



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

Juni 2022

# Entity-Relationship-Modell

Datenbanken-Vorlesung Sommersemester 2022

Prof. Dr. Eva-Maria Iwer

# GLIEDERUNG



 DATENBANKENTWURF	 PHASEN DES ENTWURFSPROZESSES	 ENTITY-RELATIONSHIP MODELL
 KONZEPTIONELLER ENTWURF MIT ER-MODELL	 QUALITÄTSMERKMALE	 WEITERE BEISPIELE
 DESIGN TECHNIKEN	 VOM ER-DIAGRAM ZU RELATIONEN	



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# DATENBANKENENTWURF

# DATENBANKENTWURF

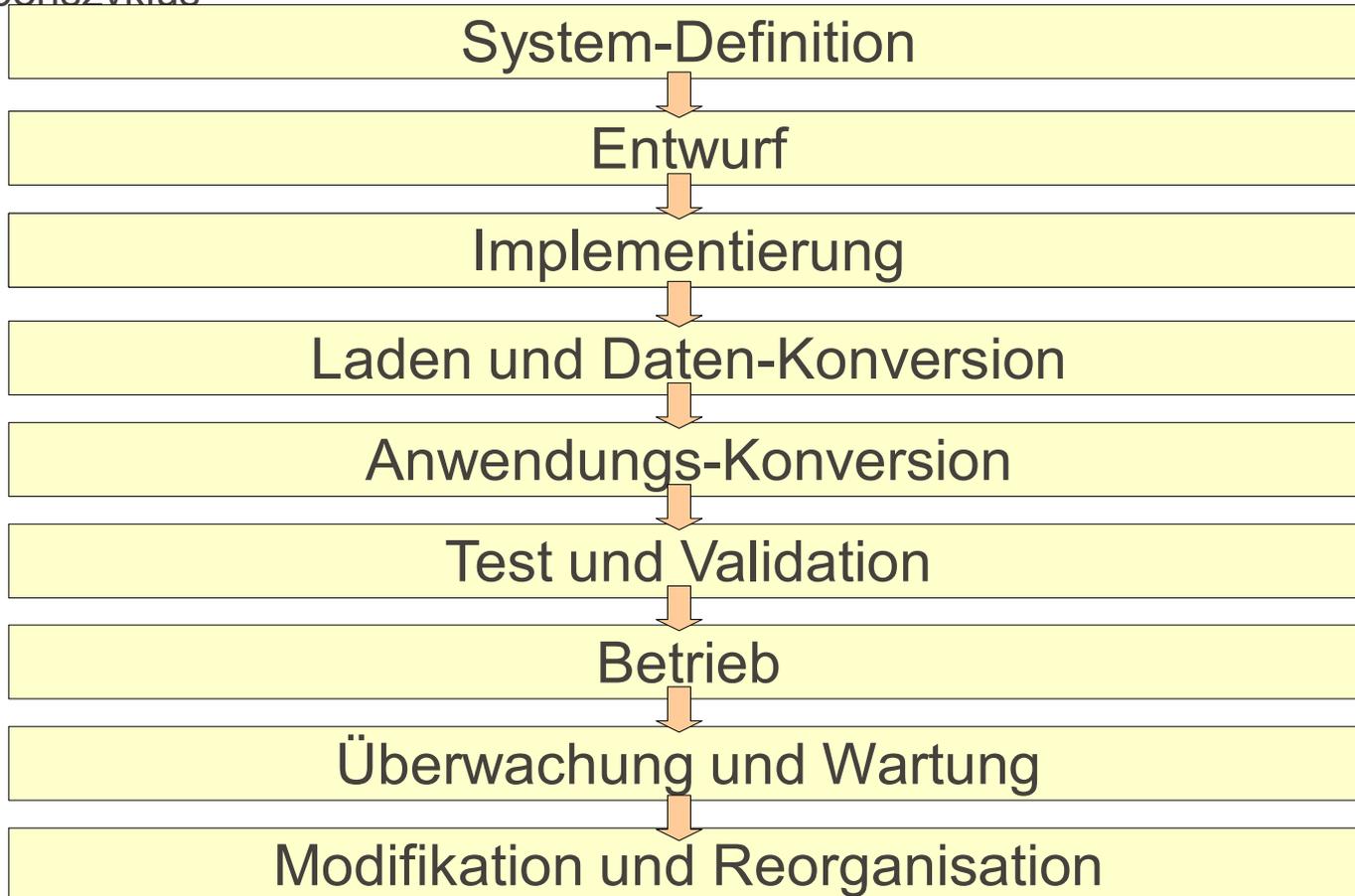


Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- **Datenbankentwurf:** Bestimmung der Struktur und des inhaltlichen Aufbaus
- Immer nur für eine spezielle Anwendung
- Herstellung geeigneter Abstraktionen von gewissen realen Gegebenheiten

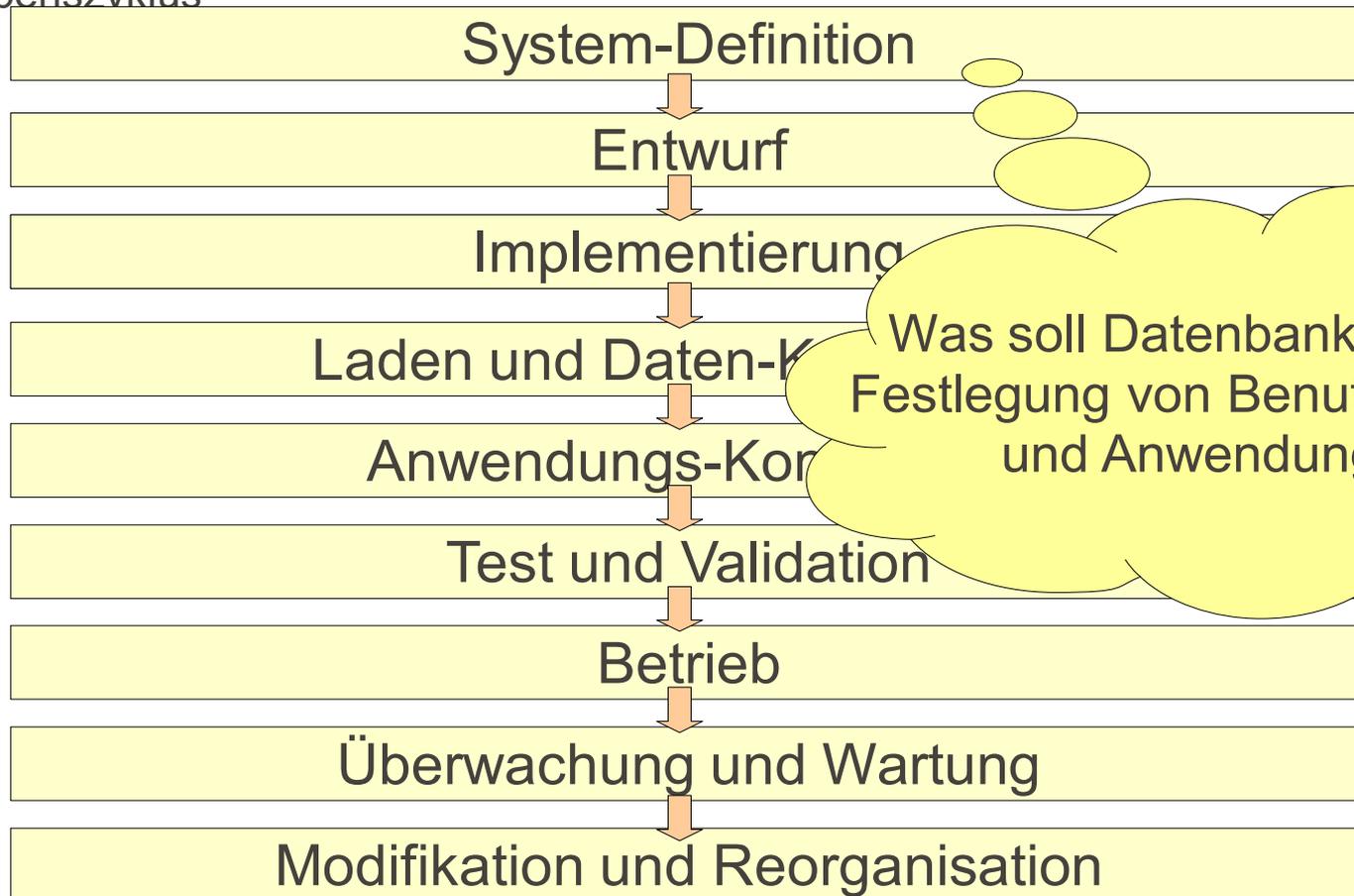


- Lebenszyklus





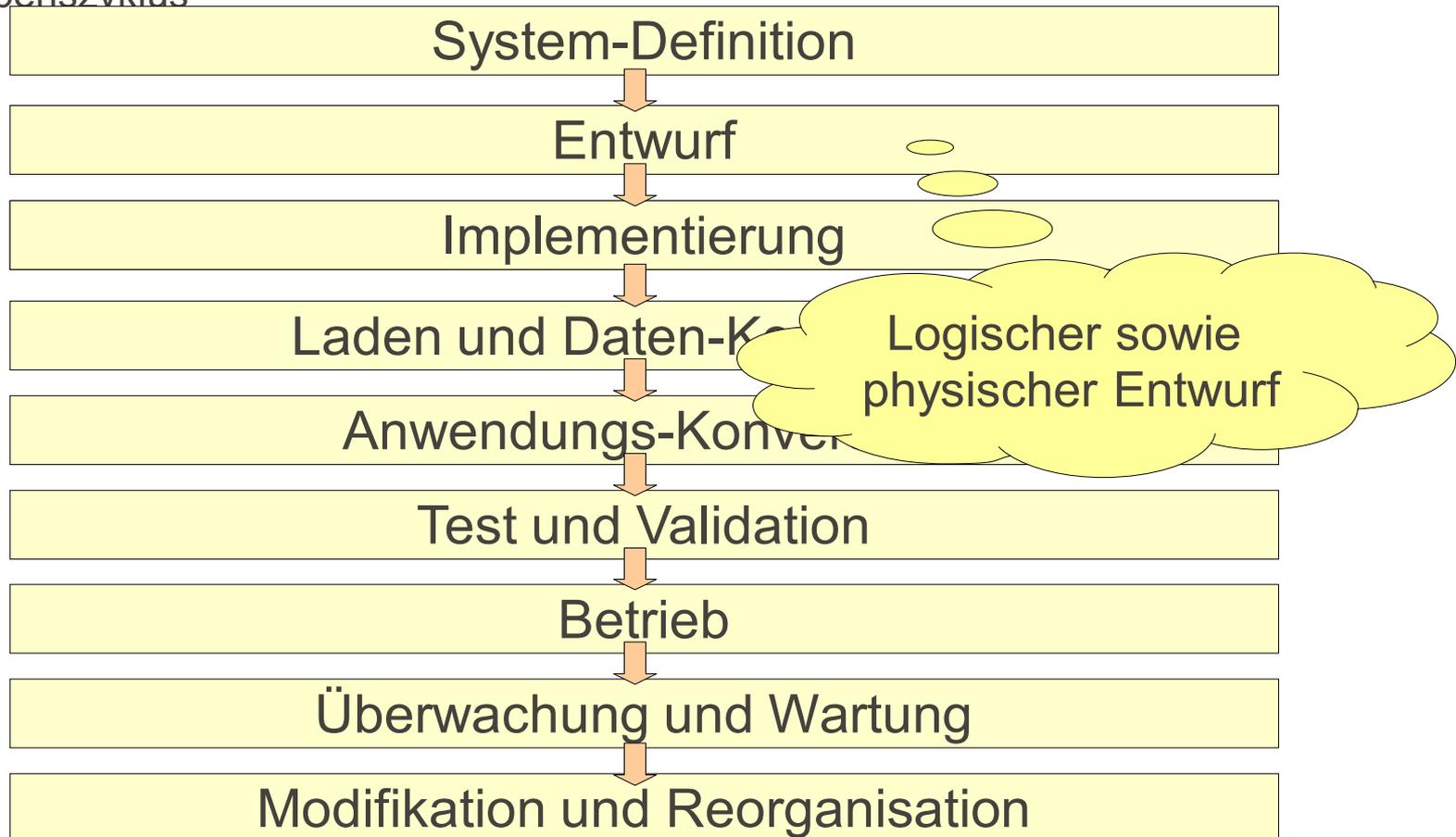
- Lebenszyklus



Was soll Datenbank aufnehmen,  
Festlegung von Benutzergruppen  
und Anwendungen

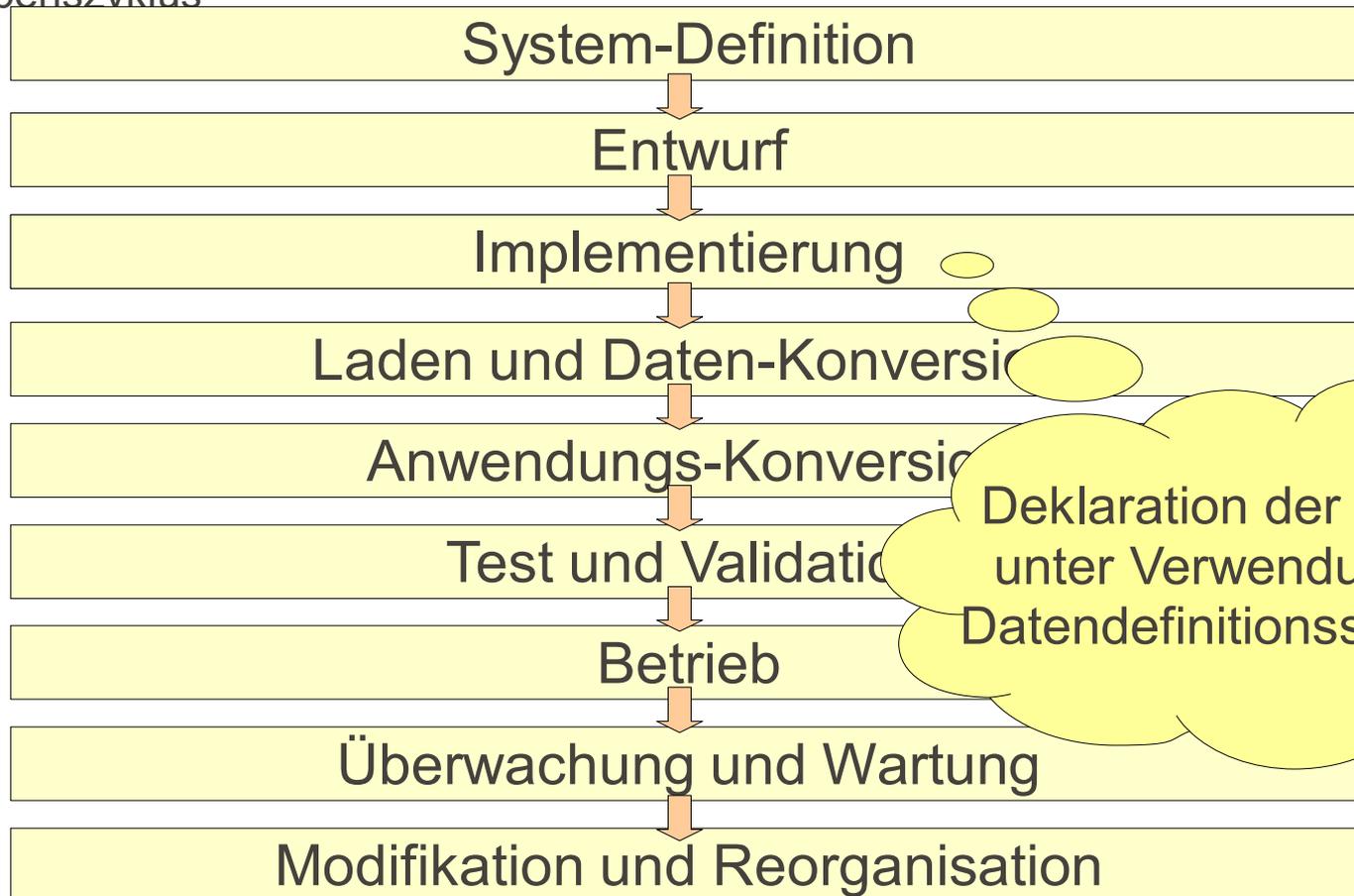


- Lebenszyklus





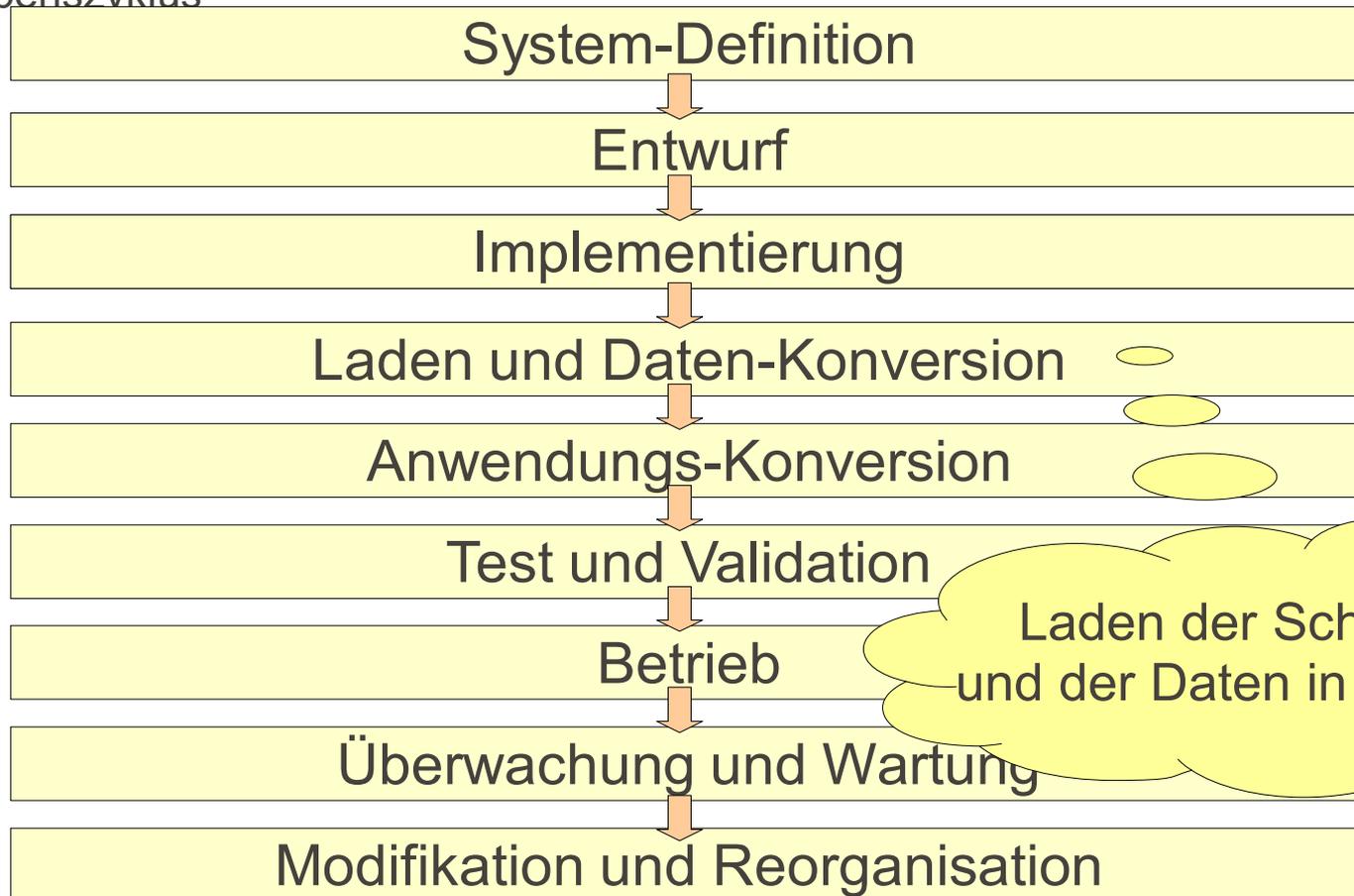
- Lebenszyklus



Deklaration der Schemata unter Verwendung von Datendefinitionssprachen



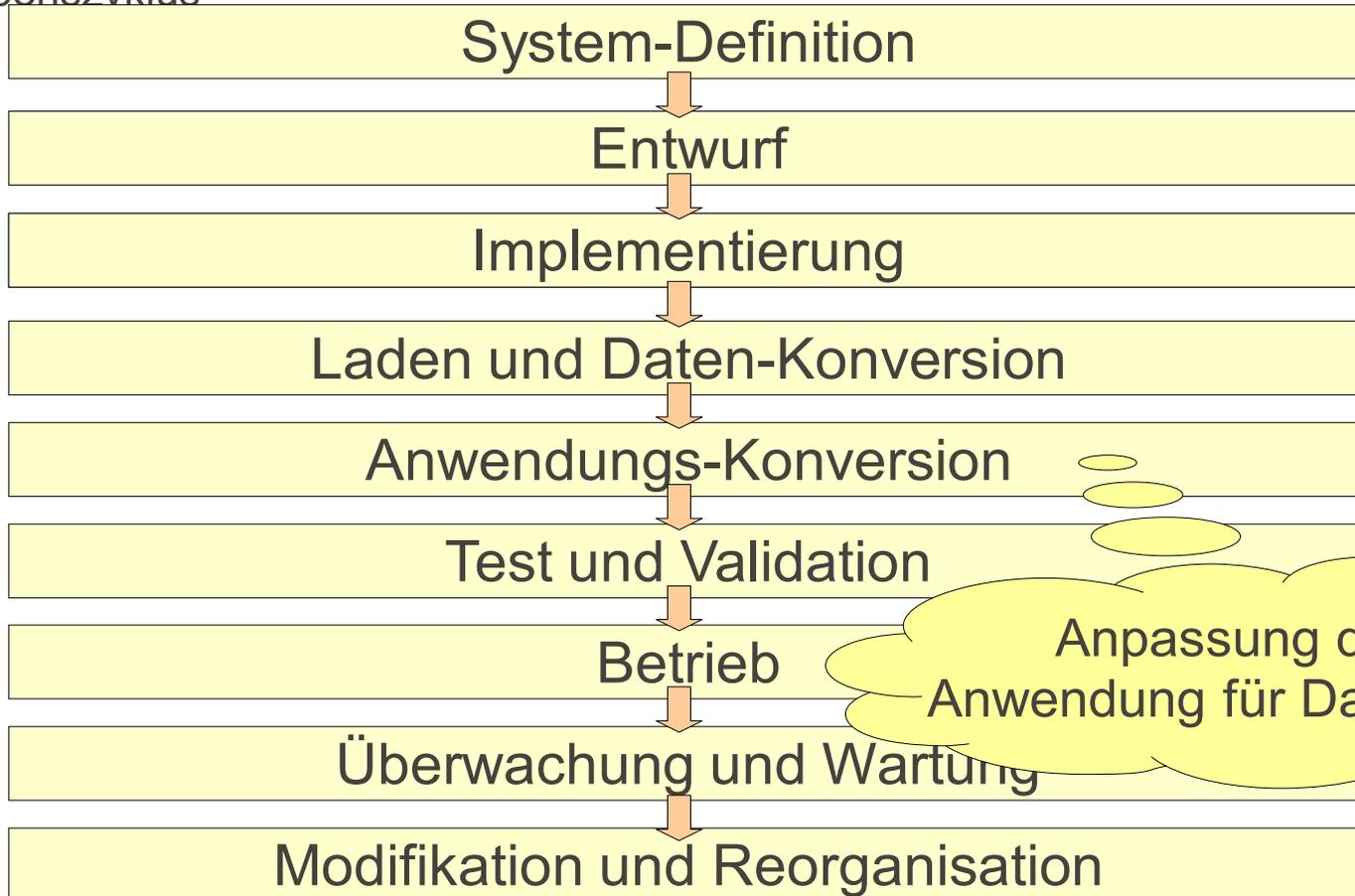
- Lebenszyklus



Laden der Schemata  
und der Daten in Datenbank



- Lebenszyklus



Anpassung der Anwendung für Datenbank



- Lebenszyklus





## Qualitätssicherung

- **Vollständigkeit:** wenn alle relevanten Eigenschaften und Aspekte des Anwendungsbereichs erfasst sind.
  - Prüfung:
    - Alle gegebenen Anforderungen prüfen, ob in Schema enthalten
    - Prüfen, ob wirklich alles im Schema für Anwendung notwendig
- **Korrektheit:** Datenmodell in der richtigen Weise verwendet (syntaktische oder semantische Korrektheit)



## Qualitätssicherung

- **Minimalität:** minimal, falls
  - Jeder Aspekt nur einmal vorkommt
  - Kein Konzept ohne Informationsverlust entfernt werden kann
- → Keine Redundanz vorhanden
- **Lesbarkeit:** in natürlicher Weise und leicht verständlich, selbsterklärend



## Qualitätssicherung

- **Modifizierbarkeit:** es müssen evtl. neue Anforderungen eingebaut werden, modularer Aufbau
- **Normalisierung:** Herstellung einer gewünschten Normalform aus Relationenmodell – für übersichtliche Struktur und Vermeidung von Redundanzen



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# PHASEN DES ENTWURFSPROZESSES

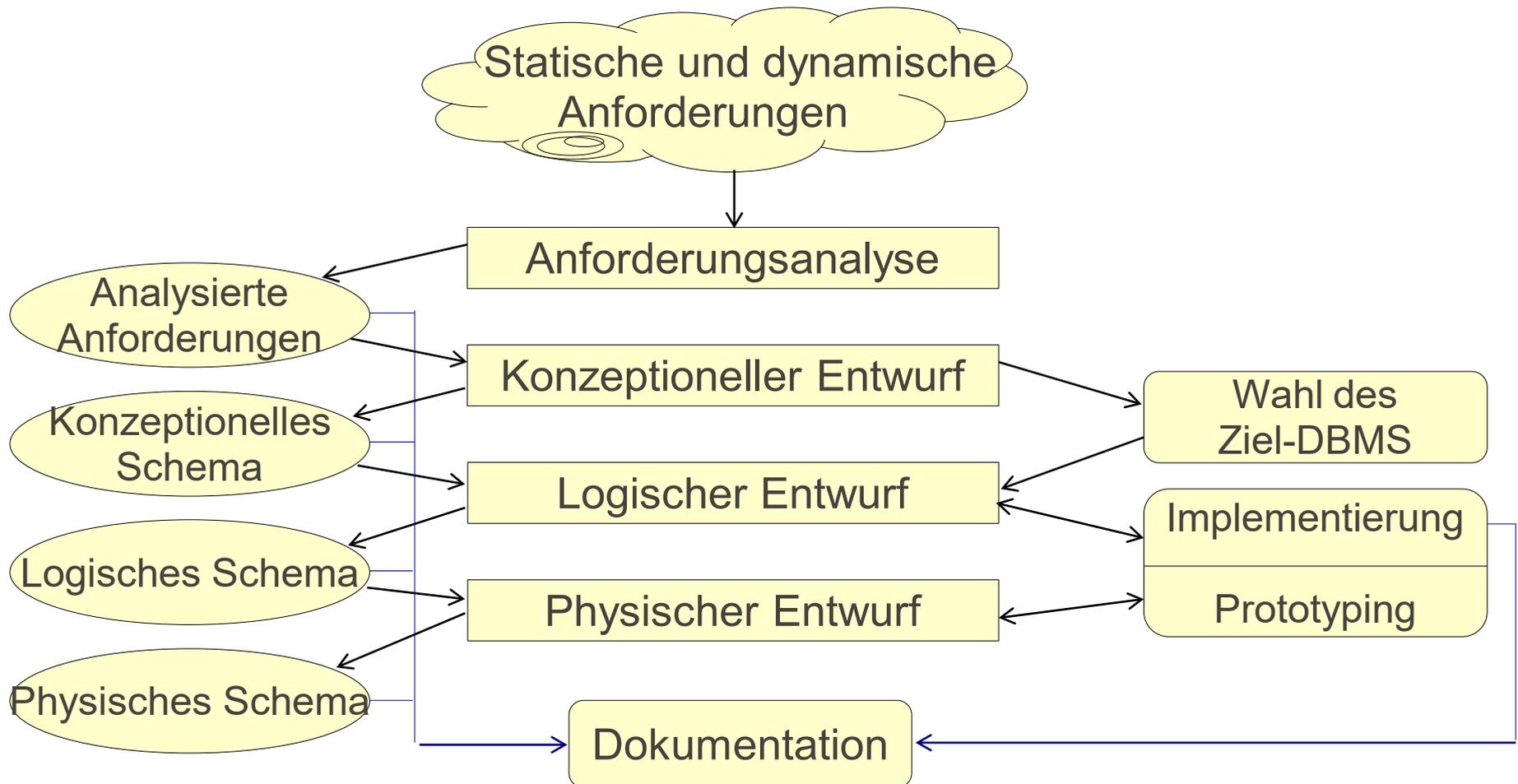
# PHASEN DES ENTWURFSPROZESSES



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Herstellung eines formalen Abbilds einer gegebenen Realwelt oder eines Ausschnitts

# PHASEN DES ENTWURFSPROZESSES

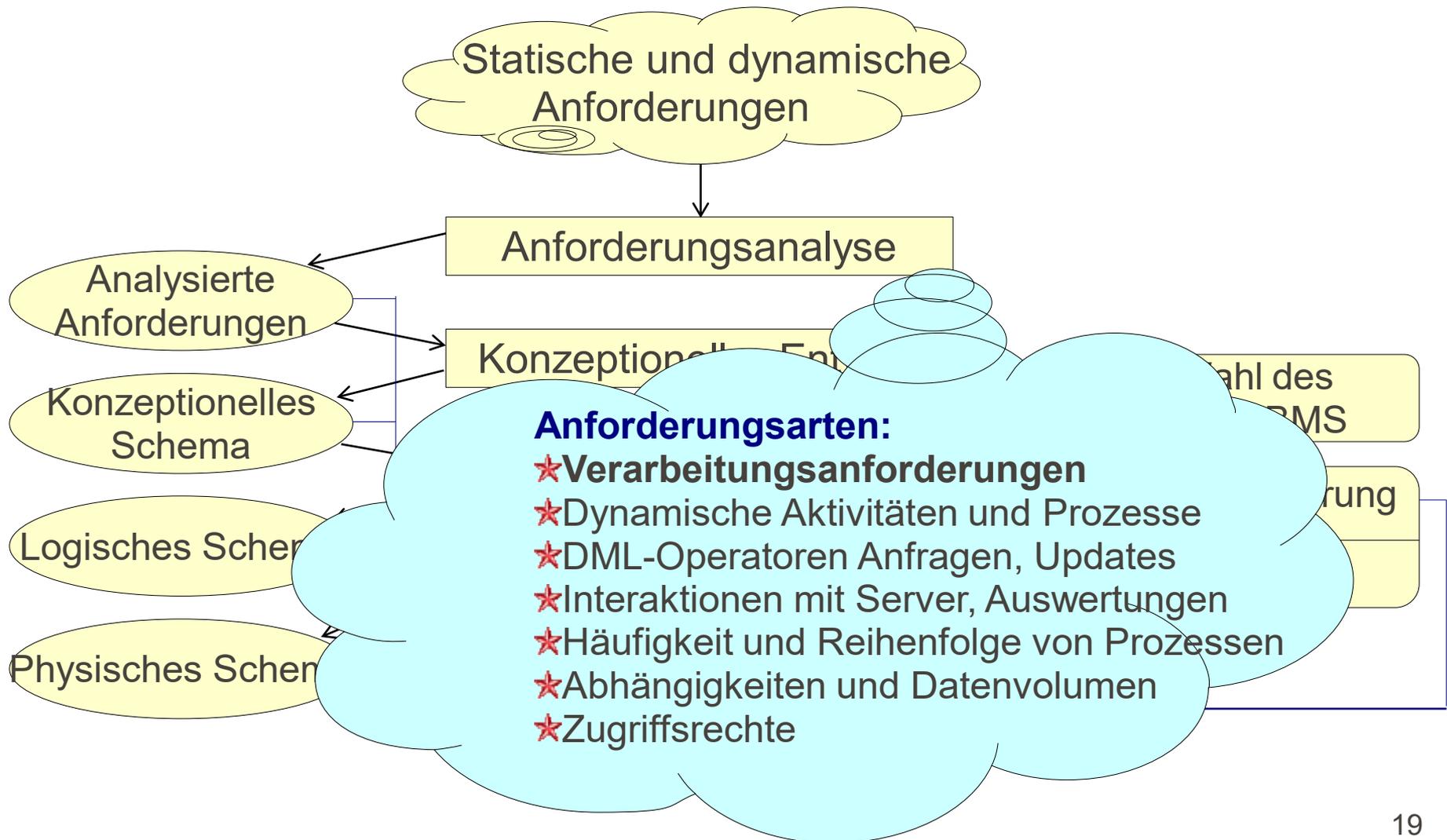


# PHASEN DES ENTWURFSPROZESSES

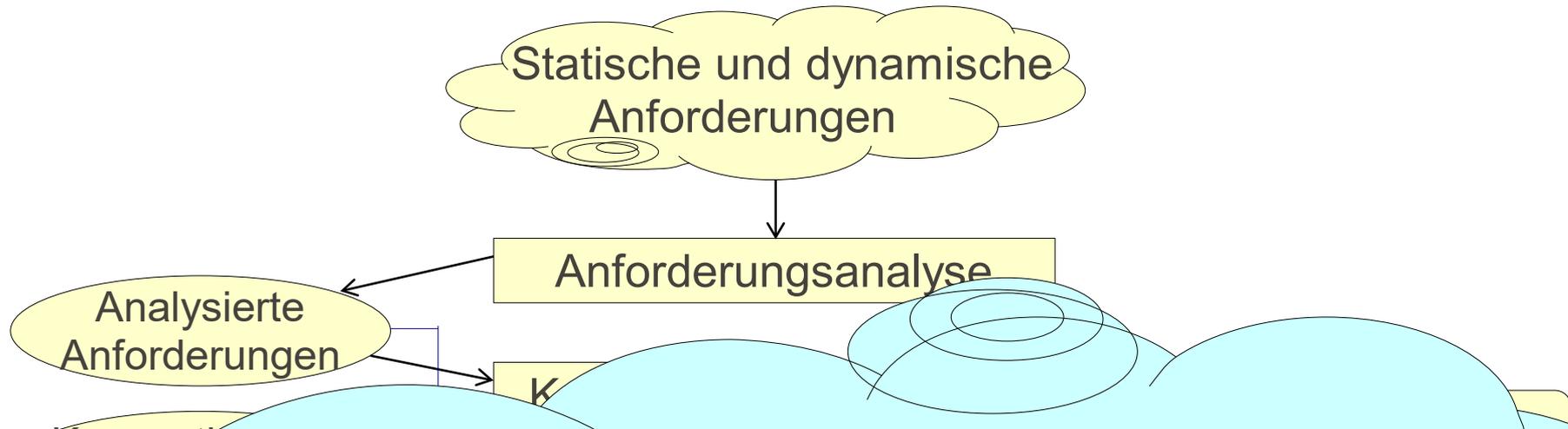




# PHASEN DES ENTWURFSPROZESSES



# PHASEN DES ENTWURFSPROZESSES



## Aktivitäten

### 1. Identifikation von Benutzergruppen und Anwendungsbereichen:

Bestimmung einer Person, die Gruppe oder Bereich repräsentiert

### 2. Sichten existierender Dokumentation:

Verfahrensweisen-Handbücher, Software-Handbücher, Organigramme

Was beeinflusst die Anforderungsanalyse?

Gibt es bereits existierende Datenbanken bzw. Entwürfe

### 3. Fragebögen und Interviews:

Von potenziellen Benutzern auszufüllen

Fragen über Struktur der Daten und Wichtigkeit bzw.

Prioritäten von Applikationen



# PHASEN DES ENTWURFSPROZESSES





## PHASEN DES ENTWURFSPROZESSES

### **Beispiel:** *Konzeptioneller Entwurf* Einzelsicht-orientiert Bottom-up

- Für jeden Benutzer oder Benutzergruppe eine Sicht als abstraktes Datenmodell erstellen
  - Die so erstellten Sichten sind bewusst verschieden zu Ziel-DBMS
  - Zur Modellierung wird Entity-Relationship-Modell genutzt
- Integration der Einzelsichten
  - Erstellung von Globalansicht der Datenbank
  - Konstruktion von mehreren ER-Schemata oder eines ER-Schemas
  - Analyse zeigt Inkonsistenzen, Redundanzen und Konflikte
  - Namensgebung, teilweise oder ganze Übereinstimmungen
  - Globalsicht mit bestehenden Abhängigkeiten oder Beziehungen



## PHASEN DES ENTWURFSPROZESSES

### **Beispiel:** *Konzeptioneller Entwurf* Einzelsicht-orientiert Bottom-up

- Für jeden Benutzer oder Benutzergruppe eine Sicht als abstraktes Datenmodell erstellen

- Die so erstellten Sichten werden in einem relationalen DBMS verwaltet

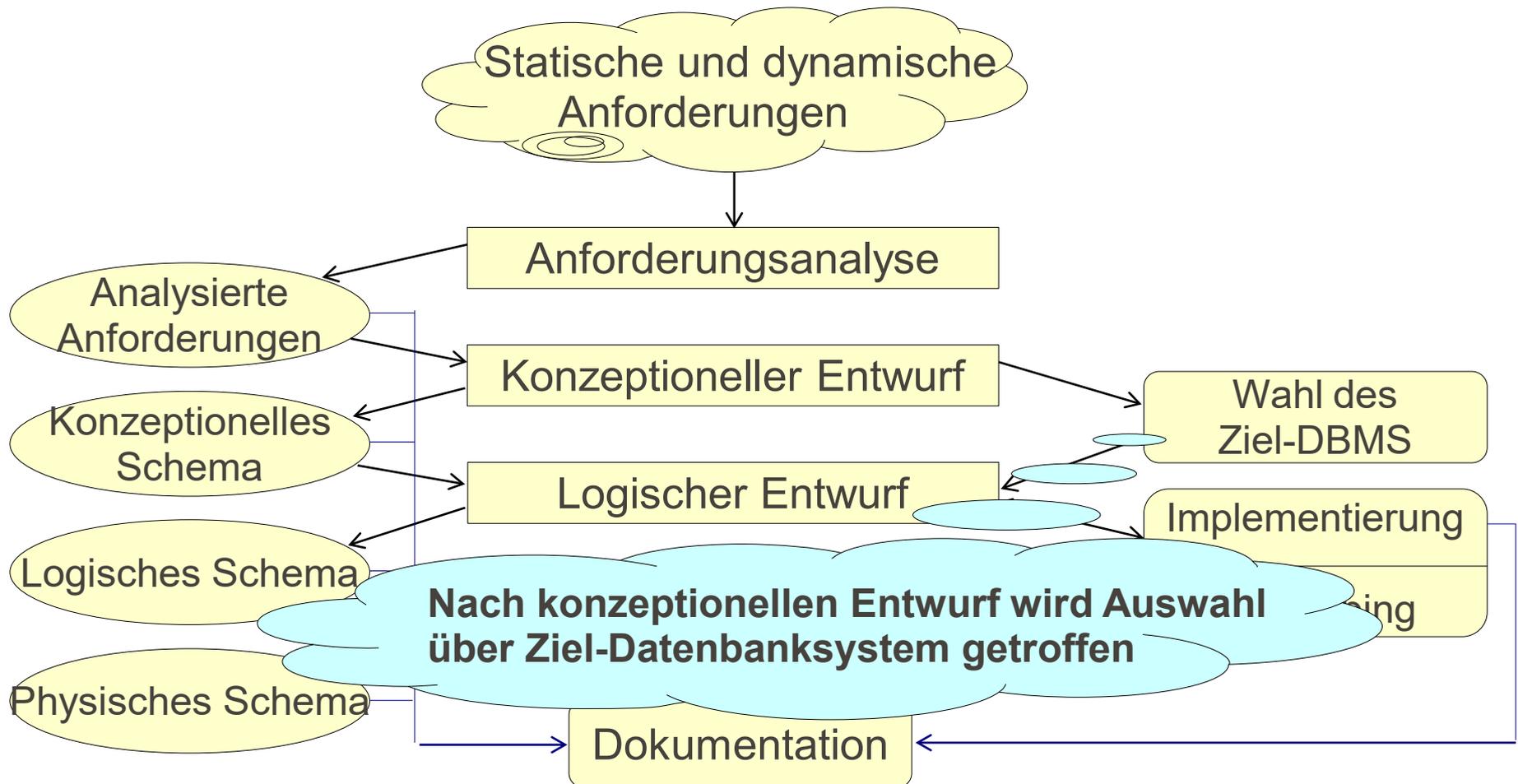
**bottom-up:** erst einzelne Schemata, dann Verallgemeinerung bis zu einem einzigen großen Schema

**top-down:** Modellierung von großen Informationsblöcken und dann immer weitere Detaillierung

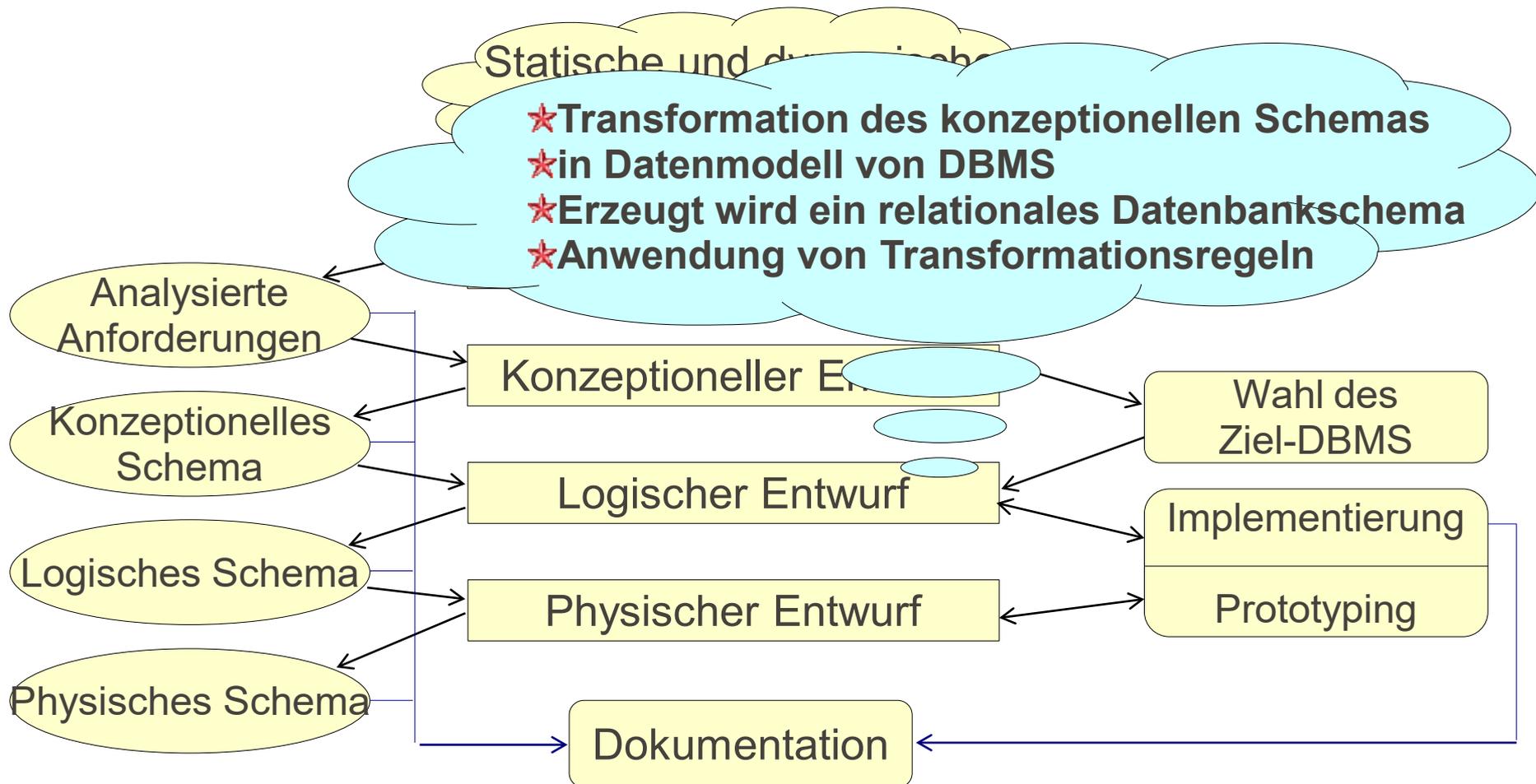
- **Integrität**

- Analyse ergibt Inkonsistenzen, Redundanzen und Konflikte
- Namensgebung, teilweise oder ganze Übereinstimmungen

# PHASEN DES ENTWURFSPROZESSES

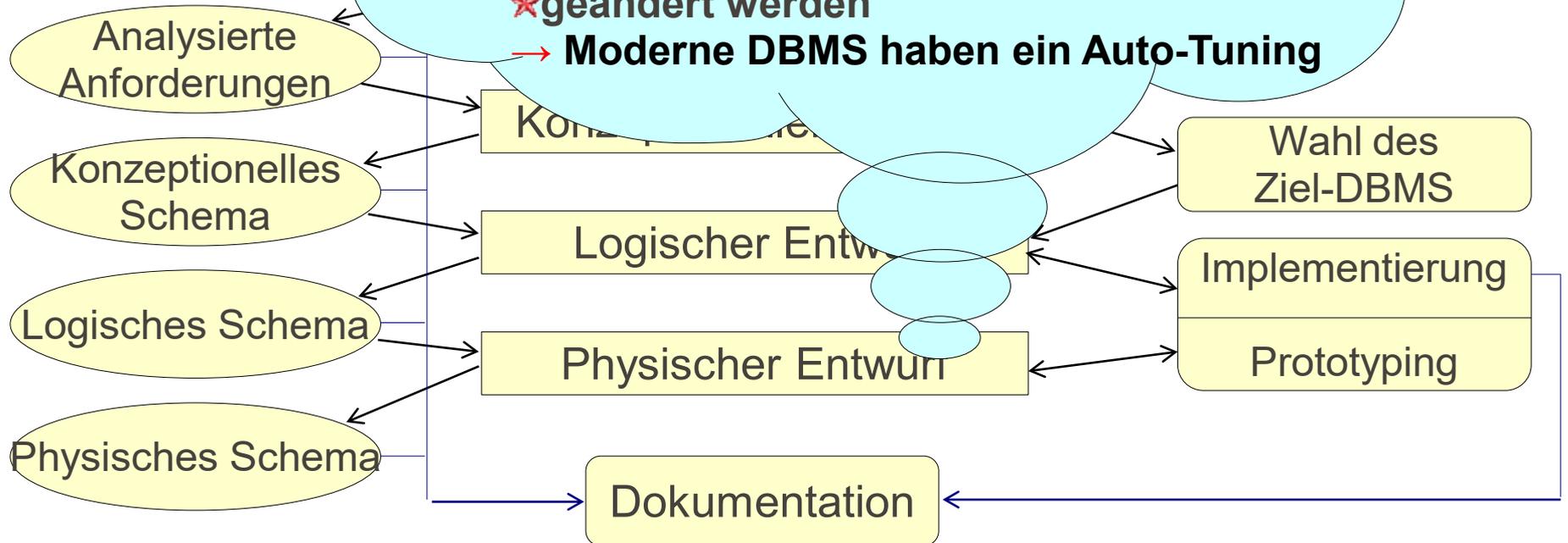


# PHASEN DES ENTWURFSPROZESSES



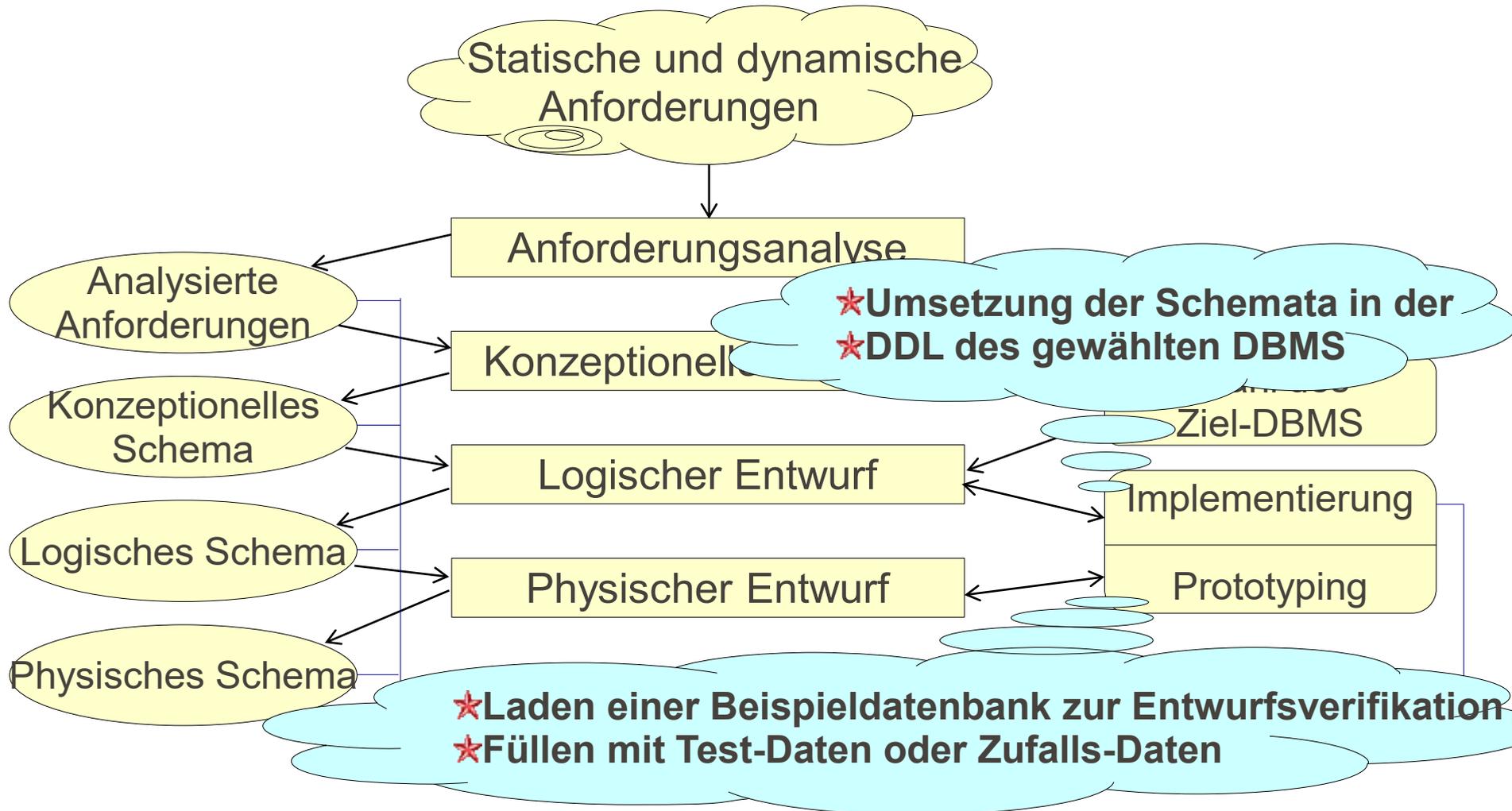
# PHASEN DES ENTW

- ★ Definition des internen Schemas
  - ★ Minimierung der Zugriffe auf Sekundärspeicher
  - ★ (I/O Kosten)
  - ★ Effizienter Zugriff auf Daten
  - ★ Konzeptionelles Modell lässt sich nicht auf
  - ★ Speicherstrukturen abbilden
  - ★ Datenbank muss entsprechend getuned werden
  - ★ Während des Betriebs muss evtl. am Tuning
  - ★ geändert werden
- **Moderne DBMS haben ein Auto-Tuning**

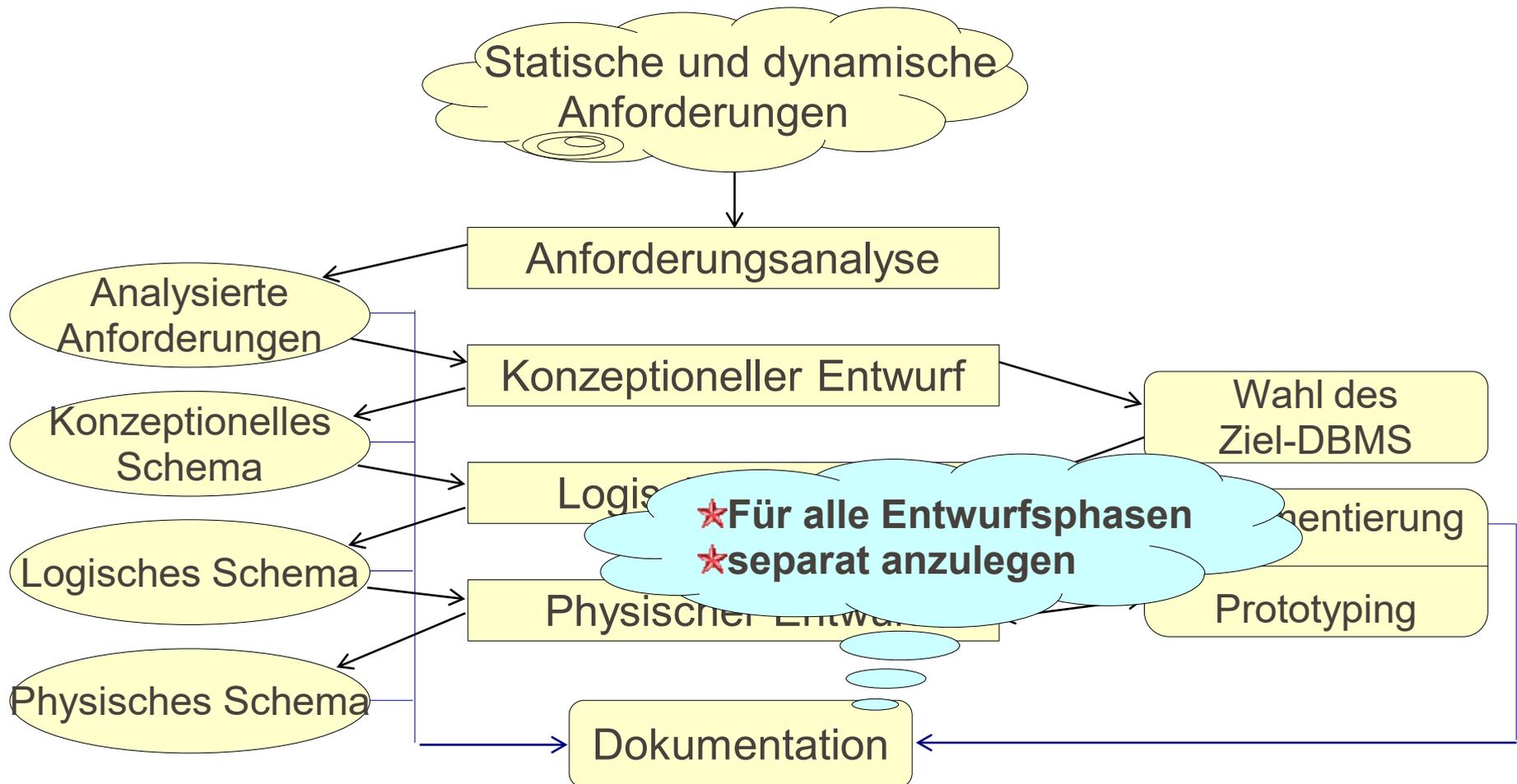




# PHASEN DES ENTWURFSPROZESSES



# PHASEN DES ENTWURFSPROZESSES





Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# ENTITY-RELATIONSHIP MODELL



# ENTITY-RELATIONSHIP-MODELL

- ER-Modell
- 1976 von Peter Chen vorgeschlagen
- Datenbankunabhängiges Modell
- Entities
  - Wohlunterscheidbare Dinge der realen Welt
  - Entities (engl.): Dateneinheit
  - z.B. Person, Auto, Stadt
- Entity-Set
  - Ähnliche oder vergleichbare Entities (z.B. alle Angestellten eines Betriebs)



# ENTITY-RELATIONSHIP-MODELL

- **Beispiel:** Bücher einer Bücherei

<i>Attribut</i>	<i>Domain</i>
<i>InvNr</i>	siebenstellige Zahl
<i>Autor</i>	Zeichenreihe der variablen Länge 12
<i>Titel</i>	Zeichenreihe der variablen Länge 50
<i>Verlag</i>	Zeichenreihe der festen Länge 2 oder 3
<i>Jahr</i>	vierstellige Zahl zwischen 1950 und 2020

- Ein einzelnes Buch ist ein **Entity**
- Die Menge aller Bücher in der Bücherei sind ein **Entity-Set**
- Die Attribute eines Buches bilden den **Entity-Typ** (analog der Attribute oder Properties der Klasse)



# ENTITY-RELATIONSHIP-MODELL

- Entity-Typen
  - Struktur von Entities, beschrieben durch deren Attribute
- Attribute
  - Entities besitzen Attribute (Farbe, Geburtsdatum, Adresse)
  - Konkrete Ausprägungen sind Werte (engl. Values)
  - Alle zugelassenen Werte sind der Wertebereich (engl. Domain)

# ZIEL VON ER MODELL

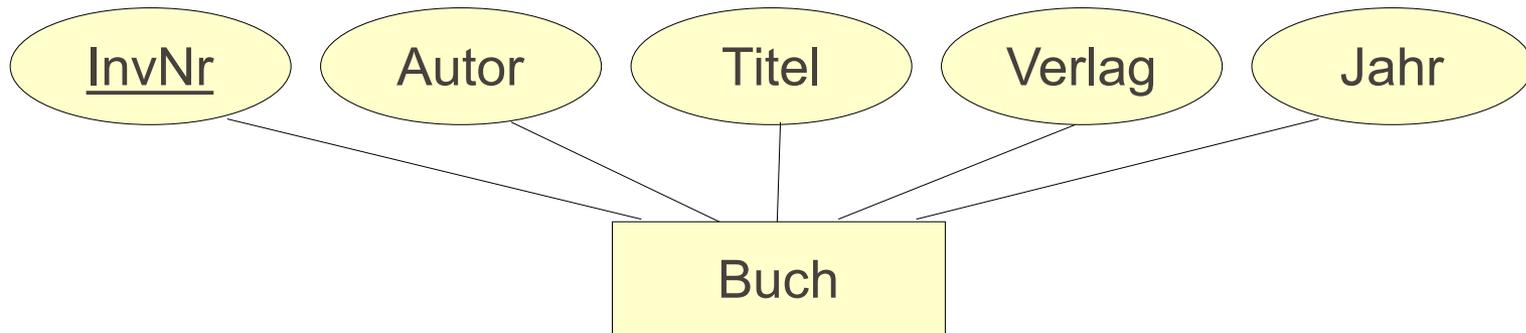


- Design der Datenbank
- Entity-Relationship Diagramm
- Später: umwandlung von ER zu DB Design.



# ENTITY-RELATIONSHIP-MODELL

- Graphische Veranschaulichung
  - **Entities** werden als Rechtecke dargestellt
  - Attribute sind mit Rechteck verbundene Ovale

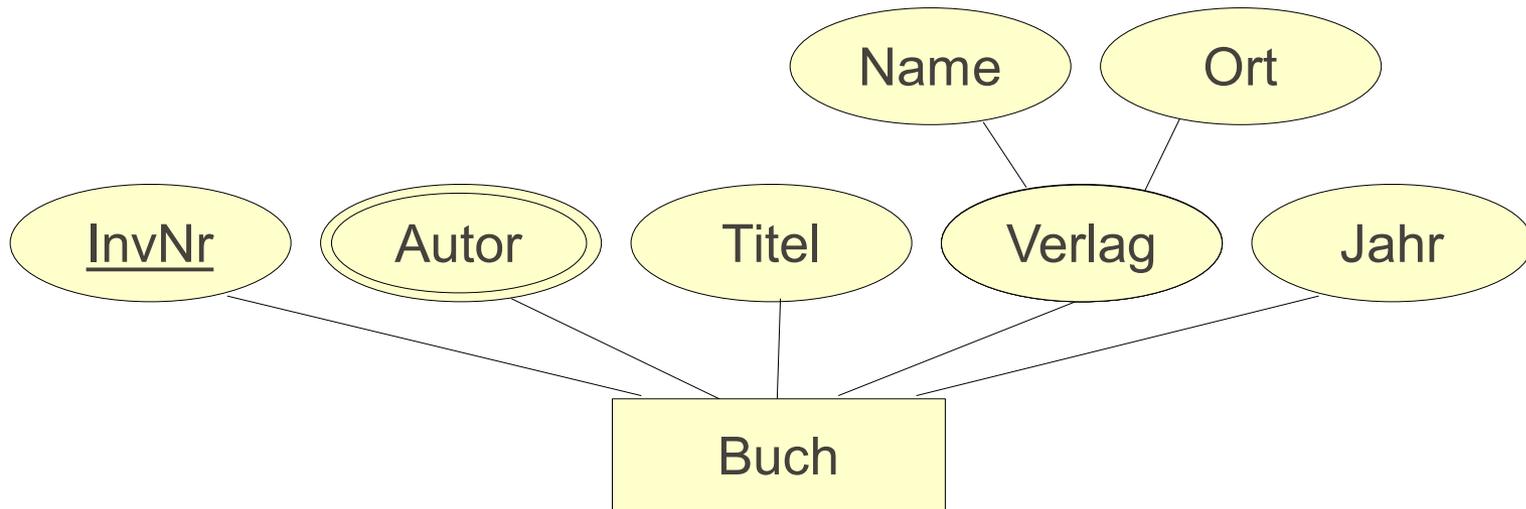


- Unzureichende Beschreibung von der Realität von Büchern
  - Ein Buch kann mehrere Autoren haben
  - Ein Verlag setzt sich aus Name und Ort zusammen



# ENTITY-RELATIONSHIP-MODELL

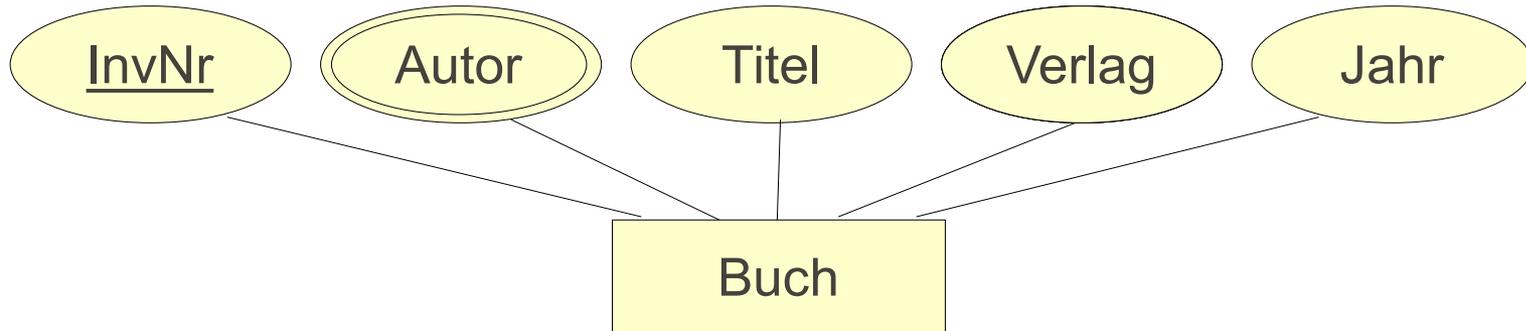
- Mehrwertige Attribute → Doppeloval
- Zusammengesetzte Attribute → Ovale mit Kanten zu Zusammensetzung verbunden





# ENTITY-RELATIONSHIP-MODELL

- Unterstreichungen → Attribut als eindeutige Identifikation (Schlüsselattribut)



- Mehrere Attribute können in einen Schlüssel einbezogen werden
- Es können mehrere Schlüssel existieren
  - Einer wird Primärschlüssel ausgezeichnet
  - Andere Sekundärschlüssel



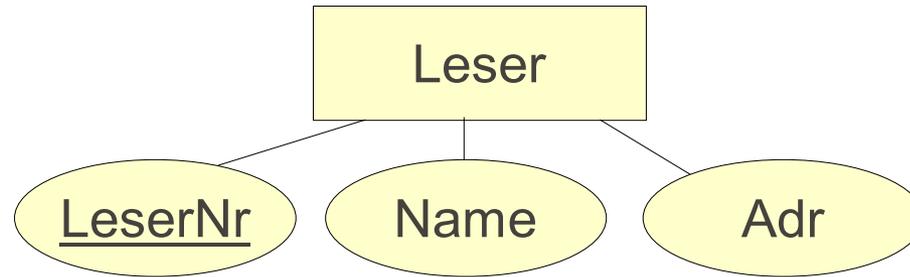
## SCHLÜSSEL (KEYS)

- Vermeidung von Redundanz
- Ein Schlüssel ist eine Menge von Attributen für ein Entity Set, sodass keine zwei Entitys in allen Attributen des Schlüssels übereinstimmen
  - Es ist erlaubt für zwei Entitys, dass sie sich auf eine Teilmenge der Schlüsselattribute übereinstimmen, aber nicht auf alle Schlüsselattribute.
- Für jedes Entity Set MUSS ein Schlüssel festgelegt werden.

# ENTITY-RELATIONSHIP-MODELL



- Weiteres Entity



# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

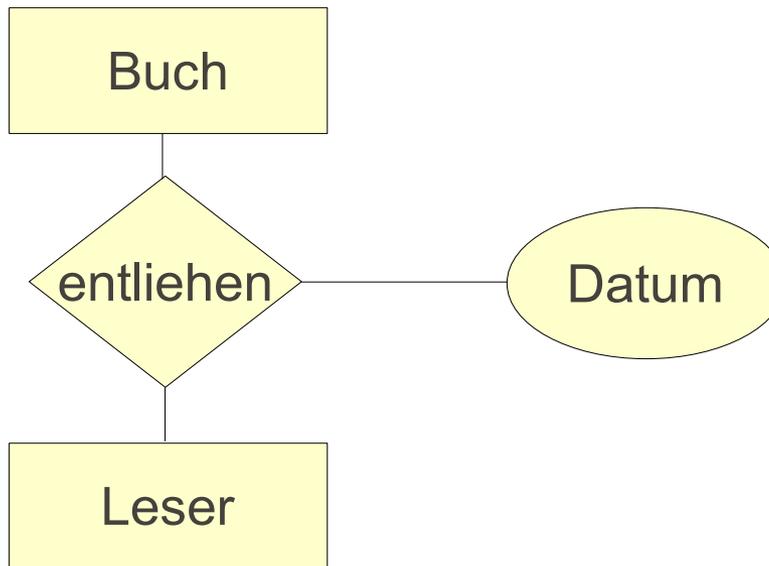
- **Beziehungen** (engl. Relationship)
  - Bsp: Bücher werden von Lesern 'entliehen'
  - Ein Buch steht mit einem bestimmten Leser in Beziehung
  - Beziehungen können eigene Attribute haben
  - Die Beziehung 'entliehen' hat z.B. Attribut Rückgabedatum
  - Beziehungen werden durch eine Raute, welche Namen enthält, repräsentiert
  - Kanten verbinden die beteiligten Entity-Deklarationen
  - Attribute werden durch Ovale dargestellt

# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- **Beispiel:** Beziehung zw. Büchern und Lesern



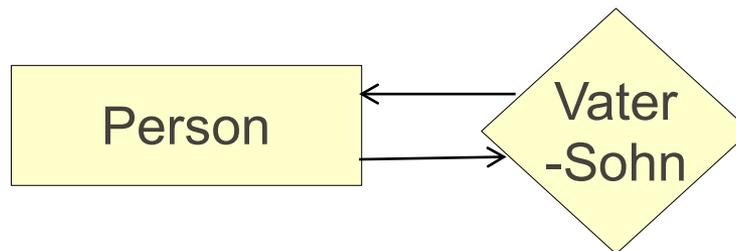
- Attribute von Entities zwecks Übersichtlichkeit weggelassen

# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- Kanten sind ungerichtet, außer wenn es rekursive Beziehungen mit Rollenangaben sind
- **Beispiel:** Rekursive Beziehungen zwischen Personen

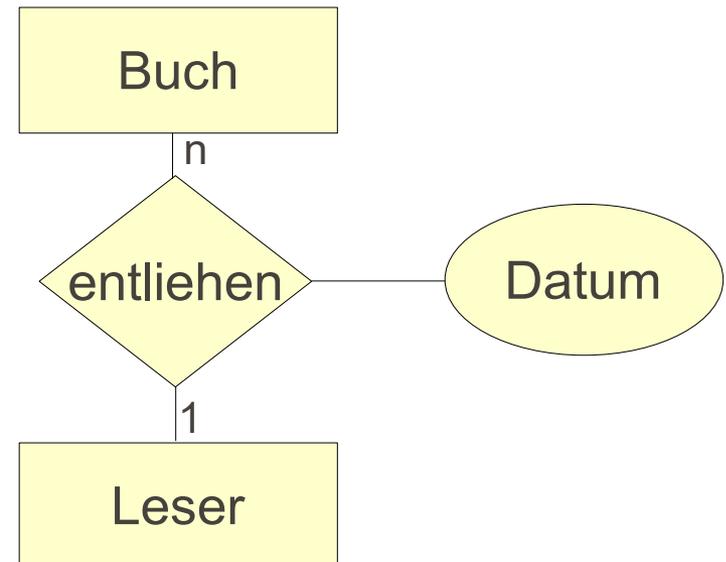


# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- Die **Komplexität** definiert die Anzahl der in Beziehung stehenden Entities
- Komplexität einer Beziehung
  - Wie oft darf die Beziehung auftreten?
  - Mögliche Werte 1:1 1:n m:n
  - Problem, es können keine Höchstwerte angegeben werden

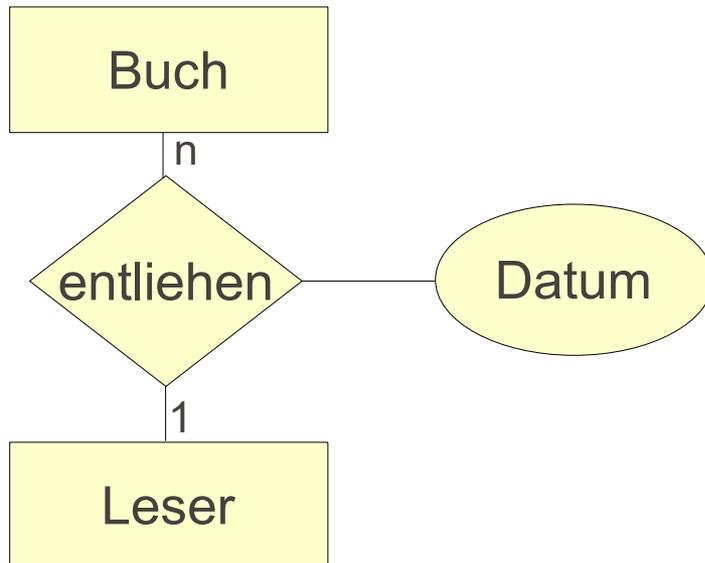


# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim



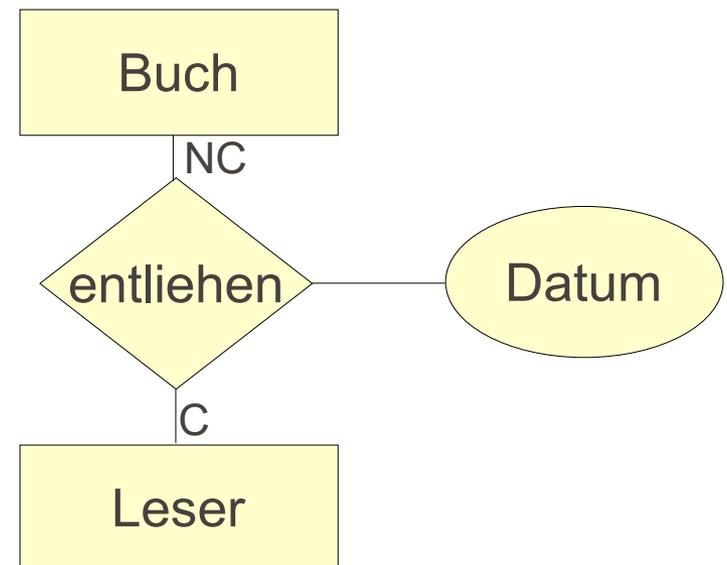
- Ein Leser kann mehrere Bücher ausleihen
- Bücher können immer nur von einem Leser ausgeliehen werden

# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- Alternative Schreibweise
  - 1 – genau eins
  - C – keins oder eins
  - N – mindestens ein, auch beliebig viele (oder M)
  - NC – keins, eins, beliebig viele (oder MC)

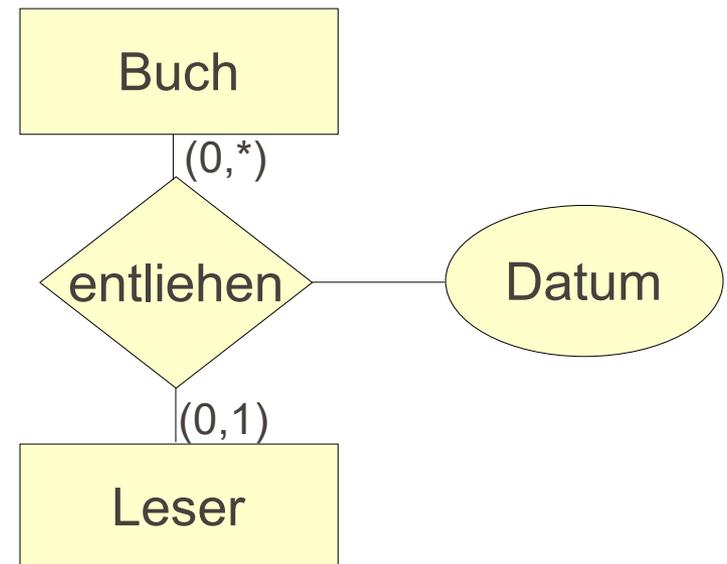


# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- Alternative Schreibweise
  - (min,max)-Notation
  - \* – unbegrenzt



# ENTITY-RELATIONSHIP-MODELL

## BEZIEHUNGEN



- Schwache Entität
  - Eine Entität kann ohne die Existenz einer anderen Entität nicht existieren (manchmal hat nur die Entität eine doppelte Linie)



# ENTITY-RELATIONSHIP-MODELL

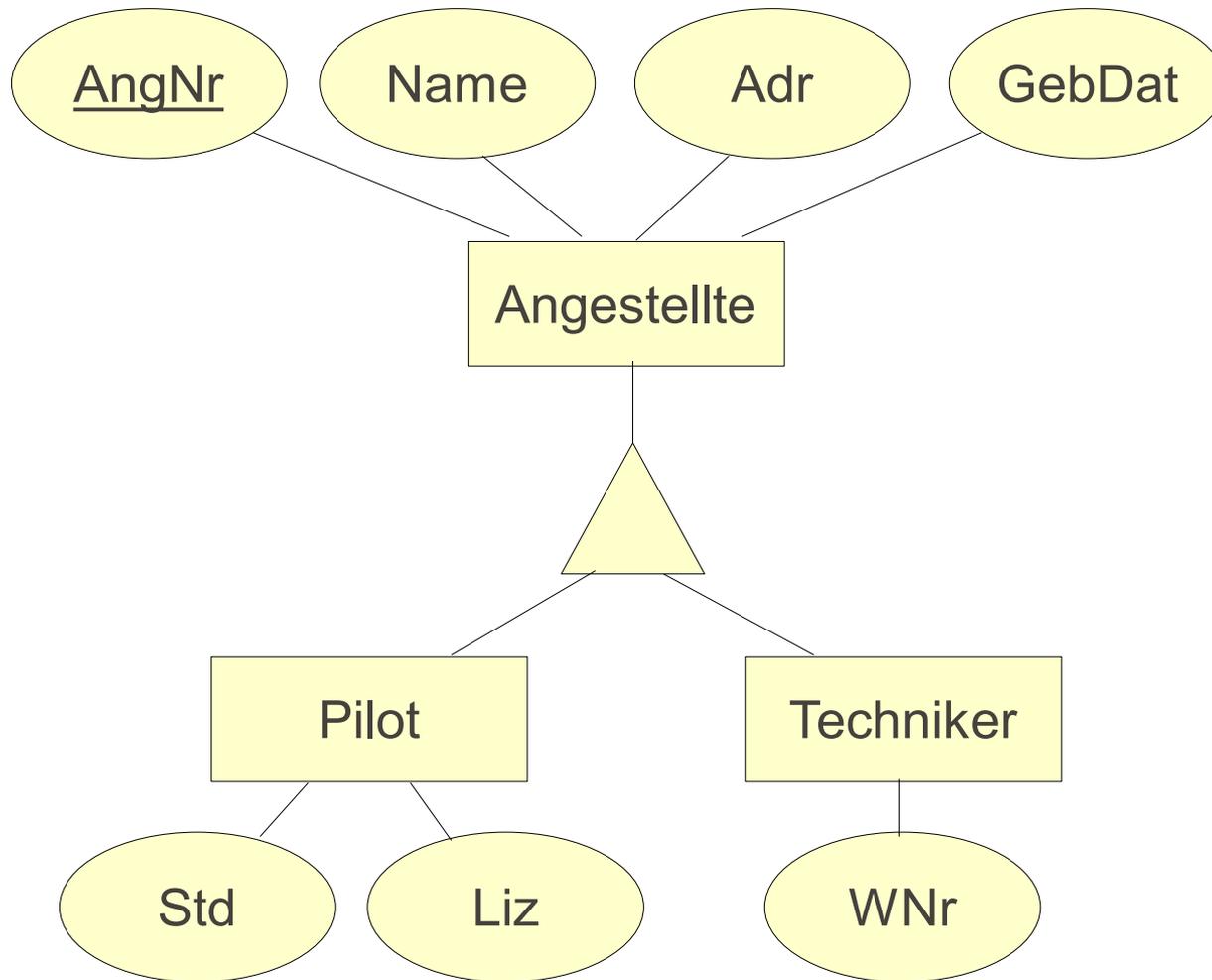
## IS-A-BEZIEHUNGEN



- Angestellte von Fluggesellschaft
- *Angestellte* = (*{AngNr, Name, Adresse, GebDat, ...}*,  
*{AngNr}*)
- Spezialisierung von Angestellten
  - Piloten, zusätzlich Flugst. (Std) und Fluglizenz (Liz)
  - Techniker, zusätzlich Wartungsteam (WNr)
- Alle Attribute werden an Spezialisierung vererbt
- Als Dreieck auf Verallgemeinerung zeigend, mit Kanten verbunden

# ENTITY-RELATIONSHIP-MODELL

## IS-A-BEZIEHUNGEN (SUBCLASSES)



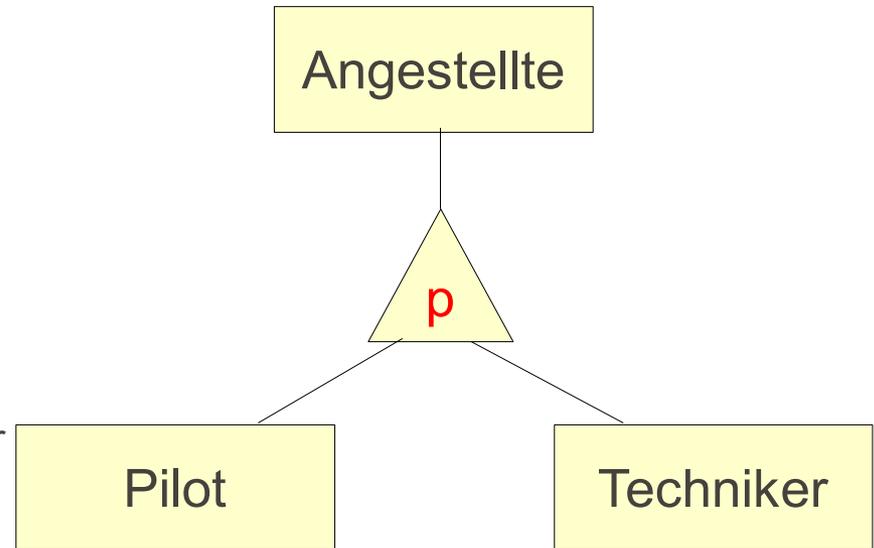
# ENTITY-RELATIONSHIP-MODELL

## IS-A-BEZIEHUNGEN



- Drei Arten von Entities:
  - Piloten
  - Techniker
  - Angestellte, weder Piloten noch Techniker

- → nicht vollständig in Spezialform zerlegbar
- → gilt als **partiell** (Gegenteil zu **total**)
- Im Dreieck mit **p** oder **t** angegeben (im Englischen mit „is a“)



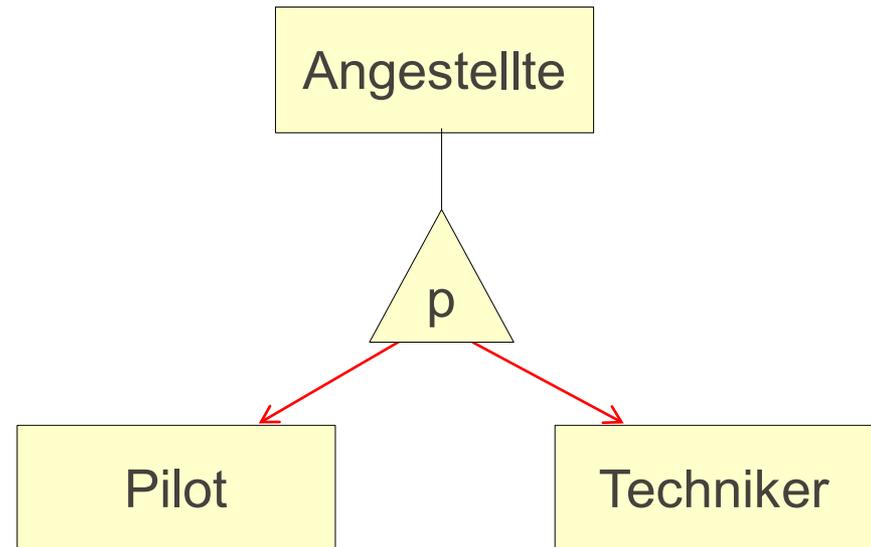
# ENTITY-RELATIONSHIP-MODELL

## IS-A-BEZIEHUNGEN



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Piloten und Techniker haben keine gemeinsamen Elemente, d.h. sie sind **disjunkt**
- Gerichtete Pfeile
  - **disjunkt**, Verallgemeinerung "von oben" (Pfeile von oben nach unten)
  - **nicht disjunkt**, Verallgemeinerung "von unten" (Pfeile von unten nach oben)

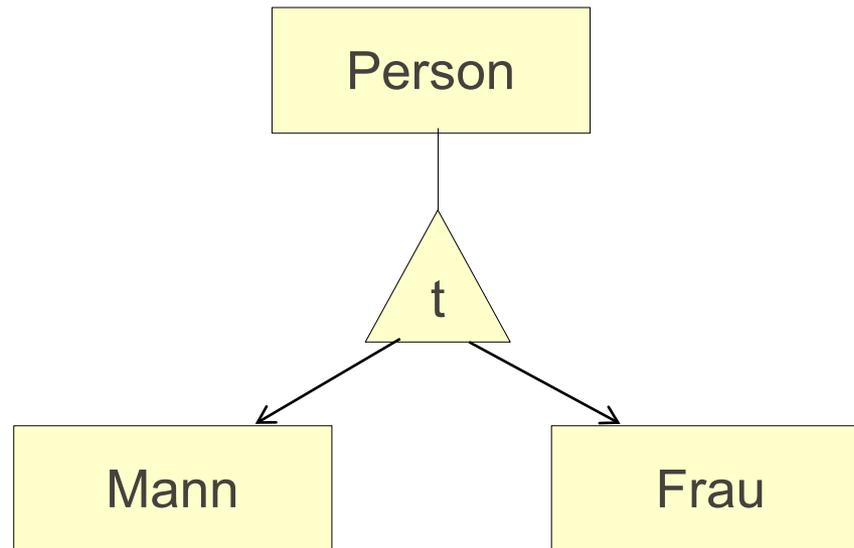


# ENTITY-RELATIONSHIP-MODELL

## IS-A-BEZIEHUNGEN



- **Beispiel:** Totale, disjunkte Spezialisierung

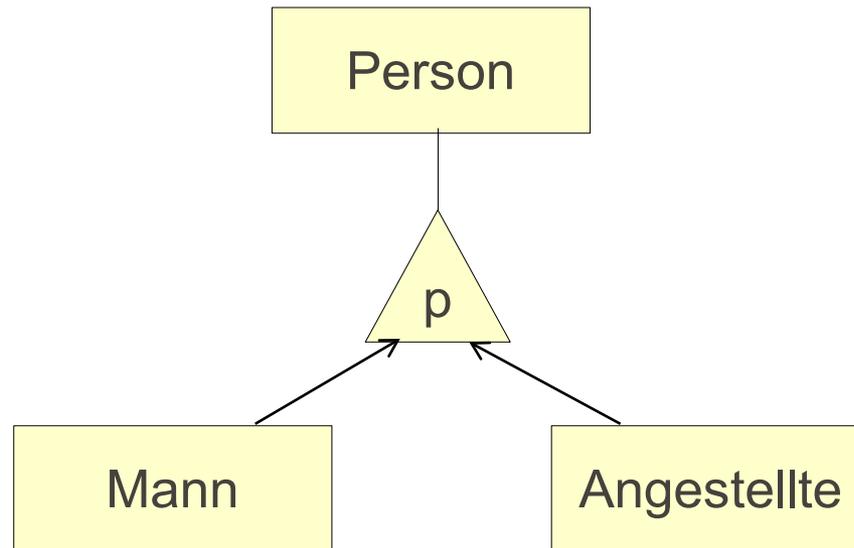


# ENTITY-RELATIONSHIP-MODELL

## IS-A-BEZIEHUNGEN



- **Beispiel:** Partielle, nicht disjunkte Spezialisierung





## E/R VS. OBJECT-ORIENTED SUBCLASSES

- In OO, Objekte sind nur in einer Klasse.
  - Unterklassen erben von Masterklassen.
- Im Gegensatz dazu, E/R Entities haben Repräsentative in allen Unterklassen zu denen sie gehören.
  - **Regel:** Wenn ein Entity e in einer Unterklasse repräsentiert ist, dann ist e auch in der Oberklasse repräsentiert (und rekursiv bis zur Wurzel des Baumes).



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# KONZEPTIONELLER ENTWURF MIT ER-MODELL

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



- Vorgehensweise zur Erstellung einer konzeptionellen Globalsicht
- top-down
  - Beginnt bei großen Informationsblöcken
  - Weitere Detaillierung - schrittweise Verfeinerung

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## 1. Entity-Verfeinerung

- a) Ein bereits existierender Entity-Typ wird durch neue Typen mit relevanten Beziehungen ersetzt.
- b) Ein bereits existierender Entity-Typ wird spezialisiert in Subtypen.
- c) Ein bereits existierender Entity-Typ wird in voneinander unabhängige Typen zerlegt, welche weder miteinander in Beziehung stehen noch Spezialisierungen voneinander darstellen.
- d) Ein Entity-Typ wird mit Attributen versehen, und unter diesem wird ein Primärschlüssel ausgezeichnet.

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## 2. Relationship-Verfeinerung

- a) Ein existierender Relationship-Typ wird in zwei oder mehr Relationships zwischen den beteiligten Entitäten zerlegt.
- b) Ein existierendes Relationship wird durch eine Folge von Beziehungen (unter Hinzuziehung weiterer Entity-Typen) ersetzt.
- c) Ein Relationship wird mit Attributen versehen.

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## 3. Attribut-Verfeinerung

- a) Ein Attribut einer Entität bzw. eines Relationships wird durch mehrere Attribute ersetzt.
- b) Ein Attribut wird durch ein zusammengesetztes Attribut ersetzt.

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## Beispiel: Mediengroßhandel

- Verkauf von Büchern, Filmen, Tonträgern, elektronischen Artikeln (mp3)

# KONZEPTIONELLER ENTWURF MIT ER-MODELL

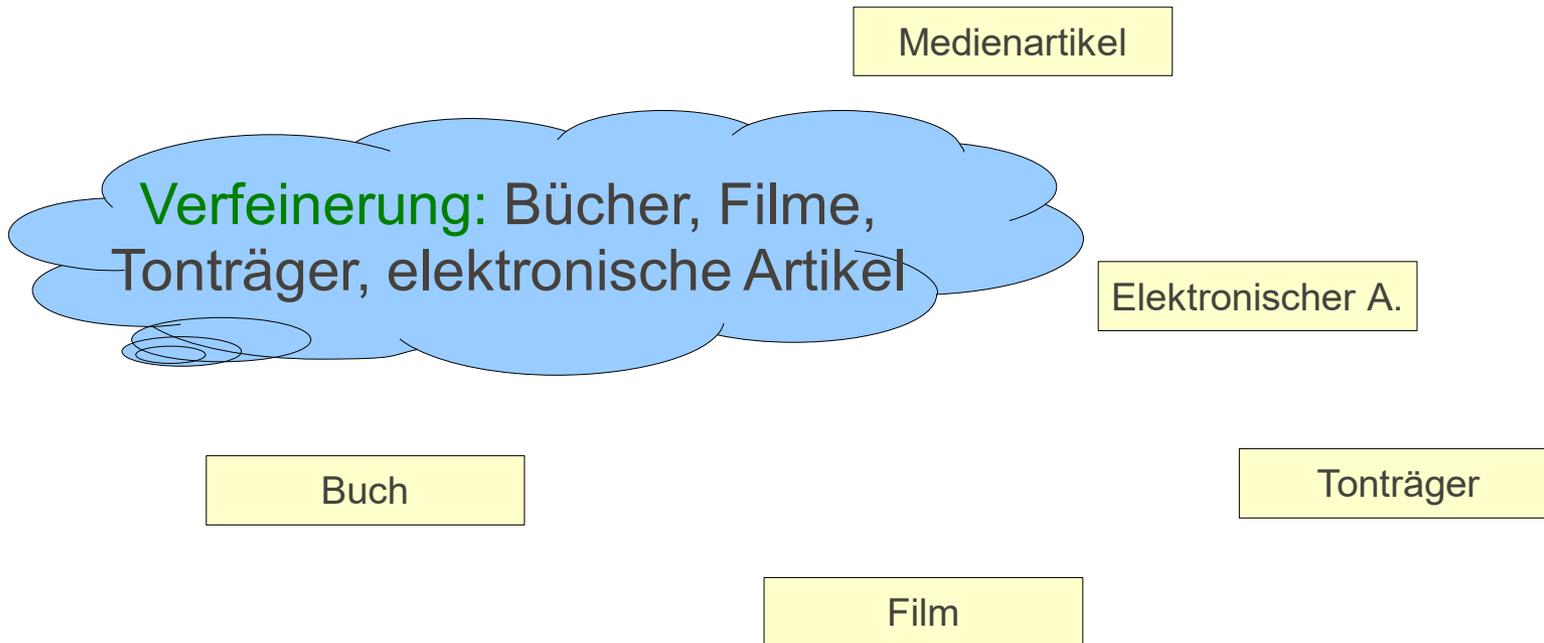


Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

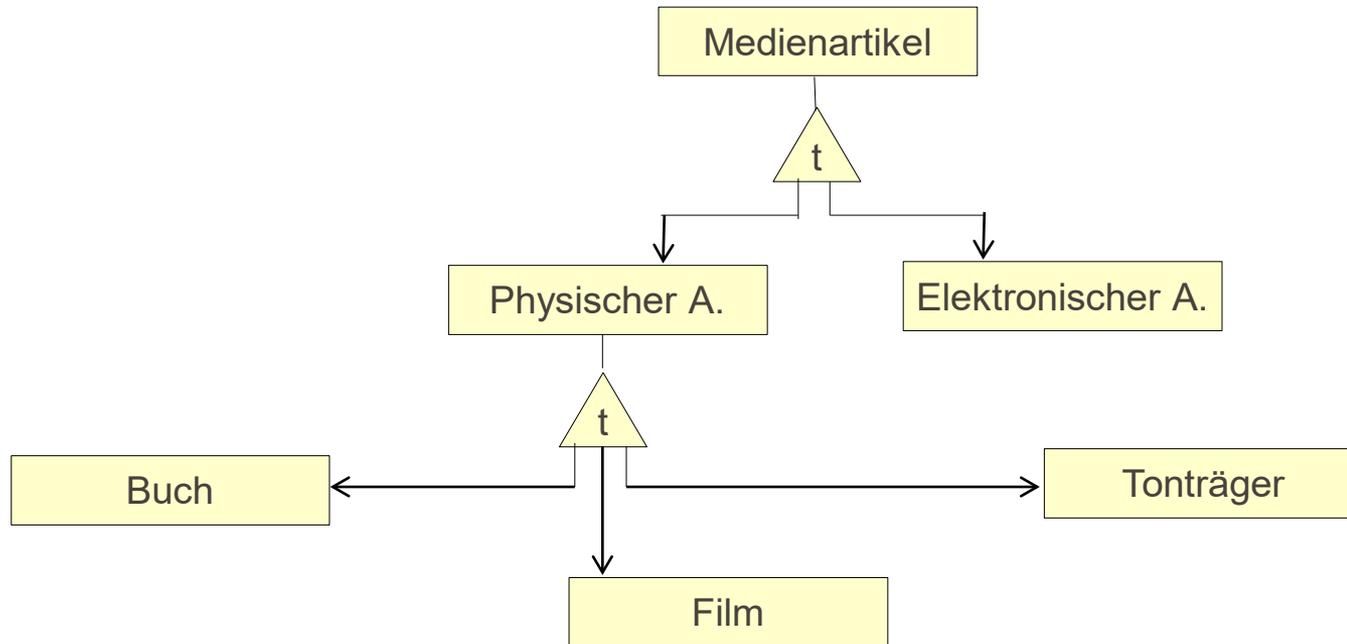
Medienartikel

Verkauf von Medienartikeln

# KONZEPTIONELLER ENTWURF MIT ER-MODELL

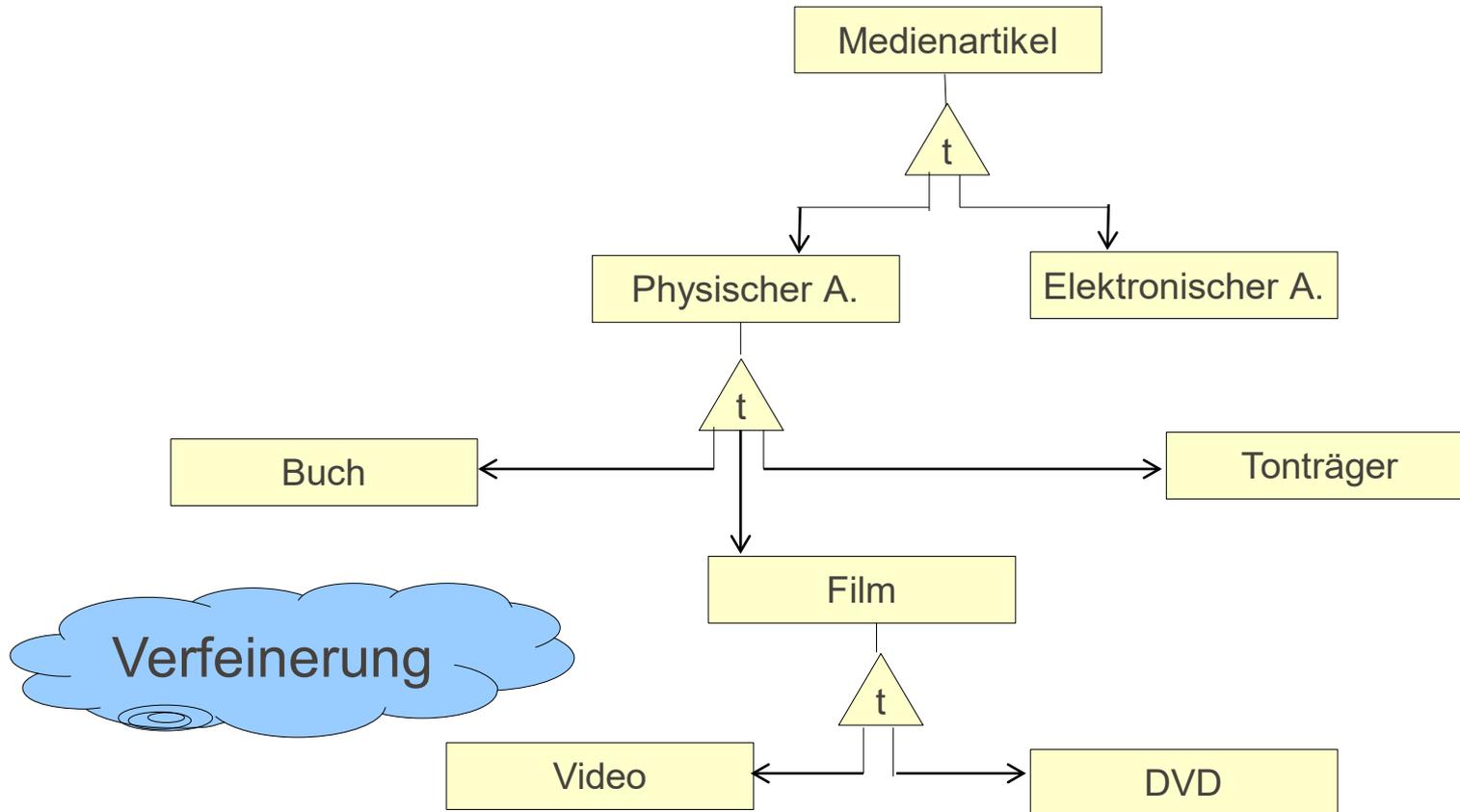


# KONZEPTIONELLER ENTWURF MIT ER-MODELL

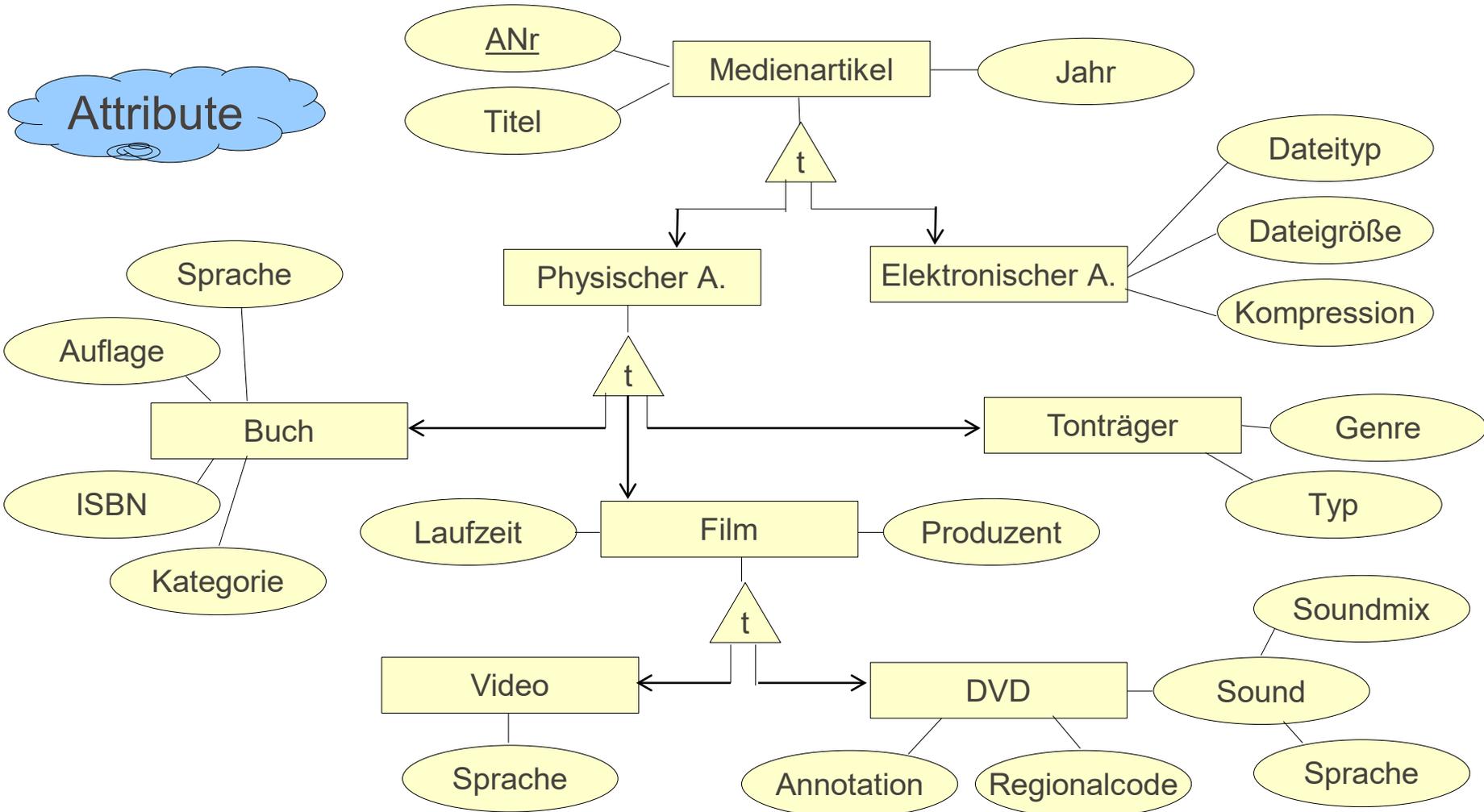


Relationen hinzufügen,  
zusätzliche Entities

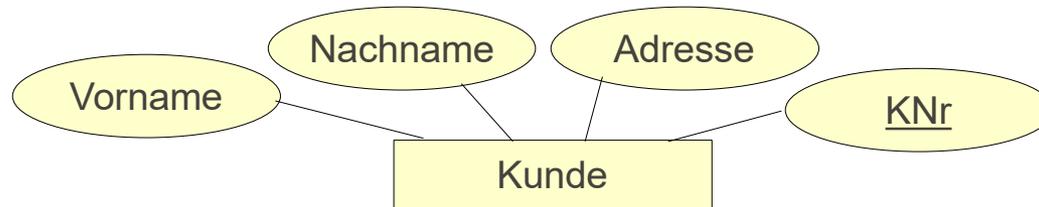
# KONZEPTIONELLER ENTWURF MIT ER-MODELL



# KONZEPTIONELLER ENTWURF MIT ER-MODELL

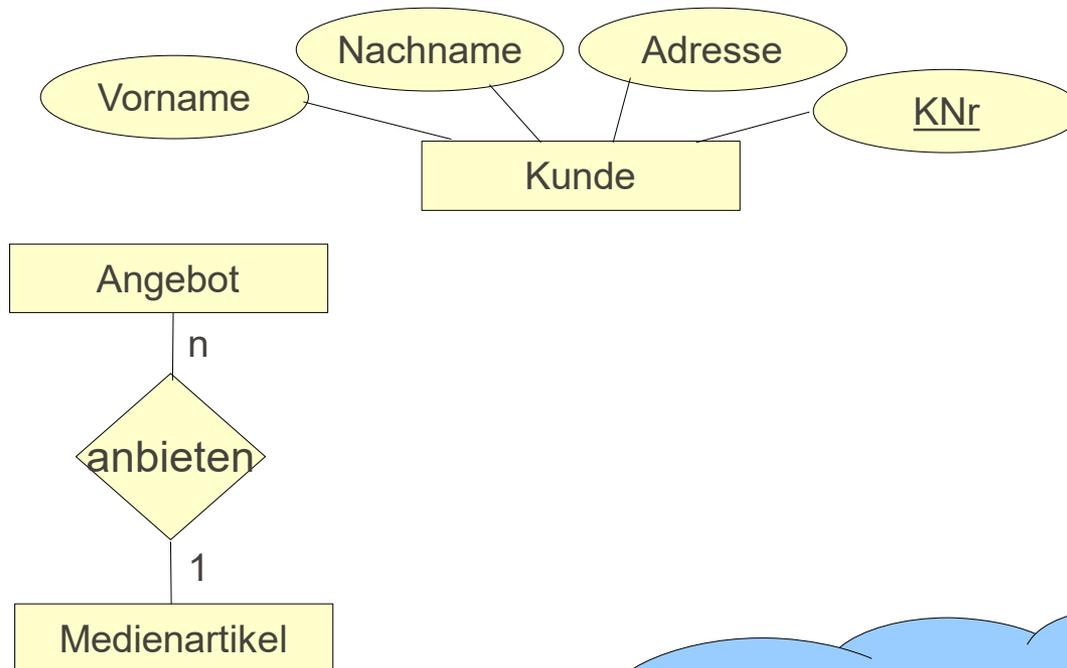


# KONZEPTIONELLER ENTWURF MIT ER-MODELL



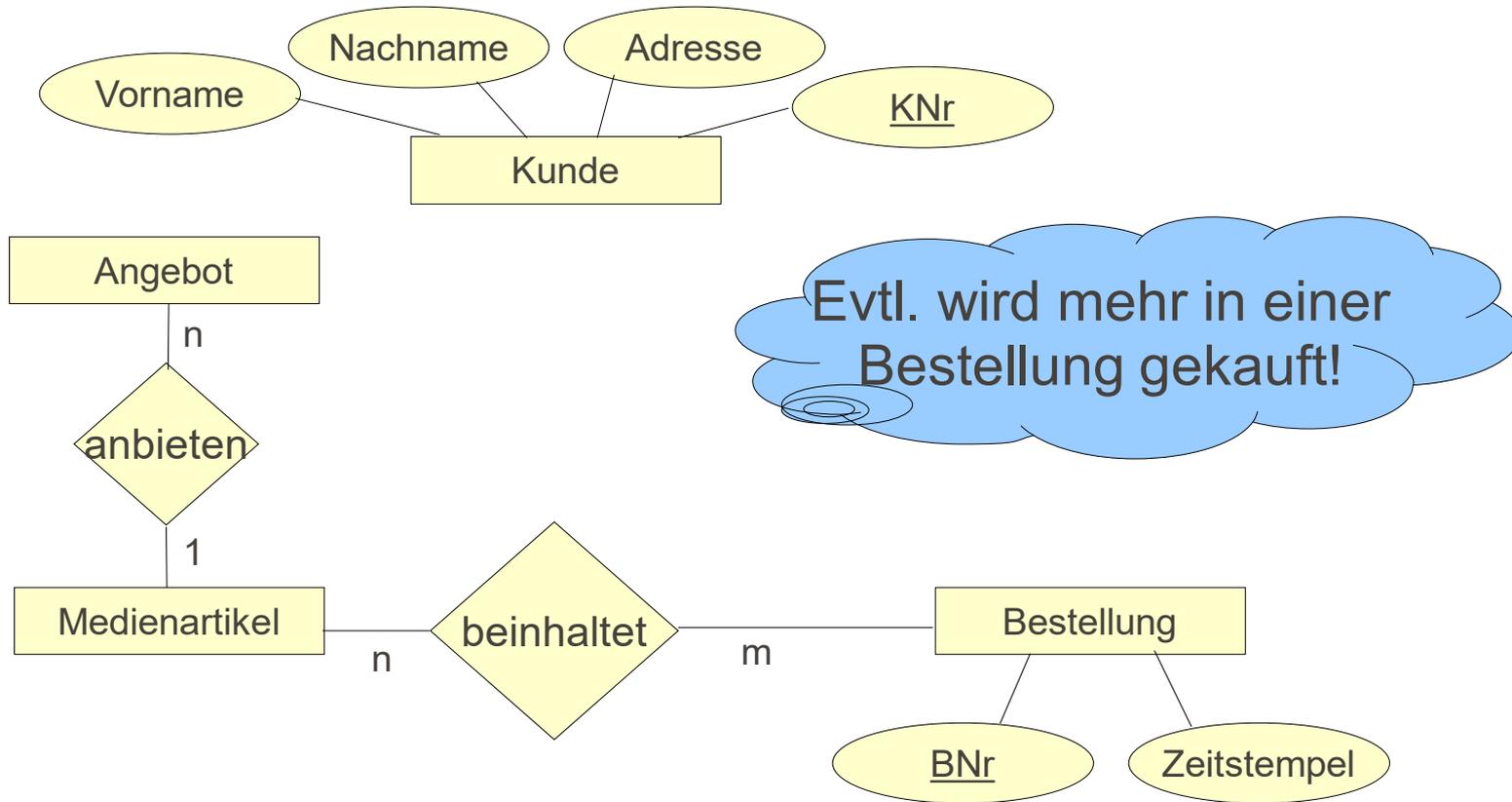
Kunden  
mit Attributen

# KONZEPTIONELLER ENTWURF MIT ER-MODELL

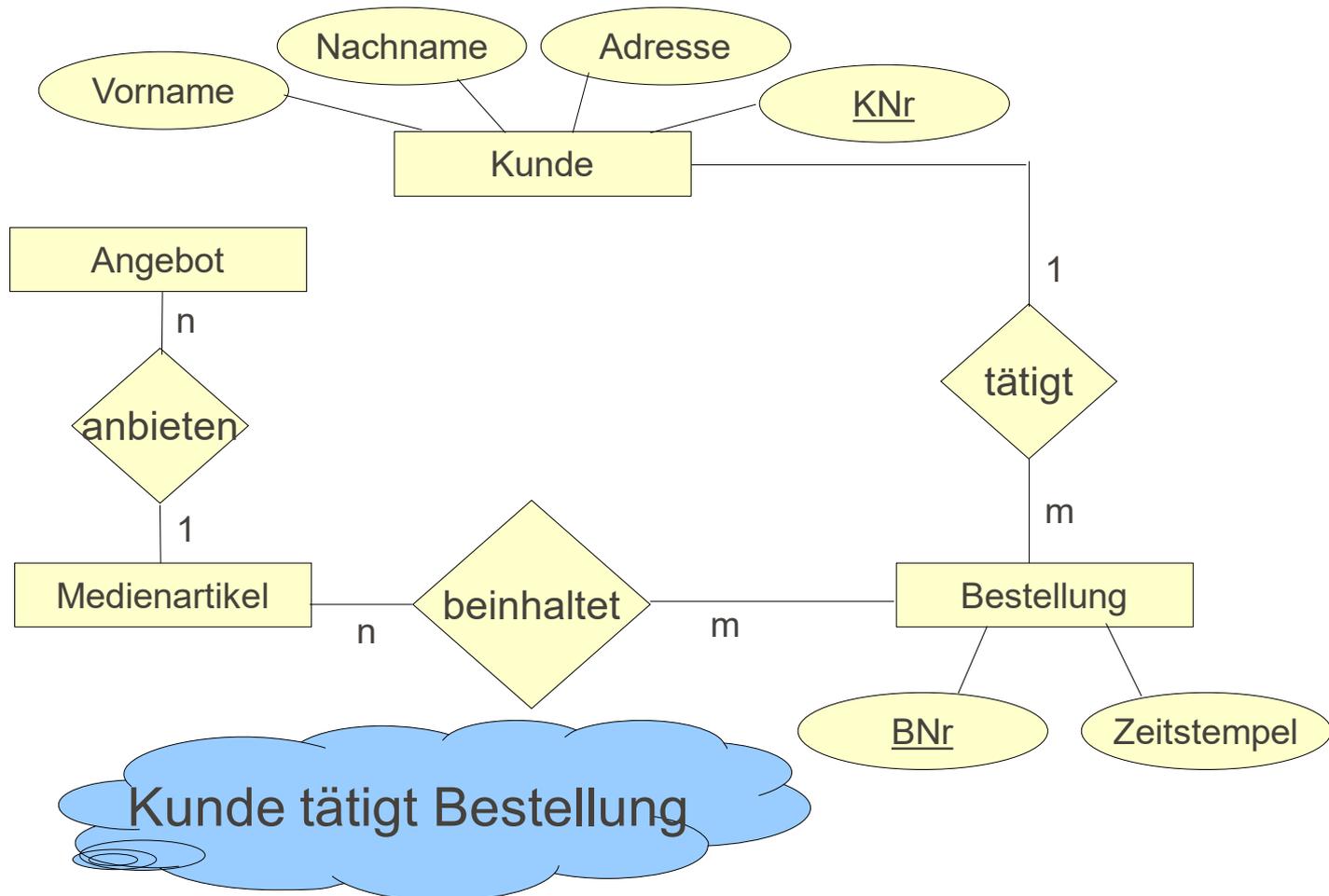


Einem Kunden wird ein Artikel angeboten

# KONZEPTIONELLER ENTWURF MIT ER-MODELL



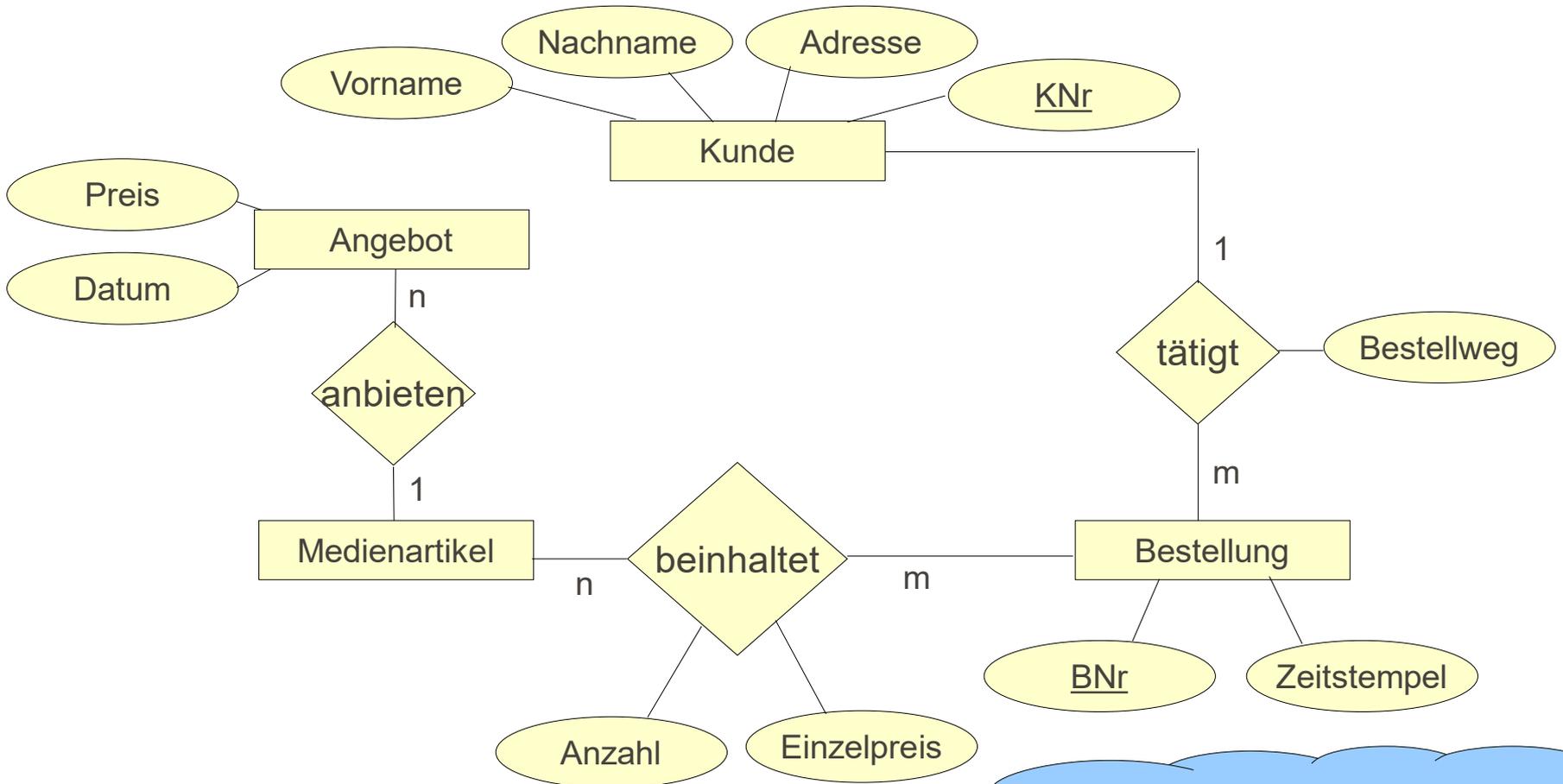
# KONZEPTIONELLER ENTWURF MIT ER-MODELL



# KONZEPTIONELLER ENTWURF MIT ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim



Attribute für Relationships



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# QUALITÄTSMERKMALE

# QUALITÄTSMERKMALE BEI ER-MODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- **Vollständigkeit:** nur durch genauen Vergleich mit gegebenen Anwendung
- **Korrektheit:**
  - **syntaktisch:** Definitionen und Festlegungen in zulässiger Weise genutzt,
  - **semantisch:** Konzepte (Entity, Relation, Attribut) gemäß Ihrer Definition angewendet (*Häufiger Fehler:* Verwendung eines Attributes anstelle einer Entity)
- **Minimalität:** nur auf informelle Weise prüfbar (Können bestimmte Werte aus anderen abgeleitet werden?)

# QUALITÄTSMERKMALE BEI ER-MODELL



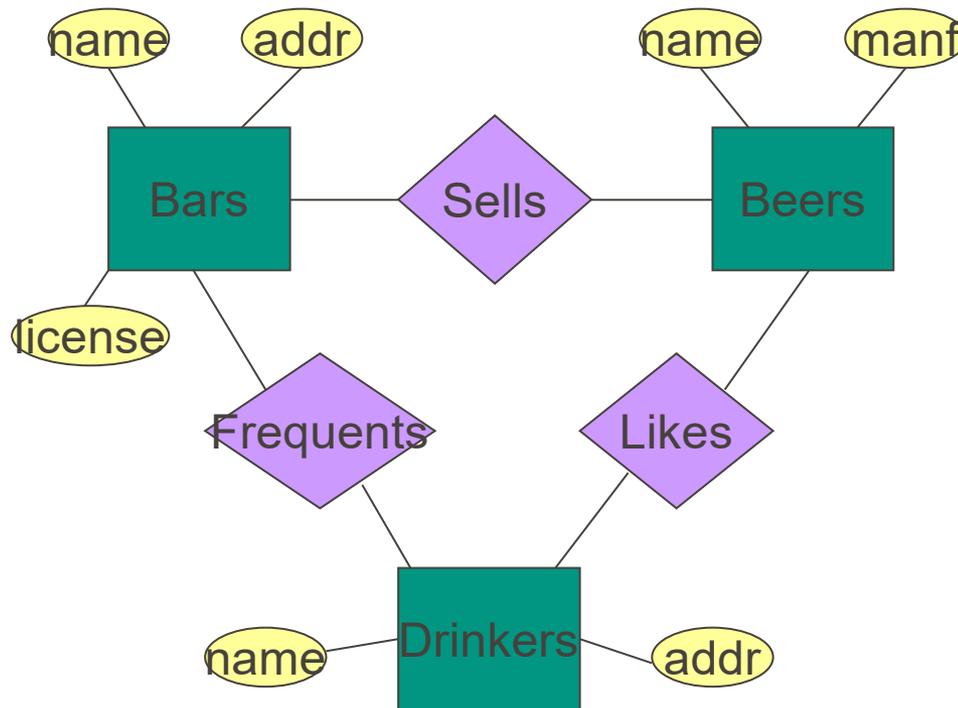
- **Lesbarkeit:** Ästhetische Kriterien
  - Rechtecke und Rauten gleich groß, Kanten horizontal oder vertikal
  - Spezialisierung beginnend mit allgemeinem oben
  - Symmetrien betonen
  - Kreuzungsfrei
  - Wahl der Bezeichner
- **Modifizierbarkeit:**
  - Dokumentation
  - Größere Einheiten identifizierbar
  - Teildiagramme



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# WEITERE BEISPIELE

# WEITERE BEISPIELE



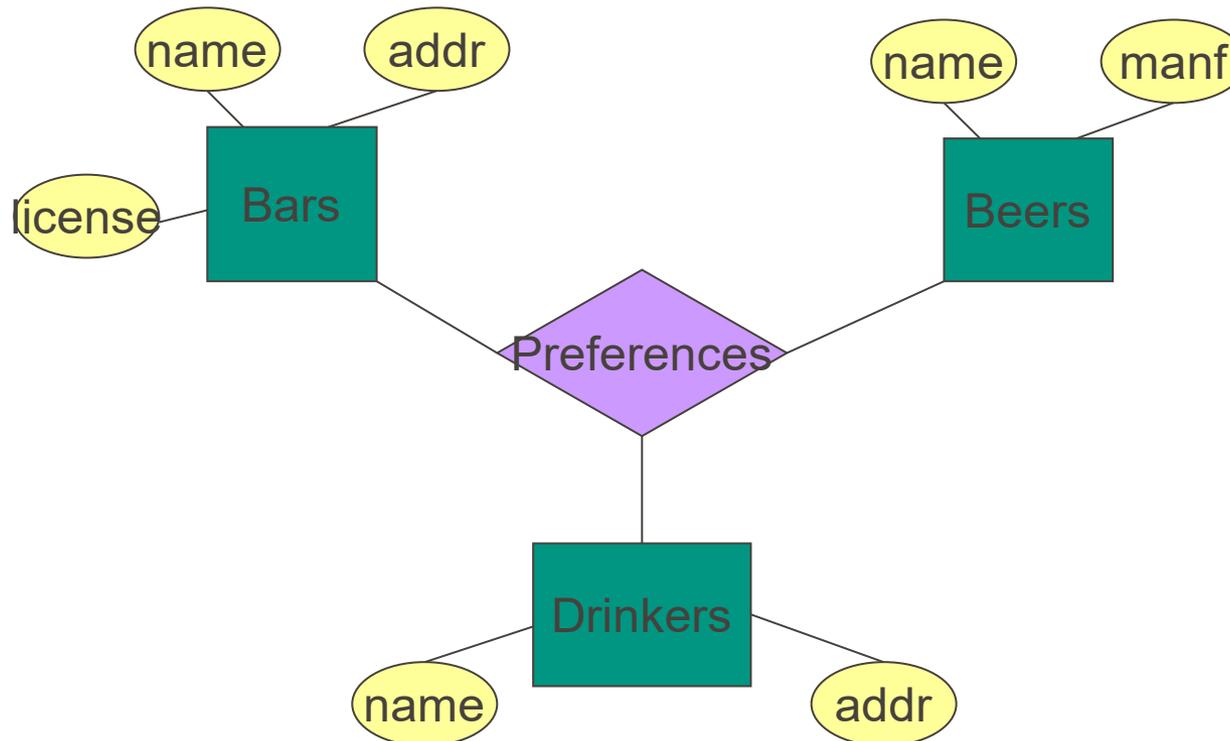
Bars verkaufen Bier.

Drinkers mögen einige Biere.

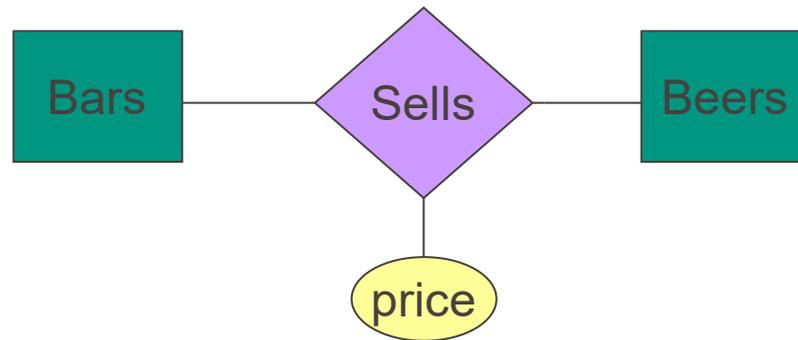
Drinkers besuchen bestimmte Bars.



## BEISPIEL: 3-WEGE BEZIEHUNG

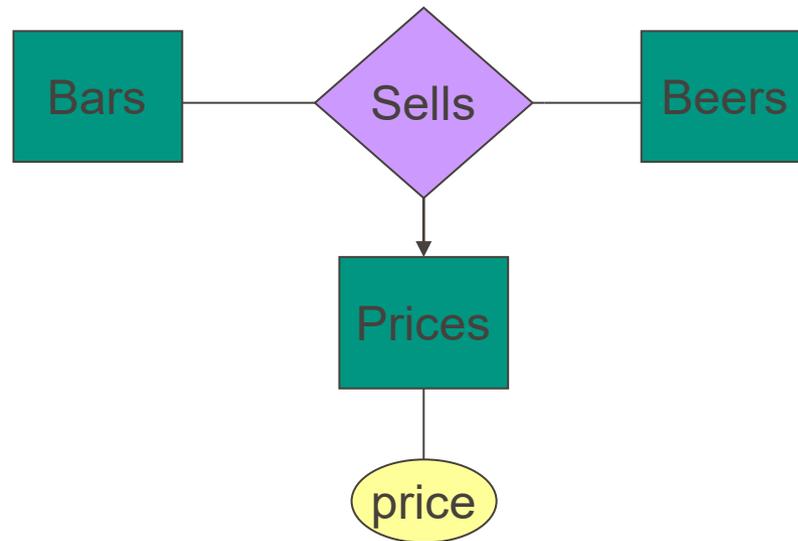


## BEISPIEL: BEZIEHUNG MIT EINEM ATTRIBUTE

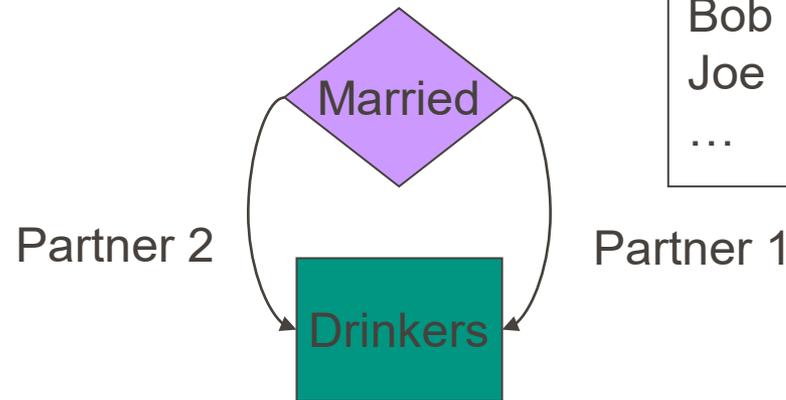


Price ist eine Funktion von sowohl Bar als auch Beer.

# BEISPIEL: ENTFERNEN DES ATTRIBUTES VON DER BEZIEHUNG



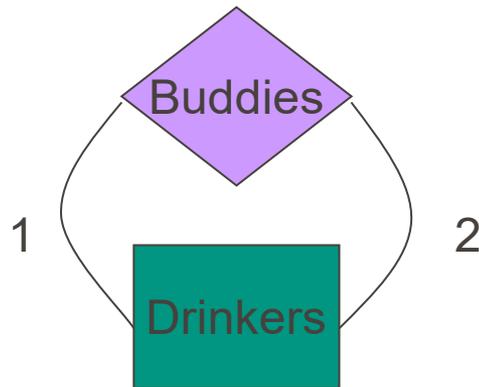
# BEISPIEL: ROLLEN



Relationship Set

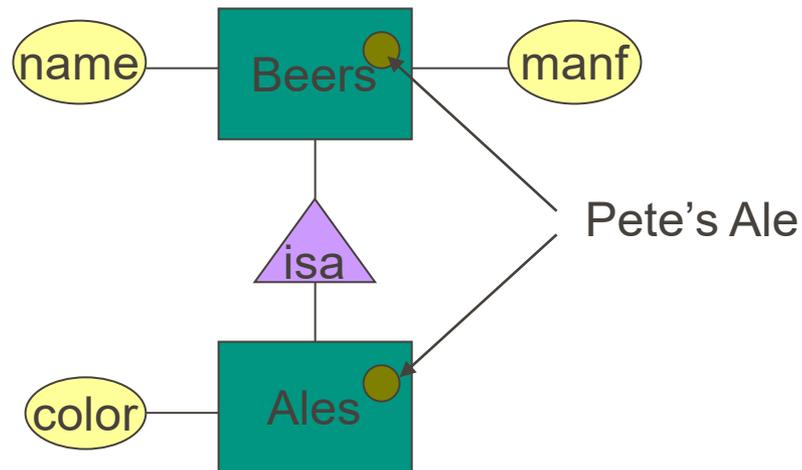
<u>Partner 1</u>	<u>Partner 2</u>
Bob	Ann
Joe	Sue
...	...

# BEISPIEL: ROLLEN



Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
...	...

# BEISPIEL: IS A BEZIEHUNG

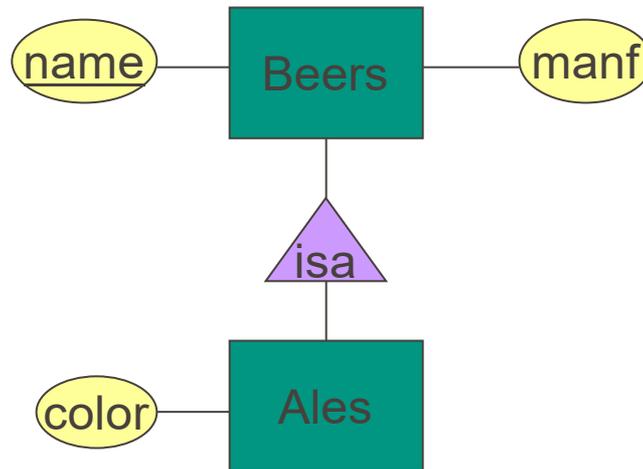




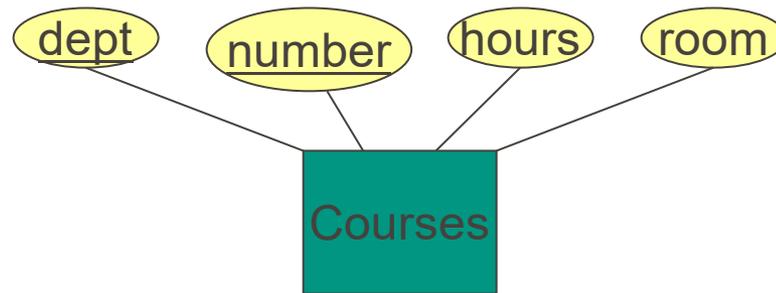
## SCHLÜSSEL IN EINER IS A BEZIEHUNG

- Nur die Hauptklasse (root entity) hat einen Schlüssel, dieser muss für alle Unterklassen in der Hierarchy gelten.

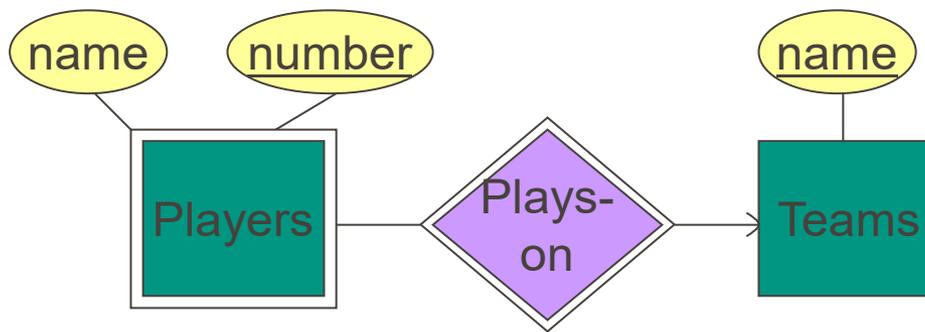
# BEISPIEL: NAME IST DER SCHLÜSSEL FÜR BEERS



# BEISPIEL: EIN MULTI-ATTRIBUTE SCHLÜSSEL



# BEISPIEL: WEAK ENTITY SETS





Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

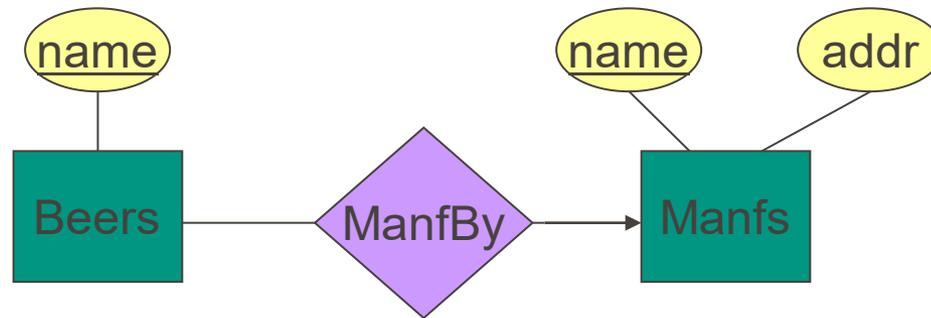
# DESIGN TECHNIKEN



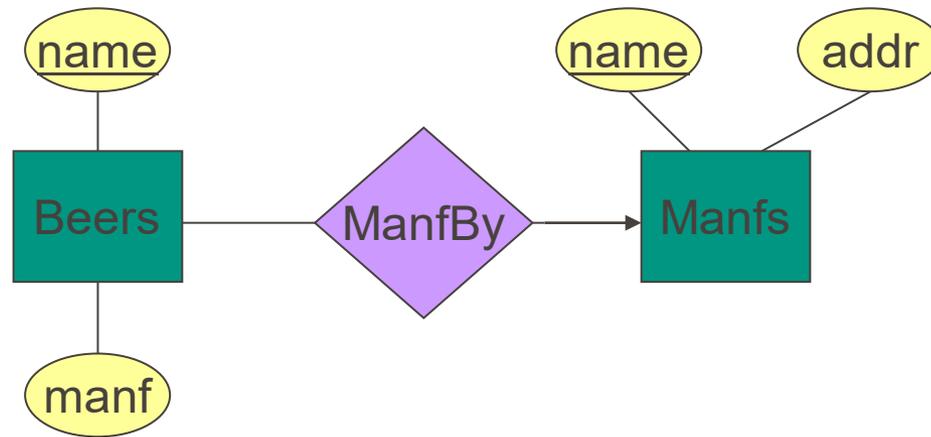
# DESIGN TECHNIKEN

1. VERMEIDEN SIE REDUDANZEN
2. LIMITIEREN SIE SCHWACHE ENTITY SETs
3. BENUTZEN SIE KEIN ENTITY SET, WENN EIN ATTRIBUT REICHEN WÜRD
4. JEDES ENTITY SET SOLLTE MINDESTENS 2 ATTRIBUTE HABEN
5. JEDES ENTITY SET BRAUCHT EINEN SCHLÜSSEL
6. JEDES ENTITY SET BRAUCHT EINE VERBINDUNG
7. WENN MÖGLICH, NATÜRLICHE SCHLÜSSEL WÄHLEN (mehr dazu in der nächsten Vorlesung)

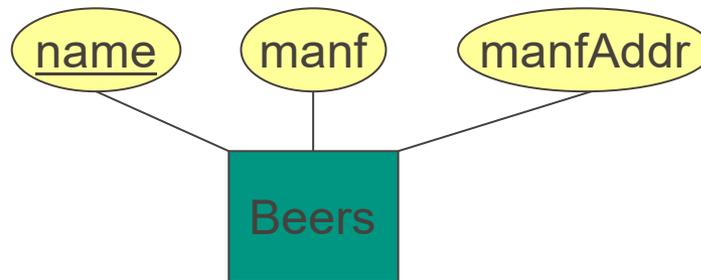
# BEISPEL: GUTES DESIGN



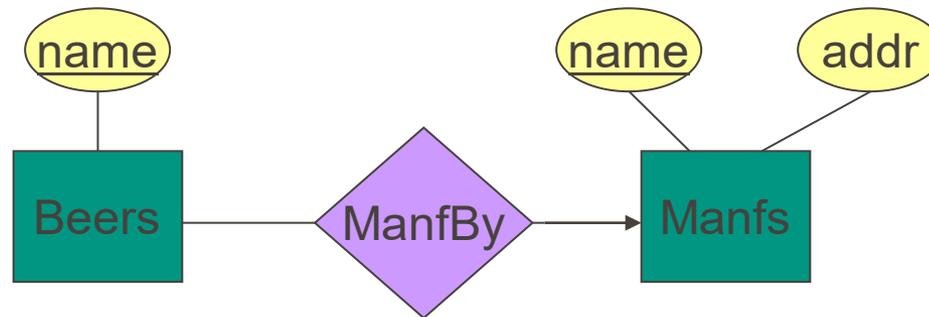
# BEISPIEL: SCHLECHTES DESIGN



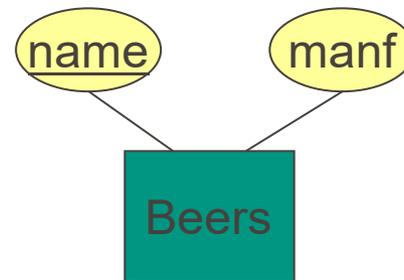
# BEISPIEL: SCHLECHTES DESIGN



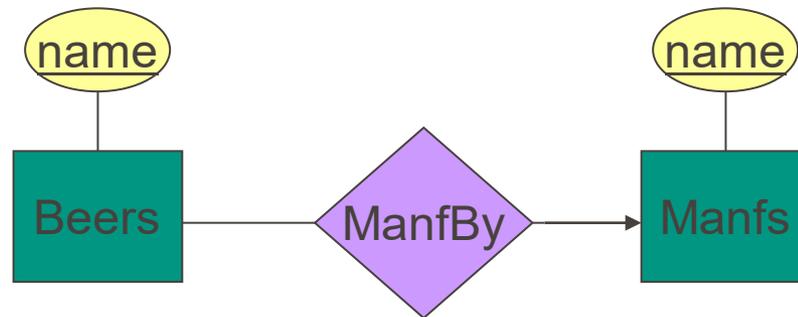
# BEISPIEL: GUTES DESIGN



# BEISPIEL: GUTES DESIGN



# BEISPIEL: SCHLECHTES DESIGN





Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

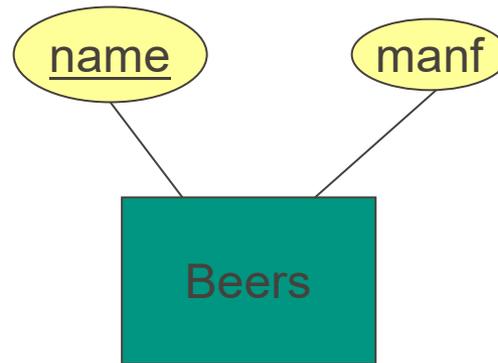
# VOM ER-DIAGRAM ZU RELATIONEN



# VOM ER-DIAGRAM ZU RELATIONEN

- Entity Set -> Relation
  - Attribute -> Attribute.
- Beziehungen -> Relation mit folgenden Attributen:
  - Die Schlüsselattribute der verbundenen Entity Sets
  - Die Beziehungsattribute

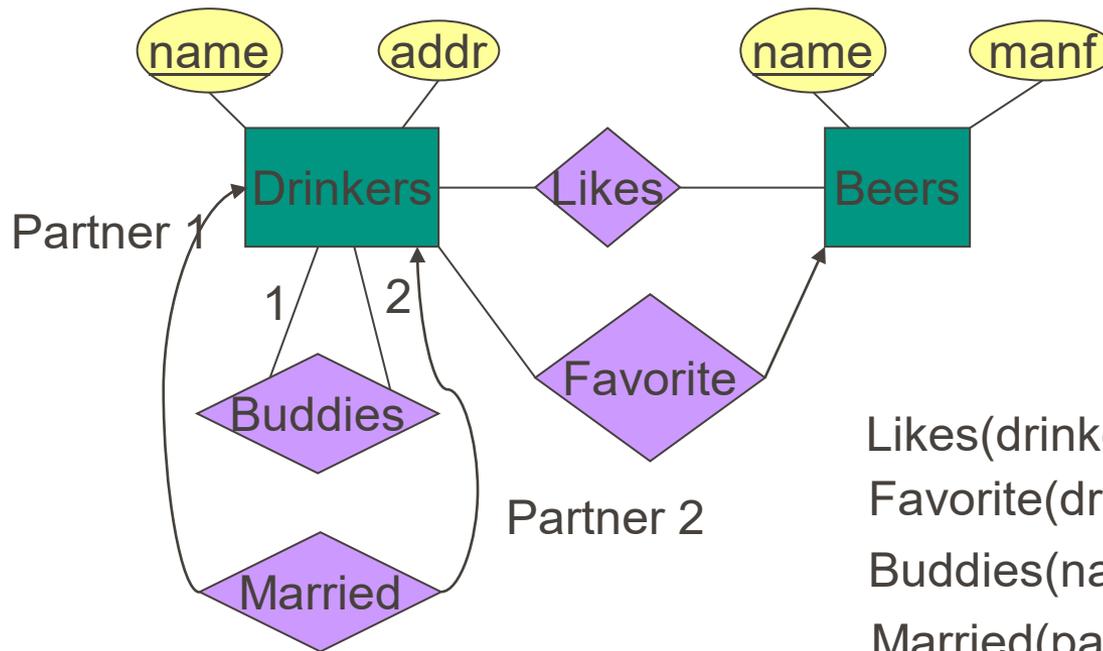
# ENTITY SET -> RELATION



Relation: Beers(name, manf)



# RELATIONSHIP -> RELATION



Likes(drinker, beer)

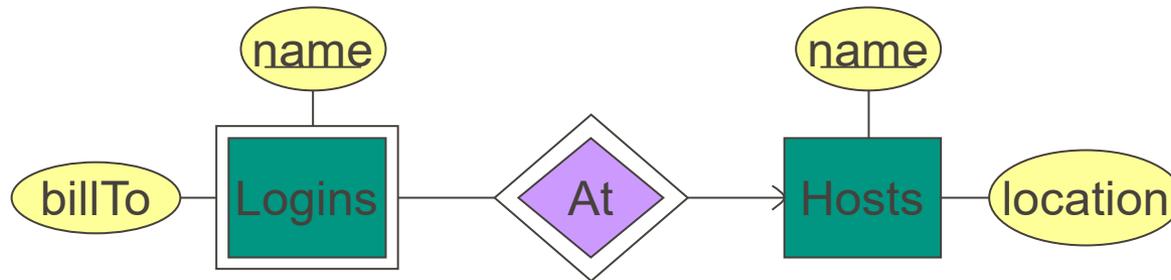
Favorite(drinker, beer)

Buddies(name1, name2)

Married(partner1, partner2)



# BEISPIEL: WEAK ENTITY SET -> RELATION



Hosts(hostName, location)

Logins(loginName, hostName, billTo)

~~At(loginName, hostName, hostName2)~~

At wird Teil von  
Logins

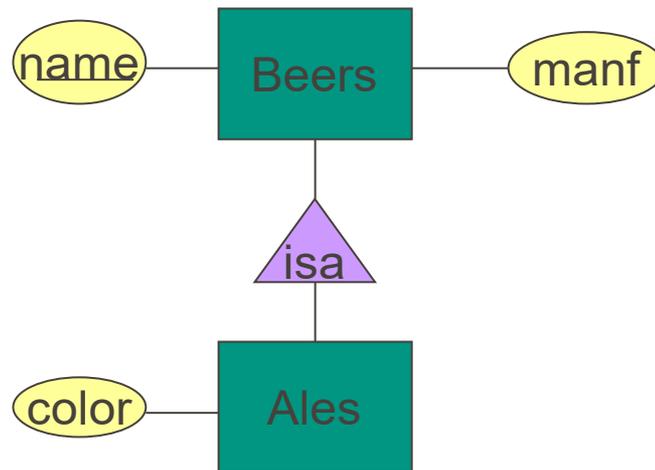
Muss das Gleiche sein



## SUBCLASSES: 2 MÖGLICHKEITEN

1. *Object-oriented* : Eine Relation per Teilmenge der Subklassen, mit allen relevanten Attributen.
2. *NULLS*: Eine Relation; Entities haben NULL-Werte in den Attributen, welche nicht zu ihnen gehören.
3. *E/R style* : Eine Relation für jede Subklasse:
  - Schlüssel Attribut(e)
  - Attribute der Subklasse

# BEISPIEL: SUBCLASS -> RELATIONS





# OBJECT-ORIENTED

- Beers

name	manf
Bud	Anheuser- Busch

- Ales

name	manf	color
Summerbrew	Pete's	dark

- Gut, wenn man Abfragen hat wie: „Finde die Farben von Ales die von Pete's hergestellt werden“



## ER STYLE

- Beers

name	manf
Bud	Anheuser- Busch
Summerbrew	Pete's

- Ales

name	color
Summerbrew	dark

- Gut, wenn man Abfragen hat wie: „Finde alle Biere (inkl. Ales) welche von Pete's hergestellt werden“

# USING NULLS



- Beers

name	manf	color
Bud	Anheuser-Busch	NULL
Summerbrew	Pete's	dark

Beers

Benutzt am wenigsten Speicher,  
verwendet aber sehr viele NULL-Werte

# LITERATUR



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Vossen, Gottfried: Datenmodelle, Datenbank-sprachen und Datenbankmanagementsysteme, 5. Auflage, Oldenburg Wissenschaftsverlag, 2008
- Kudraß, Thomas: Taschenbuch Datenbanken, Hanser, 2007
- Database System: The Complete Book