

# **ISDA 02**

## **ER Modellierung**

Prof. Dr. Volker Markl

Folienmaterial von Prof. Dr. Felix Naumann  
und Dr. Ralf-D. Kutsche



Fachgebiet Datenbanksysteme und Informationsmanagement  
Technische Universität Berlin

<http://www.dima.tu-berlin.de/>

- Vorstellung des Fachgebiets
- Organisatorisches
- Datenbanken und Informationssystem
- Das Semester an einem Beispiel
- Ausblick auf das Semester
- Tutorien



Kapitel 1 des Lehrbuchs

- **Motivation und Einbettung**
- Begriffe und Definitionen
- ER-Diagramme
- Modellierung von Nebenbedingungen
- Schwache Entitytypen

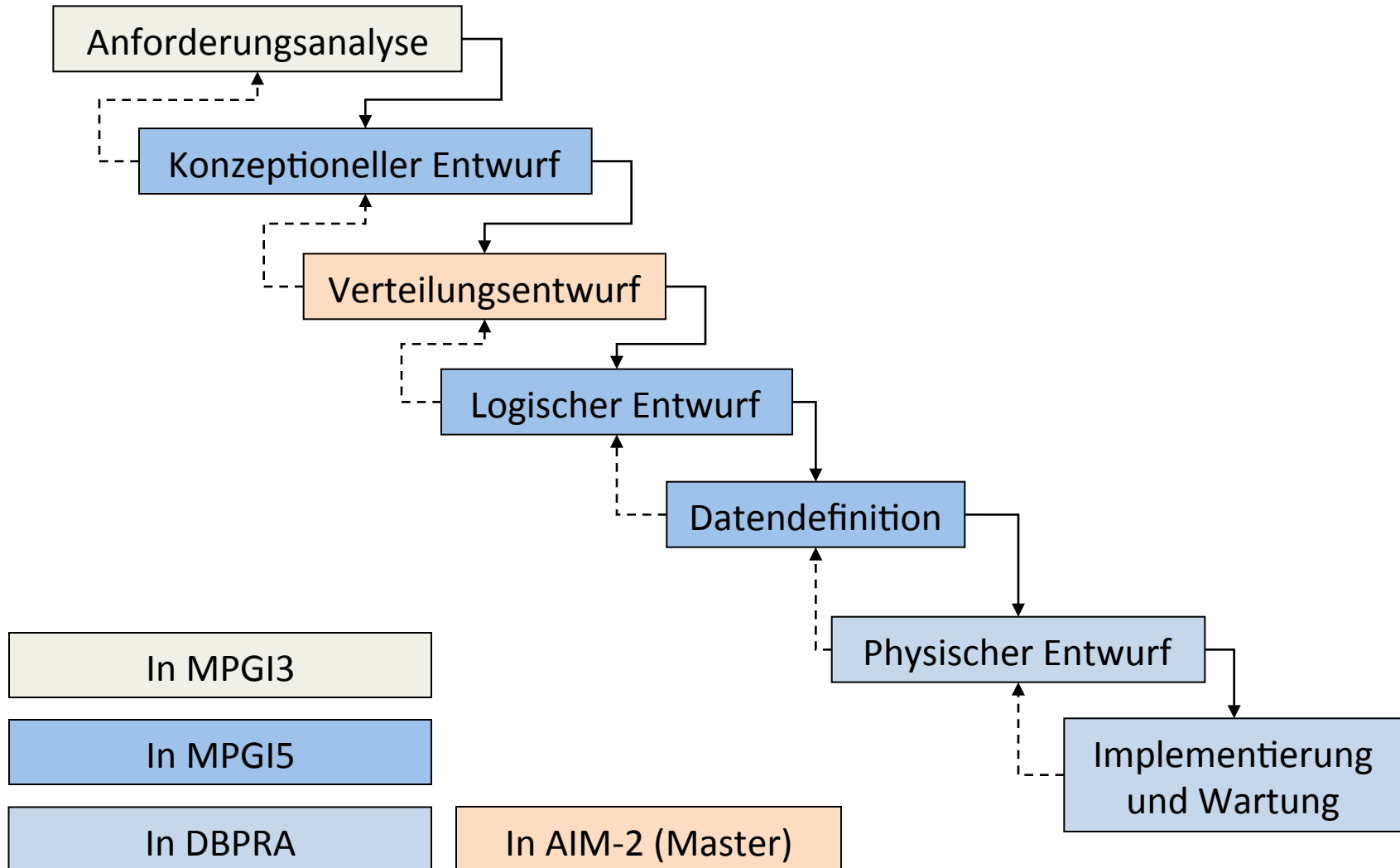


Kapitel 4 des Lehrbuchs

- Datenbankdesign als „Skizze“ (korrekt: Informationsmodell) mittels ER-Modellierung
  - Welche Daten? Welche Beziehungen?
  - Nicht wie Daten erzeugt oder verändert werden.
- Informationsmodelle für relationale Datenbanken meist als „Entity-Relationship-Diagramme“
  - ER-Diagramme
- In 2 Wochen: Überführung von ER-Diagrammen in das relationale Modell
  - Relationen, Attribute, Integritätsbedingungen
- In 4 Wochen: Überführung von Relationenschemata in SQL Ausdrücke
  - **CREATE TABLE ...**

- Datenhaltung für **mehrere** Anwendungssysteme und **mehrere** Jahre
- daher: besondere Bedeutung
- Anforderungen an Entwurf
  - Anwendungsdaten jeder Anwendung sollen aus Daten der Datenbank ableitbar sein.
    - Möglichst effizient
  - Nur „vernünftige“ (wirklich benötigte) Daten sollen gespeichert werden.
    - Auch: Datenschutz
  - Nicht-redundante Speicherung

- Abfolge von Entwurfsdokumenten
  - Von abstrakter Beschreibung
  - Bis tatsächlichen Realisierung in einem DBMS
  - Verschiedene Beschreibungsformalismen
    - ER, Relationenmodell, SQL DDL, usw.
- In jedem Schritt
  - Informationserhaltung
  - Konsistenzerhaltung

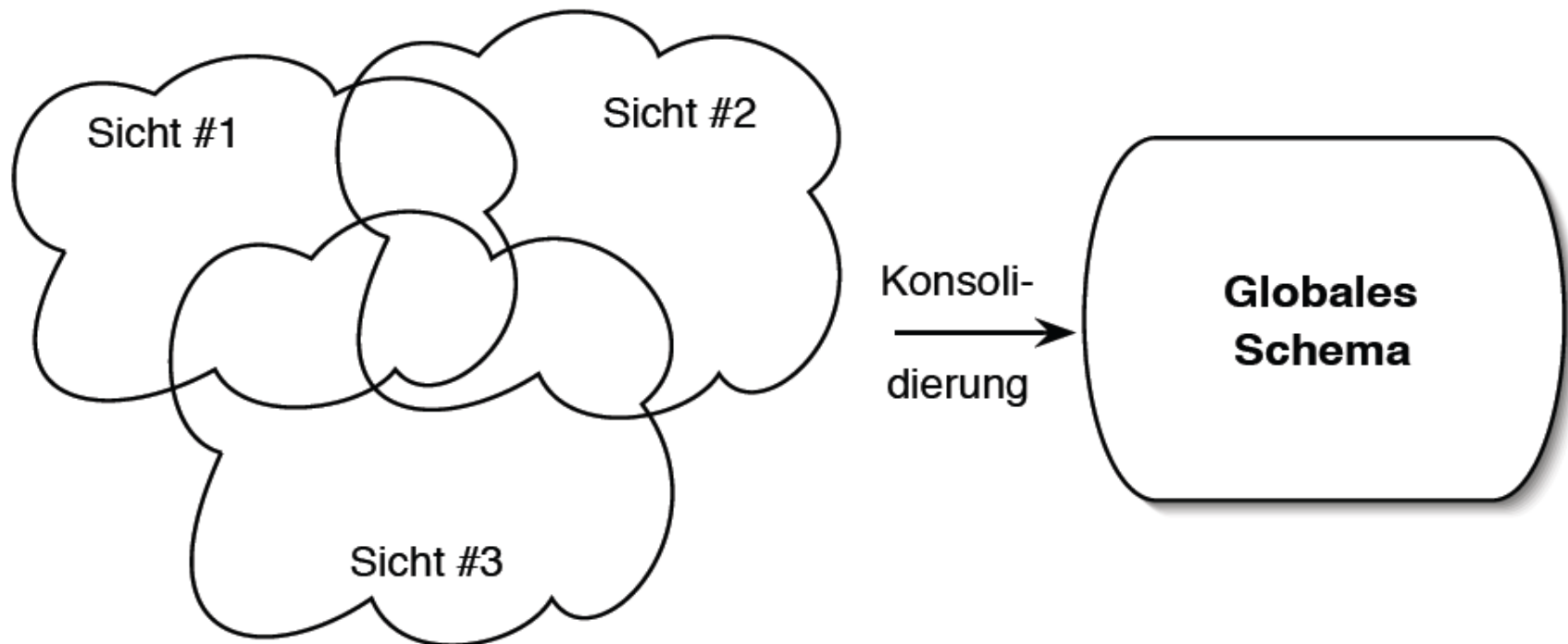


- Vorgehensweise
  - Sammlung des Informationsbedarfs in den Fachabteilungen
- Ergebnis
  - informale Beschreibung des Fachproblems
    - Texte, tabellarische Aufstellungen, Formblätter, ...
  - Trennen der Information über Daten (Datenanalyse) von den Information über Funktionen (Funktionsanalyse)
- „Klassischer“ DB-Entwurf
  - nur Datenanalyse und Folgeschritte
- Funktionsentwurf
  - siehe Methoden des Software Engineering (MPGI3)



- Erste formale Beschreibung des Fachproblems
  - UoD: Universe of Discourse (Diskursbereich)
- Sprachmittel: semantisches Datenmodell
  - ER
- Vorgehensweise
  - Modellierung von Sichten z.B. für verschiedene Fachabteilungen
  - Analyse der vorliegenden Sichten in Bezug auf Konflikte
    - Namenskonflikte (Synonyme, Homonyme)
    - Typkonflikte
    - Bedingungskonflikte
    - Strukturkonflikte
  - Integration der Sichten in ein Gesamtschema
- Ergebnis
  - konzeptionelles Gesamtschema, z.B.(E)ER-Diagramm

- Analyse der vorliegenden Sichten in Bezug auf Konflikte
- Integration der Sichten in ein Gesamtschema



- Namenskonflikte
  - Homonyme: Schloss; Hahn; Kunde
  - Synonyme: Auto, KFZ, Fahrzeug
- Typkonflikte
  - verschiedene Strukturen für das gleiche Element
- Wertebereichskonflikte
  - verschiedene Wertebereiche für ein Element
- Bedingungskonflikte
  - z.B. verschiedene Schlüssel für ein Element
- Strukturkonflikte
  - gleicher Sachverhalt durch unterschiedliche Konstrukte ausgedrückt

- Sollen Daten auf mehreren Rechnern verteilt vorliegen, muss Art und Weise der verteilten Speicherung festgelegt werden.
- z.B. bei einer Relation
  - **KUNDE (KNr, Name, Adresse, PLZ, Konto)**
  - horizontale Verteilung
    - **KUNDE\_1 (KNr, Name, Adresse, PLZ, Konto)**  
**where PLZ < 50.000**
    - **KUNDE\_2 (KNr, Name, Adresse, PLZ, Konto)**  
**where PLZ >= 50.000**
  - vertikale Verteilung (Verbindung über KNr Attribut)
    - **KUNDE\_Adr (KNr, Name, Adresse, PLZ)**
    - **KUNDE\_Konto (KNr, Konto)**

- Sprachmittel: Datenmodell des ausgewählten „Realisierungs“-DBMS
  - z.B. DB2, Informix => relationales Modell
  - z.B. Tamino => XML
- Vorgehensweise:
  - (automatische) Transformation des konzeptionellen Schemas
    - z.B. ER in relationales Modell
  - Verbesserung des relationalen Schemas anhand von Gütekriterien
    - Normalisierung, Redundanzvermeidung, ...
- Ergebnis: logisches Schema, z.B. Sammlung von Relationenschemata
- Siehe nächsten Foliensatz: Relationaler Entwurf

- Umsetzung des logischen Schemas in ein konkretes Schema
- Sprachmittel:
  - DDL und DML eines DBMS
    - z.B. Oracle, DB2, SQL Server
  - Datenbankdeklaration in der DDL des DBMS
  - Realisierung der Integritätssicherung
  - Definition der Benutzersichten

**CREATE TABLE ...**  
**CREATE VIEW ...**

- Ergänzen des physischen Entwurfs um Zugriffsunterstützung zur Effizienzverbesserung
  - z.B. Definition von Indizes
  - **CREATE INDEX ...**
- Index
  - Zugriffspfad: Datenstruktur für zusätzlichen, schlüsselbasierten Zugriff auf Tupel (<Schlüsselattributwert, Tupeladresse>)
  - meist als B\*-Baum realisiert
- Beispiel
  - Tabelle mit 10 GB Daten, Festplattentransferrate ca. 10 MB/s
  - Operation: Suchen eines Tupels (Selektion)
  - Implementierung: sequentielles Durchsuchen
  - Aufwand:  $10.240/10 = 1.024 \text{ sec.} = 17 \text{ min.}$

Wird im Datenbankpraktikum behandelt

- Wartung des DBMS
  - Parameter, Festplatten, etc.
- Weitere Optimierung der physischen Ebene
- Anpassung an neue Anforderungen
- Anpassung an neue Systemplattformen
- Portierung auf neue Datenbankmanagementsysteme
  
- Kostenaufwändigste Phase
- Software Engineering

Wird im Datenbankpraktikum behandelt



- Motivation und Einbettung
- **Begriffe und Definitionen**
- ER-Diagramme
- Modellierung von Nebenbedingungen
- Schwache Entitytypen



- Nach Peter P. Chen 1976
  - The entity-relationship model – towards a unified view of data. ACM TODS
- Standardmodell in der frühen Entwurfsphase

## The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN

Massachusetts Institute of Technology

---

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

**Key Words and Phrases:** database design, logical view of data, semantics of data, data models, entity-relationship model, relational model, Data Base Task Group, network model, entity set model, data definition and manipulation, data integrity and consistency

**CR Categories:** 3.50, 3.70, 4.33, 4.34

---

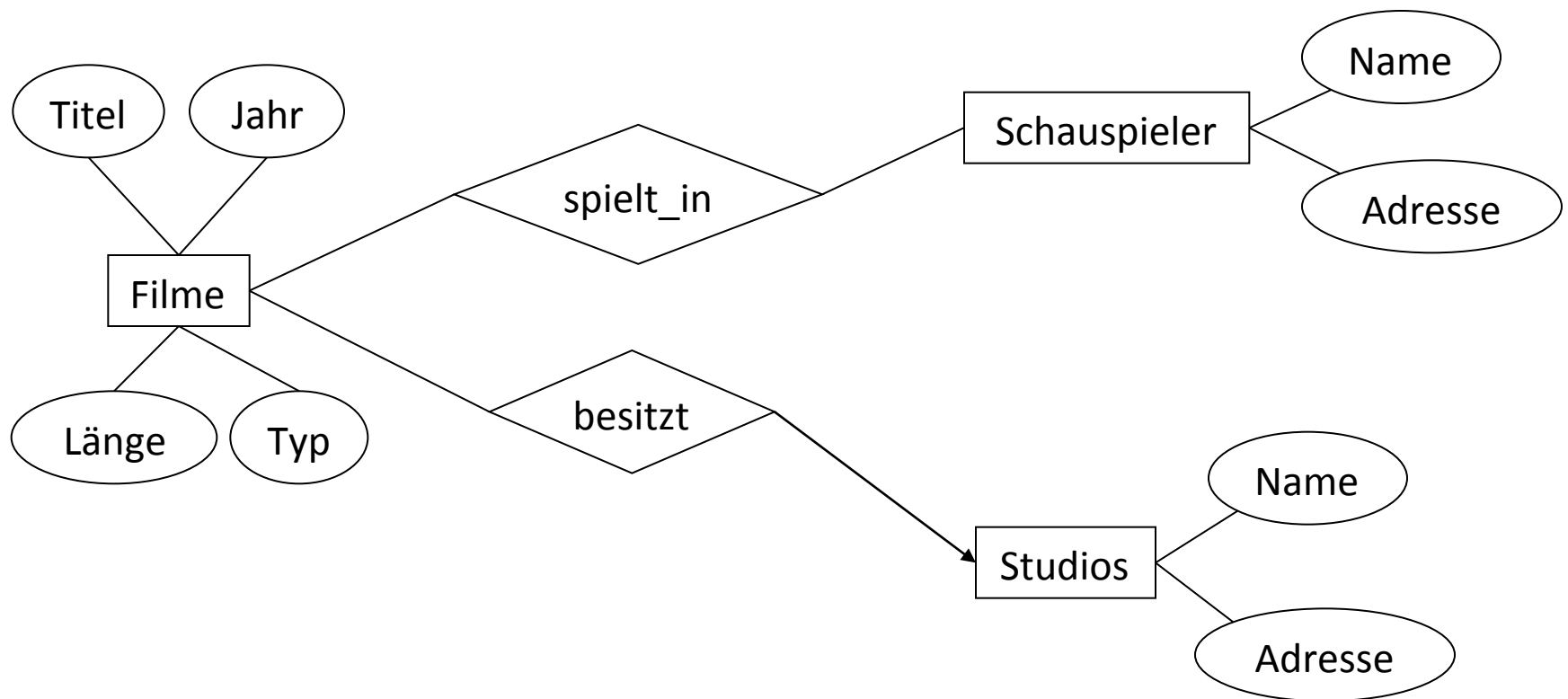


- Entity
  - Ein Ding / Objekt der realen oder der Vorstellungswelt
  - Nicht direkt darstellbar, sondern nur über Eigenschaften beobachtbar
- Entitytyp (*entity set*)
  - Eine Klasse (Menge) für gleichartige Objekte
- Relationship
  - Beschreibt Beziehungen zwischen Entities
  - Meist binär
- Relationshiptyp
  - Eine Klasse für gleichartige Beziehungen
- Attribut
  - repräsentiert eine Eigenschaft von Entities oder von Relationships
  - Zunächst nur primitive Datenwerte (String, Integer, ...) und Operationen darauf
    - Später auch komplexe Attribute

- Entitytyp: Rechteck
- Relationshiptyp: Raute
- Attribut: Oval (oder: „Nadel“-Notation)
- Kanten verbinden Entitytypen mit Attributen
- Kanten verbinden Entitytypen mit Relationshiptypen

- Motivation und Einbettung
- Begriffe und Definitionen
- **ER-Diagramme**
- Modellierung von Nebenbedingungen
- Schwache Entitytypen
- Erweitertes ER-Modell
- Designprinzipien
- Sichtintegration

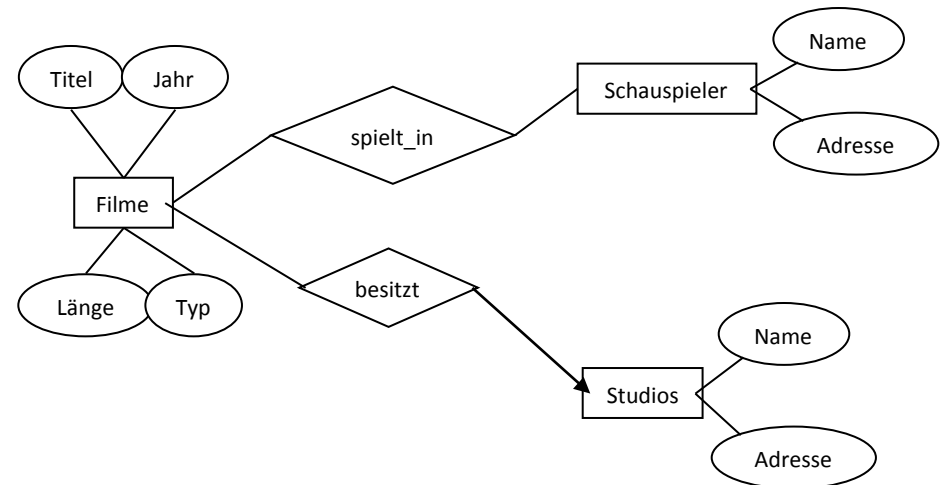




- ER-Diagramme beschreiben Datenbankschemata, die eine oder viele Instanzen haben (werden).
- Der „Wert“ eines Entitytypen ist die (endliche) Menge der zugehörigen Entities.
- Jeder Entity hat bestimmte Werte für seine Attribute.
- Die Instanz eines  $n$ -ären Relationshiptypen ist eine Menge von Listen der Länge  $n$ .
- Dies alles ist nur abstrakte Denkhilfe.
  - Modellierung im relationalen Modell unterscheidet sich.
  - Speicherung in einem DBMS unterscheidet sich.

Filme

Titel	Jahr	Länge	Typ
Basic Instinct	1992	127	Farbe
Total Recall	1990	113	Farbe
Dead Man	1995	121	s/w



Schauspieler

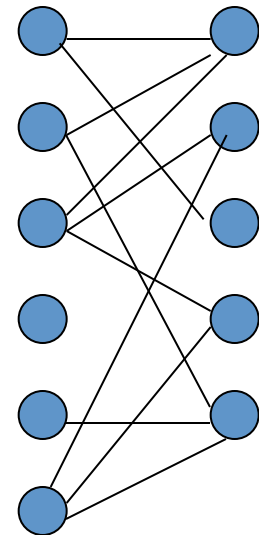
Name	Adresse
Sharon Stone	Hollywood
Arnold Schwarzenegger	Sacramento
Johnny Depp	Paris

spielt\_in

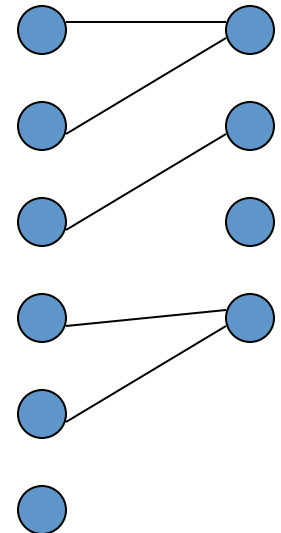
Name	Titel
Sharon Stone	Total Recall
Sharon Stone	Basic Instinct
Arnold Schwarzenegger	Total Recall
Johnny Depp	Dead Man



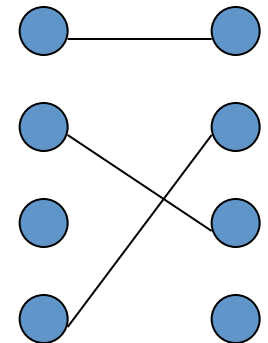
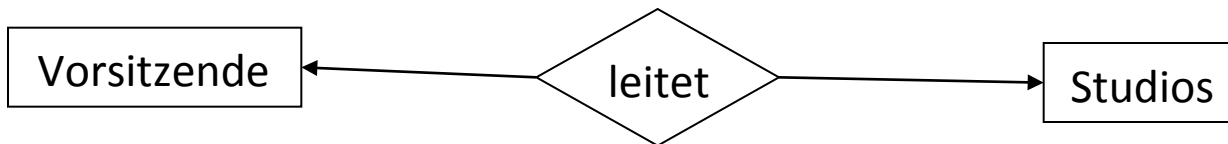
- Allgemein: Ein binärer Relationship kann beliebig viele Entities des einen Typen mit beliebig vielen des anderen Typen verbinden.
  - Ein Schauspieler kann in mehreren Filmen spielen.
  - In einem Film spielen mehrere Schauspieler.
  
- „Kann“: D.h. nicht jeder Entity muss mit einem anderen verbunden sein.
  - In einem Animationsfilm spielen keine Schauspieler.
  
- m:n Beziehung
  
- Einschränkungen („Spezialfälle“)
  - 1:n Beziehung
  - 1:1 Beziehung



- Ein Entity vom Typ  $E$  kann mit beliebig vielen Entities des Typs  $F$  verbunden sein.
- Ein Entity vom Typ  $F$  kann mit höchstens einem Entity des Typs  $E$  verbunden sein.
- Beispiel
  - Ein Studio kann die Rechte an mehreren Filmen besitzen.
  - Ein Film kann nur von einem Studio besessen werden.
- „Kann“: D.h. nicht jeder Entity muss mit einem anderen verbunden sein.
  - Ein neues Studio besitzt noch keinen Film.
- Darstellung mittels eines Pfeils zur „1er“ Seite.



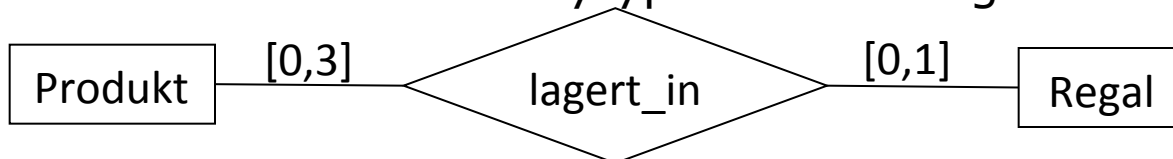
- Ein Entity vom Typ  $E$  kann mit höchstens einem Entity des Typs  $F$  verbunden sein.
- Ein Entity vom Typ  $F$  kann mit höchstens einem Entity des Typs  $E$  verbunden sein.
- Beispiel
  - Ein Studio kann nur von einem Vorsitzenden geleitet werden.
  - Ein Vorsitzender kann nur ein Studio leiten.
- „Kann“: D.h. nicht jeder Entity muss mit einem anderen verbunden sein.
  - Ein Studio kann (vorübergehend) keinen Vorsitzenden haben.



## ■ [min,max]-Notation (Lookup-Semantik)

- Schränkt die möglichen Teilnahmen von Instanzen der beteiligten Entitytypen an der Beziehung ein.

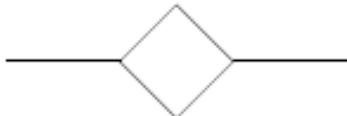
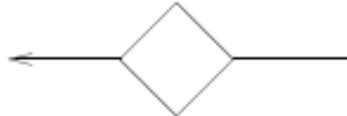
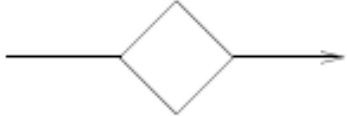
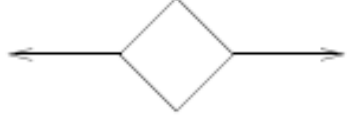
- Wie oft steht ein Entitytyp in Beziehung?



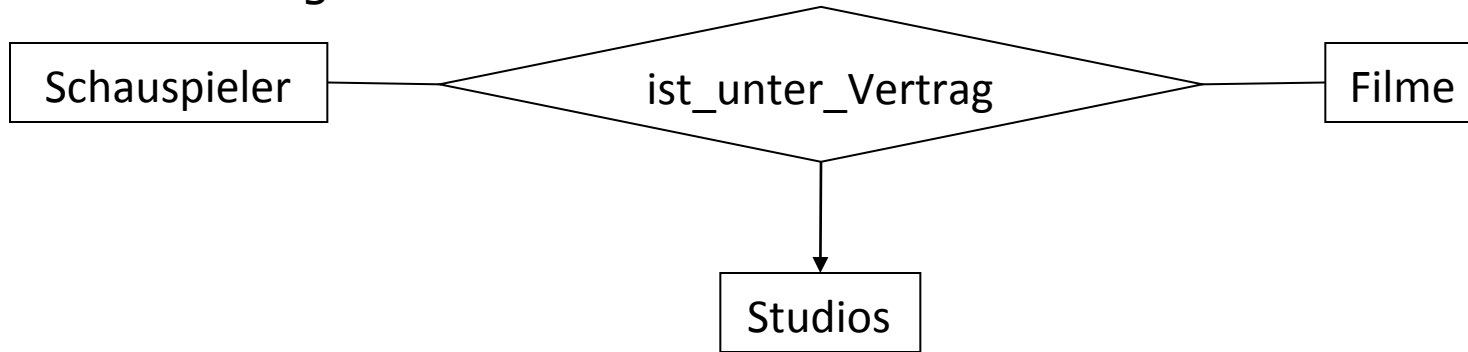
- (Ausverkaufte) Produkte sind keinem Regal zugeordnet.
- Ein Produkt kann höchstens in einem Regal gelagert werden.
- Ein Regal kann leer sein und maximal 3 Produkte Lagern.

## ■ Numerische Notation (Partizipationssemantik)



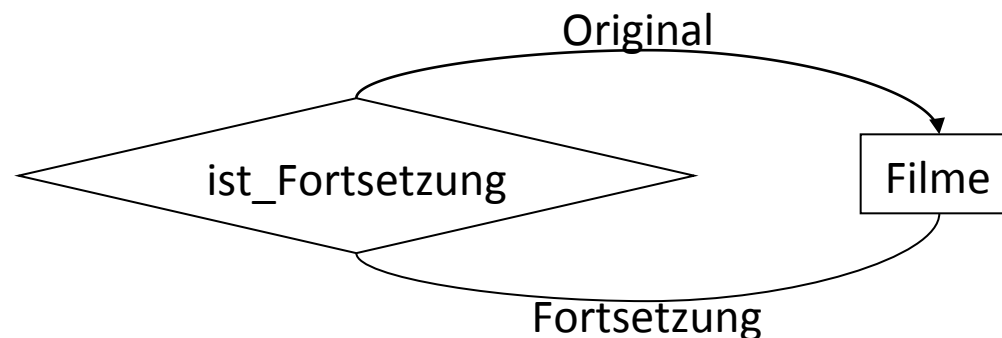
Beziehungsm.	(min1, max1)	(min2, max2)	grafische Notation
many-to-many	(0,*)	(0,*)	
one-to-many	(0,*)	(0,1)	
many-to-one	(0,1)	(0,*)	
one-to-one	(0,1)	(0,1)	

- Relationships zwischen mehr als zwei Entities
- Ein Schauspieler steht bei einem Studio für einen bestimmten Film unter Vertrag

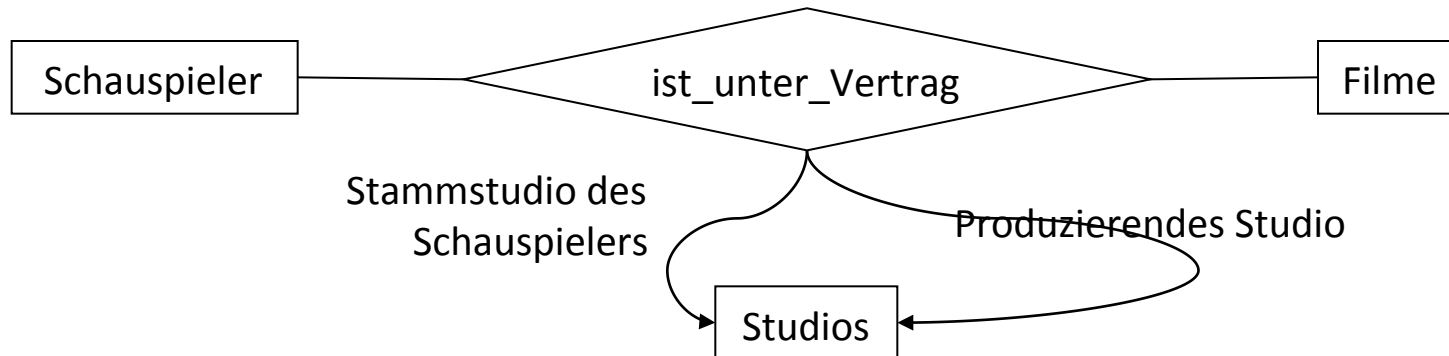


- Instanz der Relationship kann man als Tripel darstellen.
- Kardinalitäten: Jede Kombination von Schauspieler und Film kann nur mit einem Studio in Beziehung stehen.

- Entitytypen können mehr als einmal in einem Relationship-Typen auftauchen.
  - Entsprechend werden mehrere Kanten gemalt.
  - Jede Kante entspricht einer anderen Rolle.
  - Die Kanten werden mit den entsprechenden Rollen annotiert.
  - Bsp: Fortsetzungen von Filmen
    - Beverly Hills Cop (& II, III, ...)
    - Ein Film kann viele Fortsetzungen haben.
    - Eine Fortsetzung hat aber nur ein Original.



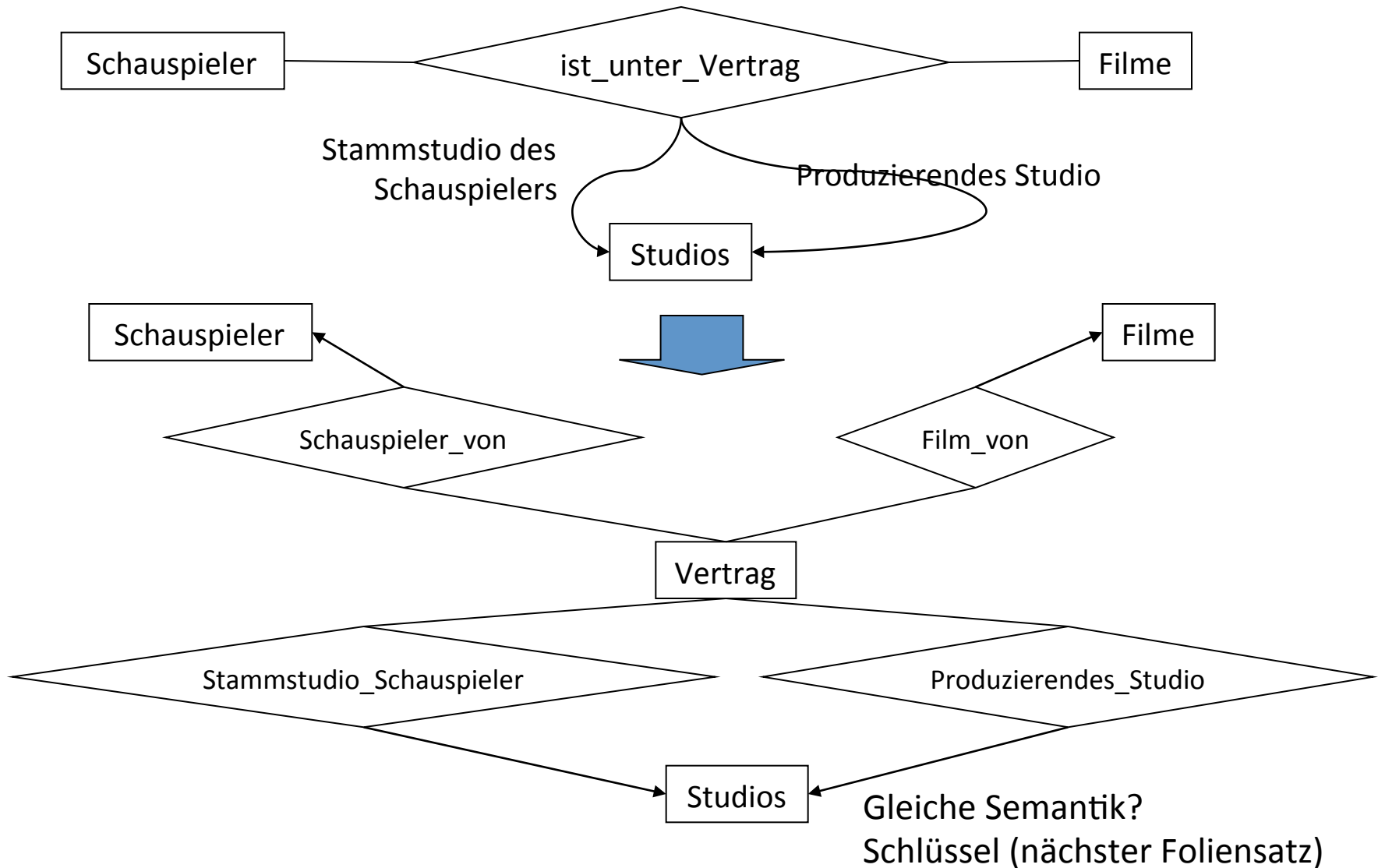
- Stammstudio eines Schauspielers erlaubt einem anderen Studio den Schauspieler für einen bestimmten Film auszuleihen



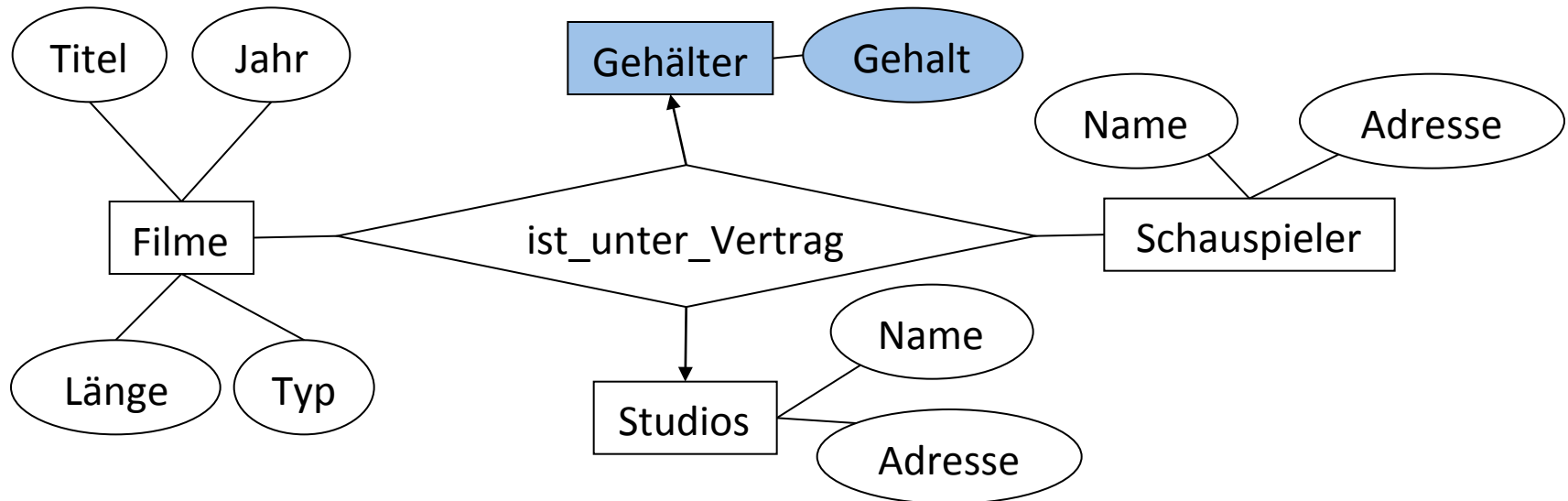
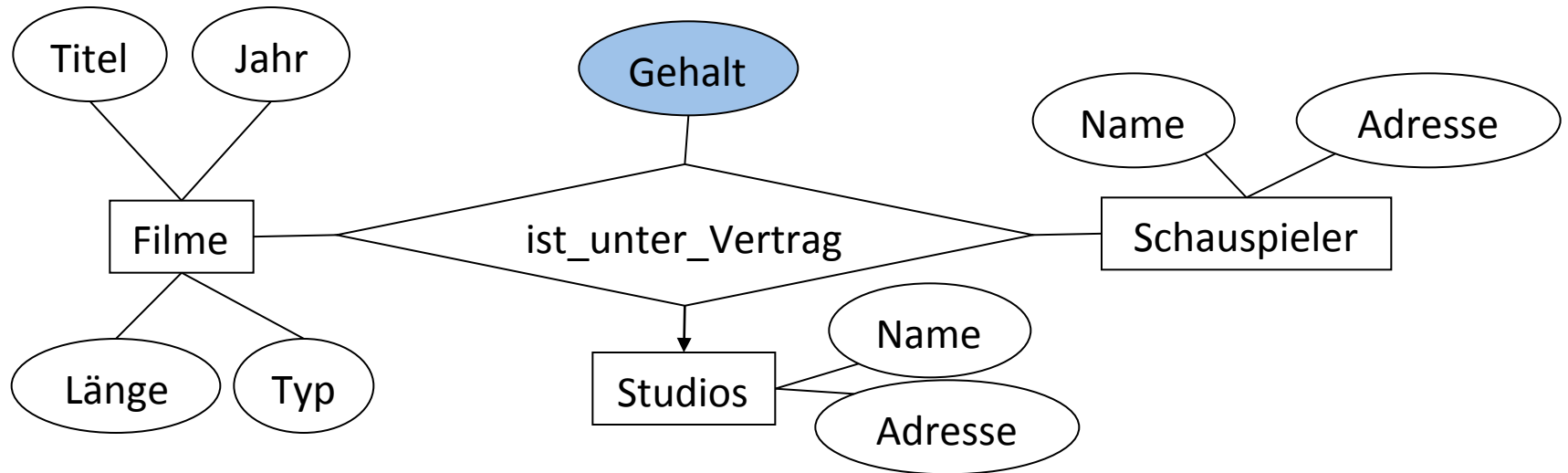
- Kardinalitäten:
  - Gegeben Schauspieler, Film und produzierendes Studio, ist das ausleihende Stammstudio eindeutig (höchstens ein Studio ist das ausleihende Stammstudio).
  - Gegeben Schauspieler, Film und Stammstudio ist das produzierende Studio eindeutig (höchstens ein Studio kann einen Film produzieren).
  - Gegeben Schauspieler, Stammstudio und produzierendes Studio könnte es mehrere Filme geben, die in dieser Konstellation gedreht werden.
  - Gegeben Film, Stammstudio und produzierendes Studio könnte es mehrere Schauspieler geben, die in dieser Konstellation ausgeliehen werden.



- Umwandlung n-ärer Relationships in binäre Relationships
  - Erstellung eines neuen, verbindenden Entitytyps
  - Neue n:1 Relationships zwischen dem neuen Entitytyp und den alten Entitytypen
  - Falls ein Entitytyp mehrere Rollen spielt, entsteht pro Rolle ein Relationshiptyp.
  - Attribute des Relationshiptyps werden an den neuen Entitytyp angehängt.

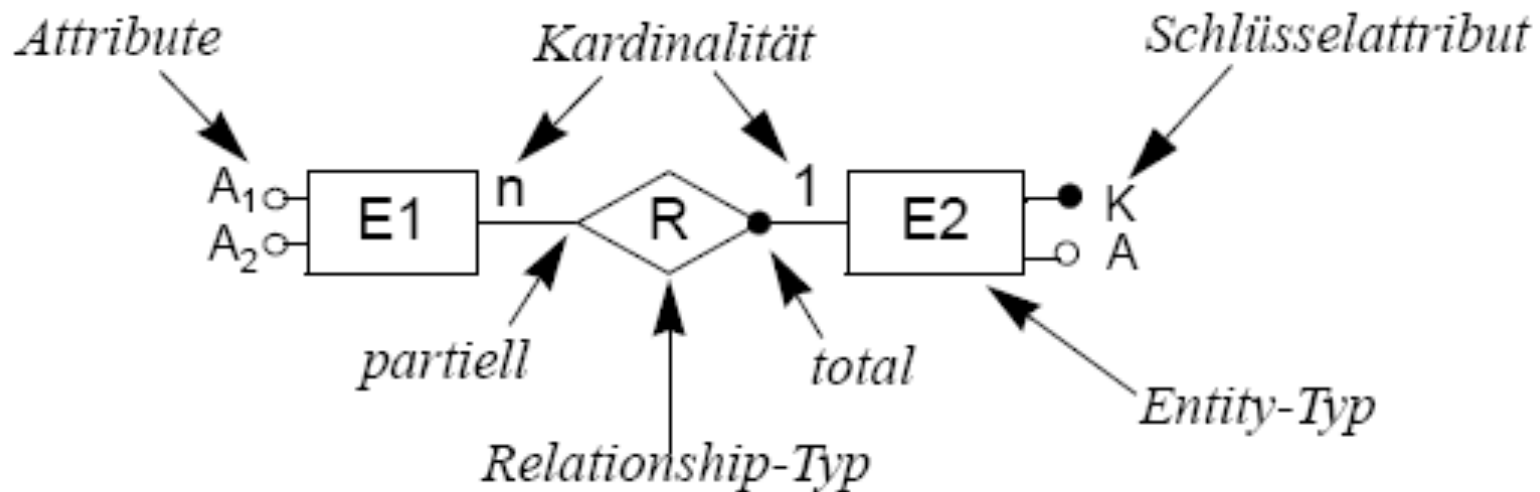


- In manchen Fällen ist es hilfreich, Relationship-Typen Attribute zuzuordnen.
  - Bsp: In dem Drehvertrag wird ein Gehalt festgestellt.
    - Zuordnung zu Schauspieler? Er könnte für verschiedene Filme unterschiedliche Gehälter bekommen.
    - Zuordnung zum Film? Verschiedene Schauspieler könnten unterschiedliche Gehälter bekommen.
    - Zuordnung zum Studio? Es könnte verschiedenen Schauspielern unterschiedliche Gehälter zahlen.



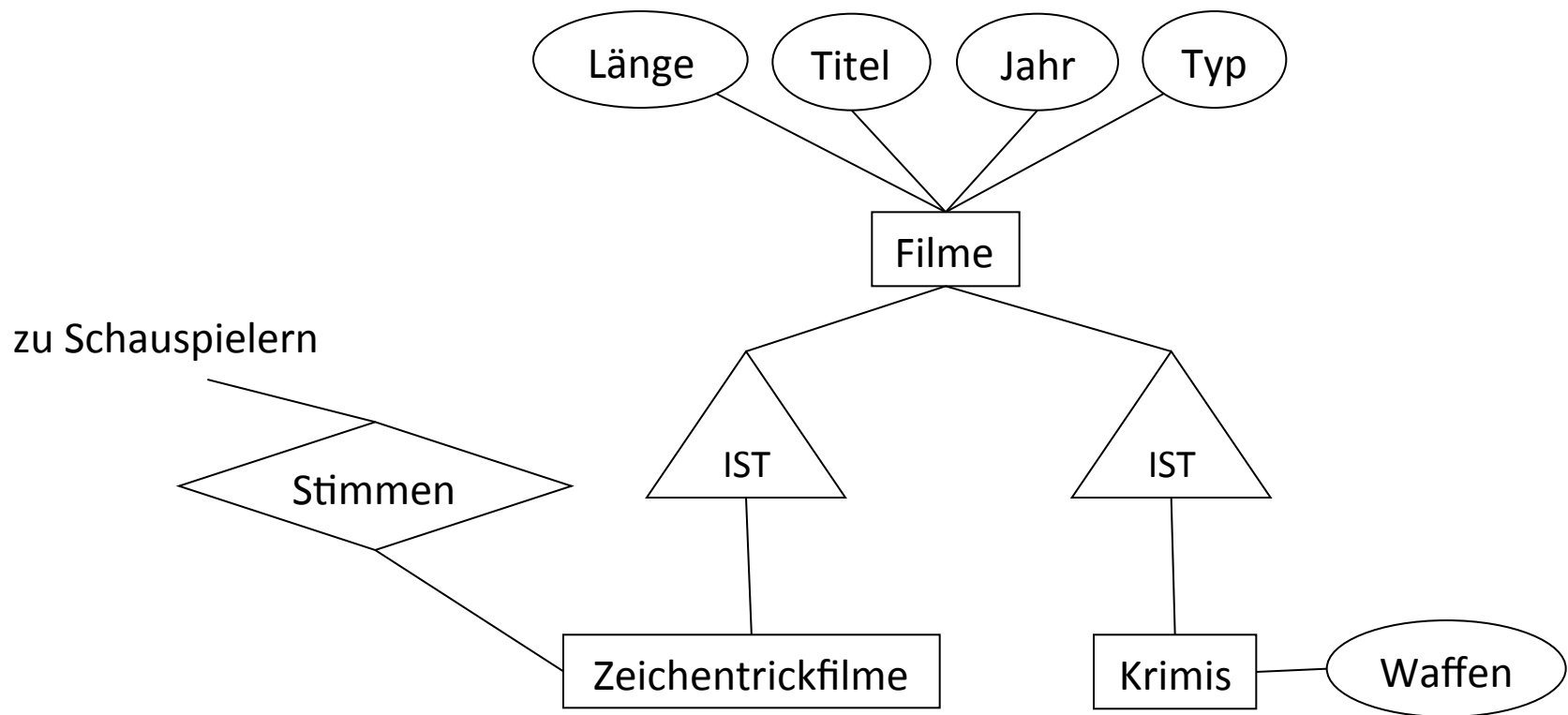
zuerst nur:

- Flache E/R-Diagramme (d.h. noch keine Abstraktionskonzepte) mit Entity-Typen, Relationship-Typen, Attributen und Integritätsbedingungen



- Subtyp / Subklasse
  - Spezialfall / Spezialisierung
  - Weniger Entities
  - Mehr Attribute
  - Evtl. mehr Relationships
  
- Besonderer Relationshiptyp ( -> Abstraktionskonzepte )
  - IST / IST-EIN (is-a)
  - Darstellung durch Dreieck
    - Spitze zeigt zur Generalisierung
  - Immer 1:1
    - Trotzdem keine Pfeile

- Subklassen zu Filmen
  - Zeichentrickfilme
  - Krimis

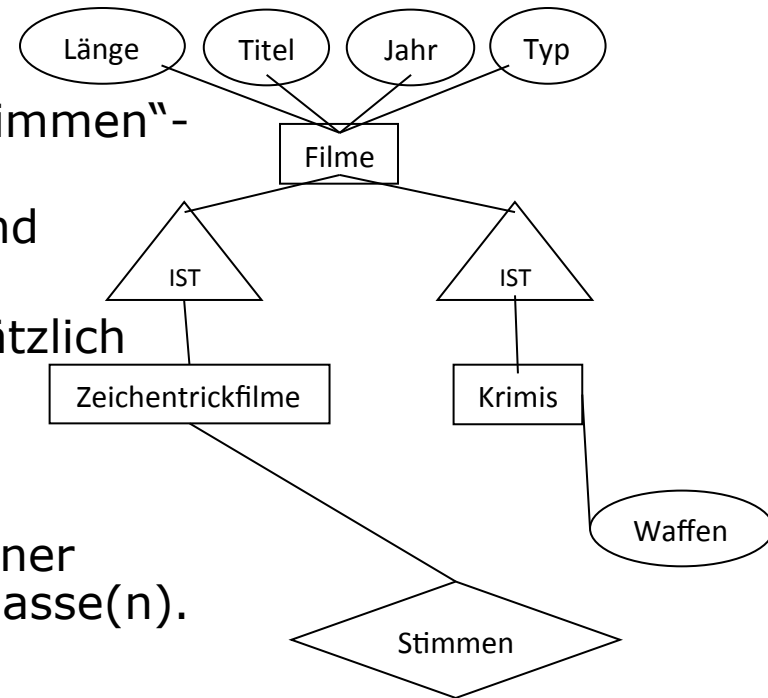


- IST-Beziehungen nur als Bäume
  - Keine Mehrfachvererbung
- Ein Entity kann aus mehreren Komponenten des IST-Baumes bestehen.

- „Krieg der Sterne“ hat vier Attribute.
- „Cinderella“ hat vier Attribute und „Stimmen“-Relationships.
- „Der dritte Mann“ hat vier Attribute und zusätzlich das Attribut „Waffen“.
- „Roger Rabbit“ hat vier Attribute, zusätzlich das Attribut „Waffen“ und „Stimmen“-Relationships.

- Anders als objektorientierte Modelle

- In OO sind Objekte immer in genau einer Klasse; Subklassen erben von Superklasse(n).
- In ER sind Entities in allen Subklassen repräsentiert in die sie gehören.
- In ER ist ein Entity in einer Subklasse auch automatisch in den Superklassen repräsentiert.

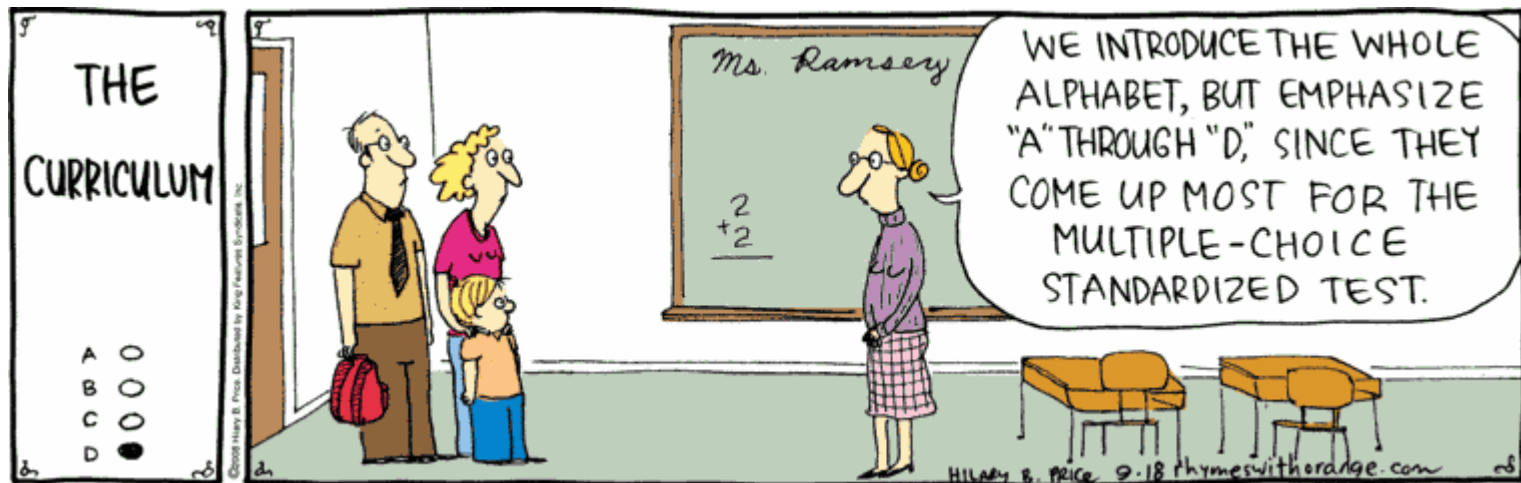




- Motivation und Einbettung
- Begriffe und Definitionen
- ER-Diagramme
- **Modellierung von Nebenbedingungen**
- Schwache Entitytypen



- Bitte erstellen Sie eine Multiple Choice Aufgabe zum Thema E/R Modellierung
  - Formulieren Sie eine Frage und 3 Antworten (A, B, C)
  - Davon sollte mindestens eine Antwort richtig und mindestens eine Antwort falsch sein
- Geben Sie die Aufgaben an Ihren rechten Nachbarn. Diskutieren Sie gemeinsam und markieren Sie die richtigen Lösungen
- Geben Sie am Ende der Vorlesung Ihre Aufgabe bei mir ab



- Schlüssel
  - Ein oder mehrere Attribute
  - Werte identifizieren eindeutig ein Entity.
  
- Referentielle Integrität
  - Existenz des referenzierten Entities
  - Entspricht „dangling pointer“
  
- Domänen
  - Einschränkung des Wertebereichs
  
- Allgemeine Nebenbedingungen (Assertions)
  - Z.B. nicht mehr als 10 Schauspieler pro Film
  
- Nebenbedingungen sind Teil des Schemas. Sie leiten sich nicht aus den Daten ab!

- Ein Schlüssel ist eine (minimale) Menge von Attributen eines Entitytyps, für die gilt, dass keine zwei Entities gleiche Werte in allen Schlüsselattributen haben.
  - Einige Attributwerte können übereinstimmen.
  - Oft nur ein Attribut
- Für jeden Entitytyp muss ein Schlüssel angegeben werde.
- Es kann mehr als einen Schlüssel für einen Entitytyp geben.
  - Üblich: Primärschlüssel auswählen
- Bei IST-Beziehungen muss die Wurzel-Superklasse sämtliche Schlüsselattribute enthalten.
- Darstellung durch Unterstreichen der Attributnamen

- Filme
  - „Titel“ als Schlüssel?
    - King Kong
  - Titel und Jahr als Schlüssel?
    - Eventuell kann ein Film dann nicht gespeichert werden.
    - In IMDB: 275 Doppelte
- Schauspieler
  - „Name“ oder „Name“ und „Adresse“
- Studio
  - Name
- In der Praxis: Modellierung eines speziellen, numerischen Schlüsselattributs.
  - ISBN, SSN, Imma-Nummer,...

Steuerzentrale - DB2COPY1

Steuerzentrale Ausgewählt Editieren Sicht Tools Hilfe

Objektsicht Befehlseditor 1

Befehle Abfrageergebnisse Zugriffsplan

Diese Ergebnisse werden mit Hilfe von Aktualisierungen und Löschungen mit Su...

TITLE	DATE	3
Shadow Chasers	1985	2
Shogun	1980	2
Shriners Hospit...	2000	2
Sinatra: The Cl...	2002	2
Sirens	1995	2
Skag	1980	2
Skin Deep	2000	2
Skin Deep	2003	2
Skyport	1960	2
Something Is D...	1988	2
Space Invaders	1999	2
Sparks	1998	2
Special Delivery	2003	2
Spring	2002	2
Srečna porodica	1979	2
Sternbergs - Är...	1999	2
Still Life	2004	2
Straight Up	1997	2
Strange Days	1998	2
Strange World	1999	2
Striker	1976	2
Studio 5-B	1989	2
Summerland	2004	2
Sunset Beat	1990	2
Supercarrier	1988	2
Sweet Heart	1996	2
Take Five	1987	2
Talk to Me	2000	2
Tarzan: The E...	1996	2
Tæskeholdet	1997	2
Temptation	1968	2
Tenshi no tama...	1985	2
Tenspeed and ...	1980	2
Test The	2001	2

Star Wars (1977) - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.imdb.com/title/tt0076759/

Star Wars (1977)

IMDb

Earth's Biggest Movie Database™

Home | Top Movies | Photos | Independent Film | Browse | Help

Search the IMDb

All

go

Result: 1 of 108

[ < x > ? ]

[More searches](#) | [Tips](#)

[IMDbPro.com free trial](#)

Showing page 1 of 39

Overview

- main details
- combined details
- full cast and crew
- company credits

Awards & Reviews

- user comments
- external reviews
- newsgroup reviews
- awards & nominations
- user ratings
- recommendations
- message board

Plot & Quotes

- plot summary
- plot keywords
- Amazon.com summary
- memorable quotes

**Star Wars (1977)**

Directed by [George Lucas](#)

Writing credits [George Lucas](#) (written by)

[Add to MyMovies](#) [Photo Gallery](#) [IMDbPro Details](#)

**Photo Gallery (35 photos)**

[more](#)

Genre: [Action](#) / [Adventure](#) / [Fantasy](#) / [Sci-Fi](#) (more)

Tagline: A long time ago in a galaxy far, far away... ([more](#))

SHOP STAR WARS

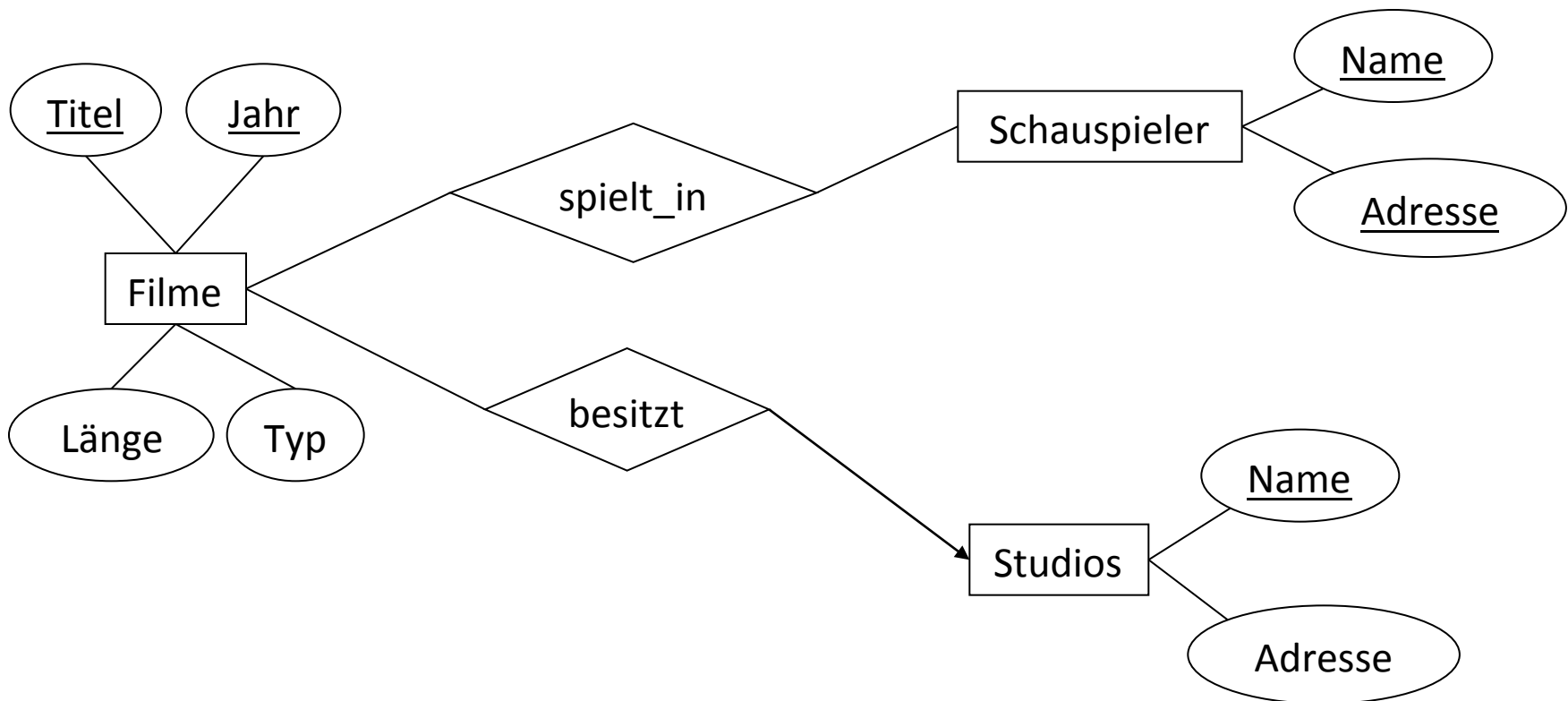
[DVD](#) [VHS](#) [CD](#)

[amazon.com](#)

Fertig

2.513s Adblock

- In ER-Diagrammen kann nur ein Schlüssel notiert werden.



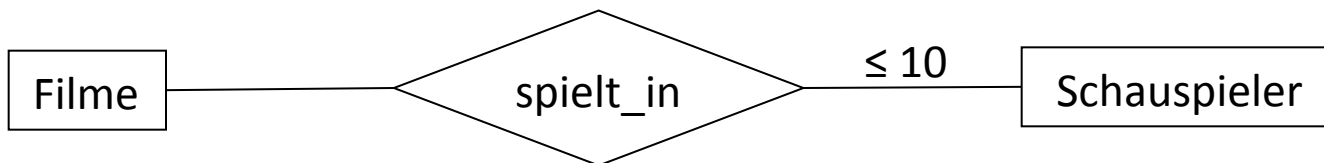
- Schlüssel: Höchstens ein bestimmter Wert für ein Attribut
- Referentielle Integrität: Genau ein bestimmter Wert
- Bsp. n:1 Relationship zwischen „Filme“ und „Studios“
  - Ein Film kann zu höchstens einem Studio gehören.
  - Aber ein Film muss zu keinem Studio gehören.
  - Auch wenn ein Film zu einem Studio gehört, muss dieses nicht in der DB repräsentiert sein.
  - Referentielle Integrität erzwingt die Existenz und Repräsentation des Studios
- „Erzwingen“
  - Bei Einfügen/Ändern eines Films muss entsprechendes Studio vorhanden sein.
  - Ein Studio darf nicht gelöscht werden, solange es noch Filme besitzt.
  - Oder: Wenn ein Studio gelöscht wird, werden auch alle entsprechenden Filme gelöscht.
  - Verschiedene Einstellungen im DBMS





- Jeder Film muss von genau einem Studio besessen werden.
- Jeder Vorsitzende muss genau ein Studio leiten.
- Jedes Studio wird von höchstens einem Vorsitzenden geleitet.
  - Eventuell aber von keinem

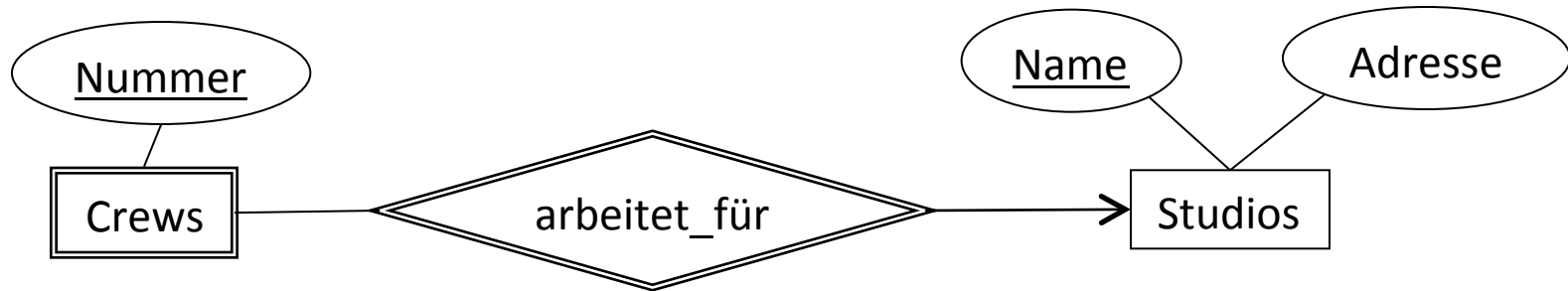
- Ohne formale Notation im ER-Diagramm
  - Datentyp
    - Integer, String, ...
  - Wertebereich / Domäne
    - $\leq 100$ , {Krimi,Doku,Zeichentrick}
  - Länge eines Attributes
    - Stringlänge  $< 25$
  - Kardinalität von Relationships
    - Höchstens 10 Schauspieler pro Film



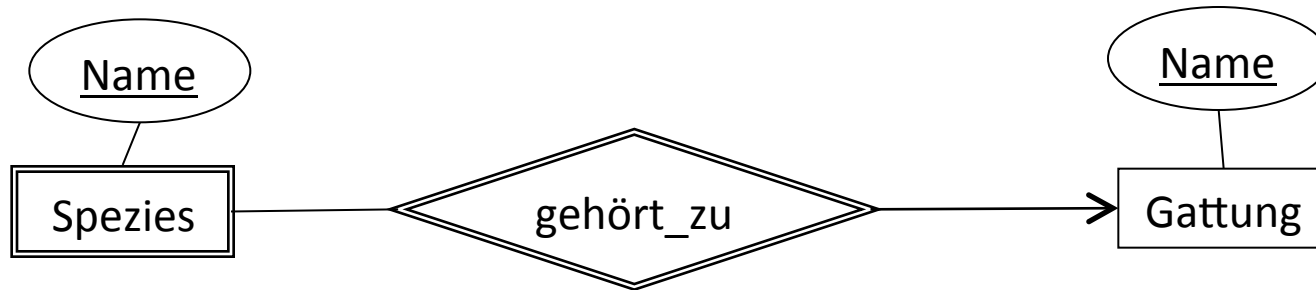
- Motivation und Einbettung
- Begriffe und Definitionen
- ER-Diagramme
- Modellierung von Nebenbedingungen
- **Schwache Entitytypen**



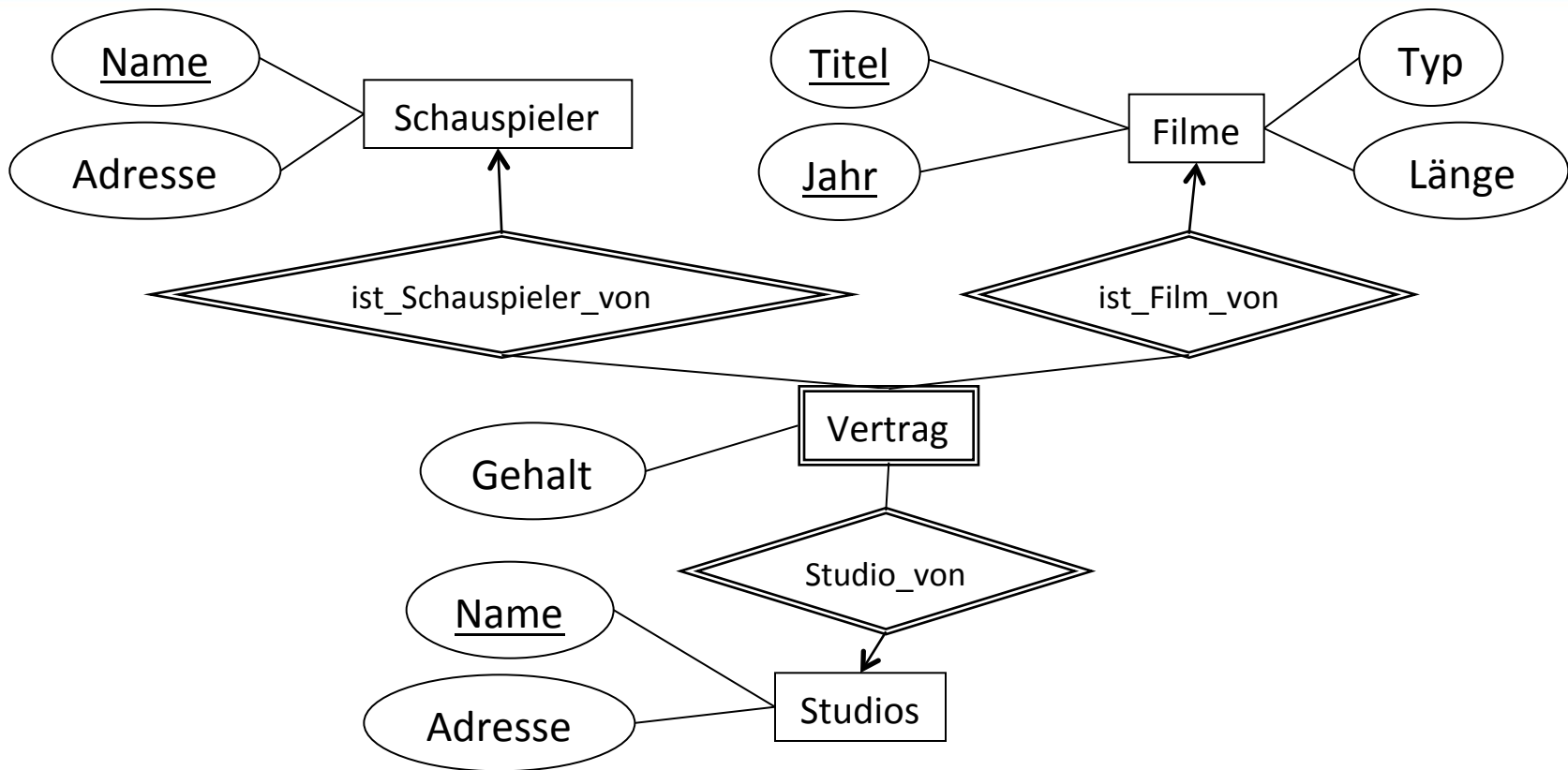
- In bestimmten Situationen können Entities nicht allein anhand ihrer Attribute identifiziert werden:
  1. Falls sie in eine nicht-IST-Hierarchie fallen.
  2. Entities, die zur Eliminierung n-ärer Relationships erschaffen wurden.
- Ein Entitytyp ist schwach wenn es zur eindeutigen Identifizierung eines Entities nötig ist, eine oder mehr n:1 Relationships zu folgen und den Schlüssel der verwandten Entities hinzuzunehmen.



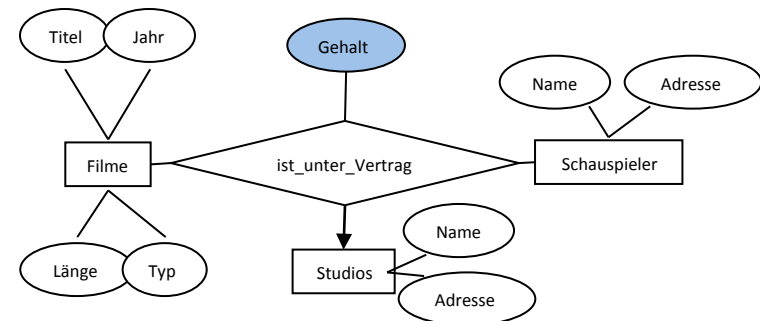
- Ein Studio beschäftigt mehrere Filmcrews.
- Filmcrews werden mit einer Nummer versehen.
- Verschiedene Studios könnten eigene Crews mit gleichen Nummern beschäftigen.
  - Nummer ist also kein Schlüssel
- Nimmt man den Schlüssel der Studios hinzu ist eine eindeutige Identifizierung möglich.



- Eine Spezies ist definiert durch den Namen der Gattung und des Spezies.
  - Gattung: homo
  - Spezies: homo sapiens



- Fall 2: Auflösung einer ternären Relationship
- Vertrag hat kein Attribut, das Teil des Schlüssels ist.

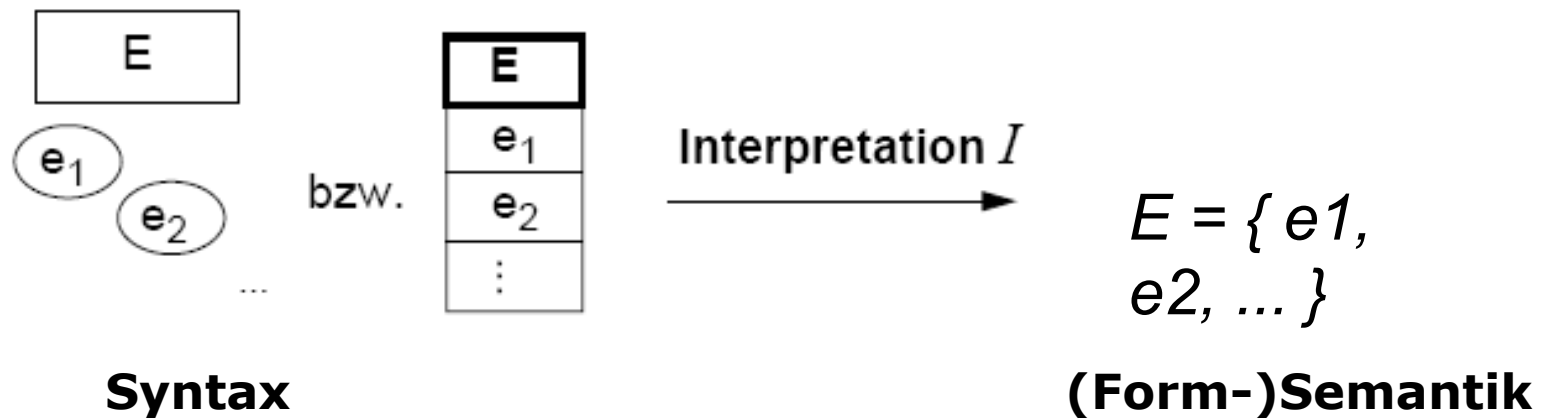


- Falls E ein schwacher Entitytyp ist, besteht sein Schlüssel aus...
  - ... null oder mehr eigenen Attributen
  - ... und den Schlüsselattributen von Entitytypen, die über bestimmte n:1 Relationshiptypen, den „unterstützenden Relationshiptypen“ erreicht werden können.
    - supporting relationships
- Unterstützende Relationshiptypen
  - n:1 vom schwachen Entitytypen zu einem anderen Entitytypen
  - Es muss referentielle Integrität gelten.
  - Falls referenzierter Entitytyp wiederum schwach ist, werden (rekursiv) weitere Schlüsselattribute übernommen.



- Mathematische Formalisierung, d.h. (Form-) Semantik oder Interpretation der Konzepte der E/R-Modellierung als auf eine sichere Grundlage unserer (graphischen) Modellierungstechnik mit den Konzepten:
  - Entity bzw. Entity-Typ,
  - Relationship bzw. Relationship-Typ
  - Attribut und Schlüsselattributsowie
  - Integritätsbedingungen
- Hierzu reichen im wesentlichen die allseits bekannten Grundbegriffe Menge, Relation, Abbildung sowie deren Eigenschaften und etwas Prädikatenlogik aus.

- Ein Entity-Typ  $E$  wird mathematisch interpretiert als eine Menge  $\mathcal{E}$  von Elementen  $e_1, e_2, \dots$ , die ihrerseits die Interpretation der zu  $E$  gehörigen Entities  $e_1, e_2, \dots$  sind, d.h. der in einem modellierten Diskursbereich relevanten Informationselemente.



- Ein  $n$ -ärer Relationship-Typ  $R$  zwischen den Entity-Typen  $E_1, E_2, \dots, E_n$  — kurz geschrieben als:  $R(E_1, E_2, \dots, E_n)$  — wird interpretiert als

$n$ -stellige Relation  $R \subseteq E_1 \times E_2 \times \dots \times E_n$ ,

wobei die  $E_i$  die jeweiligen Interpretationen der  $E_i$  ( $i=1, \dots, n$ ) seien.

- Eine einzelne Relationship  $r$  des Typs  $R(E_1, E_2, \dots, E_n)$  ist demzufolge zu interpretieren als ein

$n$ -Tupel  $(e_1, e_2, \dots, e_n) \in R$ ,

wobei die  $e_i$  hier natürlich die Interpretationen der jeweiligen Entities  $e_i$  sind.

- Ein Entity-Attribut  $a$  zu einem Entity-Typ  $E$  wird interpretiert als Abbildung  $a : E \rightarrow \text{dom}(a)$  ,

wobei  $\text{dom}(a)$  den Wertebereich des Attributs  $a$  bezeichne und  $E$  die Interpretation von  $E$  sei.

Typische Wertebereiche sind Zahlenräume wie die natürlichen Zahlen  $\mathbb{N}$ , die ganzen Zahlen  $\mathbb{Z}$  oder Teilmengen davon sowie die Menge aller über einem Alphabet  $\Sigma$  bildbaren Zeichenketten  $\Sigma^*$  (bzw. nicht-leeren Zeichenketten  $\Sigma^+$ ) oder Teilmengen davon.

- Ein Relationship-Attribut  $a$  zu einem Relationship-Typ  $R(E_1, E_2, \dots, E_n)$  wird analog zu den Entity-Attributen interpretiert als Abbildung  $a : E_1 \times E_2 \times \dots \times E_n \rightarrow \text{dom}(a)$  .
- Ist  $a$  ein Schlüsselattribut zum Entity-Typ  $E$  , so ist seine Interpretation  $a$  eine *injektive* Abbildung, d.h. zwei verschiedene Entities besitzen stets auch verschiedene Attributwerte, in Formeln:

$$\forall e_1, e_2 \in E : e_1 \neq e_2 \Rightarrow a(e_1) \neq a(e_2) .$$

- Zu den Integritätsbedingungen werden logische Formeln angegeben, deren klassische Interpretation in der Prädikatenlogik gerade die Semantik des entsprechenden Konzepts angibt.

- Kardinalitäten

- 1:n – Kardinalität, 1:1 – Kardinalität

Die Integritätsbedingung 1:n für einen Relationship-Typ  $R(E1, E2)$  bedeutet, dass folgende Formel erfüllt sein muss:

$$\forall e1, e2 \in E1, e \in E2 : (e1, e) \in R \wedge (e2, e) \in R \Rightarrow e1 = e2$$

Der Fall n:1 ist hierzu genau symmetrisch, der Fall 1:1 erfordert das gleichzeitige Erfülltsein beider Bedingungen, d.h. die logische Konjunktion beider Formeln.

- Totalität

Die Formalisierung der Integritätsbedingung links-total eines Relationship-Typs  $R(E1, E2)$  wird durch folgende Formel geleistet (die Fälle rechts-total bzw. bi-total analog):

$$\forall e1 \in E1 \exists e2 \in E2 : (e1, e2) \in R$$

- Motivation und Einbettung
- Begriffe und Definitionen
- ER-Diagramme
- Modellierung von Nebenbedingungen
- Schwache Entitytypen
- In den Tutorien
  - Erweitertes ER-Modell
  - Designprinzipien
  - Sichtintegration

In der nächsten Veranstaltung:

- Relationaler Entwurf  
(Kapitel 2 und 3 des Lehrbuches)
  - Das Relationale Modell
  - Von ER-Diagrammen zu Relationenschemata
  - Konvertierung von Spezialisierung
  - Funktionale Abhängigkeiten (FDs)
  - Ableitungsregeln für FDs
  - Normalformen

