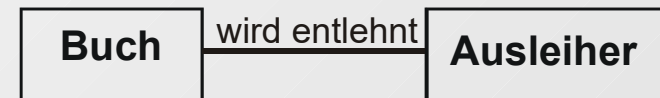
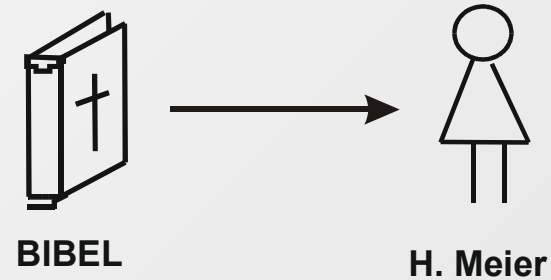
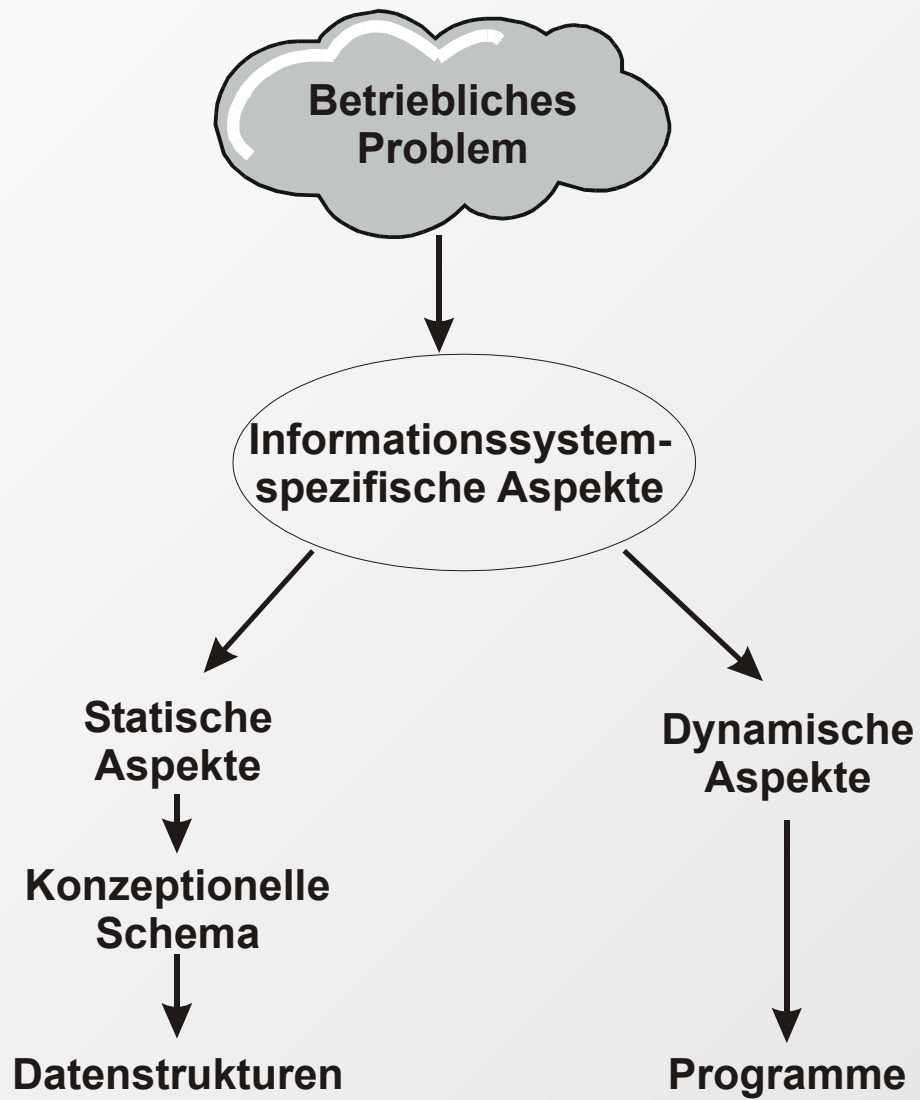


Der DB-Entwurf

Foliensatz 2

DI(FH) Gerald Aistleitner, 2015/16



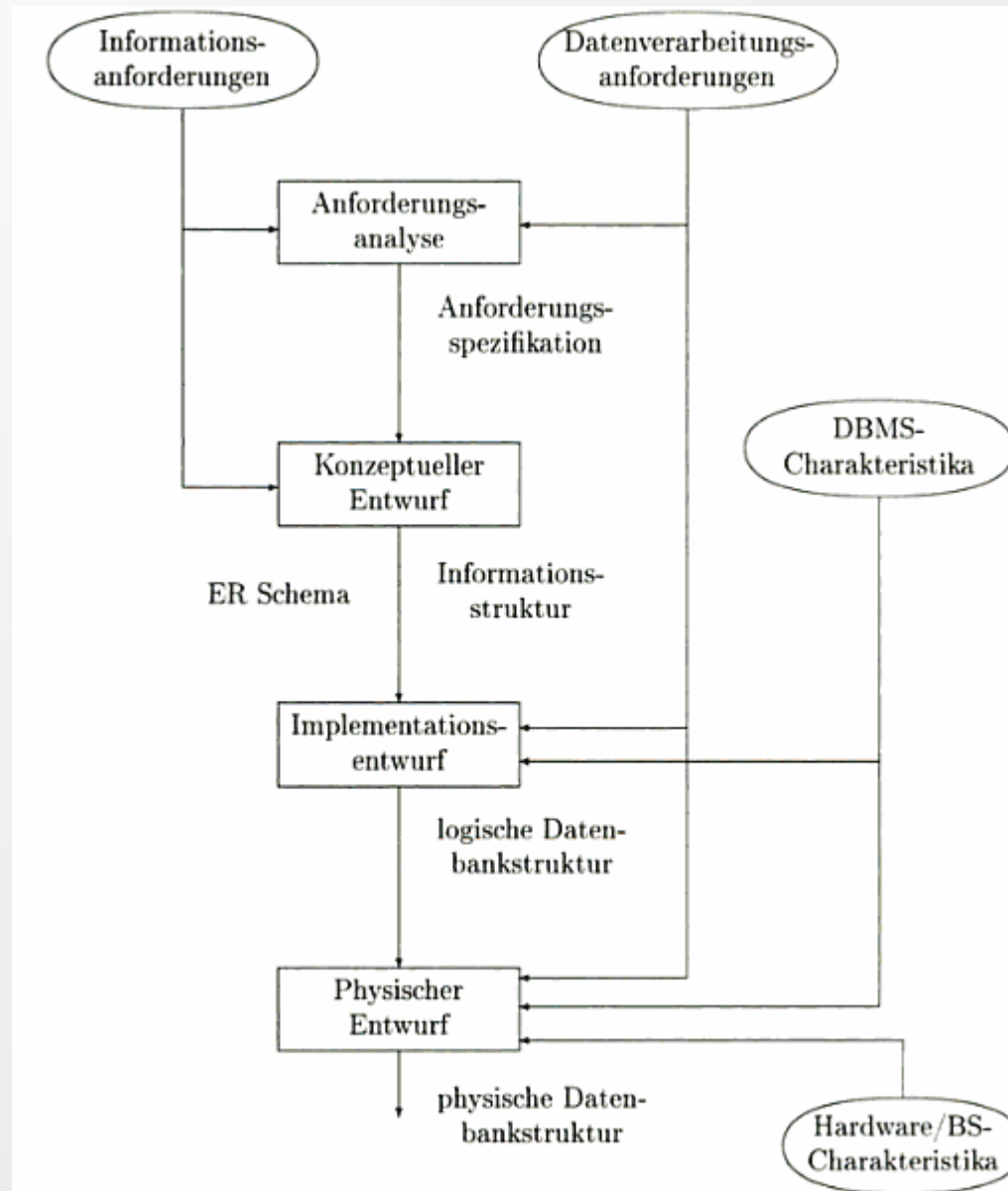


Abb. 2.2: Die Phasen des Datenbankentwurfs

Einleitendes Beispiel

Problemstellung

In einem Unternehmen werden Informationen über Mitarbeiter und über die Projekte, die in Bearbeitung stehen, gespeichert.

Beschreibung dieser realen Situation in Tabellenform:

Personal-nummer	Name	Ort	Abteilung	Abteilungs-name	Projektnr	Projektname	Zeit
100	Susi	Linz	20	Org	23c	Kontobuchungen	40
101	Sabine	Wels	20	Org	23y	OP Liste	50
					30a	Umrechnung	5
102	Franz	Steyr	21	SW Entw.	23x	Saldenliste	30
					30a	Umrechnung	10
103	Otto	Linz	22	Test	23x	Saldenliste	10
					23y	OP Liste	10
					30a	Umrechnung	10

Mögliche Probleme ??

Einleitendes Beispiel

Probleme

- Mehrfache Speicherung der Zuordnung
Abteilungsnummer → Abteilungsname
- Mehrfache Speicherung der Zuordnung
Projektnummer → Projektname
- Neuaufnahme eines Datensatzes
(z.B. neue Abteilung)
- Löschen eines Datensatzes
(z.B. Löschen des Datensatzes mit Pers.Nr 100 →
Verlust der Zuordnung 23c zu Kontobuchungen)

Einleitendes Beispiel

Lösung

Unterteilung der Information in mehrere Tabellen

Projekt#	Projektname
23c	Kontobuchungen
23y	OP Liste
23x	Saldenliste
30a	Umrechnung

Pers#	Name	Ort	Abt#
100	Susi	Linz	20
101	Sabine	Wels	20
102	Franz	Steyr	21
103	Otto	Linz	22

Pers#	Projekt#	Zeit
100	23c	40
101	23y	50
101	30a	5
102	23x	30
102	30a	10
103	23x	10
103	23y	10
103	30a	10

Allgemeines

- Erstellung einer korrekten, vollständigen und konsistenten Repräsentation der realen Gegebenheiten
- Später soll davon ausgehend eine SW-Implementierung durchgeführt werden können.
- Mittels Sprache schwer zu beschreiben → Formale Hilfsmittel benötigt
- Grundsätzlich können alle Arten von Informationen verwaltet / strukturiert werden.
- Erst durch Kombination der Daten mit ihrer Semantik entsteht Information, die strukturiert wird.

Entity Relationship Diagramme (ERD) / Entität

Eine **Entität** ist ein individuelles und *identifizierbares* Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt.

Sofern eine Beziehung zwischen Entitäten eine Bedeutung hat, kann auch ein individuelles Exemplar einer solchen Beziehung als Entität aufgefasst werden.

Beispiele:

- Max Muster
- Die Ausgabe des „Standard“ vom 15.10.2015
- Ein Buch mit der ISBN 123456789
- Die Ehe A-B

ERD / Entitätsmenge

Modellierung erfolgt nicht auf Basis von Entitäten (wäre vollständig nicht möglich)!

→ **Gruppierung** von Entitäten mit gleichen oder ähnlichen Merkmalen, aber unterschiedlichen Merkmalswerten, zu **Entitätsmengen**.

Beispiele:

- Angestellte der Firma XY
- Einzelnummern des Standards
- Bücher in der Schulbibliothek
- Eheschlüsse auf dem Standesamt YZ

Darstellung im ERD mittels Rechtecke:



Kunde

ERD / Entitäten, Entitätstypen, Entitätsmengen

- **Disjunkte Entitätsmenge**
keine Entität kommt in mehr als einer Entitätsmenge vor
- **Überlappende Entitätsmenge**
gewisse Entitäten gehören gleichzeitig zu mehr als einer Entitätsmenge (Bsp. STUDENTEN, DOKTORANDEN, ASSISTENTEN)

ERD / Entitäten, Entitätstypen, Entitätsmengen

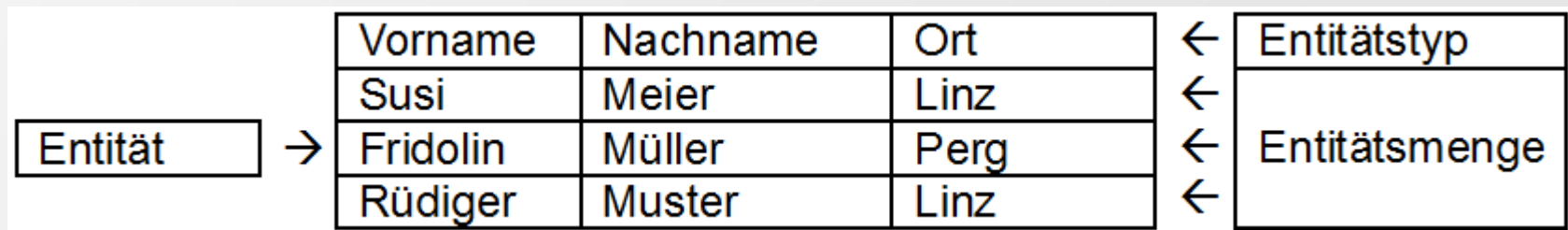
- **Fundamentale Entitäten**
(Kernentität, unabhängige E., Regular Entity)
stellen einen Sachverhalt unabhängig von anderen Entitätsmengen dar
- **Abhängige Entitäten** (Weak Entity)
enthalten nur dann sinnvolle Datenwerte, wenn in der entsprechenden Kernentität die benötigten Daten enthalten sind (Bsp. STADT → STRASSE)

ERD / Entitäten, Entitätstypen, Entitätsmengen

- **Entitätstyp**

beschreibt eine Entitätsmenge, deren Elemente idente Attribute besitzen. Jeder Entitätstyp wird durch seinen Namen und seine Attribute beschrieben.

Sämtliche Entitäten eines bestimmten Entitätstyp werden als Entitätsmenge bezeichnet.



ERD / Beziehungen (Assoziationen)

Eine Assoziation (EM1, EM2) legt fest, wie viele Entitäten aus EM2 einer Entität aus EM1 zugeordnet sein können.

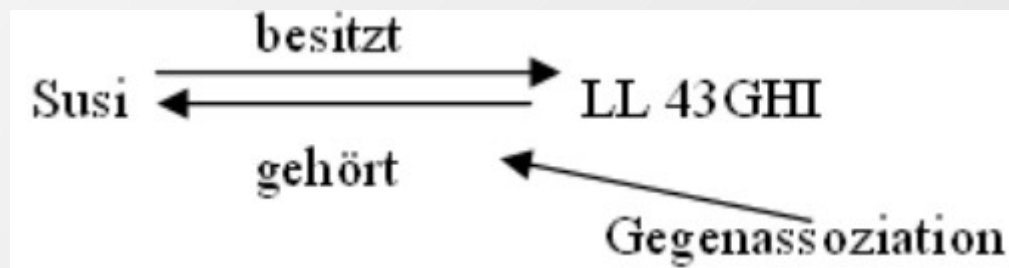
(Ober- und Untergrenzen, da vielfach variabel)

4 Typen von Assoziationen beim DB-Entwurf:

Assoziationstyp (EM1, EM2)	Entitäten aus EM2, die der Menge EM1 zugeordnet sind
1: einfache Assoziation	genau eine (1)
c: konditionelle Assoziation	keine oder eine (0, 1)
m: multiple Assoziation	mindestens eine (≥ 1)
mc: multiple-konditionelle Assoziation	keine, eine oder mehrere (≥ 0)

ERD / Beziehungen (Assoziationen)

Kombiniert man eine Assoziation (EM1, EM2) mit ihrer Gegen-Assoziation (EM2, EM1), so ergibt sich eine Beziehung (relationship) zwischen den beiden betrachteten Entitätsmengen.



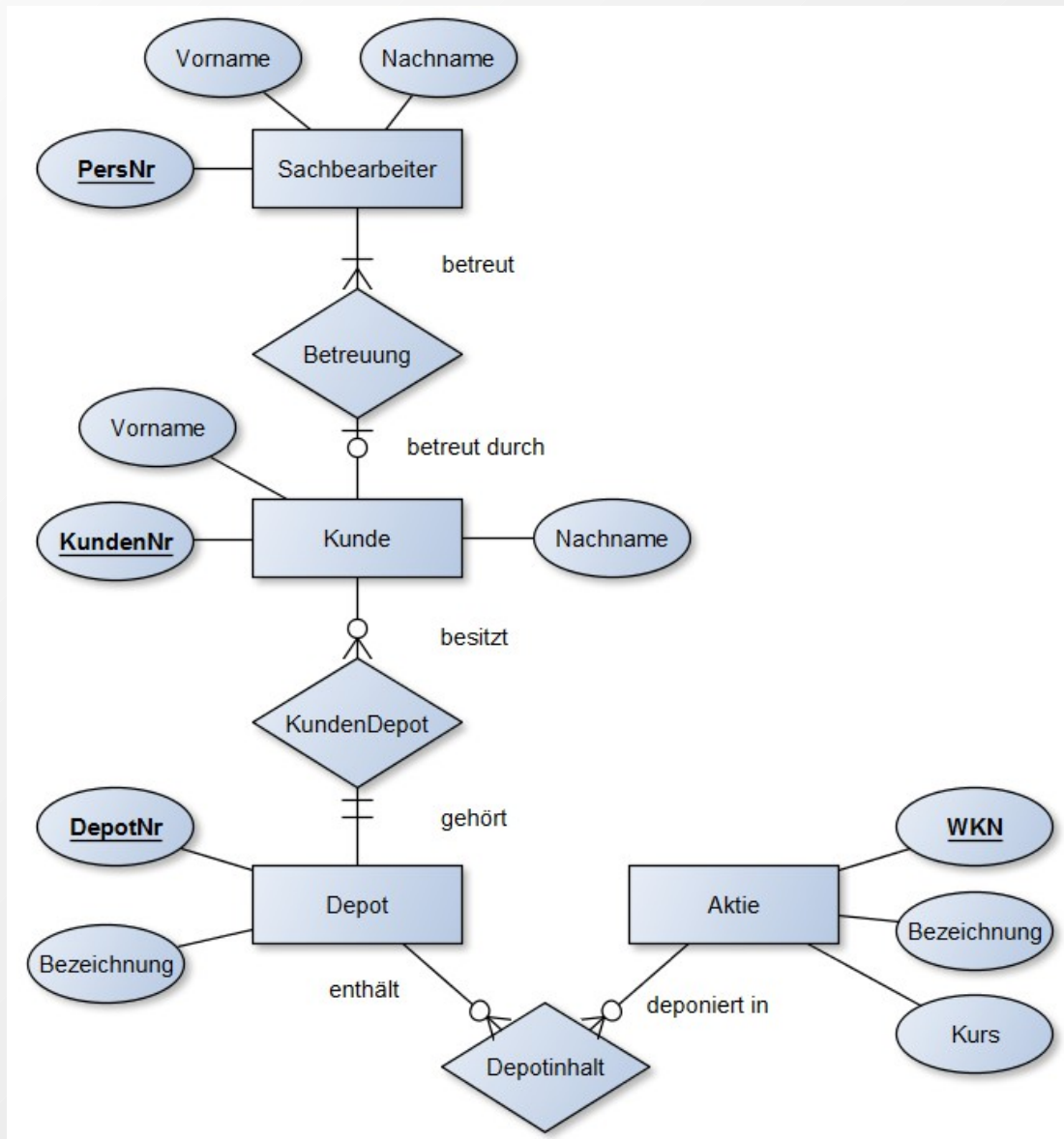
ERD / Beziehungen (Assoziationen)

Bei 4 Assoziationstypen ergeben sich 16 mögliche Arten der Beziehung zwischen 2 Entitätsmengen, von denen 10 verschieden sind (ohne Symmetrien):

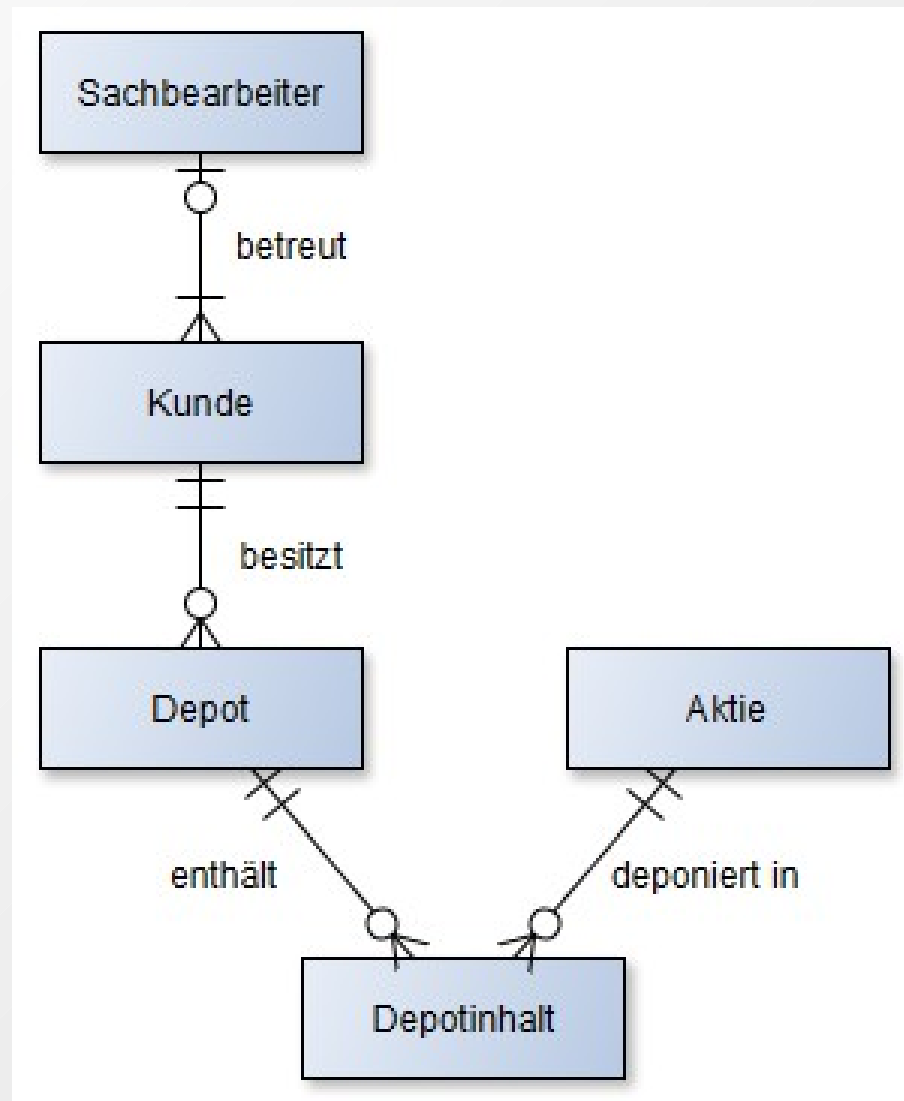
Entitätsmenge 1	Entitätsmenge 2	Beziehungstyp	Beziehung
Ehefrau	Ehemann	1 - 1	Hochzeit
Abteilungen	Personal	c - 1	Abteilungsleiter
Personal	Abteilungen	m - 1	Abteilungszugehörigkeit
Kinder	Ehepaare	mc - 1	Familienzugehörigkeit
Frauen	Männer	c - c	Heirat
Personen	Parteien	m - c	Parteizugehörigkeit
Angestellte	Angestellte	mc - c	ist Vorgesetzter
Standorte	Standorte	m - m	Distanz
Vorlesungen	Studenten	mc - m	Einschreibung
Personen	Personen	mc - mc	Freundschaften

1	c	m	mc	
1:1	1:c	1:m	1:mc	1
	c:c	c:m	c:mc	c
		m:m	m:mc	m
			mc:mc	mc

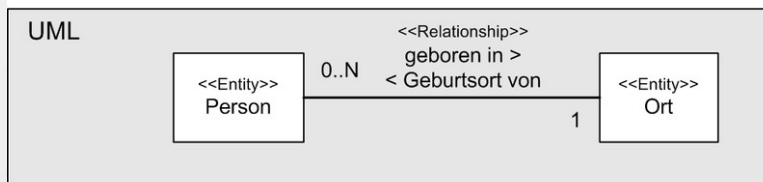
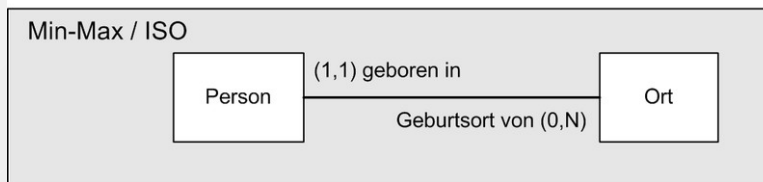
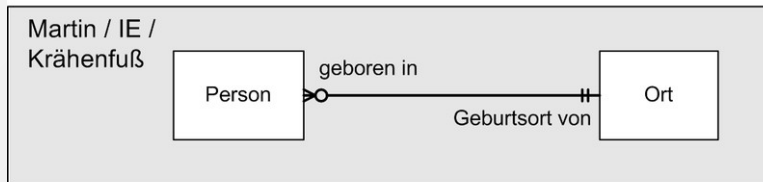
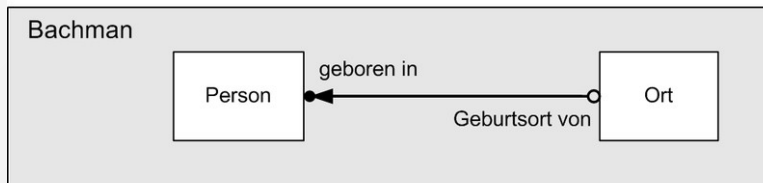
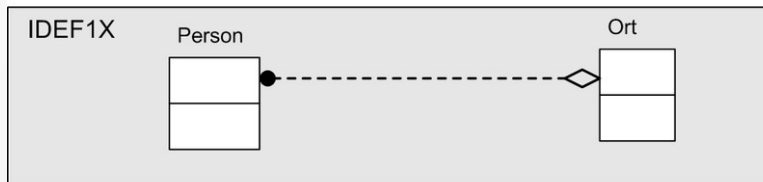
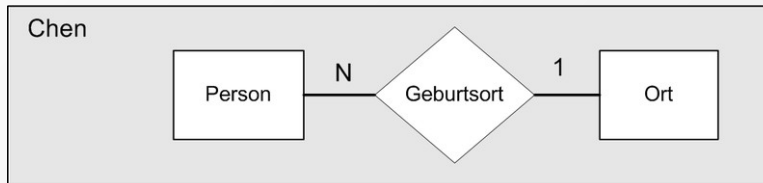
ERD - Beispiel Wertpapiere



ERD – Bsp. Wertpapiere – Erw. Relationenmodell



ERD- Übersicht über Notationen

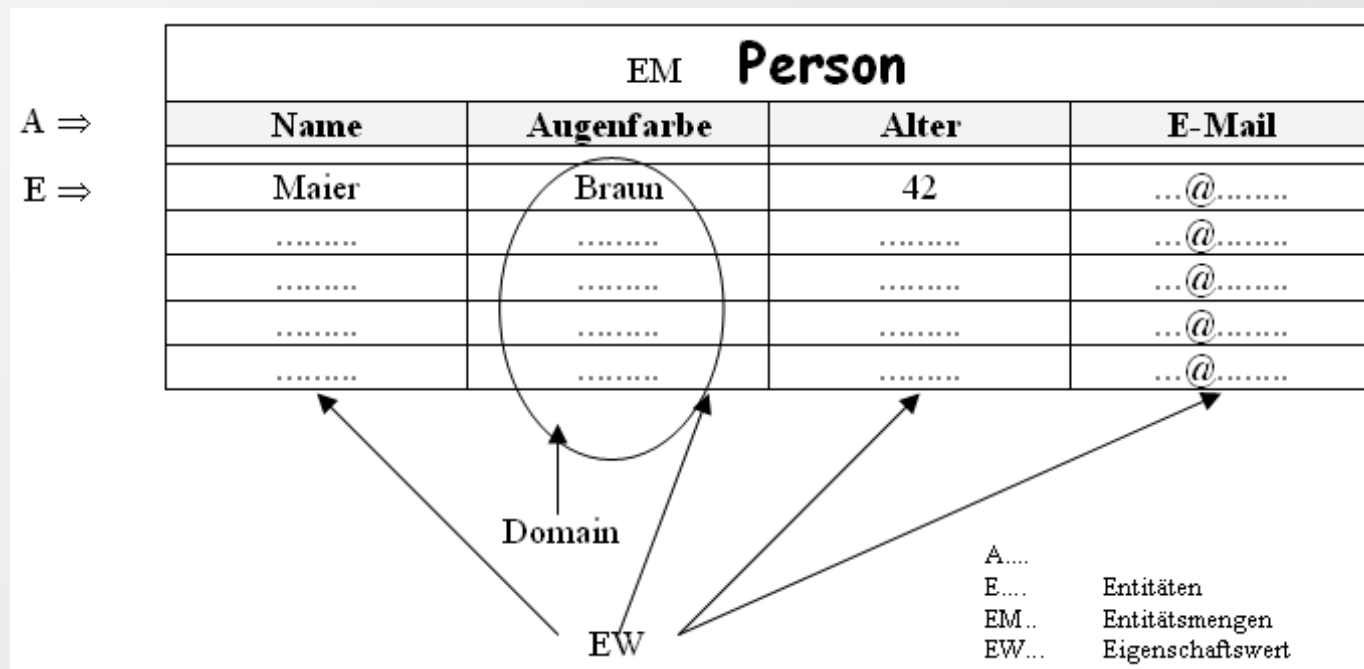


ERD / Wertebereich, Attribute, Formatierung

- **Wertebereich** = Domain
ist eine Menge von verschiedenen Datenwerten des selben skalaren Datentyps
- Beispiele:
 - ganze Zahlen (Integer)
 - Zeichen des ASCII-Zeichensatzes
 - 'Mo', 'Di', 'Mi',
 - 1..23
 - Zeichenkette der Länge 10

ERD / Wertebereich, Attribute, Formatierung

- **Attribut**
beschreibt eine Eigenschaft der Entitäten einer Entitätsmenge mit Eigenschaftswerten.
- Eigenschaftswerte können nur bestimmte Werte annehmen, die in einem Wertebereich definiert sind.



ERD / Wertebereich, Attribute, Formatierung




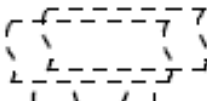









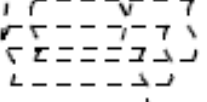

- **Formatierte Beschreibung**
Beschreibung eines Attributs erfolgt durch die Angabe einer oder mehrerer erlaubter Ausprägungen aus dem zulässigen Wertebereich dieses Attributes.
- **Beispiele:**
 - Lohnklasse: 17
 - Ruhetag: 'Di'
 - Kinder: 'Max', 'Susi'
- **Unformatierte Beschreibung**
wird als Text durchgeführt
- **ZB besondere Merkmale:** 'Narbe über dem re. Auge'

ERD / Namensgebung

- Innerhalb eines Modells / firmenintern eindeutig
- Namensgebung von Entitäten:
 - stets im Singular (je nach Quelle)
 - Abkürzungen vermeiden
 - Begriffe der Geschäftswelt verwenden
 - keine EDV Fachausdrücke (Zielgrp.=Fachbereich)
- Namensgebung von Beziehungen:
 - Verben
 - in beiden Richtungen Namen vergeben

ERD / Namensgebung

- Vermeiden von Synonymen, Homonymen und Äquipollenzen

	Synonym	Homonym	Äquipollenzen	Vagheiten	falsche Bezeichner
Intension (Inhalt der Begriffe)					
Zeichenebene (Bezeichner)					
Extension (Umfang der Begriffe)					
Beispiel	Firma, Unternehmen, Unternehmung	Wertschöpfungs- kosten	Zweckauf- wand, Grundkosten	SGF, SGE	relativer Marktanteil
Aktion	Kontrollieren	Beseitigen	Aufdecken	Klären	Ersetzen

ERD / Definitionen im Modell

- Um Kommunikationsschwierigkeiten zu vermeiden ist eine exakte Definition aller Entitäten sinnvoll
→ Kommunikation IT ↔ Fachbereich
- Beispiel:
Kunde: **jeder**, der **etwas** von uns kauft
 - jeder: kann ein Mitarbeiter Kunde sein?
 - etwas: Verkauf einer Lagerhalle → neuer Kunde?
 - kauft: Wann verkauft? (Bezahlung, Lieferung, ...)
 - wann: sind Interessenten schon Kunden?

ERD / Definitionen im Modell / Kriterien

- Klarheit und Kürze

Negativbeispiel:

Kunden sind eine Gruppe, bestehend aus ein oder mehreren Personen, die nach außen als eine Unternehmung auftreten und mit der wir in Geschäftsbeziehung stehen, in dem wir ihnen Produkte verkaufen, die sie benötigen, um eine oder mehrere ihrer Geschäftsfunktionen erfüllen zu können.

- Vollständigkeit

- Präzision

- Konsistenz

Negativbeispiel:

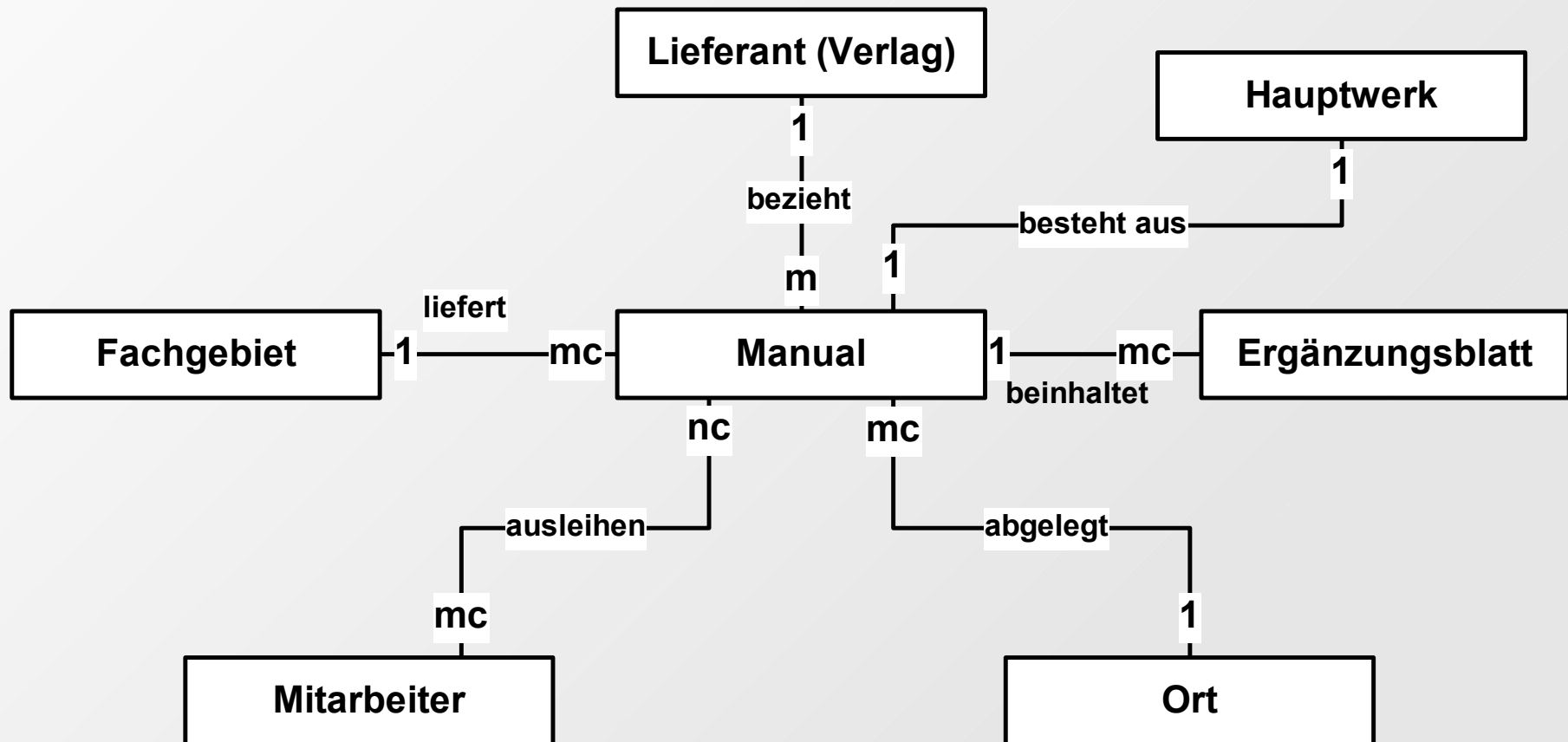
Kunde = Organisation, die unsere Produkte kauft

Wiederverkäufer = Organisation, die von uns Produkte einkauft, um diese an den **Kunden** weiterzuverkaufen

Übung – ERD Diagramm für techn. Manual

- Ein Manual besteht aus dem Hauptwerk und evtl. aus weiteren Ergänzungsblättern
- Ein Manual bezieht sich auf ein bestimmtes Fachgebiet
- Jedes Manual ist samt seinen Ergänzungsblättern an einem bestimmten Ort abgelegt
- Ein Manual kann samt Ergänzungsblättern von den Mitarbeitern der Firma ausgeliehen werden.
- Ein Manual samt Ergänzungsblättern wird von einem bestimmten Lieferanten geliefert

Übung - ERD Diagramm für techn. Manual



Das relationale Modell

- Überleitung von der abstrakten Ebene der Entitätsmengen auf die „datennähere“ Ebene der Relationen
- Jede Entitätsmenge wird durch eine Relation und darin durch eine Gruppe ihrer Attribute dargestellt

Prinzipien:

- Unterscheidung zw. Logischen / physischen Aspekten
- Strukturelle Einfachheit (erleichtert Kommunikation)
- Mengenorientiertes Verarbeiten

Das relationale Modell – Relationales Schema R

Entitätsmenge:

PERSON

Relation(enschema)

Attribute A	→	Person	←	Relationenname
		PersNr	←	Relationenschema R
		Name		
		Plz		
		Ort		
Tupel t	→	27	←	Relation r
		Mair		
		4600		
		Wels		
		17		
		Ana		
		4020		
		Linz		
		45		
		Zopf		
		4020		
		Linz		

Ist eine endliche Menge von Attributen $A_1, A_2, A_3, \dots, A_n$ gegeben, so nennen wir sie ein relationales Schema R , das aus all diesen Attributen besteht.

Das heißt $R = \{A_1, A_2, A_3, \dots, A_n\}$.

Das relationale Modell – Domäne D

Entitätsmenge:

PERSON

Relation(enschema)

Attribute A	→	Person	←	Relationenname
		PersNr	←	Relationenschema R
		Name		
		Plz		
		Ort		
Tupel t	→	27	←	Relation r
		17		
		Ana		
		4020		
		Linz		
		45		
		Zopf		
		4020		
		Linz		

Mit jedem dieser Attribute verbunden ist eine nichtleere Menge D_i ($1 \leq i \leq n$), die man den Wertebereich des Attributs A_i nennt und als $\text{Dom}(A_i)$ bezeichnet.

Sei D eine neue Menge, die als Vereinigung der Wertebereiche (Definitionsbereiche) aller Attribute definiert ist. Mit anderen Worten gilt

$$D = D_1 \cup D_2 \cup \dots \cup D_n.$$

Das relationale Modell – Relation r / Tupel t

Entitätsmenge:

PERSON

Relation(enschema)

Attribute A	→	Person	←	Relationenname
		PersNr	←	Relationenschema R
		Name		
		Plz		
		Ort		
Tupel t	→	27	←	Relation r
		Mair		
		4600		
		Wels		
		17		
		Ana		
		4020		
		Linz		
		45		
		Zopf		
		4020		
		Linz		

Wir definieren eine Relation r des relationalen Schemas R als eine endliche Menge von Zuordnungen (t_1, t_2, \dots, t_k) von R in D . Die einzelnen Zuordnungen t_i nennt man Tupel oder n -Tupel.

Für jedes dieser Tupel muss der Wert in einer bestimmten Zeile A_i , bezeichnet als $t(A_i)$, ein Element des Wertebereichs von A_i sein.

Das relationale Modell - Relation

Neben der vorher angeführten Definition der Zuordnung von Tupeln zu Relationen kann man Relationen ganz allgemein definieren als:

Ausgehend von n Wertebereichen D_i , die nicht notwendigerweise disjunkt sein müssen (d.h. dieselben Werte dürfen in mehreren Wertebereichen vorkommen), ist eine **Relation R als Teilmenge eines kartesischen Produkts** definiert:

$$R \subset D_1 \times D_2 \times \dots \times D_n$$

→ Teilmenge: keine Doubletten!

→ jede Zeile einer Relation bildet ein solches Tupel

Das relationale Modell - Schlüsselkonzept

- Schlüssel ist fundamentales Konzept des relationalen Modells
- Mechanismus zur Suche nach Tupeln in der Tabelle
- Per Definition sind alle Elemente einer Menge verschieden, d.h. alle Tupel einer Relation müssen sich in ihren Werten unterscheiden!
- Es existieren normalerweise **Untermengen von Attributen**, die diese Bedingung erfüllen und in keinen zwei Tupeln dieselbe Kombination von Werten aufweisen.

Das relationale Modell - Schlüsselkonzept

- Für solche Attribut-Untermengen (bezeichnet als SK) gilt für jegliche zwei Tupel t_1 und t_2 :
$$t_1(SK) \neq t_2(SK)$$
- Solche Attribut-Untermengen SK werden als Superschlüssel (superkey) eines Relationenschemas R bezeichnet, die aus einem oder mehreren Attributen zusammengesetzt sind und deren Wertekombination den eindeutigen Zugriff auf einer Zeile der Tabelle ermöglicht.
- Ein Superschlüssel kann jedoch redundante Attribute aufweisen!
=> besser brauchbar ist daher ein Schlüssel (key)

Das relationale Modell - Schlüsselkonzept

- Ein Schlüssel K eines Relationenschemas R ist ein Superschlüssel von R mit der zusätzlichen Eigenschaft, dass bei Weglassen eines Attributes die verbleibende Attributmenge K' kein Superschlüssel mehr ist.
- Erfüllt also folgende Bedingungen:
 - Eindeutigkeitsbedingung
 - Minimalitätsbedingung
- Beispiel:
Relation Schüler = {SozVersNr, Name, Alter}
{SozVersNr, Name, Alter} → Superschlüssel, kein Key
{SozVersNr} → Superschlüssel und Schlüssel

Das relationale Modell - Schlüsselkonzept

- Jede Tabelle kann mehrere Schlüssel besitzen, die als **Schlüsselkandidaten** (candidate key) bezeichnet werden.
- Ein **Primärschlüssel** (primary key) ist ein bestimmter, willkürlich festgelegter Schlüssel, der aus den vorhanden Schlüsselkandidaten ausgewählt wird.
- Jene Schlüsselkandidaten, die nicht als Primärschlüssel definiert werden, nennt man **Alternativschlüssel** (alternate keys).
- Primärschlüsselattribute dürfen niemals NULL sein!

Das relationale Modell - Schlüsselkonzept

- Eine Menge von Attributen, die in einer anderen Tabelle den Primärschlüssel bildet, wird als Fremdschlüssel (foreign key) bezeichnet.
- Dabei müssen folgende Bedingungen erfüllt sein:
 - Ein Fremdschlüssel und der entspr. Primärschlüssel müssen über den gleichen Wertebereich (Domain) definiert sein
 - Die Fremdschlüssel-Werte müssen den korrespondierenden Primärschlüsselwerten entsprechen oder können NULL als Wert aufweisen
- Fremdschlüssel können zugleich auch Teil eines Primärschlüssels sein (zB KLASSE(KlassenID,...) und SCHÜLER(KlassenID,KatalogNr,Name,...))

Überleitung eines ERDs in ein relationales Modell

1) Jede Entitätsmenge wird als Tabelle mit einem Primärschlüssel dargestellt

Strukturregel 1 (SR1): Bei der Darstellung von Entitätsmengen durch Relationen muss für jede Relation ein Primärschlüssel existieren.

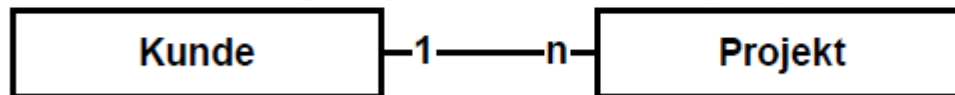
z.B.

EM Mitarbeiter	→	Mitarbeiter (<u>PersNr</u> , ...)
EM Projekt	→	Projekt (<u>ProjektId</u> , ...)
EM Kunde	→	Kunde (<u>KundenNr</u> , ...)

2) 1:n Beziehungen werden Fremdschlüssel

Als Fremdschlüssel in der abhängigen Relation (n-Seite) wird der Primärschlüssel der unabhängigen Relation (1-Seite) eingefügt; er verweist daher auf eine Zeile (Row) in dieser Relation ähnlich einem Pointer oder einem Link.

z.B.: Beziehung Projekt-Kunde → Projekt(..., KundenNr, ...)



➡ Warum nicht Kunde(..., ProjektId, ...)

Überleitung eines ERDs in ein relationales Modell

3)m:n Beziehungen werden assoziative Tabellen

→ Beispiel ad-hoc erweitern

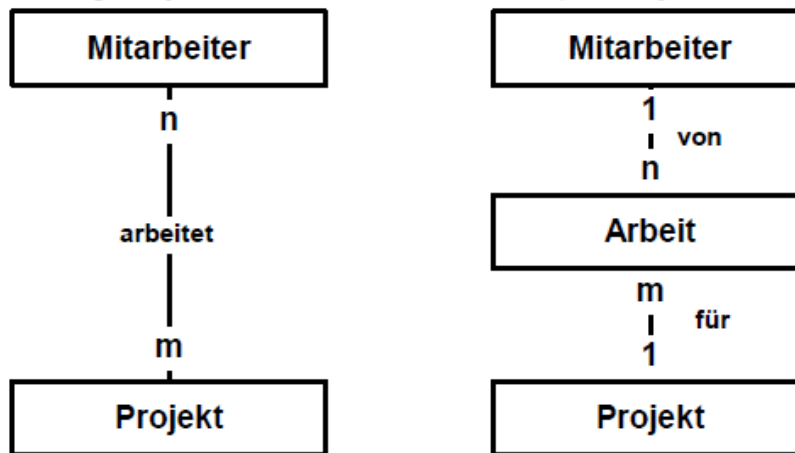
→ Warum nicht Projekt(..., PersNr, ...)

→ Warum nicht Mitarbeiter(..., ProjektId,...)

Die neu gebildete assoziative Relation enthält die Primärschlüssel der beiden in Beziehung stehenden Relationen. Sie erhält ihren Primärschlüssel entweder direkt als Kombination der beiden Fremdschlüssel oder es wird („künstlich“) ein neuer Primärschlüssel zugeteilt; die beiden Fremdschlüssel werden dann als Nichtschlüsselattribute verwendet.

Beispiel 1:

Beziehung Projekt-Mitarbeiter → Arbeit (... , ProjektId, PersNr, ...)



Überleitung eines ERDs in ein relationales Modell

4)Eigenschaften werden Spalten

Alle Eigenschaften werden als Attribute abgebildet; die Eigenschaftswerte finden sich dann in der Spalte „darunter“.

Beispiel:

Mitarbeiter(..., Name, Vorname, Adresse, Plz, Ort, ...)
Projekt(..., Bezeichnung, Startdatum, Endedatum, ...)
Kunde(..., Name1, Name2, Adresse, Plz, Ort, Umsatz, ...)
Arbeit(..., AnteilArbeitszeit, ...)

5)Jede Beziehung wird zu einem Fremdschlüssel

6)Normalisierung (folgt später)

7)Aggregation (folgt später)