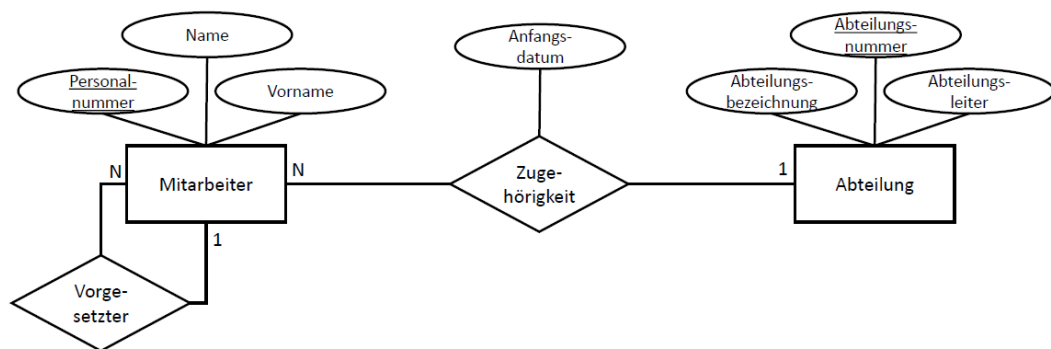


## 4 Entity-Relationship-Modell

1. Entitäten und Beziehungen identifizieren
2. Entitäten und Beziehungen modellieren
3. Super- und Subtypen modellieren
4. N:M-Beziehungen transformieren
5. Attribute hinzufügen
6. Schlüsselattribute definieren

### 4.1 Chen

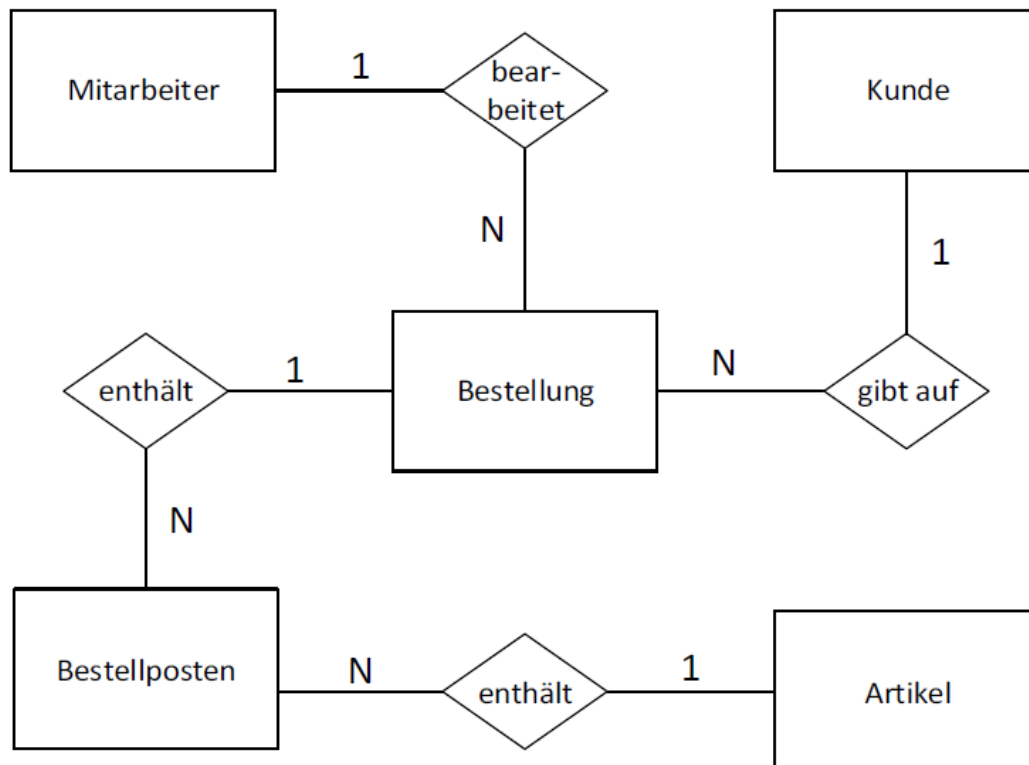
#### 4.1.1 Einführung



##### 4.1.2 Beispiele

Ein Online-Spielzeugwarengeschäft verkauft über das Internet Spielzeug. Ein Kunde gibt mehrere Bestellungen auf. Eine Bestellung enthält mehrere Artikel bzw. jeder Artikel kann auf verschiedenen Bestellungen erscheinen. Die Bestellungen werden von Mitarbeitern bearbeitet, wobei jede Bestellung durch genau einen Mitarbeiter bearbeitet wird. Mitarbeiter bearbeiten jedoch mehrere Bestellungen.

Entwerfen Sie ein Datenmodell nach dem ERM von Chen. Verzichten Sie auf Attribute und Schlüssel.



## 4. Entity-Relationship-Modell

Entwerfen Sie für folgenden Sachverhalt ein ERM nach Chen:

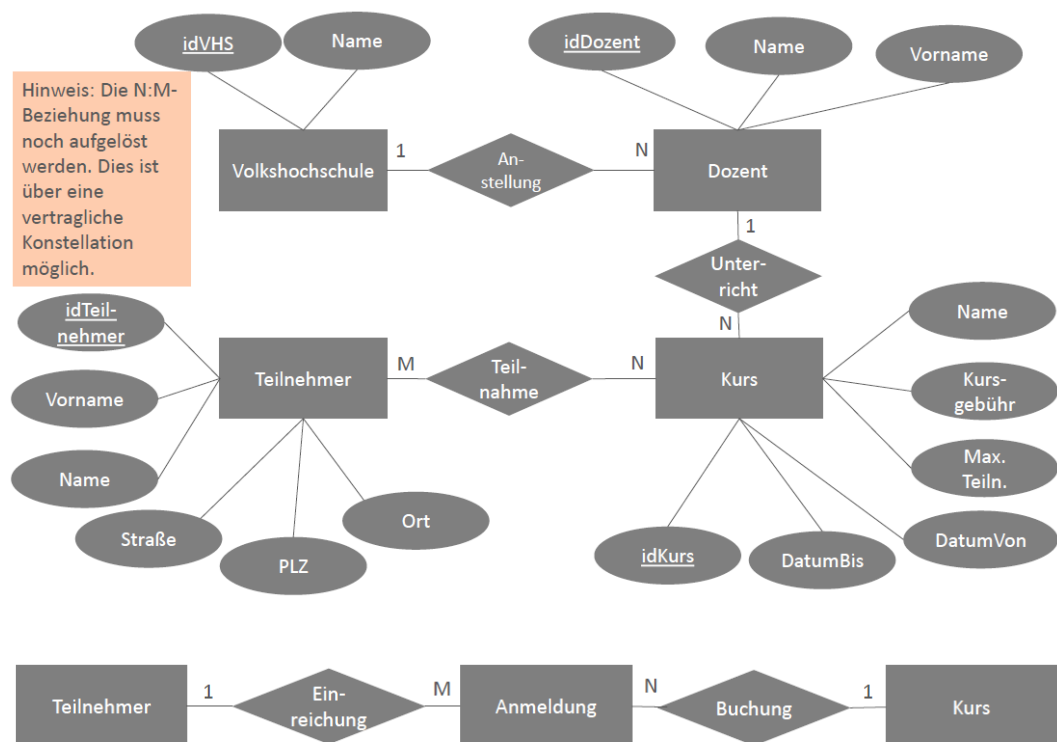
Für einen Verbund von Volkshochschulen soll eine neue Verwaltungssoftware eingeführt werden. An einer Volkshochschule sind mehrere Dozenten beschäftigt. Ein Dozent unterrichtet mehrere Kurse. An den Kursen können mehrere Teilnehmer teilnehmen. Jeder Teilnehmer kann auch mehrere Kurse besuchen.

Jede Volkshochschule verfügt über einen Namen.

Zu den Dozenten werden Name und Vorname abgelegt.

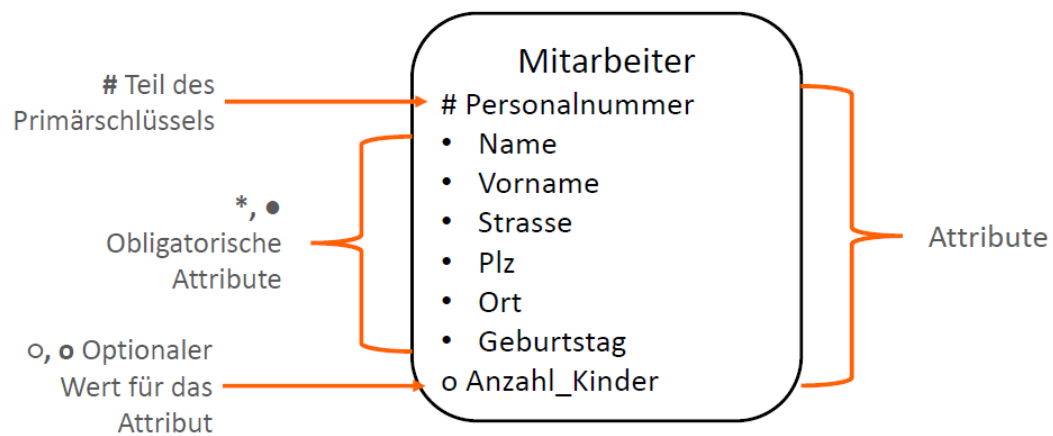
Zu den Kursen sind Name, Kursgebühr, maximale Teilnehmeranzahl, Anfangs- und Enddatum zu speichern.

Von den Teilnehmern sollen Name, Vorname, Straße, PLZ, Ort und Telefon hinterlegt werden.



## 4.2 Barker

1. Entitätstypen und Beziehungen identifizieren
2. Grobentwurf erstellen
3. Super- und Subtypen modellieren
4. N:M-Beziehungen auflösen
5. Attribute ergänzen und charakterisieren (optional oder obligatorisch)
6. Schlüsselattribute festlegen oder hinzufügen



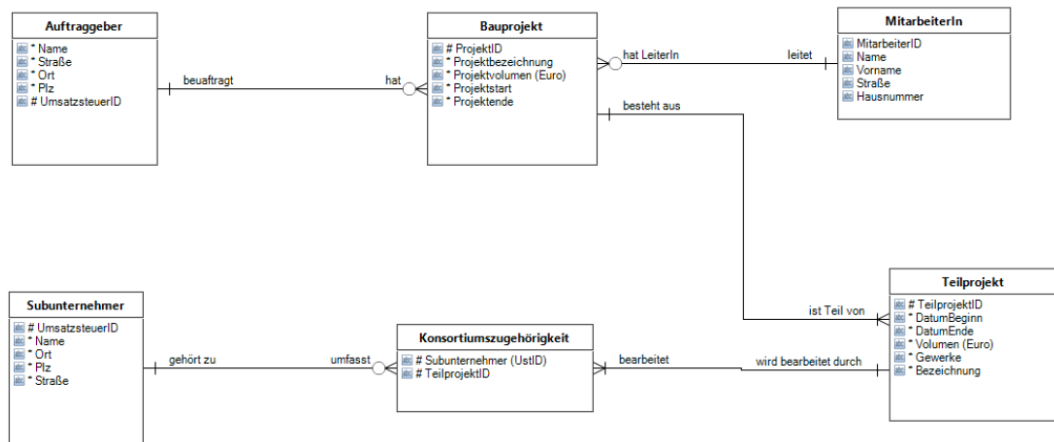
	1 Genau ein	C ein oder kein	M ein oder mehrere	CM kein, ein oder mehrere
1 Genau ein				
C ein oder kein				
M ein oder mehrere				
CM kein, ein oder mehrere				

## 4. Entity-Relationship-Modell

### 4.2.1 Übung

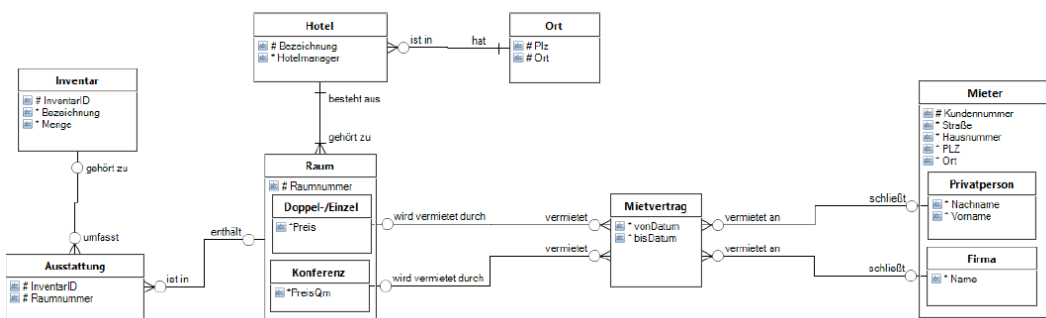
Die Bau GmbH will ein neues System für die Verwaltung von Bauprojekten einführen. Bauprojekte werden durch Auftraggeber beauftragt. Jedes Bauprojekt wird durch einen Mitarbeiter der Bau GmbH geleitet. Die Bau GmbH fungiert als Generalunternehmer und vergibt Unteraufträge an Subunternehmer. Die Teilprojekte, die die Subunternehmer durchführen, sind immer einem Gewerke zuordbar (z.B. Erdarbeiten, Mauerarbeiten, Dachdeckerarbeiten, Zimmererarbeiten). Ein Projekt kann mehrere Teilprojekte des selben Gewerkes umfassen. Kleine Subunternehmen bedienen in der Regel nur ein Gewerke, einige größere Subunternehmen bearbeiten jedoch mehrere Gewerke. Manche Teilprojekte werden auch gemeinschaftlich durch Subunternehmen bearbeitet.

Entwerfen Sie ein Datenmodell nach dem ERM von Barker.



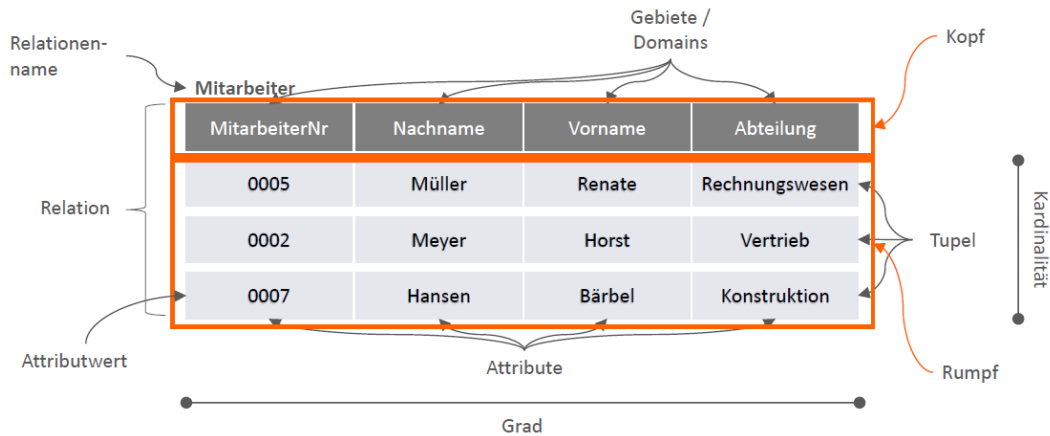
Eine Hotelkette hat mehrere Hotels in verschiedenen Orten. An machen Orten sind mehrere Hotels. Jedes Hotel wird von einem Hotelmanager geleitet. Jedes Hotel hat mehrere Doppel- und Einzelzimmer und verschiedene Konferenzräume. In jedem dieser Räume befindet sich Inventar, wie z.B. Stühle, Betten, Fernseher. Inventar hat eine ID, eine Bezeichnung und eine Anzahl an verfügbaren Inventarstücken. Ein Doppelzimmer kostet pro Übernachtung 220 Euro, ein Einzelzimmer 150 Euro und jeder Konferenzraum pro Quadratmeter 10 Euro. Die Räume werden sowohl an Privatpersonen vermietet, als auch an Firmen, sofern es sich um Konferenzräume handelt.

Entwerfen Sie ein Datenmodell nach der ERM von Barker.



## 5 Relationsmodell

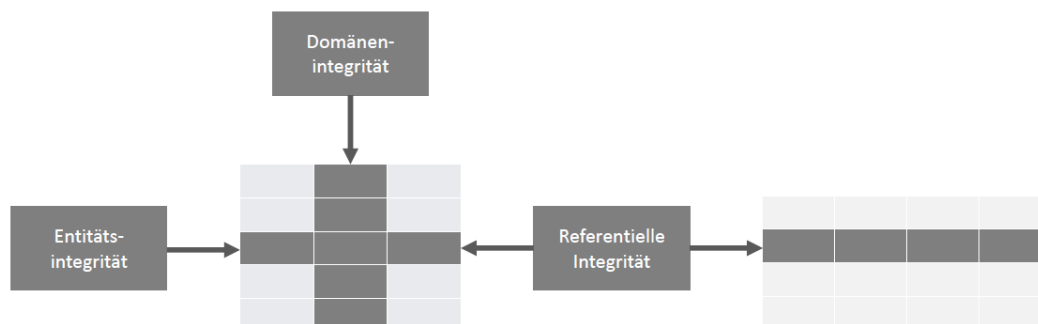
### 5.1 Grundlagen



Relationen weisen vier Eigenschaften auf:

- Es gibt keine doppelten Tupel
- Tupel sind nicht geordnet (etwa von oben nach unten)
- Attribute sind nicht geordnet (etwa von links nach rechts)
- Alle Attribute sind atomar

#### 5.1.1 Integritätsbedingungen



## 6 Logischer Entwurf

Das Relationenmodell kennt nur ein Strukturierungskonzept:

Von Entitäten und Beziehungen...

...zu Relationen

### Vom ERM zum Relationenmodell in mehreren Schritten.

Schritt 1: Für jeden (starken) Entitätstyp im ERM wird eine Relation erzeugt.

Schritt 2: Für jeden schwachen Entitätstyp wird eine Relation erzeugt.

Schritt 3: Für jede 1:1-Beziehung (a) übernimmt den Primärschlüssel einer Tabelle als Fremdschlüssel in die andere Tabelle oder (b) es wird eine gemeinsame Tabelle gebildet.

Schritt 4: Für jeden 1:N-Beziehungstyp bezieht man den Primärschlüssel der 1er-Relation als Fremdschlüssel in die N-er-Relation ein.

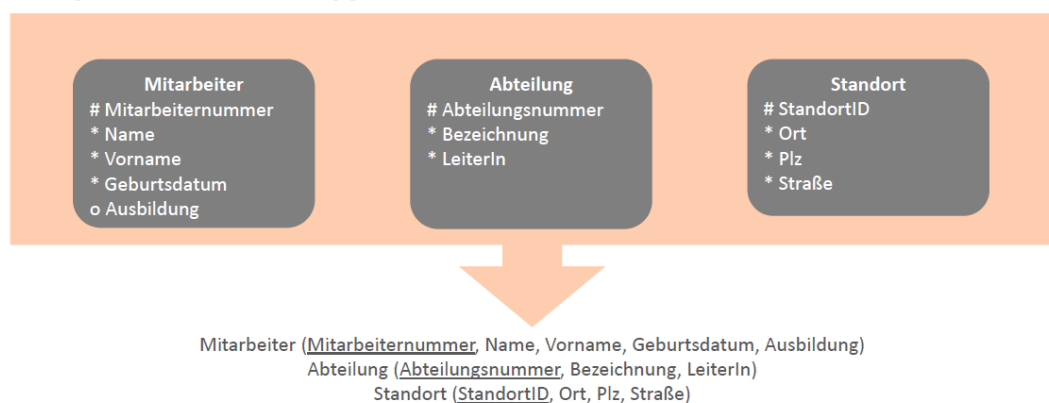
Schritt 5: Für jeden M:N-Beziehungstyp wird eine neue Relation erzeugt (Hinweis: N:M-Beziehungen sollten bereits im ERM aufgelöst sein!).

Schritt 6: Für jedes mehrwertige Attribut wird eine neue Relation erzeugt.

Schritt 7: Beziehungen zwischen Sub- und Superklassen werden in Relationen überführt.

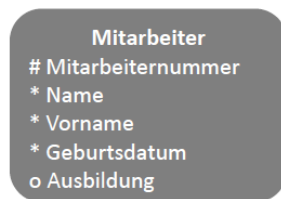
Quelle: Elamsri, Navathe (2009), S. 236 ff., mit Ergänzungen und Anpassungen

### Für jeden Entitätstyp im ERM wird eine Relation erstellt.

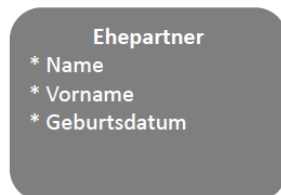


## 6. Logischer Entwurf

Für jede schwache Entität wird eine Relation erzeugt, wobei der Primärschlüssel der zugehörigen starken Entität aufgenommen wird.

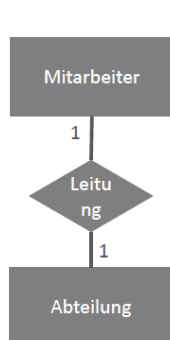


Mitarbeiter(Mitarbeiternummer, Name, Vorname, Geburtsdatum, Ausbildung)



Ehepartner(Name, Vorname, Mitarbeiternummer, Geburtsdatum)

Bei der Umwandlung von 1:1-Beziehungen bestehen drei Optionen.



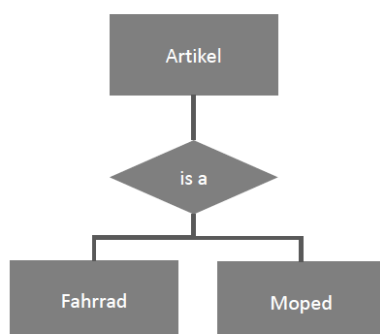
Ausgangssituation

Mitarbeiter(Nummer, Name, Vorname, Ausbildung)

Abteilung(Bezeichnung, Nummer)

1. Option: Es wird eine gemeinsame Tabelle gebildet.  
Mitarbeiter(NummerPersonal, Name, Vorname, Ausbildung, Abteilungsbezeichnung, NummerAbteilung)
2. Option: Der Primärschlüssel von ‚Mitarbeiter‘ wird als Fremdschlüssel in ‚Abteilung‘ übernommen.  
Mitarbeiter(Nummer, Name, Vorname, Ausbildung)  
Abteilung(Bezeichnung, Nummer, NummerPersonal)
3. Option: Der Primärschlüssel von ‚Abteilung‘ wird als Fremdschlüssel in ‚Mitarbeiter‘ übernommen.  
Mitarbeiter(Nummer, Name, Vorname, Ausbildung, NummerAbteilung)  
Abteilung(Bezeichnung, Nummer)

Sub- und Supertypen werden als einzelne Tabellen mit dem Primärschlüssel des Supertyps als Primär-/Fremdschlüssel der Supertypen abgebildet.



Artikel (Artikelnummer, Preis, Bezeichnung)

Fahrrad(AnzahlGänge, Farbe, Größe, Artikelnummer)

Moped(Leistung, Farbe, Hersteller, Artikelnummer)



## 6.1.1 Überblick

### 1:1-Beziehungen

- Eine gemeinsame Tabelle bilden
- Primärschlüssel der einen Tabelle als Fremdschlüssel in die andere Tabelle übernehmen oder umgekehrt
- Sub-/Supertypen i.d.R. eine Supertypentabelle mit allgemeinen Attributen und n Subtypentabellen mit dem Primärschlüssel der Supertypentabelle als Fremd- und Primärschlüssel

### 1:N-Beziehungen

- Primärschlüssel der 1-er-Tabelle als Fremdschlüssel in die n-er-Tabelle

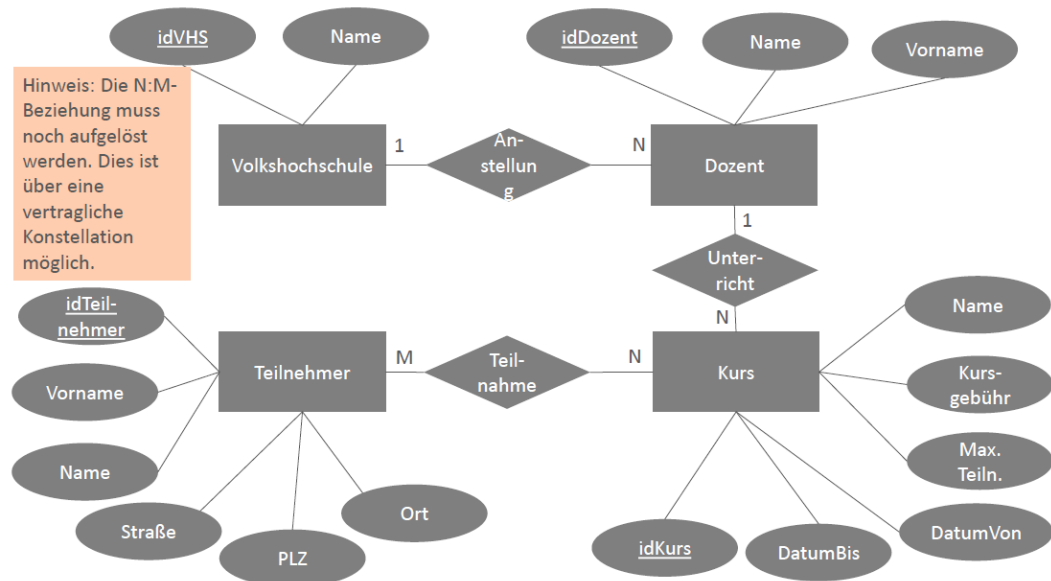
### N:M-Beziehungen

- Neue Tabelle mit 1:N-Beziehungen auf die neue Tabelle. Primärschlüssel der beiden 1-er-Tabellen gehen als Fremdschlüssel in die neue Tabelle. Beide Fremdschlüssel bilden zusammen den Primärschlüssel

### Schwache Beziehungen

- Primärschlüssel + Fremdschlüssel bilden gemeinsamen starken Primärschlüssel

## 6.1.2 Übung



## 6. Logischer Entwurf

### Entitäten überführen

Volkshochschule(idVHS, Name)

Dozent(idDozent, Name, Vorname)

Kurs(idKurs, Name, Kursgebühr, MaxTn, DatumVon, DatumBis)

Teilnehmer(idTeilnehmer, Vorname, Name, Straße, PLZ, Ort)

### 1:N-Beziehungen überführen

Volkshochschule(idVHS, Name)

Dozent(idDozent, Name, Vorname, *idVHS*)

Kurs(idKurs, Name, Kursgebühr, MaxTn, DatumVon, DatumBis, *idDozent*)

Teilnehmer(idTeilnehmer, Vorname, Name, Straße, PLZ, Ort)

### M:N-Beziehungen überführen

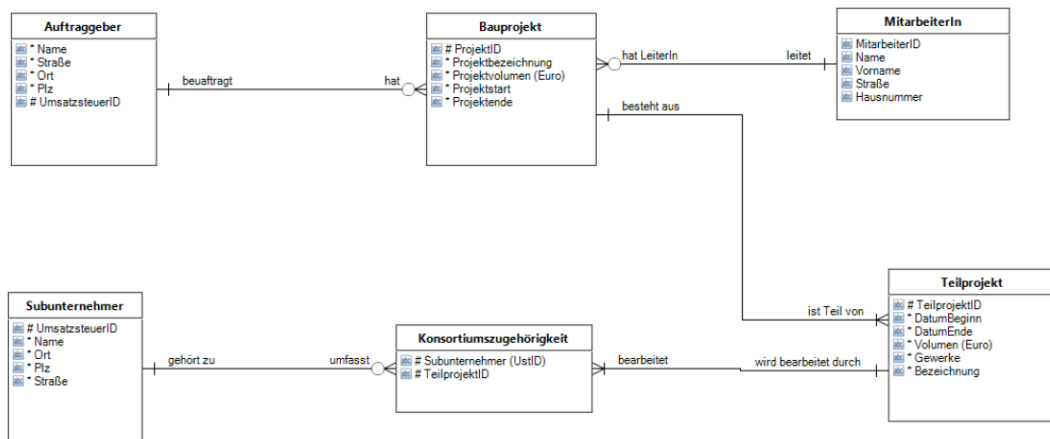
Volkshochschule(idVHS, Name)

Dozent(idDozent, Name, Vorname, *idVHS*)

Kurs(idKurs, Name, Kursgebühr, MaxTn, DatumVon, DatumBis, *idDozent*)

Teilnehmer(idTeilnehmer, Vorname, Name, Straße, PLZ, Ort)

Teilnahme(idTeilnehmer, *idKurs*)



Auftraggeber(UmsatzsteuerID, Name, Straße, Ort, Plz)

Bauprojekt(ProjektID, Projektbezeichnung, Projektvolumen, Projektstart, Projektende, Auftraggeber, LeiterIn)

MitarbeiterIn(MitarbeiterID, Name, Vorname, Straße, Hausnummer)

Teilprojekt(TeilprojektID, DatumBeginn, DatumEnde, Volumen, Gewerke, Bezeichnung, ProjektID)

Subunternehmer(UmsatzsteuerID, Name, Ort, Plz, Straße)

Konsortiumszugehörigkeit(Subunternehmer, TeilprojektID)

## 7 Relationale Entwurfstheorie

### 7.1 Normalformen

#### 1. Normalform



Eine Tabelle darf nur aus Attributen mit **atomaren Werten** bestehen und es darf **keine Wiederholgruppen** geben

#### 2. Normalform

Alle **Nichtschlüssel-Attribute** sind nicht nur von einem Teil, sondern vom gesamten Primärschlüssel **voll funktional abhängig**.

Bezieht sich ausschließlich auf Tabellen mit zusammengesetztem Primärschlüssel.

#### 3. Normalform

Alle Nichtschlüssel-Attribute sind ausschließlich vom Primärschlüssel funktional abhängig, und **nicht transitiv** über ein Nichtschlüssel-Attribut.

**Änderungsanomalie:** Änderungen des gleichen Werts müssen an mehreren Stellen vorgenommen werden.

**Einfügeanomalie:** Vorhandene Daten müssen an anderer Stelle wiederholt eingefügt werden.

**Löschanomalie:** Daten müssen an mehreren Stellen gelöscht werden.

## Getränk

Bezeichnung	Farbe
Kaffee	schwarz
Coca-Cola	schwarz
Mineralwasser	-
Milch	weiss

**Funktionale Abhängigkeit** | Wenn zwei Tupel gleiche Werte für alle Attribute in der Menge von Attributen (Attributsmenge)  $\alpha$  haben, dann müssen auch die Werte ihrer Attribute in der Menge von Attributen (Attributsmenge)  $\beta$  übereinstimmen.

Anders formuliert: Wenn für zwei Tupel gilt  $t_1[\alpha] = t_2[\alpha]$  und gleichzeitig auch  $t_1[\beta] = t_2[\beta]$  gilt, sind diese funktional abhängig.

Es folgt:  $\alpha \rightarrow \beta$

## Getränk

Bezeichnung	Geschmack	Farbe
Kaffee	bitter	schwarz
Coca-Cola	süß	schwarz
Mineralwasser	neutral	-
Milch	neutral	weiss

**Volle funktionale Abhängigkeit** | Eine volle funktionale Abhängigkeit liegt vor, wenn jedes Element in  $\beta$  von der kompletten Menge  $\alpha$ , nicht von einer echten Teilmenge von  $\alpha$ , funktional abhängig ist.

**Partielle funktionale Abhängigkeit** | Eine partielle funktionale Abhängigkeit liegt vor, wenn es ein Element (Attribut) in  $\beta$  gibt, dass nur von einer echten Teilmenge der Attribute aus  $\alpha$  abhängt. Anders formuliert: Die funktionale Abhängigkeit bleibt auch dann erhalten, wenn ein Element aus  $\alpha$  entfernt wird.

## 7. Relationale Entwurfstheorie

Getränk	Geschmack	Farbe	Hersteller	Preis	Herkunftsland	Lieferdauer in Wochen
Kaffee	bitter	schwarz	GutRöst	3,99	Brasilien	6
Coca-Cola	süß	schwarz	SüßeSachen	1,50	Brasilien	6
Mineralwasser	-	-	QuellWell	0,99	Deutschland	1
Milch	neutral	weiß	NaturPur	0,69	Deutschland	1
Kaffee	bitter	schwarz	TeureBohne	4,99	Brasilien	6
Coca-Cola	süß	schwarz	QuellWell	1,99	Brasilien	6

Wir könnten vermuten: Getränke → Lieferdauer

Bei genauer Betrachtung sehen wir: Getränke → Herkunftsland, Herkunftsland → Lieferdauer

Daher haben wir eine transitive funktionale Abhängigkeit: Getränk → Herkunftsland → Lieferdauer

**Transitive funktionale Abhängigkeit** | Wenn A den Wert von B bestimmt und B den Wert von C bestimmt, dann bestimmt auch A den Wert von C

1. Ein Attribut A ist von einem Attribut B **funktional abhängig**, wenn zu jedem Wert von B eindeutig der Wert von A bestimmt werden kann.
2. Von **voller funktionaler** Abhängigkeit spricht man, wenn ein Attribut A von einer Attributkombination B komplett funktional abhängig ist, und nicht nur von einem Teil dieser Attributkombination.
3. **Transitive** Abhängigkeit zwischen zwei Attributen A und C liegt vor, wenn ein Attribut A von einem Attribut B eindeutig bestimmt wird, das Attribut B aber wiederum von einem Attribut C eindeutig bestimmt wird.

## 7.2 Schlüssel

**Schlüssel sind Attribute oder Attributkombinationen, die Objekte eines Objekttyps eindeutig identifizieren.**



- **Eindeutigkeitsbedingung (-einschränkung):** Eine Entität lässt sich eindeutig identifizieren (Elmasri, Navathe (2009), S. 66).
- **Schlüssel:** "Minimale Menge von Attribute, die die zugeordnete Entitäten eindeutig identifizieren. Diese Menge kann sich aus einem oder mehreren Schlüsseln zusammensetzen." (Faeskorn-Woyke 2007, S. 80)
- **Forderung nach Minimalität:** Ein Primärschlüssel soll eine minimale Anzahl von Attributen enthalten. Kein Attribut darf wegfallen, um die Eindeutigkeit aller Entitäten zu gewährleisten.
- **Primärschlüssel:** Ausgewählter Schlüsselkandidat zur eindeutigen Identifizierung eines Objekts. Hinweis: Nicht jedes Entity-Relationship-Model sieht die Existenz von Primärschlüsseln vor (siehe hierzu Elmasri, Navathe (2017), S. 99).
- **Schlüsselkandidat:** Gibt es mehrere Schlüssel, so handelt es sich um Schlüsselkandidaten. Schlüsselkandidaten sind somit Attribute, die eine Entität / ein Objekt eindeutig beschreiben.
- **Zusammengesetzter Schlüssel:** Schlüssel, der aus einer Kombination von Attributen resultiert (basierend auf Elmasri, Navathe (2009), S. 66).
- **Künstliche Schlüssel (auch: Surrogatschlüssel):** Bei einem künstlichen Schlüssel handelt es sich um ein zusätzliches Attribut, dass keine Entsprechung in der realen Welt hat. Dabei handelt es sich in der Regel um einen einfachen Zähler (z.B. Kundennummer, Personalnummer).

### 7.2.1 Eindeutiger Schlüssel

Eine Attributskombination  $L$  wird **eindeutiger Schlüssel** der Relation  $R(A_1, A_2, \dots, A_n)$  genannt, wenn

- 1)  $L \rightarrow (A_1, A_2, \dots, A_n)$  und
- 2)  $(A_1, A_2, \dots, A_n)$  von keiner echten Teilmenge von  $L$  funktional abhängig ist.

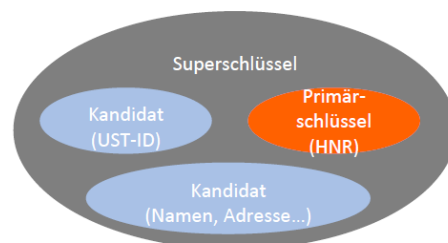
#### Superschlüssel – Schlüsselkandidat – Primärschlüssel.

**Superschlüssel** | Kombination von Attributen in einer Relation, die die Tupel in dieser Relation eindeutig identifizieren.

**Schlüsselkandidat** | Minimale Teilmenge der Attribute eines Superschlüssels, welche die Identifizierung der Tupel ermöglicht.

**Eindeutiger Schlüssel** | Attributskombination einer Relation von denen die übrigen Attribute voll funktional abhängig sind.

**Primärschlüssel** | Ausgewählte Schlüsselkandidat



### 7.3 Zerlegung

**Die Zerlegung von Relationen muss verlustfrei und abhängigkeitsstreu erfolgen.**

**Kriterium der Verlustlosigkeit (auch Wiederherstellbarkeit):**

Die in der ursprünglichen Relationsausprägung einer Relation enthaltenen Informationen müssen aus den Ausprägungen der neuen Relationen rekonstruierbar sein.

**Kriterium der Abhängigkeitserhaltung:**

Die für die ursprüngliche Relation geltenden Abhängigkeiten müssen auf die neuen Relationen übertragbar sein.

## 7.3.1 1. Normalform

Kunde (unnormalisiert)

Kunden-nummer	Name	Vorname	Strasse	Plz	Ort	Kind	Alter
1	Bolte	Bertram	Busweg 12	44444	Kohlscheidt	Katja, Ursula, Karl	5, 8, 12
2	Muster	Hans	Musterweg 12	22222	Karlstadt		
3	Wiegerich	Frieda	Wanderstr. 89	33333	Rettrich	Ursula, Enzo	16, 9

Überführung in 1. Normalform

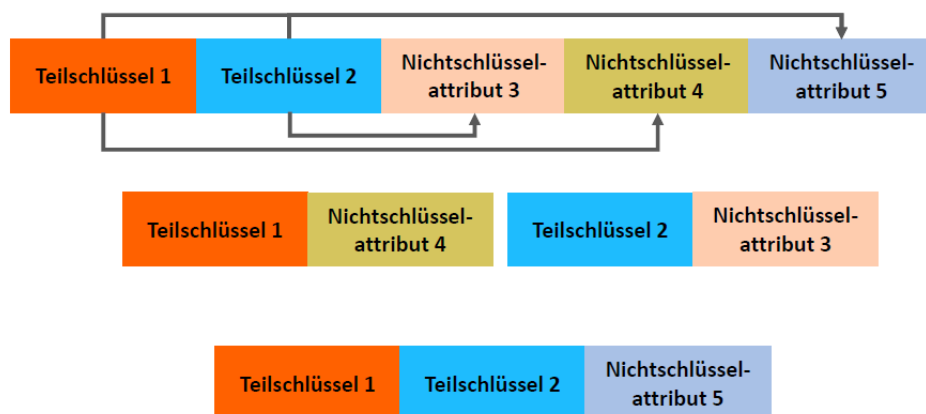
Kunde (1. NF)

Kunden-nummer	Name	Vorname	Strasse	Haus-nummer	Plz	Ort	Kind	Alter
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Katja	5
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Ursula	8
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Karl	12
2	Muster	Hans	Musterweg	12	22222	Karlstadt		
3	Wiegerich	Frieda	Wanderstr.	89	33333	Rettrich	Ursula	16
3	Wiegerich	Frieda	Wanderstr.	89	33333	Rettrich	Enzo	9

Kunde (Kundennummer, Name, Vorname, Strasse, Hausnummer, Plz, Ort, Kind, Alter)

## 7.3.2 2. Normalform

Ist die zweite Normalform nicht gegeben,  
wird diese durch Zerlegung der Relationen erzeugt.



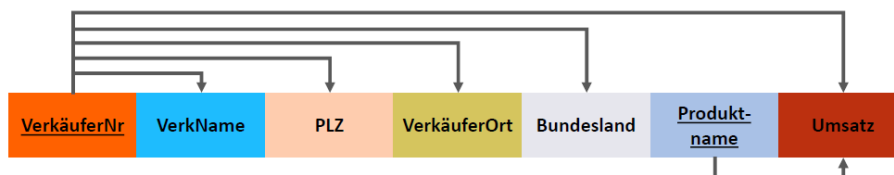
## Relationen bei der Fashion AG müssen aufgeteilt werden.

VerkäuferNr	VerkName	PLZ	VerkäuferOrt	Produktname	Umsatz	Bundesland
V1	Meier	22222	Hamm	Bluse	50	NRW
V1	Meier	22222	Hamm	T-Shirt	20	NRW
V1	Meier	22222	Hamm	Jeans	80	NRW
V2	Schneider	33333	Minden	Pullover	30	NDS
V2	Schneider	33333	Minden	Jeans	80	NDS
V3	Müller	44444	Hannover	Pullover	30	NDS

VerkäuferNr	VerkName	PLZ	VerkäuferOrt	Bundesland
V1	Meier	22222	Hamm	NRW
V2	Schneider	33333	Minden	NDS
V3	Müller	44444	Hannover	NDS

VerkäuferNr	Produktname	Umsatz
V1	Bluse	50
V1	T-Shirt	20
V1	Jeans	80

Ist die zweite Normalform nicht gegeben, wird diese durch Zerlegung der Relationen erzeugt.



VerkäuferNr	VerkName	PLZ	VerkäuferOrt	Bundesland
-------------	----------	-----	--------------	------------

VerkäuferNr	Produktname	Umsatz
-------------	-------------	--------

## RunIT GmbH überführt in die 2. Normalform.

MAVorname	MaNachname	MaNr	ProduktNr	ProduktBez	BearbZeit	Abteilung	AbtNr
Hendrik	Schröder	01	P1	Gamer XPO	8	Heim-PC	0002
Christine	Bauer	02	P2	Gamer XPP	5	Heim-PC	0002
Susanne	Müller	03	P3	DeepLearn 1	24	Industrie-PC	0001
Stefan	Lindner	04	-	-	-	Finanzen	0003
Hanna	Hansen	05	P1	Gamer XPO	6	Heim-PC	0002
Hendrik	Schröder	01	P1	Gamer XPP	6	Heim-PC	0002

MAVorname	MaNachname	MaNr	Abteilung	AbtNr
Hendrik	Schröder	01	Heim-PC	0002
Christine	Bauer	02	Heim-PC	0002
Susanne	Müller	03	Industrie-PC	0001
Stefan	Lindner	04	Finanzen	0003
Hanna	Hansen	05	Heim-PC	0002

ProduktNr	ProduktBez	MaNr	ProduktNr	BearbZeit
P1	Gamer XPO	01	P1	8
P2	Gamer XPP	02	P2	5
P3	DeepLearn 1	03	P3	24
		05	P1	6
		01	P2	6



## 7. Relationale Entwurfstheorie

Kunde (1. NF)

Kunden-nummer	Name	Vorname	Strasse	Haus-nummer	Plz	Ort	Kind	Alter
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Katja	5
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Ursula	8
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt	Karl	12
2	Muster	Hans	Musterweg	12	22222	Karlstadt		
3	Wiegerich	Frieda	Wanderstr.	89	33333	Rettrich	Ursula	16
3	Wiegerich	Frieda	Wanderstr.	89	33333	Rettrich	Enzo	9

Überführung in 2. Normalform

Kunde (2. NF)

Kunden-nummer	Name	Vorname	Strasse	Haus-nummer	Plz	Ort
1	Bolte	Bertram	Busweg	12	44444	Kohlscheidt
2	Muster	Hans	Musterweg	12	22222	Karlstadt
3	Wiegerich	Frieda	Wanderstr.	89	33333	Rettrich

Kind (2. NF)

Kunden-nummer	Kind	Alter
1	Katja	5
1	Ursula	8
1	Karl	12
3	Ursula	16
3	Enzo	9

### 7.3.3 3. Normalform

3. Normalform: Nichtschlüssel-Attribute sind lediglich vom Primärschlüssel funktional abhängig.



VerkäuferNr	VerkName	PLZ	Ort	Bundesland
V1	Meier	22222	Hamm	NRW
V2	Schneider	33333	Minden	NDS
V3	Müller	44444	Hannover	NDS

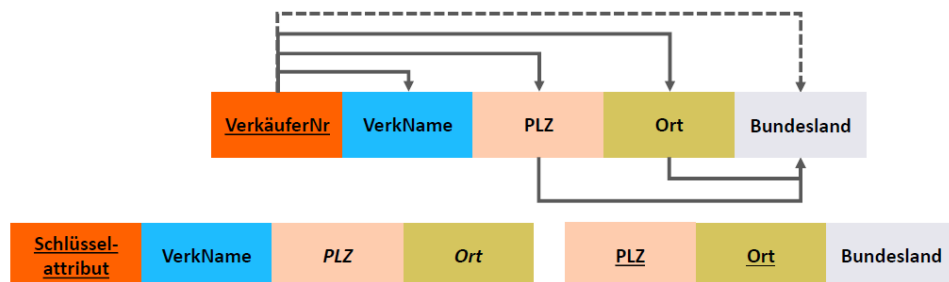


VerkäuferNr	VerkName	PLZ	Ort
V1	Meier	22222	Hamm
V2	Schneider	33333	Minden
V3	Müller	44444	Hannover



PLZ	Ort	Bundesland
22222	Hamm	NRW
33333	Minden	NDS
44444	Hannover	NDS

Transitive Abhängigkeit wird durch Aufspaltung aufgelöst!"



Vorteile:

Ändert sich der Wohnort des Kunden und dieser zieht in ein anderes Bundesland, so muss das Bundesland, sofern der Ort schon angelegt ist, nicht wiederholt geändert werden.

Werden Landesgrenzen verschoben (z.B. Hamburg wird Teil von Schleswig-Holstein) muss die Änderung nur einmalig vorgenommen werden.

Kommt ein neuer Kunde aus dem gleichen Ort dazu, so wird dieser auch automatisch dem richtige Bundesland zugeordnet.

## 7. Relationale Entwurfstheorie

MAVorname	MaNachname	MaNr	Abteilung	AbtNr
Hendrik	Schröder	01	Heim-PC	0002
Christine	Bauer	02	Heim-PC	0002
Susanne	Müller	03	Industrie-PC	0001
Stefan	Lindner	04	Finanzen	0003
Hanna	Hansen	05	Heim-PC	0002



MAVorname	MaNachname	MaNr	AbtNr
Hendrik	Schröder	01	0002
Christine	Bauer	02	0002
Susanne	Müller	03	0001
Stefan	Lindner	04	0003
Hanna	Hansen	05	0002



Abteilung	AbtNr
Industrie-PC	0001
Finanzen	0003
Heim-PC	0002